

Computational Development of the Strategies to Explore Molecular Machines and the Molecular Space for Desired Properties using Machine Learning

by

Ghule Siddharth Sambhaji
10CC17A26010

A thesis submitted to the
Academy of Scientific & Innovative Research
for the award of the degree of
DOCTOR OF PHILOSOPHY
in
SCIENCE

Under the supervision of
Dr. Kumar Vanka



CSIR-National Chemical Laboratory, Pune



Academy of Scientific and Innovative Research
AcSIR Headquarters, CSIR-HRDC campus
Sector 19, Kamla Nehru Nagar, Ghaziabad,
U.P. – 201 002, India

May 2022

Certificate

This is to certify that the work incorporated in this Ph.D. thesis entitled, “Computational Development of the Strategies to Explore Molecular Machines and the Molecular Space for Desired Properties using Machine Learning”, submitted by Ghule Siddharth Sambhaji to the Academy of Scientific and Innovative Research (AcSIR) in fulfillment of the requirements for the award of the Degree of Doctor of Philosophy In Science, embodies original research work carried-out by the student. We, further certify that this work has not been submitted to any other University or Institution in part or full for the award of any degree or diploma. Research material(s) obtained from other source(s) and used in this research work has/have been duly acknowledged in the thesis. Image(s), illustration(s), figure(s), table(s) etc., used in the thesis from other source(s), have also been duly cited and acknowledged.



(Signature of Student)

Name: Ghule Siddharth
Sambhaji

Date: 11-05-2022

(Signature of Co-Supervisor)

No



(Signature of Supervisor)

Name: Dr. Kumar Vanka

Date: 11-05-2022

STATEMENTS OF ACADEMIC INTEGRITY

I Ghule Siddharth Sambhaji, a Ph.D. student of the Academy of Scientific and Innovative Research (AcSIR) with Registration No. 10CC17A26010 hereby undertake that, the thesis entitled “Computational Development of the Strategies to Explore Molecular Machines and the Molecular Space for Desired Properties using Machine Learning” has been prepared by me and that the document reports original work carried out by me and is free of any plagiarism in compliance with the UGC Regulations on “*Promotion of Academic Integrity and Prevention of Plagiarism in Higher Educational Institutions (2018)*” and the CSIR Guidelines for “*Ethics in Research and in Governance (2020)*”.



Signature of the Student

Date : 11-05-2022

Place : Pune

It is hereby certified that the work done by the student, under my/our supervision, is plagiarism-free in accordance with the UGC Regulations on “*Promotion of Academic Integrity and Prevention of Plagiarism in Higher Educational Institutions (2018)*” and the CSIR Guidelines for “*Ethics in Research and in Governance (2020)*”.



Signature of the Co-supervisor

No

Signature of the Supervisor

Name : Dr. Kumar Vanka

Date : 11-05-2022

Place : Pune

Dedicated to My Family

Acknowledgment

Like an adventure, my PhD life was full of surprises, up, and downs. I ventured into unknowns and discovered treasures. I met amazing people along the way. Now, end in sight, I feel deeply indebted to those who have inspired and supported me during my graduate study.

First, I would like to express my deep gratitude to my research supervisor, Dr. Kumar Vanka, for his constant guidance and support. Without him, it would have been impossible to reach this point. Furthermore, I am also grateful for the academic freedom he has given me, which motivated me to grow as an independent researcher. Also, his teaching, writing, and communication skills have been inspirational to me throughout my doctoral research.

I would also thank my present Doctoral Advisory Committee members, Dr. Nayana Vaval, Dr. Dr. Kavita Joshi, and DAC chairperson Dr. Paresh Dhepe for their insightful suggestions and feedback. I am also grateful to Dr. Ashish Lele, director, CSIR-NCL, and Dr. Ashwini Kumar Nangia, former director of CSIR-NCL. Moreover, I take the opportunity to thank the present and former heads of the Physical and Materials Chemistry Division for their support and providing all facilities during my PhD.

I want to extend my gratitude to all teachers who had taught me during my PhD course work at CSIR-NCL: Dr. Kumar Vanka, Dr. Syan Bagchi, Dr. Leelavati Narlikar, Dr. Kavita Joshi, and Dr. T.G. Ajithkumar, as well as other scientists in NCL. Let me extend my warm thanks to my research collaborators Dr. Leelavati Narlikar, Dr. Syan Bagchi, and Dr. Kavita Joshi. Moreover, I owe to thank my University and school teachers for their support and motivation.

I also acknowledge all the non-academic staff of CSIR-NCL, and AcSIR, for their support and help during my work. Without the funding, this PhD journey would not have been possible; hence I would like to express my gratitude to the Council of Scientific & Industrial Research (CSIR) for the fellowship. Moreover, I want to thank the whole scientific community for being a source of inspiration and motivation.

No words are enough to thank my friend, Indranil, who has helped and supported me at various stages. I also extend my thanks to Tamal, Vipin, and Shailja for their help whenever needed during my PhD. Also, my special thanks to Soumya for helping me a lot. Moreover, it is my pleasure to thank my past and present lab mates; Jugal, Yuvraj, Mrityunjay, Vipin, Shailja, Ruchi, Anagh, Subhrashis, Himanshu, Priyam, and Soumya.

I want to thank the most important people in my life, my family. No words are sufficient to describe their love, affection, and support. I would like to thank my all family members for their mental and emotional support. I consider myself blessed to have such a beautiful family around me.

Finally, I express my gratitude to the Almighty for the blessings.

- **Ghule Siddharth Sambhaji**

Table of Contents

Abbreviations.....	i
Physical Constants	iii
Chapter 1: Brief Introduction to the Molecular Space and Strategies for its Exploration	2
1.1 Importance of the Scientific Discoveries and Their Connection to Molecules	3
1.2 Molecular Space and Motivation for its Exploration	4
1.3 Conventional Strategies for the Exploration of Molecular Space.....	5
1.3.1 Experimental Approaches	5
1.3.2 Computational Approaches.....	6
1.3.3 Algorithmic Approaches	7
1.4 Motivation for the Development of the Exploration Strategies Based on Machine Learning Algorithms	8
1.5 Redox Flow Batteries (RFBs)	10
1.6 Transcription Regulation.....	11
1.7 Molecular Machines	12
1.8 Statement of problem (aims & objectives).....	14
1.9 Organization of the Thesis	16
1.10 References	18
Chapter 2: Fundamentals of Machine Learning.....	30
2.1 Introduction	31
2.2 Brief History of Machine Learning	33
2.3 Types of Machine Learning Problems	34
2.3.1 Classification.....	34
2.3.2 Regression.....	34
2.3.3 Clustering	34
2.4 Types of Machine Learning Algorithms	35
2.4.1 Supervised Learning Algorithms	35
2.4.2 Unsupervised Learning Algorithms	35
2.4.3 Reinforcement Learning Algorithm.....	35
2.5 Machine Learning Workflow	36
2.5.1 Data Collection	36
2.5.2 Data Processing.....	36
2.5.3 Feature Engineering	38
2.5.4 Model Training	38

2.5.5	Model Evaluation / Deployment	39
2.5.6	Model Selection	40
2.6	Brief Description of Commonly Used Machine Learning Algorithms	41
2.6.1	Linear Regression	41
2.6.2	Ridge Regression	42
2.6.3	Lasso Regression	43
2.6.4	Logistic Regression.....	44
2.6.5	Naïve Bayes	45
2.6.6	Support Vector Machines	46
2.6.7	Support Vector Regression	48
2.6.8	Decision Trees	48
2.6.9	Random Forests	50
2.6.10	Artificial Neural Networks	50
2.6.11	Automatic Relevance Determination Regression	51
2.6.12	Gaussian Process Regression.....	51
2.6.13	Kernel Ridge Regression	51
2.6.14	K-Means Algorithm	51
2.6.15	Principal Component Analysis	52
2.7	Brief Description of Modern Machine Learning Methods.....	52
2.7.1	Reinforcement Learning	52
2.7.2	Recurrent Neural Networks	53
2.7.3	Convolutional Neural Networks	54
2.7.4	Variational Autoencoders	55
2.7.5	Generative Adversarial Networks.....	55
2.8	Computational Methods	56
2.9	References	58
Chapter 3: Machine Learning the Redox Potentials of Phenazine Derivatives: A Comparative Study on Molecular Features		
3.1	Introduction	67
3.2	Materials and Methods	70
3.2.1	Dataset.....	70
3.2.2	Feature Generation.....	70
3.2.3	Machine Learning Models	72
3.2.4	Hyperparameter Tuning	73
3.2.5	Evaluation Metrics	73

3.2.6	MSE and MAE Threshold	74
3.2.7	K-Fold Cross-Validation.....	74
3.2.8	Feature Importance Score	75
3.2.9	Pipeline	75
3.3	Results and Discussion.....	77
3.3.1	Analysis of the Best-Performing Models.....	77
3.3.2	Assessment of Model Performance on Four Feature Sets	78
3.3.3	Cross-Validation and Out-of-Sample Performance	79
3.3.4	Feature Importance Analysis	82
3.3.5	Effect of Feature Size on Model Performance.....	86
3.3.6	Assessment of Model Performance on Limited Number of Features.....	87
3.3.7	Analysis of the Predictive Performance with respect to Individual Functional Groups	90
3.3.8	Error Analysis	91
3.4	Conclusions	93
3.5	References	95
Chapter 4: Predicting the Redox Potentials of Phenazine Derivatives Using a Hybrid DFT-ML Approach.....		102
4.1	Introduction	103
4.2	Materials and Methods	105
4.2.1	Computational Details	105
4.2.2	Data Generation	105
4.2.3	Hyperparameter Optimization	107
4.2.4	Machine Learning Models	108
4.2.5	Evaluation Metrics	109
4.2.6	Feature Selection.....	109
4.2.7	Feature Importance Analysis	110
4.3	Results and Discussion.....	111
4.3.1	Test-set Performance	111
4.3.2	Prediction on Multiple Functional Group Test-sets.....	112
4.3.3	Feature Importance Analysis	114
4.3.4	Structure–Functional Relationship.....	117
4.3.5	Identification of the Promising Phenazine Derivatives for Analyte	121
4.4	Conclusions	123
4.5	References	124

Chapter 5: Investigating Combinatorial Binding of Transcription Factors using Unsupervised Machine Learning Models	128
5.1 Introduction	129
5.2 Materials and Methods	133
5.2.1 Data Generation	133
5.2.2 Models.....	133
5.3 Results and Discussion.....	141
5.3.1 Selecting the Appropriate Normalization Method.....	141
5.3.2 Employing LDA to Cluster Regions from Simulated and ChIP-seq Datasets.	142
5.3.3 Employing HDP to Cluster Regions from Simulated and ChIP-seq Datasets.	145
5.3.4 Employing NPLB to Cluster Regions from Simulated and ChIP-seq Datasets	148
5.3.5 Applying LDA, HDP, and NPLB to Cluster Regions from the DNase-seq Dataset	150
5.4 Conclusions	154
5.5 References	155
Chapter 6: An Algorithmic Development of the Strategy for Quantifying Rotational Motion in Molecular Machines	165
6.1 Introduction	166
6.2 Materials and Methods	168
6.2.1 Systems	168
6.2.2 Computational Details	168
6.2.3 Terminologies	168
6.3 Results and Discussion.....	170
6.3.1 Development of an Algorithm for Quantifying the Net Relative Rotation in Molecular Machines.....	170
6.3.2 Development of an Algorithm for Quantifying the Net Relative Translation in Molecular Machines.....	172
6.3.3 Verification of the Algorithm Developed for Quantifying Rotation	172
6.3.4 Verification of the Algorithm Developed for Quantifying Translation.....	175
6.3.5 Investigating Rotational and Translational Motion in Rotaxane	176
6.3.6 Investigating Issues with the Algorithm Developed for Quantifying Rotational Motion	179
6.3.7 Resolving Issues Related to the Rotation of Ring Atoms.....	183
6.3.8 Attempting to Resolve Issues Related to the Rotation of Track Atoms	188

6.3.9	Re-verification of the Improved Algorithm Developed for Quantifying Rotational Motion in Molecular Machines	193
6.3.10	Investigating Rotational Motion of Only the Ring in the Molecular Machine	195
6.4	Pseudocode of the algorithm developed for quantifying the net absolute rotation of the ring	202
6.5	Conclusions	203
6.6	References	204
Chapter 7: Summary and Future Outlook		208
7.1	Focus of this Thesis	208
7.2	Future Outlook	210
7.3	References	213
ABSTRACT		215
Details of the publications emanating from the thesis work		216

List of Figures

Figure 1.1. Visualization of molecular space along the axis of the desired property.	5
Figure 1.2. Timeline of the development of molecular space exploration strategies. The x-axis represents the time, whereas the y-axis represents the amount of statistical data required for each strategy.	8
Figure 2.1. Relationship between artificial intelligence, machine learning, deep learning, and data science.	32
Figure 2.2. A brief history of machine learning.	34
Figure 2.3. Types of machine learning problems.	35
Figure 2.4. Typical machine learning workflow.	36
Figure 2.5. Bias-variance trade-off.	40
Figure 2.6. Visualizing linear regression in two dimensions.	41
Figure 2.7. The logistic function.	45
Figure 2.8. Support Vector Machine.	47
Figure 2.9. Decision Tree.	48
Figure 2.10. Artificial Neural Network.	51
Figure 2.11. In reinforcement learning, the agent interacts with the environment through action, which causes the environment to transition to a new state and generate a reward.	53
Figure 2.12. Simple recurrent neural network.	54
Figure 2.13. Arrangement of layers in a typical CNN.	54
Figure 2.14. Structure of variational autoencoders.	55
Figure 2.15. Generative Adversarial Networks consist of a generator and a discriminator. The generator generates fake data, whereas, the discriminator tries to identify real data from fake data.	56
Figure 3.1. Schematic diagram of a typical redox flow battery.	68
Figure 3.2. Pictorial representation of the training and evaluation pipeline.	76
Figure 3.3. Machine learning prediction of redox potential (y-axis) vs. true redox potential (x-axis) of the three best-performing models in each feature set. The title of each plot indicates the model name, its rank, and the corresponding feature set used for training in brackets.	77

Figure 3.4. 10-Fold cross-validation performance of twenty models. Models were trained on all features from the corresponding feature set.	81
Figure 3.5. Test-set performance of twenty models. Models were trained on all features from the corresponding feature set.	82
Figure 3.6. Feature importance histograms of '2d' feature set.	84
Figure 3.7. Feature importance histograms of '3d' feature set.	84
Figure 3.8. Feature importance histograms of 'fp' feature set.	85
Figure 3.9. Feature importance histograms of '2d+3d+fp' feature set.	85
Figure 3.10. Model performance vs. number of features.	86
Figure 3.11. Test-set performance of twenty models trained on top-5, 10, 15, and 20 features from '2d+3d+fp' feature set. The top most important features were selected based on the random forest score. Full feature set performance is shown for reference.	88
Figure 3.12. Test-set performance of twenty models trained on the five most important features from the corresponding feature set. Features were selected based on the random forest score.	89
Figure 3.13. Functional group (FG) vs. Mean Absolute Percentage Error (MAPE) on the test-set. (a) MAPE is averaged over FG and feature sets. (b) MAPE is averaged over only FG. The test-set predictions were obtained from the corresponding best-performing model.	90
Figure 3.14. Distribution of functional groups in training and test sets.	91
Figure 3.15. Distribution of redox potential (a) of the whole dataset. (b) of -CN, -NO ₂ functional groups in training-set.	92
Figure 3.16. MAPE vs. redox potential. The final MAPE on the y-axis was calculated by averaging the MAPE obtained from the best-performing model in each feature set. The red curve depicts the normalized distribution of redox potential (i.e., density) for the whole dataset.	92
Figure 4.1. Plots showing machine learning predictions. ML predictions (y-axis) vs. DFT redox potentials (x-axis) on (a) internal test-set, (b) external test-set. The gray dashed line corresponds to the perfect predictions.	103
Figure 4.2. Plots showing machine learning predictions on internal test-set (y-axis) vs. DFT redox potentials (x-axis). Gray dash line corresponds to the perfect predictions.	111
Figure 4.3. Plots showing machine learning predictions on external test-set (y-axis) vs. DFT redox potentials (x-axis). Gray dash line corresponds to the perfect predictions.	112

Figure 4.4. Plots showing machine learning predictions on two functional group test-set (y-axis) vs. DFT redox potentials (x-axis). Gray dash line corresponds to the perfect predictions.	113
Figure 4.5. Plots showing machine learning predictions on three functional group test-set (y-axis) vs. DFT redox potentials (x-axis). Gray dash line corresponds to the perfect predictions.	113
Figure 4.6. Plots showing machine learning predictions on three functional group test-set (y-axis) vs. DFT redox potentials (x-axis). The combined dataset (training-set + two functional group test-set) was used for the training. Gray dash line corresponds to the perfect predictions.	114
Figure 4.7. R^2 vs. number of descriptors. R^2 was computed using the internal test-set. In this study, we identified a few issues with ARDR and GP. Despite the high predictive performance, ARDR is not a reliable model as it places very high weight on one feature (i.e., 'PEOE_VSAI'). Similarly, GP is not a reliable model as it becomes unstable when a small number of features are used. We encountered divided by zero errors in the kernel function during the analysis with GP model.	115
Figure 4.8. Top ten features (y-axis) vs. mean feature importance score (x-axis). Feature importance scores were rescaled between 0 to 1. Error bars represent the standard deviation obtained from 100 repetitions.	116
Figure 4.9. Redox Potential vs. 'PEOE_VSAI'	117
Figure 4.10. Examples from the training-set showing the effect of charge delocalization on 'PEOE_VSAI'. Values of 'PEOE_VSAI' and DFT redox potential in volts are also shown. Mol IDs were assigned to identify derivatives from the corresponding dataset.	118
Figure 4.11. Examples showing positive and negative shifts with respect to parent phenazine. DFT redox potentials and shifts in volts are also shown. Mol IDs were assigned to identify derivatives from the corresponding dataset.	118
Figure 4.12. Examples showing the effect of similar types of functional groups on the redox potential. DFT redox potentials and shifts in volts are also shown. Mol IDs were assigned to identify derivatives from the corresponding dataset.	119
Figure 4.13. Numbering of the positions in phenazine derivatives	120
Figure 5.1. Schematic diagram of a regulatory region containing different regulatory elements.	129
Figure 5.2. The Z-score normalized module-motif matrix of ChIP-seq (CTCF) data obtained from LDA (a) normalized along rows (module) and (b) normalized along columns (motif).	141

Figure 5.3. The Z-score normalized module-motif matrix of ChIP-seq (115 TFs) data obtained from HDP (a) normalized along columns (TFs) and (b) normalized along rows (modules).	142
Figure 5.4. The module-motif matrix obtained from LDA (eta = 0.01) corresponding to the ChIP-seq (CTCF) data. Each cell in the heatmap represents the z-score of the motif count (normalized along the rows) of a module (row).	144
Figure 5.5. The module-TF matrix obtained from LDA corresponding to the ChIP-seq (115 TFs) data. Each cell in the heatmap represents the z-score of the TF binding site count (normalized along the columns) of a TF (column) in a module (row).	145
Figure 5.6. The module-motif matrix obtained from HDP (eta = 0.1) corresponding to the ChIP-seq (CTCF) data. Each cell in the heatmap represents the z-score of the motif count (normalized along the rows) of a module (row).	147
Figure 5.7. Module-TF matrix obtained from HDP corresponding to the ChIP-seq (115 TFs) data. Each cell in the heatmap represents the z-score of the TF binding site count (normalized along the columns) of a TF (column) in a module (row).	148
Figure 5.8. The module-motif matrix obtained from NPLB corresponding to the ChIP-seq (CTCF) data. Each cell in the heatmap represents the z-score of the motif count (normalized along the rows) of a module (row).	149
Figure 5.9. The module-TF matrix obtained from NPLB corresponding to the ChIP-seq (115 TFs) data. Each cell in the heatmap represents the z-score of the TF binding site count (normalized along the columns) of a TF (column) in a module (row).	150
Figure 5.10. The module-motif matrix obtained from LDA corresponding to the DNase-seq data. Each cell in the heatmap represents the z-score of the motif count (normalized along the columns) of a motif (column) in a module (row).	151
Figure 5.11. The module-motif matrix obtained from HDP corresponding to the DNase-seq data. Each cell in the heatmap represents the z-score of the motif count (normalized along the columns) of a motif (column) in a module (row).	151
Figure 5.12. The module-motif matrix obtained from NPLB corresponding to the DNase-seq data. Each cell in the heatmap represents the z-score of the motif count (normalized along the columns) of a motif (column) in a module (row).	151
Figure 6.1. Artificial test system.	172
Figure 6.2. Plots showing expected and predicted rotation on the y-axis and manual rotation on the x-axis for the ring and the track corresponding to test-1. The red line represents predicted rotation from the algorithm, and the blue line denotes expected rotation.	173
Figure 6.3. Plots showing the expected and predicted rotation on the y-axis and manual rotation on the x-axis for the ring and the track corresponding to test-2. The red line	

represents the predicted rotation from the algorithm, and the blue line denotes the expected rotation. 173

Figure 6.4. Plots showing expected and predicted rotation on the y-axis and manual rotation on the x-axis for the ring and the track corresponding to test-3. The red line represents predicted rotation from the algorithm, and the blue line denotes expected rotation..... 174

Figure 6.5. Plots showing expected and predicted rotation on the y-axis and manual rotation on the x-axis for the ring and the track corresponding to test-4. The red line represents predicted rotation from the algorithm, and the blue line denotes expected rotation..... 174

Figure 6.6. Plots showing expected and predicted rotation on the y-axis and manual rotation on the x-axis for the ring and the track corresponding to test-5. The red line represents the predicted rotation from the algorithm, and the blue line denotes the expected rotation..... 175

Figure 6.7. Plots showing expected and predicted rotation on the y-axis and manual rotation on the x-axis for the ring and the track corresponding to test-6. The red line represents the predicted rotation from the algorithm, and the blue line denotes the expected rotation..... 175

Figure 6.8. Plots showing expected and predicted translation on the y-axis and manual translation on the x-axis for the ring and the track corresponding to test-7. The red line represents the predicted translation from the algorithm, and the blue line denotes the expected translation..... 176

Figure 6.9. Plots showing the expected and the predicted translation on the y-axis and the manual translation on the x-axis for the ring and track corresponding to test-8. The red line represents the predicted translation from the algorithm, and the blue line denotes the expected translation..... 176

Figure 6.10. Net relative rotation of the ring in rotaxane simulated for 44,697 steps. 177

Figure 6.11. Net relative translation of the ring in rotaxane system simulated for 44,697 steps..... 177

Figure 6.12. The net relative rotation of the ring in the rotaxane system simulated for 93,132 steps at 1600 K..... 178

Figure 6.13. Net relative translation of the ring in rotaxane system simulated for 93,132 steps at 1600 K..... 178

Figure 6.14. Net absolute rotation of the ring and track in the rotaxane system. 179

Figure 6.15. Rotation of ring and track atoms in rotaxane. (a) Net absolute rotation of ring atoms. (b) Net relative rotation of ring atoms. (c) Net absolute rotation of track atoms. 180

Figure 6.16. Visualization of the rotation axis in a randomly selected time step. Orange circles represent the ring atoms. The green arrow coming out of the figure (towards the

viewer) is the rotation axis identified by the algorithm. The plane perpendicular to the rotation axis is shown in blue.....	181
Figure 6.17. Distribution of instantaneous absolute rotation of ring atoms in the rotaxane system.	182
Figure 6.18. Visualization of a time step having a maximum value of instantaneous absolute rotation for a randomly selected ring atom. The rotation of ring atoms is represented by red circles. The size of the red circle corresponds to the amount of rotation.	182
Figure 6.19. Visualizing linear regression in (i) one dimension and (ii) in three dimensions.	183
Figure 6.20. Net absolute rotation of the ring atoms obtained from linear regression strategy.	184
Figure 6.21. Net absolute rotation of the ring obtained from linear regression strategy.	184
Figure 6.22. Visualization of the rotation axis obtained from the axis optimization strategy.	185
Figure 6.23. Examples showing valid and invalid time steps according to the cylinder test.	186
Figure 6.24. The net absolute rotation of the ring in a rotaxane system.	186
Figure 6.25. The net absolute rotation of ring atoms in the rotaxane system obtained after the incorporation of the axis optimization strategy and the cylinder test.	187
Figure 6.26. The net relative rotation of the ring in the rotaxane system obtained after the incorporation of the axis optimization strategy and the cylinder test.	187
Figure 6.27. The net relative rotation of the ring atoms in the rotaxane system obtained after the incorporation of the axis optimization strategy and the cylinder test.	187
Figure 6.28. Visualizing the maximum and minimum rotation of all ring atoms obtained without algorithmic improvements and with algorithmic improvements. Algorithmic improvements include axis optimization strategy and cylinder test.	188
Figure 6.29. Net absolute rotation of the track in the rotaxane system obtained after algorithmic improvements. Algorithmic improvements include axis optimization strategy and cylinder test.	189
Figure 6.30. Net absolute rotation of the track atoms in the rotaxane system obtained after algorithmic improvements. Algorithmic improvements include axis optimization strategy and cylinder test.	189
Figure 6.31. The strategy used for identifying track atoms. (a) The previous strategy employed two infinite planes, leading to the inclusion of distant track atoms. The yellow	

sheet represents the area between two planes (b) The new strategy uses a sphere around the center of rotation to identify the track atoms. The red circle represents the ring, and the green curve represents the track..... 190

Figure 6.32. The net absolute rotation of the track in the rotaxane system obtained after incorporating the new procedure for identifying local track atoms. 191

Figure 6.33. The net absolute rotation of track atoms in the rotaxane system obtained after incorporating the new procedure for identifying local track atoms. 191

Figure 6.34. Investigating rotational motion in the catenane system simulated at 1500 K with solvent for 50,000 steps. (a) The net absolute rotation of track atoms. (b) The net relative rotation of the ring. (c) The net relative rotation of all ring atoms. 192

Figure 6.35. Plots showing expected and predicted rotation on the y-axis and manual rotation on the x-axis for the ring and the track corresponding to Scheme-1. The red line represents the predicted rotation from the algorithm, and the blue line denotes the expected rotation. (a) and (b) verification results on the ring and the rack from the artificial test system, respectively. (c) and (d) verification results on the ring and the track from the rotaxane test system, respectively. 193

Figure 6.36. Net absolute rotation of track atoms in a simulated artificial test system. 194

Figure 6.37. Scatter plot of the angular deviation of the rotation axis with respect to the initial rotation axis. 195

Figure 6.38. Artificial test system containing only the ring..... 195

Figure 6.39. Plots showing expected and predicted rotation on the y-axis and manual rotation on the x-axis for the artificial test system containing only a ring. Verification was performed with and without translation of the ring. The red line represents predicted rotation from the algorithm, and the blue line denotes the expected rotation..... 196

Figure 6.40. Net absolute rotation of the ring atoms in the simulated rotaxane test system containing only the ring. 196

Figure 6.41. Effect of the track on the rotation of the ring in the rotaxane system simulated at 1300 K with solvent and counterions. (a) Net absolute rotation of ring atoms in the presence of the track. (b) Net absolute rotation of ring atoms without the track. (c) Net absolute rotation of the ring computed after decreasing the radius of the cylinder. (d) Angular deviation of the rotation axis with respect to the initial rotation axis. 198

Figure 6.42. Visualizing the distortion of the ring in a rotaxane system without track (a) Ring at time step 0 (b) Ring at time step 20,000. 198

Figure 6.43. Plots showing net absolute rotation of the ring with and without solvent in the rotaxane system simulated at 1300 K in the presence of counterions. 199

Figure 6.44. Net absolute rotation of the ring with and without counterions in the rotaxane system simulated at 1300 K without solvent.	200
Figure 6.45. Box plots showing the distribution of net absolute rotation of the ring in rotaxane system simulated by varying the number of counterions at 1300 K without solvent.	200
Figure 6.46. Net absolute rotation of the ring atoms in the rotaxane system simulated at 1300 K without solvent (a) with counterions. (b) without counterions.	200
Figure 6.47. Net absolute rotation of the ring in the rotaxane system simulated with and without solvent and counterions at 1300 K.....	201
Figure 6.48. Net absolute rotation of the ring atoms in the rotaxane system simulated at 1300 K (a) with solvent and counterions. (b) without solvent and counterions.	201
Figure 7.1. A representation of the research work presented in the thesis.	210

List of Tables

Table 1.1. Advantages and disadvantages of molecular space exploration strategies.	9
Table 2.1. Some commonly used evaluation metrics in regression and classification tasks.	39
Table 3.1. Representative structures from training-set/test-set. Mol IDs were assigned to identify derivatives from the corresponding dataset.	70
Table 3.2. Feature sets.	71
Table 3.3. List of all features.	72
Table 3.4. List of Models.	73
Table 3.5. Threshold values.	74
Table 3.6. Fifteen best-performing models. Models were trained on all features from the corresponding feature set.	78
Table 3.7. Test-set performance of the best-performing models in each feature set. Models were trained on all features from the corresponding feature set.	79
Table 3.8. Training and test set performance of four feature sets averaged over all models except linear regression. Models were trained on all features from the corresponding feature set.	80
Table 4.1. Representative structures from external test-set. Mol IDs were assigned to identify derivatives from the corresponding dataset.	106
Table 4.2. Representative structures from multiple functional group test-sets. Mol IDs were assigned to identify derivatives from the corresponding dataset.	107
Table 4.3. Parameter grids used during hyperparameter optimization.	108
Table 4.4. Average model performance on external validation-set at different values of 'k'.	110
Table 4.5. Values of performance metrics on internal and external test-sets. Numbers were rounded upto two decimals	112
Table 4.6. Examples showing the effect of the absolute value of a single functional group shift on the redox potential of derivatives containing different types of functional groups. DFT redox potentials and shifts in volts are also shown. Mol IDs were assigned to identify derivatives from the corresponding dataset.	120
Table 4.7. Top five anolyte candidates predicted using DFT, KRR, and SVR from the external test-set. SVR and KRR were trained on the phenazine derivatives containing a single type of functional group per derivative. Mol IDs, and redox potentials predicted from DFT	

and ML models are shown below the respective candidates. Mol IDs were assigned to identify derivatives from the corresponding test-set. Derivatives are arranged in increasing order of their redox potential. Redox potentials are given in the unit of volt. 122

Table 5.1. Distribution of motifs across three modules in the simulated dataset. Cells represent motif counts. 133

Table 5.2. List of hyperparameters used during training. An array in front of some hyperparameters represents different values tested during training. 136

Table 5.3. The list of some important transcription factors found in the K562 cell line. This list serves as a reference to understand the functional roles of the modules discovered in this study. 138

Table 5.4. The module-motif matrix predicted by LDA corresponding to the simulated dataset. 143

Table 5.5. The module-motif matrix predicted by HDP ($\eta = 0.1$, $\gamma_a = 5$, $\gamma_b = 0.1$, $\alpha_a = 0.1$, and $\alpha_b = 1.653$) corresponding to the simulated dataset. 145

Table 5.6. Module-motif matrix predicted by NPLB corresponding to the simulated dataset. 148

Table 5.7. Summary of results. The tick (✓) represents the successful identification of expected regulatory modules by the model, whereas the cross (✗) represents a failure of the same. 153

Table 6.1. List of different conditions under which rotaxane system was simulated. 168

Table 6.2. Simulation profile of the artificial test system. 194

Table 6.3. Simulation profile of rotaxane test system containing only the ring. 196

Abbreviations

ML	Machine Learning
DNA	Deoxyribonucleic Acid
RNA	Ribonucleic Acid
TF	Transcription Factor
ENCODE	Encyclopedia of DNA Elements
HTT	High Throughput Technologies
NGS	Next-Generation Sequencing
ChIP-seq	Chromatin Immunoprecipitation (ChIP) assay with Sequencing
DNase-seq	DNase I hypersensitive sites Sequencing
MD	Molecular Dynamics
DFT	Density Functional Theory
RFB	Redox Flow Battery
TSS	Transcription Start Site
AMMs	Artificial Molecular Machines
MIMs	Mechanically Interlocked Molecules
SVM	Support Vector Machines
SVR	Support Vector Regression
ARDR	Automatic Relevance Determination Regression
GP	Gaussian Processes Regression
KRR	Kernel Ridge Regression
DME	Dimethoxyethane
MSE	Mean Squared Error
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error

CV	Cross-Validation
FG	Functional Group
LDA	Latent Dirichlet Allocation
HDP	Hierarchical Dirichlet Processes
NPLB	No Promoter Left Behind
AIMD	<i>ab initio</i> Molecular Dynamics
QM	Quantum Mechanics
B3LYP	Becke, 3-parameter, Lee–Yang–Parr
HF	Hartree–Fock
COG	Center of Geometry

Physical Constants

Avogadro's Constant	$(N_A) = 6.02214129 \times 10^{23} \text{ mol}^{-1}$
Atomic Mass Unit	$(u) = 1.660538921 \times 10^{-27} \text{ kg}$
Boltzmann's Constant	$(k) = 1.3806488 \times 10^{-23} \text{ JK}^{-1}$
Bohr Radius	$(a_0) = 5.291772109 \times 10^{-11} \text{ m}$
Elementary Charge	$(e) = 1.602176565 \times 10^{-19} \text{ C}$
Gas Constant	$(R) = 8.3144621 \text{ JK}^{-1}\text{mol}^{-1}$
Mass of Electron	$(m_e) = 9.10938291 \times 10^{-31} \text{ kg}$
Mass of Proton	$(m_p) = 1.672621777 \times 10^{-27} \text{ kg}$
Mass of Neutron	$(m_n) = 1.674927351 \times 10^{-27} \text{ kg}$
Rydberg Constant	$(R) = 1.097373157 \times 10^7 \text{ m}^{-1}$
Speed of Light	$(c) = 2.99792458 \times 10^8 \text{ ms}^{-1}$
Planck's Constant	$(h) = 6.62606957 \times 10^{-34} \text{ Js}$
Faraday constant	$(F) = 96485.33212 \text{ C}\cdot\text{mol}^{-1}$

Chapter 1

Brief Introduction to the Molecular Space and Strategies for its Exploration

Chapter 1

Brief Introduction to the Molecular Space and Strategies for its Exploration

Abstract

The progress of society is tied to the progress of science. Scientific progress has resulted in technological advancements, improvement in the quality of life, and an increase in the average life span. Discovery is an essential aspect of scientific progress. Scientific discoveries involve discovering new materials, new scientific phenomena, or insight into a known phenomenon. Quite often, scientific discoveries are associated with molecules and require exploration through the available space for a class of molecules (i.e., molecular space). This chapter provides a brief introduction to the molecular space and the conventional strategies for its exploration. A large amount of data has been generated due to advancements in experimental and computational tools. However, conventional strategies do not take advantage of these datasets. Furthermore, they also suffer from certain issues, making them inefficient for the exploration of large molecular space. On the other hand, machine learning algorithms can learn from the data and provide accurate predictions in a short amount of time. This chapter outlines the issues with the conventional strategies and motivates the reader to develop efficient strategies based on machine learning algorithms.

1.1 Importance of the Scientific Discoveries and Their Connection to Molecules

Scientific progress is perhaps the most important facet of today's society. Over the past two hundred years, humanity has experienced growth at a pace that few could have imagined. This has primarily been driven by the scientific discoveries that fuelled significant economic and technological developments. Science has, for example, cured diseases, brought us closer through travel and modern communications technology, and helped us better understand and respond to environmental challenges. New scientific discoveries are still required to tackle the challenges we face today. Scientific discoveries of new materials, new phenomena, or an improved understanding of the known phenomena allow us to develop new technologies, solve practical problems and make informed decisions — both individually and collectively. For example, the discovery of the structure of DNA (deoxyribonucleic acid) was a fundamental breakthrough in biology. It formed the underpinnings of all biomedical research, including DNA fingerprinting, genetically engineered crops, and the diagnosis of genetic diseases.¹⁻³ The discovery of metals and alloys was critical to the technological progress of society. It is impossible to imagine the world today without metals. Global annual steel production reached 1864 million tonnes (Mt) in 2020.⁴ However, scientific discovery is not an easy process. Historically, scientific discoveries and technological advancements resulted from serendipity from decades of experimentation. A most famous example of this is the discovery of penicillin, an antibiotic effective against bacterial infections.⁵ In 1928, Alexander Fleming, during his investigation on staphylococci bacteria, left one petri dish open and went on holiday. After returning, he observed blue-green mold in the petri dish that had killed all the surrounding bacteria. The mold contained an antibiotic capable of killing harmful bacteria. At the time, Fleming's discovery did not garner much scientific attention, and it took another decade before penicillin was made available for use. The serendipitous discovery of the shape memory effect in nitinol is another example. Nitinol is a class of metal alloys with unique properties such as superplasticity and shape memory. William J. Buehler, who was a metallurgist at Naval Ordnance Laboratory (NOL)⁶, discovered the shape memory effect in nitinol while searching for a suitable material capable of sustaining the heat of re-entry in the earth's atmosphere for the nose core of the Polaris missile. In 1959, He inspected about 60 alloys, including nitinol. During testing, he intentionally dropped the cold ingots of nitinol on the floor. He expected to hear a bell-like ring. Instead, he heard a thud-like sound after dropping. He thought the ingot had some internal flaws, so he dropped the hot ingot on the floor, which made the expected bell-like ring.⁷ However, the hot ingot, after cooling, made a similar dull thud-like sound. This observation made Buehler realize the presence of double states in the nitinol. He continued experimenting with it. The shape memory effect of nitinol was not discovered until 1961, when one of his colleagues used a lighter to straighten the bent nitinol rod during a meeting. Now, nitinol is widely used in many applications such as biomedical, actuators, aerospace, automotive, and MEMS (micro-electromechanical system) devices.⁸ Thus, the discovery of new materials or phenomena is not only scientifically important but also essential to technological developments. New scientific discovery (material or phenomenon) is often connected with some material. For example, the discovery of superconductivity in mercury, the discovery of penicillin, the first antibiotic molecule; insight into the shape memory effect due to the discovery of nitinol, the discovery of BaTiO₃, the first commercial ferroelectric and piezoelectric material etc.^{5,6,9,10} Molecules lie at the heart of materials such as battery materials,

catalytic materials, polymeric materials. Their properties primarily depend on the composition of elements and the structure of the constituent molecules.

1.2 Molecular Space and Motivation for its Exploration

Molecular space is a collection of known and unknown molecules that differ in their atoms, bonding patterns, and configurations. It is a subspace of a large chemical space that includes all known and unknown elemental compositions, molecules, conformations and configurations, and chemical reactions.¹¹ In this thesis work, we have investigated materials composed of molecules. Therefore, we restrict ourselves to the molecular subspace. Molecular space is a high-dimensional space of molecules in which each axis represents different properties associated with the molecules. It is possible to discover new molecules for a particular application if one moves along the axis of desired property in the molecular space, as shown in Figure 1.1. Similarly, if we are interested in maximizing or minimizing more than one property, we can move along the multiple axes of the desired properties in molecular space. Besides materials discovery, molecular space can also reveal new patterns, such as the functional relationships between molecules. For example, in cells, gene expression is regulated through the binding of unique proteins known as transcription factors (TF) to DNA sequences (motifs). Different TFs bind to different motifs. However, certain groups of TFs bind to similar motifs and regulate the same genes.¹² These functionally similar TFs occupy nearby positions in molecular space. Visualizing the molecular space of these TFs can reveal different classes of functionally similar TFs. Thus, molecular space can help us discover new molecules and scientific phenomena or improve our understanding of the known scientific phenomenon. However, discovering new molecules or phenomena requires exploration through the molecular space. It is important to answer two fundamental questions before exploration: (i) How large is molecular space? (ii) How to explore the molecular space? These two aspects are not decided independently. Molecular space dictates the exploration strategy, and often exploration strategy may restrict the space that can be explored. Even though a universal molecular space exists, we generally restrict ourselves to a smaller subspace for efficient exploration. Therefore, we define a molecular space corresponding to a given application, which is a subspace of a large molecular space. Cayley, the inventor of graph theory, first attempted to estimate the size of a molecular space containing acyclic branched hydrocarbons in 1875.¹³ He calculated the number of possible acyclic branched hydrocarbons using graph-theoretical algorithms. Later, similar attempts were made to estimate the number of possible molecules of certain types. The size of molecular space for organic “drug-like” molecules is estimated to lie between 10^{18} to 10^{200} molecules; however, the commonly reported number for the size of molecular space is 10^{60} for the molecules obeying Lipinski’s rule-of-five.^{14,15} Although it is impossible to enumerate the entire molecular space, efforts have been made to compile known molecules into a collection. These collections are known as databases representing known parts of the molecular space. For example, PubChem is a publicly available database containing molecular structures and bioassay data. It is maintained by the U.S. National Center for Biotechnology Information (NCBI) and, as of August 2018, contains 111 million unique chemical structures.¹⁶ Chempidder is a similar database owned by the Royal Society of Chemistry. It gives access to over 100 million chemical structures.¹⁷ ZINC is a database of commercially-available drug-like compounds. It contains over 750 million purchasable compounds.¹⁸ Additionally, small and more specialized databases are also

available. Similarly, datasets of hundreds of complete genome sequences are available today due to high throughput genome sequencing. For instance, the ENCODE project has a large collection of transcriptomic datasets.¹⁹ GenBank is a public database that contains an annotated collection of DNA sequences.²⁰ KEGG is another database that contains information on biomolecular pathways and the molecular building blocks of genes, proteins, and drug molecules.²¹ The structural genomics efforts have also resulted in a corresponding increase in the number of macromolecular structures (e.g., proteins). The Protein Data Bank (PDB) is a database for the three-dimensional structural data of large biological molecules, such as proteins and nucleic acids.²²

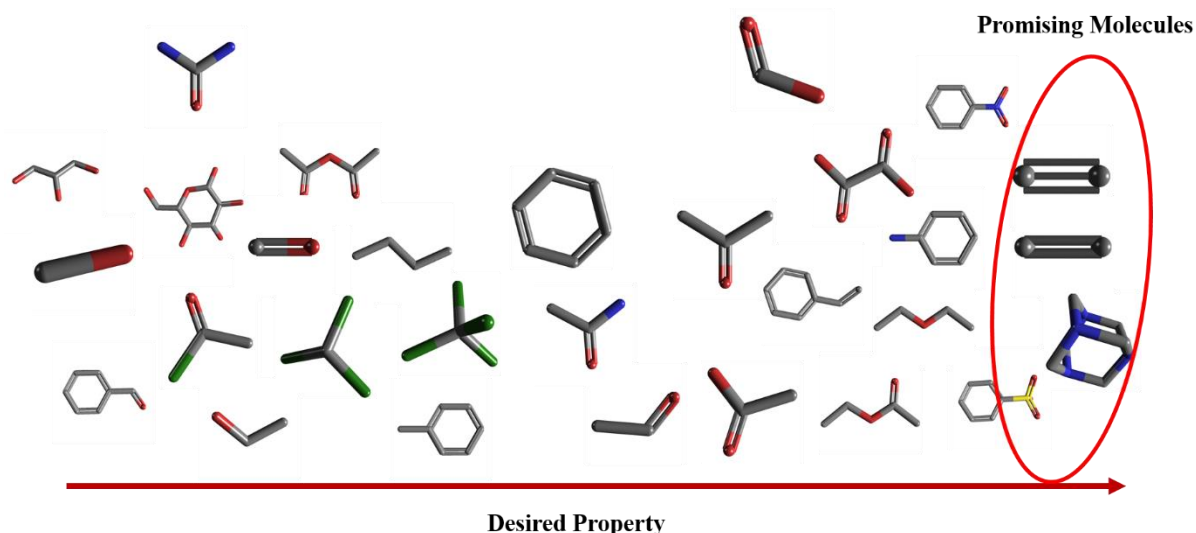


Figure 1.1. Visualization of molecular space along the axis of the desired property.

1.3 Conventional Strategies for the Exploration of Molecular Space

1.3.1 Experimental Approaches

Experimentation is the most reliable and robust methodology for scientific discoveries and verification of theories. Important scientific discoveries have been made through experiments. Even with the advent of computation tools, experiments are still a significant part of today's science. However, experimentation is not an easy task. Science in the early 20th century was primarily experimental. Scientists had to carry out many experiments that required a significant investment of time and resources. They had to wait patiently for one of the experiments to show promising results. We hear the stories of only successes. There are hundreds or even thousands of failed experiments behind a successful discovery. The famous quote from Thomas Edison nicely summarises the difficulties in early scientific discoveries *"I have not failed. I've just found 10,000 ways that won't work"*.²³ In recent years, advances in technologies have accelerated the field of experimental research. High-throughput and combinatorial approaches have made it possible to carry out thousands of experiments in parallel.²⁴ The early report on high throughput technologies (HTT) was published by Hanak in the 1970s.²⁵ He demonstrated the benefits of HTT in the search for new materials. In 1995, Xiang, Schultz, and co-workers reinitiated the successful application of combinatorial methodologies in materials science.²⁶ With the innovation of automated HTT, experimental approaches have been applied to explore

the molecular space of a variety of materials such as drug molecules, ferroelectric materials, catalytic materials, superconducting materials etc.²⁶⁻²⁹ The field of biology is inherently combinatorial (DNA, proteins, genes), thus lending itself naturally to the combinatorial approach. In life sciences, the combinatorial approaches started in the mid-1980s when Geyson first published a spatially resolved library of 96 peptides synthesized on microtiter plates.^{30,31} Recently, next-generation sequencing (NGS) has made genome sequencing more affordable and readily available.³² NGS include high throughput and massively parallel DNA-sequencing technologies such as ChIP-seq, DNase-seq, ATAC-seq, RNA-seq, etc.³³⁻³⁵ NGS technologies are currently being used for whole-genome sequencing, investigation of gene expression profile, epigenetics, cellular heterogeneity, the discovery of protein-binding sites and non-coding RNAs (ribonucleic acid), diagnosis of disease etc.^{32,34,36} Thus, high throughput approaches have been adopted in various fields of science for the exploration of molecular space.

1.3.2 Computational Approaches

The experimental approach is inherently limited due to the high costs of expensive instruments and time-consuming protocols. Historically, scientists could synthesize a small number (i.e., a few hundred) of materials in a year. With the advent of HTT, it is now possible to synthesize thousands of compounds in a very short amount of time.³⁷ However, the size of molecular space could easily reach the order of 10^{10-20} . Even with the advancement in combinatorial synthesis and HTT, it is impossible to explore such a large molecular space. Therefore, it is necessary to develop computational tools for the efficient exploration of molecular space. The scientific field observed a paradigm shift from empirical evidence to theories in the 20th century. The development of quantum mechanics led to the scientific revolution. Quantum mechanics changed the fundamental understanding of physics.³⁸ Theories developed in the early 20th century built the foundation of science and provided answers to unexplained phenomena. The second shift took place towards the end of the 20th century due to the advancement and innovation in computing. In the mid-1950s, electronic computers became available for general use by physicists and chemists. Chemists started using computers to obtain quantitative information about the behavior of molecules *via* numerical approximations to the solution of the Schrödinger equation.³⁹ Due to the enormous increase in speed, low cost, and development of efficient algorithms, computational tools started acquiring a central place in the research. Today, electronic structure codes have evolved from handling a few tens of atoms to more realistic simulations with thousands of atoms.⁴⁰ Now, computational studies are being carried out in many areas of science, including physics, chemistry, materials science, and biology.^{41,42} With supercomputers, scientists can simulate systems from microscopic scale to macroscopic scale (from femtoseconds to milliseconds). Molecular dynamics (MD) simulations are now possible on billions of atoms, with force fields promising near-quantum mechanical accuracy.⁴³ Computational tools allow scientists to investigate phenomena under conditions that are impossible to create in a laboratory. Now, it is possible to reliably automate first-principle calculations, especially those based on more cost-effective approximations such as Kohn-Sham density functional theory (DFT). High-throughput DFT calculations have been employed to explore molecular space for solar materials, carbon capture and gas storage materials, topological insulators, battery materials, catalytic materials, hydrogen storage materials etc.⁴⁴⁻⁵¹ Furthermore, combined experimental and computational approaches have

also been investigated for the exploration of molecular space in order to discover molecules with desired properties.⁵²⁻⁵⁵

1.3.3 Algorithmic Approaches

The experimental and computational strategies are generally employed to explore the known molecular space. Innovative algorithmic approaches have also been developed to navigate unknown molecular space. Algorithmic strategies involve the generation of new molecules guided by the desired property. For example, genetic algorithms combine molecule generation with a fitness function in iterative cycles to generate new molecules.⁵⁶ The first example of the development of a genetic algorithm for molecule generation is the SPROUT algorithm developed by Johnson and co-workers.⁵⁷ It is capable of growing molecules for a targeted protein binding site. SPROUT selects synthetically feasible molecules having maximum fitness as estimated by docking. Several other approaches have been developed based on genetic algorithms such as EVOLUATOR, Skelgen, TOPAS, and multi-objective optimization algorithms - GANDI and MEGA.⁵⁸⁻⁶² Algorithms that do not restrict themselves in synthetically feasible space could generate structurally more innovative molecules. Such an attempt to generate molecules independent of synthetic feasibility was made by Gasteiger *et al.* They reported an innovative molecular breeding algorithm based on the recombination of different molecular fragments.⁶³ This algorithm was used for generating molecules that maximize common features of two starting molecules by optimizing a fitness function that depends on Pareto rank and Tanimoto similarity. Genetic algorithms capable of breeding random fragments, generating any target molecule, and evolving molecular populations to maximum fitness by iterative cycles were also reported.⁶⁴⁻⁶⁶ As molecules can be efficiently represented through graph structures, Stadler *et al.* proposed exploration strategies based on graph-theoretical algorithms.⁶⁷ They also demonstrated the applicability of their strategy using key examples of complex chemical networks from sugar chemistry and the realm of metabolic networks. Among all the algorithms, genetic algorithms have been investigated more than others to explore molecular space.⁶⁸

Figure 1.2 shows the timeline of the development of molecular space exploration strategies discussed in this chapter. It also depicts the amount of statistical data required for each strategy, it can be seen that the amount of statistical data increases as we move from experimental approaches to machine learning approaches. Machine learning approaches are data-driven approaches. Therefore, they require a relatively large amount of data. In the following section, we discuss the motivation for developing exploration strategies based on the machine learning algorithm.

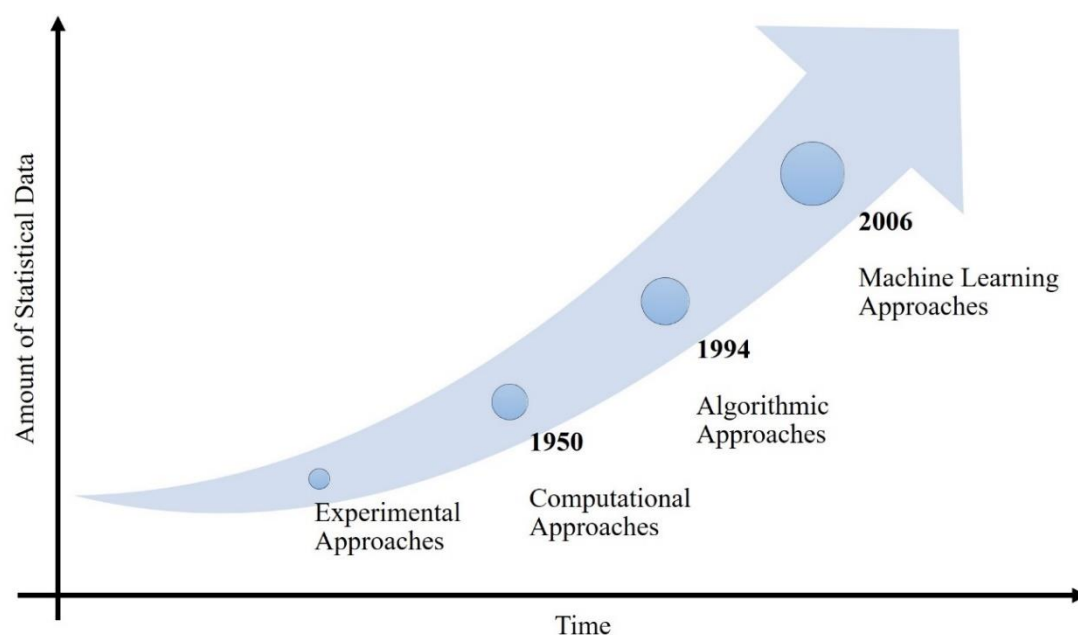


Figure 1.2. Timeline of the development of molecular space exploration strategies. The x-axis represents the time, whereas the y-axis represents the amount of statistical data required for each strategy.

1.4 Motivation for the Development of the Exploration Strategies Based on Machine Learning Algorithms

Exploration of molecular space to discover new materials or phenomena is not only scientifically important but also critical to technological developments. Experimental approaches pose high requirements in terms of equipment, infrastructure, and researcher expertise. On the other hand, computational approaches require large computing clusters to run electronic structure calculations. The computational approach is inefficient because no explicit use can be made of previous calculations for the new system. Typically, calculations can take anywhere from a few hours to months, and the accuracy of the solution depends on the level of theory. Although high-throughput approaches have enabled the rapid exploration of molecular space, it was quickly realized that even with HTT, only a small fraction of the molecular space could be explored. Even today, experimental and computational approaches depend to some extent on intuition and serendipity. Thus, large-scale experimental and computational studies are time-consuming and inefficient. Although algorithmic approaches (particularly genetic algorithms) are faster than experimental and computational approaches, they also have some disadvantages: (i) performance depends on the size and quality of the initial population, (ii) they may get stuck in local optima resulting in suboptimal solution, (iii) formulation of the fitness function is critical to the quality of results, (iv) results are susceptible to the choice parameters, any inappropriate choice would produce meaningless molecules.^{69,70} Due to all these issues, it typically takes 10-20 years for the discovery of new materials from the initial research to its first use.⁷¹ The advances in computational tools and experimental techniques have resulted in the generation of a large amount of data. Significant efforts have been made to collect and organize extensive datasets (into a database) generated from experimental and computational studies. Several databases corresponding to different applications are available today. Efficient methods are required to interrogate, analyze, process,

and infer knowledge from the existing databases. This decade has seen a rapid rise in the popularity of machine learning algorithms in scientific and industrial domains. Machine learning (ML) algorithms can handle large datasets and extract hidden patterns in high-dimensional data through accurate statistical models. Machine learning algorithms can make reliable predictions when in-depth knowledge is incomplete or inaccurate, when the amount of data is too large, or when there exist exceptions to the general rule. ML algorithms learn from empirical data by modeling the linear and non-linear relationships in the data. Machine learning approaches have thus been applied to numerous problems in science and engineering.^{72–76}

In Table 1.1, we compare various advantages and disadvantages of the molecular space exploration strategies discussed in this chapter.

Table 1.1. Advantages and disadvantages of molecular space exploration strategies.

Exploration Strategy	Advantages	Disadvantages
Experimental Approaches	<ul style="list-style-type: none"> • High accuracy • Most reliable • Used as a reference for the development of other strategies 	<ul style="list-style-type: none"> • Time-consuming • Resource intensive • Requires expensive instruments
Computational Approaches	<ul style="list-style-type: none"> • Faster than experimental approaches • Does not require expensive instruments • Able to investigate the behavior of the systems under extreme conditions 	<ul style="list-style-type: none"> • Accuracy depends on the level of theory • Calculations may take hours to months • Requires large computing clusters
Algorithmic Approaches	<ul style="list-style-type: none"> • Faster than computational approaches • Large computing clusters are not necessary • Possible to navigate unknown molecular space 	<ul style="list-style-type: none"> • Performance depends on the size and quality of the initial population • May get stuck in local optima resulting in a suboptimal solution • Formulation of the fitness function is critical • Susceptible to the choice parameters, any inappropriate choice would produce meaningless molecules
Machine Learning Approaches	<ul style="list-style-type: none"> • Accuracy could be improved to the level of the experimental approach • Faster than experimental, computational, and algorithmic approaches depending on the application 	<ul style="list-style-type: none"> • Requires a large quantity of data • Requires computing resources • Accuracy depends on the quality of training data

In this thesis work, we have investigated machine learning approaches for the exploration of three molecular spaces corresponding to different applications: (i) battery materials based on phenazine derivatives, (ii) biomolecules (DNA, proteins), and (iii) molecular machines. We provide below a brief background necessary to understand these molecular spaces and their corresponding applications.

1.5 Redox Flow Batteries (RFBs)

Redox flow batteries (RFBs) have emerged as promising grid-scale energy storage systems due to adjustable storage capacity, long service life for repeated charge-discharge cycles, high round-trip efficiency, fast response, low cost, low environmental impact, etc.⁷⁷⁻⁸¹ Inarguably, the major advantage of RFBs is their ability to decouple energy storage from power output, unlike other batteries where the two are correlated.⁸² This property provides substantial flexibility for designing flow battery systems according to the requirements of a particular application. The power output of RFB can be controlled by changing the size and number of electrochemical cells. The capacity of RFB could be modified by changing the size of storage tanks.^{83,84} These characteristics make RFBs an attractive candidate for grid-scale energy storage applications. In RFBs, liquid redox-active materials are circulated between electrolyte tanks and electrochemical cells. RFB consists of two storage tanks containing cathode and anode redox-active species dissolved in an electrolyte solution. The electrolyte solutions in the positive and negative compartments are termed catholyte and anolyte, respectively. These storage tanks are connected to an electrochemical cell (or current collector) *via* pumps. The electrochemical cell consists of porous electrodes separated by an ion-selective membrane. During operation, electrolytes containing redox-active species are pumped to the electrochemical cell, where redox-active species undergo oxidation or reduction depending on the charge/discharge cycle. Then, electrolytes are circulated back to their storage tanks.^{85,86} So far, transition metal-based redox flow batteries (such as vanadium, iron, and chromium) have found some commercial success.^{86,87} One of the early reports on RFBs comes from Japan as far back as 1971.⁸⁸ NASA in 1973 founded the Lewis Research Center and developed contracts with industries to investigate RFBs. NASA studied a variety of redox couples, but their focus was on the iron-chromium RFBs. They developed the iron-chromium (Fe-Cr) RFB with a 1 kW power output and 13 kWh capacity from 1973 to 1982.⁸⁹ This battery validated many desirable properties, but NASA scientists encountered some issues such as poor electrochemical reversibility of chromium and cross-contamination of redox-active species. Hence, in 1981, NASA shifted its research efforts from system design to more fundamental studies of RFBs. In 1984, NASA decided to shut down the program. Around 1980 in Japan, interest grew for electrochemical storage to complement other grid-scale energy storage systems. This storage technology was further developed in Japan within the Moonlight Project of the New Energy and Industrial Technology Development Organization (NEDO).⁹⁰ In this project, Fe-Cr redox systems in hydrochloric acid were investigated. Since 1985, Skyllas-Kazacos *et al.* have carried out significant research and development on the vanadium redox flow battery (VRFB).⁸⁶ They showed the first successful demonstration of redox flow batteries utilizing vanadium in each half cell. Large-scale VRFB plants have been installed globally by manufacturers for various applications.^{77,91} Among all the RFB technologies, VRFBs are commercially most successful today.

1.6 Transcription Regulation

Genes play an important role in determining physical traits.⁹² They carry information that makes our hair curly or straight, legs long or short, and even dictates how we might smile or laugh. Inside the cells, genes regulate biological processes critical for survival, such as development, reproduction, aging, and differentiation. Genes perform their tasks through proteins encoded in their DNA strands. However, proteins are not directly synthesized from DNA. First, information from the DNA is converted into RNA molecules through a process known as transcription.⁹³ Then, RNA molecules are used for making proteins through another process known as translation.⁹⁴ Transcription and translation are crucial steps in gene expression. The successful execution of biological processes requires an accurate and carefully orchestrated set of steps that depend on the precise spatial and temporal expression of genes. Any deregulation of gene expression often results in disease. The gene expression in eukaryotes is regulated at multiple levels, including transcription, elongation, mRNA processing, transportation, translation, and stability.¹² However, it is believed that most regulation occurs at the transcription level.⁹⁵ The discussion in this thesis is concerned with the eukaryotic transcription process. In eukaryotes, RNA polymerase II is responsible for the transcription of protein-coding genes.⁹⁶ Jacob and Monod in 1961 showed that the transcription of genes is controlled through the binding of special proteins known as Transcription Factors (TFs) to the specific sequences on the DNA (motifs).⁹⁷ These motifs are called transcription regulatory elements. Transcription regulatory elements include (i) promoters, (ii) enhancers, (iii) silencers, (iv) insulators, and (v) locus control regions.⁹⁸ Below, we give a brief introduction to these regulatory elements:

(i) Promoters: Promoter is a sequence of DNA that acts as an on-off switch for a gene. Promoter consists of two elements:

(a) Core Promoter: The DNA sequence located near the gene where transcriptional machinery assembles is known as a core promoter. It dictates the position of the Transcription Start Site (TSS) and the direction of transcription.⁹⁹ Examples of core promoters include TATA, Inr, DPE, and BRE.

(b) Proximal Promoter Elements: The proximal promoter elements serve as binding sites for activators. Activators are TFs that help in the assembly of transcriptional machinery.¹² Proximal promoter is located in a region immediately upstream from the core promoter.

(ii) Enhancers: Enhancers are regulatory elements that interact with promoters to control the transcription of the target gene. Enhancers regulate transcription in a spatial and temporally specific manner.¹⁰⁰ They generally increase the transcription of the target gene and function independent of the distance and orientation relative to the promoter element.¹⁰¹ Typically, enhancers are situated quite distally from the core promoter; thus, they are the long-range transcriptional control elements.¹⁰² Enhancers often act in a modular fashion.¹⁰⁰ A single promoter could be acted upon by many enhancer elements at different times or in different tissues. Enhancers usually consist of relatively closely grouped clusters of transcription factor binding sites.¹⁰³

(iii) Silencers: Silencers are sequence-specific elements with a negative effect on transcription.¹⁰⁴ They share most of the characteristics attributed to enhancers. Typically, they act independently of orientation and distance from the promoter, although some position-dependent silencers have also been found. Silencers can be situated close to a promoter or far from their target gene. Certain transcription factors called repressors bind the silencers. Repressors may recruit other TFs called corepressors for their function.¹⁰⁵ Number of studies have been conducted to understand the mechanism of repression. In some cases, it was observed that repressors might block the binding of nearby activators, directly competing for the same binding site.^{106,107} Repressors might function by preventing activators from accessing a promoter through recruitment of histone-modifying or chromatin-stabilizing factors that establish a repressive chromatin structure, as reported in another study.¹⁰⁸ Alternately, a repressor may block the assembly of the transcription machinery, as suggested by another study.¹⁰⁹

(iv) Insulators: Insulators function as boundary elements blocking genes from being affected by the transcriptional activity in the nearby genes.¹¹⁰ They partition the genome into distinct regulatory domains. Insulators have two essential characteristics: (a) they block communication between the enhancers and promoters, and (b) they can stop the spread of repressive chromatin structure. However, the precise mechanism of insulator activity is not entirely known. Generally, insulators function in a position-dependent and orientation-independent manner.

(v) Locus Control Regions (LCRs): Locus control regions are groups of regulatory elements responsible for regulating a cluster of genes.¹¹¹ They regulate tissue-specific, physiological expression of the linked transgene in a position-independent manner. LCRs typically consist of many regulatory elements such as enhancers, silencers, insulators, etc. The collective binding of TFs to different regulatory elements in LCR defines their functional activity on gene expression. The most significant property of LCRs is their strong enhancer activity. Similar to enhancers and silencers, LCRs can regulate gene expression from a distance in a position-independent fashion.¹¹²

1.7 Molecular Machines

Molecular machines play a central role in the fundamental biological processes critical for the survival of living organisms. Kinesin is a molecular motor found in eukaryotic cells that transports molecules inside the cell.¹¹³ Proteins are the workhorses of the cells. They are essential for cellular growth, metabolism, maintenance, and reproduction. Protein synthesis would not be possible without the ribosome, a natural molecular machine.¹¹⁴ Flagella are rotary molecular machines found in bacteria capable of unidirectional angular motion.¹¹⁵ Biological molecular machines are incredibly efficient compared to the corresponding artificial systems.¹¹⁶ Researchers believe understanding the mechanism of the biological molecular machines is essential for designing efficient systems. When we build molecular machines whose motions can be controlled in the desired environment, it would potentially impact all aspects of fundamental and applied science. However, it requires an improved understanding of the dynamics at a molecular level. A molecular machine could be defined as an assembly of different molecular components exhibiting mechanical movements in response to external stimuli.¹¹⁷

Molecular machines can be classified into two broad classes:

(i) Biological Molecular Machines: Biological molecular machines are some of the most efficient machines in nature. They are typically composed of protein molecules. Examples of biological molecular machines include F₀F₁-ATPase, kinesin, myosin, and dynein. Biological molecular machines function as energy transducers converting energy from one form to another. They rely on adenosine triphosphate (ATP) for the energy input. These machines have been carrying out essential functions inside and outside the cell. A rotary molecular machine, F₀F₁-ATP synthase, generates ATP, an essential energy-supplying molecule. It consists of two motors — the F₀-ATPase motor domain containing the proton channel and the soluble F₁-ATPase motor containing three catalytic sites.¹¹⁸ Other biological molecular machines such as myosin, kinesin, dynein, and the related proteins function as linear motors transporting molecular cargos along the polymeric structures.¹¹⁹ Myosin transports cargo along the actin filaments in muscles and other cells, whereas kinesin and dynein transport cargo along the microtubules. Myosin delivers power to all our voluntary and involuntary muscle activities. The transport motors convert the chemical energy of ATP into mechanical motion. These molecular machines also play an important role in cell division.

(ii) Artificial Molecular Machines (AMMs): Although there have been reports dating back to the 1970s and 1980s on the synthesis of artificial molecular systems exhibiting particular conformational changes, the field of AMMs really began in 1991 with the report by J. Fraser Stoddart on the molecular shuttle.¹²⁰ In the report, the authors investigated dynamical motion in rotaxane. Rotaxane consists of a ring mechanically interlocked onto an axle by bulky stoppers. They showed that the ring moves between two preferred binding sites due to random thermal motion. It was realized that the threaded (i.e., mechanically interlocked) structure of a rotaxane could potentially allow for the large-amplitude motion of molecular components in a controlled manner. Although mechanically interlocked molecular structures are not necessary for building AMMs, they provided the first step toward the practical synthesis of molecular architectures through which well-defined molecular motions could be controlled, studied, and utilized. In the early 1980s, Jean-Pierre Sauvage revolutionized the strategy of synthesis through the use of template methods to assemble mechanically linked molecules such as catenanes and rotaxanes.¹²¹ The groups of Sauvage, Stoddart, and others contributed to the development of new mechanically interlocked molecules based on rotaxane and catenanes from 1992 to 2007.^{117,120–123} They also invented novel strategies for switching the positions of components in rotaxane and catenane architectures under different stimuli such as light, temperature, charge etc.^{122,124,125}

AMMs could be further classified into three categories:¹²⁶

(a) Mechanically Interlocked Molecules (MIMs): This class of artificial molecular machines includes molecules in which different molecular components are interlocked. E.g., rotaxanes and catenanes.¹²⁰

(b) Molecular Switches: Molecular switches include molecules capable of inducing reversible transitions between different states in response to external stimuli. E.g., light-responsive molecular switches such as azobenzene.¹²⁷

(c) Molecular Motors: This class of molecule machines includes AMMs that undergo unidirectional motion under external stimuli. Molecular motors are fundamentally different from molecular switches. In contrast to molecular switches, molecular motors could drive the system away from equilibrium and continuously perform work in nonequilibrium environments for a complete cycle. In other words, no useful work is done in a molecular switch when components return to their original position. Whereas when components return to their original position in a molecular motor, any work that has been done is not undone. Researchers have built various artificial molecular motors inspired by biological motors.¹²⁶

The challenge in the artificial molecular machine is to design systems where the controlled motion of components results in useful tasks. Molecular machines should be designed according to the environment they are expected to operate. Molecular machines cannot simply mimic the mechanism of their macroscopic counterparts. Various factors such as random thermal motion, heat dissipation, solvation, momentum, and inertia affect the motion at a molecular level (i.e., nanometre scale). The forces influencing the dynamics at the nanoscale are not those we commonly encounter in the macroscopic world. Inertia, which depends on the mass of a particle, dominates the motion of large objects. As particle size decreases, momentum and gravity become less relevant, and viscous forces and Brownian motion become important. Molecular motors exploit random thermal fluctuations for directional motion by employing ratchet mechanisms.¹²⁵ Thought experiments such as Feynman's ratchet-and-pawl, Maxwell's demon, and Smoluchowski's trapdoor have investigated different strategies to cause the directional motion of Brownian particles.^{128,129} The second law of thermodynamics states that the Brownian motion resulting from thermal energy cannot be harnessed to produce useful work. Brownian motion has a disruptive effect on small objects. It was estimated that a molecule experiences thermal noise power that is at least eight orders of magnitude higher than the power obtained from a typical biochemical reaction fuelling the molecular motor.¹³⁰ Unfortunately, Brownian motion cannot be stopped except at 0 K; the only solution is to exploit random motion. This is what bimolecular machines do: they use external chemical energy to bias thermodynamics, making the movement more probable in one direction (i.e., directional). Thus, molecular machines required an external input of energy for the operations. The most obvious way to supply energy to a chemical system is through a reactant ("fuel") that undergoes an exoergonic reaction. This is what happens in biomolecular machines, which are typically powered by ATP. In ATP synthase, chemical energy is supplied in the form of a transmembrane proton gradient.¹³¹ In 2016, an artificial molecular machine capable of autonomous motion powered by chemical fuel was prepared by David Leigh and co-workers.¹³² Such an advancement in controlling the motion at a molecular level has opened doors for innovations and scientific discoveries.

1.8 Statement of problem (aims & objectives)

Scientific discoveries (i.e., the discovery of new material or phenomena or insight into known phenomena) are often associated with novel molecules. They are essential for the progress of society. However, scientific discoveries require exploration through molecular space. Traditionally, exploration of molecular space has been carried out through experimental, computational, and, in some cases, algorithmic approaches. However, experimental and computational approaches are time-consuming and expensive due to inherent limitations.

Although algorithmic approaches are relatively fast, they are highly sensitive to the parameters and starting conditions and may end up producing a lot of useless molecules. Thus, traditional approaches are not suitable for exploring the large and ever-expanding molecular space to meet the technological demands of society. Recently, machine learning (ML) algorithms have shown superior performance in many applications. ML algorithms are capable of handling and extracting knowledge from large existing datasets. High throughput approaches have resulted in the generation of a huge amount of data. Thus, approaches based on ML algorithms are more suitable for the efficient exploration of molecular space. However, a single ML strategy may not work for all molecular spaces. The choice of the exploration strategy depends on the molecular space and the given application. One may need to develop a new strategy if existing strategies do not work. The aim of this work was to demonstrate the applicability of the machine algorithms to explore different molecular spaces for discovering promising molecules or new phenomena or get insight into known phenomena guided by the desired property (i.e., application).

In this work, we have attempted to address the following questions that arise during the development of an exploration strategy:

How does the molecular space depend on the given application?

How do we develop new strategies for the exploration of molecular space using machine learning algorithms?

Which approach is more suitable? Computational, algorithmic, machine learning, or a combined approach?

Which machine learning algorithms are better suited for exploring a given molecular space?

How does a machine learning algorithm help in reducing the time required for the computation of a complex quantity?

Is it possible to use a small dataset to develop an exploration strategy based on machine learning algorithms?

Is it possible to use the developed strategies for new scientific discoveries?

Objectives:

Identify the scientific problems/applications that require exploration of the molecular space.

Construct the molecular spaces corresponding to each application.

Develop the exploration strategies for these molecular spaces using supervised and unsupervised machine learning algorithms and computational approaches if required.

If required, develop a new algorithm to address certain aspects of the exploration strategy.

Use the developed strategy to (i) discover potential molecules from the molecular space and/or (ii) discover new phenomena and/or (iii) gain insight into new or known scientific phenomena.

1.9 Organization of the Thesis

This thesis is divided into seven chapters. A brief introduction to each chapter is provided below with the chapter titles.

Chapter-1: Brief Introduction to the Molecular Space and Strategies for its Exploration

This chapter briefly introduces the molecular space and conventional strategies for its exploration. We have outlined the issues associated with the conventional strategies and highlighted the need for efficient strategies that utilize previous data to explore the molecular space. We propose that machine learning approaches are more suited for exploring molecular spaces for which datasets are available. We have also briefly described the molecular spaces investigated in this thesis.

Chapter-2: Fundamentals of Machine Learning

This chapter discusses the fundamentals of machine learning algorithms, such as supervised and unsupervised learning methods. Also, commonly used algorithms and different aspects and components of a typical machine learning workflow have been described briefly.

Chapter-3: Machine Learning the Redox Potentials of Phenazine Derivatives: A Comparative Study on Molecular Features

This chapter describes the development of machine-learning models to explore the molecular space containing battery materials. In particular, we have investigated molecular space containing phenazine derivatives, which are promising redox-active candidates for redox flow batteries (RFBs). We employed twenty linear and non-linear machine-learning models for the prediction of the redox potential (desired property) of phenazine derivatives in dimethoxyethane (DME) solvent using a small dataset of 189 molecules. The models achieved excellent performance on the unseen data (i.e., $R^2 > 0.98$, $MSE < 0.008 \text{ V}^2$ and $MAE < 0.07 \text{ V}$). Furthermore, the predictive performance of four types of molecular features (i.e., 2D, 3D, and molecular fingerprints) was analyzed. It was observed that the 2D molecular features were most informative and achieved the highest prediction accuracy.

Chapter-4: Predicting the Redox Potentials of Phenazine Derivatives Using a Hybrid DFT-ML Approach

This chapter demonstrates a hybrid DFT-ML approach to explore the molecular space containing green battery materials (i.e., phenazine derivatives). We developed four machine learning (ML) models to predict the redox potentials of phenazine derivatives in dimethoxyethane using density functional theory (DFT) and a small training data of 151 phenazine derivatives having only one type of functional group per molecule (20 unique groups). Despite being trained on the molecules with a single type of functional group, the models were able to predict the redox potentials of derivatives containing multiple and different types of functional groups with good accuracy ($R^2 > 0.7$).

Chapter-5: Investigating Combinatorial Binding of Transcription Factors using Unsupervised Machine Learning Models

This chapter describes the development of an unsupervised machine learning approach to explore the molecular space containing DNA regions and special proteins called transcription

factors (TFs) obtained from next-generation sequencing (NGS). We have investigated topic models (Latent Dirichlet Allocation and the Hierarchical Dirichlet Processes) and a recently developed No Promoter Left Behind (NPLB) approach to cluster the DNA regions obtained from the ChIP-seq and DNase-seq data of the K562 cell line. The results showed that the models could identify the commonly occurring regulatory modules in the K562 cell line. The identified regulatory modules contained the transcription factors that generally co-occur or are functionally similar. NPLB was most successful in identifying regulatory modules from both the ChIP-seq datasets investigated in this study. We also obtained the regulatory modules from DNase-seq data using topic models.

Chapter-6: An Algorithmic Development of the Strategy for Quantifying Rotational Motion in Molecular Machines

This chapter demonstrates the development of an algorithm for quantifying the rotational motion (desired property) in molecular machines. In particular, mechanically interlocked molecules having a ring and a track were investigated (i.e., rotaxane and catenane). We also investigated linear regression, a machine learning algorithm, during the development. We performed several tests to verify the algorithm using an artificial test system and a rotaxane test system. It was observed that the developed algorithm could reasonably quantify the absolute rotation of the ring in rotaxane and catenane. We also investigated the effect of the track, solvent, and counterions on the rotation of the ring.

Chapter-7: Summary and Future Outlook

This chapter provides conclusions and the future outlook that can be developed from the work done and reported in this thesis.

1.10 References

- (1) Ely Hepfer, C.; Piperberg, J. B.; Farganis, G. M. An Introduction to DNA Fingerprinting. *Am. Biol. Teach.* **1993**, *55* (4), 216–221. <https://doi.org/10.2307/4449636>.
- (2) Gould, F. Evolutionary Biology and Genetically Engineered Crops. *Bioscience* **1988**, *38* (1), 26–33. <https://doi.org/10.2307/1310643>.
- (3) Antonarakis, S. E. Diagnosis of Genetic Disorders at the DNA Level. <http://dx.doi.org/10.1056/NEJM198901193200305> **2010**, *320* (3), 153–163. <https://doi.org/10.1056/NEJM198901193200305>.
- (4) Global crude steel output decreases by 0.9% in 2020 | worldsteel <https://www.worldsteel.org/media-centre/press-releases/2021/Global-crude-steel-output-decreases-by-0.9--in-2020.html> (accessed Jan 1, 2022).
- (5) Ban, T. A. The Role of Serendipity in Drug Discovery. *Dialogues Clin. Neurosci.* **2006**, *8* (3), 335. <https://doi.org/10.31887/DCNS.2006.8.3/TBAN>.
- (6) KAUFFMAN, G. B.; MAYO, I. The Story of Nitinol: The Serendipitous Discovery of the Memory Metal and Its Applications. *Chem. Educ.* **1997**, *22* (2), 1–21. <https://doi.org/10.1007/S00897970111A>.
- (7) A Brief History of Nitinol - Kellogg's Research Labs <https://www.kelloggsresearchlabs.com/2018/01/10/a-brief-history-of-nitinol/> (accessed Jan 13, 2022).
- (8) Chaudhari, R.; Vora, J. J.; Parikh, D. M. A Review on Applications of Nitinol Shape Memory Alloy. **2021**, 123–132. https://doi.org/10.1007/978-981-33-4176-0_10.
- (9) Van Delft, D.; Kes, P. The Discovery of Superconductivity. *Phys. Today* **2010**, *63* (9), 38–43.
- (10) Buscaglia, V.; Buscaglia, M. T.; Canu, G. BaTiO₃-Based Ceramics: Fundamentals, Properties and Applications. *Encycl. Mater. Tech. Ceram. Glas.* **2021**, 3–3, 311–344. <https://doi.org/10.1016/B978-0-12-803581-8.12132-0>.
- (11) Gromski, P. S.; Henson, A. B.; Granda, J. M.; Cronin, L. How to Explore Chemical Space Using Algorithms and Automation. *Nat. Rev. Chem.* **2019**, *3* (2), 119–128. <https://doi.org/10.1038/s41570-018-0066-y>.
- (12) Maston, G. A.; Evans, S. K.; Green, M. R. Transcriptional Regulatory Elements in the Human Genome. *Annu. Rev. Genomics Hum. Genet.* **2006**, *7*, 29–59. <https://doi.org/10.1146/annurev.genom.7.080505.115623>.
- (13) Reymond, J.-L.; Awale, M. Exploring Chemical Space for Drug Discovery Using the Chemical Universe Database. *ACS Chem. Neurosci.* **2012**, *3* (9), 649–657. <https://doi.org/10.1021/cn3000422>.
- (14) Reymond, J. L. The Chemical Space Project. *Acc. Chem. Res.* **2015**, *48* (3), 722–730. <https://doi.org/10.1021/AR500432K>.
- (15) Drew, K. L. M.; Baiman, H.; Khwaounjoo, P.; Yu, B.; Reynisson, J. Size Estimation of Chemical Space: How Big Is It? *J. Pharm. Pharmacol.* **2012**, *64* (4), 490–495. <https://doi.org/10.1111/J.2042-7158.2011.01424.X>.

- (16) Kim, S.; Chen, J.; Cheng, T.; Gindulyte, A.; He, J.; He, S.; Li, Q.; Shoemaker, B. A.; Thiessen, P. A.; Yu, B.; Zaslavsky, L.; Zhang, J.; Bolton, E. E. PubChem in 2021: New Data Content and Improved Web Interfaces. *Nucleic Acids Res.* **2021**, *49* (D1), D1388–D1395. <https://doi.org/10.1093/nar/gkaa971>.
- (17) ChemSpider | Search and share chemistry <http://www.chemspider.com/> (accessed Jan 17, 2022).
- (18) ZINC <https://zinc.docking.org/> (accessed Jan 17, 2022).
- (19) Dunham, I.; Kundaje, A.; Aldred, S. F.; Collins, P. J.; Davis, C. A.; Doyle, F.; Epstein, C. B.; Frietze, S.; Harrow, J.; Kaul, R.; Khatun, J.; Lajoie, B. R.; Landt, S. G.; Lee, B. K.; Pauli, F.; Rosenbloom, K. R.; Sabo, P.; Safi, A.; Sanyal, A.; Shores, N.; Simon, J. M.; Song, L.; Trinklein, N. D.; Altshuler, R. C.; Birney, E.; Brown, J. B.; Cheng, C.; Djebali, S.; Dong, X.; Ernst, J.; Furey, T. S.; Gerstein, M.; Giardine, B.; Greven, M.; Hardison, R. C.; Harris, R. S.; Herrero, J.; Hoffman, M. M.; Iyer, S.; Kellis, M.; Kheradpour, P.; Lassmann, T.; Li, Q.; Lin, X.; Marinov, G. K.; Merkel, A.; Mortazavi, A.; Parker, S. C. J.; Reddy, T. E.; Rozowsky, J.; Schlesinger, F.; Thurman, R. E.; Wang, J.; Ward, L. D.; Whitfield, T. W.; Wilder, S. P.; Wu, W.; Xi, H. S.; Yip, K. Y.; Zhuang, J.; Bernstein, B. E.; Green, E. D.; Gunter, C.; Snyder, M.; Pazin, M. J.; Lowdon, R. F.; Dillon, L. A. L.; Adams, L. B.; Kelly, C. J.; Zhang, J.; Wexler, J. R.; Good, P. J.; Feingold, E. A.; Crawford, G. E.; Dekker, J.; Elnitski, L.; Farnham, P. J.; Giddings, M. C.; Gingeras, T. R.; Guigó, R.; Hubbard, T. J.; Kent, W. J.; Lieb, J. D.; Margulies, E. H.; Myers, R. M.; Stamatoyannopoulos, J. A.; Tenenbaum, S. A.; Weng, Z.; White, K. P.; Wold, B.; Yu, Y.; Wrobel, J.; Risk, B. A.; Gunawardena, H. P.; Kuiper, H. C.; Maier, C. W.; Xie, L.; Chen, X.; Mikkelsen, T. S.; Gillespie, S.; Goren, A.; Ram, O.; Zhang, X.; Wang, L.; Issner, R.; Coyne, M. J.; Durham, T.; Ku, M.; Truong, T.; Eaton, M. L.; Dobin, A.; Tanzer, A.; Lagarde, J.; Lin, W.; Xue, C.; Williams, B. A.; Zaleski, C.; Röder, M.; Kokocinski, F.; Abdelhamid, R. F.; Alioto, T.; Antoshechkin, I.; Baer, M. T.; Batut, P.; Bell, I.; Bell, K.; Chakraborty, S.; Chrast, J.; Curado, J.; Derrien, T.; Drenkow, J.; Dumais, E.; Dumais, J.; Duttagupta, R.; Fastuca, M.; Fejes-Toth, K.; Ferreira, P.; Foissac, S.; Fullwood, M. J.; Gao, H.; Gonzalez, D.; Gordon, A.; Howald, C.; Jha, S.; Johnson, R.; Kapranov, P.; King, B.; Kingswood, C.; Li, G.; Luo, O. J.; Park, E.; Preall, J. B.; Presaud, K.; Ribeca, P.; Robyr, D.; Ruan, X.; Sammeth, M.; Sandhu, K. S.; Schaeffer, L.; See, L. H.; Shahab, A.; Skancke, J.; Suzuki, A. M.; Takahashi, H.; Tilgner, H.; Trout, D.; Walters, N.; Wang, H.; Hayashizaki, Y.; Reymond, A.; Antonarakis, S. E.; Hannon, G. J.; Ruan, Y.; Carninci, P.; Sloan, C. A.; Learned, K.; Malladi, V. S.; Wong, M. C.; Barber, G. P.; Cline, M. S.; Dreszer, T. R.; Heitner, S. G.; Karolchik, D.; Kirkup, V. M.; Meyer, L. R.; Long, J. C.; Maddren, M.; Raney, B. J.; Grassegger, L. L.; Giresi, P. G.; Battenhouse, A.; Sheffield, N. C.; Showers, K. A.; London, D.; Bhinge, A. A.; Shestak, C.; Schaner, M. R.; Kim, S. K.; Zhang, Z. Z.; Mieczkowski, P. A.; Mieczkowska, J. O.; Liu, Z.; McDaniell, R. M.; Ni, Y.; Rashid, N. U.; Kim, M. J.; Adar, S.; Zhang, Z.; Wang, T.; Winter, D.; Keefe, D.; Iyer, V. R.; Zheng, M.; Wang, P.; Gertz, J.; Vielmetter, J.; Partridge, E. C.; Varley, K. E.; Gasper, C.; Bansal, A.; Pepke, S.; Jain, P.; Amrhein, H.; Bowling, K. M.; Anaya, M.; Cross, M. K.; Muratet, M. A.; Newberry, K. M.; McCue, K.; Nesmith, A. S.; Fisher-Aylor, K. I.; Pusey, B.; DeSalvo, G.; Parker, S. L.; Balasubramanian, S.; Davis, N. S.; Meadows, S. K.; Eggleston, T.; Newberry, J. S.; Levy, S. E.; Absher, D. M.; Wong, W. H.; Blow, M. J.; Visel, A.; Pennachio, L. A.; Petrykowska, H. M.; Abyzov, A.; Aken, B.; Barrell, D.; Barson, G.; Berry, A.; Bignell, A.; Boychenko, V.; Bussotti, G.; Davidson, C.; Despacio-Reyes, G.; Diekhans, M.; Ezkurdia, I.; Frankish,

- A.; Gilbert, J.; Gonzalez, J. M.; Griffiths, E.; Harte, R.; Hendrix, D. A.; Hunt, T.; Jungreis, I.; Kay, M.; Khurana, E.; Leng, J.; Lin, M. F.; Loveland, J.; Lu, Z.; Manthravadi, D.; Mariotti, M.; Mudge, J.; Mukherjee, G.; Notredame, C.; Pei, B.; Rodriguez, J. M.; Saunders, G.; Sboner, A.; Searle, S.; Sisu, C.; Snow, C.; Steward, C.; Tapanari, E.; Tress, M. L.; Van Baren, M. J.; Washietl, S.; Wilming, L.; Zadissa, A.; Zhang, Z.; Brent, M.; Haussler, D.; Valencia, A.; Addleman, N.; Alexander, R. P.; Auerbach, R. K.; Balasubramanian, S.; Bettinger, K.; Bhardwaj, N.; Boyle, A. P.; Cao, A. R.; Cayting, P.; Charos, A.; Cheng, Y.; Eastman, C.; Euskirchen, G.; Fleming, J. D.; Grubert, F.; Habegger, L.; Hariharan, M.; Harmanci, A.; Iyengar, S.; Jin, V. X.; Karczewski, K. J.; Kasowski, M.; Lacroute, P.; Lam, H.; Lamarre-Vincent, N.; Lian, J.; Lindahl-Allen, M.; Min, R.; Miotto, B.; Monahan, H.; Moqtaderi, Z.; Mu, X. J.; O'Geen, H.; Ouyang, Z.; Patacsil, D.; Raha, D.; Ramirez, L.; Reed, B.; Shi, M.; Slifer, T.; Witt, H.; Wu, L.; Xu, X.; Yan, K. K.; Yang, X.; Struhl, K.; Weissman, S. M.; Penalva, L. O.; Karmakar, S.; Bhanvadia, R. R.; Choudhury, A.; Domanus, M.; Ma, L.; Moran, J.; Victorsen, A.; Auer, T.; Centanin, L.; Eichenlaub, M.; Gruhl, F.; Heermann, S.; Hoeckendorf, B.; Inoue, D.; Kellner, T.; Kirchmaier, S.; Mueller, C.; Reinhardt, R.; Schertel, L.; Schneider, S.; Sinn, R.; Wittbrodt, B.; Wittbrodt, J.; Jain, G.; Balasundaram, G.; Bates, D. L.; Byron, R.; Canfield, T. K.; Diegel, M. J.; Dunn, D.; Ebersol, A. K.; Frum, T.; Garg, K.; Gist, E.; Hansen, R. S.; Boatman, L.; Haugen, E.; Humbert, R.; Johnson, A. K.; Johnson, E. M.; Kuttyavin, T. V.; Lee, K.; Lotakis, D.; Maurano, M. T.; Neph, S. J.; Neri, F. V.; Nguyen, E. D.; Qu, H.; Reynolds, A. P.; Roach, V.; Rynes, E.; Sanchez, M. E.; Sandstrom, R. S.; Shafer, A. O.; Stergachis, A. B.; Thomas, S.; Vernot, B.; Vierstra, J.; Vong, S.; Wang, H.; Weaver, M. A.; Yan, Y.; Zhang, M.; Akey, J. M.; Bender, M.; Dorschner, M. O.; Groudine, M.; MacCoss, M. J.; Navas, P.; Stamatoyannopoulos, G.; Beal, K.; Brazma, A.; Flicek, P.; Johnson, N.; Lusk, M.; Luscombe, N. M.; Sobral, D.; Vaquerizas, J. M.; Batzoglou, S.; Sidow, A.; Hussami, N.; Kyriazopoulou-Panagiotopoulou, S.; Libbrecht, M. W.; Schaub, M. A.; Miller, W.; Bickel, P. J.; Banfai, B.; Boley, N. P.; Huang, H.; Li, J. J.; Noble, W. S.; Bilmes, J. A.; Buske, O. J.; Sahu, A. D.; Kharchenko, P. V.; Park, P. J.; Baker, D.; Taylor, J.; Lochovsky, L. An Integrated Encyclopedia of DNA Elements in the Human Genome. *Nat.* 2012 4897414 **2012**, 489 (7414), 57–74.
<https://doi.org/10.1038/nature11247>.
- (20) Sayers, E. W.; Cavanaugh, M.; Clark, K.; Ostell, J.; Pruitt, K. D.; Karsch-Mizrachi, I. GenBank. *Nucleic Acids Res.* **2020**, 48 (D1), D84–D86.
<https://doi.org/10.1093/NAR/GKZ956>.
- (21) Ogata, H.; Goto, S.; Sato, K.; Fujibuchi, W.; Bono, H.; Kanehisa, M. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res.* **1999**, 27 (1), 29–34.
<https://doi.org/10.1093/NAR/27.1.29>.
- (22) RCSB PDB: Homepage <https://www.rcsb.org/> (accessed Apr 3, 2022).
- (23) Thomas Alva Edison - Oxford Reference
<https://www.oxfordreference.com/view/10.1093/acref/9780191826719.001.0001/q-oro-ed4-00003960> (accessed Apr 3, 2022).
- (24) Krska, S. W.; DiRocco, D. A.; Dreher, S. D.; Shevlin, M. The Evolution of Chemical High-Throughput Experimentation to Address Challenging Problems in Pharmaceutical Synthesis. *Acc. Chem. Res.* **2017**, 50 (12), 2976–2985.
https://doi.org/10.1021/ACS.ACCOUNTS.7B00428/ASSET/IMAGES/ACS.ACCOUNTS.7B00428.SOCIAL.JPEG_V03.

- (25) Hanak, J. J. The “Multiple-Sample Concept” in Materials Research: Synthesis, Compositional Analysis and Testing of Entire Multicomponent Systems. *J. Mater. Sci. 1970 511* **1970**, 5 (11), 964–971. <https://doi.org/10.1007/BF00558177>.
- (26) Xiang, X. D.; Sun, X.; Briceño, G.; Lou, Y.; Wang, K. A.; Chang, H.; Wallace-Freedman, W. G.; Chen, S. W.; Schultz, P. G. A Combinatorial Approach to Materials Discovery. *Science (80-.)*. **1995**, 268 (5218), 1738–1740. <https://doi.org/10.1126/SCIENCE.268.5218.1738>.
- (27) Wildey, M. J.; Haunso, A.; Tudor, M.; Webb, M.; Connick, J. H. High-Throughput Screening. *Annu. Rep. Med. Chem.* **2017**, 50, 149–195. <https://doi.org/10.1016/BS.ARCM.2017.08.004>.
- (28) Chang, H.; Gao, C.; Takeuchi, I.; Yoo, Y.; Wang, J.; Schultz, P. G.; Xiang, X. D.; Sharma, R. P.; Downes, M.; Venkatesan, T. Combinatorial Synthesis and High Throughput Evaluation of Ferroelectric/Dielectric Thin-Film Libraries for Microwave Applications. *Appl. Phys. Lett.* **1998**, 72 (17), 2185. <https://doi.org/10.1063/1.121316>.
- (29) Holzwarth, A.; Schmidt, H.-W.; Maier, W. F. Detection of Catalytic Activity in Combinatorial Libraries of Heterogeneous Catalysts by IR Thermography. *Angew. Chemie Int. Ed.* **1998**, 37 (19), 2644–2647. [https://doi.org/https://doi.org/10.1002/\(SICI\)1521-3773\(19981016\)37:19<2644::AID-ANIE2644>3.0.CO;2-#](https://doi.org/https://doi.org/10.1002/(SICI)1521-3773(19981016)37:19<2644::AID-ANIE2644>3.0.CO;2-#).
- (30) Geysen, H. M.; Meloen, R. H.; Barteling, S. J. Use of Peptide Synthesis to Probe Viral Antigens for Epitopes to a Resolution of a Single Amino Acid. *Proc. Natl. Acad. Sci. U. S. A.* **1984**, 81 (13), 3998–4002. <https://doi.org/10.1073/PNAS.81.13.3998>.
- (31) Lam, K. S.; Renil, M. From Combinatorial Chemistry to Chemical Microarray. *Curr. Opin. Chem. Biol.* **2002**, 6 (3), 353–358. [https://doi.org/10.1016/S1367-5931\(02\)00326-5](https://doi.org/10.1016/S1367-5931(02)00326-5).
- (32) Metzker, M. L. Sequencing Technologies - the next Generation. *Nat. Rev. Genet.* **2010**, 11 (1), 31–46. <https://doi.org/10.1038/NRG2626>.
- (33) Park, P. J. ChIP–Seq: Advantages and Challenges of a Maturing Technology. *Nat. Rev. Genet.* 2009 1010 **2009**, 10 (10), 669–680. <https://doi.org/10.1038/nrg2641>.
- (34) Ansorge, W. J. Next-Generation DNA Sequencing Techniques. *N. Biotechnol.* **2009**, 25 (4), 195–203. <https://doi.org/10.1016/J.NBT.2008.12.009>.
- (35) Song, L.; Crawford, G. E. DNase-Seq: A High-Resolution Technique for Mapping Active Gene Regulatory Elements across the Genome from Mammalian Cells. *Cold Spring Harb. Protoc.* **2010**, 2010 (2), pdb.prot5384. <https://doi.org/10.1101/PDB.PROT5384>.
- (36) Qin, D. Next-Generation Sequencing and Its Clinical Application. *Cancer Biol. Med.* **2019**, 16 (1), 4. <https://doi.org/10.20892/J.ISSN.2095-3941.2018.0055>.
- (37) Vervoort, N.; Goossens, K.; Baeten, M.; Chen, Q. Recent Advances in Analytical Techniques for High Throughput Experimentation. *Anal. Sci. Adv.* **2021**, 2 (3–4), 109–127. <https://doi.org/10.1002/ANSA.202000155>.
- (38) Szabo, A.; Ostlund, N. S. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*; Courier Corporation, 2012.

- (39) The Emergence of Computational Chemistry | Mathematical Challenges from Theoretical/Computational Chemistry | The National Academies Press <https://www.nap.edu/read/4886/chapter/4> (accessed Jan 14, 2022).
- (40) Hafner, J. Ab-Initio Simulations of Materials Using VASP: Density-Functional Theory and Beyond. *J. Comput. Chem.* **2008**, *29* (13), 2044–2078. <https://doi.org/10.1002/JCC.21057>.
- (41) Togo, A.; Tanaka, I. First Principles Phonon Calculations in Materials Science. *Scr. Mater.* **2015**, *108*, 1–5. <https://doi.org/10.1016/J.SCRIPTAMAT.2015.07.021>.
- (42) Van Mourik, T.; Bühl, M.; Gageot, M. P. Density Functional Theory across Chemistry, Physics and Biology. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **2014**, *372* (2011). <https://doi.org/10.1098/RSTA.2012.0488>.
- (43) Kadau, K.; Germann, T. C.; Lomdahl, P. S. MOLECULAR DYNAMICS COMES OF AGE: 320 BILLION ATOM SIMULATION ON BlueGene/L. <http://dx.doi.org/10.1142/S0129183106010182> **2011**, *17* (12), 1755–1761. <https://doi.org/10.1142/S0129183106010182>.
- (44) Yu, L.; Zunger, A. Identification of Potential Photovoltaic Absorbers Based on First-Principles Spectroscopic Screening of Materials. *Phys. Rev. Lett.* **2012**, *108* (6), 068701. <https://doi.org/10.1103/PHYSREVLETT.108.068701>/FIGURES/4/MEDIUM.
- (45) Lin, L. C.; Berger, A. H.; Martin, R. L.; Kim, J.; Swisher, J. A.; Jariwala, K.; Rycroft, C. H.; Bhowan, A. S.; Deem, M. W.; Haranczyk, M.; Smit, B. In Silico Screening of Carbon-Capture Materials. *Nat. Mater.* *2012 117* **2012**, *11* (7), 633–641. <https://doi.org/10.1038/nmat3336>.
- (46) Wilmer, C. E.; Leaf, M.; Lee, C. Y.; Farha, O. K.; Hauser, B. G.; Hupp, J. T.; Snurr, R. Q. Large-Scale Screening of Hypothetical Metal–Organic Frameworks. *Nat. Chem.* *2011 42* **2011**, *4* (2), 83–89. <https://doi.org/10.1038/nchem.1192>.
- (47) Lin, H.; Wray, L. A.; Xia, Y.; Xu, S.; Jia, S.; Cava, R. J.; Bansil, A.; Hasan, M. Z. Half-Heusler Ternary Compounds as New Multifunctional Experimental Platforms for Topological Quantum Phenomena. *Nat. Mater.* *2010 97* **2010**, *9* (7), 546–549. <https://doi.org/10.1038/nmat2771>.
- (48) Ceder, G. Opportunities and Challenges for First-Principles Materials Design and Applications to Li Battery Materials. *MRS Bull.* *2010 359* **2011**, *35* (9), 693–701. <https://doi.org/10.1557/MRS2010.681>.
- (49) Hansen, E. W.; Neurock, M. First-Principles-Based Monte Carlo Simulation of Ethylene Hydrogenation Kinetics on Pd. *J. Catal.* **2000**, *196* (2), 241–252. <https://doi.org/10.1006/JCAT.2000.3018>.
- (50) Greeley, J.; Jaramillo, T. F.; Bonde, J.; Chorkendorff, I.; Nørskov, J. K. Computational High-Throughput Screening of Electrocatalytic Materials for Hydrogen Evolution. *Nat. Mater.* *2006 511* **2006**, *5* (11), 909–913. <https://doi.org/10.1038/nmat1752>.
- (51) Wolverton, C.; Siegel, D. J.; Akbarzadeh, A. R.; Ozoli, V. Discovery of Novel Hydrogen Storage Materials: An Atomic Scale Computational. *J. Phys. Condens. Matter* **2008**, *20* (6), 064228. <https://doi.org/10.1088/0953-8984/20/6/064228>.
- (52) Gronau, G.; Krishnaji, S. T.; Kinahan, M. E.; Giesa, T.; Wong, J. Y.; Kaplan, D. L.; Buehler, M. J. A Review of Combined Experimental and Computational Procedures

- for Assessing Biopolymer Structure–Process–Property Relationships. *Biomaterials* **2012**, *33* (33), 8240–8255. <https://doi.org/10.1016/J.BIOMATERIALS.2012.06.054>.
- (53) Lipinski, C. A.; Lombardo, F.; Dominy, B. W.; Feeney, P. J. Experimental and Computational Approaches to Estimate Solubility and Permeability in Drug Discovery and Development Settings. *Adv. Drug Deliv. Rev.* **1997**, *23* (1–3), 3–25. [https://doi.org/10.1016/S0169-409X\(96\)00423-1](https://doi.org/10.1016/S0169-409X(96)00423-1).
- (54) Kumar, A.; Voet, A.; Zhang, K. Y. J. Fragment Based Drug Design: From Experimental to Computational Approaches. *Curr. Med. Chem.* **2012**, *19* (30), 5128–5147. <https://doi.org/10.2174/092986712803530467>.
- (55) Zhang, X.; Chen, A.; Chen, L.; Zhou, Z. 2D Materials Bridging Experiments and Computations for Electro/Photocatalysis. *Adv. Energy Mater.* **2022**, *12* (4), 2003841. <https://doi.org/10.1002/AENM.202003841>.
- (56) Willett, P. Genetic Algorithms in Molecular Recognition and Design. *Trends Biotechnol.* **1995**, *13* (12), 516–521. [https://doi.org/10.1016/S0167-7799\(00\)89015-0](https://doi.org/10.1016/S0167-7799(00)89015-0).
- (57) Gillet, V. J.; Newell, W.; Mata, P.; Myatt, G.; Sike, S.; Zsoldos, Z.; Johnson, A. P. SPROUT: Recent Developments in the de Novo Design of Molecules. *J. Chem. Inf. Comput. Sci.* **1994**, *34* (1), 207–217. <https://doi.org/10.1021/CI00017A027>.
- (58) Lameijer, E. W.; Kok, J. N.; Bäck, T.; Ijzerman, A. P. The Molecule Evuator. An Interactive Evolutionary Algorithm for the Design of Drug-Like Molecules. *J. Chem. Inf. Model.* **2006**, *46* (2), 545–552. <https://doi.org/10.1021/CI050369D>.
- (59) Firth-Clark, S.; Willems, H. M. G.; Williams, A.; Harris, W. Generation and Selection of Novel Estrogen Receptor Ligands Using the De Novo Structure-Based Design Tool, SkelGen. *J. Chem. Inf. Model.* **2005**, *46* (2), 642–647. <https://doi.org/10.1021/CI0502956>.
- (60) Schneider, G.; Lee, M. L.; Stahl, M.; Schneider, P. De Novo Design of Molecular Architectures by Evolutionary Assembly of Drug-Derived Building Blocks. *J. Comput. Mol. Des.* **2000**, *14* (5), 487–494. <https://doi.org/10.1023/A:1008184403558>.
- (61) Dey, F.; Caflisch, A. Fragment-Based de Novo Ligand Design by Multiobjective Evolutionary Optimization. *J. Chem. Inf. Model.* **2008**, *48* (3), 679–690. https://doi.org/10.1021/CI700424B/SUPPL_FILE/CI700424B-FILE021.PDF.
- (62) Nicolaou, C. A.; Apostolakis, J.; Pattichis, C. S. De Novo Drug Design Using Multiobjective Evolutionary Graphs. *J. Chem. Inf. Model.* **2009**, *49* (2), 295–307. https://doi.org/10.1021/CI800308H/ASSET/IMAGES/CI800308H.SOCIAL.JPEG_V03.
- (63) Brown, N.; McKay, B.; Gilardoni, F.; Gasteiger, J. A Graph-Based Genetic Algorithm and Its Application to the Multiobjective Evolution of Median Molecules. *J. Chem. Inf. Comput. Sci.* **2004**, *44* (3), 1079–1087. <https://doi.org/10.1021/CI034290P>.
- (64) Globus, A. I.; Lawton, J.; Wipke, T. Automatic Molecular Design Using Evolutionary Techniques. *Nanotechnology* **1999**, *10* (3), 290. <https://doi.org/10.1088/0957-4484/10/3/312>.
- (65) Douguet, D.; Thoreau, E.; Grassy, G. A Genetic Algorithm for the Automated Generation of Small Organic Molecules: Drug Design Using an Evolutionary Algorithm. *J. Comput. Mol. Des.* **2000**, *14* (5), 449–466.

<https://doi.org/10.1023/A:1008108423895>.

- (66) Pierce, A. C.; Rao, G.; Bemis, G. W. BREED: Generating Novel Inhibitors through Hybridization of Known Ligands. Application to CDK2, P38, and HIV Protease. *J. Med. Chem.* **2004**, *47* (11), 2768–2775.
<https://doi.org/10.1021/JM030543U/ASSET/IMAGES/MEDIUM/JM030543UN00001.GIF>.
- (67) Andersen, J. L.; Flamm, C.; Merkle, D.; Stadler, P. F. Generic Strategies for Chemical Space Exploration. *Int. J. Comput. Biol. Drug Des.* **2014**, *7* (2–3), 225–258.
<https://doi.org/10.1504/IJCBDD.2014.061649>.
- (68) Leardi, R. Genetic Algorithms in Chemometrics and Chemistry: A Review. *J. Chemom.* **2001**, *15* (7), 559–569. <https://doi.org/10.1002/CEM.651>.
- (69) Yang, X.-S. Genetic Algorithms. *Nature-Inspired Optim. Algorithms* **2021**, 91–100.
<https://doi.org/10.1016/B978-0-12-821986-7.00013-5>.
- (70) Katoch, S.; Chauhan, S. S.; Kumar, V. A Review on Genetic Algorithm: Past, Present, and Future. *Multimed. Tools Appl.* **2021**, *80* (5), 8091–8126.
<https://doi.org/10.1007/S11042-020-10139-6/FIGURES/8>.
- (71) Liu, Y.; Zhao, T.; Ju, W.; Shi, S.; Shi, S.; Shi, S. Materials Discovery and Design Using Machine Learning. *J. Mater.* **2017**, *3* (3), 159–177.
<https://doi.org/10.1016/J.JMAT.2017.08.002>.
- (72) Raccuglia, P.; Elbert, K. C.; Adler, P. D. F.; Falk, C.; Wenny, M. B.; Mollo, A.; Zeller, M.; Friedler, S. A.; Schrier, J.; Norquist, A. J. Machine-Learning-Assisted Materials Discovery Using Failed Experiments. *Nat. 2016 5337601* **2016**, *533* (7601), 73–76.
<https://doi.org/10.1038/nature17439>.
- (73) Nielsen, M. K.; Ahneman, D. T.; Riera, O.; Doyle, A. G. Deoxyfluorination with Sulfonyl Fluorides: Navigating Reaction Space with Machine Learning. *J. Am. Chem. Soc.* **2018**, *140* (15), 5004–5008.
https://doi.org/10.1021/JACS.8B01523/SUPPL_FILE/JA8B01523_SI_002.ZIP.
- (74) Lecun, Y.; Bengio, Y.; Hinton, G.; Gregor K And Lecun, Y.; Icm1; F, K. D.; Philbin, J.; Cvpr; Schuster, M.; Chen, Z. PERSPECTIVES Special Topic: Machine Learning Deep Learning for Natural Language Processing: Advantages and Challenges. *11. Sprechmann P, Bronstein AM Sapiro G. IEEE TPAMI* **2018**, *5* (1), 22–24.
<https://doi.org/10.1093/nsr/nwx099>.
- (75) Fujiyoshi, H.; Hirakawa, T.; Yamashita, T. Deep Learning-Based Image Recognition for Autonomous Driving. *IATSS Res.* **2019**, *43* (4), 244–252.
<https://doi.org/10.1016/J.IATSSR.2019.11.008>.
- (76) Hähnel, P.; Mareček, J.; Monteil, J.; O’Donncha, F. Using Deep Learning to Extend the Range of Air Pollution Monitoring and Forecasting. *J. Comput. Phys.* **2020**, *408*, 109278. <https://doi.org/10.1016/J.JCP.2020.109278>.
- (77) Alotto, P.; Guarneri, M.; Moro, F. Redox Flow Batteries for the Storage of Renewable Energy: A Review. *Renew. Sustain. Energy Rev.* **2014**, *29*, 325–335.
<https://doi.org/10.1016/j.rser.2013.08.001>.
- (78) da Silva Lima, L.; Quartier, M.; Buchmayr, A.; Sanjuan-Delmás, D.; Laget, H.; Corbisier, D.; Mertens, J.; Dewulf, J. Life Cycle Assessment of Lithium-Ion Batteries

- and Vanadium Redox Flow Batteries-Based Renewable Energy Storage Systems. *Sustain. Energy Technol. Assessments* **2021**, *46*, 101286. <https://doi.org/10.1016/j.seta.2021.101286>.
- (79) Díaz-Ramírez, M. C.; Ferreira, V. J.; García-Armingol, T.; López-Sabirón, A. M.; Ferreira, G. Environmental Assessment of Electrochemical Energy Storage Device Manufacturing to Identify Drivers for Attaining Goals of Sustainable Materials 4.0. *Sustain.* **2020**, *Vol. 12*, Page 342 **2020**, *12* (1), 342. <https://doi.org/10.3390/SU12010342>.
- (80) Dunn, B.; Kamath, H.; Tarascon, J.-M. Electrical Energy Storage for the Grid: A Battery of Choices. *Science* (80-.). **2011**, *334* (6058), 928–935. <https://doi.org/10.1126/science.1212741>.
- (81) Sánchez-Díez, E.; Ventosa, E.; Guarnieri, M.; Trovò, A.; Flox, C.; Marcilla, R.; Soavi, F.; Mazur, P.; Aranzabe, E.; Ferret, R. Redox Flow Batteries: Status and Perspective towards Sustainable Stationary Energy Storage. *J. Power Sources* **2021**, *481*, 228804. <https://doi.org/10.1016/j.jpowsour.2020.228804>.
- (82) Weber, A. Z.; Mench, M. M.; Meyers, J. P.; Ross, P. N.; Gostick, J. T.; Liu, Q. Redox Flow Batteries: A Review. *J. Appl. Electrochem.* **2011**, *41* (10), 1137–1164. <https://doi.org/10.1007/S10800-011-0348-2/FIGURES/15>.
- (83) Schwenzer, B.; Zhang, J.; Kim, S.; Li, L.; Liu, J.; Yang, Z. Membrane Development for Vanadium Redox Flow Batteries. *ChemSusChem* **2011**, *4* (10), 1388–1406. <https://doi.org/10.1002/CSSC.201100068>.
- (84) Ponce de León, C.; Frías-Ferrer, A.; González-García, J.; Szánto, D. A.; Walsh, F. C. Redox Flow Cells for Energy Conversion. *J. Power Sources* **2006**, *160* (1), 716–732. <https://doi.org/10.1016/J.JPOWSOUR.2006.02.095>.
- (85) Qi, Z.; Koenig, G. M. Review Article: Flow Battery Systems with Solid Electroactive Materials. *J. Vac. Sci. Technol. B, Nanotechnol. Microelectron. Mater. Process. Meas. Phenom.* **2017**, *35* (4), 040801. <https://doi.org/10.1116/1.4983210>.
- (86) Skyllas-Kazacos, M.; Chakrabarti, M. H.; Hajimolana, S. A.; Mjalli, F. S.; Saleem, M. Progress in Flow Battery Research and Development. *J. Electrochem. Soc.* **2011**, *158* (8), R55. <https://doi.org/10.1149/1.3599565>.
- (87) Kear, G.; Shah, A. A.; Walsh, F. C. Development of the All-Vanadium Redox Flow Battery for Energy Storage: A Review of Technological, Financial and Policy Aspects. *Int. J. Energy Res.* **2012**, *36* (11), 1105–1120. <https://doi.org/10.1002/er.1863>.
- (88) Bartolozzi, M. Development of Redox Flow Batteries. A Historical Bibliography. *J. Power Sources* **1989**, *27* (3), 219–234. [https://doi.org/10.1016/0378-7753\(89\)80037-0](https://doi.org/10.1016/0378-7753(89)80037-0).
- (89) THALLER, L. Redox Flow Cell Energy Storage Systems. **1979**. <https://doi.org/10.2514/6.1979-989>.
- (90) Shigematsu, T. SEI Technical Rev. **2011**.
- (91) Leung, P.; Li, X.; Ponce De León, C.; Berlouis, L.; Low, C. T. J.; Walsh, F. C. Progress in Redox Flow Batteries, Remaining Challenges and Their Applications in Energy Storage. *RSC Adv.* **2012**, *2* (27), 10125–10156. <https://doi.org/10.1039/c2ra21342g>.

- (92) Genetics Basics | CDC <https://www.cdc.gov/genomics/about/basics.htm> (accessed Apr 4, 2022).
- (93) Transcription <https://www.genome.gov/genetics-glossary/Transcription> (accessed Apr 4, 2022).
- (94) Translation <https://www.genome.gov/genetics-glossary/Translation> (accessed Apr 4, 2022).
- (95) Regulation of Transcription in Eukaryotes - The Cell - NCBI Bookshelf <https://www.ncbi.nlm.nih.gov/books/NBK9904/> (accessed Apr 4, 2022).
- (96) Nikolov, D. B.; Burley, S. K. RNA Polymerase II Transcription Initiation: A Structural View. *Proc. Natl. Acad. Sci. U. S. A.* **1997**, *94* (1), 15–22. <https://doi.org/10.1073/PNAS.94.1.15/ASSET/BC607FD5-4F96-441E-8A0D-1F67DBF7083A/ASSETS/GRAPHIC/PQ0173055004.JPEG>.
- (97) Jacob, F.; Monod, J. Genetic Regulatory Mechanisms in the Synthesis of Proteins. *J. Mol. Biol.* **1961**, *3* (3), 318–356. [https://doi.org/10.1016/S0022-2836\(61\)80072-7](https://doi.org/10.1016/S0022-2836(61)80072-7).
- (98) Riethoven, J.-J. M. Regulatory Regions in DNA: Promoters, Enhancers, Silencers, and Insulators BT - Computational Biology of Transcription Factor Binding. *Methods Mol Biol* **2010**, *674*, 33–42.
- (99) Smale, S. T.; Kadonaga, J. T. The RNA Polymerase II Core Promoter. <http://dx.doi.org/10.1146/annurev.biochem.72.121801.161520> **2003**, *72*, 449–479. <https://doi.org/10.1146/ANNUREV.BIOCHEM.72.121801.161520>.
- (100) Atchison, M. L. Enhancers: Mechanisms of Action and Cell Specificity. <http://dx.doi.org/10.1146/annurev.cb.04.110188.001015> **2003**, *4*, 127–153. <https://doi.org/10.1146/ANNUREV.CB.04.110188.001015>.
- (101) Banerji, J.; Rusconi, S.; Schaffner, W. Expression of a β -Globin Gene Is Enhanced by Remote SV40 DNA Sequences. *Cell* **1981**, *27* (2), 299–308. [https://doi.org/10.1016/0092-8674\(81\)90413-X](https://doi.org/10.1016/0092-8674(81)90413-X).
- (102) Field, A.; Adelman, K. Evaluating Enhancer Function and Transcription. <https://doi.org/10.1146/annurev-biochem-011420-095916> **2020**, *89*, 213–234. <https://doi.org/10.1146/ANNUREV-BIOCHEM-011420-095916>.
- (103) Reményi, A.; Schöler, H. R.; Wilmanns, M. Combinatorial Control of Gene Expression. *Nat. Struct. Mol. Biol.* *2004* **119** **2004**, *11* (9), 812–815. <https://doi.org/10.1038/nsmb820>.
- (104) Ogbourne, S.; Antalis, T. M. Transcriptional Control and the Role of Silencers in Transcriptional Regulation in Eukaryotes. *Biochem. J.* **1998**, *331* (1), 1–14. <https://doi.org/10.1042/BJ3310001>.
- (105) Privalsky, M. L. The Role of Corepressors in Transcriptional Regulation by Nuclear Hormone Receptors. <http://dx.doi.org/10.1146/annurev.physiol.66.032802.155556> **2004**, *66*, 315–360. <https://doi.org/10.1146/ANNUREV.PHYSIOL.66.032802.155556>.
- (106) Harris, M. B.; Mostecky, J.; Rothman, P. B. Repression of an Interleukin-4-Responsive Promoter Requires Cooperative BCL-6 Function *. *J. Biol. Chem.* **2005**, *280* (13), 13114–13121. <https://doi.org/10.1074/JBC.M412649200>.

- (107) Li, L.; He, S.; Sun, J. M.; Davie, J. R. Gene Regulation by Sp1 and Sp3. <https://doi.org/10.1139/o04-045> **2011**, 82 (4), 460–471. <https://doi.org/10.1139/O04-045>.
- (108) Srinivasan, L.; Atchison, M. L. YY1 DNA Binding and PcG Recruitment Requires CtBP. *Genes Dev.* **2004**, 18 (21), 2596–2601. <https://doi.org/10.1101/GAD.1228204>.
- (109) Chen, L.; Widom, J. Mechanism of Transcriptional Silencing in Yeast. *Cell* **2005**, 120 (1), 37–48. <https://doi.org/10.1016/J.CELL.2004.11.030/ATTACHMENT/00D1781C-8B43-4431-93B8-B1C4341008BF/MMC3.PDF>.
- (110) Gaszner, M.; Felsenfeld, G. Insulators: Exploiting Transcriptional and Epigenetic Mechanisms. *Nat. Rev. Genet.* 2006 79 **2006**, 7 (9), 703–713. <https://doi.org/10.1038/nrg1925>.
- (111) Li, Q.; Peterson, K. R.; Fang, X.; Stamatoyannopoulos, G. Locus Control Regions. *Blood* **2002**, 100 (9), 3077–3086. <https://doi.org/10.1182/BLOOD-2002-04-1104>.
- (112) Cook, P. R. Nongenic Transcription, Gene Regulation and Action at a Distance. *J. Cell Sci.* **2003**, 116 (22), 4483–4491. <https://doi.org/10.1242/JCS.00819>.
- (113) Ali, I.; Yang, W. C. The Functions of Kinesin and Kinesin-Related Proteins in Eukaryotes. *Cell Adh. Migr.* **2020**, 14 (1), 139. <https://doi.org/10.1080/19336918.2020.1810939>.
- (114) Lafontaine, D. L. J.; Tollervey, D. The Function and Synthesis of Ribosomes. *Nat. Rev. Mol. Cell Biol.* 2001 27 **2001**, 2 (7), 514–520. <https://doi.org/10.1038/35080045>.
- (115) Aizawa, S. I. Flagella. *Mol. Med. Microbiol. Second Ed.* **2015**, 1–3, 125–146. <https://doi.org/10.1016/B978-0-12-397169-2.00007-X>.
- (116) Wagoner, J. A.; Dill, K. A. Mechanisms for Achieving High Speed and Efficiency in Biomolecular Machines. *Proc. Natl. Acad. Sci. U. S. A.* **2019**, 116 (13), 5902–5907. https://doi.org/10.1073/PNAS.1812149116/SUPPL_FILE/PNAS.1812149116.SAPP.PDF.
- (117) Balzani, V.; Credi, A.; Raymo, F. M.; Stoddart, J. F. Artificial Molecular Machines. *Angew. Chemie Int. Ed.* **2000**, 39 (19), 3348–3391. [https://doi.org/https://doi.org/10.1002/1521-3773\(20001002\)39:19<3348::AID-ANIE3348>3.0.CO;2-X](https://doi.org/https://doi.org/10.1002/1521-3773(20001002)39:19<3348::AID-ANIE3348>3.0.CO;2-X).
- (118) Deckers-Hebestreit, G.; Altendorf, K. THE F0F1-TYPE ATP SYNTHASES OF BACTERIA: Structure and Function of the F0 Complex. <https://doi.org/10.1146/annurev.micro.50.1.791> **2003**, 50, 791–824. <https://doi.org/10.1146/ANNUREV.MICRO.50.1.791>.
- (119) Howard, J. Molecular Motors: Structural Adaptations to Cellular Functions. *Nat.* 1997 3896651 **1997**, 389 (6651), 561–567. <https://doi.org/10.1038/39247>.
- (120) Schill, G.; Henschel, R.; Neubauer, H.; Ziircher, C.; Vetter, W.; Beckmann, W.; Schweichert, N.; Fritz, H.; Harrison, I. T.; Harrison, S.; Ogino, H.; Ogina, H.; Ohata, K.; Yamanari, K.; Shimura, Y.; Rao, T. V. S.; Lawrence, D. S. A Molecular Shuttle. *J. Am. Chem. Soc.* **1991**, 113 (13), 5131–5133. <https://doi.org/10.1021/JA00013A096>.
- (121) Chambron, J. C.; Dietrich-Buchecker, C.; Hemmert, C.; Khemiss, A. K.; Mitchell, D.; Sauvage, J. P.; Weiss, J. Interlacing Molecular Threads on Transition Metals. *Pure*

- Appl. Chem.* **1990**, 62 (6), 1027–1034. <https://doi.org/10.1351/PAC199062061027>.
- (122) Kay, E. R.; Leigh, D. A. Rise of the Molecular Machines. *Angew. Chemie - Int. Ed.* **2015**, 54 (35), 10080–10088. <https://doi.org/10.1002/anie.201503375>.
- (123) Koumura, N.; Zijlstra, R. W. J.; Van Delden, R. A.; Harada, N.; Feringa, B. L. Light-Driven Monodirectional Molecular Rotor. *Nat. 1999 4016749* **1999**, 401 (6749), 152–155. <https://doi.org/10.1038/43646>.
- (124) Mavroidis, C.; Dubey, A.; Yarmush, M. L. Molecular Machines. *Annual Review of Biomedical Engineering*. Annual Reviews July 15, 2004, pp 363–395. <https://doi.org/10.1146/annurev.bioeng.6.040803.140143>.
- (125) Kay, E. R.; Leigh, D. A.; Zerbetto, F. Synthetic Molecular Motors and Mechanical Machines. *Angew. Chemie Int. Ed.* **2007**, 46 (1–2), 72–191. <https://doi.org/10.1002/ANIE.200504313>.
- (126) Shi, Z.-T.; Zhang, Q.; Tian, H.; Qu, D.-H. Driving Smart Molecular Systems by Artificial Molecular Machines. *Adv. Intell. Syst.* **2020**, 2 (5), 1900169. <https://doi.org/10.1002/AISY.201900169>.
- (127) Norikane, Y.; Tamaoki, N. Light-Driven Molecular Hinge: A New Molecular Machine Showing a Light-Intensity-Dependent Photoresponse That Utilizes the Trans-Cis Isomerization of Azobenzene. *Org. Lett.* **2004**, 6 (15), 2595–2598. https://doi.org/10.1021/OL049082C/SUPPL_FILE/OL049082CSI20040608_100711.PDF.
- (128) Zheng, J.; Zheng, X.; Zhao, Y.; Xie, Y.; Yam, C.; Chen, G.; Jiang, Q.; Chwang, A. T. Maxwell's Demon and Smoluchowski's Trap Door. *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.* **2007**, 75 (4), 041109. <https://doi.org/10.1103/PHYSREVE.75.041109/FIGURES/6/MEDIUM>.
- (129) Magnasco, M. O.; Stolovitzky, G. Feynman's Ratchet and Pawl. *J. Stat. Phys.* 1998 933 **1998**, 93 (3), 615–632. <https://doi.org/10.1023/B:JOSS.0000033245.43421.14>.
- (130) Astumian, R. D.; Hänggi, P. Brownian Motors. *Phys. Today* **2007**, 55 (11), 33. <https://doi.org/10.1063/1.1535005>.
- (131) Baroncini, M.; Casimiro, L.; de Vet, C.; Groppi, J.; Silvi, S.; Credi, A. Making and Operating Molecular Machines: A Multidisciplinary Challenge. *ChemistryOpen* **2018**, 7 (2), 169–179. <https://doi.org/10.1002/OPEN.201700181>.
- (132) Wilson, M. R.; Solà, J.; Carlone, A.; Goldup, S. M.; Lebrasseur, N.; Leigh, D. A. An Autonomous Chemically Fuelled Small-Molecule Motor. *Nat. 2016 5347606* **2016**, 534 (7606), 235–240. <https://doi.org/10.1038/nature18013>.

Chapter 2
Fundamentals of Machine Learning

Chapter 2

Fundamentals of Machine Learning

Abstract

Machine Learning (ML) is a subfield of Artificial Intelligence (AI), which can be defined as an ability of a system to extract knowledge from the data rather than being explicitly programmed. With a rapid increase in storage capacity and processing power, machine learning has shown impressive results in many fields of science and technology that surpass traditional methods. Machine learning algorithms have been successfully applied to image recognition, natural language processing, robotics, protein folding, and the prediction of novel materials. Machine learning algorithms are particularly advantageous when expert knowledge is unavailable or incomplete. In this chapter, we have briefly discussed the fundamental concepts of machine learning. First, we give a brief introduction to machine learning and related fields. Then, we discuss a brief history of machine learning, three types of common machine learning problems, and machine learning algorithms. A typical workflow of a machine project has been discussed next, followed by a brief description of some popular machine learning algorithms. Finally, computational methods used in this thesis work have been discussed.

2.1 Introduction

The past decade has seen an immense increase in the techniques and applications powered by machine learning (ML). It has impacted many industries, including autonomous driving, health care, finance, manufacturing, and energy harvesting.¹⁻⁴ At the same time, researchers around the world are using machine learning to discover new phenomena or further the understanding of the known scientific phenomena. Machine learning is considered as a new paradigm in science, and one of the disruptive technologies of this age.⁵ Machine learning has been successfully applied to seemingly unsolvable problems such as protein folding.⁶ It aims to extract knowledge from the data and utilize it to make decisions on new, unseen data. Machine learning generally requires a large amount of data. Therefore, the success of ML in recent years could be attributed to the generation of huge datasets and improvements in the technologies for data management.⁷ The considerable number of applications shows that the industry has significantly benefited from machine learning. However, the spread of ML techniques in the scientific community took some time due to a fundamental difference in goals. Scientists want to understand the mechanism behind a phenomenon and are interested in building intuitive, interpretable models. On the other hand, machine learning does the opposite: most machine learning algorithms create very complex models, making it difficult to extract and interpret the learned knowledge. ML models are generally considered a black box. Nevertheless, one cannot deny the power of machine learning to produce surprisingly good results that surpass traditional methods.

Learning is the process of acquiring new behaviors or modifying existing behaviors, values, and knowledge. The theory of personal learning states that humans learn from their past experiences. The field of artificial intelligence aims to develop a system capable of thinking and learning on its own. Research into artificial intelligence led to the development of machine learning algorithms that can learn from data. In his famous book, Tom Mitchell defines machine learning as follows:⁸

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E .

In simple words, it means learning from data. Data is analogous to past experience in human learning. Machine learning is also known as statistical learning due to its foundation in statistics. Before moving forward, it is essential to distinguish between artificial intelligence, machine learning, deep learning, and data science. Figure 2.1 depicts how the artificial intelligence, machine learning, and deep learning fields relate to each other.⁹

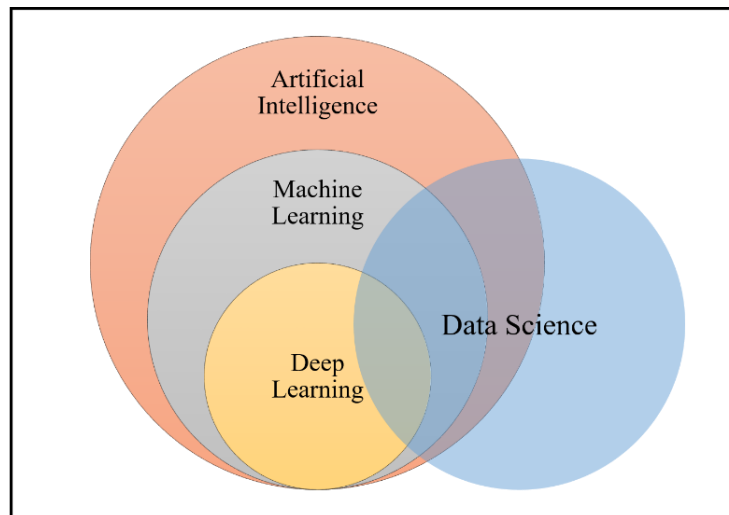


Figure 2.1. Relationship between artificial intelligence, machine learning, deep learning, and data science.

- **Artificial Intelligence (AI):** Artificial intelligence is the ability of machines to learn and understand data, make informative decisions by discovering hidden patterns in the data, and make inferences that could otherwise be challenging for humans to make manually. AI enables machines to adapt to a new situation that has not been encountered previously. In other words, AI is a collection of mathematical algorithms that tries to simulate human intelligence and impart computers an ability to comprehend relationships between various types of data and extract knowledge from them to make conclusions or decisions that are most likely to achieve the desired goal.
- **Machine Learning (ML):** Machine learning is a subfield of AI. It consists of algorithms that can learn hidden patterns from massive amounts of data. Learning from past experience is an essential property of human intelligence. ML algorithms impart a similar ability to an intelligent system. In ML algorithms, the data used for learning is known as training data. The uniqueness of machine learning lies in the ability to understand the general pattern that is not only applicable to training data but also to other unseen data. Machine learning algorithms can be grouped into two categories: (i) Traditional or Shallow learning algorithms and (ii) Deep learning algorithms. Traditional or Shallow learning algorithms are the algorithms developed prior to the advent of deep learning. These algorithms rely more on expert knowledge than deep learning algorithms.
- **Deep Learning (DL):** Deep learning is one of the most popular branches of machine learning. Today, many applications are powered by deep learning algorithms such as face recognition, language translation, and recommendation systems.¹⁰⁻¹² Deep learning algorithms are based on artificial neural networks that resemble neurons in our brain. Deep learning algorithms can make highly accurate predictions, surpassing the performance of conventional methods in various fields. The success of DL algorithms comes from their ability to learn meaningful features directly from the raw data. In contrast, the performance of traditional machine learning algorithms depends on feature engineering, which requires domain knowledge. Feature engineering has been the bottleneck in many machine learning applications. Deep learning essentially removed the need for feature engineering. However,

it requires vast data and computing resources for the training. The availability of large datasets and Graphical Processing Units (GPUs) have fueled the growth of deep learning.

- **Data Science:** On the other hand, data science is a multidisciplinary field that involves extracting insights from large datasets. It uses scientific methods, processes, and algorithms to extract knowledge and insight from noisy, structured, and unstructured data. It combines domain expertise, data analytics skills, programming skills, mathematics, and machine learning to understand data and draw inferences. Data science encompasses preparing data for analysis, formulating problems, analyzing data, and developing data-driven solutions.

It can be seen that deep learning is a subfield of machine learning, which itself is a subfield of the larger field of artificial intelligence. While data science is an entirely different area, it uses some elements from machine learning as well as deep learning. The goal of the machine learning model is to identify patterns from the data to explain unseen data as accurately as possible. Machine learning has found its importance due to its power to identify complex patterns in data. Therefore, difficult tasks could be modeled easily with machine learning. For example, it is tough to program a computer to identify human faces. However, it can easily be done using machine learning algorithms. The machine learning approach is advantageous when:

- It is challenging to construct systems that require specific detailed skills or expert knowledge tuned to a specific task.
- Systems are required to adapt and customize themselves to individual users automatically. For example, personalized email filters.
- New knowledge must be discovered from large databases. For example, medical text mining for disease diagnosis.
- Human expertise does not exist. For example, navigating on mars.
- Humans are unable to explain their expertise. For example, speech recognition.
- Solution changes in time. For example, the stock market.

Thus, machine learning is instrumental when expert knowledge is not available or incomplete, and the data is large or too complex for human analysis.

2.2 Brief History of Machine Learning

Although machine learning has recently gained popularity, the foundation was laid in the mid-20th century when Alan Turing invented the “Turing Test” to test a machine’s ability to exhibit intelligent behavior equivalent to, or indistinguishable from, that of humans.¹³ Later, in 1958, Frank Rosenblatt proposed the perceptron model for face recognition, the basic unit of all deep learning models.¹⁴ In 1967, a simple yet very effective nearest neighbor algorithm was implemented for classification tasks.¹⁵ Around 1969, Seymour Papert and Minsky proved that perceptron was incapable of learning non-linear functions; this led to a decline in AI/machine learning research.¹⁶ In the 1960s, a multilayer perceptron capable of learning non-linear functions was proposed, but the AI community still lacked an efficient algorithm to train multiple layers.¹⁷ The period from 1969 to 1984 is known as the AI winter. Around 1990, the research focus shifted from solving AI to solving practical problems using data; this led to the development of many traditional/shallow learning algorithms (e.g., random forest, SVM,

boosting).¹⁸ The AI winter ended when, in 1986, a simple and efficient algorithm for the training of multilayer neural networks was designed.¹⁹ Backpropagation was a revolutionary idea, but the real success of deep learning was realized in the early 21st century when GPUs were used for training neural networks. Deep learning surpassed all the traditional approaches in image recognition, natural language processing, and even in toxicity prediction.^{10,20,21} These examples demonstrate the power of deep learning models. A brief history of machine learning is shown in Figure 2.2.

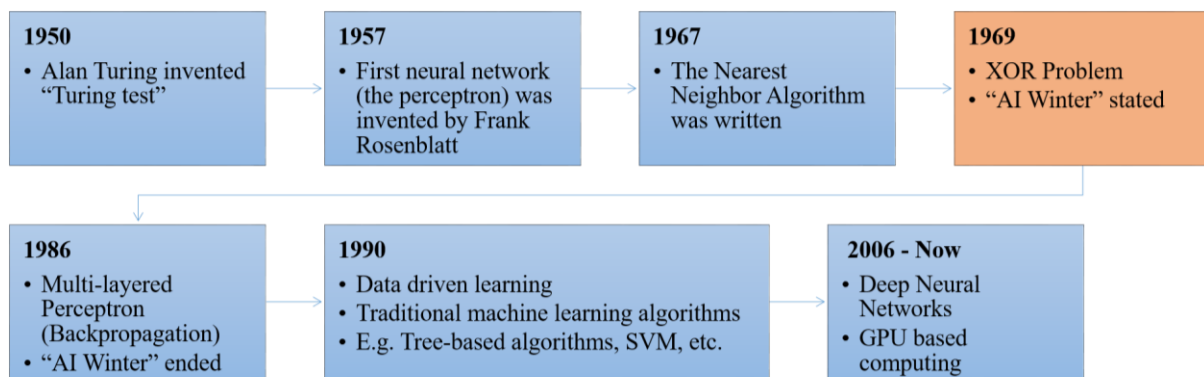


Figure 2.2. A brief history of machine learning.

2.3 Types of Machine Learning Problems

Before applying machine learning algorithms, it is crucial to formulate a given problem into an appropriate type. Machine learning can solve three types of problems (Figure 2.3), which are discussed below.²²

2.3.1 Classification

In this problem, the output variable to be predicted is categorical. The goal of classification is to assign data points to a fixed number of discrete classes such as Yes/No, and Male/Female. The problem could be a binary or multi-class classification problem depending on the number of output classes. For example, classifying incoming emails as spam or ham, face recognition in which output classes are more than two, etc.

2.3.2 Regression

In regression, the output variable to be predicted is continuous, e.g., scores of a student and the weight of a person. Regression involves modeling the target variable based on independent variables. It is used to understand the relationship between target and independent variables.

2.3.3 Clustering

Clustering involves grouping given data into different clusters or categories. The goal of clustering is to group similar data points into the same clusters and segregate dissimilar data points into the clusters that are farthest from each other. There is no pre-defined notion of label allocated to the groups/clusters formed. One needs to inspect the formed clusters to understand their meaning. An example of clustering includes customer segmentation for marketing.

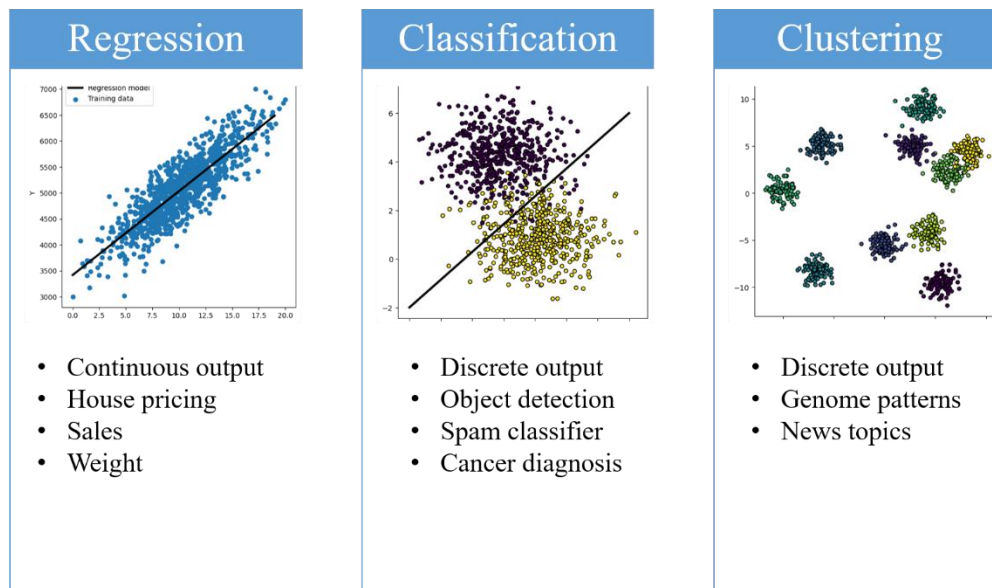


Figure 2.3. Types of machine learning problems.

2.4 Types of Machine Learning Algorithms

Machine learning algorithms can be classified into three categories.²³

2.4.1 Supervised Learning Algorithms

Supervised learning algorithms are the type of machine learning algorithms that use a labeled dataset to learn general patterns describing the given data. This dataset (referred to as the training dataset) includes both the target variable and the input data. From this, the supervised learning algorithm seeks to build a model that can predict the values of the target variable for a new dataset. Supervised models create a mapping between input and target variables. Supervised learning algorithms typically solve classification and regression problems. Some commonly used supervised machine learning algorithms include k-nearest neighbors, naïve Bayes, decision trees, linear regression, support vector machines, and neural networks.^{24–28}

2.4.2 Unsupervised Learning Algorithms

Unsupervised learning algorithms train only on the input data without their labels. Clustering problems are typically solved using unsupervised machine learning algorithms. They separate input data into different groups based on similarity and dissimilarity between data points. Unsupervised machine learning algorithms have also been employed for dimensionality reduction. Unsupervised algorithms find hidden structures present within the data. Examples of unsupervised machine learning algorithms include k-means, principal component analysis, hierarchical clustering, and topic models.^{29–32}

2.4.3 Reinforcement Learning Algorithm

Reinforcement learning algorithms learn by trial and error through interaction with the environment. It involves training an agent using the concept of rewards and penalties without specifying how to accomplish a given task. The reinforcement learning agent learns appropriate

behavior that maximizes the reward from its past experience. These types of algorithms are commonly used in robotics. Some of the reinforcement learning algorithms include Markov decision processes, Deep Q-Network, Proximal Policy Optimization.³³

2.5 Machine Learning Workflow

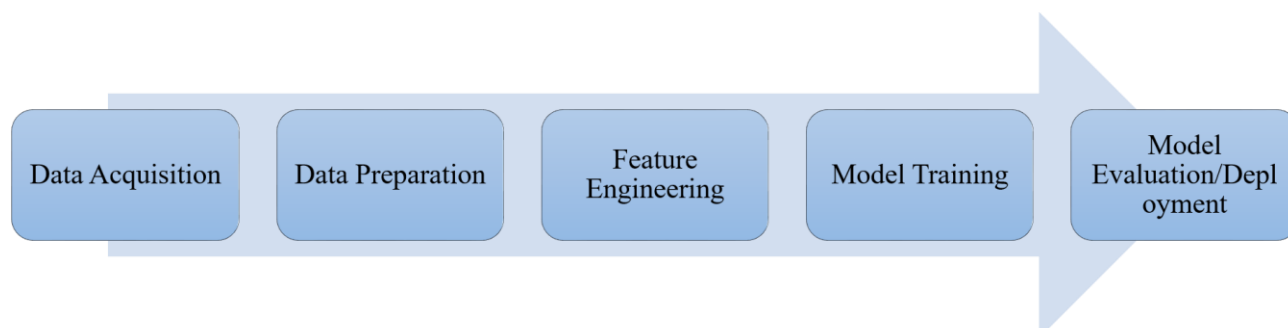


Figure 2.4. Typical machine learning workflow.

2.5.1 Data Collection

This is the very first step in any machine learning project.³⁴ The data for a machine learning project is collected based on the problem. Machine learning models learn from the data; therefore, data quality is by far the most critical aspect of a machine learning project. High-quality data helps the model learn general patterns that apply to unseen data. On the other hand, low-quality data may contain noise that can mislead the algorithm into learning noise rather than a general pattern. Data is selected by considering its type, quality, source, and format. These days thousands of open and private databases are easily accessible. People from various fields have made efforts to compile datasets corresponding to their application domain. These databases provide a good starting point for many machine learning problems. Some famous open data repositories include the UC Irvine Machine Learning Repository³⁵ and Kaggle datasets.³⁶ In some cases, when data is not available, one might need to generate the data through surveys, experimentation, or computational tools. Lack of data is a common situation in fundamental scientific research. Researchers realized this issue and created many open databases such as the Open Quantum Materials Database, the Materials Project, the Harvard Clean Energy Project, the Inorganic Crystal Structure Database, PubChem, ZINC, ENCODE, GenBank, and KEGG.³⁷⁻⁴⁵ In addition to these, text mining has also been used for retrieving data from the literature.⁴⁶

2.5.2 Data Processing

Quite often, the required data is collected from multiple sources. Different databases store data in different formats. Furthermore, the required data format depends on the machine learning algorithm utilized for a given application. For example, convolution neural networks require data in a 2D image format, whereas linear regression needs data represented in a 1D array. When the data has been collected from multiple sources, it becomes crucial to unify their format and select a suitable representation for the machine learning algorithm. Furthermore, data obtained from the web or databases may contain noise, errors, outliers, and invalid values. Such data will make ML algorithms harder to detect underlying patterns. Thus, data needs to be

cleaned before feeding to an ML algorithm. Data cleaning is one of the most time-consuming steps of the machine learning workflow.⁴⁷ Data cleaning does not have a well-defined structure but generally involves the following steps:

Fix rows and columns: Delete summary rows such as total and subtotal rows. Delete unnecessary rows such as header rows and footer rows. Delete extra rows such as column numbers, indicators, blank rows, and page numbers. Merge columns for creating unique identifiers if needed. Eg., merge state and city into a full address. Add column names if missing. Rename columns consistently, such as abbreviations and encoded columns. Delete unnecessary columns. Align misaligned columns when datasets may have shifted columns.

Fix missing values: First, identify values that indicate missing data and are not yet recognized by the software. For example, blank strings, “NA”, “XX”, “999”. Then, one should try replacing missing values from reliable sources as much as possible. However, if one cannot, then it is better to keep missing values as such rather than exaggerating the existing rows/columns. Rows could be deleted if the number of missing values is insignificant, as this would not impact the analysis. Columns could be removed if the missing values are quite significant in number.

Standardize values: We should also ensure that all observations under a variable have a common and consistent unit. Eg., convert lbs to kgs, miles/hr to km/hr, etc. Many machine algorithms (such as linear regression, SVM, k-nearest neighbors, and logistic regression) are sensitive to the scale of features. Therefore, appropriate feature scaling is required for robust prediction. Various methods such as standardization, normalization, and min-max scaling are available for feature scaling.

Fix invalid values: A dataset can contain invalid values in various forms. Some of the values could be truly invalid. For example, a string “tr8ml” in a variable containing mobile numbers would make no sense and hence would be better removed. Similarly, a height of 11 ft would be an invalid value in a set containing the heights of children. On the other hand, some invalid values can be corrected. Eg., a numeric value with a data type of string could be converted to its original numeric type. Issues might arise due to misinterpretation of the encoding of a file, thus showing junk characters where there were valid characters. This could be corrected by specifying the proper encoding or converting the dataset to the accurate format before importing.

Convert Datatypes: Sometime, a feature may have an incorrect datatype. For example, the height, which is a numerical variable, is represented as a string. Furthermore, most machine learning algorithms require all features in integer or float format. Therefore, we should convert variables into their appropriate datatype. Label or one-hot encodings are commonly used for converting categorical variables into numbers.

Filter data: In this step, identical rows and columns are removed. We can also pick rows and columns relevant to the analysis, for example, based on date.

2.5.3 Feature Engineering

The performance of machine learning algorithms, particularly traditional machine learning algorithms, depends on features present in the data. Different features contain different information about the individual data points. Feature engineering is the process of creating new and informative features from raw data. It also involves converting data into a more suitable representation for machine learning algorithms. The suitable representation of the input data dictates the accuracy of the algorithm. In this thesis work, we are dealing with molecules. There are multiple choices available for the representation of molecules, such as SMILES,⁴⁸ InChI,⁴⁹ molecular graphs,⁵⁰ and arrays of molecular features. Arrays of molecular features include one-dimensional arrays of 2D molecular features, 3D molecular features, and molecular fingerprints.^{51–53} Biomolecules such as DNA and RNA are represented as a two-dimensional array of probability weight matrices or one-dimensional arrays of either base pairs or protein binding sites.^{54–56} Thus, feature engineering involves the generation of different molecular features. Feature engineering requires insight into both the scientific problem and the mechanism of a machine learning algorithm. Traditional machine learning algorithms (shallow learning algorithms) require manual feature engineering and selection. Manual feature engineering needs domain expertise that is not always available and has high labor and computational cost. Therefore, manual feature engineering is not always an ideal solution. On the other hand, deep learning algorithms have eliminated the need for manual feature engineering. These algorithms can identify and generate important features directly from the raw data.⁵⁷

2.5.4 Model Training

This step constitutes the selection of appropriate algorithms, model training, and hyperparameter tuning. Before the training, the data is split into three sets: (i) training-set, (ii) test-set, and (iii) validation-set. If the size of the data is small, then the validation-set is generally not created. The typical size of the validation and test set ranges somewhere from 30% to 5%, depending on the size of the whole dataset. Algorithm selection depends on the problem, data, and the size of the training-set. For the prediction problem with labeled data, we would choose one of the supervised learning algorithms. If we want to identify groups within data, unsupervised machine learning algorithms are more suitable. Although there are no rules for model selection, general guidelines exist. One should investigate deep learning models along with others when the size of the training-set is sufficiently large.²² On the other hand, for small datasets, traditional or shallow learning algorithms are more suitable. Training a machine learning algorithm involves adjusting model parameters to achieve high performance on the training data defined by a loss or cost function. The cost function measures the amount of deviation between model predictions and the actual values. It guides the ML algorithm during the search for parameters. Parameter search in the ML algorithm is carried out using some optimization method. For example, gradient descent is the most commonly used optimization method in ML algorithms.⁵⁸ Apart from parameters, ML models also contain other parameters known as hyperparameters that are not part of the model itself. However, they still have an impact on training and prediction. Hyperparameters are not updated during the training. Thus, they require manual tuning. Hyperparameter tuning is generally performed with a validation-set when the size of the dataset is large; otherwise, a training-set is used.⁵⁹

2.5.5 Model Evaluation / Deployment

After training, we need a way to estimate the reliability of the models in test scenarios not covered by the training data. “Hold-Out Strategy” and “Cross-Validation” are the two commonly employed methods for assessing model performance.⁶⁰ The basic idea is the same in both methods – to keep aside some data that will not in any way influence the model building. The part of the data that is kept aside is then used as a ‘proxy’ for the unknown (as far as the model we have built is concerned) test data on which we want to estimate the performance of the model. Finally, the performance of the trained ML model is evaluated on the test-set. It is crucial to evaluate a model on the test-set to understand its generalizability. If the performance on the test-set is satisfactory, then we can deploy the model to obtain predictions on the new datasets. Model deployment is particularly important in industries, whereas model evaluation is generally the last step in research domains. Some commonly used evaluation metrics for regression and classification are given in Table 2.1 below. In the formulas below, N represents the total number of data points, \hat{y}_i denotes the predicted value of i -th sample and the y_i denotes the corresponding true value.

Table 2.1. Some commonly used evaluation metrics in regression and classification tasks.

Regression	$\text{Coefficient of Determination } (R^2) = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2}$
	$\text{Mean Squared Error } (MSE) = \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{N}$
	$\text{Mean Absolute Error } (MAE) = \frac{\sum_{i=1}^N y_i - \hat{y}_i }{N}$
	$\text{Mean Absolute Percentage Error } (MAPE) = 100 * \frac{\sum_{i=1}^N \frac{ y_i - \hat{y}_i }{ y_i }}{N}$
Classification	$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$ <p>where, TP is the number of true positives TN is the number of true negatives FP is the number of false positives and FN is the number of false negatives</p>
	$\text{Precision} = \frac{TP}{TP + FP}$
	$\text{Recall} = \frac{TP}{TP + FN}$
	$\text{Specificity} = \frac{TN}{TN + FP}$
	$F1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$
	AUC (Area Under The Curve)- ROC (Receiver Operating Characteristics) curve

2.5.6 Model Selection

Model selection is not depicted as a separate step in Figure 2.4. However, it is often part of either model training or evaluation steps. Here, we briefly discuss an important aspect of model selection — the bias-variance trade-off. The central issue in all of the machine learning is “how do we extrapolate what has been learned from a finite amount of data to all possible inputs ‘of the same kind’?”. We build models from some training data. However, the training data is always finite. On the other hand, the model is expected to have learned ‘enough’ about the entire domain from where the data points can possibly come. Clearly, in almost all realistic scenarios, the domain is infinitely large. How do we ensure that our model is as good as we think based on its performance on the training data, even when we apply it to the infinitely many data points that the model has never ‘seen’ (been trained on)? A predictive model has to be as simple as possible, but no simpler. Often referred to as the Occam’s Razor, this is not just a convenience but a fundamental tenet of all machine learning. Overfitting is a phenomenon where a model becomes way too complex than what is warranted for the task at hand and, as a result, suffers from bad generalization properties. Overfitting is generally addressed through regularization techniques. By contrast, an underfitted model fails to capture the relationships between the variables in the data adequately. This could be due to an incorrect choice of model type, incomplete or incorrect assumptions about the data, too few parameters in the model, or an incomplete training process. The ‘variance’ of a model is the variance in its output on some test data with respect to changes in the training dataset. In other words, variance refers to the degree of changes in the model itself with respect to changes in the training data. Bias quantifies how accurate the model is likely to be on future (test) data. The left-hand side of Figure 2.5 illustrates bias and variance using a target shooting analogy. A model whose shots are clustered together is one that has a small variance and one whose shots are close to the “bull’s eye” has a small bias. A ‘consistent’ shooter will have a small variance, and a ‘good’ shooter will have a small bias. So, in other words, variance is about consistency, and bias is about accuracy. The right-hand side of Figure 2.5 illustrates the typical trade-off between bias and variance — low complexity models have high bias, and low variance and high complexity models have low bias but high variance. The goal of model selection is to find a ‘best’ model that balances both bias and variance and achieves a reasonable degree of predictability (low variance) without compromising too much on the accuracy (bias).⁶¹ Model selection generally includes feature selection, hyperparameter tuning, and regularization.

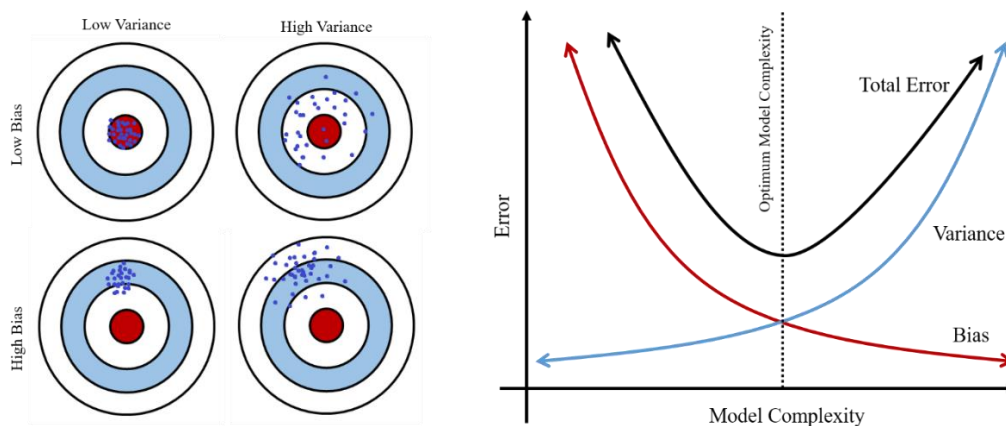


Figure 2.5. Bias-variance trade-off.

2.6 Brief Description of Commonly Used Machine Learning Algorithms

2.6.1 Linear Regression

Linear regression is a regression technique (model) that assumes a linear relationship between the predictor and the target variable.⁶² It is one of the simplest machine learning algorithms used when the target variable depends linearly on independent variables. The target variable is represented as a linear combination of independent variables in linear regression. Based on the number of independent variables, there are two types of linear regression (i) Simple Linear Regression and (ii) Multiple linear regression.

- **Simple Linear Regression:** The linear regression model with only one independent variable is known as a simple linear regression. The relationship between the target and independent variable is given by the following equation of a straight line:

$$y = \beta_0 + \beta_1 x$$

- **Multiple Linear Regression:** When there are several independent variables in the linear regression model, it is called multiple linear regression.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n$$

In the equations above, y is the dependent variable, x_i are the independent variables and β_i are weights or coefficients of the regression model.

Linear regression also assumes that the error terms are normally distributed, independent of each other, and have constant variance (homoscedasticity). The goal of linear regression is to find the best-fit line or hyperplane that captures the linear relationship between target and independent variables, as shown in Figure 2.6.

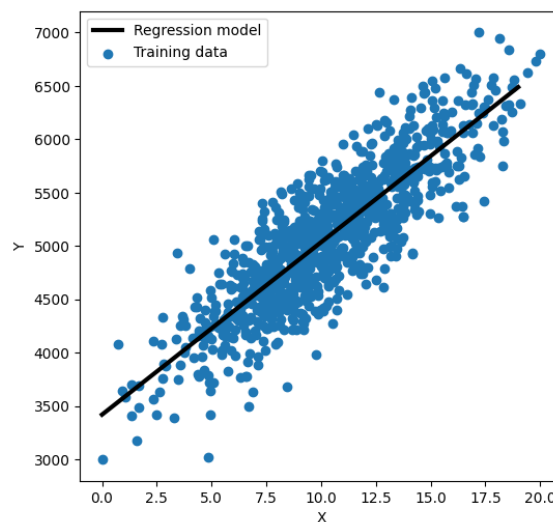


Figure 2.6. Visualizing linear regression in two dimensions.

The best-fit line or hyperplane corresponds to a set of coefficients having minimum prediction error. The coefficients are found by minimizing the expression of RSS (Residual Sum of Squares) which is equal to the sum of squares of the residual for each data point. The residual

for any data point is computed by subtracting the predicted value of the target variable from the actual value of the target variable, as given below:

$$\text{Residuals} = y_{i,true} - y_{i,pred}$$

Training linear regression model involves minimizing a cost function. The cost function in linear regression is RSS (Residual Sum of Squares):

$$\text{Linear Regression Cost Function (RSS)} = \sum_{i=1}^N (y_{i,true} - y_{i,pred})^2 = \|Y - X\hat{\beta}\|^2$$

where,

N is the number of training samples.

$y_{i,true}$ is the true value of the target variable corresponding to training sample i .

$y_{i,pred}$ is the predicted value of the target variable corresponding to training sample i .

$\hat{\beta}$ is the matrix of coefficients.

X is the matrix of independent variables corresponding to all training samples.

Y is the matrix of target values of all training samples.

We generally employ a gradient descent algorithm to minimize the RSS in linear regression. The output of gradient descent is the set of coefficients (betas) giving the lowest RSS. Gradient descent is the process of optimizing the values of coefficients by iteratively minimizing the cost function on training data. It starts with the random values for each coefficient. The cost is calculated on the training data. The coefficients are updated in the direction of minima using the learning rate as a scaling factor. The process is repeated until RSS goes below some threshold. Gradient descent works very well on a large dataset. However, we can also obtain the coefficients using a closed-form solution for the small datasets. The closed-form solution exists for linear regression, which is given by:

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

2.6.2 Ridge Regression

In linear regression, we get the best coefficients by minimizing the residual sum of squares (RSS). Similarly, with ridge regression, we estimate the model coefficients but by minimizing a different cost function. This cost function adds a penalty term to the RSS. The penalty term is the sum of squared model coefficients (i.e., L2-norm) multiplied by a regularisation parameter alpha.⁶³

$$\begin{aligned} \text{Ridge Regression Cost Function} &= \sum_{i=1}^N (y_{i,true} - y_{i,pred})^2 + \alpha \sum_{j=1}^m \beta_j^2 \\ &= \|Y - X\hat{\beta}\|^2 + \alpha \|\hat{\beta}\|_2^2 \end{aligned}$$

where,

N is the number of training samples.

$y_{i,true}$ is the true value of the target variable corresponding to training sample i .

$y_{i,pred}$ is the predicted value of the target variable corresponding to training sample i .

β_j is the coefficients corresponding to feature j .

X is the matrix of independent variables corresponding to all training samples.

Y is the matrix of target values of all training samples.

α is the coefficient of regularisation.

m is the number of features.

In the cost function, the penalty term, also called the shrinkage penalty, would be small only if the coefficients are small, i.e., close to 0. Hence, while fitting the ridge regression model, since we need to find out the model coefficients that minimize the entire cost, i.e., RSS and a penalty, it would have the effect of shrinking the model coefficients, i.e., the betas, towards 0. If alpha is 0, then the cost function would not contain the penalty term, and there will be no shrinkage of the model coefficients. They would be the same as those from linear regression. However, since alpha moves towards higher values, the shrinkage penalty increases, pushing the coefficients further towards zero, which may lead to model underfitting. Choosing an appropriate alpha becomes crucial: if it is too small, then we would not be able to solve the problem of overfitting, and with too large an alpha, we may actually end up underfitting. Ridge regression is helpful in scenarios where independent variables are highly correlated. It pushes the coefficients of unimportant features to zero, thereby reducing the overfitting and improving generalizability.

2.6.3 Lasso Regression

Ridge regression retains all the variables that are present in the data. Now, when the number of variables is very large, and the data may have unrelated or noisy variables, we may not want to keep such variables in the model. Lasso regression helps us here by performing feature selection. The primary difference between lasso and ridge regression is their penalty term. Here, the penalty term is the sum of the absolute values of all the coefficients present in the model (L1-norm).⁶⁴

$$\begin{aligned} \text{Lasso Regression Cost Function} &= \sum_{i=1}^N (y_{i,true} - y_{i,pred})^2 + \alpha \sum_{j=1}^m |\beta_j| \\ &= \|Y - X\hat{\beta}\|^2 + \alpha \|\hat{\beta}\|_1 \end{aligned}$$

where,

N is the number of training samples.

$y_{i,true}$ is the true value of the target variable corresponding to training sample i .

$y_{i,pred}$ is the predicted value of the target variable corresponding to training sample i .

β_j is the coefficients corresponding to feature j .

X is the matrix of independent variables corresponding to all training samples.

Y is the matrix of target values of all training samples.

α is the coefficient of regularisation.

m is the number of features.

As with ridge regression, lasso regression shrinks the coefficient estimates towards 0. However, there is one difference. With lasso, the penalty pushes some of the coefficient estimates to be exactly 0, provided the tuning parameter, alpha, is large enough. Hence, lasso performs feature selection. Choosing an appropriate value of alpha is critical here as well. Because of this, it is easier to interpret models generated by lasso than those generated by ridge regression.

2.6.4 Logistic Regression

Logistic Regression is a supervised machine learning model for binary classification.⁶⁵ In this model, the probability of a sample ($X = x_1, \dots, x_n$) belonging to a positive class is modelled using the logistic function over the linear combination of features, as given below:

$$P(X) = g(W^T X) = \frac{1}{1 + e^{-W^T X}}$$

where,

W is a weight vector.

X is the matrix of independent variables corresponding to all training samples.

The logistic function is also known as the sigmoid function. The advantage of the logistic function is that its output always lies between 0 and 1 for any value of X , making it a suitable candidate for probability estimation. Another advantage is that logistic function has two types of regions: (i) the regions in which probability is either small or large with diminishing returns – a small change in probability requires a more significant change in X , and (ii) the middle region where a small change in X results in a large change in probability. The middle region defines the boundary between classes. The plot of the logistic function is shown in Figure 2.7.

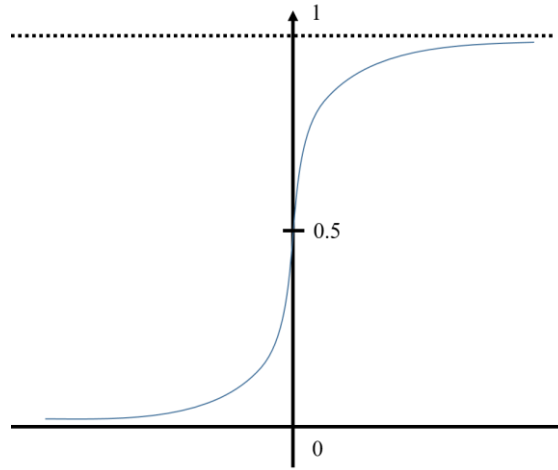


Figure 2.7. The logistic function.

Binary Cross-Entropy is used as a cost function during the training of logistic regression. Binary Cross-Entropy is the negative of mean log-likelihood. Thus, minimizing the cost function is equivalent to maximizing the likelihood.

$$\text{Logistic Regression Cost Function} = -\frac{1}{N} \sum_{i=1}^N y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)$$

where,

N is the total number of data points.

y_i is the true class of the i -th data point.

\hat{y}_i is the class predicted by the logistic regression for the i -th data point.

Gradient descent is generally employed to minimize this cost function with respect to coefficients W . After training, the class of a new sample is predicted using a cut-off, generally taken as 0.5. The sample is estimated to belong to a positive class if the probability obtained from the logistic regression is ≥ 0.5 ; else, it is estimated to belong to a negative class. Other values of cut-off may also be chosen depending on the given application.

2.6.5 Naïve Bayes

Naïve Bayes is a simple yet effective supervised machine learning algorithm for classification. Naïve Bayes is a probabilistic classifier that uses Bayes' theorem. It returns the probability of a test point belonging to a class rather than the label of the test point.²⁵ The probability of a data point X having features x_1, \dots, x_n belonging to a class C_i is given by Bayes' theorem as given below:

$$P(C_i|X) = \frac{P(X|C_i) P(C_i)}{P(X)}$$

where,

$P(C_i)$ is known as the prior probability. It is the probability of an event occurring before the collection of new data. It represents our belief. Prior probability plays an important role in the classification of new data points.

$P(X|C_i)$ represents the likelihood function. It tells the likelihood of a data point occurring in a class C_i . The conditional independence assumption is leveraged while computing the likelihood probability.

The effect of the denominator $P(X)$ is not incorporated while calculating probabilities, as it is the same for all the classes and hence, can be ignored without affecting the final outcome.

$P(C_i|X)$ is called the posterior probability, which is finally compared for all classes, and the data point is assigned to the class with the highest posterior probability.

For a dataset with all categorical features, these probabilities are simply computed by counting the number of instances/occurrences of the categorical data. Calculation of prior probability is easier than likelihood. Therefore, to simplify the computation, naïve Bayes makes the assumption that x_1, \dots, x_n are conditionally independent. Thus, it is called a naïve model as this assumption is most likely not valid in real situations. The final representation of class probability is given as:

$$\begin{aligned} P(C_i|x_1, \dots, x_n) &\propto P(x_1, \dots, x_n|C_i) P(C_i) \\ &\propto P(C_i) \prod_{j=1}^n P(x_j|C_i) \end{aligned}$$

The computation of individual $P(x_j|C_i)$ depends on the distribution of features in a given class. For example, in the text classification, where features may be word counts, they may follow a multinomial distribution. In some cases, where features are continuous, they may follow a Gaussian distribution. There is very little explicit training involved in naïve Bayes compared to other classification models. The only work that needs to be done is calculating the conditional probability of individual features and the prior probability of classes, which can be done quickly. Therefore, naïve Bayes performs well with high-dimensional data and large datasets. To estimate the class of a new data point, naïve Bayes simply chooses a class that has the highest probability for a given data point, as shown below:

$$y = \operatorname{argmax}_{C_i} P(C_i) \prod_{j=1}^n P(x_j|C_i)$$

Thus, the naïve Bayes estimate is also known as Maximum A Posteriori (MAP) estimate.

2.6.6 Support Vector Machines

Support Vector Machines, or SVMs, are a class of extremely popular classification models. Besides their ability to solve complex machine learning problems, they have numerous advantages over other classification algorithms, such as dealing with computationally heavy datasets and classifying non-linearly separable data. SVMs solve the problem of nonlinearity through kernel trick. Logically, multiple lines (hyperplanes) could perfectly separate the two classes. However, the best separator is the one that maintains the largest possible equal distance from the nearest points of both classes. Therefore, for a separator to be optimal, the margin or

the distance of the nearest point to the separator should be maximum. This is called the maximal margin classifier, as shown in Figure 2.8. The goal of SVM is to identify such maximum margin hyperplane in a high dimensional space defined by the kernel function that maps to a non-linear classifier in the original feature space.²⁷

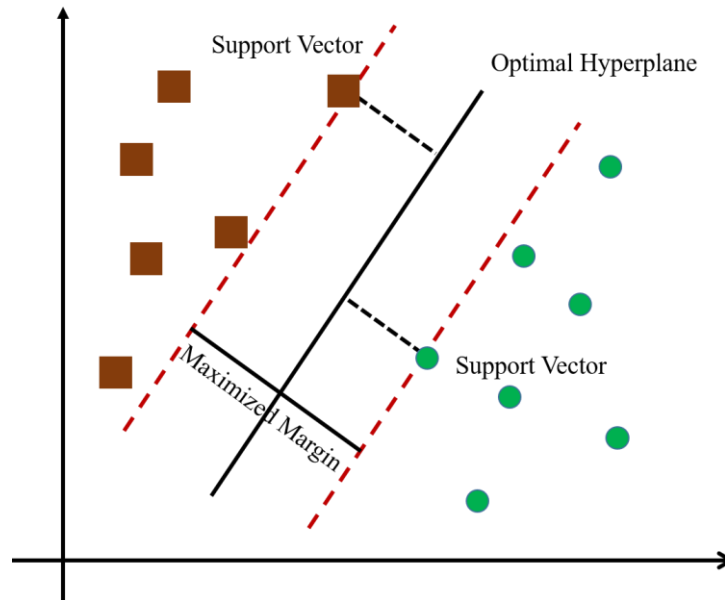


Figure 2.8. Support Vector Machine.

The points close to the hyperplane are only considered for constructing the hyperplane, and those points are called support vectors. The support vector classifier also allows certain points to be deliberately misclassified. By doing this, it is able to classify most of the points correctly in the unseen data making it more robust. The support vector classifier is also called the soft margin classifier because, instead of searching for the margin that exactly classifies each and every data point to the correct class, the soft margin classifier allows some observations to fall on the wrong side.

Support vector machines solve a constraint optimization problem defined below:

$$\min_{w, b, \zeta} \frac{1}{2} w^T w + C \sum_{i=1}^n \zeta_i$$

$$\text{subject to, } y_i(w^T \phi(x_i) + b) \geq 1 - \zeta_i,$$

$$\zeta_i \geq 0, i = 1, \dots, n$$

where,

w is the coefficient vector.

b is the intercept term.

ζ_i is the distance of the i -th data point from its correct margin boundary.

ϕ is the function that transforms input features into a high dimensional space.

C is the penalty term.

Minimizing $w^T w$ is equivalent to maximizing the margin. The support vector classifier works well even when the data is partially intermingled (i.e., data is not linearly separable). The hyperplane in a high dimension may not be perfect; therefore, SVM allows some points to be misclassified (i.e., lie at a distance from their correct margin boundary). C is the penalty term that controls the strength of the misclassification. The success of SVM is due to the fact that it does not need to transform features using ϕ . As feature transformation results in a large number of features, it makes the modeling (i.e., the learning process) computationally expensive. The key fact that makes the kernel trick possible is that — SVM only needs the inner products of the observations to find the best fit model, which can be easily computed using a kernel function without feature transformation.

2.6.7 Support Vector Regression

This model is the regression form of a support vector machine (SVM), a popular algorithm for classification tasks. Analogous to SVM, Support vector regression depends on the subset of training data and ignores the points whose predictions are close to their true values. SVM also utilizes the kernel trick and learns a hyperplane in high dimensional space to explain the relationship in the original dimensions.⁶⁶

2.6.8 Decision Trees

With their high interpretability and intuitive nature, decision trees mimic the human decision-making process and excel in dealing with categorical and continuous data. It is possible to easily explain all the factors or rules leading to a particular decision/prediction in the decision trees. Hence, decision trees are easily understood by people. Decision trees naturally represent the way we make decisions. A decision tree is similar to a flowchart that helps make decisions/predictions.²⁶ It is a predictive model that resembles an upside-down tree, as shown in Figure 2.9.

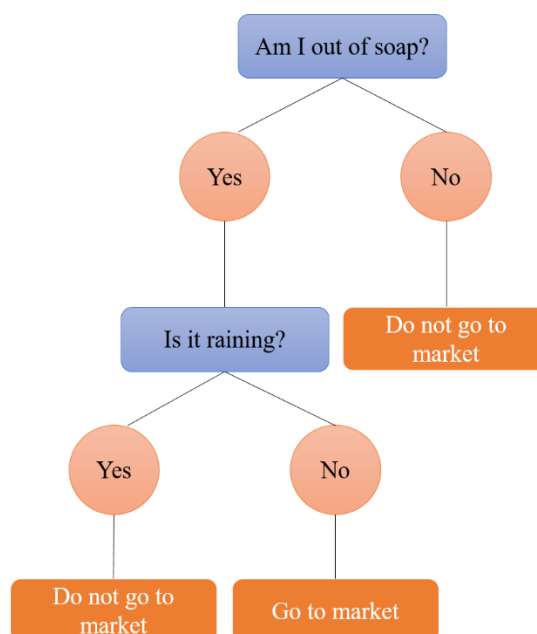


Figure 2.9. Decision Tree.

Decision tree is a supervised learning method, i.e., it has a fixed target variable, but unlike logistic regression, it is not a parametric model. It is free to learn any functional form from the training data and does not have a fixed set of parameters to define the model. Decision trees can be considered a set of if-then-else statements. One of the major advantages of using a decision tree is that it can handle both categorical and continuous features. Another advantage of decision trees is that they can be used for both classification and regression tasks. Constructing a decision tree involves the following steps:

- Recursive binary splitting/partitioning the data into smaller subsets
- Selecting the best rule from a feature for the split
- Applying the split based on the rules obtained from the features
- Repeating the process for the subsets obtained
- Continuing the process until the stopping criterion is reached
- Assigning the majority class/average value as the prediction

In order to construct a decision tree, we must know how to select a node that will lead to the best possible solution. Homogeneity/Purity is one of the factors considered while constructing a decision tree. A dataset is considered completely homogeneous for classification tasks if it contains only a single class label, which is extremely difficult to achieve in real-world datasets. While creating a decision tree, we should follow a step-by-step approach by picking an attribute first and then splitting the data such that the homogeneity of the child nodes increase after every split. Then, splitting is stopped when the resulting leaves are sufficiently homogenous. For this, we need to define the degree of homogeneity. Various methods are used for quantifying homogeneity, such as the classification error, Gini index and entropy (for classification), and MSE (for regression).

The classification error is calculated as follows:

$$E = 1 - \max(p_i)$$

The Gini index is calculated as follows:

$$G = \sum_{i=0}^k p_i(1 - p_i)$$

Entropy is calculated as follows:

$$D = - \sum_{i=0}^k p_i \log_2(p_i)$$

Where p_i is the probability of finding a point with the label i , and k is the number of classes.

CART is a famous algorithm for building decision trees with the Gini index as a splitting criterion. It includes the following steps for building a decision tree:

1. Calculate the Gini index before splitting the entire dataset.
2. Consider any one of the available features.
3. Calculate the Gini index after splitting on that particular feature for each of the levels of the features.

4. Combine the Gini index of all the levels to obtain the Gini index of the overall feature.
5. Repeat steps 2 – 5 with another feature until we have exhausted all of them.
6. Compare the Gini index across all the features and select the one that has the minimum Gini index.

2.6.9 Random Forests

A random forest algorithm combines multiple decision trees to generate the final results.⁶⁷ This process of combining more than one model to make the final decision is called ensemble learning. An ensemble tries to overcome the shortcomings of single ML models. The random forest algorithm is an ensemble of decision trees that uses bagging to generate different base models. So far, the random forest algorithm has been the most successful among the bagging ensembles. They are essentially ensembles of several decision trees. The random forest involves generating a large number of decision trees, each one on a different bootstrap sample from the training-set. The results are combined from different models for the final prediction. Bootstrapping refers to sampling from a given dataset. A bootstrap sample is generated by uniformly sampling the given dataset with replacement. A bootstrap sample typically contains 40% to 70% of the data from training-set. A random forest algorithm selects a random sample of data points (bootstrap sample) to build each tree and a random sample of features while splitting a node. Randomly selecting features ensures that each tree is diverse, an essential requirement for ensemble models. Finally, random forest aggregates the result of each model for the prediction. In classification problems, a majority vote is taken as the final prediction, whereas, in regression, an average is taken.

2.6.10 Artificial Neural Networks

This is the most important machine learning model in this age, which has evolved into a field of its own known as deep learning. Artificial neural networks have played a significant role in the success of machine learning. Deep learning is an active area of research in machine learning. The basic unit in these models is an artificial neuron modeled after a biological neuron. A single artificial neuron transforms input values using some mathematical operations and returns an output. The operation performed by the neuron is typically represented as follows:

$$y = \sigma\left(\sum_{i=1}^n (w_i x_i) + b\right)$$

Where,

x_i represents i -th feature of the input.

w_i represents a learnable weight for x_i .

b is the bias term.

σ represents an activation function.

Neural networks are inspired by the network of neurons in the brain. Millions of interconnected neurons in our brain perform various tasks; similarly, artificial networks are formed by connected layers of artificial neurons.²⁸ As shown in Figure 2.10, a typical artificial neural

network has an input layer, several hidden layers, and an output layer. They are used for classification as well as regression tasks. The weights of neural networks are learned using the backpropagation algorithm.

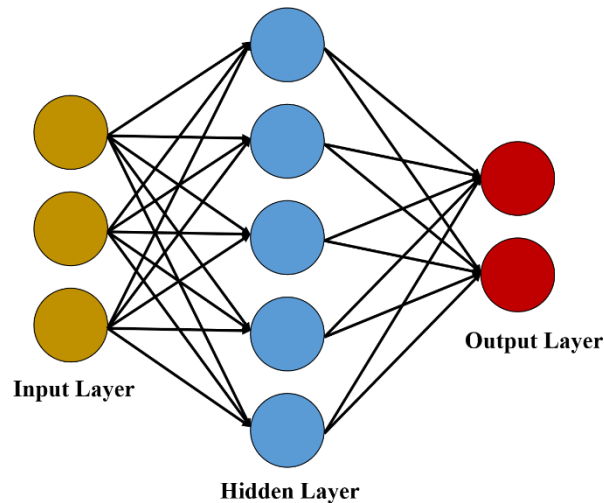


Figure 2.10. Artificial Neural Network.

2.6.11 Automatic Relevance Determination Regression

This is the probabilistic model related to the sparse Bayesian learning (SBL) framework. It assumes an axis-parallel, elliptical Gaussian distribution for each coefficient. The precision of each Gaussian distribution is drawn from the prior distribution (gamma distribution); therefore, it can lead to sparser coefficients. Thus, it is an effective tool for removing irrelevant features.^{68,69}

2.6.12 Gaussian Process Regression

It is a nonparametric Bayesian model. The nonparametric Bayesian model provides the probability distribution of parameters over all possible functions that fit the data. The prior in a Gaussian process is specified on the function space. The Gaussian process prior is a multivariate normal distribution whose mean is obtained from the data and whose covariance is specified using the kernel function. The hyperparameters of the kernel need to be optimized during the training.^{70,71} Some examples of kernel functions include RBF, matern, dot-product, exp-sine-squared, and white.⁷⁰ RBF kernel is a very popular kernel employed in many algorithms.

2.6.13 Kernel Ridge Regression

This is the extension of ridge regression with a kernel trick. In ridge regression, a linear model is learned with the L2-norm regularization. Using the kernel trick, KRR learns a linear function in the high dimensional non-linear space without actually transforming the data.⁷²

2.6.14 K-Means Algorithm

The k-means an unsupervised machine learning algorithm for dividing the N data points into K groups or clusters. It does so by calculating the distance between the data points using some

distance measure. A distance measure tells how similar two data points are — the points that are closer or more similar to each other would have a low distance, and the points which are farther or less similar to each other would have a higher distance. K-means involves the following steps:²⁹

1. Start by choosing K random points as the initial cluster centers.
2. Assign each data point to its nearest cluster center. The most common way of measuring the distance between the points is the Euclidean distance.
3. For each cluster, compute the new cluster center, which will be the mean of all cluster members.
4. Now, re-assign all the data points to the different clusters by considering the new cluster centers.
5. Keep iterating through steps 3 and 4 until no further changes are possible.

At this point, we arrive at the optimal clusters. Some of the points to be considered while implementing the k-means algorithm are (a) the choice of the initial cluster center has an impact on the final cluster composition, (b) we need to decide the number of clusters K in advance, (c) outliers have a serious impact on the performance of the algorithm and prevent optimal clustering, (d) the data needs to be standardized, and (e) the k-means algorithm cannot be employed when dealing with categorical data, as the concept of distance for categorical data does not make much sense.

2.6.15 Principal Component Analysis

Principal component analysis (PCA) is one of the most commonly used dimensionality reduction techniques. By converting large datasets into smaller ones containing fewer variables, it helps in improving model performance, visualizing complex datasets, and in many more areas. PCA performs dimensionality reduction by dropping the unnecessary variables, i.e., those that add no useful information. It converts the data by creating new features from old ones and then decides which features to consider based on information content using the variance. PCA calculates uncorrelated features (i.e., principal components) through a linear combination of original features. These principal components capture maximum information (i.e., variance) present in the data.³⁰ Dimensionality reduction is performed by choosing only those components that capture variance above a pre-defined threshold, e.g., greater than 95%. The most common application of PCA is to improve the model's performance. As principal components are uncorrelated, PCA helps us solve the problem of multicollinearity and thus model instability.

2.7 Brief Description of Modern Machine Learning Methods

2.7.1 Reinforcement Learning

Reinforcement learning involves identifying the best set of actions under different conditions that maximize the long-term rewards based on repeated interactions with the environment.⁷³ Typically, RL consists of an agent and the corresponding environment, as shown in Figure 2.11. The agent identifies the best actions by trial and error through repeated interaction with the environment. A few examples of reinforcement learning algorithms include Markov decision processes, Deep Q-Network, Proximal Policy Optimization, etc.^{33,73,74}

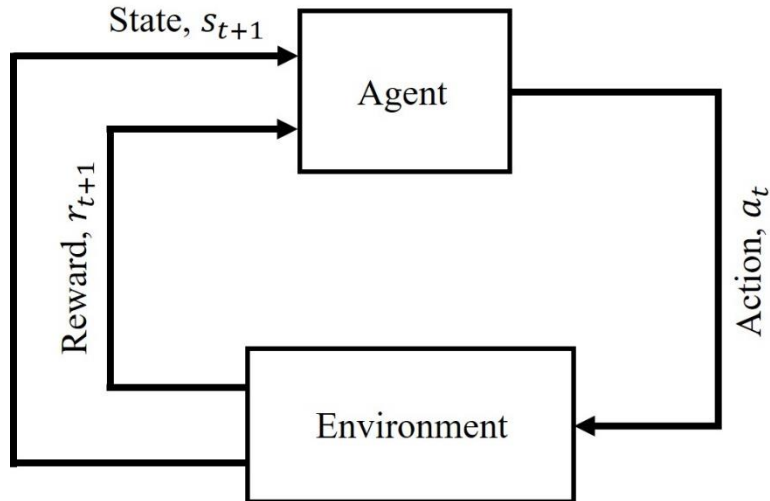


Figure 2.11. In reinforcement learning, the agent interacts with the environment through action, which causes the environment to transition to a new state and generate a reward.

During training, the agent takes an action a_t at time t , which affects the environment, causing it to transition from state s_t to s_{t+1} state. This transition results in a reward of r_{t+1} . Then, the agent takes an a_{t+1} based on the state s_{t+1} and reward r_{t+1} , continuing the cycle. The goal of the agent is to learn a mapping from states to actions, i.e., policy $\pi(a_{t+1}|s_{t+1})$ that maximizes a long-term sum of future rewards known as a value function v_π defined below.

$$v_\pi(s) = E(r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} \dots |s)$$

2.7.2 Recurrent Neural Networks

Artificial neural networks, particularly feedforward neural networks, assume that individual data points in training and test-sets are independent. This poses an issue for the datasets in which data points such as time series and sequence data are not independent. Thus, researchers introduced recurrent neural networks (i.e., RNNs), a type of neural network capable of modeling dependencies among data points. RNNs are commonly used in applications involving sequential data such as natural language processing, time series prediction, etc. RNNs have recurrent connections that pass relevant information from past data. These recurrent connections introduce the notion of time to the model. RNNs are composed of high-dimensional hidden states that work as the memory of the network. The state of the hidden layer depends on the earlier states enabling RNN to store, remember, and process data for longer time periods.⁷⁵ A simple RNN consists of an input layer, recurrent layers (i.e., hidden layers), and an output layer, as shown in Figure 2.12. RNNs are trained using a modified form of backpropagation algorithm known as backpropagation through time (BPTT).⁷⁶ However, RNNs are challenging to train. Consequently, many variations of RNNs, such as LSTM, GRU, etc., have been developed.⁷⁷

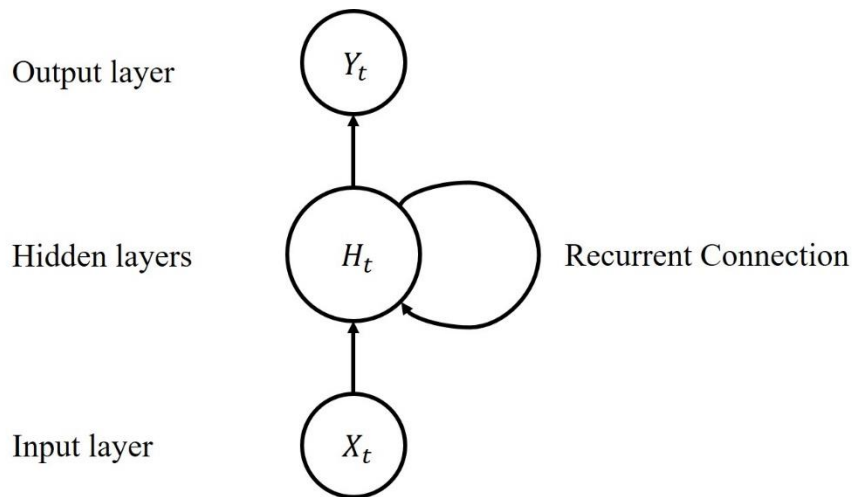


Figure 2.12. Simple recurrent neural network.

2.7.3 Convolutional Neural Networks

Convolutional Neural Networks, or CNNs, are specialized architectures that work particularly well with visual data, i.e., images and videos. They have been largely responsible for revolutionizing deep learning by setting new benchmarks for many image processing tasks that were recently considered extremely hard. Although fully connected neural networks can learn highly complex functions, their architecture does not exploit what we know about how the brain reads and processes images. For this reason, they haven't achieved any major breakthroughs in the image processing domain. CNNs had first demonstrated their extraordinary performance in the ImageNet challenge.⁷⁸ Convolutional neural networks consist series of convolution layers. These convolutional layers are composed of filters that extract various image features using convolution operation. The weights and biases of these filters are learned during training. The output from the filters is passed through the non-linear activation function, generally, ReLU. Convolutional layers are generally followed by pooling layers that aggregate the features and reduce feature dimensions. Typically, the last few layers in the CNNs are consists of fully connected layers followed by the output layer. A softmax layer is generally used as output layer for classification.⁷⁹ Figure 2.13 below depicts the structure of a typical CNN.

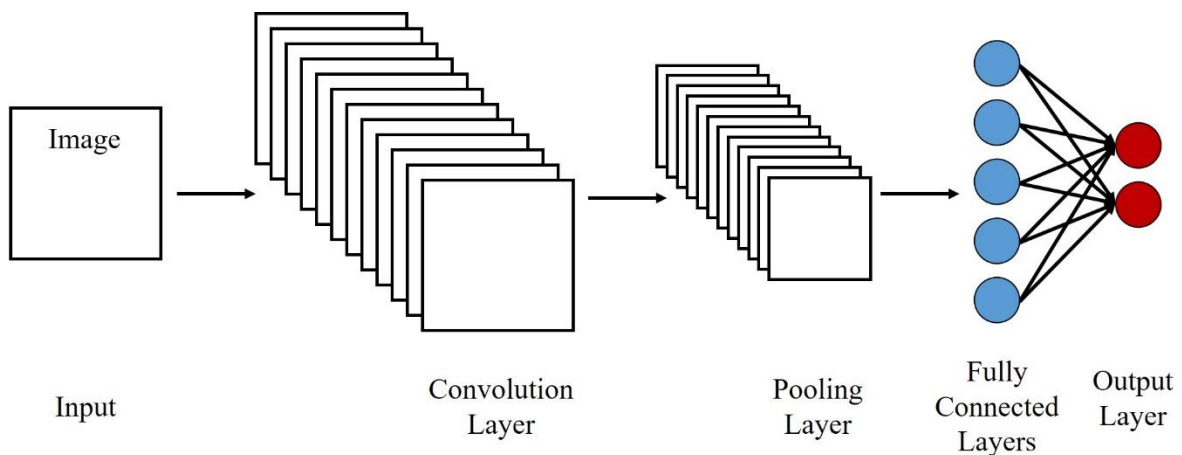


Figure 2.13. Arrangement of layers in a typical CNN.

2.7.4 Variational Autoencoders

Variational autoencoders (i.e., VAEs) are one of the widely used approaches for generative modeling and representation learning. VAEs are probabilistic generative models, they learn the true distribution of input features from the distribution of latent variables using Bayesian statistics.⁸⁰ VAEs approximate a latent space defined by mean μ and a standard deviation σ using stochastic inference. VAEs are composed of an encoder and a decoder network.⁸¹ The encoder provides a low-dimensional latent representation of the input data X at the bottleneck layer. At the same time, the decoder tries to reconstruct the input data \hat{X} . As VAEs learn the representation of input data in a continuous latent space, we are able to generate new data from VAEs.⁸² Figure 2.14 shows the structure of VAE.

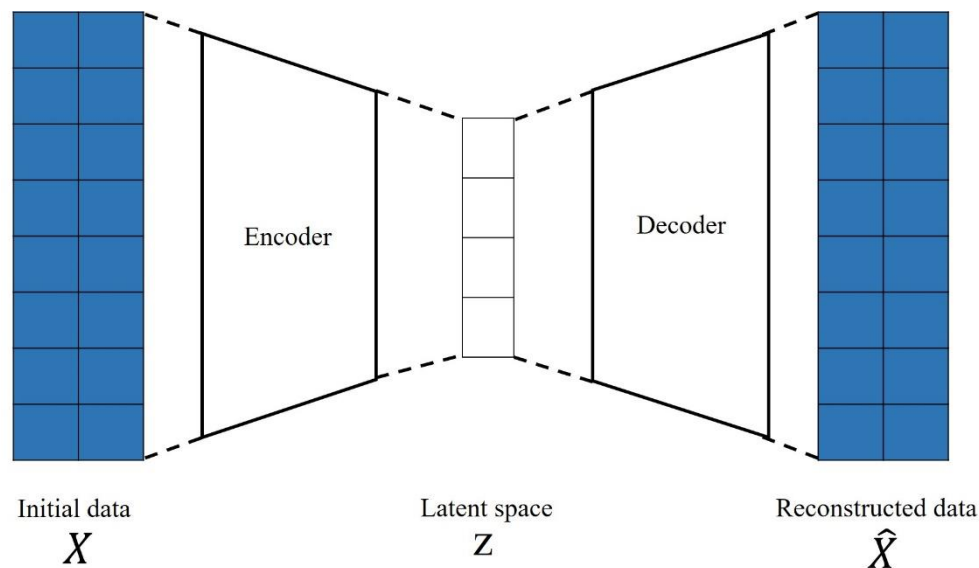


Figure 2.14. Structure of variational autoencoders.

2.7.5 Generative Adversarial Networks

Generative Adversarial Networks (i.e., GANs) are a novel class of generative models that were popularized due to their ability to generate realistic images.⁸³ GANs consist of a generator and a discriminator, as shown in Figure 2.15. The generator is a generative model used for generating fake data and capturing the probability distribution of the real data. Whereas, the discriminator is a discriminative model used for distinguishing real data from fake data. During training, the generator and discriminator compete with each other to achieve the Nash equilibrium using gradient-based optimization.⁸⁴ Generator generates fake data from noise vector, its objective is to deceive the discriminator. The discriminator is a binary classifier that receives fake data generated from the generator and real data. The objective of the discriminator is to identify real and fake data correctly. The optimal state is achieved when the discriminator fails to distinguish real data from fake data. The generator obtained at the optimal state has learned real data distribution. This generator could be used for generating data that resemble real data.⁸⁵

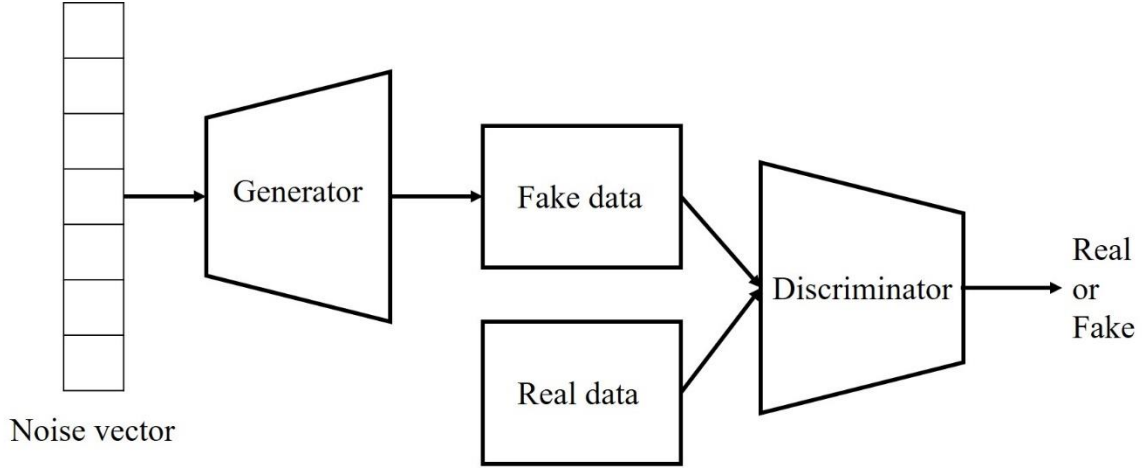


Figure 2.15. Generative Adversarial Networks consist of a generator and a discriminator. The generator generates fake data, whereas, the discriminator tries to identify real data from fake data.

2.8 Computational Methods

Supervised machine learning algorithms were employed for developing predictive models. The models investigated in this thesis work include linear regression, ridge regression, lasso, elastic-net, LARS lasso, orthogonal matching pursuit, Bayesian ridge regression, automatic relevance determination regression, passive aggressive, Huber regression, kernel ridge regression, support vector machines, Gaussian processes regression, decision trees, bagging meta-estimator, random forest, AdaBoost, gradient boosting regression, artificial neural network, and nearest neighbors regression. Supervised machine learning models were implemented using the scikit-learn Python library.⁸⁶ Hyperparameters of the models were optimized using the *'GridSearchCV'* class of the scikit-learn library⁸⁶. The mean squared error (MSE) was used as an evaluation metric during hyperparameter optimization. When necessary, feature selection was carried out using the *'SelectKBest'* class of the scikit-learn library.⁸⁷ Feature importance was computed using the *'permutation_importance'* class of the Scikit-learn library.⁸⁸ The following metrics were employed for evaluating the model performance. In the formulas below, N denotes the number of data points, \hat{y}_i denotes the predicted value of the i -th sample and y_i denotes the corresponding true value.

Coefficient of Determination (R^2):

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2}$$

$$\text{where, } \bar{y} = \frac{\sum_{i=1}^N y_i}{N}$$

Mean Squared Error (MSE):

$$MSE = \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{N}$$

Mean Absolute Error (MAE):

$$MAE = \frac{\sum_{i=1}^N |y_i - \hat{y}_i|}{N}$$

Mean Absolute Percentage Error (MAPE):

$$MAPE = 100 * \frac{\sum_{i=1}^N \frac{|y_i - \hat{y}_i|}{|y_i|}}{N}$$

The unsupervised machines learning models investigated in this thesis work include Latent Dirichlet Allocation (LDA), Hierarchical Dirichlet Processes (HDP), and No Promoter Left Behind (NPLB). LDA and HDP are frequently used for topic models. However, NPLB is a relatively new and promising clustering algorithm capable of identifying new promoters directly from the promotor sequences without any prior information on binding. We modified the typical workflow of NPLB for working with motif data. An implementation of LDA from the Gensim Python library was used in this work.⁸⁹ HDP was implemented using the hdp Python library developed by altosaar *et al.*⁹⁰ The NPLB library developed by Mitra and Narlikar was used for the development of the NPLB approach.⁹¹

2.9 References

- (1) Fujiyoshi, H.; Hirakawa, T.; Yamashita, T. Deep Learning-Based Image Recognition for Autonomous Driving. *IATSS Res.* **2019**, *43* (4), 244–252. <https://doi.org/10.1016/J.IATSSR.2019.11.008>.
- (2) Aziz, S.; Dowling, M.; Hammami, H.; Piepenbrink, A. Machine Learning in Finance: A Topic Modeling Approach. *Eur. Financ. Manag.* **2021**. <https://doi.org/10.1111/EUFM.12326>.
- (3) Wuest, T.; Weimer, D.; Irgens, C.; Thoben, K. D. Machine Learning in Manufacturing: Advantages, Challenges, and Applications. *http://mc.manuscriptcentral.com/tpmr* **2016**, *4* (1), 23–45. <https://doi.org/10.1080/21693277.2016.1192517>.
- (4) Zhang, C.; Hu, G.; Yurchenko, D.; Lin, P.; Gu, S.; Song, D.; Peng, H.; Wang, J. Machine Learning Based Prediction of Piezoelectric Energy Harvesting from Wake Galloping. *Mech. Syst. Signal Process.* **2021**, *160*, 107876. <https://doi.org/10.1016/J.YMSSP.2021.107876>.
- (5) Agrawal, A.; Choudhary, A. Perspective: Materials Informatics and Big Data: Realization of the “Fourth Paradigm” of Science in Materials Science. *APL Mater.* **2016**, *4* (5), 053208. <https://doi.org/10.1063/1.4946894>.
- (6) Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Židek, A.; Potapenko, A.; Bridgland, A.; Meyer, C.; Kohl, S. A. A.; Ballard, A. J.; Cowie, A.; Romera-Paredes, B.; Nikolov, S.; Jain, R.; Adler, J.; Back, T.; Petersen, S.; Reiman, D.; Clancy, E.; Zielinski, M.; Steinegger, M.; Pacholska, M.; Berghammer, T.; Bodenstein, S.; Silver, D.; Vinyals, O.; Senior, A. W.; Kavukcuoglu, K.; Kohli, P.; Hassabis, D. Highly Accurate Protein Structure Prediction with AlphaFold. *Nat.* **2021**, *596* (7873), 583–589. <https://doi.org/10.1038/s41586-021-03819-2>.
- (7) Jordan, M. I.; Mitchell, T. M. Machine Learning: Trends, Perspectives, and Prospects. *Science* (80-.). **2015**, *349* (6245), 255–260. https://doi.org/10.1126/SCIENCE.AAA8415/ASSET/AB2EF18A-576D-464D-B1B6-1301159EE29A/ASSETS/GRAPHIC/349_255_F5.JPEG.
- (8) Mitchell, T. M. *Machine Learning*, 1st editio.; McGraw-Hill Education, 1997.
- (9) Dhall, D.; Kaur, R.; Juneja, M. Machine Learning: A Review of the Algorithms and Its Applications. *Lect. Notes Electr. Eng.* **2020**, *597*, 47–63. https://doi.org/10.1007/978-3-030-29407-6_5.
- (10) Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2015**, *115* (3), 211–252. <https://doi.org/10.1007/S11263-015-0816-Y>.
- (11) Lecun, Y.; Bengio, Y.; Hinton, G. ; Gregor K And Lecun, Y.; Icml ; F, K. D.; Philbin, J.; Cvpr ; Schuster, M.; Chen, Z. PERSPECTIVES Special Topic: Machine Learning Deep Learning for Natural Language Processing: Advantages and Challenges. *11. Sprechmann P, Bronstein AM Sapiro G. IEEE TPAMI* **2018**, *5* (1), 22–24. <https://doi.org/10.1093/nsr/nwx099>.

- (12) Khanal, S. S.; Prasad, P. W. C.; Alsadoon, A.; Maag, A. A Systematic Review: Machine Learning Based Recommendation Systems for e-Learning. *Educ. Inf. Technol. 2019 254* **2019**, 25 (4), 2635–2664. <https://doi.org/10.1007/S10639-019-10063-9>.
- (13) Turing, A. M. Computing Machinery and Intelligence. *Parsing Turing Test Philos. Methodol. Issues Quest Think. Comput.* **2009**, 23–65. https://doi.org/10.1007/978-1-4020-6710-5_3.
- (14) Rosenblatt, F. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychol. Rev.* **1958**, 65 (6), 386–408. <https://doi.org/10.1037/H0042519>.
- (15) Fundamental Knowledge of Machine Learning | by Iftekher Aziz | Analytics Vidhya | Medium <https://medium.com/analytics-vidhya/fundamental-omachine-learning-ada28afa1bd3> (accessed Apr 21, 2022).
- (16) Strawn, G.; Strawn, C. Masterminds of Artificial Intelligence: Marvin Minsky and Seymour Papert. *IT Prof.* **2016**, 18 (06), 62–64. <https://doi.org/10.1109/MITP.2016.116>.
- (17) Rana, A.; Rawat, A. S.; Bijalwan, A.; Bahuguna, H. Application of Multi Layer (Perceptron) Artificial Neural Network in the Diagnosis System: A Systematic Review. *Proc. 2018 3rd IEEE Int. Conf. Res. Intell. Comput. Eng. RICE 2018* **2018**. <https://doi.org/10.1109/RICE.2018.8509069>.
- (18) Deep Learning 101 - Part 1: History and Background http://beamlab.org/deeplearning/2017/02/23/deep_learning_101_part1.html (accessed Apr 21, 2022).
- (19) Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. Learning Representations by Back-Propagating Errors. *Nat. 1986 3236088* **1986**, 323 (6088), 533–536. <https://doi.org/10.1038/323533a0>.
- (20) Hinton, G.; Deng, L.; Yu, D.; Dahl, G.; Mohamed, A. R.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T.; Kingsbury, B. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Process. Mag.* **2012**, 29 (6), 82–97. <https://doi.org/10.1109/MSP.2012.2205597>.
- (21) Mayr, A.; Klambauer, G.; Unterthiner, T.; Hochreiter, S. DeepTox: Toxicity Prediction Using Deep Learning. *Front. Environ. Sci.* **2016**, 0 (FEB), 80. <https://doi.org/10.3389/FENVS.2015.00080>.
- (22) Greener, J. G.; Kandathil, S. M.; Moffat, L.; Jones, D. T. A Guide to Machine Learning for Biologists. *Nat. Rev. Mol. Cell Biol. 2021 231* **2021**, 23 (1), 40–55. <https://doi.org/10.1038/s41580-021-00407-0>.
- (23) Sarker, I. H. Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Comput. Sci. 2021 23* **2021**, 2 (3), 1–21. <https://doi.org/10.1007/S42979-021-00592-X>.
- (24) Shakhnarovich, G.; Darrell, T.; Indyk, P. *Nearest-Neighbor Methods in Learning and Vision : Theory and Practice*; MIT Press, 2005.
- (25) Hand, D. J.; Yu, K. Idiot’s Bayes: Not So Stupid after All? *Int. Stat. Rev. / Rev. Int.*

- Stat.* **2001**, 69 (3), 385. <https://doi.org/10.2307/1403452>.
- (26) Kotsiantis, S. B. Decision Trees: A Recent Overview. *Artif. Intell. Rev.* 2011 394 **2011**, 39 (4), 261–283. <https://doi.org/10.1007/S10462-011-9272-4>.
- (27) Noble, W. S. What Is a Support Vector Machine? *Nat. Biotechnol.* 2006 2412 **2006**, 24 (12), 1565–1567. <https://doi.org/10.1038/nbt1206-1565>.
- (28) Jain, A. K.; Mao, J.; Mohiuddin, K. M. Artificial Neural Networks: A Tutorial. *Computer (Long. Beach. Calif.)*. **1996**, 29 (3), 31–44. <https://doi.org/10.1109/2.485891>.
- (29) Hartigan, J. A.; Wong, M. A. Algorithm AS 136: A K-Means Clustering Algorithm. *Appl. Stat.* **1979**, 28 (1), 100. <https://doi.org/10.2307/2346830>.
- (30) Abdi, H.; Williams, L. J. Principal Component Analysis. *Wiley Interdiscip. Rev. Comput. Stat.* **2010**, 2 (4), 433–459. <https://doi.org/10.1002/WICS.101>.
- (31) Blei, D. M.; Ng, A. Y.; Jordan, M. I. Latent Dirichlet Allocation. *J. Mach. Learn. Res.* **2003**, 3 (4–5), 993–1022. <https://doi.org/10.1016/b978-0-12-411519-4.00006-9>.
- (32) Teh, Y. W.; Jordan, M. I.; Beal, M. J.; Blei, D. M. Hierarchical Dirichlet Processes. *J. Am. Stat. Assoc.* **2006**, 101 (476), 1566–1581. <https://doi.org/10.1198/016214506000000302>.
- (33) Puterman, M. L. Chapter 8 Markov Decision Processes. In *Handbooks in Operations Research and Management Science*; Elsevier, 1990; Vol. 2, pp 331–434. [https://doi.org/10.1016/S0927-0507\(05\)80172-0](https://doi.org/10.1016/S0927-0507(05)80172-0).
- (34) Wei, J.; Chu, X.; Sun, X.; Xu, K.; Deng, H.; Chen, J.; Wei, Z.; Lei, M. Machine Learning in Materials Science. *InfoMat* **2019**, 1 (3), 338–358. <https://doi.org/10.1002/inf2.12028>.
- (35) UCI Machine Learning Repository <https://archive.ics.uci.edu/ml/index.php> (accessed Apr 22, 2022).
- (36) Find Open Datasets and Machine Learning Projects | Kaggle <https://www.kaggle.com/datasets?fileType=csv> (accessed Apr 22, 2022).
- (37) Kirklin, S.; Saal, J. E.; Meredig, B.; Thompson, A.; Doak, J. W.; Aykol, M.; Rühl, S.; Wolverton, C. The Open Quantum Materials Database (OQMD): Assessing the Accuracy of DFT Formation Energies. *npj Comput. Mater.* 2015 11 **2015**, 1 (1), 1–15. <https://doi.org/10.1038/npjcompumats.2015.10>.
- (38) Jain, A.; Ong, S. P.; Hautier, G.; Chen, W.; Richards, W. D.; Dacek, S.; Cholia, S.; Gunter, D.; Skinner, D.; Ceder, G.; Persson, K. A. Commentary: The Materials Project: A Materials Genome Approach to Accelerating Materials Innovation. *APL Mater.* **2013**, 1 (1), 011002. <https://doi.org/10.1063/1.4812323>.
- (39) Hachmann, J.; Olivares-Amaya, R.; Atahan-Evrenk, S.; Amador-Bedolla, C.; Sánchez-Carrera, R. S.; Gold-Parker, A.; Vogt, L.; Brockway, A. M.; Aspuru-Guzik, A. The Harvard Clean Energy Project: Large-Scale Computational Screening and Design of Organic Photovoltaics on the World Community Grid. *J. Phys. Chem. Lett.* **2011**, 2 (17), 2241–2251. https://doi.org/10.1021/JZ200866S/ASSET/IMAGES/MEDIUM/JZ-2011-00866S_0009.GIF.
- (40) Hellenbrandt, M. The Inorganic Crystal Structure Database (ICSD)—Present and

- Future. <https://doi.org/10.1080/08893110410001664882> **2014**, *10* (1), 17–22.
<https://doi.org/10.1080/08893110410001664882>.
- (41) Kim, S.; Chen, J.; Cheng, T.; Gindulyte, A.; He, J.; He, S.; Li, Q.; Shoemaker, B. A.; Thiessen, P. A.; Yu, B.; Zaslavsky, L.; Zhang, J.; Bolton, E. E. PubChem in 2021: New Data Content and Improved Web Interfaces. *Nucleic Acids Res.* **2021**, *49* (D1), D1388–D1395. <https://doi.org/10.1093/nar/gkaa971>.
- (42) ZINC <https://zinc.docking.org/> (accessed Jan 17, 2022).
- (43) ENCODE: Deciphering Function in the Human Genome <https://www.genome.gov/27551473/genome-advance-of-the-month-encode-deciphering-function-in-the-human-genome> (accessed Mar 9, 2022).
- (44) Sayers, E. W.; Cavanaugh, M.; Clark, K.; Ostell, J.; Pruitt, K. D.; Karsch-Mizrachi, I. GenBank. *Nucleic Acids Res.* **2020**, *48* (D1), D84–D86.
<https://doi.org/10.1093/NAR/GKZ956>.
- (45) Ogata, H.; Goto, S.; Sato, K.; Fujibuchi, W.; Bono, H.; Kanehisa, M. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res.* **1999**, *27* (1), 29–34.
<https://doi.org/10.1093/NAR/27.1.29>.
- (46) Cohen, K. B.; Hunter, L. Getting Started in Text Mining. *PLOS Comput. Biol.* **2008**, *4* (1), e20. <https://doi.org/10.1371/JOURNAL.PCBI.0040020>.
- (47) Chu, X.; Ilyas, I. F.; Krishnan, S.; Wang, J. Data Cleaning: Overview and Emerging Challenges. *Proc. ACM SIGMOD Int. Conf. Manag. Data* **2016**, *26-June-2016*, 2201–2206. <https://doi.org/10.1145/2882903.2912574>.
- (48) Weininger, D. SMILES, a Chemical Language and Information System: 1: Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28* (1), 31–36.
https://doi.org/10.1021/CI00057A005/ASSET/CI00057A005.FP.PNG_V03.
- (49) Heller, S. R.; McNaught, A.; Pletnev, I.; Stein, S.; Tchekhovskoi, D. InChI, the IUPAC International Chemical Identifier. *J. Cheminform.* **2015**, *7* (1), 1–34.
<https://doi.org/10.1186/S13321-015-0068-4/FIGURES/11>.
- (50) Kearnes, S.; McCloskey, K.; Berndl, M.; Pande, V.; Riley, P. Molecular Graph Convolutions: Moving beyond Fingerprints. *J. Comput. Aided. Mol. Des.* **2016**, *30* (8), 595–608. <https://doi.org/10.1007/S10822-016-9938-8/FIGURES/11>.
- (51) Moriwaki, H.; Tian, Y. S.; Kawashita, N.; Takagi, T. Mordred: A Molecular Descriptor Calculator. *J. Cheminform.* **2018**, *10* (1), 1–14.
<https://doi.org/10.1186/S13321-018-0258-Y/FIGURES/6>.
- (52) Landrum, G. RDKit: Open-source cheminformatics <https://www.rdkit.org/> (accessed Oct 23, 2021).
- (53) Muegge, I.; Mukherjee, P. An Overview of Molecular Fingerprint Similarity Search in Virtual Screening. <http://dx.doi.org/10.1517/17460441.2016.1117070> **2015**, *11* (2), 137–148. <https://doi.org/10.1517/17460441.2016.1117070>.
- (54) Bembom, O.; Bembom, O. Sequence Logos for DNA Sequence Alignments. **2014**.
- (55) Stormo, G. D. DNA Motif Databases and Their Uses. *Curr. Protoc. Bioinforma.* **2015**,

- 51 (1), 2.15.1-2.15.6. <https://doi.org/10.1002/0471250953.BI0215S51>.
- (56) Guo, Y.; Gifford, D. K. Modular Combinatorial Binding among Human Trans-Acting Factors Reveals Direct and Indirect Factor Binding. *BMC Genomics* **2017**, *18* (1), 1–16. <https://doi.org/10.1186/s12864-016-3434-3>.
- (57) Cayir, A.; Yenidogan, I.; Dag, H. Feature Extraction Based on Deep Learning for Some Traditional Machine Learning Methods. *UBMK 2018 - 3rd Int. Conf. Comput. Sci. Eng.* **2018**, 494–497. <https://doi.org/10.1109/UBMK.2018.8566383>.
- (58) Ruder, S. An Overview of Gradient Descent Optimization Algorithms. **2016**. <https://doi.org/10.48550/arxiv.1609.04747>.
- (59) Butler, K. T.; Davies, D. W.; Cartwright, H.; Isayev, O.; Walsh, A. Machine Learning for Molecular and Materials Science. *Nature* **2018**, *559* (7715), 547–555. <https://doi.org/10.1038/s41586-018-0337-2>.
- (60) Raschka, S. Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning. **2018**. <https://doi.org/10.48550/arxiv.1811.12808>.
- (61) Neal, B.; Mittal, S.; Baratin, A.; Tantia, V.; Scicluna, M.; Lacoste-Julien, S.; Mitliagkas, I. A Modern Take on the Bias-Variance Tradeoff in Neural Networks. **2018**. <https://doi.org/10.48550/arxiv.1810.08591>.
- (62) Su, X.; Yan, X.; Tsai, C. L. Linear Regression. *Wiley Interdiscip. Rev. Comput. Stat.* **2012**, *4* (3), 275–294. <https://doi.org/10.1002/WICS.1198>.
- (63) McDonald, G. C. Ridge Regression. *Wiley Interdiscip. Rev. Comput. Stat.* **2009**, *1* (1), 93–100. <https://doi.org/10.1002/WICS.14>.
- (64) Tibshirani, R. Regression Shrinkage and Selection Via the Lasso. *J. R. Stat. Soc. Ser. B* **1996**, *58* (1), 267–288. <https://doi.org/10.1111/J.2517-6161.1996.TB02080.X>.
- (65) LaValley, M. P. Logistic Regression. *Circulation* **2008**, *117* (18), 2395–2399. <https://doi.org/10.1161/CIRCULATIONAHA.106.682658>.
- (66) 1.4. Support Vector Machines — scikit-learn 1.0 documentation <https://scikit-learn.org/stable/modules/svm.html#svm-regression> (accessed Oct 23, 2021).
- (67) Biau, G.; Scornet, E. A Random Forest Guided Tour. *TEST 2016 252* **2016**, *25* (2), 197–227. <https://doi.org/10.1007/S11749-016-0481-7>.
- (68) 1.1. Linear Models — scikit-learn 1.0 documentation https://scikit-learn.org/stable/modules/linear_model.html#bayesian-regression (accessed Oct 23, 2021).
- (69) Wipf, D.; Nagarajan, S. A New View of Automatic Relevance Determination. *Adv. Neural Inf. Process. Syst.* **2007**, *20*.
- (70) 1.7. Gaussian Processes — scikit-learn 1.0 documentation https://scikit-learn.org/stable/modules/gaussian_process.html (accessed Oct 23, 2021).
- (71) Sit, H. Quick Start to Gaussian Process Regression <https://towardsdatascience.com/quick-start-to-gaussian-process-regression-36d838810319> (accessed Oct 23, 2021).
- (72) 1.3. Kernel ridge regression — scikit-learn 1.0 documentation https://scikit-learn.org/stable/modules/kernel_ridge.html (accessed Oct 23, 2021).

- learn.org/stable/modules/kernel_ridge.html (accessed Oct 23, 2021).
- (73) Shin, J.; Badgwell, T. A.; Liu, K.-H.; Lee, J. H. Reinforcement Learning – Overview of Recent Progress and Implications for Process Control. *Comput. Chem. Eng.* **2019**, *127*, 282–294. <https://doi.org/10.1016/j.compchemeng.2019.05.029>.
 - (74) Deng, J.; Sierla, S.; Sun, J.; Vyatkin, V. Reinforcement Learning for Industrial Process Control: A Case Study in Flatness Control in Steel Industry. *Comput. Ind.* **2022**, *143*, 103748. <https://doi.org/10.1016/J.COMPIND.2022.103748>.
 - (75) De Mulder, W.; Bethard, S.; Moens, M. F. A Survey on the Application of Recurrent Neural Networks to Statistical Language Modeling. *Comput. Speech Lang.* **2015**, *30* (1), 61–98. <https://doi.org/10.1016/J.CSL.2014.09.005>.
 - (76) Lipton, Z. C.; Berkowitz, J.; Elkan, C. A Critical Review of Recurrent Neural Networks for Sequence Learning. **2015**. <https://doi.org/10.48550/arxiv.1506.00019>.
 - (77) Shrestha, A.; Mahmood, A. Review of Deep Learning Algorithms and Architectures. *IEEE Access* **2019**, *7*, 53040–53065. <https://doi.org/10.1109/ACCESS.2019.2912200>.
 - (78) Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2015**, *115* (3), 211–252. <https://doi.org/10.1007/S11263-015-0816-Y>.
 - (79) Li, Z.; Liu, F.; Yang, W.; Peng, S.; Zhou, J. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Trans. Neural Networks Learn. Syst.* **2021**, 1–21. <https://doi.org/10.1109/TNNLS.2021.3084827>.
 - (80) Wei, R.; Mahmood, A. Recent Advances in Variational Autoencoders with Representation Learning for Biomedical Informatics: A Survey. *IEEE Access* **2021**, *9*, 4939–4956. <https://doi.org/10.1109/ACCESS.2020.3048309>.
 - (81) Pratella, D.; Saadi, S. A. E. M.; Bannwarth, S.; Paquis-fluckinger, V.; Bottini, S. A Survey of Autoencoder Algorithms to Pave the Diagnosis of Rare Diseases. *Int. J. Mol. Sci.* **2021**, *Vol. 22, Page 10891* **2021**, *22* (19), 10891. <https://doi.org/10.3390/IJMS221910891>.
 - (82) Girin, L.; Leglaive, S.; Bie, X.; Diard, J.; Hueber, T.; Alameda-Pineda, X. Dynamical Variational Autoencoders: A Comprehensive Review. *Found. Trends® Mach. Learn.* **2021**, *15* (1–2), 1–175. <https://doi.org/10.1561/22000000089>.
 - (83) Alqahtani, H.; Kavakli-Thorne, M.; Kumar, G. Applications of Generative Adversarial Networks (GANs): An Updated Review. *Arch. Comput. Methods Eng.* **2019**, *28* (2), 525–552. <https://doi.org/10.1007/S11831-019-09388-Y>.
 - (84) Pan, Z.; Yu, W.; Yi, X.; Khan, A.; Yuan, F.; Zheng, Y. Recent Progress on Generative Adversarial Networks (GANs): A Survey. *IEEE Access* **2019**, *7*, 36322–36333. <https://doi.org/10.1109/ACCESS.2019.2905015>.
 - (85) Gui, J.; Sun, Z.; Wen, Y.; Tao, D.; Ye, J. A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications. *IEEE Trans. Knowl. Data Eng.* **2021**. <https://doi.org/10.1109/TKDE.2021.3130191>.
 - (86) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.;

- Courapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, É. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12* (85), 2825–2830.
- (87) sklearn.feature_selection.SelectKBest — scikit-learn 1.0.2 documentation https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html (accessed Jan 7, 2022).
- (88) 4.2. Permutation feature importance — scikit-learn 1.0 documentation https://scikit-learn.org/stable/modules/permutation_importance.html#permutation-importance (accessed Oct 23, 2021).
- (89) Reh\rek, R.; Sojka, P. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*; ELRA: Valletta, Malta, 2010; pp 45–50.
- (90) blei-lab/hdp: Hierarchical Dirichlet processes. Topic models where the data determine the number of topics. This implements Gibbs sampling. <https://github.com/blei-lab/hdp> (accessed Jun 25, 2021).
- (91) Mitra, S.; Narlikar, L. No Promoter Left Behind (NPLB): Learn de Novo Promoter Architectures from Genome-Wide Transcription Start Sites. *Bioinformatics* **2016**, *32* (5), 779–781. <https://doi.org/10.1093/BIOINFORMATICS/BTV645>

Chapter 3

Machine Learning the Redox Potentials of Phenazine Derivatives: A Comparative Study on Molecular Features

Chapter 3

Machine Learning the Redox Potentials of Phenazine Derivatives: A Comparative Study on Molecular Features

Abstract

Redox Flow Batteries (RFBs) are promising candidates for green and efficient energy storage systems. However, their widespread adoption still needs further investigations into cheaper and greener alternative organic redox-active species. In this work, we have developed machine-learning models to predict the redox potentials of phenazine derivatives in DME (dimethoxyethane) solvent using a small dataset of 189 molecules. 2D, 3D, and molecular fingerprint features were computed using readily available and easy-to-use Python libraries, making our approach easily adaptable to similar work. Twenty linear and non-linear machine learning models were investigated in this work. These models achieved excellent performance on the unseen data (i.e., $R^2 > 0.98$, $MSE < 0.008 \text{ V}^2$ and $MAE < 0.07 \text{ V}$). Model performance was assessed consistently using the training and evaluation pipeline developed in this work. We showed that 2D molecular features were the most informative and achieved the best prediction accuracy among four feature sets. We also showed that often less preferred but relatively faster linear models could perform better than non-linear models when the feature set contained different types of features (i.e., 2D, 3D, and molecular fingerprints) to predict the redox potential of phenazine derivatives. Further investigations revealed that it is possible to reduce the training and inference time without sacrificing prediction accuracy by using a small subset of features. Moreover, significantly low prediction errors were observed for most functional groups. Thus, we believe that the results obtained in this work would help in the adoption of green energy by accelerating the field of materials discovery for energy storage applications.

3.1 Introduction

Today, ~85% of the world's energy demand is being fulfilled by fossil fuels.^{1,2} The limited supply of fossil fuels and the ever-increasing population has raised concerns that we might run out of fossil fuels sooner than expected.^{1,3} Furthermore, electricity production from fossil fuels is one of the major factors responsible for greenhouse gas emissions.⁴ In this age, humanity faces two major challenges: of balancing increased energy demand while reducing the environmental impact associated with energy production. In the past decades, investments and research efforts in green technology have been increased to overcome these challenges.⁵ Significant progress has already been made to access renewable energy sources.^{6,7} Renewable energy sources, being intermittent, require efficient energy storage.⁴ Improvements in the energy storage technology would not only help in the adoption of renewable energy but also help in making efficient use of non-renewable energy sources. Historically, it has been more expensive to store energy than to expand energy generation to handle increased demand.⁸ Thus, grid systems employed today are likely to fail when additional energy cannot be generated during peak demand. The massive Texas Blackout in February 2021 is an example of such a failure.⁹ It suggests that efficient energy storage technology is urgently required. Unfortunately, only 1.0% of the energy consumed worldwide can be stored with the energy storage technology accessible today.¹⁰ Furthermore, the contribution of electrochemical batteries to energy storage capacity is less than 2.0%, even though most of the devices we use every day include batteries.^{8,10} Li-ion batteries are widely used today due to their high energy density, high specific energy, long cycle life, and fast charge-discharge cycle.^{4,8,11} Unfortunately, Li-ion batteries suffer from high production costs, safety issues, and high environmental impact.^{2,12} Redox flow batteries (RFBs) have the potential to overcome drawbacks of Li-ion batteries owing to their high storage capacity, independent control over storage capacity and power, fast responsiveness, ease of scaling, room temperature operation, cost-effectiveness, high round trip efficiency, safety, and lower environmental impact.¹³⁻²⁶ RFBs are increasingly being used as energy storage devices in renewable energy systems, thereby helping in the adoption of green energy.^{15,22} A schematic diagram of the typical redox flow battery is shown in Figure 3.1. RFB consists of two storage tanks containing cathode and anode redox-active species dissolved in an electrolyte solution. The electrolyte solution in the positive and negative compartments is termed catholyte and anolyte, respectively. These storage tanks are connected to an electrochemical cell (or current collector) *via* pumps. The electrochemical cell consists of porous electrodes separated by an ion-selective membrane. During operation, electrolytes containing redox-active species are pumped to the electrochemical cell, where redox-active species undergo oxidation or reduction depending on the charge/discharge cycle. Then, electrolytes are circulated back to their storage tanks.^{13,24} So far, transition metal-based redox flow batteries (such as vanadium, iron, and chromium) have found some commercial success. However, their widespread adoption has been limited mainly due to high production cost, toxicity, and cell component corrosion associated with the use of transition metal salts.^{27,28} Therefore, redox flow batteries containing organic redox-active species are being heavily investigated due to their low production cost, access to a massive space of electroactive compounds, and low environmental impact.^{28,29} Many organic compounds such as quinones, viologens, flavins, thiazines, imides, and their derivatives have been investigated for redox-active species in both aqueous and non-aqueous RFBs.^{27,30,31} However, non-aqueous RFBs

offer large operating voltage.³⁰ Recently, phenazine derivatives have been shown to be promising redox-active candidates in non-aqueous RFBs.

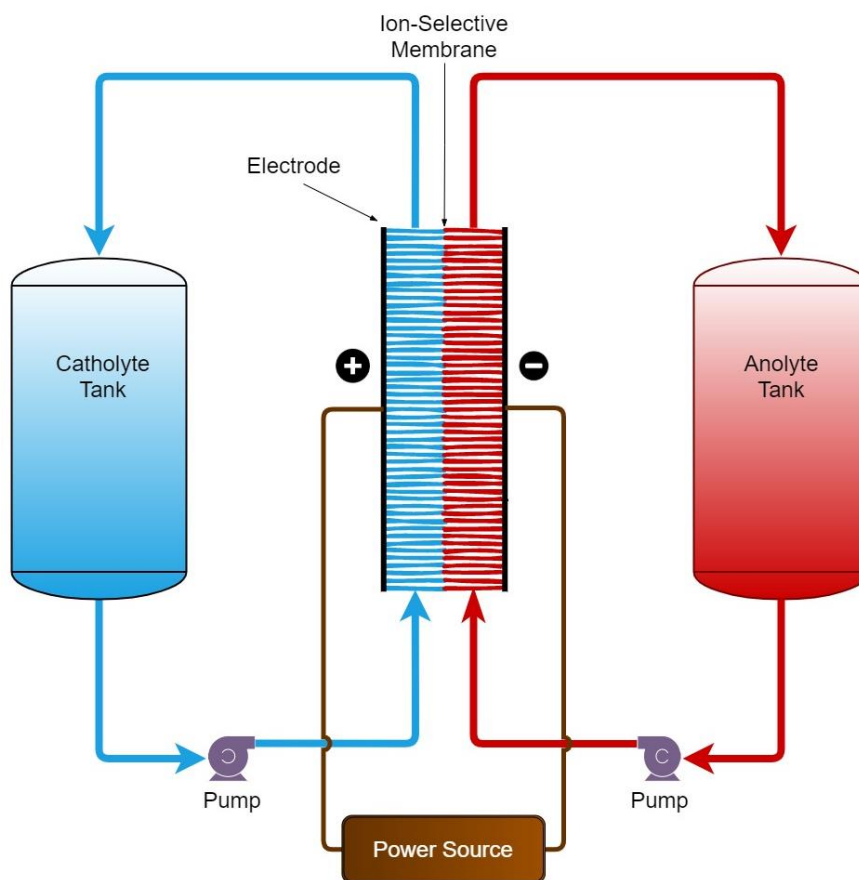


Figure 3.1. Schematic diagram of a typical redox flow battery.

Recent reports have revealed the reasons behind using phenazine derivatives as promising redox-active candidates. Romadina *et al.* synthesized phenazine derivatives having significantly negative redox potential.³² Materials with highly negative redox potential are preferred candidates for anolytes in RFBs. They showed that the non-aqueous RFB based on the synthesized phenazine derivative is capable of achieving a potential of 2.3 V, high capacities, > 95% coulombic efficiency, and good charge-discharge cycling stability after the initial 20 cycles. Mavrandonakis and co-workers, in their computational investigation, reported the most negative redox-active candidate based on phenazine for non-aqueous RFBs.²⁷ They showed that tetra-amino-phenazine (TAPZ) has 140 mV more negative potential than N-methylphthalimide (MePht), which has one of the most negative redox potentials reported so far in RFBs.³³ They also proposed all-phenazine RFB capable of reaching a high potential of 2.83 V. Furthermore, the redox potential of phenazine derivatives could be tuned easily with the addition of appropriate electron-donating or electron-withdrawing functional groups. The synthesis of phenazine derivatives is very economical than mining transition metals. Therefore, phenazine derivatives are currently being investigated as potential candidates for novel redox-active species.^{27,32}

These investigations remain primarily experimental. Unfortunately, the vast molecular space offered by organic compounds cannot be explored using experimental procedures. Quantum

mechanical DFT computations have been used heavily in chemistry research due to high accuracy but are very slow and cannot screen millions of molecules in a reasonable amount of time. Therefore, a fast and reliable method to screen millions of compounds without compromising accuracy is required. In this regard, machine-learning algorithms have shown excellent predictive accuracies along with short development and prediction times^{34–38}. Therefore, machine learning algorithms have been used extensively to screen millions of molecules in materials science and drug discovery.^{39–43} Machine learning models generally require a large amount of data for accurate predictions. When the quantity of data is limited, feature engineering is employed to generate the most informative features. These features are expected to capture the appropriate molecular information necessary to predict the target quantity. Feature engineering requires domain knowledge, relying on having access to experts.^{44–46} In small datasets, DFT-based or experimentally determined features have been used due to their high accuracy. However, some reports also explore simple features based on molecular structure.^{47–52}

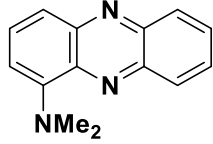
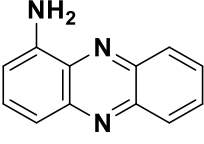
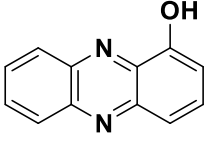
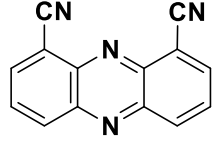
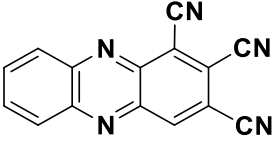
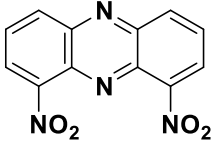
The goal of this study was to develop efficient machine learning models for predicting the redox potential of phenazine derivatives and understand the effect of different types of molecular features on prediction accuracy. We did not compute any features from DFT calculations or experimental studies to make our approach easily adaptable. The features used in this study were computed from molecular structures using readily available, easy-to-use Python libraries such as RDKit⁵³ and DeepChem.⁵⁴ These libraries have been used in other studies as well.^{55–58} Previous studies to predict redox potential using machine learning investigate only a small number of non-linear models.^{59–63} Furthermore, none of the previous studies use easily computable features from RDKit or DeepChem libraries. This study investigated twenty different linear and non-linear machine-learning models to predict the redox potential of phenazine derivatives in DME (dimethoxyethane) solvent. Linear models are generally faster to train but may not capture complex relationships between features and target variables, whereas non-linear models are capable of capturing these complex relationships but may overfit the training data and may require a considerable amount of training time. A total of 3510 features containing 2D, 3D, and molecular fingerprints were generated using RDKit, and DeepChem Python libraries. Models were trained on four feature sets described in Table 3.2 to obtain high prediction accuracy. Moreover, to understand which feature set had the best prediction accuracy, a detailed analysis of model performance was carried out using the pipeline developed in this work (described in section 3.2). The pipeline was developed to make training and evaluation easy, consistent, and automatic for all models. It combines different model training and evaluation steps into a single, convenient sub-routine. Then, the feature importance analysis was performed to identify the most important features in each feature set. After that, model performance was analyzed on small subsets of the most important features to reduce training and inference time for large datasets. Next, the prediction accuracy across different functional groups was analyzed. Finally, the sources of errors were identified. We believe that the methods used in this work are easily adaptable, and the results obtained in this study would help accelerate the discovery of novel redox-active species for energy storage applications.

3.2 Materials and Methods

3.2.1 Dataset

Data used in this study was obtained from the report by Mavrandonakis and co-workers.²⁷ The redox potentials of 189 phenazine derivatives in DME were provided in their report. These potentials were computed using DFT. Phenazine derivatives were generated from twenty unique electron-withdrawing and donating functional groups ($-\text{N}(\text{CH}_3)_2$, $-\text{NH}_2$, $-\text{OH}$, $-\text{OCH}_3$, $-\text{P}(\text{CH}_3)_2$, $-\text{SCH}_3$, $-\text{SH}$, $-\text{CH}_3$, $-\text{C}_6\text{H}_5$, $-\text{CH}=\text{CH}_2$, $-\text{F}$, $-\text{Cl}$, $-\text{CHO}$, $-\text{COCH}_3$, $-\text{CONH}_2$, $-\text{COOCH}_3$, $-\text{COOH}$, $-\text{CF}_3$, $-\text{CN}$ and $-\text{NO}_2$). Optimized 3D structures of molecules in neutral and in anionic states were also provided. Only neutral structures were used for the feature generation. However, not all compounds were supplied with their neutral structure. Therefore, compounds with missing neutral structures were removed. Thus, we ended up with 185 compounds in the final dataset. Next, 3510 different types of features were generated using RDKit and DeepChem, libraries as described below. Finally, the whole dataset was shuffled and split randomly into a training-set and a test-set, in a 7:3 ratio. This resulted in 129 samples in the training-set and 56 samples in the test-set. The term ‘Redox Potential’ in this chapter refers to the ‘Reduction Potential’ of phenazine derivatives. A few phenazine derivatives from the training-set/test-set are shown in Table 3.1.

Table 3.1. Representative structures from training-set/test-set. Mol IDs were assigned to identify derivatives from the corresponding dataset.

 Mol ID: 1	 Mol ID: 3	 Mol ID: 5
 Mol ID: 48	 Mol ID: 52	 Mol ID: 172

3.2.2 Feature Generation

For each compound, three types of features were generated: (i) 2D, (ii) 3D, and (iii) molecular fingerprints. Ten 2D features were also generated from the raw data (features with the word ‘basic’ in the suffix). The rest of the 2D features were computed using RDKit.⁵³ All 3D features were computed using RDKit. However, molecular fingerprints were computed using RDKit and DeepChem⁵⁴ libraries. These features were grouped into four sets, as shown in Table 3.2. Molecular fingerprints and some of the 3D features were one-dimensional vectors. In this study, we considered each component of the vectorial feature as an independent feature.

Therefore, a small number of unique 3D and molecular fingerprint features resulted in a large number of final features. Features having a “NaN” (Not a Number) value for any compound were removed. Also, features having identical values for all compounds were removed, as they did not contain any useful information. All 2D and 3D features computed from the RDKit library were scaled using the ‘*StandardScaler*’ class of the Scikit-learn library⁶⁴, which removes the mean and scales each feature to unit variance. A list of all features used in this study is given in Table 3.3.

Table 3.2. Feature sets.

Feature Set	Description	Number of Features
<i>2d+3d+fp</i>	Contains 2D and 3D features computed using raw data and RDKit library. Also contains molecular fingerprints computed from RDKit and DeepChem.	3510
<i>2d</i>	Contains only 2D features computed using the raw data and RDKit.	151
<i>3d</i>	Contains only 3D features computed using RDKit.	869
<i>fp</i>	Contains molecular fingerprints computed using DeepChem and RDKit.	2490

Table 3.3. List of all features.

Feature Type	Feature Names
2D	'FG_no_basic', 'FG_position_1_basic', 'FG_position_2_basic', 'FG_position_3_basic', 'FG_position_4_basic', 'FG_position_6_basic', 'FG_position_7_basic', 'FG_position_8_basic', 'FG_position_9_basic', 'MaxEStateIndex', 'MinEStateIndex', 'MaxAbsEStateIndex', 'MinAbsEStateIndex', 'qed', 'MolWt', 'HeavyAtomMolWt', 'ExactMolWt', 'NumValenceElectrons', 'MaxPartialCharge', 'MinPartialCharge', 'MaxAbsPartialCharge', 'MinAbsPartialCharge', 'FpDensityMorgan1', 'FpDensityMorgan2', 'FpDensityMorgan3', 'BCUT2D_MWHI', 'BCUT2D_MWLOW', 'BCUT2D_CHGHI', 'BCUT2D_CHGLO', 'BCUT2D_LOGPHI', 'BCUT2D_LOGPLOW', 'BCUT2D_MRHI', 'BCUT2D_MRLOW', 'BalabanJ', 'BertzCT', 'Chi0', 'Chi0n', 'Chi0v', 'Chi1', 'Chi1n', 'Chi1v', 'Chi2n', 'Chi2v', 'Chi3n', 'Chi3v', 'Chi4n', 'Chi4v', 'HallKierAlpha', 'Ipc', 'Kappa1', 'Kappa2', 'Kappa3', 'LabuteASA', 'PEOE_VSA1', 'PEOE_VSA10', 'PEOE_VSA11', 'PEOE_VSA12', 'PEOE_VSA13', 'PEOE_VSA14', 'PEOE_VSA2', 'PEOE_VSA3', 'PEOE_VSA4', 'PEOE_VSA5', 'PEOE_VSA6', 'PEOE_VSA7', 'PEOE_VSA8', 'PEOE_VSA9', 'SMR_VSA1', 'SMR_VSA10', 'SMR_VSA2', 'SMR_VSA4', 'SMR_VSA5', 'SMR_VSA6', 'SMR_VSA7', 'SMR_VSA9', 'SlogP_VSA1', 'SlogP_VSA10', 'SlogP_VSA11', 'SlogP_VSA12', 'SlogP_VSA2', 'SlogP_VSA3', 'SlogP_VSA4', 'SlogP_VSA5', 'SlogP_VSA6', 'SlogP_VSA7', 'SlogP_VSA8', 'TPSA', 'EState_VSA1', 'EState_VSA10', 'EState_VSA2', 'EState_VSA3', 'EState_VSA4', 'EState_VSA5', 'EState_VSA6', 'EState_VSA7', 'EState_VSA8', 'EState_VSA9', 'VSA_EState1', 'VSA_EState10', 'VSA_EState2', 'VSA_EState3', 'VSA_EState4', 'VSA_EState5', 'VSA_EState6', 'VSA_EState7', 'VSA_EState8', 'VSA_EState9', 'FractionCSP3', 'HeavyAtomCount', 'NHOHCount', 'NOCCount', 'NumAromaticCarbocycles', 'NumAromaticRings', 'NumHAcceptors', 'NumHDonors', 'NumHeteroatoms', 'NumRotatableBonds', 'RingCount', 'MolLogP', 'MolMR', 'fr_ArN', 'fr_Ar_COO', 'fr_Ar_OH', 'fr_COO', 'fr_COO2', 'fr_C_O', 'fr_C_O_noCOO', 'fr_NH0', 'fr_NH2', 'fr_SH', 'fr_aldehyde', 'fr_alkyl_halide', 'fr_amide', 'fr_aniline', 'fr_aryl_methyl', 'fr_benzene', 'fr_ester', 'fr_ether', 'fr_halogen', 'fr_ketone', 'fr_ketone_Topliss', 'fr_methoxy', 'fr_nitrile', 'fr_nitro', 'fr_nitro_ arom', 'fr_nitro_ arom_nonortho', 'fr_para_hydroxylation', 'fr_phenol', 'fr_phenol_noOrthoHbond', 'fr_priamide', 'fr_sulfide'
3D	'Asphericity', 'Eccentricity', 'InertialShapeFactor', 'NPR1', 'NPR2', 'PMI1', 'PMI2', 'PMI3', 'RadiusOfGyration', 'SphericityIndex', 'Autocorr3D', 'RDF', 'MORSE', 'WHIM', 'GETAWAY'
Molecular Fingerprints	'Extended Connectivity Circular Fingerprints (ECFP4)', 'MACCS Keys Fingerprint', 'RDKit Topological Fingerprint'

3.2.3 Machine Learning Models

Twenty linear and non-linear machine-learning models were investigated in this study. Machine learning models were implemented with the scikit-learn Python library.⁶⁴ A list of all models is given in Table 3.4.

Table 3.4. List of Models.

Sr. No.	Model Name	Alias
1	Linear Regression	linear_reg
2	Ridge Regression	ridge
3	Lasso	lasso
4	Elastic-Net	elastic_net
5	LARS Lasso	lasso_lars
6	Orthogonal Matching Pursuit	omp
7	Bayesian Ridge Regression	bayesian_ridge
8	Automatic Relevance Determination Regression	ARDR
9	Passive Aggressive	PA
10	Huber Regression	huber
11	Kernel Ridge Regression	kernel_ridge
12	Support Vector Regression	SVR
13	Gaussian Processes Regression	gaussian_process
14	Decision Trees	decision_tree
15	Bagging meta-estimator	bagging
16	Random Forest	random_forest
17	AdaBoost	ada_boost
18	Gradient Boosting Regression	gradient_boosting_reg
19	Artificial Neural Network	neural_network
20	Nearest Neighbors Regression	knn_reg

3.2.4 Hyperparameter Tuning

Hyperparameter tuning was performed for all models using the ‘*GridSearchCV*’ class of the scikit-learn library. ‘*GridSearchCV*’ performs a systematic search over a grid of parameters to identify the best set of parameters using cross-validation. 10-fold cross-validation with mean squared error (MSE) loss was used in this study.

3.2.5 Evaluation Metrics

The following metrics were used for evaluating the model performance. In the formulas below, N denotes the number of data points, \hat{y}_i denotes the predicted value of i -th sample and the y_i denotes the corresponding true value.

Coefficient of Determination (R^2):

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2}$$

$$\text{where, } \bar{y} = \frac{\sum_{i=1}^N y_i}{N}$$

Mean Squared Error (MSE):

$$MSE = \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{N}$$

Mean Absolute Error (MAE):

$$MAE = \frac{\sum_{i=1}^N |y_i - \hat{y}_i|}{N}$$

Mean Absolute Percentage Error (MAPE):

$$MAPE = 100 * \frac{\sum_{i=1}^N \frac{|y_i - \hat{y}_i|}{|y_i|}}{N}$$

MSE was used as an internal evaluation metric in all cross-validation procedures. Other metrics were used to get more interpretable results. The use of terms ‘Accuracy’ and ‘Performance’ in this chapter is contextual and refers to one or more metrics defined above.

3.2.6 MSE and MAE Threshold

To understand whether a model was learning or not, we determined an approximate upper bound on MSE and MAE for the training and test set. It is expected that MSE and MAE would stay below this threshold if learning were successful. It was observed that when the training fails, the model predicts a constant value (i.e., the mean of the training data). Therefore, the threshold value for MSE and MAE was determined using the mean value of training data. The threshold values are shown in Table 3.5.

Table 3.5. Threshold values.

Metric	Training-Set Threshold	Test-Set Threshold
MSE	0.47	0.44
MAE	0.6	0.56

3.2.7 K-Fold Cross-Validation

In a typical k-fold cross-validation procedure, the training-set is split into k sets of approximately equal size. Then, the model is trained on k-1 sets, leaving one set as a test-set. Then, the performance of the trained model is evaluated on the left-out test-set. This procedure is repeated for every fold, and the average performance is reported. As every data point in the

training-set is evaluated as if it belongs to the test-set, the performance obtained from cross-validation is considered a reasonable estimate of out-of-sample performance. K-fold cross-validation gives robust out-of-sample performance for the model. It is a crucial evaluation technique, especially when the size of the dataset is small, and it becomes impractical to partition data into three sets (i.e., train, validation, test). 10-fold cross-validation with MSE loss was used in this study.

3.2.8 Feature Importance Score

Feature importance scores were computed using the hyperparameter optimized models of random forest, AdaBoost, and gradient boosting regression trained on all features from the corresponding feature set.

3.2.9 Pipeline

To assess the model performance, we developed a pipeline that combines all training and evaluation components into a single procedure. Given the train and test sets as inputs, the pipeline first performs hyperparameter-tuning for all models, then evaluates the performance of the optimized models on the train and test sets, and finally combines necessary results from each step in a single dataframe. The pipeline makes the training and evaluation easy, consistent and automatic for all models across different scenarios. A pictorial representation of the pipeline is shown in Figure 3.2. Different steps in the pipeline are described below:

Input: First, training and test data are provided as inputs.

Hyperparameter Tuning: In this step, optimized parameters of all twenty models are determined using the training-set, as described in section 3.2.4.

10-Fold Cross-Validation: In this step, the cross-validation performance of optimized models is evaluated using 10-fold cross-validation on the training-set. Three metrics (i.e., R^2 , MSE, and MAE) are recorded during the cross-validation for all models.

Training and Test-set Performance: In this step, the performance of all optimized models are evaluated on the training and test set. Three metrics (i.e., R^2 , MSE, and MAE) are recorded during the evaluation for all models.

Output: In this step, results from the above steps are combined into one dataframe containing the best set of parameters, 10-fold cross-validation performance, train, and test set performance of all models.

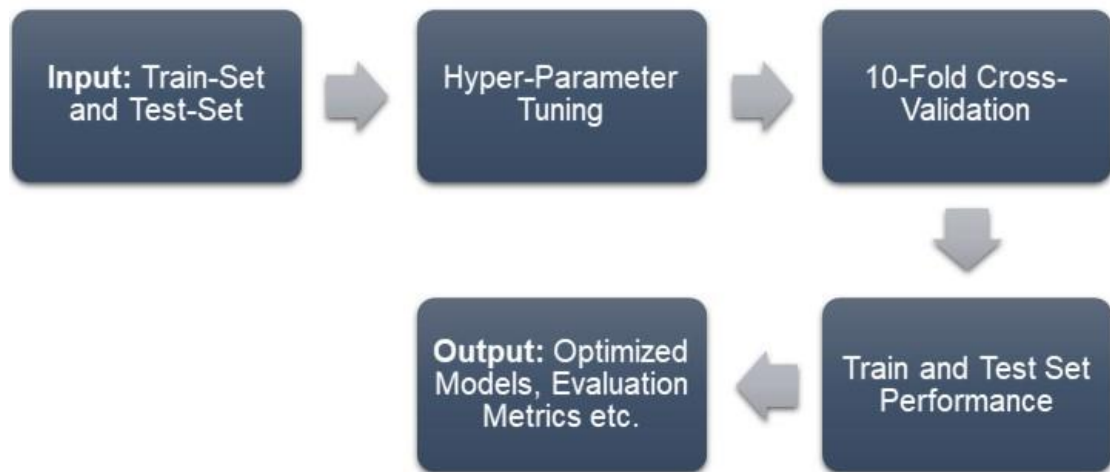


Figure 3.2. Pictorial representation of the training and evaluation pipeline.

3.3 Results and Discussion

3.3.1 Analysis of the Best-Performing Models

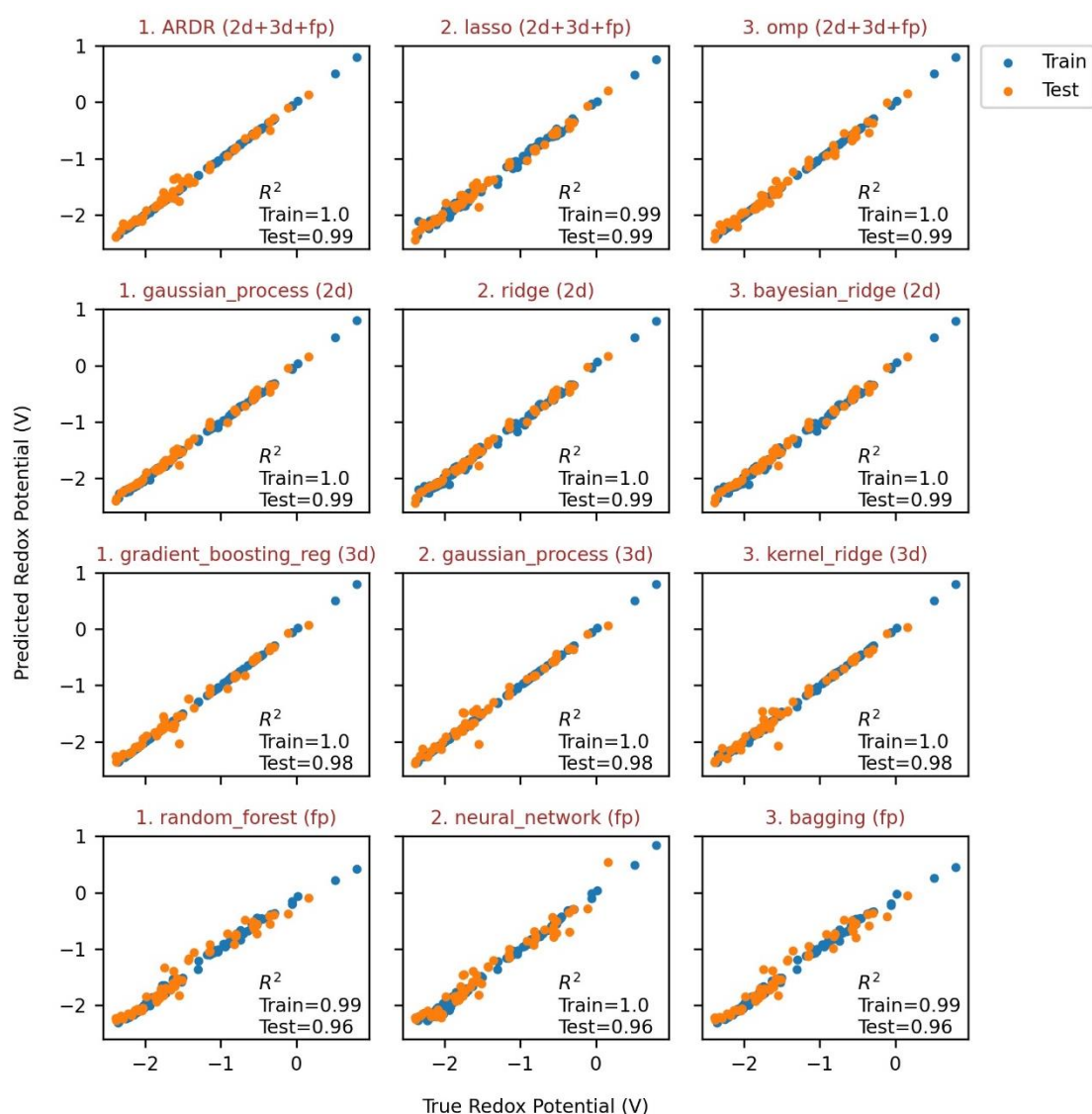


Figure 3.3. Machine learning prediction of redox potential (y-axis) vs. true redox potential (x-axis) of the three best-performing models in each feature set. The title of each plot indicates the model name, its rank, and the corresponding feature set used for training in brackets.

For accurate prediction of the redox potential of phenazine derivatives, we employed twenty different linear and non-linear machine-learning models, which are listed in Table 3.4. The whole dataset was shuffled and split randomly into train and test sets in a 7:3 ratio. The size of the training and test set was 129 and 56, respectively. Even though models were trained on a relatively small dataset, they achieved excellent performance on the unseen data (i.e., test-set). Figure 3.3 shows the redox potentials predicted by models on the y-axis and the corresponding true value of redox potentials on the x-axis. It can be seen that the majority of the models achieved an R^2 value of 0.99 on the test-set (R^2 values in the plots were rounded to two decimal places for clarity). Table 3.6 shows the fifteen best-performing models obtained in this study along with their R^2 , MSE, and MAE values on cross-validation (CV), training-set, and test-set.

All top twenty models not only had an outstanding performance on the training-set but also on the test-set (i.e., $R^2 > 0.98$, $MSE < 0.008 V^2$ and $MAE < 0.07 V$).

Table 3.6. Fifteen best-performing models. Models were trained on all features from the corresponding feature set.

Feature Set	Model Name	R^2 (CV)	MSE (CV) / V^2	MAE (CV) / V	R^2 (Train-set)	MSE (Train-set) / V^2	MAE (Train-set) / V	R^2 (Test-set)	MSE (Test-set) / V^2	MAE (Test-set) / V
<i>2d</i>	gaussian_process	0.9738	0.0078	0.0559	0.9991	0.0004	0.0136	0.9921	0.0035	0.0428
<i>2d</i>	ridge	0.9767	0.0069	0.0541	0.9960	0.0019	0.0298	0.9916	0.0037	0.0454
<i>2d</i>	bayesian_ridge	0.9751	0.0072	0.0537	0.9964	0.0017	0.0282	0.9915	0.0037	0.0452
<i>2d</i>	neural_network	0.9644	0.0108	0.0649	0.9988	0.0005	0.0133	0.9909	0.0040	0.0447
<i>2d</i>	kernel_ridge	0.9738	0.0077	0.0586	0.9947	0.0025	0.0372	0.9896	0.0046	0.0501
<i>2d</i>	omp	0.9160	0.0546	0.0798	0.9940	0.0028	0.0389	0.9876	0.0055	0.0567
<i>2d</i>	ARDR	0.9768	0.0074	0.0582	0.9956	0.0020	0.0333	0.9874	0.0055	0.0540
<i>2d+3d+fp</i>	ARDR	0.9519	0.0132	0.0600	0.9999	0.0000	0.0052	0.9873	0.0056	0.0473
<i>2d+3d+fp</i>	lasso	0.9826	0.0064	0.0593	0.9937	0.0029	0.0429	0.9868	0.0058	0.0519
<i>2d+3d+fp</i>	omp	-0.5387	0.3913	0.2317	1.0000	0.0000	0.0000	0.9861	0.0061	0.0628
<i>2d</i>	lasso	0.9768	0.0078	0.0680	0.9901	0.0046	0.0550	0.9857	0.0063	0.0599
<i>2d+3d+fp</i>	gaussian_process	0.9849	0.0058	0.0509	1.0000	0.0000	0.0021	0.9855	0.0064	0.0478
<i>2d+3d+fp</i>	bayesian_ridge	0.9851	0.0055	0.0509	0.9993	0.0003	0.0114	0.9853	0.0064	0.0483
<i>2d+3d+fp</i>	ridge	0.9858	0.0053	0.0499	0.9989	0.0005	0.0144	0.9849	0.0066	0.0494
<i>2d+3d+fp</i>	gradient_boosting_reg	0.9713	0.0141	0.0673	1.0000	0.0000	0.0009	0.9849	0.0066	0.0546

3.3.2 Assessment of Model Performance on Four Feature Sets

The performance of machine learning models depends on the type and the quality of features. Therefore, it is important to identify the best set of features that achieve high prediction accuracy. Hence, we assessed model performance on four sets of features given in Table 3.2. The goal here was to understand how different types of molecular features affected model performance. The model performance on each feature set was assessed using the pipeline described in section 3.2. Fifteen best-performing models in Table 3.6 were obtained after assessing model performance independently on four feature sets. The ‘*Feature Set*’ column in Table 3.6 shows the corresponding feature set used for the training. Gaussian processes regression trained on 2D features achieved the highest prediction accuracy in this study. The negative value of R^2 (CV) for the orthogonal matching pursuit (*omp*) model in Table 3.6 suggests that it may not generalize well to the unseen data. A similar trend was observed for a few other linear models (Figure 3.4). Nine models, including the top seven models in Table 3.6, were trained on the ‘*2d*’ feature set. The rest of the models in Table 3.6 were trained on the ‘*2d+3d+fp*’ feature set. We did not observe any model trained on either ‘*3d*’ or ‘*fp*’, even in the top twenty best-performing models. The best-performing model in each feature set, along with their test-set performance (i.e., R^2 , MSE, and MAE), is shown in Table 3.7. From the Table 3.7, we observed following order among the feature sets with respect to the prediction accuracy: $2d > 2d+3d+fp > 3d > fp$. Therefore, we conclude that 2D features are more

informative than 3D and molecular fingerprint features in predicting the redox potential of phenazine derivatives in DME. We also observed that linear models (e.g., *ARDR*, *lasso*, *omp*, *ridge*, *bayesian_ridge*) perform better than non-linear models on ‘*2d+3d+fp*’ feature set, whereas non-linear models (e.g., *gradient_boosting_reg*, *gaussian_process*, *random_forest*, *neural_network*) perform better than linear models on ‘*3d*’ and ‘*fp*’ feature sets. This observation suggests that linear models should be preferred when the feature set consists of different features (i.e., 2D, 3D, and molecular fingerprints), and non-linear models should be preferred when the feature set contains either 3D or molecular fingerprint features. Any model could be used for 2D features. Linear models are generally faster than non-linear models due to their simple structure but are not preferred due to low accuracy. The results obtained here show that linear models could give accurate predictions compared to non-linear models in certain combinations of features (‘*2d+3d+fp*’ feature set in this study). Utilizing linear models in these scenarios could significantly reduce the training and inference time.

Table 3.7. Test-set performance of the best-performing models in each feature set. Models were trained on all features from the corresponding feature set.

Featue Set	Model Name	R ² (Test-set)	MSE (Test-set) / V ²	MAE (Test-set) / V
<i>2d+3d+fp</i>	ARDR	0.9873	0.0056	0.0473
<i>2d</i>	gaussian_process	0.9921	0.0035	0.0428
<i>3d</i>	gradient_boosting_reg	0.9788	0.0093	0.0573
<i>fp</i>	random_forest	0.9583	0.0183	0.1012

3.3.3 Cross-Validation and Out-of-Sample Performance

10-fold cross-validation (CV) performance (i.e., R², MSE, and MAE) obtained from the pipeline is shown in Figure 3.4 for all twenty models. Cross-validation gives a reasonable estimate of out-of-sample performance (i.e., performance on unseen data). ‘*2d+3d+fp*’, ‘*2d*’, and ‘*3d*’ feature sets had the acceptable CV performance (i.e., MSE and MAE below their threshold value) on most models except for four linear models (i.e., *linear_reg*, *omp*, *PA*, *huber*). The computation of threshold values for MSE and MAE is described in section 3.2. These four linear models had negative R² value, high MSE, and high MAE (i.e., close to threshold) for at least one feature set. ‘*fp*’ feature set had the worst CV performance on all models. Three linear models (i.e., *omp*, *PA*, *huber*) had poor CV performance on the ‘*3d*’ feature set. Performance on the test-set (i.e., R², MSE, and MAE) obtained from the pipeline is shown in Figure 3.5 for all twenty models. As models never saw the test-set, this gives an even better estimate of out-of-sample performance than cross-validation. All feature sets had an acceptable test-set performance on all models except linear regression. Linear regression had a poor test-set performance on ‘*2d*’ and ‘*fp*’ feature sets. Furthermore, the averaged training and test set performance (i.e., R², MSE, and MAE values were averaged over all models) for each feature set are shown in Table 3.8. Values of linear regression were not considered in the average due to very high errors. In Table 3.8, ‘*2d+3d+fp*’ and ‘*3d*’ feature sets had better training-set performance and poorer test-set performance in comparison to the ‘*2d*’ feature set. This indicates that 2D features are better at generalizing to unseen data than 3D and molecular

fingerprint features. Trend analysis of Figure 3.4, Figure 3.5, and Table 3.8 revealed the previously observed order of feature set performance, $2d > 2d+3d+fp > 3d > fp$.

Table 3.8. Training and test set performance of four feature sets averaged over all models except linear regression. Models were trained on all features from the corresponding feature set.

Feature Set	R ² (Train-set)	MSE (Train-set) / V ²	MAE (Train-set) / V	R ² (Test-set)	MSE (Test-set) / V ²	MAE (Test-set) / V
$2d+3d+fp$	0.9926	0.0035	0.0301	0.9718	0.0124	0.0733
$2d$	0.9876	0.0058	0.0426	0.9729	0.0119	0.0734
$3d$	0.9917	0.0039	0.0335	0.9535	0.0204	0.0899
fp	0.9718	0.0132	0.0506	0.9028	0.0427	0.1511

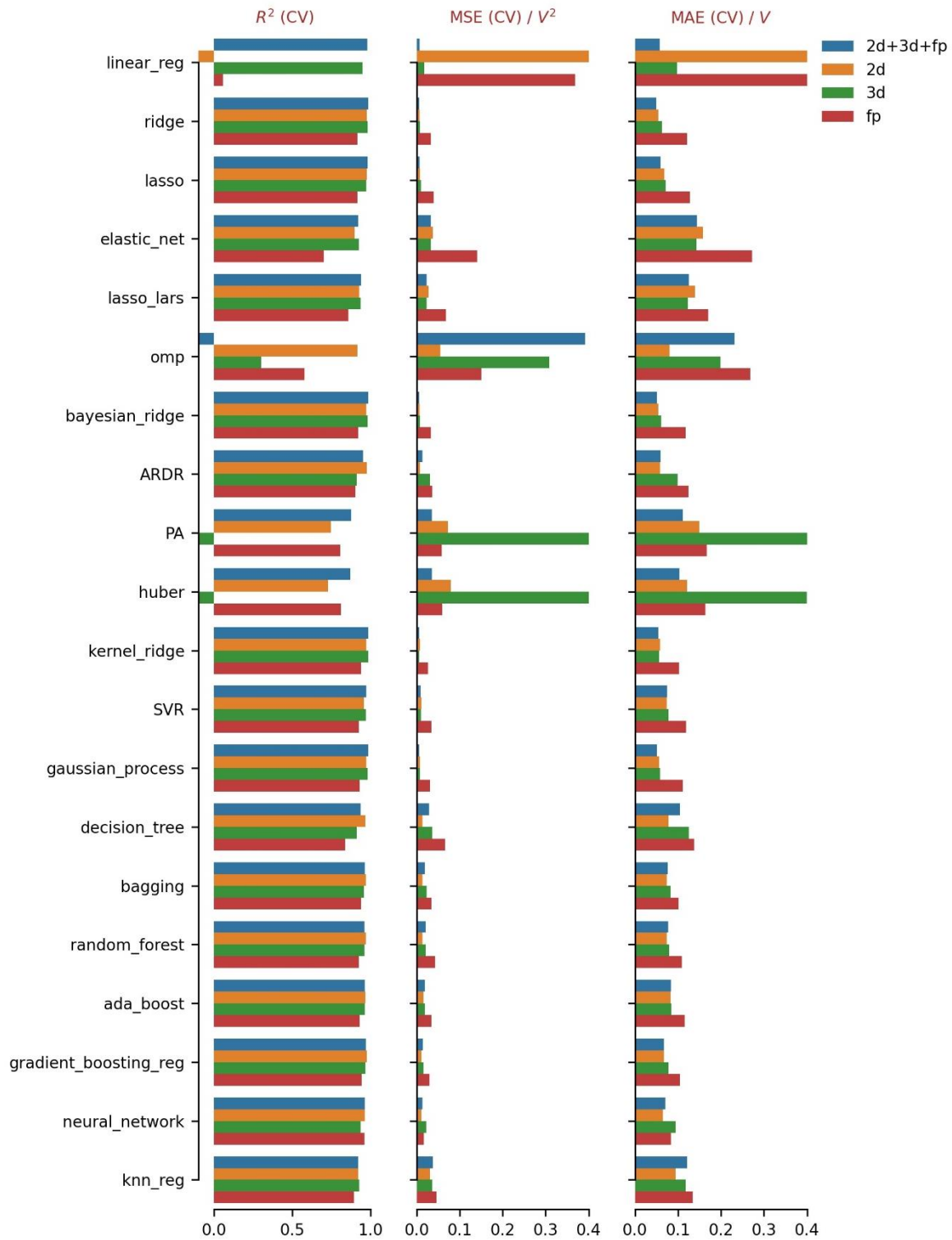


Figure 3.4. 10-Fold cross-validation performance of twenty models. Models were trained on all features from the corresponding feature set.

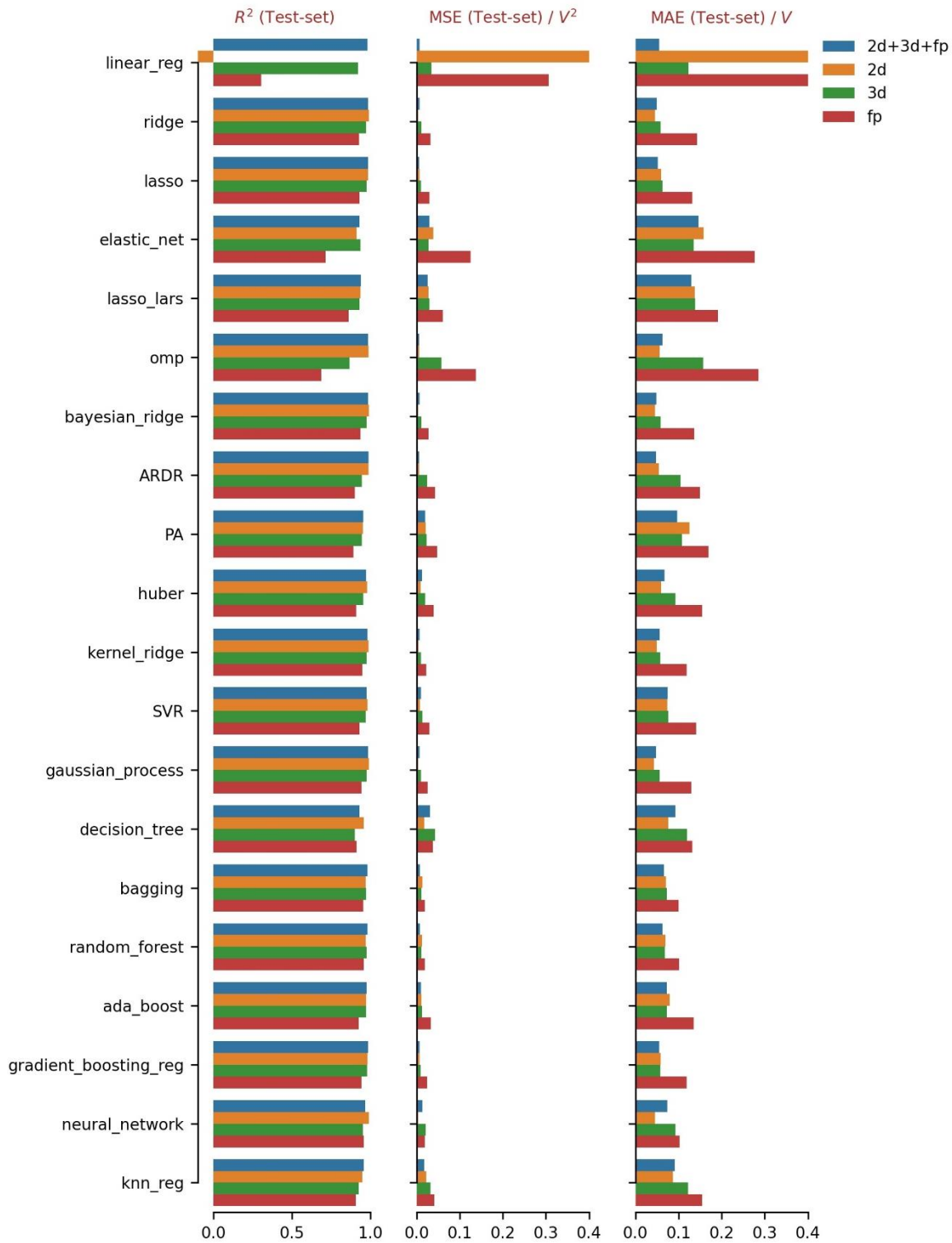


Figure 3.5. Test-set performance of twenty models. Models were trained on all features from the corresponding feature set.

3.3.4 Feature Importance Analysis

Here, we performed feature importance analysis for each feature set to identify the most important features. We used random forest, AdaBoost, and gradient boosting regression to

calculate the feature importance score. Figure 3.6 - Figure 3.9 show feature importance histograms for '2d', '3d', 'fp', and '2d+3d+fp' feature sets, respectively. Only twenty features with the highest scores are shown in the histograms. The most important features in the '2d' feature set are *SlogP_VSA4*, *fr_NH0*, *VSA_Estate3*, and *VSA_Estate4*. *SlogP_VSA4* includes the LogP⁶⁵ and Van der Waals surface area contributions from all atoms in the molecule. *fr_NH0* is the number of tertiary amines⁶⁶. *VSA_Estate3* and *VSA_Estate4* are calculated using EState indices⁶⁷ and Van der Waals surface area contributions of all the atoms in a molecule. Many graph-based features like *Kappa2*, *BertzCT*, *Chi1*, *Chi2n*, *HallKierAlpha*⁶⁸, and some chemically intuitive features like *fr_ArN* (i.e., number of N functional groups attached to aromatic ring⁶⁶), *MinPartialCharge*, *MaxAbsPartialCharge* are also observed in the top twenty features. In the case of '3d' feature set, *RDF_120*, *RDF_90*, *RDF_125 WHIM_90*, *WHIM_86* are among the top 3D features (*RDF*, *WHIM* are 1D vectors).⁶⁹ The number at the end of the feature names denotes its position in the corresponding feature vector. Some components of *MORSE* and *GETAWAY* feature vectors⁶⁹ are also observed in the top twenty features. Only two components of *Autocorr3D* were observed in one of the histograms (i.e., AdaBoost histogram), suggesting that *Autocorr3D*⁶⁹ is a relatively less important 3D feature. None of the scalar 3D features were observed in feature importance histograms, suggesting that scalar 3D features are less important than vectorial 3D features. We had only three types of fingerprints in the 'fp' feature set (i.e., RDKit, ECFP4, MACCS keys). RDKit Fingerprints are daylight-like fingerprints computed from hashing molecular subgraphs.⁶⁶ ECFP4⁷⁰ or Extended Connectivity Circular Fingerprints are computed from the bag-of-word representation of the local molecular neighborhood. Four in ECFP4 denotes the radius of the local neighborhood. MACCS keys are computed using the SMARTS-based implementation of the 166 public MACCS keys.⁷¹ Components only from RDKit and ECFP4 were among the top features, whereas only one component from the MACCS keys was observed in one of the histograms (the gradient boosting regression histogram). This indicates that the MACCS key fingerprint does not contain enough molecular information to predict the redox potential. Figure 3.9 shows the feature importance histograms for the '2d+3d+fp' feature set. This feature set contains all the features from the '2d', '3d', and 'fp' feature sets. The most important features in this feature set were also the most important features in their respective set. Top features are mainly from '3d' and '2d' feature sets, and only one component from the ECFP4 feature vector was observed in the lower end of the gradient boosting regression histogram. This again shows that molecular fingerprints are the least informative among all features. The feature importance histogram of the '2d+3d+fp' feature set contains a few '2d' features and predominantly '3d' features. This can be attributed to how fast the feature importance score diminishes from the most important features to the least important features. Feature importance scores in the '2d' feature set (Figure 3.6) diminish faster than in the '3d' feature set (Figure 3.7). This also indicates that very few 2D features are required to predict the redox potential compared to 3D features.

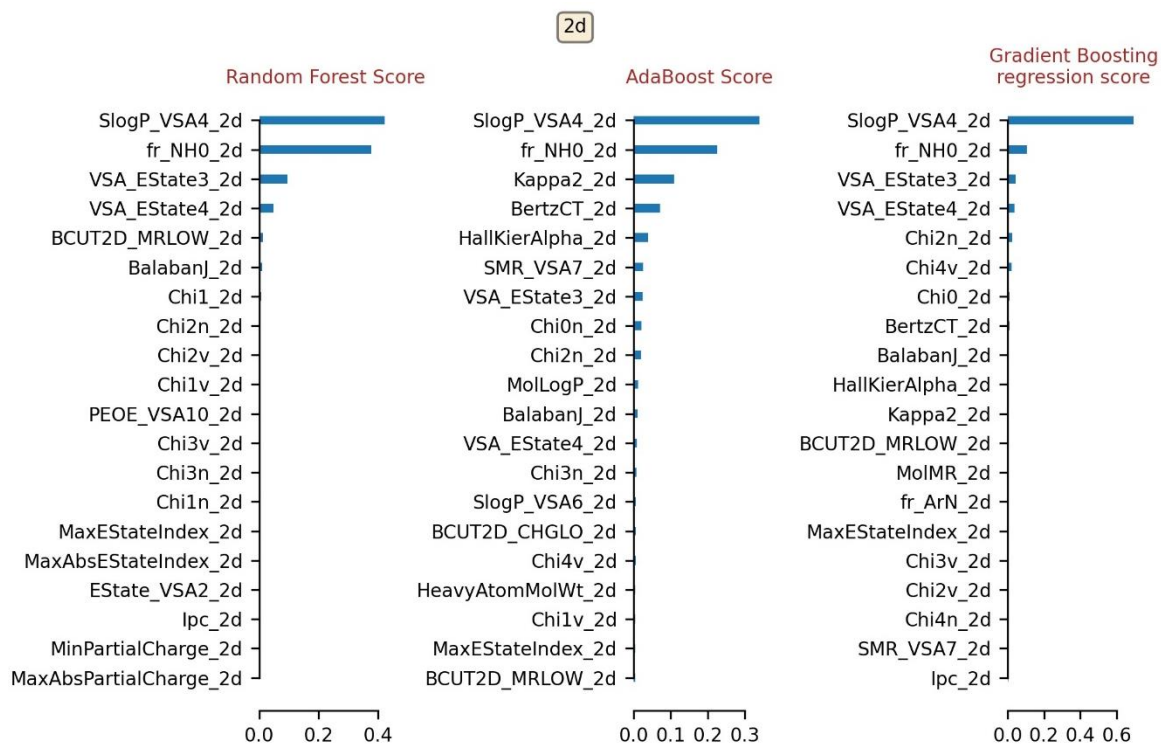


Figure 3.6. Feature importance histograms of '2d' feature set.

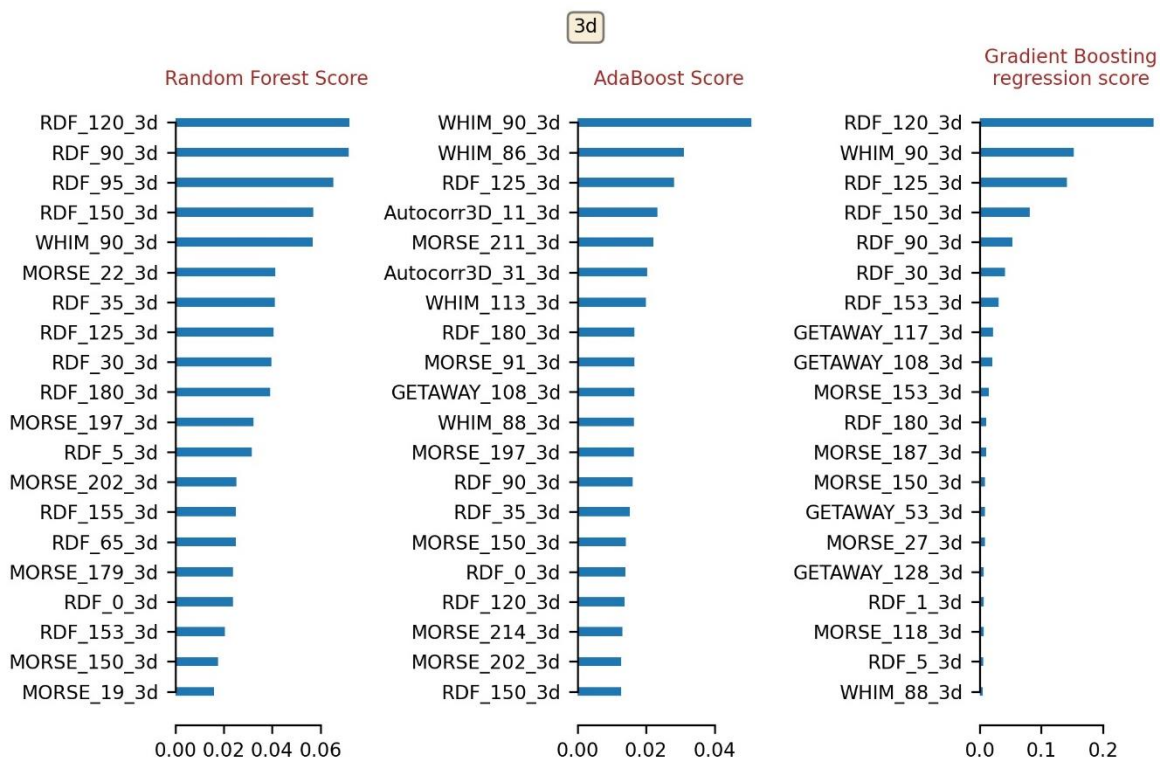


Figure 3.7. Feature importance histograms of '3d' feature set.

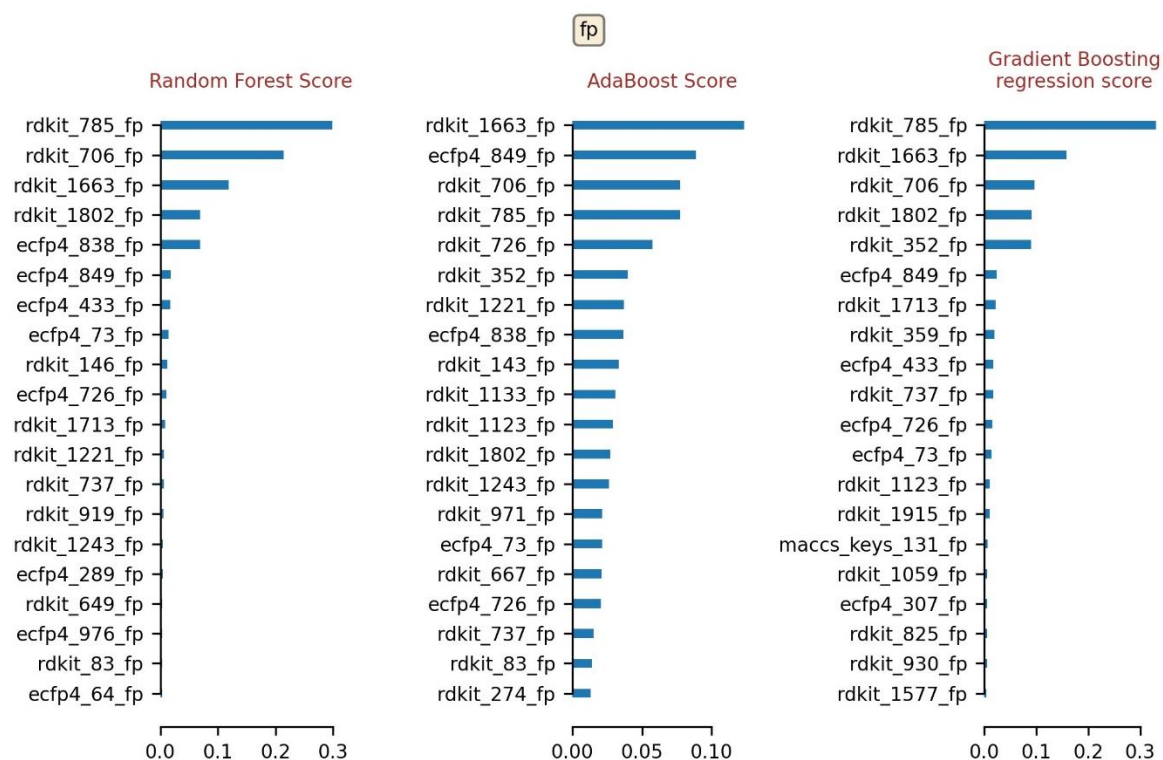


Figure 3.8. Feature importance histograms of 'fp' feature set.

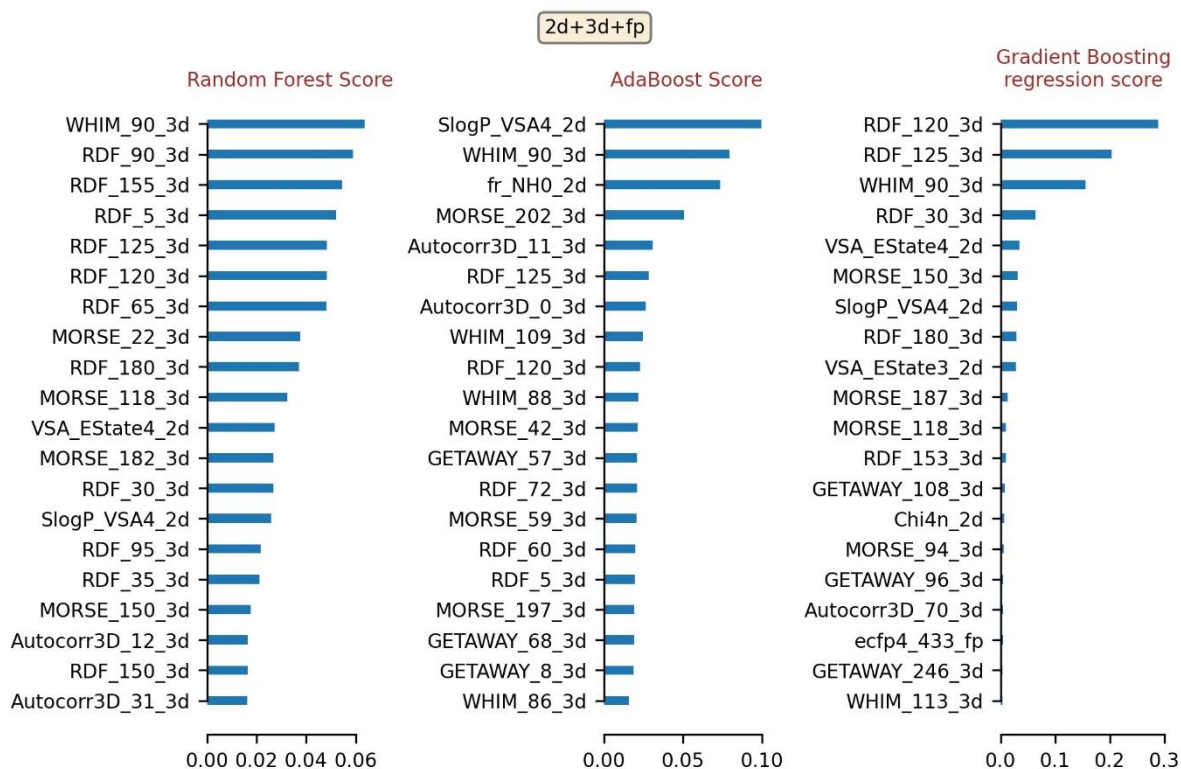


Figure 3.9. Feature importance histograms of '2d+3d+fp' feature set.

3.3.5 Effect of Feature Size on Model Performance

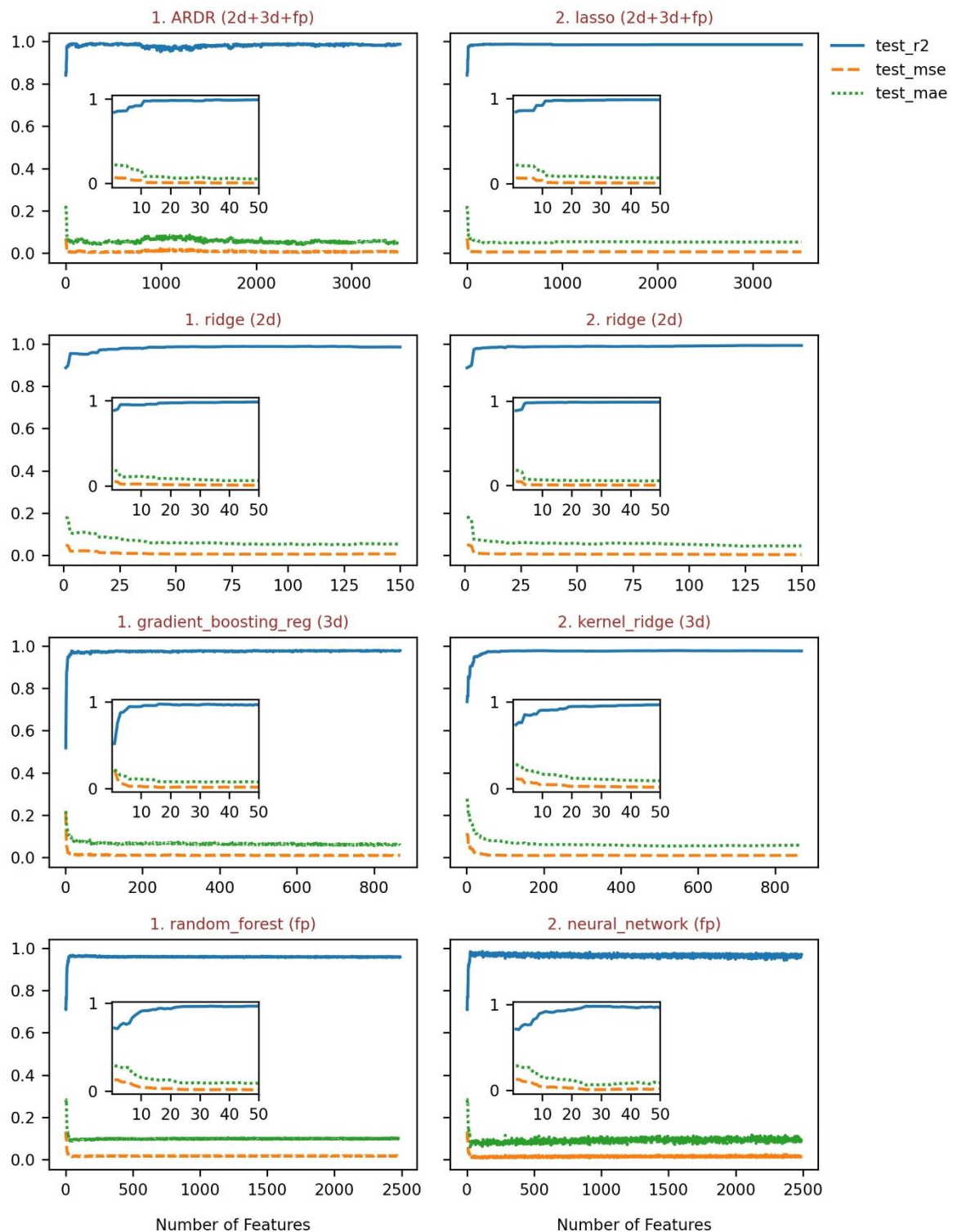


Figure 3.10. Model performance vs. number of features

Previous feature importance studies suggest that not all features may be necessary to achieve high predictive performance. To confirm this hypothesis, the two best-performing models in each feature set were selected and re-trained on the subset of features. Features were sorted in descending order based on the random forest scores. Models were re-trained starting with a

single feature to the complete set of features, and three metrics (i.e., R^2 , MSE, MAE) on the test-set were recorded. The results for each feature set are shown in Figure 3.10. Inset plots show the same data for the first 50 features. Plots corresponding to the '2d' feature set quickly saturate (after ~5 features), suggesting that only a small number of 2D features are required to predict the redox potential accurately. Plots of the '3d' feature set seem to saturate after twenty features, whereas plots corresponding to the 'fp' feature set saturate slowly and require more than twenty features to achieve similar performance. In the case of the '2d+3d+fp' feature set, plots seem to saturate around 15-20 features and look approximately similar to a linear combination of the plots from the '2d', '3d', and 'fp' feature sets. These plots clearly show that not all features are required to attain a high level of prediction accuracy.

3.3.6 Assessment of Model Performance on Limited Number of Features

To gain insight into the quality of predictions when models are trained on a limited number of features, all models were re-trained on the subset of features using the pipeline (section 3.2). Features were chosen from the array of features sorted based on the random forest score. The number of features were varied from five to twenty in a step of five. Figure 3.11 shows the test-set performance of models when trained on a small number of features from the '2d+3d+fp' feature set. The performance on a full set of features is also shown for reference. We observed that the model performance generally increases with the number of features. A few exceptions were also observed. Some models (e.g., *PA*, *huber*, *neural network*, and *knn_reg*) had better performance on the top twenty features than a full set of features. Moreover, *decision_tree* performed slightly better on the top fifteen features than a full set of features. Similar trends were observed for other feature sets as well. However, in the case of the 'fp' feature set, *SVR* and *knn_reg* showed better performance on the top twenty features than a full set of features. We also analyzed which feature set was able to achieve the highest accuracy when models were trained on a small number of features. Figure 3.12 shows the test-set performance of all models when trained on only the top five features from each feature set. '2d' feature set achieved an R^2 value as high as 0.9869 with only five features with the 'bagging' model. The '2d+3d+fp' and '3d' feature set performance was similar but sub-par to the '2d' feature set. The similarity in performance of '2d+3d+fp' and '3d' feature sets could be attributed to the similarity in their top five features (see Figure 3.7 and Figure 3.9). The 'fp' feature set performed poorly with only five features. These results were consistent across all models. Similar results were also obtained for the models trained on the top ten, fifteen, and twenty features. Furthermore, a significant decrease in the training and inference time was observed while using the limited number of features for the training. These results suggest that training and inference time could be reduced by using a small number of features while maintaining a good accuracy level.

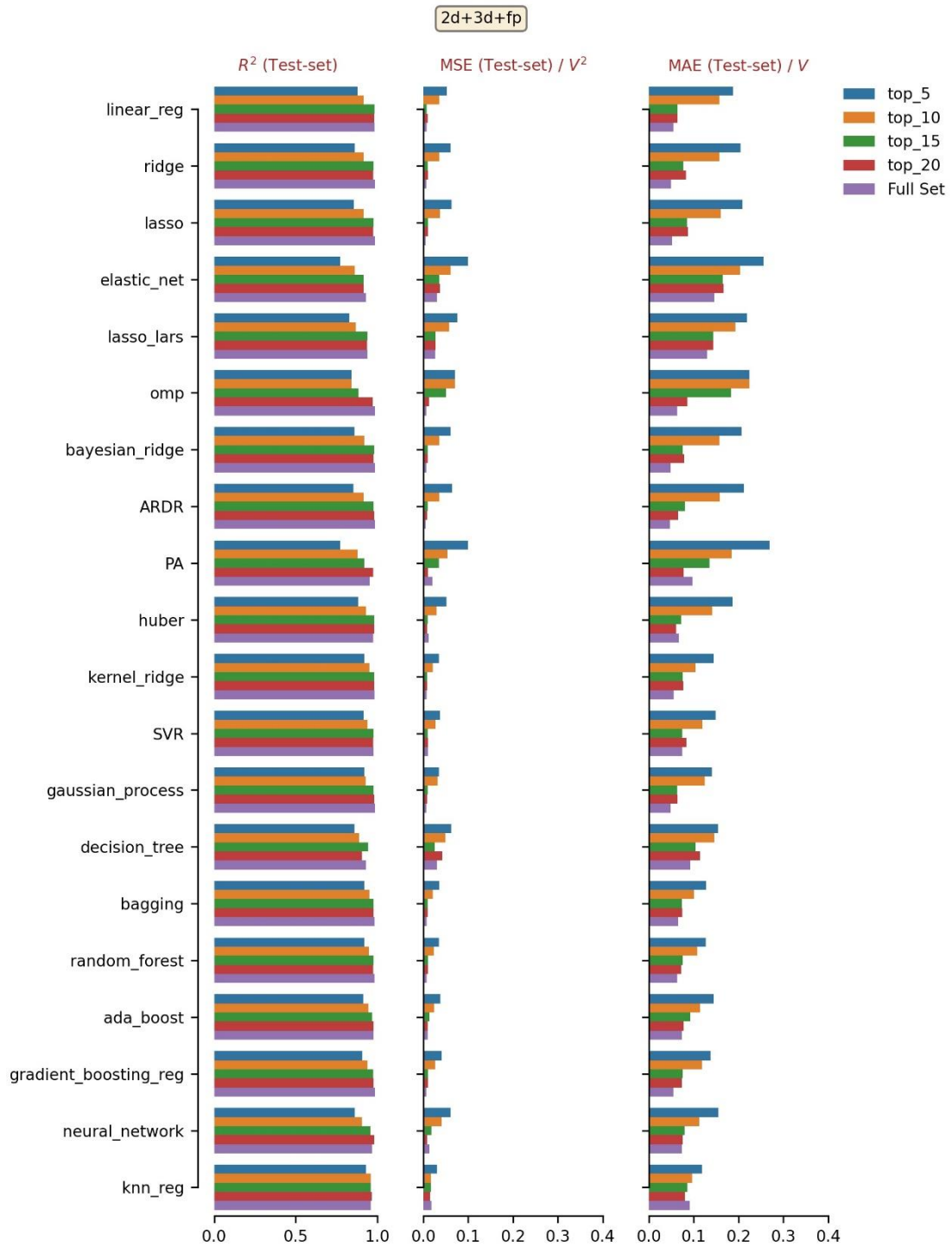


Figure 3.11. Test-set performance of twenty models trained on top-5, 10, 15, and 20 features from ‘ $2d+3d+fp$ ’ feature set. The top most important features were selected based on the random forest score. Full feature set performance is shown for reference.

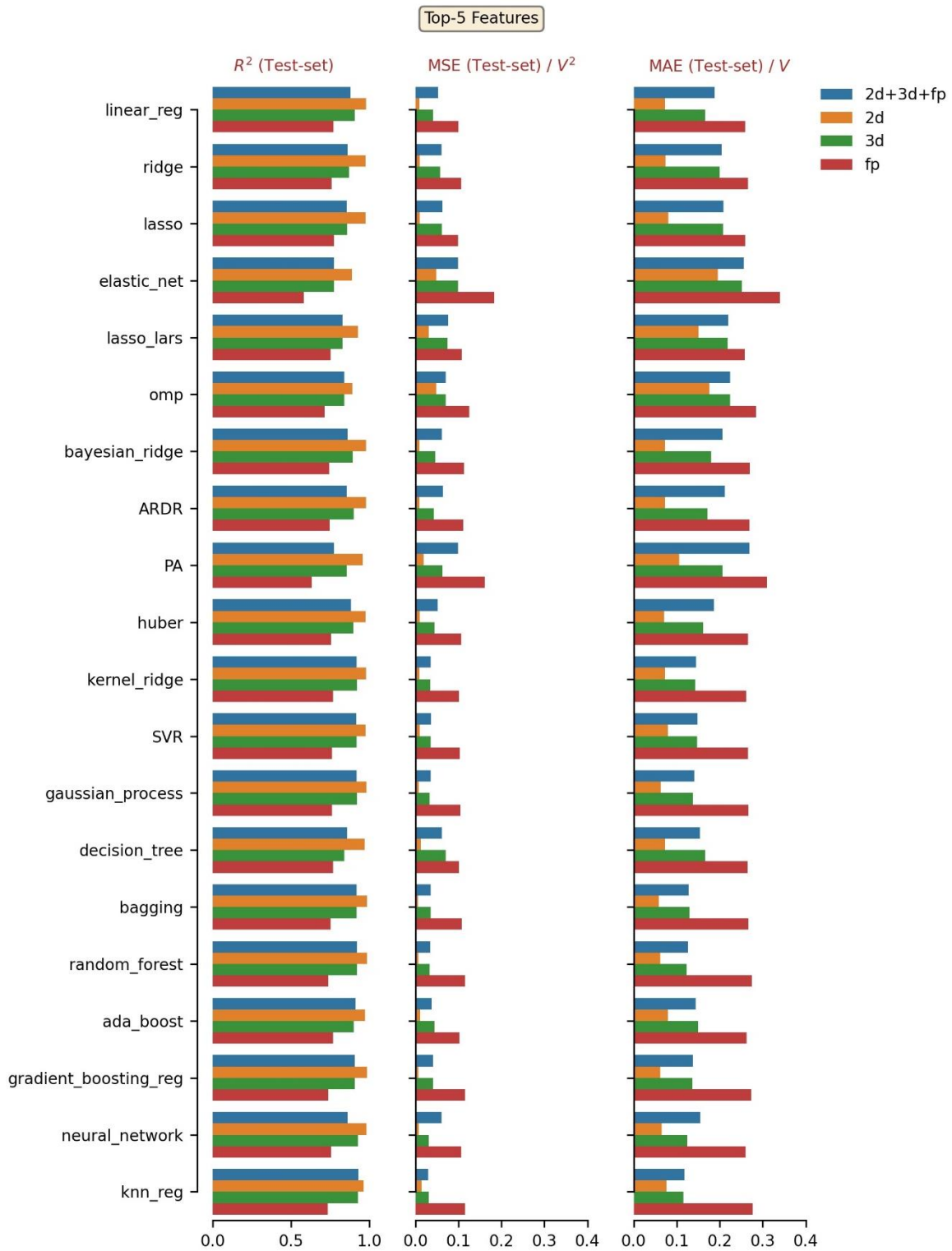


Figure 3.12. Test-set performance of twenty models trained on the five most important features from the corresponding feature set. Features were selected based on the random forest score.

3.3.7 Analysis of the Predictive Performance with respect to Individual Functional Groups

Here, we evaluate the prediction accuracy of the best-performing models in each feature set with respect to different functional groups attached to the phenazine ring. Twenty different functional groups were present in the phenazine derivatives investigated in this study. Training-set and Test-set predictions were obtained from the best performing model in each feature set (see Table 3.7). Figure 3.13 shows the Mean Absolute Percentage Error (MAPE) for each functional group (FG) present in the test-set. MAPE in Figure 3.13 (a) is averaged over FGs and four feature sets.

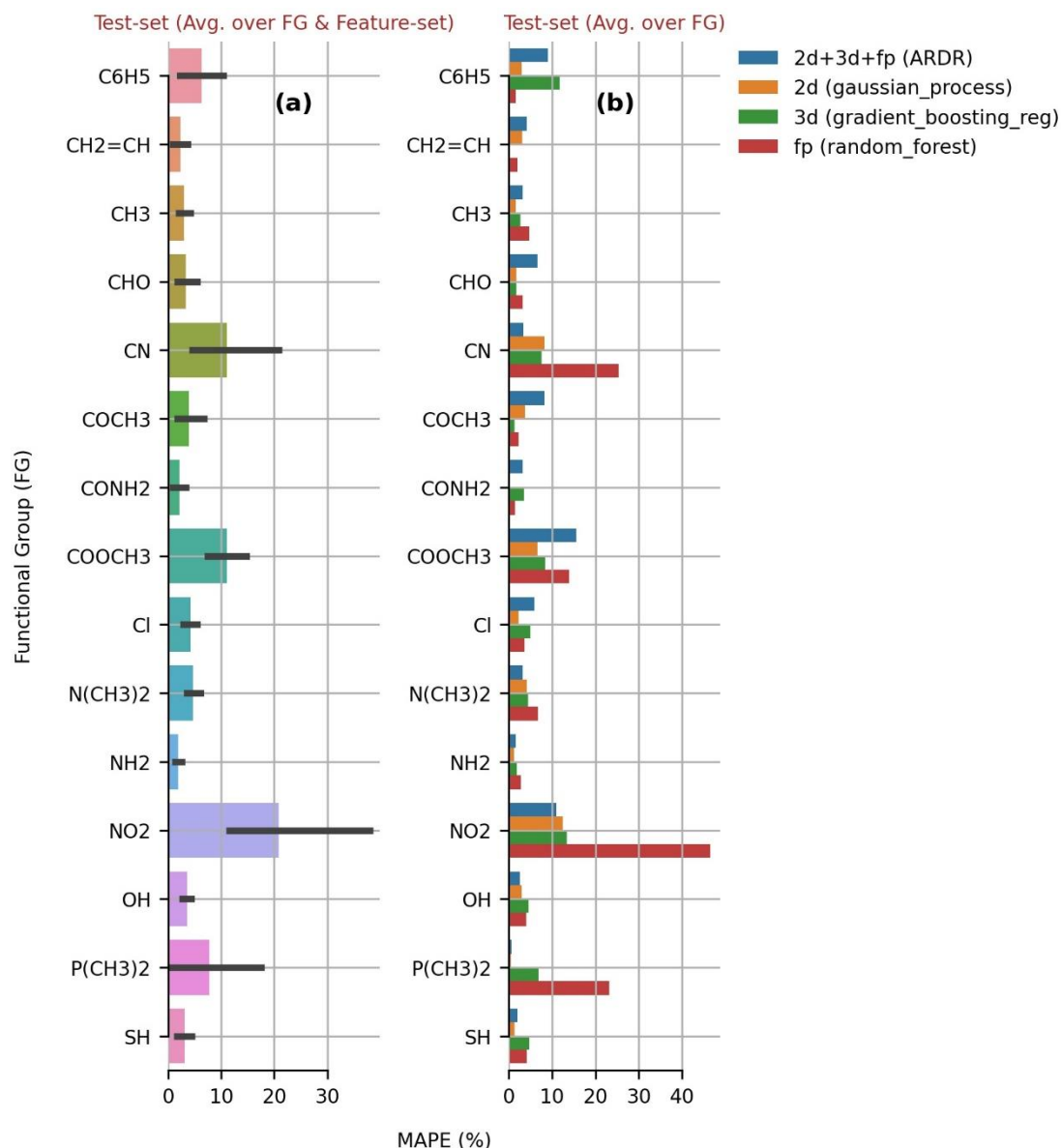


Figure 3.13. Functional group (FG) vs. Mean Absolute Percentage Error (MAPE) on the test-set. (a) MAPE is averaged over FG and feature sets. (b) MAPE is averaged over only FG. The test-set predictions were obtained from the corresponding best-performing model.

On the other hand, MAPE, shown in Figure 3.13 (b), is averaged over only FGs. MAPE for the majority of the functional groups was well below 10.0%. Even though some functional

groups appeared only once in the training-set (see Figure 3.14), most of the models were able to predict the redox potential with minimal error. The functional group $-\text{COCH}_3$ was present only in the test-set, which means that models never saw this functional group during the training. Still, the error in its prediction was less than 5.0%. This shows that models successfully learned the hidden patterns between features and redox potential from the training-set, resulting in low generalization errors.

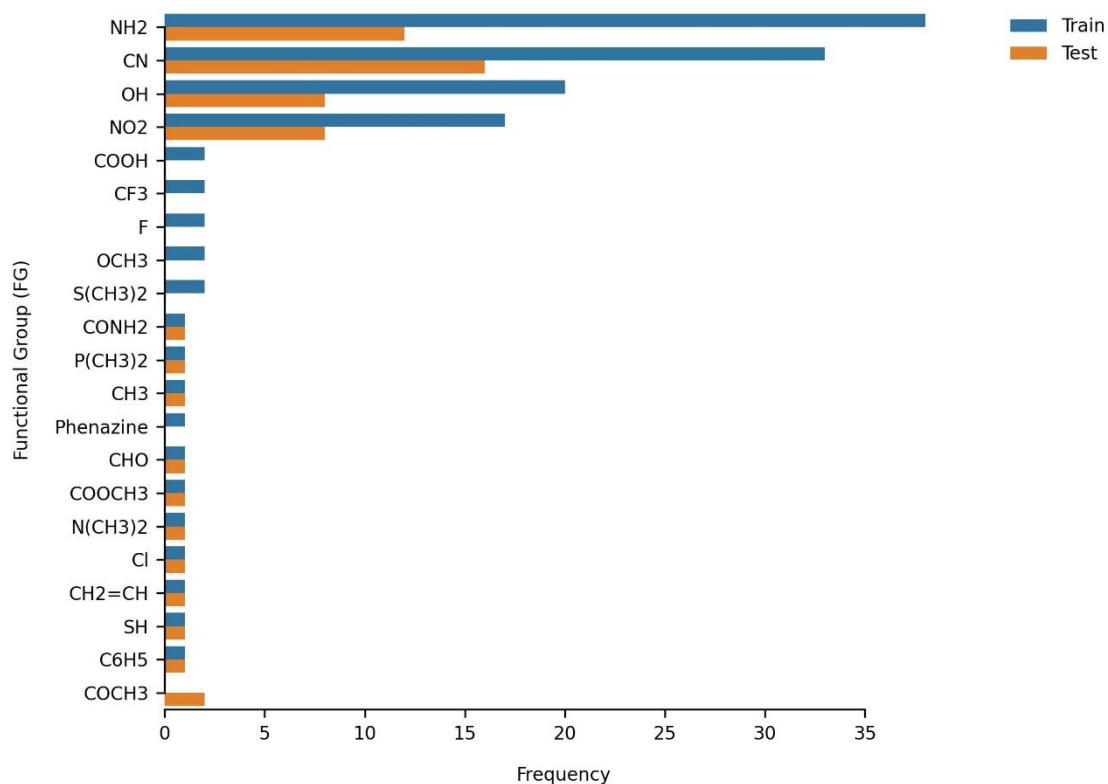


Figure 3.14. Distribution of functional groups in training and test sets.

3.3.8 Error Analysis

Even though models had low generalization errors, some functional groups exhibited relatively high MAPE, e.g., $-\text{CN}$, $-\text{NO}_2$, and $-\text{COOCH}_3$. High errors of $-\text{CN}$ and $-\text{NO}_2$ could be attributed to a small number of compounds with a redox potential close to zero. Unfortunately, the data used in this study contains a very small number of compounds with redox potential close to zero. The whole dataset contains only 18 compounds (~9.7% of the total data) having redox potential greater than -0.5 V (Figure 3.15 (a)). Therefore, models had less information to learn from the region near zero redox potential. This is why test-set predictions near zero redox potential had relatively high errors (Figure 3.16). On the other hand, having access to a large enough dataset in the region below -0.5 V, models were able to learn the hidden patterns. This resulted in low prediction errors for the compounds with redox potential below -0.5 V, even for compounds with less than one sample in the training-set. Functional groups $-\text{CN}$ and $-\text{NO}_2$ contain some compounds with redox potential greater than -0.5 V (see Figure 3.15 (b)), responsible for the high prediction errors observed in Figure 3.13. We also observed a slight

increase in the errors around -1.5 V in Figure 3.16, which could be attributed to the relatively low number of data points in the region near -1.5 V. This is why $-\text{COOCH}_3$ (avg. redox potential -1.57 V) also showed a slightly high prediction error. The red curve in Figure 3.16 shows the normalized distribution over redox potential (i.e., density) for the whole dataset.

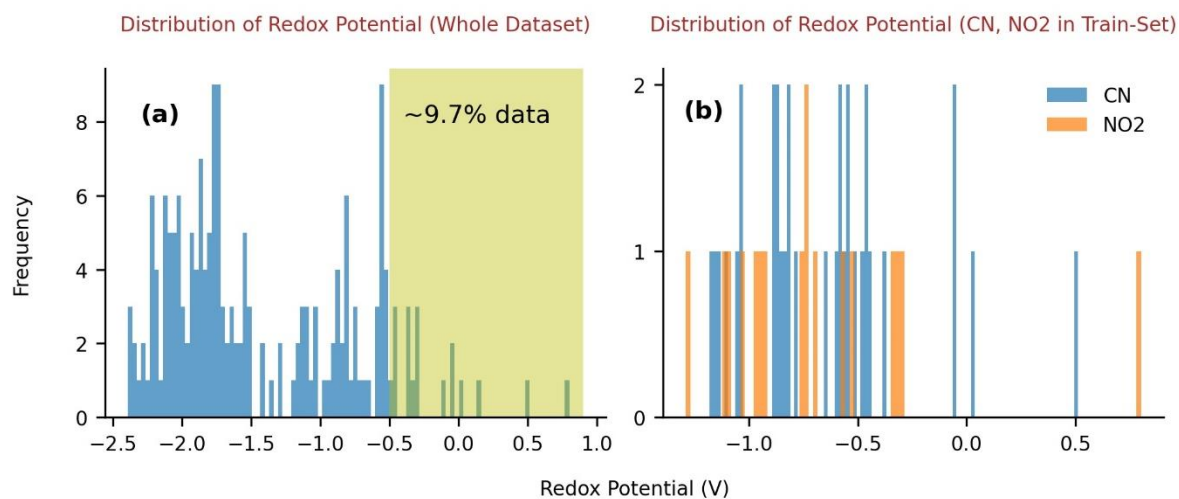


Figure 3.15. Distribution of redox potential (a) of the whole dataset. (b) of $-\text{CN}$, $-\text{NO}_2$ functional groups in training-set

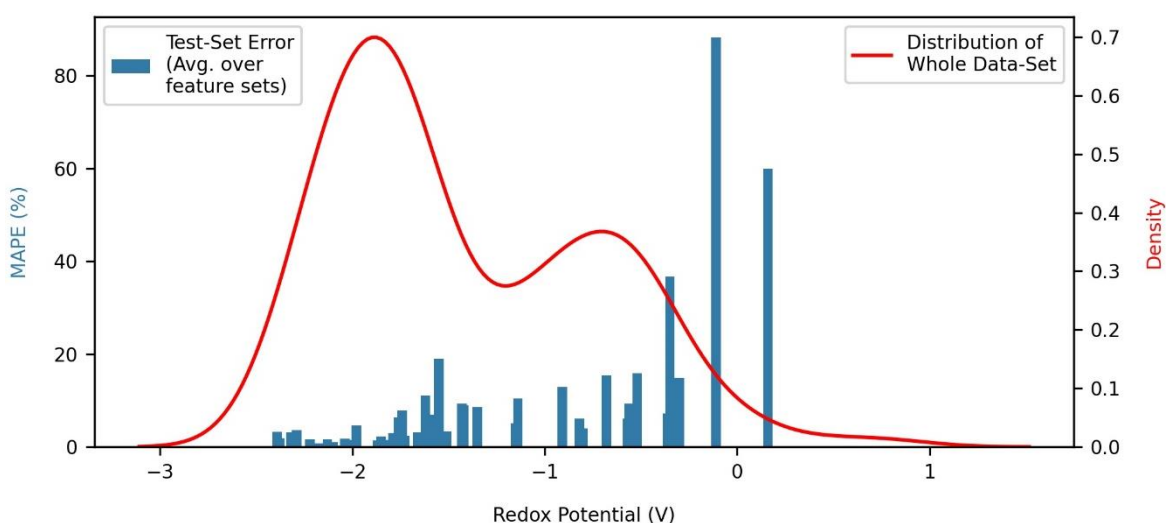


Figure 3.16. MAPE vs. redox potential. The final MAPE on the y-axis was calculated by averaging the MAPE obtained from the best-performing model in each feature set. The red curve depicts the normalized distribution of redox potential (i.e., density) for the whole dataset.

3.4 Conclusions

In this study, we have investigated twenty linear and non-linear machine learning models to predict the redox potential of phenazine derivatives in DME. Both linear and non-linear models trained on a small dataset were able to achieve excellent prediction accuracy on the test-set (i.e., $R^2 > 0.98$, $MSE < 0.008 \text{ V}^2$, and $MAE < 0.07 \text{ V}$). Features used in this study were intentionally chosen to be easily computable from open-source libraries that do not require DFT calculations or experimental measurements, making our approach readily adaptable for similar studies. Model performance was assessed on four feature sets containing different features (i.e., 2D, 3D, and molecular fingerprints) using a convenient pipeline developed in this work. This pipeline combines different training and evaluation components in a single sub-routine, making the whole process easy, consistent, and automatic for all models in different scenarios. Gaussian processes regression trained on 2D features achieved the highest prediction accuracy. The analysis of model performance on four feature sets revealed an interesting order with respect to prediction accuracy: $2d > 2d+3d+fp > 3d > fp$. Average performance analysis also showed that 2D features are better at generalizing to unseen data than 3D and molecular fingerprint features. Therefore, we conclude that 2D features capture important molecular properties necessary for predicting the redox potential of phenazine derivatives in DME solvent. It was observed that linear models out-perform non-linear models on '2d+3d+fp' feature set, whereas non-linear models perform better than linear models on '3d' and 'fp' feature sets. Therefore, for predicting the redox potential of phenazine derivatives, linear models should be preferred when the feature set contains different types of features, and non-linear models should be preferred when the feature set contains either 3D or molecular fingerprint features. Due to the simple structure, linear models have fast training and inference time but suffer from low accuracy. Results obtained here show that lower training and inference times are possible with the linear models that out-perform non-linear models when the dataset contains different types of features (i.e., 2D, 3D, and molecular fingerprints). Feature importance analysis showed that features related to Van der Waals surface areas, e.g., *SlogP_VSA4*, *fr_NH0*, *VSA_Estate3*, and *VSA_Estate4* were the most important 2D features. *RDF_120*, *RDF_90*, *RDF_125*, *WHIM_90*, and *WHIM_86* were the most important 3D features. *RDKit*, *ECFP4* were the most important molecular fingerprint features. Some features based on molecular structure and charges, e.g., *fr_ArN*, *MinPartialCharge*, and *MaxAbsPartialCharge*, were also observed during feature importance analysis. Feature importance analysis also suggested that very few 2D features are required to predict the redox potential compared to 3D and molecular fingerprint features. This observation was confirmed by re-training models with the subset of features starting from a single feature to a full set of features. Model performance was generally observed to increase with the number of features, but some exceptions were also observed for which the small number of features performed better than a full set of features. A bagging meta-estimator trained on only the top five 2D most important features was able to achieve R^2 value as high as 0.9869. A significant reduction in the training and inference time was observed while maintaining a good level of accuracy. Thus, the results obtained in this study would also help in reducing the training and inference time for similar future studies on large datasets. MAPE for most functional groups was well below 10.0%, even for the functional groups with one or zero compounds in the training set. This shows that models were able to successfully learn hidden patterns and generalize quite well to the unseen data. High test errors for three functional groups (-CN, -NO₂, and -COOCH₃) were

observed due to the small number of data points in the region around their average redox potential. With the machine learning models developed in this study, it will be possible to explore the large molecular space and identify promising phenazine derivatives containing only one type of function group per molecule in a reasonable amount of time compared to experimental or DFT methods. Furthermore, these models will reduce the number of molecules that need to be analyzed using DFT calculations in a hybrid DFT-ML approaches. Thus, we have showed that machine learning based approaches could accelerate the discovery of novel materials for energy storage applications such as RFBs.

3.5 References

- (1) Shafiee, S.; Topal, E. When Will Fossil Fuel Reserves Be Diminished? *Energy Policy* **2009**, *37* (1), 181–189. <https://doi.org/10.1016/j.enpol.2008.08.016>.
- (2) Dehghani-Saniij, A. R.; Tharumalingam, E.; Dusseault, M. B.; Fraser, R. Study of Energy Storage Systems and Environmental Challenges of Batteries. *Renew. Sustain. Energy Rev.* **2019**, *104*, 192–208. <https://doi.org/10.1016/j.rser.2019.01.023>.
- (3) Höök, M.; Tang, X. Depletion of Fossil Fuels and Anthropogenic Climate Change-A Review. *Energy Policy* **2013**, *52*, 797–809. <https://doi.org/10.1016/j.enpol.2012.10.046>.
- (4) Gür, T. M. Review of Electrical Energy Storage Technologies, Materials and Systems: Challenges and Prospects for Large-Scale Grid Storage. *Energy Environ. Sci.* **2018**, *11* (10), 2696–2767. <https://doi.org/10.1039/c8ee01419a>.
- (5) Chu, W. S.; Chun, D. M.; Ahn, S. H. Research Advancement of Green Technologies. *Int. J. Precis. Eng. Manuf.* **2014**, *15* (6), 973–977. <https://doi.org/10.1007/s12541-014-0424-8>.
- (6) Balat, H. Green Power for a Sustainable Future. *Energy Explor. Exploit.* **2007**, *25* (1), 1–25. <https://doi.org/10.1260/014459807781036403>.
- (7) Demirbas, A. Electrical Power Production Facilities from Green Energy Sources. *Energy Sources, Part B Econ. Plan. Policy* **2006**, *1* (3), 291–301. <https://doi.org/10.1080/15567240500400648>.
- (8) Dunn, B.; Kamath, H.; Tarascon, J.-M. Electrical Energy Storage for the Grid: A Battery of Choices. *Science* (80-.). **2011**, *334* (6058), 928–935. <https://doi.org/10.1126/science.1212741>.
- (9) Chung, E. What Caused the Deadly Power Outages in Texas and How Canada’s Grid Compares. *CBC News*. 2021.
- (10) Larcher, D.; Tarascon, J. M. Towards Greener and More Sustainable Batteries for Electrical Energy Storage. *Nat. Chem.* **2015**, *7* (1), 19–29. <https://doi.org/10.1038/nchem.2085>.
- (11) Koochi-Fayegh, S.; Rosen, M. A. A Review of Energy Storage Types, Applications and Recent Developments. *J. Energy Storage* **2020**, *27*, 101047. <https://doi.org/10.1016/j.est.2019.101047>.
- (12) Deng, D. Li-ion Batteries: Basics, Progress, and Challenges. *Energy Sci. Eng.* **2015**, *3* (5), 385–418. <https://doi.org/10.1002/ese3.95>.
- (13) Skyllas-Kazacos, M.; Chakrabarti, M. H.; Hajimolana, S. A.; Mjalli, F. S.; Saleem, M. Progress in Flow Battery Research and Development. *J. Electrochem. Soc.* **2011**, *158* (8), R55. <https://doi.org/10.1149/1.3599565>.
- (14) Leung, P.; Li, X.; Ponce De León, C.; Berlouis, L.; Low, C. T. J.; Walsh, F. C. Progress in Redox Flow Batteries, Remaining Challenges and Their Applications in Energy Storage. *RSC Adv.* **2012**, *2* (27), 10125–10156. <https://doi.org/10.1039/c2ra21342g>.
- (15) Sánchez-Díez, E.; Ventosa, E.; Guarnieri, M.; Trovò, A.; Flox, C.; Marcilla, R.; Soavi,

- F.; Mazur, P.; Aranzabe, E.; Ferret, R. Redox Flow Batteries: Status and Perspective towards Sustainable Stationary Energy Storage. *J. Power Sources* **2021**, *481*, 228804. <https://doi.org/10.1016/j.jpowsour.2020.228804>.
- (16) Ha, S.; Gallagher, K. G. Estimating the System Price of Redox Flow Batteries for Grid Storage. *J. Power Sources* **2015**, *296*, 122–132. <https://doi.org/10.1016/j.jpowsour.2015.07.004>.
- (17) Whitehead, A. H.; Rabbow, T. J.; Trampert, M.; Pokorny, P. Critical Safety Features of the Vanadium Redox Flow Battery. *J. Power Sources* **2017**, *351*, 1–7. <https://doi.org/10.1016/j.jpowsour.2017.03.075>.
- (18) Chen, Y.; Kang, Y.; Zhao, Y.; Wang, L.; Liu, J.; Li, Y.; Liang, Z.; He, X.; Li, X.; Tavajohi, N.; Li, B. A Review of Lithium-Ion Battery Safety Concerns: The Issues, Strategies, and Testing Standards. *J. Energy Chem.* **2021**, *59*, 83–99. <https://doi.org/10.1016/J.JECHEM.2020.10.017>.
- (19) Díaz-Ramírez, M. C.; Ferreira, V. J.; García-Armingol, T.; López-Sabirón, A. M.; Ferreira, G. Environmental Assessment of Electrochemical Energy Storage Device Manufacturing to Identify Drivers for Attaining Goals of Sustainable Materials 4.0. *Sustain.* **2020**, *Vol. 12*, *Page 342* **2020**, *12* (1), 342. <https://doi.org/10.3390/SU12010342>.
- (20) da Silva Lima, L.; Quartier, M.; Buchmayr, A.; Sanjuan-Delmás, D.; Laget, H.; Corbisier, D.; Mertens, J.; Dewulf, J. Life Cycle Assessment of Lithium-Ion Batteries and Vanadium Redox Flow Batteries-Based Renewable Energy Storage Systems. *Sustain. Energy Technol. Assessments* **2021**, *46*, 101286. <https://doi.org/10.1016/j.seta.2021.101286>.
- (21) Kear, G.; Shah, A. A.; Walsh, F. C. Development of the All-Vanadium Redox Flow Battery for Energy Storage: A Review of Technological, Financial and Policy Aspects. *Int. J. Energy Res.* **2012**, *36* (11), 1105–1120. <https://doi.org/10.1002/er.1863>.
- (22) Alotto, P.; Guarnieri, M.; Moro, F. Redox Flow Batteries for the Storage of Renewable Energy: A Review. *Renew. Sustain. Energy Rev.* **2014**, *29*, 325–335. <https://doi.org/10.1016/j.rser.2013.08.001>.
- (23) Weber, A. Z.; Mench, M. M.; Meyers, J. P.; Ross, P. N.; Gostick, J. T.; Liu, Q. Redox Flow Batteries: A Review. *J. Appl. Electrochem.* **2011**, *41* (10), 1137–1164. <https://doi.org/10.1007/S10800-011-0348-2/FIGURES/15>.
- (24) Qi, Z.; Koenig, G. M. Review Article: Flow Battery Systems with Solid Electroactive Materials. *J. Vac. Sci. Technol. B, Nanotechnol. Microelectron. Mater. Process. Meas. Phenom.* **2017**, *35* (4), 040801. <https://doi.org/10.1116/1.4983210>.
- (25) Sánchez-Díez, E.; Ventosa, E.; Guarnieri, M.; Trovò, A.; Flox, C.; Marcilla, R.; Soavi, F.; Mazur, P.; Aranzabe, E.; Ferret, R. Redox Flow Batteries: Status and Perspective towards Sustainable Stationary Energy Storage. *J. Power Sources* **2021**, *481*, 228804. <https://doi.org/10.1016/j.jpowsour.2020.228804>.
- (26) Xi, J.; Xiao, S.; Yu, L.; Wu, L.; Liu, L.; Qiu, X. Broad Temperature Adaptability of Vanadium Redox Flow Battery—Part 2: Cell Research. *Electrochim. Acta* **2016**, *191*, 695–704. <https://doi.org/10.1016/J.ELECTACTA.2016.01.165>.

- (27) De La Cruz, C.; Molina, A.; Patil, N.; Ventosa, E.; Marcilla, R.; Mavrandonakis, A. New Insights into Phenazine-Based Organic Redox Flow Batteries by Using High-Throughput DFT Modelling. *Sustain. Energy Fuels* **2020**, *4* (11), 5513–5521. <https://doi.org/10.1039/d0se00687d>.
- (28) Gentil, S.; Reynard, D.; Girault, H. H. Aqueous Organic and Redox-Mediated Redox Flow Batteries: A Review. *Curr. Opin. Electrochem.* **2020**, *21*, 7–13. <https://doi.org/10.1016/j.coelec.2019.12.006>.
- (29) Leung, P.; Shah, A. A.; Sanz, L.; Flox, C.; Morante, J. R.; Xu, Q.; Mohamed, M. R.; Ponce de León, C.; Walsh, F. C. Recent Developments in Organic Redox Flow Batteries: A Critical Review. *J. Power Sources* **2017**, *360*, 243–283. <https://doi.org/10.1016/j.jpowsour.2017.05.057>.
- (30) Cao, J.; Tian, J.; Xu, J.; Wang, Y. Organic Flow Batteries: Recent Progress and Perspectives. *Energy and Fuels* **2020**, *34* (11), 13384–13411. <https://doi.org/10.1021/acs.energyfuels.0c02855>.
- (31) Li, M.; Rhodes, Z.; Cabrera-Pardo, J. R.; Minter, S. D. Recent Advancements in Rational Design of Non-Aqueous Organic Redox Flow Batteries. *Sustain. Energy Fuels* **2020**, *4* (9), 4370–4389. <https://doi.org/10.1039/d0se00800a>.
- (32) Elena I. Romadina, Denis S. Komarov, K. J. S.; A.Troshin, P.; Romadina, E. I.; Komarov, D. S.; Stevenson, K. J.; Troshin, P. A. New Phenazine Based Anolyte Material for High Voltage Organic Redox Flow Batteries. *Chem. Commun.* **2021**, *57* (24), 2986–2989. <https://doi.org/10.1039/D0CC07951K>.
- (33) Zhang, C.; Niu, Z.; Ding, Y.; Zhang, L.; Zhou, Y.; Guo, X.; Zhang, X.; Zhao, Y.; Yu, G. Highly Concentrated Phthalimide-Based Anolytes for Organic Redox Flow Batteries with Enhanced Reversibility. *Chem* **2018**, *4* (12), 2814–2825. <https://doi.org/10.1016/J.CHEMPR.2018.08.024/ATTACHMENT/5F347F22-8121-4A57-A985-7BA8FB5B2303/MMC1.PDF>.
- (34) Butler, K. T.; Davies, D. W.; Cartwright, H.; Isayev, O.; Walsh, A. Machine Learning for Molecular and Materials Science. *Nature* **2018**, *559* (7715), 547–555. <https://doi.org/10.1038/s41586-018-0337-2>.
- (35) Schmidt, J.; Marques, M. R. G.; Botti, S.; Marques, M. A. L. Recent Advances and Applications of Machine Learning in Solid-State Materials Science. *npj Comput. Mater.* **2019**, *5* (1). <https://doi.org/10.1038/s41524-019-0221-0>.
- (36) Wei, J.; Chu, X.; Sun, X.; Xu, K.; Deng, H.; Chen, J.; Wei, Z.; Lei, M. Machine Learning in Materials Science. *InfoMat* **2019**, *1* (3), 338–358. <https://doi.org/10.1002/inf2.12028>.
- (37) Pilia, G.; Wang, C.; Jiang, X.; Rajasekaran, S.; Ramprasad, R. Accelerating Materials Property Predictions Using Machine Learning. *Sci. Rep.* **2013**, *3*, 1–6. <https://doi.org/10.1038/srep02810>.
- (38) Batra, R. Accurate Machine Learning in Materials Science Facilitated by Using Diverse Data Sources. *Nature* **2021**, *589* (7843), 524–525. <https://doi.org/10.1038/d41586-020-03259-4>.
- (39) Gómez-Bombarelli, R.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Duvenaud, D.;

- Maclaurin, D.; Blood-Forsythe, M. A.; Chae, H. S.; Einzinger, M.; Ha, D. G.; Wu, T.; Markopoulos, G.; Jeon, S.; Kang, H.; Miyazaki, H.; Numata, M.; Kim, S.; Huang, W.; Hong, S. I.; Baldo, M.; Adams, R. P.; Aspuru-Guzik, A. Design of Efficient Molecular Organic Light-Emitting Diodes by a High-Throughput Virtual Screening and Experimental Approach. *Nat. Mater.* **2016**, *15* (10), 1120–1127. <https://doi.org/10.1038/nmat4717>.
- (40) Hautier, G.; Fischer, C. C.; Jain, A.; Mueller, T.; Ceder, G. Finding Natures Missing Ternary Oxide Compounds Using Machine Learning and Density Functional Theory. *Chem. Mater.* **2010**, *22* (12), 3762–3767. <https://doi.org/10.1021/cm100795d>.
- (41) Faber, F. A.; Lindmaa, A.; Von Lilienfeld, O. A.; Armiento, R. Machine Learning Energies of 2 Million Elpasolite (ABC2D6) Crystals. *Phys. Rev. Lett.* **2016**, *117* (13), 2–7. <https://doi.org/10.1103/PhysRevLett.117.135502>.
- (42) Carrasquilla, J.; Melko, R. G. Machine Learning Phases of Matter. *Nat. Phys.* **2017**, *13* (5), 431–434. <https://doi.org/10.1038/nphys4035>.
- (43) Cavasotto, C. N.; Di Filippo, J. I. Artificial Intelligence in the Early Stages of Drug Discovery. *Arch. Biochem. Biophys.* **2021**, *698*, 108730. <https://doi.org/10.1016/j.abb.2020.108730>.
- (44) Peyton, B. G.; Briggs, C.; D’Cunha, R.; Margraf, J. T.; Crawford, T. D. Machine-Learning Coupled Cluster Properties through a Density Tensor Representation. *J. Phys. Chem. A* **2020**, *124* (23), 4861–4871. <https://doi.org/10.1021/acs.jpca.0c02804>.
- (45) Seko, A.; Hayashi, H.; Nakayama, K.; Takahashi, A.; Tanaka, I. Representation of Compounds for Machine-Learning Prediction of Physical Properties. *Phys. Rev. B* **2017**, *95* (14), 1–11. <https://doi.org/10.1103/PhysRevB.95.144110>.
- (46) Sahoo, S.; Adhikari, C.; Kuanar, M.; Mishra, B. A Short Review of the Generation of Molecular Descriptors and Their Applications in Quantitative Structure Property/Activity Relationships. *Curr. Comput. Aided-Drug Des.* **2016**, *12* (3), 181–205. <https://doi.org/10.2174/1573409912666160525112114>.
- (47) Zeiri, Y.; Fisher, D.; Lukow, S. R.; Berezutskiy, G.; Gil, I.; Levy, T. Machine Learning Improves Trace Explosive Selectivity: Application to Nitrate-Based Explosives. *J. Phys. Chem. A* **2020**, *124* (46), 9656–9664. <https://doi.org/10.1021/acs.jpca.0c05909>.
- (48) Nayak, S.; Bhattacharjee, S.; Choi, J.-H.; Cheol Lee, S. Machine Learning and Scaling Laws for Prediction of Accurate Adsorption Energy. *J. Phys. Chem. A* **2019**, *124* (1), 247–254. <https://doi.org/10.1021/acs.jpca.9b07569>.
- (49) Wei, Y.; Chin, K.; M. Barge, L.; Perl, S.; Hermis, N.; Wei, T. Machine Learning Analysis of the Thermodynamic Responses of In Situ Dielectric Spectroscopy Data in Amino Acids and Inorganic Electrolytes. *J. Phys. Chem. B* **2020**, *124* (50), 11491–11500. <https://doi.org/10.1021/acs.jpcc.0c09266>.
- (50) L. Nisbet, M.; M. Pendleton, I.; M. Nolis, G.; J. Griffith, K.; Schrier, J.; Cabana, J.; J. Norquist, A.; R. Poeppelmeier, K. Machine-Learning-Assisted Synthesis of Polar Racemates. *J. Am. Chem. Soc.* **2020**, *142* (16), 7555–7566. <https://doi.org/10.1021/jacs.0c01239>.

- (51) Wexler, R. B.; Mark P. Martinez, J.; M. Rappe, A. Chemical Pressure-Driven Enhancement of the Hydrogen Evolving Activity of Ni₂P from Nonmetal Surface Doping Interpreted via Machine Learning. *J. Am. Chem. Soc.* **2018**, *140* (13), 4678–4683. <https://doi.org/10.1021/jacs.8b00947>.
- (52) Lee, M. H. Identification of Host-Guest Systems in Green TADF-Based OLEDs with Energy Level Matching Based on a Machine-Learning Study. *Phys. Chem. Chem. Phys.* **2020**, *22* (28), 16378–16386. <https://doi.org/10.1039/d0cp02871a>.
- (53) Landrum, G. RDKit: Open-source cheminformatics <https://www.rdkit.org/> (accessed Oct 23, 2021).
- (54) Ramsundar, B.; Eastman, P.; Walters, P.; Pande, V.; Leswing, K.; Wu, Z. *Deep Learning for the Life Sciences*; O'Reilly Media, 2019.
- (55) Chaube, S.; Goverapet Srinivasan, S.; Rai, B. Applied Machine Learning for Predicting the Lanthanide-Ligand Binding Affinities. *Sci. Rep.* **2020**, *10* (1), 1–11. <https://doi.org/10.1038/s41598-020-71255-9>.
- (56) A. Pugar, J.; M. Childs, C.; Huang, C.; W. Haider, K.; R. Washburn, N. Elucidating the Physicochemical Basis of the Glass Transition Temperature in Linear Polyurethane Elastomers with Machine Learning. *J. Phys. Chem. B* **2020**, *124* (43), 9722–9733. <https://doi.org/10.1021/acs.jpcc.0c06439>.
- (57) Casey, A. D.; Son, S. F.; Billionis, I.; Barnes, B. C. Prediction of Energetic Material Properties from Electronic Structure Using 3D Convolutional Neural Networks. *J. Chem. Inf. Model.* **2020**, *60* (10), 4457–4473. <https://doi.org/10.1021/acs.jcim.0c00259>.
- (58) J. Minnich, A.; McLoughlin, K.; Tse, M.; Deng, J.; Weber, A.; Murad, N.; D. Madej, B.; Ramsundar, B.; Rush, T.; Calad-Thomson, S.; Brase, J.; E. Allen, J. AMPL: A Data-Driven Modeling Pipeline for Drug Discovery. *J. Chem. Inf. Model.* **2020**, *60* (4), 1955–1968. <https://doi.org/10.1021/acs.jcim.9b01053>.
- (59) Okamoto, Y.; Kubo, Y. Ab Initio Calculations of the Redox Potentials of Additives for Lithium-Ion Batteries and Their Prediction through Machine Learning. *ACS Omega* **2018**, *3* (7), 7868–7874. <https://doi.org/10.1021/acsomega.8b00576>.
- (60) Joshi, R. P.; Eickholt, J.; Li, L.; Fornari, M.; Barone, V.; Peralta, J. E. Machine Learning the Voltage of Electrode Materials in Metal-Ion Batteries. *ACS Appl. Mater. Interfaces* **2019**, *11* (20), 18494–18503. <https://doi.org/10.1021/acsami.9b04933>.
- (61) Allam, O.; Cho, B. W.; Kim, K. C.; Jang, S. S. Application of DFT-Based Machine Learning for Developing Molecular Electrode Materials in Li-Ion Batteries. *RSC Adv.* **2018**, *8* (69), 39414–39420. <https://doi.org/10.1039/c8ra07112h>.
- (62) Zhang, Y.; Xu, X. Machine Learning Properties of Electrolyte Additives: A Focus on Redox Potentials. *Ind. Eng. Chem. Res.* **2021**, *60* (1), 343–354. <https://doi.org/10.1021/acs.iecr.0c05055>.
- (63) Allam, O.; Kuramshin, R.; Stoichev, Z.; Cho, B. W.; Lee, S. W.; Jang, S. S. Molecular Structure–Redox Potential Relationship for Organic Electrode Materials: Density Functional Theory–Machine Learning Approach. *Mater. Today Energy* **2020**, *17*, 100482. <https://doi.org/10.1016/j.mtener.2020.100482>.

- (64) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, É. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12* (85), 2825–2830.
- (65) Wildman, S. A.; Crippen, G. M. Prediction of Physicochemical Parameters by Atomic Contributions. *J. Chem. Inf. Comput. Sci.* **1999**, *39* (5), 868–873. <https://doi.org/10.1021/ci9903071>.
- (66) Landrum, G. Getting Started with the RDKit in Python — The RDKit 2020.03.1 documentation <https://www.rdkit.org/docs/GettingStartedInPython.html#list-of-available-descriptors> (accessed Mar 31, 2021).
- (67) Hall, L. H.; Mohny, B.; Kier, L. B. The Electrotopological State: Structure Information at the Atomic Level for Molecular Graphs. *J. Chem. Inf. Comput. Sci.* **1991**, *31* (1), 76–82. <https://doi.org/10.1021/ci00001a012>.
- (68) Hall, L. H.; Kier, L. B. The Molecular Connectivity Chi Indexes and Kappa Shape Indexes in Structure-Property Modeling; John Wiley & Sons, Ltd, 2007; pp 367–422. <https://doi.org/10.1002/9780470125793.ch9>.
- (69) Todeschini, R.; Consonni, V. Descriptors from Molecular Geometry. In *Handbook of Chemoinformatics*; Wiley-VCH Verlag GmbH: Weinheim, Germany, 2008; Vol. 3, pp 1004–1033. <https://doi.org/10.1002/9783527618279.ch37>.
- (70) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50* (5), 742–754. <https://doi.org/10.1021/ci100050t>.
- (71) Durant, J. L.; Leland, B. A.; Henry, D. R.; Nourse, J. G. Reoptimization of MDL Keys for Use in Drug Discovery. *J. Chem. Inf. Comput. Sci.* **2002**, *42* (6), 1273–1280. <https://doi.org/10.1021/ci010132r>.

Chapter 4

Predicting the Redox Potentials of Phenazine Derivatives Using a Hybrid DFT-ML Approach

Chapter 4

Predicting the Redox Potentials of Phenazine Derivatives Using a Hybrid DFT-ML Approach

Abstract

This study investigates four machine learning models to predict the redox potential of phenazine derivatives in dimethoxyethane (DME) using density functional theory (DFT). A small dataset of 151 phenazine derivatives having only one type of functional group per molecule (20 unique groups) was used for the training. Prediction accuracy was improved by a combined strategy of feature selection and hyperparameter optimization using an external validation-set. The models were evaluated on the external test-set containing new functional groups and diverse molecular structures. High prediction accuracies of $R^2 > 0.74$ were obtained on the external test-set. Despite being trained on molecules with a single type of functional group, the models were able to predict the redox potentials of derivatives containing multiple and different types of functional groups with good accuracies ($R^2 > 0.7$). This type of performance for predicting redox potential from such a small and simple dataset of phenazine derivatives has never been reported before. Redox flow batteries (RFBs) are emerging as promising candidates for energy storage systems. However, new green and efficient materials are required for their widespread usage. We believe that the hybrid DFT-ML approach demonstrated in this work will help in accelerating the virtual screening of phenazine derivatives, thereby saving computational and experimental costs. Using this approach, we have identified promising phenazine derivatives for green energy storage systems such as RFB.

4.1 Introduction

This study continues our investigation into battery materials based on phenazine molecules. In the previous study, we developed several machine models to explore the molecular space of phenazine derivatives. Although the models achieved high prediction accuracy on the test-set (the internal test-set in this study), their performance was not assessed on diverse phenazine molecules obtained from different sources. Such a restricted analysis does not give insight into the generalizability of the machine learning algorithms. Therefore, in this study, we assessed model performance on the diverse phenazine derivatives obtained from different sources. The previous study showed that 2D molecular features are most accurate at predicting the redox potential of phenazine derivatives in DME (dimethoxyethane) solvent. Therefore, we used only 2D molecular features in this study. We also restricted ourselves to the four machine learning models suitable for the small datasets. The training-set containing 151 phenazine derivatives was obtained from the same DFT study (used in the previous chapter) of 189 phenazine derivatives with only one type of functional group per molecule (20 unique functional groups).¹ 2D molecular features were computed from the optimized neutral structures using the RDKit Python library.² We observed that models trained on all 208 features overfit the training data and show excellent performance on the internal test-set, whereas performance drops significantly on the external test-set containing structurally diverse functional groups (Figure 4.1).

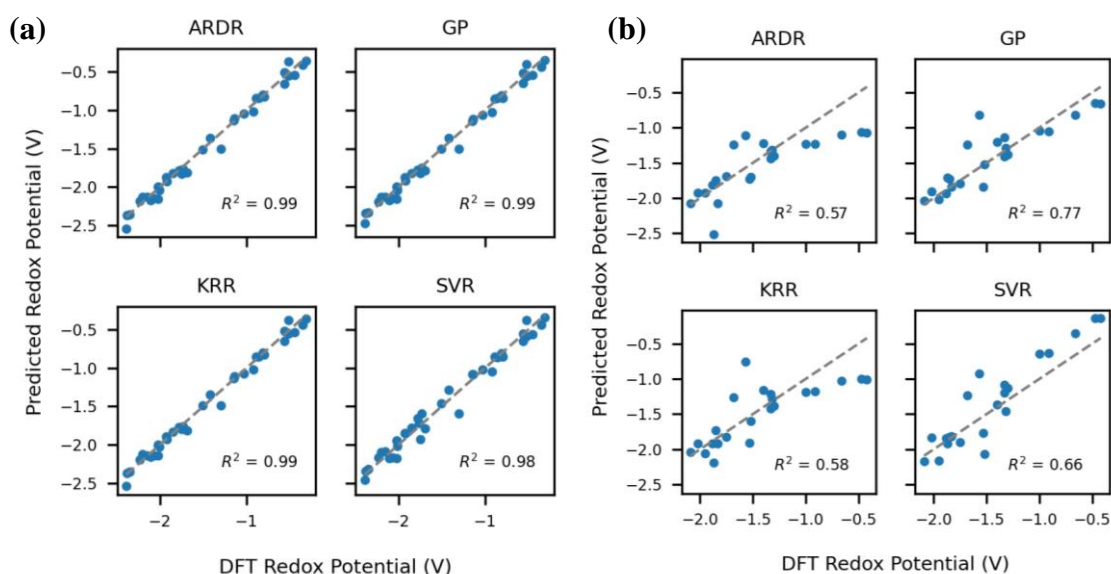


Figure 4.1. Plots showing machine learning predictions. ML predictions (y-axis) vs. DFT redox potentials (x-axis) on (a) internal test-set, (b) external test-set. The gray dashed line corresponds to the perfect predictions.

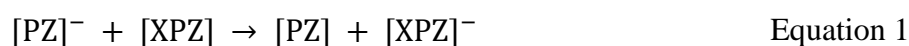
As the internal test-set comes from nearly the same distribution as the training-set, ML models showed high accuracy. In this work, we addressed overfitting through feature selection and hyperparameter optimization using an external validation-set. Then, model performance was again assessed on the external test-set to confirm the reduction in overfitting. ML models were further validated by generating test-sets containing two and three different types of functional groups per molecule (called multiple functional group test-sets). The redox potentials of the

molecules present in the external and multiple functional group test-sets were computed using DFT. Next, we carried out a feature importance analysis to understand the most influential 2D molecular features. After that, we analyzed the structure–functional relationship between phenazine derivatives and their redox potential. Finally, a few promising candidates were identified for the anolyte of RFBs from the external test-set.

4.2 Materials and Methods

4.2.1 Computational Details

The Redox potential of phenazine derivatives was computed using the DFT workflow described in the paper by Mavrandonakis *et al.*¹ All DFT calculations were performed with the Gaussian 09 software.³ Geometry optimization of neutral and reduced forms of phenazine and its derivatives were carried out in the gas phase by employing B3LYP/6-31+G(d,p) level of theory.⁴⁻⁷ Harmonic frequency analysis was performed for all the structures to confirm them as minima. Solvation effects of DME were incorporated during the single point calculations using the M06-2X functional⁸, by employing the SMD solvation model.^{9,10} The term ‘Redox Potential’ in this chapter corresponds to the ‘Reduction Potential’ with respect to unsubstituted phenazine molecule (i.e., the parent phenazine). The redox potentials of phenazine derivatives were computed using the following equations:



$$E_1^0 = -\frac{\Delta G_{(rxn,sol)}}{nF} + E_{1(ref)}^0 \quad \text{Equation 2}$$

$$\Delta G_{(rxn,sol)} = G_{([\text{XPZ}]^-,sol)}^0 + G_{([\text{PZ}],sol)}^0 - G_{([\text{XPZ}],sol)}^0 - G_{([\text{PZ}]^-,sol)}^0 \quad \text{Equation 3}$$

$$G_{(sol)}^0 = G_{(therm,gas)}^{(B3LYP)} + E_{(sol)}^{M06-2X} \quad \text{Equation 4}$$

where PZ symbolizes the parent phenazine, XPZ represents the substituted phenazine molecules, $E_{1(ref)}^0$ is the reported redox potential of parent phenazine PZ¹, $\Delta G_{(rxn,sol)}$ corresponds to the free energy change of the reaction, F is the Faraday constant, n is number of electron involved in the reduction, and $G_{(sol)}^0$ represents the final composite free energy of individual species, which was calculated by adding the free energy contribution computed at the B3LYP level of theory: $G_{(therm,gas)}^{(B3LYP)}$, to the single point energies calculated at M06-2X level of theory: $E_{(sol)}^{M06-2X}$.

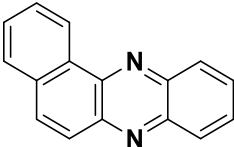
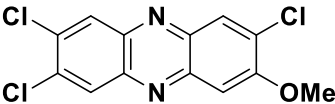
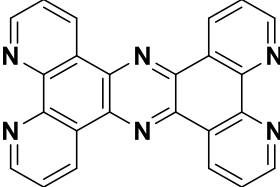
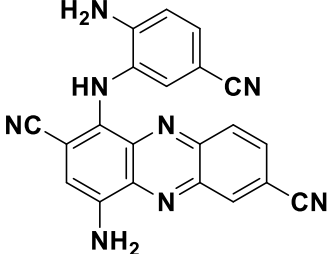
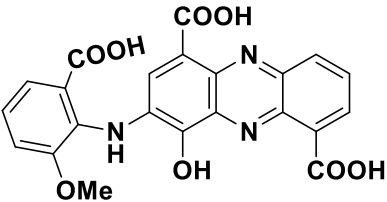
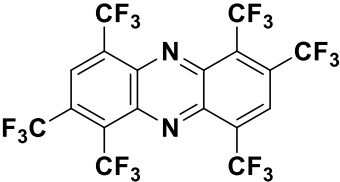
4.2.2 Data Generation

Training-Set and Internal Test-Set: These datasets are similar to those investigated in the previous chapter (chapter no. 3) — obtained from work reported by Mavrandonakis and co-workers.¹ In their report, the redox potentials of 189 phenazine derivatives were computed using DFT in DME (dimethoxyethane) solvent. These DFT redox potentials were used as a target property in this work during training and testing. Twenty unique electron-withdrawing and electron-donating functional groups were present in the dataset (–N(CH₃)₂, –NH₂, –OH, –OCH₃, –P(CH₃)₂, –SCH₃, –SH, –CH₃, –C₆H₅, –CH=CH₂, –F, –Cl, –CHO, –COCH₃, –CONH₂, –COOCH₃, –COOH, –CF₃, –CN and –NO₂). It should be noted that phenazine derivatives in this dataset contain only one type of functional group per molecule. Optimized 3D structures of derivatives in neutral and in anionic states were also provided. However, only neutral

structures were used in this study. Unfortunately, not all compounds were supplied with their neutral structure, those compounds were modeled, and their optimized structures were added to the dataset. Next, 208 different types of features were generated using RDKit Python library.² The features were scaled using the ‘*StandardScaler*’ class of the scikit-learn library,¹¹ removing the mean and scaling each feature to unit variance. Finally, the whole dataset was shuffled and split randomly into training-set and internal test-set in an 8:2 ratio (151 samples in the training-set and 38 samples in the internal test-set).

External Test-Set: This dataset was compiled from different reports studying various properties of phenazine derivatives.^{12–16} Their redox potentials were computed using DFT and used as a target property during testing. We gathered a total of 30 phenazine derivatives. Derivatives containing five or more substituted rings were removed. Also, derivatives having drastically different neutral and anion structures were removed. In the end, 22 diverse phenazine derivatives with multiple types of functional groups remained in the external test-set. Table 4.1 shows some of the structures from this dataset. It can be seen that this dataset contains unique and different structures from the training-set.

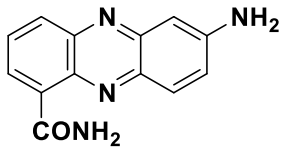
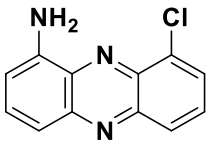
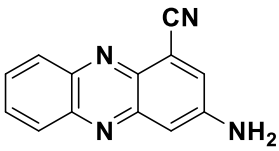
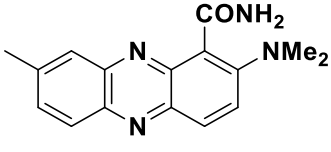
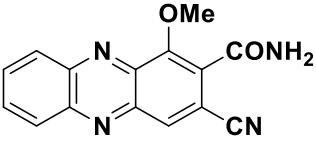
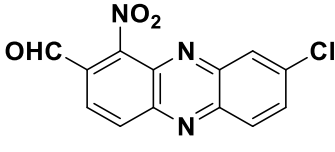
Table 4.1. Representative structures from external test-set. Mol IDs were assigned to identify derivatives from the corresponding dataset.

 <p>Mol ID: 1</p>	 <p>Mol ID: 3</p>	 <p>Mol ID: 5</p>
 <p>Mol ID: 15</p>	 <p>Mol ID: 17</p>	 <p>Mol ID: 28</p>

Multiple Functional Group Test-Sets: This dataset contains two test-sets: (i) Two functional group test-set, (ii) Three functional group test-set. These test-sets were generated by randomly choosing the position and the type of functional group from this list: (–N(CH₃)₂, –NH₂, –OH, –OCH₃, –P(CH₃)₂, –SH, –CH₃, –C₆H₅, –CH=CH₂, –F, –Cl, –CHO, –COCH₃, –CONH₂, –COOCH₃, –COOH, –CF₃, –CN and –NO₂). Twenty derivatives having two different types of functional groups per molecule were generated for two functional group test-set. Similarly, twenty derivatives having three different types of functional groups per molecule were generated for three functional group test-set. Their redox potentials were computed using DFT and used as a target property during testing. Five derivatives from two and three functional group test-sets were removed to form an external validation-set. Thus, the final size of two and three functional group test-sets was reduced from twenty to fifteen. In this report, the term

‘multiple’ refers to the derivatives containing different types and more than one functional group. Similarly, the terms ‘two functional groups’ and ‘three functional groups’ refer to the derivatives containing two different types of functional groups and three different types of functional groups per molecule, respectively. A few representative structures from these test-sets are shown in Table 4.2.

Table 4.2. Representative structures from multiple functional group test-sets. Mol IDs were assigned to identify derivatives from the corresponding dataset.

 <p>Mol ID: 19</p>	 <p>Mol ID: 9</p>	 <p>Mol ID: 7</p>
 <p>Mol ID: 12</p>	 <p>Mol ID: 14</p>	 <p>Mol ID: 5</p>

External Validation-Set: An external validation-set of ten phenazine derivatives was compiled from two and three functional group test-sets. Five derivatives from two functional group test-set and five from three functional group test-set were selected. Their redox potentials were computed using DFT and used as a target property. This validation-set does not come from the training-set. Therefore, it is termed as external validation-set. It was used for feature selection and hyperparameter optimization. The external validation-set improves generalization by transferring knowledge from the test-set to models through hyperparameters.

4.2.3 Hyperparameter Optimization

Hyperparameters of the models were optimized using the ‘*GridSearchCV*’ class of the scikit-learn library¹¹. During hyperparameter optimization, models were trained on the training-set and evaluated on the external validation-set. Mean squared error (MSE) was used as an evaluation metric for hyperparameter optimization. The grid of hyperparameters for each model is given in Table 4.3. The parameter grid was adjusted manually.

Table 4.3. Parameter grids used during hyperparameter optimization.

Model Name	Parameter grid
ARDR	alpha_1: [1e-7,1e-8,1e-9] alpha_2: [1.5,1,1e-2] lambda_1: [1e-9,1e-10,1e-11] lambda_2: [1e-3,1e-4,1e-5]
GP	kernel_list=[200*RBF(length_scale=1,length_scale_bounds=(0, 10000)) +WhiteKernel(noise_level=n,noise_level_bounds=(1e-2, 1e+1)) for l in np.linspace(0,1000,10) for n in np.linspace(0.1,1.5,10)] kernel: [RBF() + WhiteKernel(), RBF(length_scale=200.0, length_scale_bounds=(1, 10000)) + WhiteKernel(noise_level=0.1, noise_level_bounds=(1e-2, 1e+1))] + kernel_list alpha: [1e-10,1e-11,1e-12,0]
KRR	alpha: [1e-7,1e-6,1e-5] kernel: ['chi2', 'linear', 'rbf', 'laplacian', 'sigmoid', 'cosine'] gamma: [None,1e-6,1e-7,1e-8]
SVR	kernel: ['linear', 'poly', 'rbf', 'sigmoid'] C: [0.025]

4.2.4 Machine Learning Models

Following four machine-learning models were investigated in this study. These models were chosen due to their ability to generalize from small datasets. Models were implemented with the scikit-learn Python library¹¹. First, models were trained on the training-set containing all 208 features, followed by hyperparameter optimization. Then, the models were re-trained on different subsets of features to identify the number of features having the highest average performance on the external validation-set. Once the optimum number of features were identified, hyperparameter optimization was performed with the selected features to improve model performance further.

Automatic Relevance Determination Regression (ARDR): This is the probabilistic model related to the sparse Bayesian learning (SBL) framework. It assumes axis-parallel, elliptical Gaussian distribution for each coefficient. The precision of each Gaussian distribution is drawn from the prior distribution (gamma distribution); therefore, it can lead to sparser coefficients. Thus, it is an effective tool for removing irrelevant features.^{17,18}

Gaussian Process Regression (GP): It is the nonparametric Bayesian model. The nonparametric Bayesian model provides the probability distribution of parameters over all possible functions that fit the data. The prior in a Gaussian process is specified on function space. Gaussian process prior is a multivariate normal distribution whose mean is obtained from the data, and covariance is specified using the kernel function. The hyperparameters of the kernel are optimized during the training.^{19,20} We used a combination of *WhiteKernel* and

RBF kernel. *WhiteKernel* is used for specifying noise level and *RBF* kernel is a very popular kernel used in many algorithms.

Kernel ridge regression (KRR): It is the extension of ridge regression with kernel trick. In ridge regression, a linear model is leaned with the L2-norm regularization. Using the kernel trick, KRR learns a linear function in the high dimensional non-linear space without actually transforming the data.²¹

Support Vector Regression (SVR): This model is the regression form of support vector machine (SVM), a popular algorithm for classification tasks. Analogous to SVM, SVR depends on the subset of training data and ignores the points whose prediction is close to their true value. SVM also utilizes kernel trick and learns a hyperplane in the high dimensional space.²²

4.2.5 Evaluation Metrics

The following metrics were used in this study to evaluate the model performance. In the formulas below, N denotes the number of data points, \hat{y}_i denotes the predicted value of i -th sample and the y_i denotes the corresponding true value.

Coefficient of Determination (R^2):

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2}$$

$$\text{where, } \bar{y} = \frac{\sum_{i=1}^N y_i}{N}$$

Mean Squared Error (MSE):

$$MSE = \frac{\sum_{i=1}^N (y_i - \hat{y})^2}{N}$$

Mean Absolute Error (MAE):

$$MAE = \frac{\sum_{i=1}^N |y_i - \hat{y}_i|}{N}$$

The use of terms ‘Accuracy’ and ‘Performance’ in this chapter is contextual and refers to one or more metrics defined above.

4.2.6 Feature Selection

As the number of features obtained from the RDKit library was more than the size of the training-set, it was necessary to implement a feature selection strategy. It has been observed that the training-set containing more features than data points leads to overfitting.²³ Feature selection was implemented using the ‘*SelectKBest*’ class of the scikit-learn library.²⁴ The parameter ‘ k ’ of ‘*SelectKBest*’ class was obtained by evaluating the average performance of models on the external validation-set at different values of ‘ k ’. First, models were trained on the training-set containing all features, followed by hyperparameter optimization. Then, the models were re-trained on the subsets of features selected using ‘*SelectKBest*’ class

corresponding to different values of ‘ k ’. The following values for ‘ k ’ were tested: 50, 75, 100, 125, 150, 208. The average model performance at different values of ‘ k ’ on the external validation-set is shown in Table 4.4. It can be seen that the models trained on 100 selected features show the highest average performance in terms of R^2 . Therefore, these 100 features were selected for the subsequent analysis. The models trained on 100 selected features were further improved through hyperparameter optimization.

Table 4.4. Average model performance on external validation-set at different values of ‘ k ’.

Performance Metric	Values of ‘ k ’					
	50	75	100	125	150	208
R^2	0.45	0.42	0.57	0.55	0.54	0.54
MSE	0.02	0.02	0.02	0.02	0.02	0.02
MAE	0.12	0.12	0.10	0.10	0.10	0.10

4.2.7 Feature Importance Analysis

The feature importance analysis was performed using the technique known as Permutation Importance. In this technique, the values of the feature to be assessed are randomly shuffled (permuted). Then, prediction accuracy is computed on the shuffled dataset. Shuffling of feature values is equivalent to replacing the feature with noise, thereby removing its information from the dataset. Therefore, the model is expected to perform poorly on the shuffled dataset if the feature were important. The degree of importance depends on the amount of variation in the accuracy. This technique does not re-train the model; therefore, a trained model is required. The permutation importance was computed using ‘*permutation_importance*’ class of the scikit-learn library and the training-set ²⁵. This procedure was repeated 100 times to obtain reliable estimates. The feature importance scores were rescaled between 0 to 1. The mean and standard deviation of the feature scores were reported. The mean feature score was used for the ranking of individual features. The terms ‘Feature’ and ‘Descriptor’ are used interchangeably in this chapter.

4.3 Results and Discussion

4.3.1 Test-set Performance

We assessed the generalizability of the trained models (i.e., performance on the unseen data) using internal and external test-sets. Please refer to section 4.2 for the preparation of internal and external test-sets. As the internal test-set comes from the same source, it is very similar to the training-set and contains derivatives with only one type of functional group per molecule. Whereas the external test-set is compiled from multiple sources, it has very diverse phenazine derivatives with different types of functional groups. It also contains functional groups and structures not present in the training-set (e.g., -NHPH, -Br, extended conjugation). Figure 4.2 shows the performance on the internal test-set, and Figure 4.3 shows the performance on the external test-set. It can be seen that all models have excellent accuracy on the internal test-set ($R^2 > 0.98$) and high accuracy on the external test-set ($R^2 > 0.74$). GP model achieved the highest R^2 of 0.89 on the external test set. After deep analysis in section 4.3.3, it was revealed that GP is not a stable model while relatively low performing models KRR ($R^2 = 0.83$) and SVR ($R^2 = 0.85$) are more stable. Therefore, one should be careful while using the high-performing model, and the stability of the model should also be considered. The values of performance metrics on internal and external tests are shown in Table 4.5. Such a performance on the external test-set is surprising as models were trained on the phenazine derivatives having only one type of functional group. These results show that machine learning models are capable of generalizing from a very small and simple dataset.

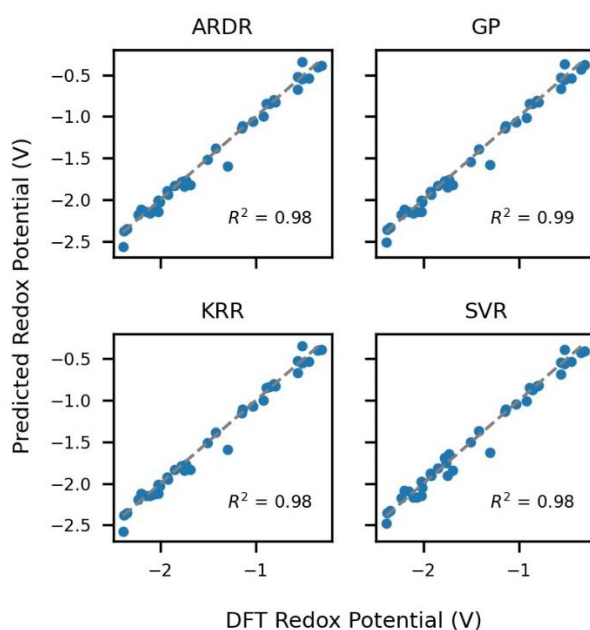


Figure 4.2. Plots showing machine learning predictions on internal test-set (y-axis) vs. DFT redox potentials (x-axis). Gray dash line corresponds to the perfect predictions.

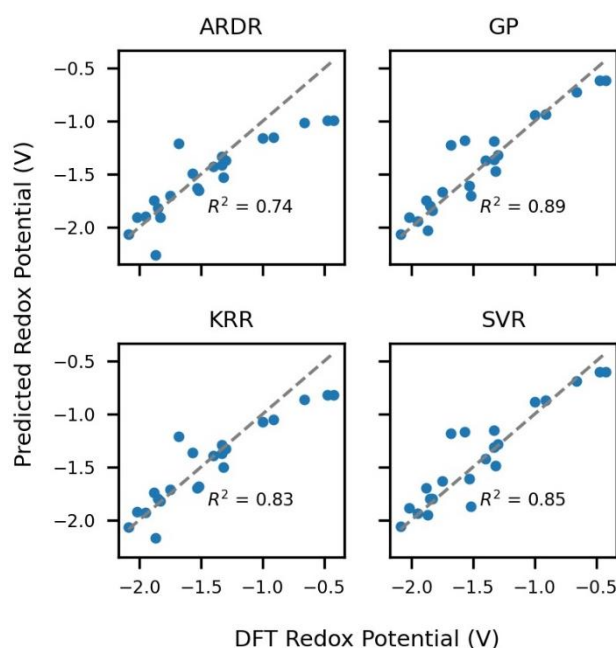


Figure 4.3. Plots showing machine learning predictions on external test-set (y-axis) vs. DFT redox potentials (x-axis). Gray dash line corresponds to the perfect predictions.

Table 4.5. Values of performance metrics on internal and external test-sets. Numbers were rounded upto two decimals

Model name	Internal test-set			External test-set		
	R^2	MSE	MAE	R^2	MSE	MAE
ARDR	0.98	0.01	0.06	0.74	0.06	0.18
GP	0.99	0.01	0.05	0.89	0.03	0.11
KRR	0.98	0.01	0.05	0.83	0.04	0.14
SVR	0.98	0.01	0.07	0.85	0.03	0.13

4.3.2 Prediction on Multiple Functional Group Test-sets

Next, we assessed the model performance on the phenazine derivatives substituted with different types of functional groups per molecule. These test-sets were generated randomly; please refer to section 4.2 for the generation of this dataset. Figure 4.4 and Figure 4.5 show the performance on the derivatives containing two and three different functional groups, respectively. It can be seen that the models performed reasonably well ($R^2 > 0.7$) even though molecules used for the training had only one type of functional group per molecule. In particular, GP models achieved the highest performance of $R^2 = 0.82$ on two functional groups test-set. Whereas ARDR achieved the highest performance of $R^2 = 0.82$ on three functional groups test-set. A deeper analysis of GP and ARDR in section 4.3.3 suggests that GP and ARDR are not very reliable models. Although KRR and SVR have relatively low performance, they are more reliable. Therefore, one should be careful while using high-performing models, and the model's reliability and stability should also be considered. Nevertheless, these results again show the surprising generalization power of machine learning models.

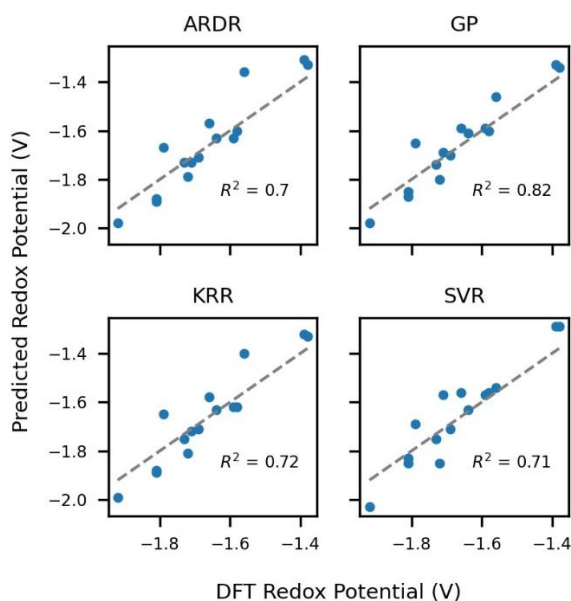


Figure 4.4. Plots showing machine learning predictions on two functional group test-set (y-axis) vs. DFT redox potentials (x-axis). Gray dash line corresponds to the perfect predictions.

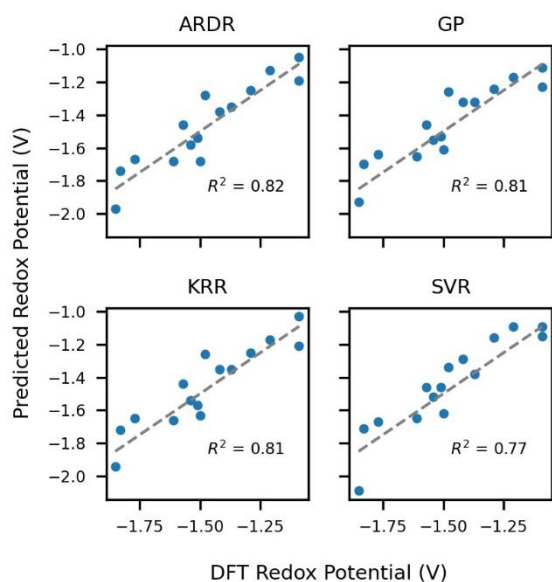


Figure 4.5. Plots showing machine learning predictions on three functional group test-set (y-axis) vs. DFT redox potentials (x-axis). Gray dash line corresponds to the perfect predictions.

Furthermore, we added these randomly generated fifteen derivatives from two functional group test-set to the training-set and re-trained the models on this new dataset of 166 derivatives. The predictive performance of this combined dataset was assessed on the same dataset of fifteen derivatives containing three functional group test-set. The results of this analysis are shown in Figure 4.6. It can be seen that the model performance has improved with the addition of more data in the training-set.

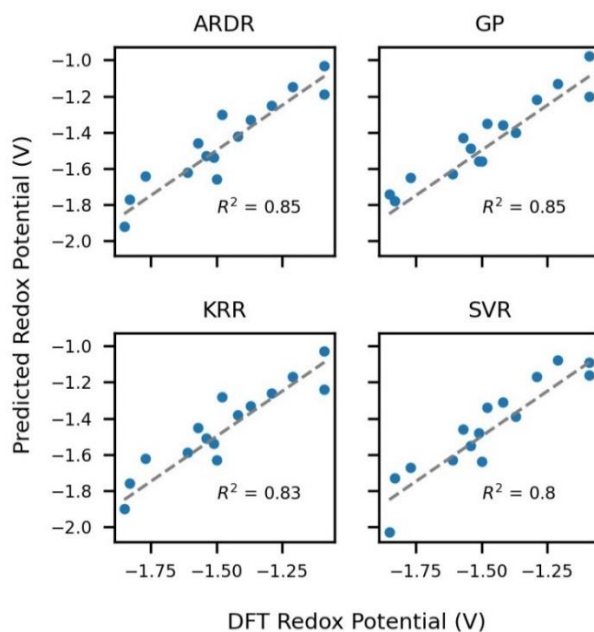


Figure 4.6. Plots showing machine learning predictions on three functional group test-set (y-axis) vs. DFT redox potentials (x-axis). The combined dataset (training-set + two functional group test-set) was used for the training. Gray dash line corresponds to the perfect predictions.

4.3.3 Feature Importance Analysis

We carried out feature importance analysis using Permutation Importance. Please refer to section 4.2 for the details on the technique. In order to understand how model performance changes with the number of descriptors, we re-trained the models on the subset of features and assessed their performance on the internal test-set. Top 50 features based on their permutation importance score were used. R^2 was used as a performance metric. The result of this analysis is shown in Figure 4.7. It can be seen that most of the models show a jump in the R^2 and have $R^2 > 0.9$ around the top ten features. The unusual behavior of GP model is attributed to the instability of the model for a small number of features. The plots in Figure 4.8 show the histograms of the top ten important features from each model. Although models show variation in feature importance, they all agree in terms of the most important feature, i.e., 'PEOE_VSA1'. Interestingly, most of the features in ARDR have small weights as ARDR tries to prune the large number of irrelevant features leading to a sparse model.^{18,26} Five out of ten features - 'MaxAbsPartialCharge', 'PEOE_VSA1', 'fr_ArN', 'fr_NH0', 'fr_NH2' are common to all models. Other variations in the feature importance scores could be attributed to the difference in the internal structure of the models. Here, we discuss some of the common features from Figure 4.8.

PEOE_VSA1: This is the sum of the approximate accessible van der Waals surface area (i.e., VSA in \AA^2) of the atoms having partial charge less than -0.30.²⁷⁻²⁹ The partial charges are computed using the PEOE method developed by Gasteiger and Marsili in 1980. Please refer to the discussion of *MaxAbsPartialCharge* for the PEOE method. Thus, this descriptor captures the information related to molecular size and the number of electron-donating functional groups

MaxAbsPartialCharge: This is the maximum value of the absolute Gasteiger partial charges present in the molecule. In 1980, Gasteiger and Marsili gave the procedure to calculate the partial charges in a molecule. That procedure is known as Partial Equalization of Orbital Electronegativities (PEOE). In this method, the charge is transferred between bonded atoms until equilibrium. The Gasteiger partial charges depend on the connectivity and the orbital electronegativity, thus capturing the electron-donating and withdrawing power of the atoms.³⁰ Electronegativity is essential information as electron-donating groups decrease the redox potential and electron-withdrawing groups increase the redox potential.¹

MinPartialCharge: This is the minimum value of the Gasteiger partial charges present in the molecule. Please refer to the discussion of *MaxAbsPartialCharge* for the properties of Gasteiger partial charges.

fr_NH0: It is the number of tertiary amines present in the molecule.

fr_ArN: It is the number of N functional groups attached to aromatic rings.

fr_NH2: It is the number of primary amines.

NHOHCount: It is the number of N-H and O-H bonds present in the molecule.

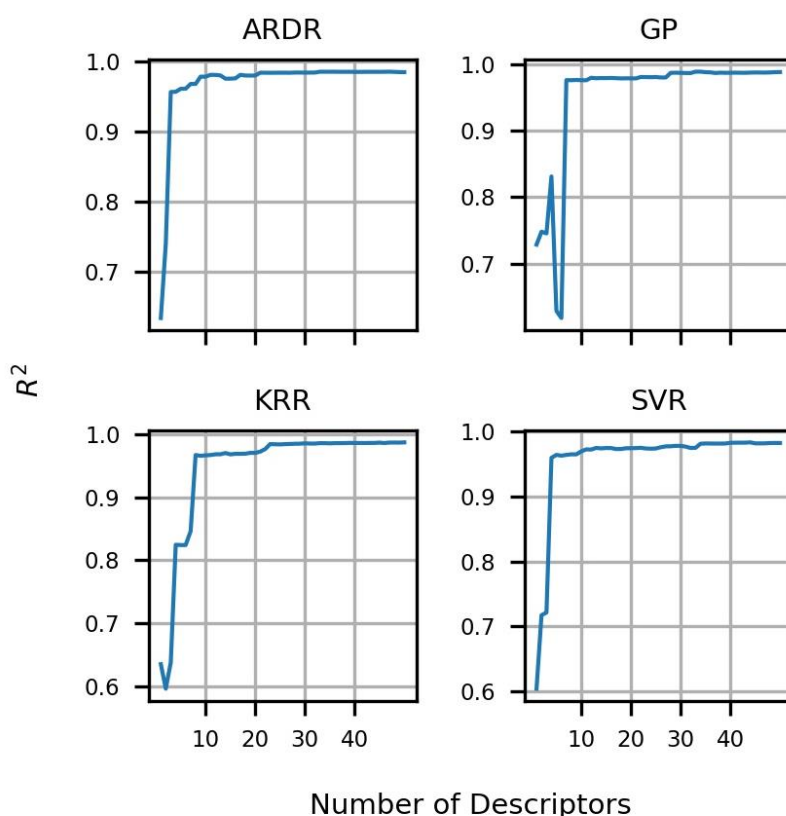


Figure 4.7. R^2 vs. number of descriptors. R^2 was computed using the internal test-set. In this study, we identified a few issues with ARDR and GP. Despite the high predictive performance, ARDR is not a reliable model as it places very high weight on one feature (i.e., *PEOE_VSA1*). Similarly, GP is not a reliable model as it becomes unstable when a small number of features are used. We encountered divided by zero errors in the kernel function during the analysis with GP model.

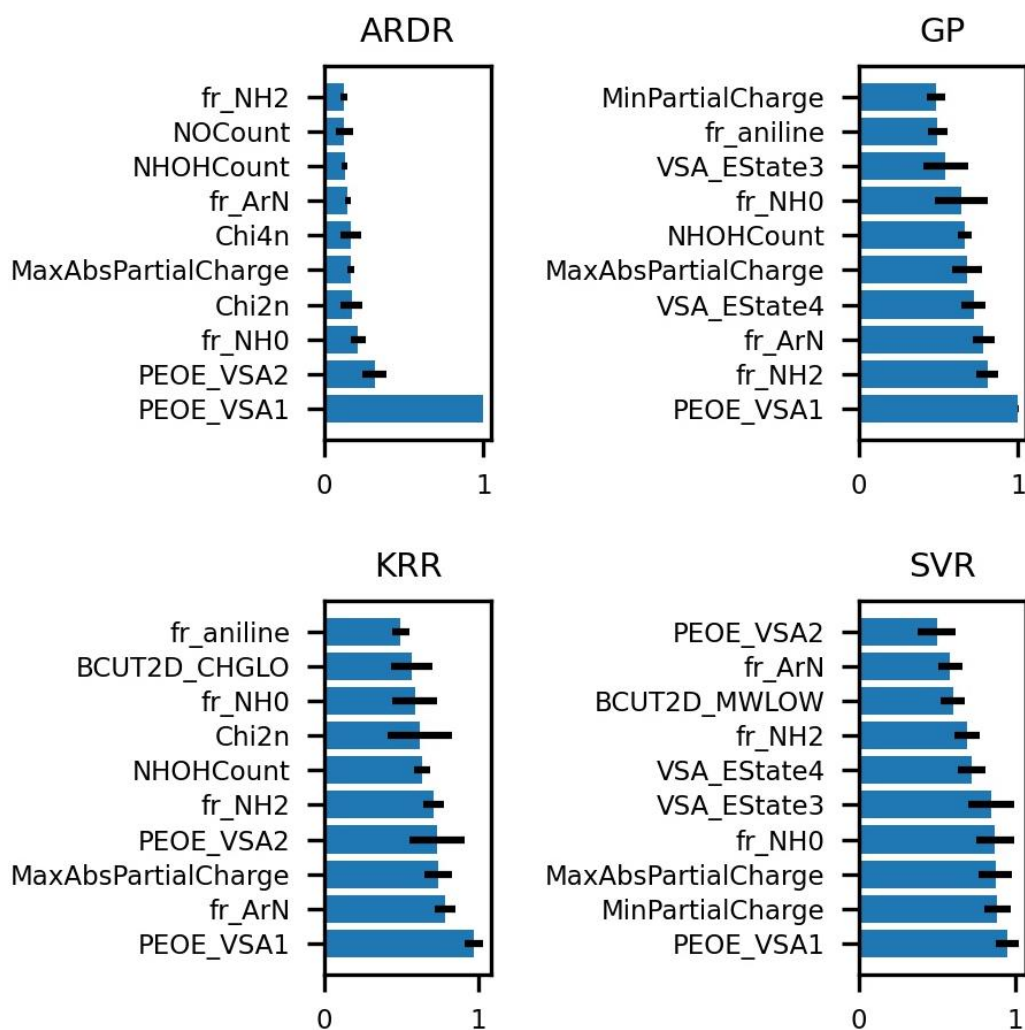


Figure 4.8. Top ten features (y-axis) vs. mean feature importance score (x-axis). Feature importance scores were rescaled between 0 to 1. Error bars represent the standard deviation obtained from 100 repetitions.

From the analysis in this section, we realized that there are some issues with the ARDR and GP models, which are outlined below. One should be very careful while using ARDR and GP models.

Issues with the ARDR model: As ARDR is related to the sparse Bayesian learning (SBL) framework, it reduces the number of irrelevant features. Unfortunately, in this case, ARDR has put a lot of weight on only one feature, i.e., ‘*PEOE_VSA1*’ (Figure 4.8). Surprisingly, ARDR also archives an accuracy of more than 0.95 R^2 only with the two important features (Figure 4.7). Although it has shown good performance on the dataset investigated in this work, it may not work for the broad molecular space. This type of behavior reduces the reliability of the model.

Issues with the GP model: From Figure 4.7, it can be seen that the model’s accuracy decreases with more features, and at around ten features, there is a significant drop in the performance.

We also encountered divided by zero errors in the kernel function during this analysis. This shows that GP may not be a very stable model in this case.

4.3.4 Structure–Functional Relationship

'*PEOE_VSAI*' is the most important descriptor common to all models. It is computed by summing over the approximate accessible van der Waals surface area (i.e., VSA in Å²) of the atoms having partial charge less than -0.30.^{27–29} Thus, the '*PEOE_VSAI*' descriptor captures the information related to molecular size and the number of electron-donating functional groups present in the molecule. From Figure 4.9, we can see that the redox potential of phenazine derivatives decreases with the increasing value of '*PEOE_VSAI*'. The Pearson correlation coefficient between '*PEOE_VSAI*' and redox potential is -0.69, supporting the previous observation. We observed that the value of '*PEOE_VSAI*' is higher for the systems having delocalization of negative partial charge. The delocalized system contains more atoms with the negative partial charge than the corresponding localized system. Thus, the number of atoms contributing to '*PEOE_VSAI*' in delocalized systems is higher than in localized ones. The effect of delocalization of partial charge on '*PEOE_VSAI*' is shown in Figure 4.10 with a few examples from the training-set. Thus, for designing better analytes, it is suggested to increase the delocalization of negative partial charge in the phenazine derivatives.

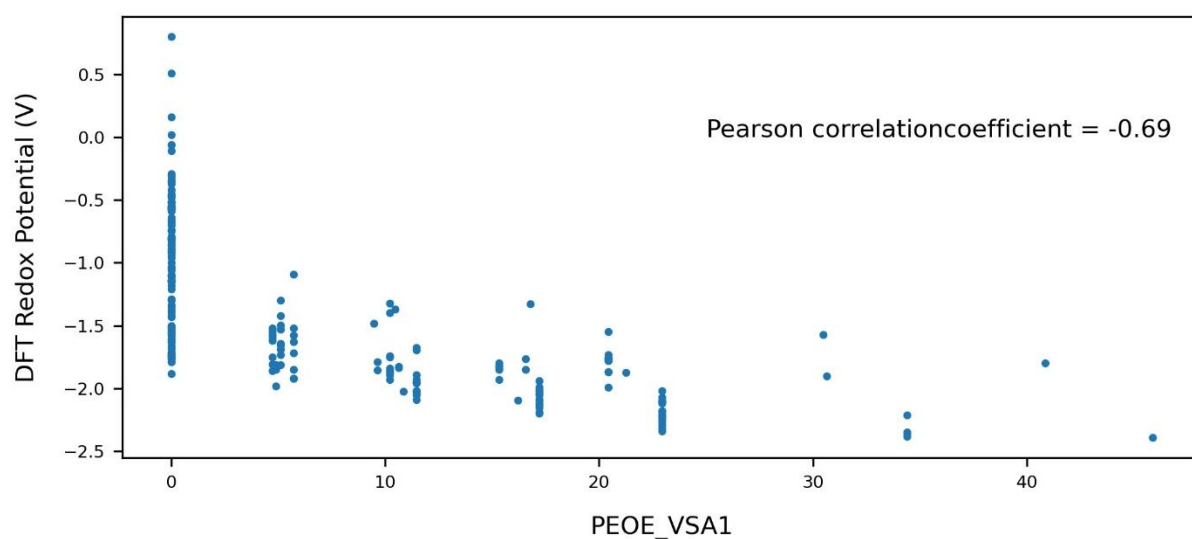


Figure 4.9. Redox Potential vs. '*PEOE_VSAI*'

Mol ID: 0	Mol ID: 3	Mol ID: 105	Mol ID: 134	Mol ID: 136
PEOE_VAS1: 0	PEOE_VAS1: 5.73	PEOE_VAS1: 17.20	PEOE_VAS1: 22.93	PEOE_VAS1: 34.40
Potential: -1.74	Potential: -1.85	Potential: -2.09	Potential: -2.32	Potential: -2.36
PEOE_VSA1 increases		➔		
Delocalization increases		➔		
Redox Potential decreases		➔		

Figure 4.10. Examples from the training-set showing the effect of charge delocalization on 'PEOE_VSA1'. Values of 'PEOE_VSA1' and DFT redox potential in volts are also shown. Mol IDs were assigned to identify derivatives from the corresponding dataset.

The redox potential of phenazine derivative depends on the type of functional group, the position of the attachment, and the number of functional groups. Two types of functional groups have been investigated in this study: (i) electron-donating and (ii) electron-withdrawing. The redox potential of parent phenazine without any functional group is -1.74 V. When the redox potential of the derivative decreases (i.e., less than -1.74 V) after the attachment of functional groups, then it is called a negative shift. Similarly, if it increases, it is called a positive shift. The shift is quantified as the difference between the redox potential of a phenazine derivative and the parent phenazine. After sorting phenazine derivatives based on the redox potential, it was observed that electron-donating groups show a negative shift, whereas electron-withdrawing groups show a positive shift. Thus, the shift corresponding to electron-donating groups is negative, and electron-withdrawing groups is positive. The redox potentials of phenazine derivatives were computed using the approach discussed in section 4.2. Equation 2 shows that the functional groups that stabilize the anionic form of phenazine derivatives have high redox potential. In contrast, those that destabilize anionic form have low redox potential. Therefore, electron-withdrawing groups show a positive shift as they stabilize the anionic form and electron-donating groups show a negative shift as they destabilize the anionic form. A few examples showing positive and negative shifts with respect to parent phenazine are shown in Figure 4.11.

Mol ID: 1	Mol ID: 3	Mol ID: 0	Mol ID: 21	Mol ID: 26
Potential: -1.85	Potential: -1.85	Potential: -1.74	Potential: -1.63	Potential: -1.50
Shift: -0.11	Shift: -0.11	Shift: 0	Shift: 0.11	Shift: 0.24
Electron-donating groups			Electron-withdrawing groups	

Figure 4.11. Examples showing positive and negative shifts with respect to parent phenazine. DFT redox potentials and shifts in volts are also shown. Mol IDs were assigned to identify derivatives from the corresponding dataset.

In the case of derivatives with multiple functional groups, if all groups are similar, then shift also corresponds to their type. For example, when the derivative contains all electron-donating groups, it shows a negative shift. Similarly, the shift is positive when the derivative contains all electron-withdrawing groups. A few examples having similar types of functional groups are shown in Figure 4.12.

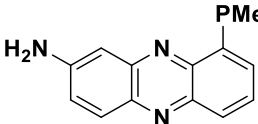
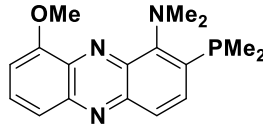
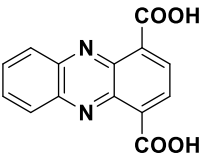
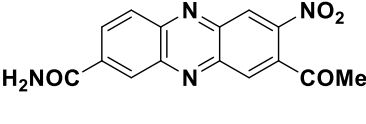
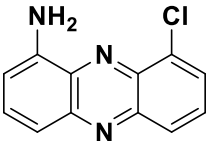
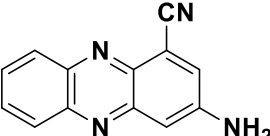
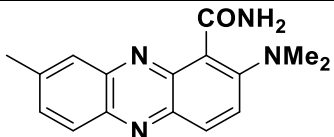
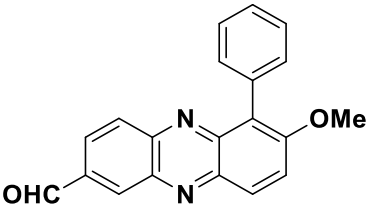
 <p>Mol ID: 9 Potential : -1.92 Shift: -0.18</p>	 <p>Mol ID: 8 Potential : -1.85 Shift: -0.11</p>	 <p>Mol ID: 7 Potential : -1.40 Shift: 0.34</p>	 <p>Mol ID: 15 Potential : -1.09 Shift: 0.65</p>
Electron-donating groups		Electron-withdrawing groups	

Figure 4.12. Examples showing the effect of similar types of functional groups on the redox potential. DFT redox potentials and shifts in volts are also shown. Mol IDs were assigned to identify derivatives from the corresponding dataset.

When derivatives contain more than one functional group that differ in their type, the shift is determined by the group having the highest absolute shift in the corresponding single functional group derivative. For example, derivative A in Table 4.6 contains -NH_2 , an electron-donating group that has a shift of -0.11 V, and -Cl an electron-withdrawing group that has the shift of 0.13 V. The absolute of the shift for -Cl is more than -NH_2 ; therefore, derivative A shows a positive shift of 0.03 V which supports our claim. A similar analysis is applicable to the derivative B, which also shows a positive shift. Derivative C contains $\text{-N(CH}_3)_2$ and -CH_3 , two electron-donating groups, and $\text{-CO(NH}_2)$, an electron-withdrawing group. The absolute shift of $\text{-N(CH}_3)_2$ is -0.24 V which is the highest among all three groups. Therefore, derivative C shows a negative shift of -0.09 V. Derivative D contains -OCH_3 and $\text{-C}_6\text{H}_5$, two electron-donating groups, and -CHO , one electron-withdrawing group. However, derivative D shows a positive shift as the absolute shift of -CHO is more than both electron-donating groups. Thus, the redox potential of phenazine derivatives containing multiple functional groups is determined by the relative strength of electron-donating or electron-withdrawing power of the individual functional groups.

Table 4.6. Examples showing the effect of the absolute value of a single functional group shift on the redox potential of derivatives containing different types of functional groups. DFT redox potentials and shifts in volts are also shown. Mol IDs were assigned to identify derivatives from the corresponding dataset.

	Phenazine Derivative	Details of the phenazine derivative	Redox potential of the corresponding single functional group derivative	Shift of the corresponding single functional group derivative
A.		Mol ID: 1 Potential: -1.71 Shift: 0.03	-NH ₂ : -1.85 -Cl: -1.61	-NH ₂ : -0.11 -Cl: 0.13
B.		Mol ID: 7 Potential: -1.58 Shift: 0.16	-NH ₂ : -1.92 -CN: -1.42	-NH ₂ : -0.18 -CN: 0.32
C.		Mol ID: 12 Potential: -1.83 Shift: -0.09	-N(CH ₃) ₂ : -1.98 -CH ₃ : -1.79 -CONH ₂ : -1.52	-N(CH ₃) ₂ : -0.24 -CH ₃ : -0.05 -CONH ₂ : 0.22
D.		Mol ID: 4 Potential: -1.54 Shift: 0.20	-OCH ₃ : -1.86 -C ₆ H ₅ : -1.76 -CHO: -1.51	-OCH ₃ : -0.12 -C ₆ H ₅ : -0.02 -CHO: 0.23

The effect of position on the redox potential of single functional group derivatives has been studied by Mavrandonakis and co-workers.¹ They showed that position does not have a significant effect for electron-withdrawing groups. However, electron-donating groups which are capable of intra-molecular hydrogen bonding show more negative shift when attached at position 2 compared to position 1. The position numbers in phenazine derivatives are shown in Figure 4.13. They also investigated the effect of the number of functional groups attached to the phenazine molecule. It was shown that the addition of more electron-withdrawing groups shifts the redox potential continuously towards positive values. However, this effect is less significant for electron-donating groups. The difference between the phenazine derivative with four amino groups and eight amino groups is very small, ~0.05 V. Whereas, the difference between the phenazine derivative with four cyano groups and eight cyano groups is ~1.23 V.

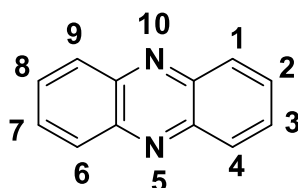


Figure 4.13. Numbering of the positions in phenazine derivatives

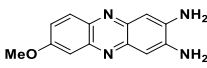
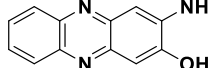
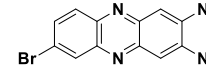
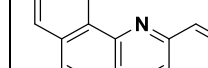
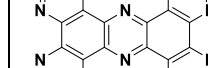
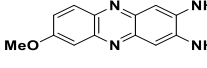
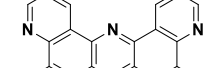
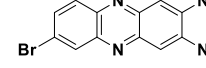
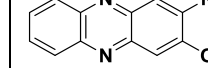
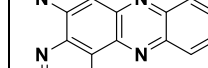
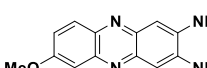
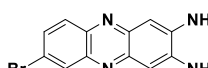
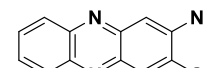
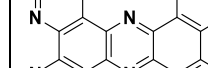
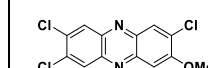
4.3.5 Identification of the Promising Phenazine Derivatives for Anolyte

In this section, we identified the top five promising candidates for anolyte in RFBs using the trained machine learning models. Models developed in this study are based on features that do not require electronic structure calculations. Therefore, these models could screen millions of molecules in a significantly small amount of time. Then, experimentation or DFT calculations could be performed on the reduced number of molecules to identify the best redox-active molecules, saving computational and experimental costs. Using this hybrid DFT-ML approach, we have identified promising phenazine derivatives for anolyte in RFBs. These promising candidates would provide a good starting point for the experimentalists. Electron-donating molecules with negative redox potential are preferred candidates for the anolyte. As KRR and SVR are stable models, the predictions here are based on them. The values of redox potentials are averaged over 100 independent iterations of data splitting and model training. Table 4.7 lists the top five phenazine derivatives from the external test-set with the most negative redox potentials obtained from DFT and two machine learning models. 4 out of 5 predictions from KRR and SVR match with DFT predictions.

Because of the finite size of training-set, machine learning models cannot give reliable results on all the datasets. Although machine learning models promise reasonably accurate predictions on the unseen datasets, their performance is still restricted to a subset of datasets, known as the applicability domain. Model predictions outside this domain cannot be trusted and often yield poor results. Models developed in this study also have their own applicability domain. If one wishes to use these models, it is important to understand their applicability domain. Here, we provide some crucial points on the applicability domain of the machine learning models developed in this study.

- These machine learning models are only applicable to the derivatives generated from phenazine molecules.
- The predictions are more reliable for the derivatives containing functional groups from the training-set (i.e., $-\text{N}(\text{CH}_3)_2$, $-\text{NH}_2$, $-\text{OH}$, $-\text{OCH}_3$, $-\text{P}(\text{CH}_3)_2$, $-\text{SCH}_3$, $-\text{SH}$, $-\text{CH}_3$, $-\text{C}_6\text{H}_5$, $-\text{CH}=\text{CH}_2$, $-\text{F}$, $-\text{Cl}$, $-\text{CHO}$, $-\text{COCH}_3$, $-\text{CONH}_2$, $-\text{COOCH}_3$, $-\text{COOH}$, $-\text{CF}_3$, $-\text{CN}$ and $-\text{NO}_2$).
- The predictions are also reliable for the derivatives containing two and three different types of functional groups from the above list.
- Machine learning models may also be applicable for the phenazine derivatives containing upto four six-membered rings that are attached either through a bond or through conjugation to the central phenazine molecule.
- Models may also be applicable to the derivatives having functional groups that are similar to the functional groups investigated in this study.

Table 4.7. Top five analyte candidates predicted using DFT, KRR, and SVR from the external test-set. SVR and KRR were trained on the phenazine derivatives containing a single type of functional group per derivative. Mol IDs, and redox potentials predicted from DFT and ML models are shown below the respective candidates. Mol IDs were assigned to identify derivatives from the corresponding test-set. Derivatives are arranged in increasing order of their redox potential. Redox potentials are given in the unit of volt.

DFT	 Mol ID: 13 DFT: -2.09	 Mol ID: 29 DFT: -2.02	 Mol ID: 12 DFT: -1.95	 Mol ID: 1 DFT: -1.88	 Mol ID: 5 DFT: -1.87
KRR	 Mol ID: 13 ML: -2.09 DFT: -2.09	 Mol ID: 5 ML: -2.09 DFT: -1.87	 Mol ID: 12 ML: -1.98 DFT: -1.95	 Mol ID: 29 ML: -1.95 DFT: -2.02	 Mol ID: 4 ML: -1.78 DFT: -1.83
SVR	 Mol ID: 13 ML: -2.06 DFT: -2.09	 Mol ID: 12 ML: -1.96 DFT: -1.95	 Mol ID: 29 ML: -1.91 DFT: -2.02	 Mol ID: 5 ML: -1.89 DFT: -1.87	 Mol ID: 3 ML: -1.81 DFT: -1.52

4.4 Conclusions

In this study, four machine learning models were employed to predict the redox potential of phenazine derivatives in dimethoxyethane (DME) using density functional theory (DFT). Models were trained on a small dataset of 151 phenazine derivatives having only one type of functional group per molecule (20 unique functional groups). The trained models achieved high accuracies ($R^2 > 0.74$) on internal as well as external test-sets containing diverse phenazine derivatives. We also showed that despite being trained on derivatives with a single type of functional group, models were able to predict the redox potentials of the derivatives containing multiple and different types of functional groups with good accuracies ($R^2 > 0.7$). Feature selection and hyperparameter optimization using the validation-set were critical strategies for performance improvement. Feature selection removed the unnecessary and noisy features. Hyperparameter optimization using an external validation-set helped in improving the generalizability of the models. The addition of fifteen derivatives from two functional group test-set in the training set improved the accuracy on the three functional group test-set. It was observed that the '*PEOE_VSAI*' descriptor was the most important molecular feature as it contains information related to molecular size and the partial charges. A deeper analysis showed that one should not rely only on the model performance but also investigate the stability and reliability of the models. From structure-functional relationship, we observed that the redox potential of derivatives containing multiple functional groups is influenced by the functional group having either strong electron-donating or strong electron-withdrawing power. Models developed in this study are based on features that do not require electronic structure calculations or experimentation. Therefore, these models could potentially screen millions of molecules in a significantly small amount of time. Then, experimentation or DFT calculations could be performed on the reduced number of molecules to identify the best molecules, saving computational and experimental costs. Using this hybrid DFT-ML approach, we have identified promising phenazine derivatives for anolyte in RFBs. These promising candidates would provide a good starting point for the experimentalists. This study shows that it is possible to develop reasonably accurate machine learning models for complex quantities such as redox potential using small and simple datasets.

4.5 References

- (1) De La Cruz, C.; Molina, A.; Patil, N.; Ventosa, E.; Marcilla, R.; Mavrandonakis, A. New Insights into Phenazine-Based Organic Redox Flow Batteries by Using High-Throughput DFT Modelling. *Sustain. Energy Fuels* **2020**, *4* (11), 5513–5521. <https://doi.org/10.1039/d0se00687d>.
- (2) Landrum, G. RDKit: Open-source cheminformatics <https://www.rdkit.org/> (accessed Oct 23, 2021).
- (3) M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, G. A. Petersson, H. Nakatsuji, X. Li, M. Caricato, A. Marenich, J. Bloino, B. G. Janesko, R. Gomperts, B. Mennucci, H. P. Hratchian, J. V. Ort, and D. J. F. Gaussian 09. Gaussian, Inc.: Wallingford CT 2016.
- (4) Lee, C.; Yang, W.; Parr, R. G. Development of the Colle-Salvetti Correlation-Energy Formula into a Functional of the Electron Density. *Phys. Rev. B* **1988**, *37* (2), 785. <https://doi.org/10.1103/PhysRevB.37.785>.
- (5) Becke, A. D. Density-functional Thermochemistry. III. The Role of Exact Exchange. *J. Chem. Phys.* **1998**, *98* (7), 5648. <https://doi.org/10.1063/1.464913>.
- (6) Tirado-Rives, J.; Jorgensen, W. L. Performance of B3LYP Density Functional Methods for a Large Set of Organic Molecules. *J. Chem. Theory Comput.* **2008**, *4* (2), 297–306. https://doi.org/10.1021/CT700248K/SUPPL_FILE/CT700248K-FILE002.PDF.
- (7) Schuchardt, K. L.; Didier, B. T.; Elsethagen, T.; Sun, L.; Gurumoorthi, V.; Chase, J.; Li, J.; Windus, T. L. Basis Set Exchange: A Community Database for Computational Sciences. *J. Chem. Inf. Model.* **2007**, *47* (3), 1045–1052. https://doi.org/10.1021/CI600510J/SUPPL_FILE/CI600510J2.TXT.
- (8) Zhao, Y.; Truhlar, D. G. Density Functionals with Broad Applicability in Chemistry. *Acc. Chem. Res.* **2008**, *41* (2), 157–167. <https://doi.org/10.1021/AR700111A>.
- (9) Papajak, E.; Truhlar, D. G. Efficient Diffuse Basis Sets for Density Functional Theory. *J. Chem. Theory Comput.* **2010**, *6* (3), 597–601. https://doi.org/10.1021/CT900566X/SUPPL_FILE/CT900566X_SI_001.PDF.
- (10) Treitel, N.; Shenhar, R.; Arahamian, I.; Sheradsky, T.; Rabinovitz, M. Calculations of PAH Anions: When Are Diffuse Functions Necessary? *Phys. Chem. Chem. Phys.* **2004**, *6* (6), 1113–1121. <https://doi.org/10.1039/B315069K>.
- (11) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, É. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12* (85), 2825–2830.
- (12) Nakagawa, R.; Nishina, Y. Simulating the Redox Potentials of Unexplored Phenazine Derivatives as Electron Mediators for Biofuel Cells. *J. Phys. Energy* **2021**, *3* (3), 034008. <https://doi.org/10.1088/2515-7655/ABEBC8>.
- (13) Miao, L.; Liu, L.; Zhang, K.; Chen, J. Molecular Design Strategy for High-Redox-Potential and Poorly Soluble n-Type Phenazine Derivatives as Cathode Materials for

- Lithium Batteries. *ChemSusChem* **2020**, *13* (9), 2337–2344. <https://doi.org/10.1002/CSSC.202000004>.
- (14) Sousa, A. C.; Martins, L. O.; Robalo, M. P. Laccases: Versatile Biocatalysts for the Synthesis of Heterocyclic Cores. *Mol.* **2021**, *Vol. 26*, Page 3719 **2021**, *26* (12), 3719. <https://doi.org/10.3390/MOLECULES26123719>.
- (15) Castro, K. P.; Clikeman, T. T.; DeWeerd, N. J.; Bukovsky, E. V.; Rippey, K. C.; Kuvychko, I. V.; Hou, G.-L.; Chen, Y.-S.; Wang, X.-B.; Strauss, S. H.; Boltalina, O. V. Incremental Tuning Up of Fluorous Phenazine Acceptors. *Chem. – A Eur. J.* **2016**, *22* (12), 3930–3936. <https://doi.org/10.1002/CHEM.201504122>.
- (16) Wang, C.; Li, X.; Yu, B.; Wang, Y.; Yang, Z.; Wang, H.; Lin, H.; Ma, J.; Li, G.; Jin, Z. Molecular Design of Fused-Ring Phenazine Derivatives for Long-Cycling Alkaline Redox Flow Batteries. *ACS Energy Lett.* **2020**, *17*, 411–417. <https://doi.org/10.1021/ACSENERGYLETT.9B02676>.
- (17) 1.1. Linear Models — scikit-learn 1.0 documentation https://scikit-learn.org/stable/modules/linear_model.html#bayesian-regression (accessed Oct 23, 2021).
- (18) Wipf, D.; Nagarajan, S. A New View of Automatic Relevance Determination. *Adv. Neural Inf. Process. Syst.* **2007**, *20*.
- (19) 1.7. Gaussian Processes — scikit-learn 1.0 documentation https://scikit-learn.org/stable/modules/gaussian_process.html (accessed Oct 23, 2021).
- (20) Sit, H. Quick Start to Gaussian Process Regression <https://towardsdatascience.com/quick-start-to-gaussian-process-regression-36d838810319> (accessed Oct 23, 2021).
- (21) 1.3. Kernel ridge regression — scikit-learn 1.0 documentation https://scikit-learn.org/stable/modules/kernel_ridge.html (accessed Oct 23, 2021).
- (22) 1.4. Support Vector Machines — scikit-learn 1.0 documentation <https://scikit-learn.org/stable/modules/svm.html#svm-regression> (accessed Oct 23, 2021).
- (23) Cao, J.; Tian, J.; Xu, J.; Wang, Y. Organic Flow Batteries: Recent Progress and Perspectives. *Energy and Fuels* **2020**, *34* (11), 13384–13411. <https://doi.org/10.1021/acs.energyfuels.0c02855>.
- (24) sklearn.feature_selection.SelectKBest — scikit-learn 1.0.2 documentation https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html (accessed Jan 7, 2022).
- (25) 4.2. Permutation feature importance — scikit-learn 1.0 documentation https://scikit-learn.org/stable/modules/permutation_importance.html#permutation-importance (accessed Oct 23, 2021).
- (26) 1.1. Linear Models — scikit-learn 1.0 documentation https://scikit-learn.org/stable/modules/linear_model.html (accessed Oct 21, 2021).
- (27) Landrum, G. Getting Started with the RDKit in Python — The RDKit 2020.03.1 documentation <https://www.rdkit.org/docs/GettingStartedInPython.html#list-of>

- available-descriptors (accessed Mar 31, 2021).
- (28) QuaSAR-Descriptor <http://www.cadaster.eu/sites/cadaster.eu/files/challenge/descr.htm> (accessed Oct 22, 2021).
- (29) rdkit.Chem.MolSurf module — The RDKit 2021.09.1 documentation <https://www.rdkit.org/docs/source/rdkit.Chem.MolSurf.html> (accessed Dec 29, 2021).
- (30) Gasteiger, J.; Marsili, M. Iterative Partial Equalization of Orbital Electronegativity—a Rapid Access to Atomic Charges. *Tetrahedron* **1980**, *36* (22), 3219–3228. [https://doi.org/10.1016/0040-4020\(80\)80168-2](https://doi.org/10.1016/0040-4020(80)80168-2).

Chapter 5

Investigating Combinatorial Binding of Transcription Factors using Unsupervised Machine Learning Models

Chapter 5

Investigating Combinatorial Binding of Transcription Factors using Unsupervised Machine Learning Models

Abstract

The appearance of an organism is determined by the genes expressed within it. Any disruption to the process of gene expression could result in severe diseases. The healthy expression of a gene requires appropriate regulation at the transcriptional level. Transcription is regulated through the binding of transcription factors (TFs) to various regulatory elements such as promoters, enhancers, silencers, and insulators present in the DNA. Next-generation sequencing technologies such as ChIP-seq can identify DNA regions containing these regulatory elements. However, ChIP-seq requires access to the antibodies of a given protein. Therefore, a large-scale assay may not always be feasible using ChIP-seq. On the other hand, DNase-seq, another sequencing technology, can identify open chromatin regions without antibodies. Therefore, DNase-seq could be cost-effective and relatively faster than ChIP-seq. Furthermore, regulatory elements could also be identified using DNase-seq, because many transcriptional regulatory elements are believed to be present in the open chromatin regions. The binding of transcription factors to regulatory regions is combinatorial— a transcription factor may bind to multiple regulatory regions, and many interacting transcription factors may simultaneously bind to a regulatory region or multiple regulatory regions. Typically, thousands of regions are obtained from a single sequencing experiment. Therefore, it is essential to group regions into biologically relevant modules for the analysis. Soft clustering methods such as topic models are better suited for discovering regulatory modules. However, to the best of our knowledge, only a handful of studies report the application of topic models to ChIP-seq and DNase-seq datasets. In this study, we have employed three unsupervised machine learning algorithms — Latent Dirichlet Allocation (LDA) and Hierarchical Dirichlet Processes (HDP), and a recently developed No Promoter Left Behind (NPLB) method to discover regulatory modules directly from the ChIP-seq and DNase-seq data of the K562 cell line. The results indicate that modules containing functionally similar TFs and regulatory elements could be discovered using topic models and NPLB from ChIP-seq and DNase-seq data without prior information on TF binding sites. Furthermore, it was observed that NPLB gives a more robust performance than topic models on the datasets analyzed in this study.

5.1 Introduction

An extensive understanding of health and diseases requires the interpretation of cellular variations at multiple levels such as genome, transcriptome, proteome, and epigenome. Cells depend on thousands of proteins to perform their tasks, such as growth and cell division. Proteins are encoded in the genes, and the process by which genetic instructions are converted into proteins is known as gene expression. The first step in this process is transcription, where information on the DNA is transferred to an RNA molecule. Then, RNA molecules are utilized to make proteins that carry out various functions inside the cell.¹ Transcription is a complex process that requires precise coordination between the special class of proteins called transcription factors (TFs) and regulatory elements present on the DNA. Regulatory elements are the regions on the genome where TFs bind and regulate the transcription, thereby affecting gene expression. Several regulatory elements control gene expression, including promoters, enhancers, silencers, insulators, and locus control regions. These regulatory elements are schematically shown in Figure 5.1. Promoters are located close to the gene where transcription machinery assembles.² They act as an on-off switch for the transcription. Enhancers control the activity of promoters, instructing when, where, and at what levels to carry out transcription. Enhancers are generally located in open chromatin regions of the genome. They play a major role in controlling gene expression.³ Insulators act as boundary elements defining a region containing regulatory elements for a gene.⁴ Silencers are negative regulatory elements that suppress gene expression. Silencers are thought to carry out their repressive action by looping to their target promoter or competing for a TF binding site with the promoter.⁵

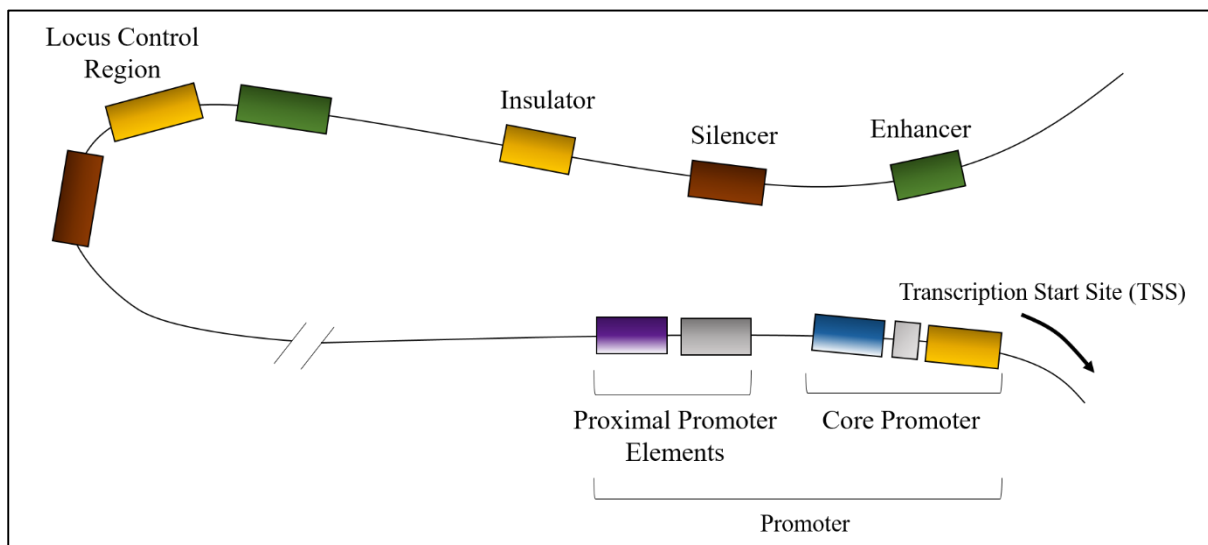


Figure 5.1. Schematic diagram of a regulatory region containing different regulatory elements.

Sometimes, a change in a gene's instruction (also known as a mutation) can cause a protein to malfunction or not be produced at all. When mutations alter a protein that plays a critical role in our body, it can disrupt normal development or cause a health condition. A health condition resulting from mutations in one or more genes is known as a genetic disorder.⁶ Some genetic mutations are so severe that they prevent an embryo from surviving until birth. These mutations occur in genes essential for the development, disrupting embryo development in its early stages.⁷⁻⁹ Mutations in the protein-coding regions have primarily been studied, as they are

directly linked with human disease. However, sequencing the coding regions of a person suspected of having a genetic disorder generally identifies only 20-25% of disease-associated mutations.¹⁰ While technological or other limitations of genome sequencing may cause failure to identify mutations in the coding region, some of the causative agents presumably lie outside the coding region, i.e., within the regulatory and other non-coding regions of the genome. Although the non-coding region does not provide instructions for making proteins, it is integral to the functioning of cells, particularly for the control of gene activity. It contains gene regulatory elements such as promoters, enhancers, silencers, and insulators.¹¹ The growing body of literature indicates that mutations in regulatory elements, like enhancers and insulators, in non-coding regions of the genome are associated with congenital genetic conditions.¹¹ Mutations in transcriptional regulatory elements and transcriptional machinery have been associated with diseases.² Somatic mutations of regulatory elements have also been investigated in cancer, describing their importance.¹² DNA variants resulting from mutations within enhancer elements genetically predispose to various common and complicated traits, such as heart disease, diabetes, cancer, obesity, hair color, etc.¹³ On the other hand, disease-causing mutations in genes oftentimes disrupt amino acids or splicing and alter the function or levels of protein production. The dominant thought is that mutations in regulatory elements cause changes in phenotypes through abnormal expression.¹⁴ In order to understand the mechanisms governing gene expression in a pathological condition, it is essential to identify regulatory elements associated with the target genes. Limitations in technology and lack of knowledge about the specific positions of regulatory elements have made researchers focus on coding regions where 85% of known disease-causing variants have been located.¹⁵

However, recent advances in genome sequencing have led to the invention of next-generation sequencing (NGS) technologies.¹⁶ NGS is a collection of technologies that use massively parallel sequencing approaches to produce millions of short-read sequences at a much cheaper cost and in a short amount of time. NGS-based approaches have been adopted to sequence mutations in the entire genome, exome, or any section of the DNA, obtain DNA copy number information, sequence the whole transcriptome, and quantify gene expression levels.¹⁷ NGS has accelerated the discovery of gene regulatory elements on a genome-wide scale. Examples of NGS-based technologies include ChIP-seq and DNase-seq.

ChIP-seq: ChIP-seq is a technique that combines chromatin immunoprecipitation (ChIP) with massively parallel DNA sequencing to identify the binding sites of proteins.¹⁸ It is used for mapping genome-wide binding sites for any protein of interest. ChIP-seq is one of the critical tools in genomics and epigenomics and has helped discover disease-associated with transcriptional regulation.¹⁹ The ChIP-seq protocol involves sample preparation and computational analysis. In the first step, cross-linked DNA is sonicated and purified with and without immunoprecipitation. Immunoprecipitation is carried out using a specific antibody. Then, DNA fragments are sequenced, which are then mapped onto a reference genome. Next, peaks are detected by comparing genomic regions with the input reads to identify significantly enriched regions.²⁰ Other genomic regions are considered as non-specific background. These peaks represent candidate positions for the targeted protein. The peaks are classified into three categories depending on their shapes: (a) “sharp mode”: these peaks are located at a specific position in the genome; (b) “broad mode”: these peaks are associated with large genomic domains; (c) “mix mode”: this includes both peak modes. Advances in sequencing technology

and analyses have enabled us to handle hundreds of ChIP samples simultaneously, revealing the high-dimensional interrelationship for regulatory elements.²¹

DNase-seq: One of the factors contributing to the cell-specific binding of TFs is chromatin structure. Chromatin is the DNA wrapped around nucleosomes linked together by DNA strands and organized structurally into accessible and inaccessible domains.²² The interaction between chromatin and transcription factor binding is not straightforward. However, accessible chromatin is generally associated with the binding of transcription factors to DNA.²³ Some transcription factors influence the chromatin structure around their binding site, which may facilitate the binding of new transcription factors.²⁴ Changes resulting from such events are the foundation for cellular processes. DNase-seq is a technique that detects Deoxyribonuclease I hypersensitive (i.e., DNase I HS) sites (open chromatin regions) across the genome by capturing DNase-digested fragments and sequencing them by high-throughput next-generation sequencing.²⁵ DNase I is an enzyme that preferentially cuts DNA at open sites. Eukaryotic DNA is packed into a repeating chain of nucleosomes.²² These nucleosomes block DNase I from nicking DNA strands, resulting in preferential sensitivity of the accessible nucleosome-free regions to the cleavage by DNase I. DNase-seq has been widely used to determine chromatin accessibility. The region near the active site is likely to have an altered nucleosome state, making DNase-seq an excellent tool for mapping genomic regulatory elements.^{25,26}

Regulatory elements could be identified using ChIP-seq experiments. However, ChIP-seq is limited to known TFs with previously derived antibodies and requires separate experimentation for each TF. Transcription regulation requires access to the chromatin regions containing regulatory elements such as promoters, enhancers, etc. Therefore, open chromatin regions are likely to have regulatory elements. DNase-seq detects these open chromatin regions. Thus, regulatory elements could also be identified using DNase-seq. The advantage of DNase-seq over ChIP-seq is that DNase-seq, being TF-agnostic, does not require access to the antibodies of individual TFs.

A fundamental question in biology is how TF-DNA interaction affects gene expression. We know that TFs bind to promoter regions proximal to the gene transcription start sites (TSSs) or distant enhancer regions that regulate expression through long-range interactions.²⁷ TF binding shows heterogeneity within cell types.²⁸ Only about two percent of the DNA in the human genome contains protein-coding genes (i.e., ~20,000 – 25,000 genes), out of which only ~1850 encode for TFs.^{2,29} The small number of transcription factors compared to genes suggests their combinatorial activity in gene regulation. It has been shown that gene expression is regulated by the combination of TFs.³⁰ The combinatorial binding of TFs dictates the spatial and temporal activity of gene regulation.^{31,32} In this study, a regulatory region is defined as a DNA segment containing one or more regulatory elements, and a regulatory module is defined as a set of TFs that bind together to similar regulatory regions. Insight into the interplay between regulatory modules is essential for understanding the complexity of gene regulation. Previous reports have shown that TFs often bind in clusters, resulting in a large number of binding sites in a regulatory region.^{33–35} These co-binding TFs may belong to distinct functional modules but come together in the regulatory regions to execute a specific task. For example, transcription is initiated through the interaction of enhancer-bound TFs, promoter-bound TFs, and other TFs that bring promoters and enhancers together, such as CTCF and cohesin. Thus, CTCF and cohesin modules are likely to co-occur with enhancer and promoter related modules.³⁶ This type of

module co-occurrence points to a modular hierarchy in the combinatorial binding of TFs in which regulatory regions may use multiple regulatory modules, and each regulatory module itself is a combination of multiple TFs.

Previous investigations into combinatorial binding could not identify modular hierarchy in TF binding because — (i) they were looking at either the regions bound by a particular TF or pairs of co-binding TFs,^{33,37} or (ii) they did not account for the modularity in TF binding during the analysis.^{38–40} Large-scale efforts such as the Encyclopedia of DNA Elements (ENCODE) have profiled *in vivo* binding of hundreds of TFs in multiple cells.³⁷ Therefore, it has become possible to discover previously unknown regulatory modules. Furthermore, advancements in machine learning have resulted in novel algorithms capable of uncovering different clusters present within the data. In this case, these clusters represent different regulatory modules. Unsupervised machine learning algorithms such as self-organizing maps (SOMs) and k-mean clustering have been employed to reveal the combinatorial binding of TFs.^{41,42} The issue with these hard clustering methods is that they model binding at a given region with a single regulatory module. Therefore, they require a large number of modules to represent binding events. Soft clustering methods such as topic models are better suited to model combinatorial binding of TFs in which each motif is assumed to be part of multiple modules. Topic models are the class of unsupervised machine learning algorithms commonly used for discovering topics from the corpus of documents. They could also be viewed as clustering algorithms for documents.

In this study, we consider a set of DNA regions to be analogous to a document corpus where each DNA segment represents a document containing different regulatory modules (i.e., topics). Topic models have been employed to investigate transcriptomic data, particularly RNA-seq.^{43–45} However, to the best of our knowledge, only a handful of studies report the application of topics models to ChIP-seq and DNase-seq datasets. Li Chen *et al.* developed a computational method based on topic models to decipher the combinatorial binding events of TFs from multiple ChIP-seq datasets.⁴⁶ Stein Aerts *et al.* developed cisTopic, a probabilistic framework to simultaneously discover co-accessible enhancers and stable cell states from sparse single-cell epigenomics data.⁴⁷ Guo and Gifford employed hierarchical Dirichlet processes to investigate the combinatorial binding of TFs.⁴⁸ We also found a lack of literature on the application of topic models to the DNase-seq data. However, being TF-agnostic, DNase-seq data might contain more information about the combinatorial binding of TFs.⁴⁹ In this study, we conducted a comparative study on three unsupervised methods, including two commonly used topic models (Latent Dirichlet Allocation and Hierarchical Dirichlet Processes) and a recently developed No Promoter Left Behind (NPLB) method to investigate the combinatorial and modular binding of TFs from ChIP-seq and DNase-seq datasets.

5.2 Materials and Methods

5.2.1 Data Generation

Simulated dataset: This dataset was obtained from the earlier work by Biswas and Narlikar.⁵⁰ The dataset includes 1000 DNA sequences of 200 bp each, sampled randomly from the non-repetitive section of the human genome. DNA sequences were implanted with five motifs randomly selected from the JASPAR2018 CORE vertebrate motif set. This dataset contains three modules, each with unique distribution over five motifs, as shown in Table 5.1.

Table 5.1. Distribution of motifs across three modules in the simulated dataset. Cells represent motif counts.

	Motif 1	Motif 2	Motif 3	Motif 4	Motif 5
Module 1	480	480	100	0	0
Module 2	0	0	478	345	0
Module 3	38	0	0	0	42

ChIP-seq (CTCF): The ChIP-seq data of the human K562 cell line for CTCF protein was obtained from the ENCODE project (ENCFF738TKN). The K562 cell line contains the bone marrow cells of a 53-year-old myelogenous leukemia patient. It has been extensively used in hematopoietic research. A bed narrowPeak file aligned to the hg19 genome was downloaded, containing the coordinates of 56,891 DNA regions enriched in CTCF. The HOMER⁵¹ motif analysis tool was used to identify the number and the positions of known motifs in each region. A region-motif matrix was constructed using the motif information obtained from the HOMER.

ChIP-seq (115 TFs): This dataset was obtained from the report by Guo and Gifford.⁴⁸ It contains ~142,960 non-overlapping co-binding regions pooled from the ChIP-seq datasets of 115 TFs in the human K562 cell line. In contrast to other datasets investigated in this report, the region-TF matrix of this dataset was constructed using binding calls or peaks instead of motif information. GPS binding calls of all TFs were pooled together to construct a region-TF for this dataset. The regions were aligned to the hg19 genome prior to identifying bind events.

DNase-seq: The DNase-seq data of the human K562 cell line was obtained from the ENCODE project (ENCFF621ZJY). The bed narrowPeak file aligned to the hg19 genome was downloaded, containing the coordinates of 378,491 DNA regions preferentially cleaved by DNase I. The HOMER motif analysis tool was used to identify the number and the positions of known motifs in each region. A region-motif matrix was constructed using the motif information obtained from the HOMER.

In this report, ‘region-TF’ and ‘module-TF’ refer to the matrices constructed using TF binding events (peaks), whereas ‘region-motif’ and ‘module-motif’ refer to the matrices constructed using annotated positions obtained from HOMER.

5.2.2 Models

The following three unsupervised machine learning algorithms were investigated in this study. They include two topic models and a recently developed No Promoter Left Behind (NPLB) model. A topic model is a type of statistical framework for discovering topics from the

collection of documents. Topic modeling is frequently used to discover hidden semantic structures within the documents. If a document belongs to a particular topic, then words related to that topic will occur more frequently than others. A document may contain more than one topic in different proportions. The topics produced by a topic model define a cluster of similar words. Here, we assume that a set of DNA regions is analogous to a document corpus in which each DNA region is a document containing different regulatory elements (i.e., words) and regulatory modules (i.e., topics). The mathematical framework of the topic models allows us to capture the underlying topics based on the statistics of the words in the documents. The latent Dirichlet allocation (LDA) and the hierarchical Dirichlet processes (HDP) are the two most commonly used topic models. An implementation of LDA from the Gensim python library was used in this work.⁵² HDP was implemented using the hdp Python library developed by altosaar *et al.*⁵³ The NPLB library developed by Mitra and Narlikar was used for the development of the NPLB approach.⁵⁴ Below, we give a brief introduction to the models used in this study:

Latent Dirichlet Allocation (LDA): LDA is one of the most widely used topic models in natural language processing. It is a generative statistical model that allows inferring hidden groups within the collection of documents.⁵⁵ In LDA, we assume that the corpus is composed of latent topics that are not directly observable, and each document has a distribution over these topics. Also, each topic is represented as a distribution over unique words. Similar words describing the same topic have a high probability in the topic distribution. At the same time, unrelated words have a low probability. LDA assumes the following generative process:

- For each topic, we randomly sample word distribution from the Dirichlet distribution, $\phi_k \sim Dir(\beta)$.
- For each document,
 - we randomly sample topic distribution from another Dirichlet distribution, $\theta_m \sim Dir(\alpha)$.
 - Then, for each word position in a document, we randomly sample a topic from the multinomial distribution corresponding to that document, $Z_{m,n} \sim Mult(\theta_m)$.
 - Then, we randomly sample a word from the multinomial distribution corresponding to that topic and repeat the process for each word position for all the documents, $w_{m,n} \sim Mult(\phi_k)$ where $k = Z_{m,n}$.

Hierarchical Dirichlet Processes (HDP): HDP is a nonparametric Bayesian model used for clustering grouped data.⁵⁶ It uses a Dirichlet process for each group of data sharing a base distribution. The base distribution is itself drawn from a Dirichlet process. A Dirichlet process is a probability distribution over probability distributions. Draws from a Dirichlet process are discrete and infinite probability measures appropriate for representing the proportions of mixture components. A common base distribution allows sharing of clusters across groups. HDP was developed by Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David Blei to address an important issue of LDA.⁵⁶ LDA requires the specification of the number of topics, which is not always known. In contrast, HDP does not assume the number of topics, and it allows a corpus to contain any number of topics while sharing the topics among different documents. HDP is an extension of LDA with the following generative process:

- First, a common base distribution is drawn from a Dirichlet process. The base distribution provides the set of all topics that can be used in a given corpus, $G_0|\gamma, H \sim DP(\gamma, H)$
- For each document in the corpus, we sample the topic distribution from another Dirichlet process using the common base distribution, $G_j|\alpha_0, G_0 \sim DP(\alpha_0, G_0)$
- From the topic distribution, we sample the multinomial distribution over words and sample words from it for each word position and repeat the process for all the documents,

$$\begin{aligned}\theta_{ji}|G_j &\sim G_j \\ x_{ji}|\theta_{ji} &\sim Mult(\theta_{ji})\end{aligned}$$

No Promoter Left Behind (NPLB): NPLB is a novel approach for identifying heterogeneous promoter architectures from high-throughput TSS data. It uses only the genomic sequence around the TSS location as input and does not require any prior information on the promoter elements. NPLB could be viewed as an unsupervised machine learning algorithm that clusters promoter sequences (i.e., DNA regions) into groups having similar architectures while simultaneously identifying important positions in the promoters for each architecture. Here, we briefly describe the NPLB model:

NPLB is concerned with partitioning n DNA regions X_1, \dots, X_n each with length l into k different architectures a_1, \dots, a_k . The j -th nucleotide in the i -th region is represented by X_i^j , where $1 \leq j \leq l$. We assume that each architecture a_u , where $1 \leq u \leq k$, has some important positions denoted by the set $I_{a_u} \subset \{1, \dots, l\}$. The model is characterized by the number of architectures k and the number of important positions present in each architecture a_u , i. e. $|I_{a_u}|$. For a given architecture, the parameters of the model are defined as follows:

- The architecture to which X_i belongs is represented as y_i and is modeled using a categorical distribution γ over $\{1, \dots, k\}$.
- Each important position j in the architecture a_u is modeled using a categorical distribution $\phi_{a_u^j}$ over the four base pairs. All other unimportant positions are modeled using a common background categorical distribution ϕ_0 . The background distribution applies to all architectures.

For a fixed model structure (i.e., hyperparameters), the parameters are learned using Gibbs sampling that maximizes the posterior distribution. The hyperparameters are determined using k-fold cross-validation by varying the total number of architectures and the number of important positions for each architecture. The model with the highest cross-validation likelihood is selected as the final model. Although NPLB was designed to cluster DNA sequences based on base pairs, we modified its typical workflow to cluster DNA regions obtained from CHIP-seq and DNase-seq data based on the presence or absence of the know TFs (motifs).

Table 5.2. List of hyperparameters used during training. An array in front of some hyperparameters represents different values tested during training.

Model Name	Dataset	Hyperparameters
LDA	Simulated dataset	α = “symmetric” η = 0.1 num_topics = 3 iterations = 1000 passes = 10
	ChIP-seq (CTCF)	α = “symmetric” η = [0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1] num_topics = 19 iterations = 2000 passes = 200
	ChIP-seq (115 TFs)	α = “symmetric” η = 0.1 num_topics = 49 iterations = 2000 passes = 2000
	DNase-seq	α = “symmetric” η = 0.1 num_topics = 49 iterations = 2000 passes = 10
HDP	Simulated dataset	η = [0.01, 0.05, 0.1, 0.5, 1] γ_a = [0.1, 1, 5] γ_b = [0.1, 1, 5] α_a = [0.1, 1, 5] α_b = Fifty equally spaced points between 1 and 5. init_topics = 3 max_iter = 20000
	ChIP-seq (CTCF)	η = [0.01, 0.05, 0.1, 0.5, 1] γ_a = 1.00 γ_b = 1.00 α_a = 1.00 α_b = 1.00 init_topics = 10 max_iter = 2000
	ChIP-seq (115 TFs)	η = 0.1 γ_a = 1.00 γ_b = 1.00 α_a = 1.00 α_b = 1.00 init_topics = 50 max_iter = 2000

	DNase-seq	eta = 0.1 gamma_a = 1.00 gamma_b = 1.00 alpha_a = 1.00 alpha_b = 1.00 init_topics = 50 max_iter = 2000
NPLB	Simulated dataset	default hyperparameters: minarch = 1 maxarch = 20 kfold = 5
	ChIP-seq (CTCF)	default hyperparameters: minarch = 1 maxarch = 20 kfold = 5
	ChIP-seq (115 TFs)	minarch = 15 maxarch = 49 kfold = 3
	DNase-seq	minarch = 15 maxarch = 30 kfold = 3

Table 5.3. The list of some important transcription factors found in the K562 cell line. This list serves as a reference to understand the functional roles of the modules discovered in this study.

TFs	Function
GATA1/2, TAL1	Master regulators of K562 cell. GATA-1 controls hematopoietic development by activating and repressing gene transcription.
CTCF, cohesin subunits RAD21 and SMC3, and ZNF143	CTCF and the cohesin complex function together to establish chromatin loops and regulate gene expression in mammalian cells. CTCF and the cohesin complex, consisting of the core subunits SMC3, SMC1, RAD21, and STAG1/SA1 or STAG2/SA2. ⁵⁷ ZNF143 is a critical factor for CTCF-bound promoter–enhancer loops. ⁵⁸
p300, RCOR1, TEAD4	Co-activators and co-repressors.
Pol2, TBP, and TAF1	Transcriptional machinery. Pol2 transcription machinery is responsible for the transcription of most of the genes in eukaryotes. ⁵⁹
Pol3	Transcriptional machinery. Pol3 is responsible for transcribing short non-coding RNAs such as tRNAs, 5S rRNA, U6 snRNA, and a limited number of others. ⁵⁹
AP-1 factors such as JUN, JUNB, JUND, FOS, FOSL	AP-1 transcription factor is composed of proteins belonging to the Jun (c-Jun, JunB, and JunD), Fos (c-Fos, FosB, Fra1, and Fra2), ATF/cyclic AMP-responsive element-binding (CREB) (ATF1–4, ATF-6, b-ATF, ATFx), and Maf family (c-Maf, MafA, MafB, MafG/F/K, and Nrl). ⁶⁰
MAF, BACH1, NFE2	The MAFs are members of the basic leucine zipper (bZIP) family of transcription factors. Small MAF proteins combine with NFE2, NFE2L1, BACH1, BACH2. ⁶¹

MYC, MAX, USF, E2F6	c-Myc plays a pivotal role in important cellular processes such as proliferation, suppression of differentiation, and apoptosis. C-Myc interacts with Max and binds to the E-box. USF also binds to the E-box to regulate the expression of different target genes. ⁶²
SCL	SCL is an essential regulator at several levels in the hematopoietic hierarchy, whose inappropriate regulation contributes to the development of pediatric T-cell acute lymphoblastic leukaemia. ⁶³
SPI1 (also known as PU.1), SP3 and ELF1	Regulates the expression of the SCL gene. ⁶⁴
FOX	FOX (forkhead box) proteins are a family of transcription factors that play important roles in regulating the expression of genes involved in cell growth, proliferation, differentiation, and longevity. Many FOX proteins are important for embryonic development. ⁶⁵ FOX transcription factors are evolutionarily conserved in organisms ranging from yeast to humans.
STAT	The STAT protein family are intracellular transcription factors that mediate many aspects of cellular immunity, proliferation, apoptosis, and differentiation. ⁶⁶
RUNX	The Runx family of transcription factors (Runx1, Runx2, and Runx3) are highly conserved and involved in various developmental and cellular processes, such as cell proliferation, differentiation, and blood and blood-related cell lineages, during the developmental and adult stages of life. ⁶⁷
KLF/SP	(KLF/SP) transcription factors play key roles in critical biological processes, including stem cell maintenance, cell proliferation, embryonic development, tissue differentiation, and metabolism.

	KLFs are transcriptional activators or repressors. ⁶⁸
IRF	Interferon regulatory factors (IRFs) are a family of transcription factors that regulate many aspects of innate and adaptive immune responses. ⁶⁹
HOX	HOX are capable of binding to enhancers through which they either activate or repress hundreds of other genes. Hox transcription factors (TFs) are determinants in the specification of cell fates during development. ⁷⁰
ZNF	ZNF proteins show diverse regulation mechanisms on various downstream genes by recruiting different chromatin modifiers. Some ZNF proteins work as transcriptional repressors by recruiting co-repressors. ZNF proteins are the largest and most diverse transcription factor family in the human genome. ⁷¹
DLX	The DLX family encodes homeodomain transcription factors related to the Drosophila distal-less (DII) gene. The family is involved in a number of developmental features such as jaws and limbs, and craniofacial morphogenesis. ^{72,73}
Sox	SOX TFs govern diverse cellular processes during development, such as maintaining the pluripotency of stem cells, cell proliferation, cell fate decisions/germ layer formation, as well as terminal cell differentiation into tissues and organs. ⁷⁴
Elk, ELF, ETS	The ETS transcription factor family is one of the largest families of TFs. It includes subfamilies such as ELF, ETS, SPI1, etc. Elk1 is a subclass of the ETS subfamily. ^{75,76}

5.3 Results and Discussion

In this study, we have investigated three unsupervised clustering approaches to group DNA regions from ChIP-seq and DNase-seq data based on the number of motifs or binding sites of transcription factors. We have carried out a comparative study on LDA, HDP, and NPLB, which are unsupervised machine learning models. LDA and HDP are frequently used for topic modeling. However, NPLB is a relatively new and promising clustering algorithm capable of identifying new promoters directly from the promotor sequences. First, we identified the appropriate normalization method for the respective dataset. Then, we employed these models to determine regulatory modules from the simulated and ChIP-seq datasets and assessed their performance. Finally, we applied these models to the DNase-seq data and compared their results.

5.3.1 Selecting the Appropriate Normalization Method

The unsupervised machine learning algorithms investigated in this work cluster DNA regions into different groups (modules). To understand their meaning, we visualized the module-motif or module-TF matrix. Each cell in a module-motif and module-TF matrix represents the number of motifs and the number of binding sites of a TF, respectively. An analysis based on the counts is inherently biased, as it highlights only the motifs that are present in abundant quantity. Therefore, it is important to address the variability arising from the difference in the number of regions and motifs. Normalization methods are generally used to remove such artifacts and compare different modules and motifs. However, a proper normalization method is crucial for the interpretation of the results. The choice of the normalization method generally depends on the data and the type of analysis.⁷⁷ We have used z-score normalization in this work. However, there is a slight difference in the way we normalize each module-motif/module-TF matrix. Based on the dataset, we normalize either rows (modules) or columns (motifs/TFs). For ChIP-seq (CTCF), we normalise each row (modules) whereas for ChIP-seq (115 TFs) and DNase-seq, we normalise columns (motifs/TFs). The reason for normalizing rows instead of columns in ChIP-seq (CTCF) was an increase in noise over the expected signal when columns were normalized, as shown in Figure 5.2. In contrast, we observed a loss of expected signals when the rows were normalized in ChIP-seq (115 TFs), as shown in Figure 5.3. For DNase-seq, we chose to normalize columns, as we expect them to contain motifs from multiple regulatory modules similar to ChIP-seq (115 TFs) data.

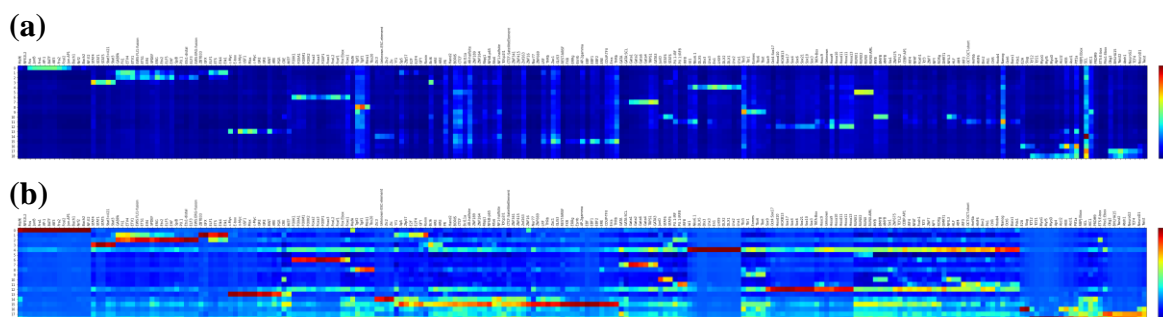


Figure 5.2. The Z-score normalized module-motif matrix of ChIP-seq (CTCF) data obtained from LDA (a) normalized along rows (module) and (b) normalized along columns (motif).

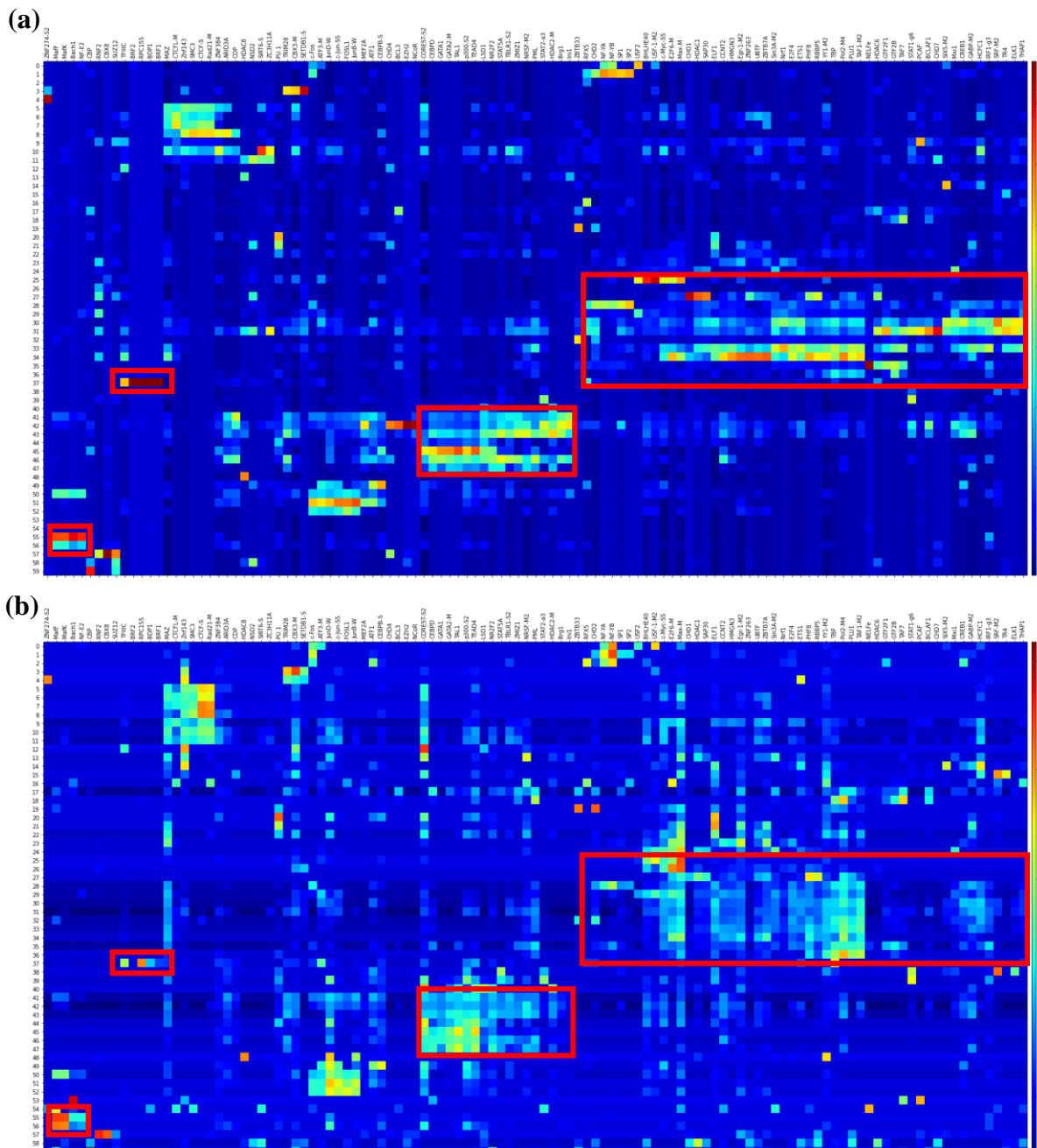


Figure 5.3. The Z-score normalized module-motif matrix of ChIP-seq (115 TFs) data obtained from HDP (a) normalized along columns (TFs) and (b) normalized along rows (modules).

5.3.2 Employing LDA to Cluster Regions from Simulated and ChIP-seq Datasets

Simulated Dataset: The simulated dataset contains 1000 random non-repetitive regions from the human genome, each of length 200 bp. Five motifs from the JASPAR database were planted in these regions. Then, the regions were grouped into three modules such that each module had a different distribution over motifs. The simulated dataset was converted into a bag-of-words representation before training. Then, the LDA was trained using a symmetric prior over document-topic distribution for ten passes. The values of important hyperparameters used during the training are shown in Table 5.2. The true distribution of five motifs in each module

is given in Table 5.1. On the other hand, Table 5.4 shows the distribution of motifs as predicted from LDA. It can be seen that the predicted modules perfectly match the true modules. Thus, LDA correctly identified the modules present in the simulated dataset.

Table 5.4. The module-motif matrix predicted by LDA corresponding to the simulated dataset.

	Motif 1	Motif 2	Motif 3	Motif 4	Motif 5
Module 1	480	480	100	0	0
Module 2	0	0	478	345	0
Module 3	38	0	0	0	42

ChIP-seq (CTCF): The ChIP-seq dataset of the CTCF protein on the human K562 cell line was obtained from the ENCODE project (ENCFF738TKN). The number and the position of known motifs were computed using the HOMER motif analysis tool. The annotated regions were used for constructing a region-motif matrix. The region-motif matrix contains the number of motifs of each TF in each region. Then, the data was converted into a bag-of-words representation before training. Next, the LDA was trained using a symmetric prior over document-topic distribution for 200 passes. The values of important hyperparameters used during the training are given in Table 5.2. After training, the module-motif matrix was obtained by assigning the most probable topic to each region. The module-motif matrix represents the number of motifs of each TF in each module. The heatmap of the module-motif matrix is shown in Figure 5.4. The rows and columns were clustered using hierarchical clustering to group together similar modules and TFs. Below, we list some regulatory modules identified by LDA. The modules are represented using a few important constituent motifs from the module-motif matrix. Please refer to Table 5.3 for the functional importance of the regulatory modules:

- GATA1, GATA2, and related members from the GATA family
- AP-1 factors such as Jun-AP1, JunB, Fos, Fosl2
- n-Myc, Max, MNT, c-Myc, USF1
- SCL transcription factor
- FOX family of transcription factors such as FOXK2, FoxL2, Foxa2, FOXA1, FOXM1
- STAT family of transcription factors such as STAT1, STAT5, Stat3, STAT4
- Hox family of transcription such as Hoxa11, Hoxd11, Hoxa13, Hoxd10.
- DLX family of transcription such as DLX2, DLX1, DLX3

It can be seen that LDA was able to identify some regulatory modules present in the K562 cell line from the ChIP-seq (CTCF) data. We expected a strong signal for the CTCF motif as it was present in almost ~43% (24,605) of the regions. However, the signal for the CTCF motif was relatively low compared to other motifs. In contrast, we observed a strong signal for other motifs even though they were present in a relatively small number of regions. For example, motifs such as AP-1, JunB, Fos, c-Myc, n-Myc, and all motifs from GATA, STAT, and Fox families were present in less than ~20% of the regions; still, we observed a strong signal for these motifs compared to the CTCF motif. Thus, LDA failed to identify commonly occurring

CTCF modules from the ChIP-seq (CTCF) data with strong signal intensity. We also tested different values of eta (0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, and 1), and the results were similar. The module-motif matrix in Figure 5.4 corresponds to eta = 0.01.

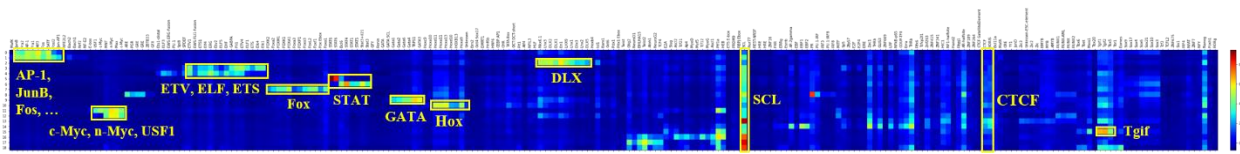


Figure 5.4. The module-motif matrix obtained from LDA (eta = 0.01) corresponding to the ChIP-seq (CTCF) data. Each cell in the heatmap represents the z-score of the motif count (normalized along the rows) of a module (row).

ChIP-seq (115 TFs): This dataset was obtained from the report by Guo and Gifford containing the pooled and merged regions from the ChIP-seq data on 115 TFs in the human K562 cell line.⁴⁸ The file containing the binding site counts of 115 TFs in each region was used for generating the bag-of-words representation. Then, LDA was trained using the symmetric prior over the document-topic distribution for 2000 passes. The values of important hyperparameters used during the training are shown in Table 5.2. After training, a module-TF matrix learned during the inference was obtained. The module-TF matrix represents the number of binding sites of each TF in each module. The heatmap of the module-TF matrix is shown in Figure 5.5. The rows and columns were clustered using hierarchical clustering to group together similar modules and TFs. Below, we list some regulatory modules identified by LDA. The modules are represented using a few important constituent TFs from the module-TF matrix. Please refer to Table 5.3 for the functional importance of the regulatory modules:

- GATA1, GATA2, and TAL1; and the enhancer-binding co-activator p300
- Transcriptional machinery Pol2, TBP, and TAF1
- USF2, USF-1-M2, c-Myc, E2f6
- CTCF, cohesin subunits RAD21 and SMC3, and ZNF143
- Pol3 transcriptional machinery
- AP-1 factors such as JunD, c-Jun, FOSL1, JunB
- MafF, MafK, Bach1, NF-E2

LDA was able to identify some commonly occurring regulatory modules in the K562 cell line, such as master regulators, CTCF/cohesin subunits, Pol2 machinery etc. Most of the regulatory modules discovered by LDA match with the modules reported by Guo and Gifford.⁴⁸ Thus, LDA successfully identified some commonly occurring regulatory modules from ChIP-seq (115 TFs) data.

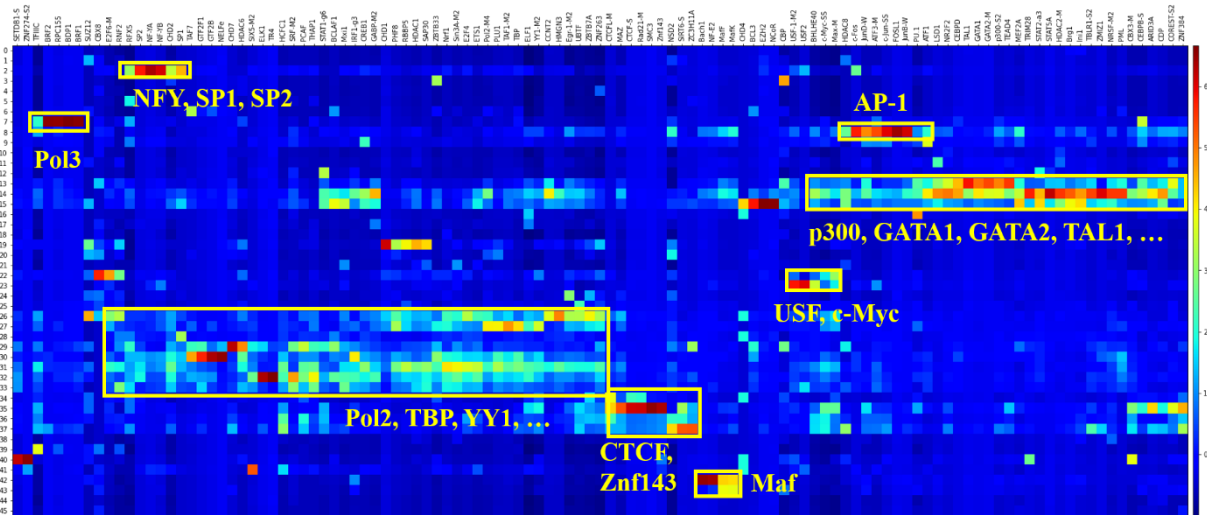


Figure 5.5. The module-TF matrix obtained from LDA corresponding to the ChIP-seq (115 TFs) data. Each cell in the heatmap represents the z-score of the TF binding site count (normalized along the columns) of a TF (column) in a module (row).

5.3.3 Employing HDP to Cluster Regions from Simulated and ChIP-seq Datasets

Simulated Dataset: We applied the hierarchical Dirichlet process (HDP) to cluster regions from the simulated dataset. HDP is a nonparametric Bayesian model used for topic modeling. It uses a Dirichlet process to represent each group of data that shares a base distribution. The base distribution is itself drawn from a Dirichlet process. The simulated dataset was converted into a bag-of-words representation before training. Then, HDP was trained for a maximum of 20,000 iterations. The values of important hyperparameters used during the training are shown in Table 5.2. The advantage of HDP over LDA is that HDP automatically discovers the number of topics present in the corpus. However, LDA requires the specification of the number of topics during training. After the training, we obtained a module-motif matrix, with cells representing motif counts in the modules. Table 5.5 shows the module-motif matrix obtained from the HDP. We can see that the predicted modules match reasonably well with the true modules. However, the two regions from module 1 had been misplaced in module 3. We also tested different values of hyperparameters listed in Table 5.2 but failed to obtain a hundred percent accurate result. Nevertheless, we believe a hundred percent accurate result could be obtained from extensive testing of different values hyperparameters. The module-motif matrix shown in Table 5.5 is most the accurate result obtained from HDP in this study which corresponds to $\eta = 0.1$, $\gamma_a = 5$, $\gamma_b = 0.1$, $\alpha_a = 0.1$, and $\alpha_b = 1.653$. Thus, HDP was able to identify the modules present in the simulated dataset with high accuracy.

Table 5.5. The module-motif matrix predicted by HDP ($\eta = 0.1$, $\gamma_a = 5$, $\gamma_b = 0.1$, $\alpha_a = 0.1$, and $\alpha_b = 1.653$) corresponding to the simulated dataset.

Hdp_Pred	Motif 1	Motif 2	Motif 3	Motif 4	Motif 5
Module 1	478	480	100	0	
Module 2	0	0	478	345	0
Module 3	40	0	0	0	42

3.3.2 ChIP-seq (CTCF): We applied HDP to cluster the regions obtained from ChIP-seq (CTCF) data. A region-motif matrix representing motif counts was computed using the HOMER motif analysis tool. The region-motif matrix was converted into a bag-of-words representation before training. Then, HDP was trained for a maximum of 2000 iterations. The values of important hyperparameters used during the training are given in Table 5.2. The module-motif matrix containing the number of each motif in each module was generated after training. Hierarchical clustering was further applied to cluster rows and columns of the module-motif matrix. The final module-motif matrix is shown in Figure 5.6. We list below a few regulatory modules identified by HDP. The modules are represented using some important constituent motifs from the module-motif matrix; please refer to Table 5.3 for the functional importance of the regulatory modules:

- Elk, ELF, ETS
- Foxa3, FOX, FOXA1, and related TFs
- Hox family
- TRPS1, GATA
- SCL
- AP-1 factors such as Jun-AP1, Fos, Fosl2, JunB, Fos
- c-Myc, MNT, Max, n-Myc, USF1
- STAT
- RUNX
- Sox

We can see that HDP was able to identify some regulatory modules present in the K562 cell line. Modules discovered by HDP were very similar to those discovered by LDA. However, the intensity of the signal for many motifs was less than the LDA. We also noticed that some modules clustered together in LDA had been split into multiple modules by the HDP, such as Elk, ETS, ETV; STAT; Fox; Hox; Tgif; and Myc, USF, which could be attributed to a large number of modules discovered by HDP (number of modules = 66) compared to LDA (number of modules = 19). Similar to LDA, HDP also failed to identify the modules containing CTCF motifs with strong signal intensity from the ChIP-seq data targeted for the CTCF protein. We also tested different values of η (0.01, 0.05, 0.1, 0.5, and 1), and the results were similar. The module-motif matrix in Figure 5.6 corresponds to $\eta = 0.1$.

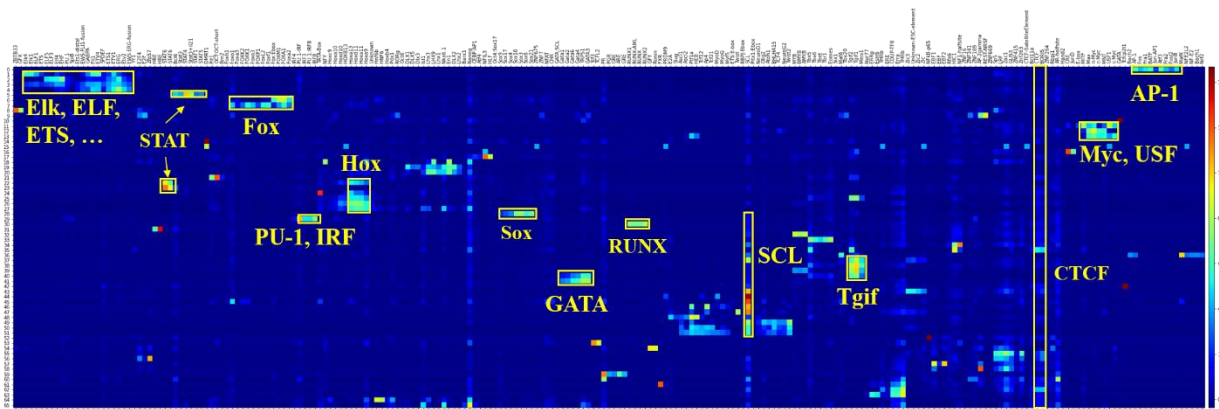


Figure 5.6. The module-motif matrix obtained from HDP ($\eta = 0.1$) corresponding to the ChIP-seq (CTCF) data. Each cell in the heatmap represents the z-score of the motif count (normalized along the rows) of a module (row).

ChIP-seq (115 TFs): The ChIP-seq data of 115 different TFs in the human K562 cell line was obtained from the report by Guo and Gifford.⁴⁸ The binding site counts of the TFs in each region were converted into a bag-of-words representation before training. Then, HDP was trained for a maximum of 2000 iterations. The values of important hyperparameters used during the training are given in Table 5.2. After training, the module-TF matrix containing binding site counts of each TF in each module was obtained from HDP. Rows and columns of the module-TF matrix were clustered using hierarchical clustering. The final module-TF matrix is shown in Figure 5.7. We list below a few regulatory modules identified by HDP. The modules are represented using some important constituent TFs from the module-TFs matrix; please refer to Table 5.3 for the functional importance of the regulatory modules:

- Master regulators GATA1, GATA2, and TAL1; and the enhancer-binding co-activator p300
- Transcriptional machinery Pol2, TBP, and TAF1
- USF2, USF-1-M2, c-Myc, E2f
- CTCF, cohesin subunits RAD21 and SMC3, and ZNF143
- Pol3 transcriptional machinery
- AP-1 factors such as JunD, c-Jun, FOSL1, JunB
- MafF, MafK, Bach1, NF-E2

The modules discovered by HDP were easy to interpret and reveal functionally similar groups of co-binding TFs. The identified modules from K562 cells capture the known set of TFs that generally appear to interact with each other. The regulatory modules discovered by HDP match very well with the modules reported by Guo and Gifford.⁴⁸ Thus, we have validated our HDP approach by reproducing the reported results.

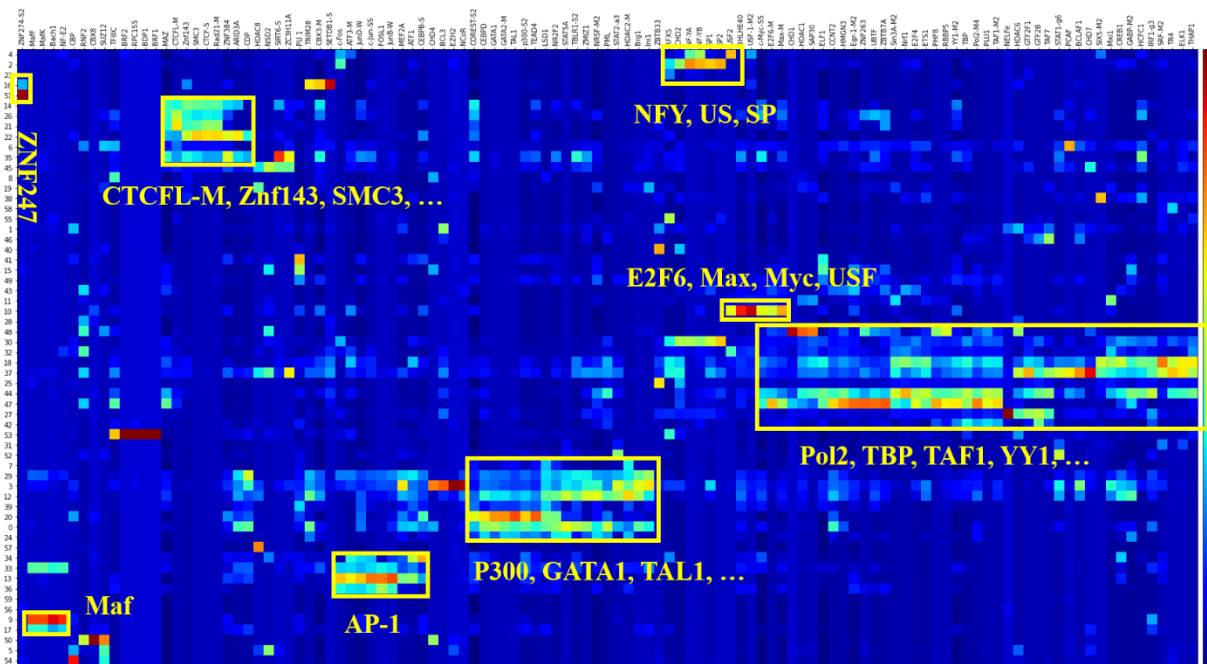


Figure 5.7. Module-TF matrix obtained from HDP corresponding to the ChIP-seq (115 TFs) data. Each cell in the heatmap represents the z-score of the TF binding site count (normalized along the columns) of a TF (column) in a module (row).

5.3.4 Employing NPLB to Cluster Regions from Simulated and ChIP-seq Datasets

Simulated Dataset: No Promoter Left Behind (NPLB) is an efficient, organism-independent unsupervised learning method for *de novo* promoter and module discovery. NPLB discovers known and unknown promoter elements from DNA sequences and clusters them into functionally similar groups. Although NPLB was designed to work with the DNA sequences, we have shown that it could also be used for clustering regions based on the presence or absence of the motifs. As NPLB requires a fasta file, we generated one from the region-motif matrix before training. “A” and “T” bases in the fasta file represent the presence and the absence of motifs in a sequence, respectively. Then, NPLB was trained on the fasta file with the default hyperparameters. A file containing assignments of the regions to modules was generated after training. We computed the module-motif matrix containing the motif counts from the region-module assignment file. Table 5.6 shows the module-motif matrix obtained from NPLB. We observed that the predicted modules perfectly match the true modules (Table 5.1). Thus, NPLB successfully identified the modules present in the simulated dataset.

Table 5.6. Module-motif matrix predicted by NPLB corresponding to the simulated dataset.

True	Motif 1	Motif 2	Motif 3	Motif 4	Motif 5
Module 1	480	480	100	0	0
Module 2	0	0	478	345	0
Module 3	38	0	0	0	42

ChIP-seq (CTCF): NPLB was employed to cluster regions obtained from ChIP-seq data on the K562 cell line enriched in CTCF protein. The number and the position of known motifs were computed using the HOMER motif analysis tool. The region-motif matrix containing the

motif count of TFs was obtained from the output files generated by HOMER. Next, we generated a fasta file from the region-motif matrix for training. “A” and “T” bases in the fasta file represent the presence and the absence of motifs in a sequence, respectively. Then, NPLB was trained on the region-motif matrix with the default hyperparameters. After training, the module-motif matrix containing the number of each motif in each module was obtained from the region-module assignment file. Modules and motifs were further grouped using hierarchical clustering. The final module-motif matrix is shown in Figure 5.8. We list below a few regulatory modules identified by NPLB. The modules are represented using some important constituent motifs from the module-motif matrix; please refer to Table 5.3 for the functional importance of the regulatory modules:

- ELF, ETV, ETS
- c-Myc, MNT, USF, Max, c-Myc, nMyc
- CTCF, BORIS, THRb, Zac1
- SCL
- RUNX, RUNX1, RUNX2, RUNX-AML
- Stat3, STAT4, STAT1, STAT5
- FOXA1, FOXM1, Foxo1, Foxa2, Foxa3, FOXP1, FOXK1, FoxL2, Foxf1
- Gata1, Gata2, Gata6, Gata4, TRPS1, GATA3
- AP-1, Fra2, Fosl2, Jun-AP1, JunB, Fra1, BATF, Fos, Atf3

NPLB identified some regulatory modules commonly found in the K562 cell line. The regulatory modules constitute the motifs of interacting co-binding TFs. Particularly, NPLB was able to discover modules containing CTCF, the target protein in the ChIP-seq data. Thus, NPLB successfully revealed the expected regulatory modules of the known co-binding and interacting TFs from ChIP-seq (CTCF) data in the K562 cell line.

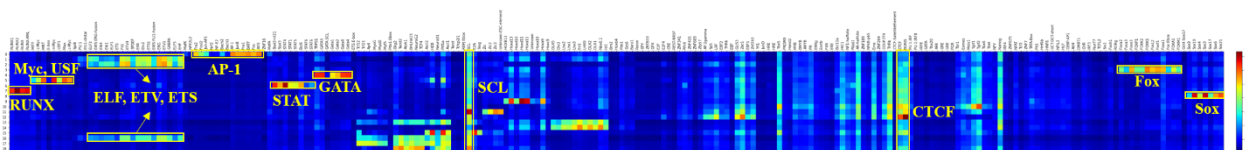


Figure 5.8. The module-motif matrix obtained from NPLB corresponding to the ChIP-seq (CTCF) data. Each cell in the heatmap represents the z-score of the motif count (normalized along the rows) of a module (row).

ChIP-seq (115 TFs): We employed NPLB to cluster the regions obtained from the ChIP-seq data of 115 TFs. A fasta file was generated from the region-TF matrix for training. “A” and “T” bases in the fasta file represent the presence and the absence of TF in a sequence, respectively. Then, NPLB was trained for a maximum of 49 modules using 3-fold cross-validation. After training, the module-TF matrix containing the binding site count of each TF in each module was obtained from the region-module assignment file. Modules and TFs were further grouped using hierarchical clustering. The final module-TF matrix is shown in Figure

5.9. We list below a few regulatory modules identified by NPLB. The modules are represented using some important constituent TFs from the module-TF matrix; please refer to Table 5.3 for the functional importance of the regulatory modules:

- Master regulators GATA1, GATA2, and TAL1; and the enhancer-binding co-activator p300
- Transcriptional machinery Pol2, TBP, and TAF1
- USF2, USF-1-M2, c-Myc, E2f6
- CTCF, cohesin subunits RAD21 and SMC3, and ZNF143
- Pol3 transcriptional machinery
- AP-1 factors such as JunD, c-Jun, FOSL1, JunB
- MafF, MafK, Bach1, NF-E2

The modules discovered by NPLB were very similar to those discovered by HDP. Modules were easy to interpret and revealed functionally similar groups of co-binding TFs. The modules discovered by NPLB capture known sets of interacting and co-binding TFs in the K562 cell line. Furthermore, NPLB discovered many of the regulatory modules reported by Guo and Gifford.⁴⁸

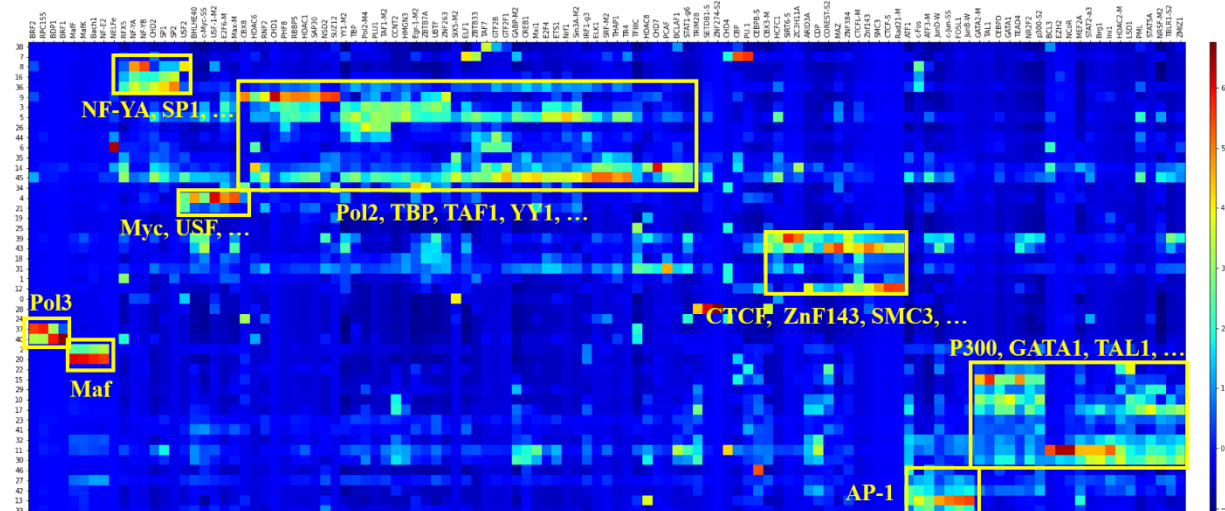


Figure 5.9. The module-TF matrix obtained from NPLB corresponding to the ChIP-seq (115 TFs) data. Each cell in the heatmap represents the z-score of the TF binding site count (normalized along the columns) of a TF (column) in a module (row).

5.3.5 Applying LDA, HDP, and NPLB to Cluster Regions from the DNase-seq Dataset

Next, we applied LDA, HDP, and NPLB to identify regulatory modules present in the DNase-seq data. The DNase-seq data of the human K562 cell line was obtained from the ENCODE project (ENCFF621ZJY). It contained the coordinates of 378,491 DNA regions preferentially cleaved by DNase I. First, the number and the position of known motifs were computed using the HOMER motif analysis tool. The region-motif matrix containing the motif counts of TFs was obtained from the output files generated by HOMER. The region-motif matrix was

converted into the bag-of-words representation for LDA and HDP. On the other hand, a fasta file was generated from the region-motif matrix for NPLB. “A” and “T” bases in the fasta file represent the presence and the absence of motifs in a sequence, respectively. Then, LDA, HDP, and NPLB were trained to identify the regulatory modules from the DNase-seq data. The values of important hyperparameters for each model used during the training are shown in Table 5.2. Rows and columns of the module-motif matrix obtained after training were further grouped using hierarchical clustering. Figure 5.10 - Figure 5.12 show the module-motif matrix obtained from LDA, HDP, and NPLB, respectively.

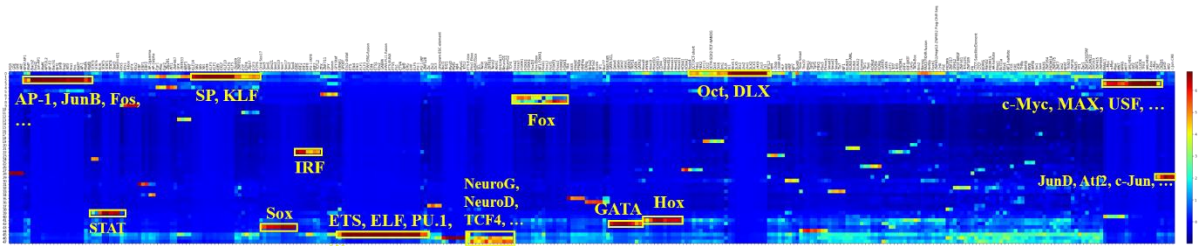


Figure 5.10. The module-motif matrix obtained from LDA corresponding to the DNase-seq data. Each cell in the heatmap represents the z-score of the motif count (normalized along the columns) of a motif (column) in a module (row).

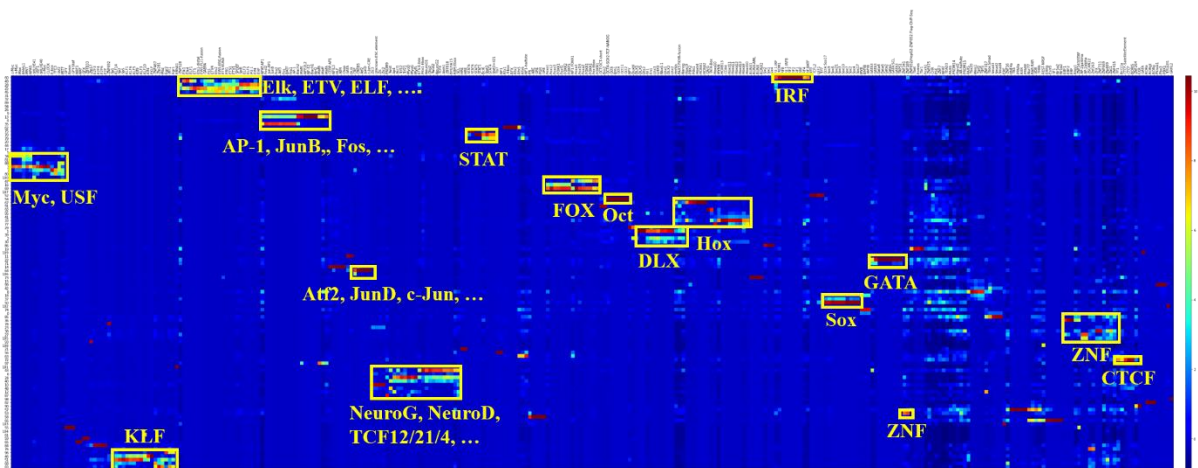


Figure 5.11. The module-motif matrix obtained from HDP corresponding to the DNase-seq data. Each cell in the heatmap represents the z-score of the motif count (normalized along the columns) of a motif (column) in a module (row).

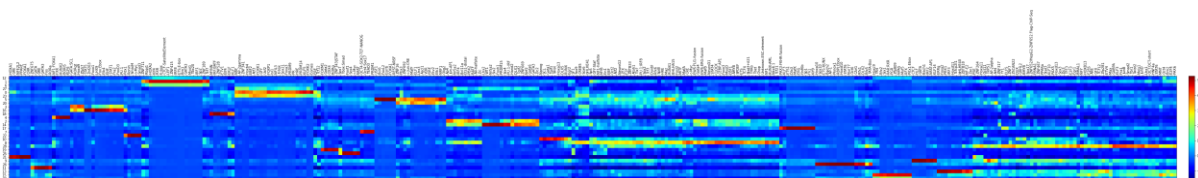


Figure 5.12. The module-motif matrix obtained from NPLB corresponding to the DNase-seq data. Each cell in the heatmap represents the z-score of the motif count (normalized along the columns) of a motif (column) in a module (row).

Some regulatory modules identified by LDA and HDP from the DNase-seq data are listed below. The modules are represented using some important constituent motifs from the module-motif matrix; please refer to Table 5.3 for the functional importance of the regulatory modules:

- AP-1, JunB, Fos
- SP, KLF
- c-Myc, n-Myc USF, MAX
- ETS, ELF, Elk, PU.1
- JunD, Atf2, c-Jun
- FOX
- Hox
- GATA
- STAT
- IRF
- Sox
- Oct
- DLX

We observed that modules discovered by HDP were very similar to those identified by LDA. However, modules containing CTCF and various ZNF motifs were only identified by HDP. We also observed that a module containing Oct and DLX motifs in LDA had been separated into two distinct sub-modules in HDP, suggesting the possibility of different functional roles. Furthermore, it was observed that some modules in LDA have been resolved into multiple modules in HDP. Another difference was observed between the modules containing NeuroG/D motifs. In LDA, NeuroG/D modules were accompanied by only the TCF4 motifs, whereas in HDP, NeuroG/D was accompanied by TCF12/21/4 motifs. The number of modules discovered by HDP was almost twice that of LDA, making analysis cumbersome. HDP produced 110 modules compared to 49 in LDA. However, more modules also helped us identify functionally distinct modules. Thus, LDA and HDP were able to identify some commonly found regulatory modules in the K562 cell line from Dnase-seq data. Unfortunately, due to a large number of sequences, the NPLB algorithm could not converge in a given time. Therefore, there was no structure to the modules identified by NPLB. Nevertheless, given its performance on the ChIP-seq datasets, we believe NPLB could discover regulatory modules from DNase-seq data. However, further improvement in the speed of NPLB is required, which is under investigation. The results are summarized in Table 5.7.

Table 5.7. Summary of results. The tick (✓) represents the successful identification of expected regulatory modules by the model, whereas the cross (✗) represents a failure of the same.

Model	Dataset			
	Simulated	ChIP-seq (CTCF)	ChIP-seq (115 TFs)	DNase-seq
LDA	✓	✗	✓	✓
HDP	✓	✗	✓	✓
NPLB	✓	✓	✓	✗

5.4 Conclusions

We employed three unsupervised learning models (LDA, HDP, and NPLB) to cluster the regions obtained from ChIP-seq and DNase-seq. In most cases, the discovered clusters (modules) represent a group of commonly interacting co-binding TFs. The co-occurrence of regulatory modules in the datasets suggests modular hierarchy in the combinatorial binding of TFs in regulatory regions. The results corresponding to each model could be summarised as follows:

- LDA accurately identified the distribution of motifs in the simulated dataset. LDA discovered some regulatory modules commonly found in the K562 cell line from the ChIP-seq (CTCF) data. However, it failed to identify the modules containing the CTCF motif with strong signal intensity from the ChIP-seq (CTCF) data. LDA was able to discover many regulatory modules from the ChIP-seq (115 TF) dataset.
- HDP was able to identify the distribution of motifs in the simulated dataset with high accuracy. It discovered some regulatory modules commonly found in the K562 cell line but failed to identify the modules containing CTCF motif with strong signal intensity from the ChIP-seq targeted for CTCF protein. In contrast to LDA, some modules discovered by HDP were split into multiple modules, possibly due to a large number of modules. Its applicability to discover regulatory modules from ChIP-seq (115 TF) is shown by Guo and Gifford and verified by us.
- NPLB correctly identified the distribution of motifs in the simulated data. NPLB successfully discovered the modules containing CTCF along with other commonly occurring regulatory modules from the ChIP-seq (CTCF) data. It also discovered many regulatory modules from the ChIP-seq (115 TFs) dataset.
- Finally, we employed LDA, HDP, and NPLB to identify regulatory modules from DNase-seq data. LDA and HDP identified very similar modules, including commonly found regulatory modules in the K562 cell line. However, modules containing CTCF and ZNF motifs were only observed in HDP. Furthermore, HDP offered more resolution in modules than LDA. Unfortunately, due to a large number of sequences, NPLB could not converge in a given time, and therefore, it failed to identify the regulatory modules in DNase-seq. Nevertheless, given its performance on the ChIP-seq datasets, we believe NPLB can discover regulatory modules in DNase-seq data. Further improvement in the speed of NPLB is required, which is under investigation.

Thus, we have shown that regulatory and functionally similar modules could be discovered using the topics models from ChIP-seq and DNase-seq data. NPLB identified expected modules in the simulated and ChIP-seq datasets when LDA and HPD failed on at least one of the datasets. NPLB was designed to extract promoter elements directly from promoter sequences without prior information. For this work, we modified the typical workflow of NPLB to handle sequences represented in the form of motifs. This robust performance of NPLB could be attributed to the ability of NPLB to ignore the noise present in the data.

5.5 References

- (1) Gene Expression | Learn Science at Scitable <https://www.nature.com/scitable/topicpage/gene-expression-14121669/> (accessed Mar 8, 2022).
- (2) Maston, G. A.; Evans, S. K.; Green, M. R. Transcriptional Regulatory Elements in the Human Genome. *Annu. Rev. Genomics Hum. Genet.* **2006**, *7*, 29–59. <https://doi.org/10.1146/annurev.genom.7.080505.115623>.
- (3) Chatterjee, S.; Ahituv, N. Gene Regulatory Elements, Major Drivers of Human Disease. <https://doi.org/10.1146/annurev-genom-091416-035537> **2017**, *18*, 45–63. <https://doi.org/10.1146/ANNUREV-GENOM-091416-035537>.
- (4) Gaszner, M.; Felsenfeld, G. Insulators: Exploiting Transcriptional and Epigenetic Mechanisms. *Nat. Rev. Genet.* **2006**, *7* (9), 703–713. <https://doi.org/10.1038/nrg1925>.
- (5) Segert, J. A.; Gisselbrecht, S. S.; Bulyk, M. L. Transcriptional Silencers: Driving Gene Expression with the Brakes On. *Trends Genet.* **2021**, *37* (6), 514–527. <https://doi.org/10.1016/J.TIG.2021.02.002>.
- (6) Genetic Disorders <https://www.genome.gov/For-Patients-and-Families/Genetic-Disorders> (accessed Mar 9, 2022).
- (7) Whitsett, J. A.; Wert, S. E.; Trapnell, B. C. Genetic Disorders Influencing Lung Formation and Function at Birth. *Hum. Mol. Genet.* **2004**, *13* (suppl_2), R207–R215. <https://doi.org/10.1093/HMG/DDH252>.
- (8) Shamseldin, H. E.; Tulbah, M.; Kurdi, W.; Nemer, M.; Alsahan, N.; Al Mardawi, E.; Khalifa, O.; Hashem, A.; Kurdi, A.; Babay, Z.; Bubshait, D. K.; Ibrahim, N.; Abdulwahab, F.; Rahbeeni, Z.; Hashem, M.; Alkuraya, F. S. Identification of Embryonic Lethal Genes in Humans by Autozygosity Mapping and Exome Sequencing in Consanguineous Families. *Genome Biol.* **2015**, *16* (1), 1–7. <https://doi.org/10.1186/S13059-015-0681-6/FIGURES/2>.
- (9) Hurd, E. A.; Capers, P. L.; Blauwkamp, M. N.; Adams, M. E.; Raphael, Y.; Poucher, H. K.; Martin, D. M. Loss of Chd7 Function in Gene-Trapped Reporter Mice Is Embryonic Lethal and Associated with Severe Defects in Multiple Developing Tissues. *Mamm. Genome* **2007**, *18* (2), 94–104. <https://doi.org/10.1007/S00335-006-0107-6>.
- (10) Scacheri, C. A.; Scacheri, P. C. Mutations in the Non-Coding Genome. *Curr. Opin. Pediatr.* **2015**, *27* (6), 659. <https://doi.org/10.1097/MOP.0000000000000283>.
- (11) Perenthaler, E.; Yousefi, S.; Niggli, E.; Barakat, T. S. Beyond the Exome: The Non-Coding Genome and Enhancers in Neurodevelopmental Disorders and Malformations of Cortical Development. *Front. Cell. Neurosci.* **2019**, *13*, 352. <https://doi.org/10.3389/FNCEL.2019.00352/BIBTEX>.
- (12) Melton, C.; Reuter, J. A.; Spacek, D. V.; Snyder, M. Recurrent Somatic Mutations in Regulatory Regions of Human Cancer Genomes. *Nat. Genet.* **2015**, *47* (7), 710. <https://doi.org/10.1038/NG.3332>.
- (13) Rice, G.; Rebeiz, M. Evolution: How Many Phenotypes Do Regulatory Mutations

- Affect? *Curr. Biol.* **2019**, *29* (1), R21–R23.
<https://doi.org/10.1016/J.CUB.2018.11.027>.
- (14) Scacheri, C. A.; Scacheri, P. C. Mutations in the Noncoding Genome. *Curr. Opin. Pediatr.* **2015**, *27* (6), 659–664. <https://doi.org/10.1097/MOP.0000000000000283>.
 - (15) Gilissen, C.; Hoischen, A.; Brunner, H. G.; Veltman, J. A. Disease Gene Identification Strategies for Exome Sequencing. *Eur. J. Hum. Genet.* **2012**, *20* (5), 490. <https://doi.org/10.1038/EJHG.2011.258>.
 - (16) Goodwin, S.; McPherson, J. D.; McCombie, W. R. Coming of Age: Ten Years of next-Generation Sequencing Technologies. *Nat. Rev. Genet.* *2016 176* **2016**, *17* (6), 333–351. <https://doi.org/10.1038/nrg.2016.49>.
 - (17) Barzon, L.; Lavezzo, E.; Militello, V.; Toppo, S.; Palù, G. Applications of Next-Generation Sequencing Technologies to Diagnostic Virology. *Int. J. Mol. Sci.* **2011**, *12* (11), 7861. <https://doi.org/10.3390/IJMS12117861>.
 - (18) Park, P. J. ChIP–Seq: Advantages and Challenges of a Maturing Technology. *Nat. Rev. Genet.* *2009 1010* **2009**, *10* (10), 669–680. <https://doi.org/10.1038/nrg2641>.
 - (19) Raha, D.; Hong, M.; Snyder, M. ChIP-Seq: A Method for Global Identification of Regulatory Elements in the Genome. *Curr. Protoc. Mol. Biol.* **2010**, *Chapter 21* (SUPPL. 91). <https://doi.org/10.1002/0471142727.MB2119S91>.
 - (20) Nakato, R.; Sakata, T. Methods for ChIP-Seq Analysis: A Practical Workflow and Advanced Applications. *Methods* **2021**, *187*, 44–53. <https://doi.org/10.1016/J.YMETH.2020.03.005>.
 - (21) Nakato, R.; Shirahige, K. Recent Advances in ChIP-Seq Analysis: From Quality Management to Whole-Genome Annotation. *Brief. Bioinform.* **2017**, *18* (2), 279–290. <https://doi.org/10.1093/BIB/BBW023>.
 - (22) DNA Packaging: Nucleosomes and Chromatin | Learn Science at Scitable <https://www.nature.com/scitable/topicpage/dna-packaging-nucleosomes-and-chromatin-310/> (accessed Mar 9, 2022).
 - (23) Coux, R. X.; Owens, N. D. L.; Navarro, P. Chromatin Accessibility and Transcription Factor Binding through the Perspective of Mitosis. *Transcription* **2020**, *11* (5), 236. <https://doi.org/10.1080/21541264.2020.1825907>.
 - (24) Guertin, M. J.; Lis, J. T. Mechanisms by Which Transcription Factors Gain Access to Target Sequence Elements in Chromatin. *Curr. Opin. Genet. Dev.* **2013**, *23* (2), 116. <https://doi.org/10.1016/J.GDE.2012.11.008>.
 - (25) Song, L.; Crawford, G. E. DNase-Seq: A High-Resolution Technique for Mapping Active Gene Regulatory Elements across the Genome from Mammalian Cells. *Cold Spring Harb. Protoc.* **2010**, *2010* (2), pdb.prot5384. <https://doi.org/10.1101/PDB.PROT5384>.
 - (26) Liu, Y.; Fu, L.; Kaufmann, K.; Chen, D.; Chen, M. A Practical Guide for DNase-Seq Data Analysis: From Data Management to Common Applications. *Brief. Bioinform.* **2019**, *20* (5), 1865–1877. <https://doi.org/10.1093/BIB/BBY057>.
 - (27) Lee, C.; Wang, K.; Qin, T.; Sartor, M. A. Testing Proximity of Genomic Regions to Transcription Start Sites and Enhancers Complements Gene Set Enrichment Testing.

- Front. Genet.* **2020**, *11*, 199. <https://doi.org/10.3389/FGENE.2020.00199/BIBTEX>.
- (28) Sharmin, M.; Bravo, H. C.; Hannenhalli, S. Heterogeneity of Transcription Factor Binding Specificity Models within and across Cell Lines. *Genome Res.* **2016**, *26* (8), 1110–1123. <https://doi.org/10.1101/GR.199166.115/-/DC1>.
- (29) What is noncoding DNA?: MedlinePlus Genetics <https://medlineplus.gov/genetics/understanding/basics/noncodingdna/> (accessed Feb 17, 2022).
- (30) Stanojevic, D.; Small, S.; Levine, M. Regulation of a Segmentation Stripe by Overlapping Activators and Repressors in the Drosophila Embryo. *Science* (80-.). **1991**, *254* (5036), 1385–1387. <https://doi.org/10.1126/SCIENCE.1683715>.
- (31) Georges, A. B.; Benayoun, B. A.; Caburet, S.; Veitia, R. A. Generic Binding Sites, Generic DNA-Binding Domains: Where Does Specific Promoter Recognition Come From? *FASEB J.* **2010**, *24* (2), 346–356. <https://doi.org/10.1096/FJ.09-142117>.
- (32) Weingarten-Gabbay, S.; Segal, E. The Grammar of Transcriptional Regulation. *Hum. Genet.* **2014**, *133* (6), 701–711. <https://doi.org/10.1007/S00439-013-1413-1>.
- (33) Gerstein, M. B.; Kundaje, A.; Hariharan, M.; Landt, S. G.; Yan, K. K.; Cheng, C.; Mu, X. J.; Khurana, E.; Rozowsky, J.; Alexander, R.; Min, R.; Alves, P.; Abyzov, A.; Addleman, N.; Bhardwaj, N.; Boyle, A. P.; Cayting, P.; Charos, A.; Chen, D. Z.; Cheng, Y.; Clarke, D.; Eastman, C.; Euskirchen, G.; Fietze, S.; Fu, Y.; Gertz, J.; Grubert, F.; Harmanci, A.; Jain, P.; Kasowski, M.; Lacroute, P.; Leng, J.; Lian, J.; Monahan, H.; O'geen, H.; Ouyang, Z.; Partridge, E. C.; Patacsil, D.; Pauli, F.; Raha, D.; Ramirez, L.; Reddy, T. E.; Reed, B.; Shi, M.; Slifer, T.; Wang, J.; Wu, L.; Yang, X.; Yip, K. Y.; Zilberman-Schapira, G.; Batzoglou, S.; Sidow, A.; Farnham, P. J.; Myers, R. M.; Weissman, S. M.; Snyder, M. Architecture of the Human Regulatory Network Derived from ENCODE Data. *Nat.* **2012**, *489* (7414), 91–100. <https://doi.org/10.1038/nature11245>.
- (34) Yip, K. Y.; Cheng, C.; Bhardwaj, N.; Brown, J. B.; Leng, J.; Kundaje, A.; Rozowsky, J.; Birney, E.; Bickel, P.; Snyder, M.; Gerstein, M. Classification of Human Genomic Regions Based on Experimentally Determined Binding Sites of More than 100 Transcription-Related Factors. *Genome Biol.* **2012**, *13* (9), 1–22. <https://doi.org/10.1186/GB-2012-13-9-R48/TABLES/5>.
- (35) Roy, S.; Ernst, J.; Kharchenko, P. V.; Kheradpour, P.; Negre, N.; Eaton, M. L.; Landolin, J. M.; Bristow, C. A.; Ma, L.; Lin, M. F.; Washietl, S.; Arshinoff, B. I.; Ay, F.; Meyer, P. E.; Robine, N.; Washington, N. L.; Di Stefano, L.; Berezhikov, E.; Brown, C. D.; Candéias, R.; Carlson, J. W.; Carr, A.; Jungreis, I.; Marbach, D.; Sealfon, R.; Tolstorukov, M. Y.; Will, S.; Alekseyenko, A. A.; Artieri, C.; Booth, B. W.; Brooks, A. N.; Dai, Q.; Davis, C. A.; Duff, M. O.; Feng, X.; Gorchakov, A. A.; Gu, T.; Henikoff, J. G.; Kapranov, P.; Li, R.; MacAlpine, H. K.; Malone, J.; Minoda, A.; Nordman, J.; Okamura, K.; Perry, M.; Powell, S. K.; Riddle, N. C.; Sakai, A.; Samsonova, A. A.; Sandler, J. E.; Schwartz, Y. B.; Sher, N.; Spokony, R.; Sturgill, D.; van Baren, M.; Wan, K. H.; Yang, L.; Yu, C.; Feingold, E.; Good, P.; Guyer, M.; Lowdon, R.; Ahmad, K.; Andrews, J.; Berger, B.; Brenner, S. E.; Brent, M. R.; Cherkas, L.; Elgin, S. C. R.; Gingeras, T. R.; Grossman, R.; Hoskins, R. A.; Kaufman, T. C.; Kent, W.; Kuroda, M. I.; Orr-Weaver, T.; Perrimon, N.; Pirrotta, V.; Posakony,

- J. W.; Ren, B.; Russell, S.; Cherbas, P.; Graveley, B. R.; Lewis, S.; Micklem, G.; Oliver, B.; Park, P. J.; Celniker, S. E.; Henikoff, S.; Karpen, G. H.; Lai, E. C.; MacAlpine, D. M.; Stein, L. D.; White, K. P.; Kellis, M.; Booth, B.; Comstock, C. L. G.; Dobin, A.; Drenkow, J.; Dudoit, S.; Dumais, J.; Fagegaltier, D.; Ghosh, S.; Hansen, K. D.; Jha, S.; Langton, L.; Lin, W.; Miller, D.; Tenney, A. E.; Wang, H.; Willingham, A. T.; Zaleski, C.; Zhang, D.; Acevedo, D.; Bishop, E. P.; Gadel, S. E.; Jung, Y. L.; Kennedy, C. D.; Lee, O. K.; Linder-Basso, D.; Marchetti, S. E.; Shanower, G.; Nègre, N.; Grossman, R. L.; Auburn, R.; Bellen, H. J.; Chen, J.; Domanus, M. H.; Hanley, D.; Heinz, E.; Li, Z.; Meyer, F.; Miller, S. W.; Morrison, C. A.; Scheftner, D. A.; Senderowicz, L.; Shah, P. K.; Suchy, S.; Tian, F.; Venken, K. J. T.; White, R.; Wilkening, J.; Zieba, J.; Nordman, J. T.; Orr-Weaver, T. L.; DeNapoli, L. C.; Ding, Q.; Eng, T.; Kashevsky, H.; Li, S.; Prinz, J. A.; Hannon, G. J.; Hirst, M.; Marra, M.; Rooks, M.; Zhao, Y.; Bryson, T. D.; Perry, M. D.; Kent, W. J.; Lewis, S. E.; Barber, G.; Chateigner, A.; Clawson, H.; Contrino, S.; Guillier, F.; Hinrichs, A. S.; Kephart, E. T.; Lloyd, P.; Lyne, R.; McKay, S.; Moore, R. A.; Mungall, C.; Rutherford, K. M.; Ruzanov, P.; Smith, R.; Stinson, E. O.; Zha, Z.; Artieri, C. G.; Malone, J. H.; Jiang, L.; Mattiuzzo, N.; Feingold, E. A.; Good, P. J.; Guyer, M. S.; Lowdon, R. F. Identification of Functional Elements and Regulatory Circuits by *Drosophila* ModENCODE. *Science* (80-.). **2010**, *330* (6012), 1787–1797. https://doi.org/10.1126/SCIENCE.1198374/SUPPL_FILE/CONSORTIUM.SOM.PDF.
- (36) Kagey, M. H.; Newman, J. J.; Bilodeau, S.; Zhan, Y.; Orlando, D. A.; Van Berkum, N. L.; Ebmeier, C. C.; Goossens, J.; Rahl, P. B.; Levine, S. S.; Taatjes, D. J.; Dekker, J.; Young, R. A. Mediator and Cohesin Connect Gene Expression and Chromatin Architecture. *Nat. 2010 4677314* **2010**, *467* (7314), 430–435. <https://doi.org/10.1038/nature09380>.
- (37) Dunham, I.; Kundaje, A.; Aldred, S. F.; Collins, P. J.; Davis, C. A.; Doyle, F.; Epstein, C. B.; Frietze, S.; Harrow, J.; Kaul, R.; Khatun, J.; Lajoie, B. R.; Landt, S. G.; Lee, B. K.; Pauli, F.; Rosenbloom, K. R.; Sabo, P.; Safi, A.; Sanyal, A.; Shores, N.; Simon, J. M.; Song, L.; Trinklein, N. D.; Altshuler, R. C.; Birney, E.; Brown, J. B.; Cheng, C.; Djebali, S.; Dong, X.; Ernst, J.; Furey, T. S.; Gerstein, M.; Giardine, B.; Greven, M.; Hardison, R. C.; Harris, R. S.; Herrero, J.; Hoffman, M. M.; Iyer, S.; Kellis, M.; Kheradpour, P.; Lassmann, T.; Li, Q.; Lin, X.; Marinov, G. K.; Merkel, A.; Mortazavi, A.; Parker, S. C. J.; Reddy, T. E.; Rozowsky, J.; Schlesinger, F.; Thurman, R. E.; Wang, J.; Ward, L. D.; Whitfield, T. W.; Wilder, S. P.; Wu, W.; Xi, H. S.; Yip, K. Y.; Zhuang, J.; Bernstein, B. E.; Green, E. D.; Gunter, C.; Snyder, M.; Pazin, M. J.; Lowdon, R. F.; Dillon, L. A. L.; Adams, L. B.; Kelly, C. J.; Zhang, J.; Wexler, J. R.; Good, P. J.; Feingold, E. A.; Crawford, G. E.; Dekker, J.; Elnitski, L.; Farnham, P. J.; Giddings, M. C.; Gingeras, T. R.; Guigó, R.; Hubbard, T. J.; Kent, W. J.; Lieb, J. D.; Margulies, E. H.; Myers, R. M.; Stamatoyannopoulos, J. A.; Tenenbaum, S. A.; Weng, Z.; White, K. P.; Wold, B.; Yu, Y.; Wrobel, J.; Risk, B. A.; Gunawardena, H. P.; Kuiper, H. C.; Maier, C. W.; Xie, L.; Chen, X.; Mikkelsen, T. S.; Gillespie, S.; Goren, A.; Ram, O.; Zhang, X.; Wang, L.; Issner, R.; Coyne, M. J.; Durham, T.; Ku, M.; Truong, T.; Eaton, M. L.; Dobin, A.; Tanzer, A.; Lagarde, J.; Lin, W.; Xue, C.; Williams, B. A.; Zaleski, C.; Röder, M.; Kokocinski, F.; Abdelhamid, R. F.; Alioto, T.; Antoshechkin, I.; Baer, M. T.; Batut, P.; Bell, I.; Bell, K.; Chakraborty, S.; Chrast, J.; Curado, J.; Derrien, T.; Drenkow, J.; Dumais, E.; Dumais, J.; Dutttagupta, R.; Fastuca, M.; Fejes-Toth, K.; Ferreira, P.; Foissac, S.; Fullwood, M. J.; Gao, H.; Gonzalez, D.; Gordon, A.; Howald, C.; Jha, S.; Johnson, R.; Kapranov, P.; King, B.; Kingswood, C.; Li, G.; Luo, O. J.; Park, E.; Preall, J. B.; Presaud, K.; Ribeca, P.; Robyr, D.; Ruan, X.;

Sammeth, M.; Sandhu, K. S.; Schaeffer, L.; See, L. H.; Shahab, A.; Skancke, J.; Suzuki, A. M.; Takahashi, H.; Tilgner, H.; Trout, D.; Walters, N.; Wang, H.; Hayashizaki, Y.; Reymond, A.; Antonarakis, S. E.; Hannon, G. J.; Ruan, Y.; Carninci, P.; Sloan, C. A.; Learned, K.; Malladi, V. S.; Wong, M. C.; Barber, G. P.; Cline, M. S.; Dreszer, T. R.; Heitner, S. G.; Karolchik, D.; Kirkup, V. M.; Meyer, L. R.; Long, J. C.; Maddren, M.; Raney, B. J.; Grasfeder, L. L.; Giresi, P. G.; Battenhouse, A.; Sheffield, N. C.; Showers, K. A.; London, D.; Bhinge, A. A.; Shestak, C.; Schaner, M. R.; Kim, S. K.; Zhang, Z. Z.; Mieczkowski, P. A.; Mieczkowska, J. O.; Liu, Z.; McDaniell, R. M.; Ni, Y.; Rashid, N. U.; Kim, M. J.; Adar, S.; Zhang, Z.; Wang, T.; Winter, D.; Keefe, D.; Iyer, V. R.; Zheng, M.; Wang, P.; Gertz, J.; Vielmetter, J.; Partridge, E. C.; Varley, K. E.; Gasper, C.; Bansal, A.; Pepke, S.; Jain, P.; Amrhein, H.; Bowling, K. M.; Anaya, M.; Cross, M. K.; Muratet, M. A.; Newberry, K. M.; McCue, K.; Nesmith, A. S.; Fisher-Aylor, K. I.; Pusey, B.; DeSalvo, G.; Parker, S. L.; Balasubramanian, S.; Davis, N. S.; Meadows, S. K.; Eggleston, T.; Newberry, J. S.; Levy, S. E.; Absher, D. M.; Wong, W. H.; Blow, M. J.; Visel, A.; Pennachio, L. A.; Petrykowska, H. M.; Abyzov, A.; Aken, B.; Barrell, D.; Barson, G.; Berry, A.; Bignell, A.; Boychenko, V.; Bussotti, G.; Davidson, C.; Despacio-Reyes, G.; Diekhans, M.; Ezkurdia, I.; Frankish, A.; Gilbert, J.; Gonzalez, J. M.; Griffiths, E.; Harte, R.; Hendrix, D. A.; Hunt, T.; Jungreis, I.; Kay, M.; Khurana, E.; Leng, J.; Lin, M. F.; Loveland, J.; Lu, Z.; Manthravadi, D.; Mariotti, M.; Mudge, J.; Mukherjee, G.; Notredame, C.; Pei, B.; Rodriguez, J. M.; Saunders, G.; Sboner, A.; Searle, S.; Sisu, C.; Snow, C.; Steward, C.; Tapanari, E.; Tress, M. L.; Van Baren, M. J.; Washietl, S.; Wilming, L.; Zadissa, A.; Zhang, Z.; Brent, M.; Haussler, D.; Valencia, A.; Addleman, N.; Alexander, R. P.; Auerbach, R. K.; Balasubramanian, S.; Bettinger, K.; Bhardwaj, N.; Boyle, A. P.; Cao, A. R.; Cayting, P.; Charos, A.; Cheng, Y.; Eastman, C.; Euskirchen, G.; Fleming, J. D.; Grubert, F.; Habegger, L.; Hariharan, M.; Harmanci, A.; Iyengar, S.; Jin, V. X.; Karczewski, K. J.; Kasowski, M.; Lacroute, P.; Lam, H.; Lamarre-Vincent, N.; Lian, J.; Lindahl-Allen, M.; Min, R.; Miotto, B.; Monahan, H.; Moqtaderi, Z.; Mu, X. J.; O'Geen, H.; Ouyang, Z.; Patacsil, D.; Raha, D.; Ramirez, L.; Reed, B.; Shi, M.; Slifer, T.; Witt, H.; Wu, L.; Xu, X.; Yan, K. K.; Yang, X.; Struhl, K.; Weissman, S. M.; Penalva, L. O.; Karmakar, S.; Bhanvadia, R. R.; Choudhury, A.; Domanus, M.; Ma, L.; Moran, J.; Vectorsen, A.; Auer, T.; Centanin, L.; Eichenlaub, M.; Gruhl, F.; Heermann, S.; Hoeckendorf, B.; Inoue, D.; Kellner, T.; Kirchmaier, S.; Mueller, C.; Reinhardt, R.; Schertel, L.; Schneider, S.; Sinn, R.; Wittbrodt, B.; Wittbrodt, J.; Jain, G.; Balasundaram, G.; Bates, D. L.; Byron, R.; Canfield, T. K.; Diegel, M. J.; Dunn, D.; Ebersol, A. K.; Frum, T.; Garg, K.; Gist, E.; Hansen, R. S.; Boatman, L.; Haugen, E.; Humbert, R.; Johnson, A. K.; Johnson, E. M.; Kuttyavin, T. V.; Lee, K.; Lotakis, D.; Maurano, M. T.; Neph, S. J.; Neri, F. V.; Nguyen, E. D.; Qu, H.; Reynolds, A. P.; Roach, V.; Rynes, E.; Sanchez, M. E.; Sandstrom, R. S.; Shafer, A. O.; Stergachis, A. B.; Thomas, S.; Vernot, B.; Vierstra, J.; Vong, S.; Wang, H.; Weaver, M. A.; Yan, Y.; Zhang, M.; Akey, J. M.; Bender, M.; Dorschner, M. O.; Groudine, M.; MacCoss, M. J.; Navas, P.; Stamatoyannopoulos, G.; Beal, K.; Brazma, A.; Flicek, P.; Johnson, N.; Lusk, M.; Luscombe, N. M.; Sobral, D.; Vaquerizas, J. M.; Batzoglou, S.; Sidow, A.; Hussami, N.; Kyriazopoulou-Panagiotopoulou, S.; Libbrecht, M. W.; Schaub, M. A.; Miller, W.; Bickel, P. J.; Banfai, B.; Boley, N. P.; Huang, H.; Li, J. J.; Noble, W. S.; Bilmes, J. A.; Buske, O. J.; Sahu, A. D.; Kharchenko, P. V.; Park, P. J.; Baker, D.; Taylor, J.; Lochovsky, L. An Integrated Encyclopedia of DNA Elements in the Human Genome. *Nat.* 2012 4897414 **2012**, 489 (7414), 57–74.
<https://doi.org/10.1038/nature11247>.

- (38) Segal, E.; Raveh-Sadka, T.; Schroeder, M.; Unnerstall, U.; Gaul, U. Predicting Expression Patterns from Regulatory Sequence in Drosophila Segmentation. *Nat.* 2007 4517178 **2008**, 451 (7178), 535–540. <https://doi.org/10.1038/nature06496>.
- (39) Bilu, Y.; Barkai, N. The Design of Transcription-Factor Binding Sites Is Affected by Combinatorial Regulation. *Genome Biol.* **2005**, 6 (12), R103. <https://doi.org/10.1186/GB-2005-6-12-R103>.
- (40) Morgan, X. C.; Ni, S.; Miranker, D. P.; Iyer, V. R. Predicting Combinatorial Binding of Transcription Factors to Regulatory Elements in the Human Genome by Association Rule Mining. *BMC Bioinformatics* **2007**, 8. <https://doi.org/10.1186/1471-2105-8-445>.
- (41) Xie, D.; Boyle, A. P.; Wu, L.; Zhai, J.; Kawli, T.; Snyder, M. XDynamic Trans-Acting Factor Colocalization in Human Cells. *Cell* **2013**, 155 (3), 713. <https://doi.org/10.1016/j.cell.2013.09.043>.
- (42) Xu, J.; Shao, Z.; Glass, K.; Bauer, D. E.; Pinello, L.; Van Handel, B.; Hou, S.; Stamatoyannopoulos, J. A.; Mikkola, H. K. A.; Yuan, G. C.; Orkin, S. H. Combinatorial Assembly of Developmental Stage-Specific Enhancers Controls Gene Expression Programs during Human Erythropoiesis. *Dev. Cell* **2012**, 23 (4), 796–811. <https://doi.org/10.1016/J.DEVCEL.2012.09.003/ATTACHMENT/2A363F58-399B-4808-9275-C43BE12E99EC/MMC9.XLS>.
- (43) Zhao, Y.; Cai, H.; Zhang, Z.; Tang, J.; Li, Y. Learning Interpretable Cellular and Gene Signature Embeddings from Single-Cell Transcriptomic Data. *Nat. Commun.* 2021 121 **2021**, 12 (1), 1–15. <https://doi.org/10.1038/s41467-021-25534-2>.
- (44) Valle, F.; Osella, M.; Caselle, M. A Topic Modeling Analysis of TCGA Breast and Lung Cancer Transcriptomic Data. *Cancers* 2020, Vol. 12, Page 3799 **2020**, 12 (12), 3799. <https://doi.org/10.3390/CANCERS12123799>.
- (45) Valle, F.; Osella, M.; Caselle, M. Multiomics Topic Modeling for Breast Cancer Classification. *Cancers* 2022, Vol. 14, Page 1150 **2022**, 14 (5), 1150. <https://doi.org/10.3390/CANCERS14051150>.
- (46) Yang, G.; Yang, G.; Ma, A.; Qin, Z. S.; Chen, L.; Chen, L. Application of Topic Models to a Compendium of ChIP-Seq Datasets Uncovers Recurrent Transcriptional Regulatory Modules. *Bioinformatics* **2020**, 36 (8), 2352–2358. <https://doi.org/10.1093/BIOINFORMATICS/BTZ975>.
- (47) Bravo González-Blas, C.; Minnoye, L.; Papisokrati, D.; Aibar, S.; Hulselmans, G.; Christiaens, V.; Davie, K.; Wouters, J.; Aerts, S. CisTopic: Cis-Regulatory Topic Modeling on Single-Cell ATAC-Seq Data. *Nat. Methods* 2019 165 **2019**, 16 (5), 397–400. <https://doi.org/10.1038/s41592-019-0367-1>.
- (48) Guo, Y.; Gifford, D. K. Modular Combinatorial Binding among Human Trans-Acting Factors Reveals Direct and Indirect Factor Binding. *BMC Genomics* **2017**, 18 (1), 1–16. <https://doi.org/10.1186/s12864-016-3434-3>.
- (49) Crawford, G. E.; Holt, I. E.; Mullikin, J. C.; Tai, D.; Green, E. D.; Wolfsberg, T. G.; Collins, F. S.; Blakesley, R.; Bouffard, G.; Young, A.; Masiello, C. Identifying Gene Regulatory Elements by Genome-Wide Recovery of DNase Hypersensitive Sites. *Proc. Natl. Acad. Sci. U. S. A.* **2004**, 101 (4), 992–997. <https://doi.org/10.1073/PNAS.0307540100/ASSET/7E6B2E71-40B7-4BE5-9750-54276C8281B8/ASSETS/GRAPHIC/ZPQ0030435060005.JPEG>.

- (50) Biswas, A.; Narlikar, L. A Universal Framework for Detecting Cis-Regulatory Diversity in DNA Regulatory Regions. *bioRxiv* **2020**, 2020.10.26.354522. <https://doi.org/10.1101/2020.10.26.354522>.
- (51) Heinz, S.; Benner, C.; Spann, N.; Bertolino, E.; Lin, Y. C.; Laslo, P.; Cheng, J. X.; Murre, C.; Singh, H.; Glass, C. K. Simple Combinations of Lineage-Determining Transcription Factors Prime Cis-Regulatory Elements Required for Macrophage and B Cell Identities. *Mol. Cell* **2010**, *38* (4), 576–589. <https://doi.org/10.1016/j.molcel.2010.05.004>.
- (52) Reh, R.; Sojka, P. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*; ELRA: Valletta, Malta, 2010; pp 45–50.
- (53) blei-lab/hdp: Hierarchical Dirichlet processes. Topic models where the data determine the number of topics. This implements Gibbs sampling. <https://github.com/blei-lab/hdp> (accessed Jun 25, 2021).
- (54) Mitra, S.; Narlikar, L. No Promoter Left Behind (NPLB): Learn de Novo Promoter Architectures from Genome-Wide Transcription Start Sites. *Bioinformatics* **2016**, *32* (5), 779–781. <https://doi.org/10.1093/BIOINFORMATICS/BTV645>.
- (55) Blei, D. M.; Ng, A. Y.; Jordan, M. I. Latent Dirichlet Allocation. *J. Mach. Learn. Res.* **2003**, *3* (4–5), 993–1022. <https://doi.org/10.1016/b978-0-12-411519-4.00006-9>.
- (56) Teh, Y. W.; Jordan, M. I.; Beal, M. J.; Blei, D. M. Hierarchical Dirichlet Processes. *J. Am. Stat. Assoc.* **2006**, *101* (476), 1566–1581. <https://doi.org/10.1198/016214506000000302>.
- (57) Li, Y.; Huang, W.; Niu, L.; Umbach, D. M.; Covo, S.; Li, L. Characterization of Constitutive CTCF/Cohesin Loci: A Possible Role in Establishing Topological Domains in Mammalian Genomes. *BMC Genomics* **2013**, *14* (1), 553. <https://doi.org/10.1186/1471-2164-14-553>.
- (58) Zhou, Q.; Yu, M.; Tirado-Magallanes, R.; Li, B.; Kong, L.; Guo, M.; Tan, Z. H.; Lee, S.; Chai, L.; Numata, A.; Benoukraf, T.; Fullwood, M. J.; Osato, M.; Ren, B.; Tenen, D. G. ZNF143 Mediates CTCF-Bound Promoter–Enhancer Loops Required for Murine Hematopoietic Stem and Progenitor Cell Function. *Nat. Commun.* **2021**, *12* (1), 1–12. <https://doi.org/10.1038/s41467-020-20282-1>.
- (59) Arimbasseri, A. G.; Rijal, K.; Maraia, R. J. Comparative Overview of RNA Polymerase II and III Transcription Cycles, with Focus on RNA Polymerase III Termination and Reinitiation. *Transcription* **2014**, *5* (1), 1–13. <https://doi.org/10.4161/TRNS.27369>.
- (60) Gazon, H.; Barbeau, B.; Mesnard, J. M.; Peloponese, J. M. Hijacking of the AP-1 Signaling Pathway during Development of ATL. *Front. Microbiol.* **2018**, *8* (JAN), 2686. <https://doi.org/10.3389/FMICB.2017.02686/BIBTEX>.
- (61) Kannan, M. B.; Solovieva, V.; Blank, V. The Small MAF Transcription Factors MAFF, MAFK and MAFK: Current Knowledge and Perspectives. *Biochim. Biophys. Acta - Mol. Cell Res.* **2012**, *1823* (10), 1841–1846. <https://doi.org/10.1016/J.BBAMCR.2012.06.012>.
- (62) Walhout, A. J. M.; Gubbels, J. M.; Bernards, R.; Van Der Vliet, P. C.; Timmers, H. T.

- M. C-Myc/Max Heterodimers Bind Cooperatively to the E-Box Sequences Located in the First Intron of the Rat Ornithine Decarboxylase (ODC) Gene. *Nucleic Acids Res.* **1997**, *25* (8), 1493. <https://doi.org/10.1093/NAR/25.8.1493>.
- (63) Lécuyer, E.; Hoang, T. SCL: From the Origin of Hematopoiesis to Stem Cells and Leukemia. *Exp. Hematol.* **2004**, *32* (1), 11–24. <https://doi.org/10.1016/J.EXPHEM.2003.10.010>.
- (64) Bockamp, E. O.; Fordham, J. L.; Göttgens, B.; Murrell, A. M.; Sanchez, M. J.; Green, A. R. Transcriptional Regulation of the Stem Cell Leukemia Gene by PU.1 and Elf-1*. *J. Biol. Chem.* **1998**, *273* (44), 29032–29042. <https://doi.org/10.1074/JBC.273.44.29032>.
- (65) Golson, M. L.; Kaestner, K. H. Fox Transcription Factors: From Development to Disease. *Development* **2016**, *143* (24), 4558. <https://doi.org/10.1242/DEV.112672>.
- (66) Clevenger, C. V. Roles and Regulation of Stat Family Transcription Factors in Human Breast Cancer. *Am. J. Pathol.* **2004**, *165* (5), 1449. [https://doi.org/10.1016/S0002-9440\(10\)63403-7](https://doi.org/10.1016/S0002-9440(10)63403-7).
- (67) De Bruijn, M.; Dzierzak, E. Runx Transcription Factors in the Development and Function of the Definitive Hematopoietic System. *Blood* **2017**, *129* (15), 2061–2069. <https://doi.org/10.1182/BLOOD-2016-12-689109>.
- (68) Pollak, N. M.; Hoffman, M.; Goldberg, I. J.; Drosatos, K. Krüppel-Like Factors: Crippling and Uncrippling Metabolic Pathways. *JACC Basic to Transl. Sci.* **2018**, *3* (1), 132. <https://doi.org/10.1016/J.JACBTS.2017.09.001>.
- (69) Jefferies, C. A. Regulating IRFs in IFN Driven Disease. *Front. Immunol.* **2019**, *10* (MAR), 325. <https://doi.org/10.3389/FIMMU.2019.00325/BIBTEX>.
- (70) Carnesecchi, J.; Pinto, P. B.; Lohmann, I. Hox Transcription Factors: An Overview of Multi-Step Regulators of Gene Expression. *Int. J. Dev. Biol.* **2018**, *62* (11–12), 723–732. <https://doi.org/10.1387/IJDB.180294IL>.
- (71) Jen, J.; Wang, Y. C. Zinc Finger Proteins in Cancer Progression. *J. Biomed. Sci.* **2016**, *23* (1), 1–9. <https://doi.org/10.1186/S12929-016-0269-9/FIGURES/1>.
- (72) Panganiban, G.; Rubenstein, J. L. R. Developmental Functions of the Distal-Less/Dlx Homeobox Genes. *Development* **2002**, *129* (19), 4371–4386. <https://doi.org/10.1242/DEV.129.19.4371>.
- (73) Merlo, G. R.; Zerega, B.; Paleari, L.; Trombino, S.; Mantero, S.; Levi, G. Multiple Functions of Dlx Genes. *Int. J. Dev. Biol.* **2004**, *44* (6), 619–626. <https://doi.org/10.1387/IJDB.11061425>.
- (74) Stevanovic, M.; Drakulic, D.; Lazic, A.; Ninkovic, D. S.; Schwirtlich, M.; Mojsin, M. SOX Transcription Factors as Important Regulators of Neuronal and Glial Differentiation During Nervous System Development and Adult Neurogenesis. *Front. Mol. Neurosci.* **2021**, *14*, 51. <https://doi.org/10.3389/FNMOL.2021.654031/BIBTEX>.
- (75) ETS transcription factor family - Wikipedia https://en.wikipedia.org/wiki/ETS_transcription_factor_family (accessed Mar 31, 2022).
- (76) ELK1 - Wikipedia <https://en.wikipedia.org/wiki/ELK1> (accessed Mar 31, 2022).

- (77) Luecken, M. D.; Theis, F. J. Current Best Practices in Single-Cell RNA-Seq Analysis: A Tutorial. *Mol. Syst. Biol.* **2019**, *15* (6), e8746.
<https://doi.org/10.15252/MSB.20188746>.

Chapter 6

An Algorithmic Development of the Strategy for Quantifying Rotational Motion in Molecular Machines

Chapter 6

An Algorithmic Development of the Strategy for Quantifying Rotational Motion in Molecular Machines

Abstract

Molecular machines are ubiquitous in living organisms. They carry out essential cellular tasks such as growth, metabolism, and cell division. Biological molecular machines are the most efficient machines that exist in nature. Apart from efficiency, directionality is one of the crucial properties of molecular machines. Researchers are investigating ways to create artificial molecular machines which are efficient and directional to perform desired tasks. There has been some success in creating directional molecular machines in the past few decades. However, achieving directionality at a molecular scale is still challenging and requires insight into dynamics. Previous experimental and computational studies have primarily investigated translational directionality in molecular machines. Due to the large size of these systems, computational studies have been mainly carried out using either molecular dynamics (MD) or the QM/MM approach. In this work, we have investigated rotational directionality in rotaxane and catenane systems using *ab initio* molecular dynamics. We have developed an algorithm for quantifying rotational directionality in mechanically interlocked molecular machines. We have also investigated linear regression, a machine learning algorithm, during the development. The developed algorithm captures the rotation of the ring and ring atoms. The algorithm was employed to investigate the rotation of the ring in a rotaxane system. The results indicate that in a rotaxane system, the ring distorts in the absence of the track, that the solvent mainly affects the direction of the rotation, and that the counterions influence the magnitude of the rotation.

6.1 Introduction

“There’s plenty of room at the bottom”¹, a famous quote by Feynman, still holds half a century later. Even today, we do not have a complete understanding of many molecular systems, such as molecular machines. The naturally occurring molecular machines perform various tasks in our body, e.g., transportation of cargo (kinesin), synthesis of ATP (ATP synthase), or the replication of DNA (DNA polymerase).² They are vital to the survival of all living organisms. The efficiency with which these molecules perform their tasks is astonishing.³ If we could make molecular machines that perform the desired tasks, we could revolutionize the whole health care industry. Many diseases, such as cancer, could be cured. Owing to such potential, researchers worldwide are working in the growing area of artificial molecular machines (AMMs). Biological molecular machines have inspired researchers to create similar systems. Their goal is to develop molecular machines whose motions can be controlled to perform the desired tasks. The creation of AMMs is a challenging task. It requires expertise in various fields of science, from experimental to theoretical. Nonetheless, remarkable progress has already been made in the synthesis of AMMs, with significant contributions from Jean-Pierre Sauvage, Stoddart, and Feringa.⁴⁻⁶ More than three decades back, Sauvage and Stoddart introduced mechanically interlocked molecules (MIMs), which became the building blocks of many AMMs. This led to the development of rotaxanes, catenanes, and other MIMs. In the last two decades, various molecular architectures, such as nanocars,⁷ windmills,⁸ and shuttles⁵ have been prepared, taking inspiration from their macroscopic counterpart. Although the molecular machines mimic macroscopic structures, this does not mean that they can necessarily perform a similar function at the molecular scale. Matter behaves differently at different length scales. Molecular machines need to be designed according to the operating environment. A molecular machine can produce useful work only when it is directional. For instance, if a nanocar moves an equal distance, once forward and then backward, it will end up in the same position, canceling any work performed. Useful work is only obtained when a machine is directional. In 1991 and 1994, with the aid of ingenious synthetic design, Stoddart synthesized molecular shuttles and showed for the first time that motion could be controlled at a molecular level.^{5,9} Another significant contribution to the area of AMMs came from Feringa. His group, in 1999, synthesized a new class of AMMs based on overcrowded alkenes.⁶ These molecules showed unidirectional rotational motion upon exposure to light, so they were named light-driven motors. It was one of the first examples of unidirectional rotational motion in AMMs. In 2016, due to their significant scientific contributions to the field of AMMs, the Royal Swedish Academy of Sciences jointly awarded the Nobel Prize in Chemistry to Jean-Pierre Sauvage, Ben Feringa, and J. Fraser Stoddart. Later, in 2016, translational directionality driven by fuel in AMMs was achieved by the group of David Leigh.¹⁰ With the help of novel synthetic designs, they successfully created a mechanically interlocked system consisting of a ring and a track. They called it ‘Autonomous Chemically Fuelled Small-Molecule Motor’ because translational directionality could only be seen as long as a fuel is present in the system. In this system, the ring unidirectionally translates over the track (Note: Leigh *et al.* referred to this motion as rotational motion, but it is essentially a translational motion on the track, and we will be referring to it as translation in this study).

Even though significant progress has been made to achieve controlled molecular motion, introducing directionality at a molecular scale is still challenging. A good insight into the

dynamics of AMMs is required. Understanding the factors responsible for the directionality is crucial in solving this challenge. Experimentalists have successfully achieved the directionality in few AMMs, but further development would need contributions from theoretical and computational experts. There has been considerable theoretical and computational work on molecular machines.¹¹ However, computational investigations into molecular machines have been challenging due to the large size of such systems. Therefore, all-atom quantum-mechanical (QM) calculations are not feasible. Thus, computational studies have been primarily carried out using molecular dynamics (MD) or the QM/MM approach.^{12–14} The all-atom QM calculations have been only possible in small molecular motors.^{15,16} These previous studies provide mechanistic insights into the dynamical properties of AMMs at an atomic scale.

In this study, we use *ab initio* molecular dynamics (AIMD) simulations to gain insight into the dynamics of a class of molecular machines known as mechanically interlocked molecules (MIMs). We have investigated directionality in the rotaxane system synthesized by Stoddart¹⁷ and the catenane system synthesized by David Leigh.¹⁰ In contrast to our approach, earlier reports on the computational studies of rotaxane and catenane employed a semiempirical QM/MM approach.^{18–25} These studies investigated the energetics, the translational motion, and various other properties of rotaxane and catenane systems. The main focus in the previous studies has been on the translational directionality in the MIMs. However, there has been experimental evidence for the rotational directionality in such molecules.^{17,26,27} We observed a lack of computational investigations into the rotational dynamics of these systems. Traditionally, relative motion in catenane systems is called rotation. However, we view such a motion in catenane as a translation motion. In this study, the motion that does not require movement of the center of mass is called rotation. Thus, rotational motion in rotaxane and catenane are essentially similar, except the ring in rotaxane rotates relative to the linear track and in catenane relative to the circular track. So far, rotational directionality in MIMs has been given very little attention. The molecules with rotational directionality would prove essential in many applications such as catalysis, drug design, etc.

The goal of this study was to develop an algorithm for investigating rotational motion in molecular machines (i.e., MIMs containing a ring and a track). In this chapter, first, we discuss the development of an algorithm for quantifying rotational motion in molecular machines. Then we discuss the several tests performed for verifying the algorithm using an artificial test system. After verification, we have analyzed the rotational motion in the rotaxane system using the developed algorithm. During the analysis, we identified a few issues with the algorithm, particularly with the atomic rotation. We then investigated various strategies to resolve these issues, including a machine learning algorithm (i.e., linear regression). We managed to resolve the issue with the rotation of ring atoms. Unfortunately, we were unable to resolve the issue with the rotation of the track, so we decided to investigate the rotation of only the ring in the rotaxane system. We conclude this study by investigating the effect of the track, solvent, and counterions on the rotation of the ring in a rotaxane system.

6.2 Materials and Methods

6.2.1 Systems

The following systems were investigated in this study:

Rotaxane: This system was synthesized by Stoddart.¹⁷ It contains a ring and a linear track. We also added four PF₆⁻ counterions to neutralize the total charge in the system. This system was simulated under different conditions using *ab initio* molecular dynamics, as shown in Table 6.1. Water was used as a solvent during simulations. This system is also known as [2]Rotaxane.

Table 6.1. List of different conditions under which rotaxane system was simulated.

	Internal system name	Temperature (K)	Solvent	Number of counterions	Simulation Time (ps)
Rotaxane system containing a ring and a track	molecular_rotor_b3lyp.md	1300	No	4	25
	molecular_rotor_b3lyp_withoutcounter_ion.md	1300	No	0	25
	molecular_rotor_b3lyp_with_solvent.md/ merged_scr_1_2_3_4_low_temp	1300	Yes	4	22.348
	molecular_rotor_b3lyp_with_solvent.md/ scr_hf_1600	1600	Yes	4	46.566
	ro_ci_1_removed	1300	No	3	25
	ro_ci_2_removed	1300	No	2	25
Rotaxane system containing only ring	ring	1300	Yes	0	25
	ring_without_dielectric	1300	No	0	25
	ring_with_solvent_with_ci_4	1300	No	4	25

Catenane: This system was synthesized by David Leigh in 2016.¹⁰ It contains a ring and a circular track. It was simulated at 1500 K for 25 ps with dichloromethane as solvent. We did not add any counterions to the system as it was neutral. This system is also known as [2]Catenane.

6.2.2 Computational Details

All Simulations were carried out using the B3LYP method and a 3-21g basis set except “molecular_rotor_b3lyp_with_solvent.md/scr_hf_1600”, which was simulated using the Hartree–Fock (HF) method due to time constraints. The implicit solvent was included when necessary. All simulations use a time interval of 0.5 fs. The TeraChem software package^{28–30} was used for the simulations. Scripts for the verification and analyses were written in Python. Note that ‘Frame’ or ‘Frame Number’ in this chapter refers to the time step of the simulation.

6.2.3 Terminologies

In this study, we developed a few terminologies related to the rotational motion in molecular machines. We briefly describe these terminologies below for the ring; however, they are equally applicable to the track:

- **Rotation of the Ring:** It is the average rotation of all ring atoms.

- **Ring Atoms Rotation:** The rotation of individual atoms of the ring is defined as the ring atoms rotation.

The “ring atoms rotation” and “rotation of the ring” are further classified into “absolute rotation” and “relative rotation”. We describe these terminologies below for the “rotation of ring”. However, these are equally applicable to “ring atoms rotation”, “rotation of the track”, and “track atoms rotation”. In the descriptions below, the value of instantaneous rotation between the two consecutive time steps t_1 and t_2 is always assigned to the second time step, i.e., t_2 .

- **Ring Absolute Rotation:** It is the rotation of the ring without consideration of the track.
 - **Ring Instantaneous Absolute Rotation:** It is defined as the absolute rotation of the ring between two consecutive time steps.
 - **Ring Net Absolute Rotation:** The sum of instantaneous absolute rotation of the ring at each time step is defined as ring net absolute rotation.
- **Ring Relative Rotation:** It is defined as the rotation of the ring relative to the track. The relative rotation of the ring is computed by subtracting the absolute rotation of the track from the absolute rotation of the ring.
 - **Ring Instantaneous Relative Rotation:** It is the relative rotation of the ring between two consecutive time steps.

$$\begin{aligned}
 & \textit{Ring Instantaneous Relative Rotation} \\
 & = \textit{Ring Instantaneous Absolute Rotation} \\
 & - \textit{Track Instantaneous Absolute Rotation}
 \end{aligned}$$

- **Ring Net Relative Rotation:** It is the sum of instantaneous relative rotation of the ring at each time step.

$$\begin{aligned}
 & \textit{Ring Net Relative Rotation} \\
 & = \sum \textit{Ring Instantaneous Relative Rotation}
 \end{aligned}$$

It is possible to extend these terminologies for translational motion as well.

6.3 Results and Discussion

6.3.1 Development of an Algorithm for Quantifying the Net Relative Rotation in Molecular Machines

Here, we attempt to develop an algorithm to quantify the rotational motion in molecular machines using *ab initio* molecular dynamics (AIMD). The algorithm has been developed for mechanically interlocked molecules (MIMs) such as rotaxanes and catenanes. The input to the algorithm is a molecular dynamics trajectory containing configurations of the molecular machine at different time steps (i.e., t_1, \dots, t_N , where N is the total number of steps). In particular, the molecular dynamics (MD) trajectory of the rotaxane system has been analyzed for algorithmic development. In this study, we are concerned with the systems containing a ring and a track. Motion in such systems is composed of simultaneous rotation and translation of the ring and track. Although both the ring and track are free to move, the motion that is of primary interest is the one in which the relative rotation or translation of the ring with respect to the track is non-zero. The identical motion of the ring and track results in zero relative motion, which may result in zero useful work. Furthermore, the system may produce zero useful work even if it shows non-zero relative motion at each step. For example, if a ring translates by amount l relative to the track and then translates by $-l$, the net work would be zero even though the ring showed a non-zero relative motion at every step. Thus, net non-zero relative motion is of importance. However, relative motion cannot occur without the absolute motion of individual components in the system. Therefore, in this study, we focus on quantifying the net absolute and relative rotation of the ring with respect to the track. The net relative rotation of the ring over an entire period of dynamics is computed by summing over the instantaneous relative rotation between two consecutive time steps. As ring and track are non-rigid systems, their motion is computed by averaging over all the atoms. From the MD trajectory of a rotaxane, we observed that the track stays close to the center of the ring. Therefore, we assumed that most of the rotation is concentrated along an axis that is approximately parallel to the track and perpendicular to the approximate plane of the ring. Rotation along other axes would be less probable due to steric hindrance. Hence, we have ignored the rotation along other axes in this development. The majority of the development is concerned with calculating absolute rotation between two time steps, referred to as the first time step (i.e., t_1) and the second time step (i.e., t_2) in this discussion. The algorithm for quantifying the net relative rotation of the ring with respect to track consists of the following steps:

Step-1. Identification of the Ring and the Track: To compute the relative rotation of the ring with respect to the track, it is necessary first to identify the set atoms belonging to the ring and the track. It is possible to identify these atoms using visualization softwares manually. However, due to a large number of atoms, the process is cumbersome, which may lead to human error. Thus, we have developed an automated approach for identifying the atoms belonging to the ring and the track. This approach requires only two inputs: (i) the Cartesian coordinates of the system at any time step and (ii) the indices of two atoms, one from the ring and one from the track. The indices of these two atoms belonging to the ring and the track could be easily identified using any visualization software. First, we convert the Cartesian coordinates of the system into the “MOL” format, as it contains the information on atomic

bonding lacking in Cartesian coordinates. Then, we convert all the molecules into graph representations in which nodes represent atoms and edges represent a bond. Next, we identify connected subgraphs and loop over them to identify subgraphs containing ring atoms and track atoms using the indices of the two atoms given as input. This step is performed only once during the calculation. The indices of the ring and track atoms obtained from this step are stored in separate lists. We use these lists in subsequent calculations.

Step-2. Identification of the Axis of Rotation and the Center of Rotation: Rotational motion requires the specification of the axis of rotation and the center of rotation (i.e., point through which the rotation axis passes). We are interested in computing rotation along the axis that is approximately perpendicular to the ring. Therefore, we define the axis of rotation as a vector from the center of geometry of the ring at t_1 to the center of geometry of the ring at t_2 . The center of rotation is defined as the center of geometry of the ring at t_1 . There is an issue with the current approach: when the ring does not translate, the axis of rotation becomes a zero vector. We address this issue with the following strategy:

Let $\overrightarrow{COG_1^R}$ and $\overrightarrow{COG_2^R}$ be the center of geometry of the ring at t_1 and t_2 , respectively.

Similarly, let $\overrightarrow{COG_1^T}$ and $\overrightarrow{COG_2^T}$ be the center of geometry of the track at t_1 and t_2 , respectively.

Let \vec{R} be the rotation axis.

- Define $\vec{R} = \overrightarrow{COG_2^R} - \overrightarrow{COG_1^R}$
- If $\vec{R} = \vec{0}$ then define $\vec{R} = \overrightarrow{COG_2^T} - \overrightarrow{COG_1^T}$
- If $\vec{R} = \vec{0}$ then define $\vec{R} = \overrightarrow{COG_2^T} - \overrightarrow{COG_1^R}$

Step-3. Alignment of the Ring and the Track: It is easier to compute rotation when the axis of rotation is parallel to one of the x, y, or z axis and the center of rotation lies at the origin. In this step, we transform the system at t_1 such that the axis of the rotation is along the x-axis and the center of rotation lies at the origin. Then, we apply identical transformations to the system at t_2 .

Step-4. Calculation of the Absolute Rotation of the Ring: This step is relatively straightforward. We calculate the instantaneous absolute rotation of all atoms of the ring between t_1 and t_2 by taking a projection in the y-z plane. Then, the absolute rotation of the ring is computed by averaging the instantaneous absolute rotation of all ring atoms. In this development, we consider rotation only along the x-axis and discard other components of the rotation.

Step-5. Calculation of the Absolute Rotation of the Track: In this step, the rotation of the track is calculated by following the same procedure used for the ring. The only difference is that the track is trimmed before the calculation. During trimming, only those atoms of the track are considered that are within 2 Å distance from the center of rotation.

Step-6. Calculation of the Net Relative Rotation of the ring: Then, the instantaneous absolute rotation of the track is subtracted from the instantaneous absolute rotation of the ring to obtain the instantaneous relative rotation of the ring. The rotation between any two time steps is defined as instantaneous rotation. Steps 2-5 are repeated for all consecutive time steps, and the net rotation is computed by summing over the instantaneous rotation between all

consecutive time steps. The consecutive time steps in this study have a difference of ten. In other words, we skip ten steps during the computation of instantaneous absolute and relative rotation.

6.3.2 Development of an Algorithm for Quantifying the Net Relative Translation in Molecular Machines

We have also developed an algorithm for quantifying translation in molecular machines. The purpose of this development was to investigate translational motion along with the rotational motion when required. As translation motion in molecular machines is a well-studied phenomenon, the main focus of this study is on rotational motion, and we have not carried out a detailed investigation into translation. The development of the algorithm for translation was relatively straightforward. Translational motion is computed using the displacement of the center of mass of the ring and the track between consecutive time steps. The net translation is calculated by summing up the instantaneous translation at each time step. Also, in this case, the translation of the ring is computed relative to the track.

6.3.3 Verification of the Algorithm Developed for Quantifying Rotation

In order to verify the algorithm developed for quantifying rotational motion, we created an artificial test system containing a planar circular ring and a planar track, as shown in Figure 6.1. The ring consists of ten carbon atoms and ten nitrogen atoms, whereas the track consists of nine carbon atoms and nine sulfur atoms.

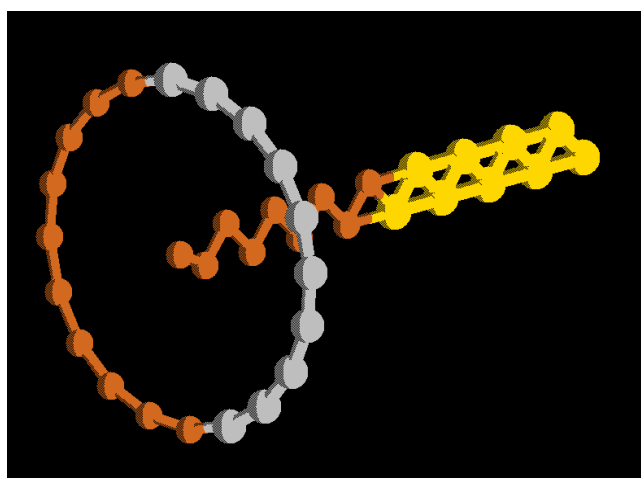


Figure 6.1. Artificial test system.

Then, we assessed the algorithm using the six tests described below. These tests were performed first on the ring and then on the track. Each test involves manual rotation of the ring and the track along one of the three axes. A few tests also involve translation along with rotation. We try to predict the rotation using the developed algorithm in these tests. The accuracy of instantaneous absolute rotation between two consecutive steps (i.e., instantaneous absolute rotation) guarantees the accuracy of relative rotation. Therefore, it is sufficient to assess the accuracy of absolute rotation. The following tests assess the accuracy of only instantaneous absolute rotation. When the system (i.e., ring or track) is rotated along the x-axis, the expected rotation is equal to the manual rotation (i.e., we expect to obtain the line: $y = x$).

However, the ring can also rotate along the y and z axis. As we are considering rotation only along the x-axis, the rotation along the y and z axis should not give rise to any rotational components along the x-axis (at least in the test system). Otherwise, it will introduce errors in the calculation. Thus, we expect rotation along the y and z axis to have zero or negligible rotation along the x-axis. If the algorithm is working correctly, we expect the predicted rotation to match the expected rotation in the tests below.

Test-1: In this test, we manually rotated the ring and the track along the x-axis from -90 degrees to +90 degrees without translation. The results of this test are shown in Figure 6.2. We observed that the predicted rotation matches perfectly with the expected rotation for the ring. On the other hand, for the track, the predicted rotation does not match the expected rotation. The reason for this is the absence of translation in the system. Neither ring translates nor track, so there is no possible way for the algorithm to determine the axis of rotation. However, it is a rare situation in the molecular dynamics of a system of this size where internal and external forces are constantly acting on the atoms.

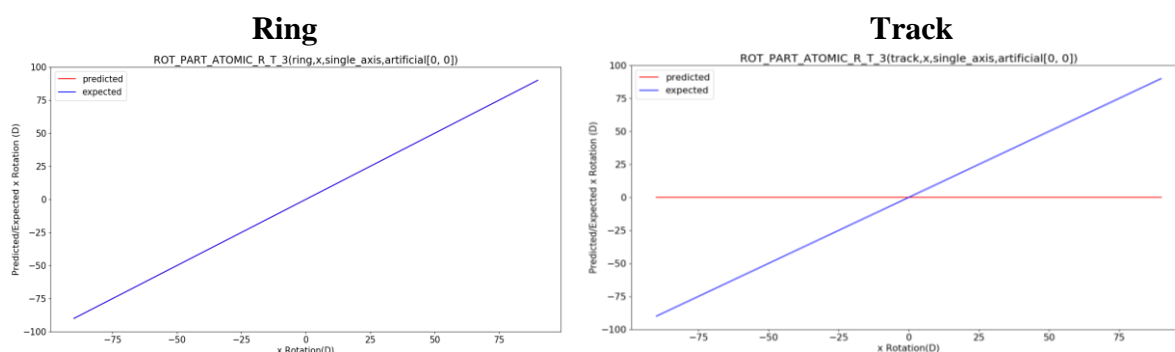


Figure 6.2. Plots showing expected and predicted rotation on the y-axis and manual rotation on the x-axis for the ring and the track corresponding to test-1. The red line represents predicted rotation from the algorithm, and the blue line denotes expected rotation.

Test-2: In this test, we manually rotated the ring and the track along the x-axis from -90 degrees to +90 degrees. We also translated the ring and the track by -10 Å and +10 Å, respectively. The results of this test are shown in Figure 6.3. We can see that the predicted rotation matches perfectly with the expected rotation for both ring and track.

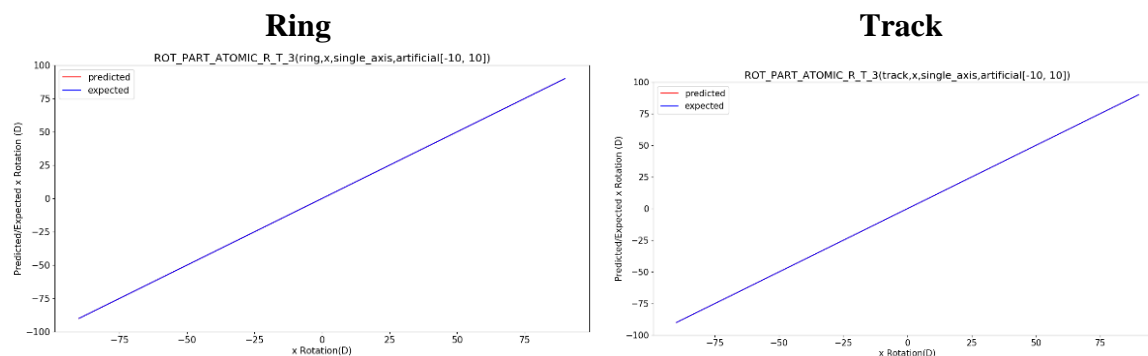


Figure 6.3. Plots showing the expected and predicted rotation on the y-axis and manual rotation on the x-axis for the ring and the track corresponding to test-2. The red line represents the predicted rotation from the algorithm, and the blue line denotes the expected rotation.

Test-3: In this test, we manually rotated the ring and the track along the y-axis from -90 degrees to +90 degrees without translation. The results of the test are shown in Figure 6.4. We expect zero predicted rotation along the x-axis from the algorithm. We can see that the predicted rotation matches perfectly with the expected rotation for both ring and track.

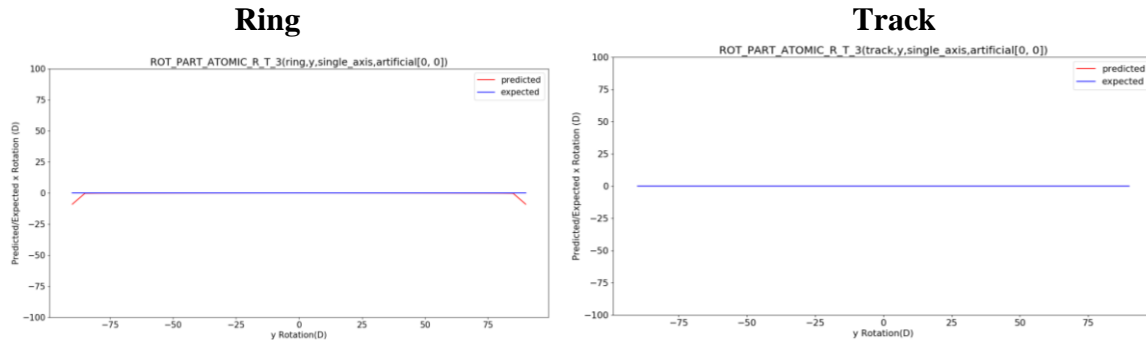


Figure 6.4. Plots showing expected and predicted rotation on the y-axis and manual rotation on the x-axis for the ring and the track corresponding to test-3. The red line represents predicted rotation from the algorithm, and the blue line denotes expected rotation.

Test-4: In this test, we manually rotated the ring and the track along the y-axis from -90 degrees to +90 degrees. We also translated the ring and the track by -10 \AA and $+10 \text{ \AA}$, respectively. The results of this test are shown in Figure 6.5. We expect zero predicted rotation along the x-axis from the algorithm. We observed that the predicted rotation matches perfectly with the expected rotation for the ring. Whereas for the track, the predicted rotation does not match the expected rotation. This type of error cannot be eliminated because the plane of the track is not perpendicular to the axis of rotation, and that is why any rotation along the y-axis gives rise to rotational components along the x-axis. Again, we need not worry about this error as its value would be small if the actual rotation is small (i.e., $\text{error} \rightarrow 0$ as $\text{rotation} \rightarrow 0$). In the dynamics, most of the time, rotation between two timesteps is of the order of 0.1 degrees or even less.

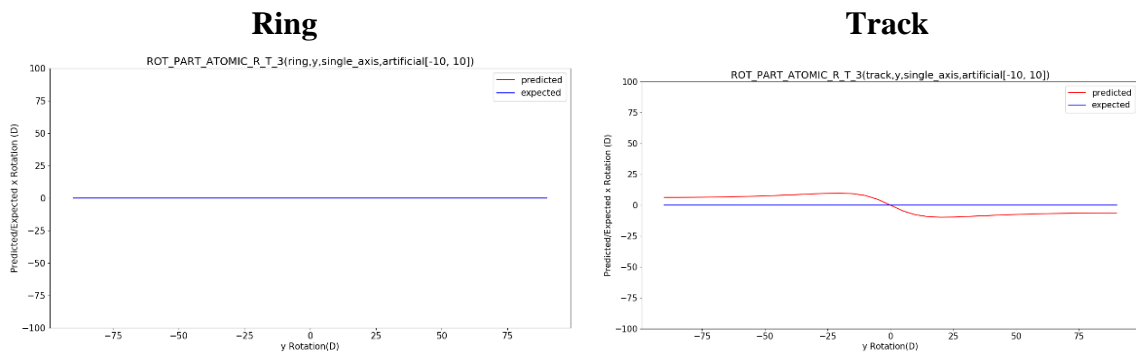


Figure 6.5. Plots showing expected and predicted rotation on the y-axis and manual rotation on the x-axis for the ring and the track corresponding to test-4. The red line represents predicted rotation from the algorithm, and the blue line denotes expected rotation.

Test-5: In this test, we manually rotated the ring and the track along the z-axis from -90 degrees to +90 degrees without translation. The results of this test are shown in Figure 6.6. We expect zero predicted rotation along the x-axis from the algorithm. We can see that the predicted rotation matches perfectly with the expected rotation for both ring and track.

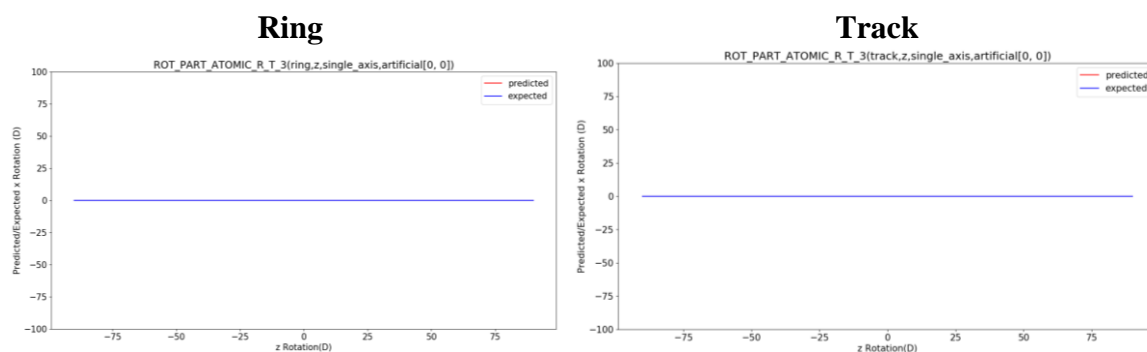


Figure 6.6. Plots showing expected and predicted rotation on the y-axis and manual rotation on the x-axis for the ring and the track corresponding to test-5. The red line represents the predicted rotation from the algorithm, and the blue line denotes the expected rotation.

Test-6: In this test, we manually rotated the ring and the track along the z-axis from -90 degrees to +90 degrees. We also translated the ring and the track by -10 Å and +10 Å, respectively. The results of this test are shown in Figure 6.7. We expect zero predicted rotation along the x-axis from the algorithm. We can see that the predicted rotation matches perfectly with the expected rotation for both ring and track.

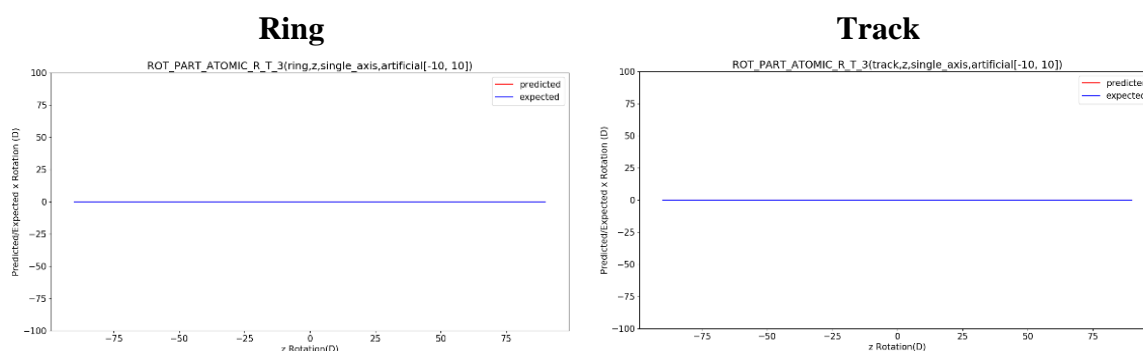


Figure 6.7. Plots showing expected and predicted rotation on the y-axis and manual rotation on the x-axis for the ring and the track corresponding to test-6. The red line represents the predicted rotation from the algorithm, and the blue line denotes the expected rotation.

6.3.4 Verification of the Algorithm Developed for Quantifying Translation

We used the same artificial test system composed of a planar circular ring and a planar track for the verification. As the translation motion is not the quantity of interest in this study, we performed only a limited number of tests for the translation only along the x-axis, as described below.

Test-7: We manually translated the ring and the track along the x-axis from -10 Å to +10 Å without any rotation. The results of this test are shown in Figure 6.8. If the algorithm is working properly, we expect the predicted translation to match the expected translation (i.e., $y = x$ line). We can see that the predicted translation matches perfectly with the expected translation for both the ring and the track.

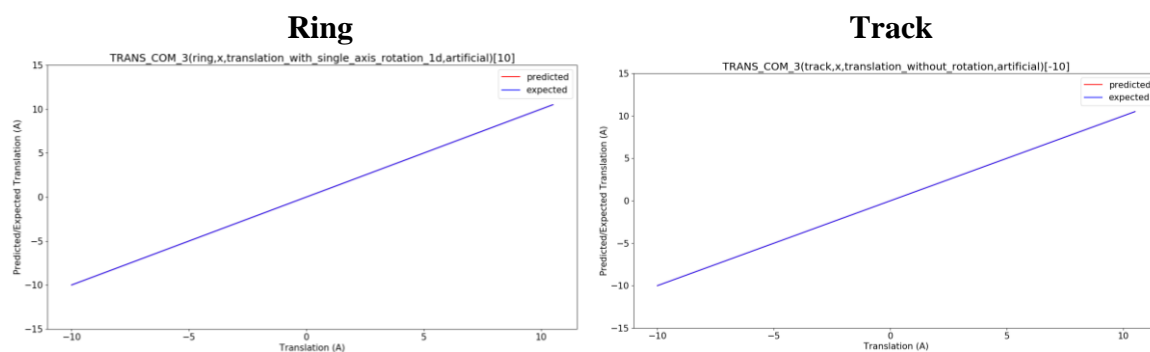


Figure 6.8. Plots showing expected and predicted translation on the y-axis and manual translation on the x-axis for the ring and the track corresponding to test-7. The red line represents the predicted translation from the algorithm, and the blue line denotes the expected translation.

Test-8: We manually translated the ring and the track along the x-axis from -10 \AA to $+10 \text{ \AA}$. We also rotated the systems by $+10$ degrees. The results of this test are shown in Figure 6.9. If the algorithm is working properly, we expect the predicted translation to match the manual translation (i.e., $y = x$ line). We observed that the predicted translation matches perfectly with the expected translation for both the ring and the track.

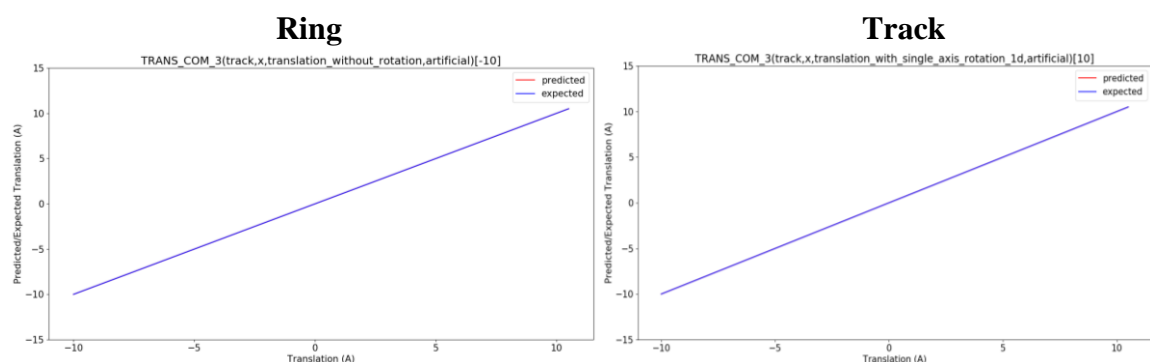


Figure 6.9. Plots showing the expected and the predicted translation on the y-axis and the manual translation on the x-axis for the ring and track corresponding to test-8. The red line represents the predicted translation from the algorithm, and the blue line denotes the expected translation.

6.3.5 Investigating Rotational and Translational Motion in Rotaxane

After verifying the algorithm on an artificial test system, we decided to investigate rotational motion in the rotaxane system synthesized by Stoddart *et al.*¹⁷ Using NMR studies, they showed that the ring in a rotaxane system rotates by 180 degrees relative to the track. We simulated the rotaxane system at 1300 K and 1600 K with the solvent for 44,697 steps to find computation evidence for this observation. Then, the simulated systems were analyzed using the developed algorithms. The results of the analysis are shown in Figure 6.10 and Figure 6.11. Simulation at high temperature (i.e., at 1600 K) was carried out to accelerate the dynamics in the system.

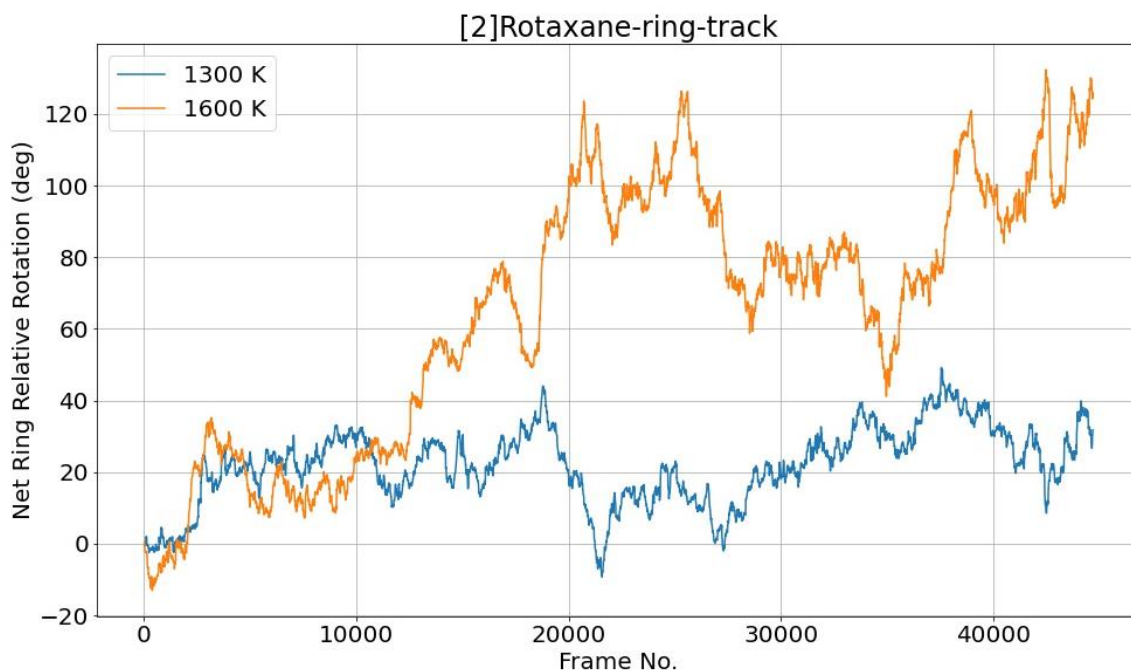


Figure 6.10. Net relative rotation of the ring in rotaxane simulated for 44,697 steps.

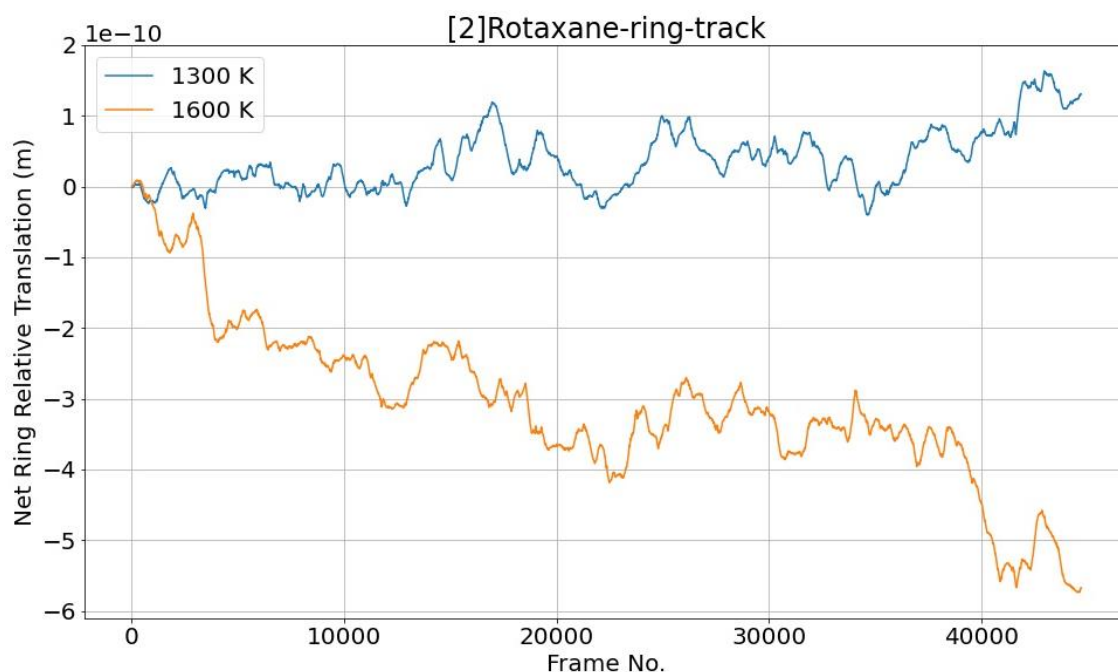


Figure 6.11. Net relative translation of the ring in rotaxane system simulated for 44,697 steps.

Figure 6.10 contains plots of the net relative rotation of the ring at two different temperatures (i.e., 1300 K and 1600 K). At low temperature, the ring shows the maximum net relative rotation of 49.20 degrees at step number 37,550 and a net relative rotation of 31.67 degrees in the last step (i.e., 44,690). At high a temperature, the maximum net relative rotation shown by the ring is 132.35 degrees at step number 42,480 and 125.82 degrees in the last step (i.e., 44,690). Similarly, the plots of the net relative translation of the ring at 1300 K and 1600 K are shown in Figure 6.11. It can be seen that the magnitude of rotation and translation is higher at 1600 K as compared to 1300 K. The increase in rotational and translational values could be

attributed to the accelerated dynamics at high temperature. However, temperature itself is not responsible for the rotational and translational directionality because the system is always at a thermal equilibrium during the whole simulation.³¹ Increased temperature has merely accelerated the dynamics in the system. In other words, given enough time, the system would show similar behavior even at low temperatures. Thus, plots in Figure 6.10 and Figure 6.11 showed the evidence for rotational and translational directionality in the rotaxane system at low and high temperatures. These results do not show enough evidence for the 180-degree rotation, as the simulations were performed only for 44,697 steps. Owing to the increasing nature of the plot at 1600 K, we expect that the system might show 180-degree rotation beyond 44,697 steps. We, therefore, carried out an extended simulation of rotaxane at 1600 K for up to 93,132 steps. This extended simulation was re-analyzed using the developed algorithms. The results of the analysis are shown in Figure 6.12 and Figure 6.13.

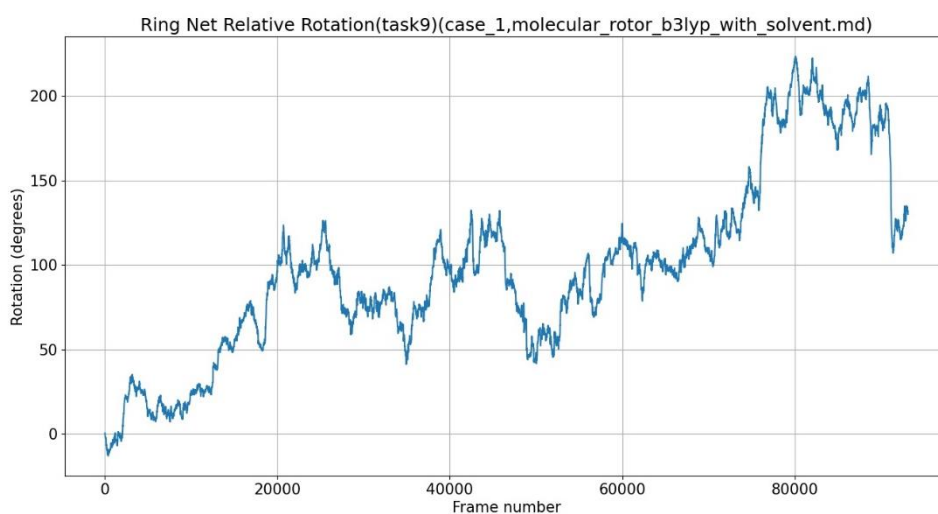


Figure 6.12. The net relative rotation of the ring in the rotaxane system simulated for 93,132 steps at 1600 K.

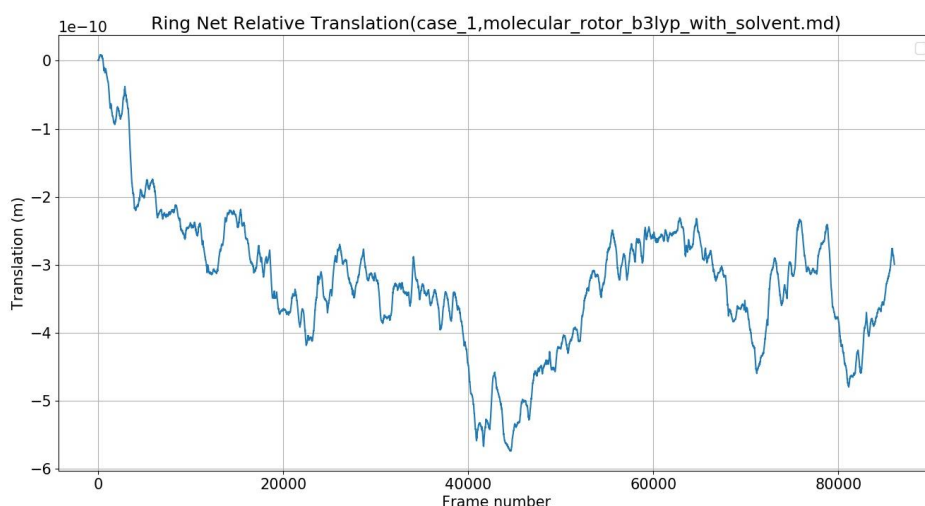


Figure 6.13. Net relative translation of the ring in rotaxane system simulated for 93,132 steps at 1600 K.

From Figure 6.12, we observed that the rotaxane system crosses 180 degrees at step no. 76,270. This could be considered as a potential evidence for the 180-degree rotation reported by

Stoddart *et al.*¹⁷ The plot of the translational directionality in an extended simulation is also shown for reference in Figure 6.13. During further investigations, we identified a few issues with the current algorithm. These issues have been discussed in the next section. Therefore, we note that the results obtained and the conclusions drawn in this section may not be correct. Further improvements in the algorithm are required to validate these results.

6.3.6 Investigating Issues with the Algorithm Developed for Quantifying Rotational Motion

The accuracy of relative rotation depends on the accuracy of the absolute rotation. Therefore, we investigated the net absolute rotation of both ring and track. The analyses in this and the following sections have been conducted primarily on the rotaxane system simulated at 1600 K with solvent and four counterions for the 93,132 steps. Therefore, “rotaxane system” in the current and subsequent sections refers to this system unless mentioned otherwise. Figure 6.14 shows the plots of the net absolute rotation of the ring and the track.

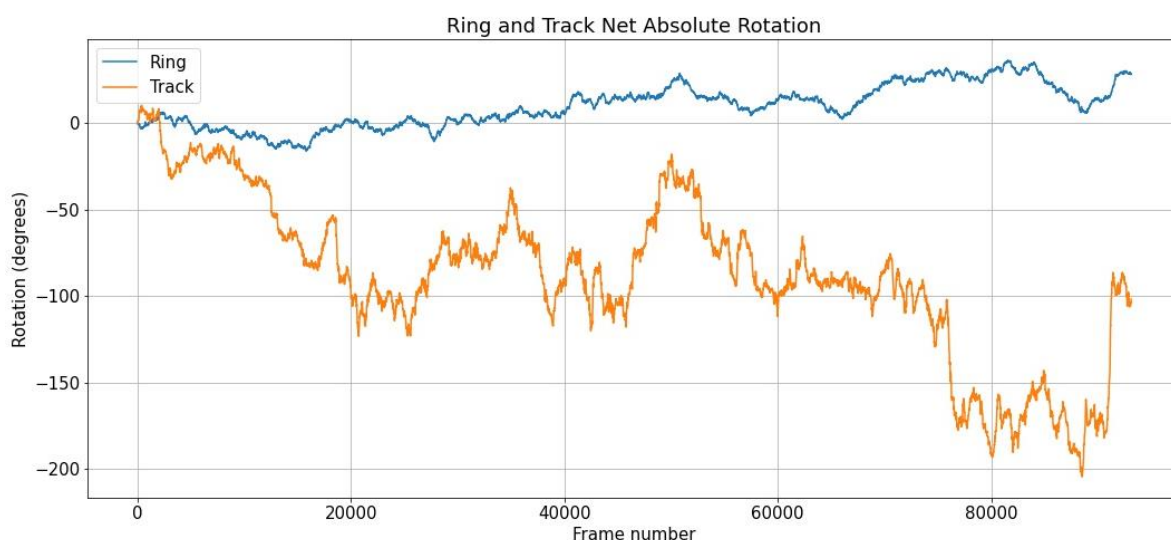


Figure 6.14. Net absolute rotation of the ring and track in the rotaxane system.

We observed that the magnitude of the net absolute rotation of the ring is smaller than the track at all the time steps. It looks like the track contributes more to the relative rotation of the ring than the ring itself in a rotaxane system. We also observed that the ring and track rotate in opposite directions, with the rotation of the ring being positive and the rotation of the track being negative. Till now, we have focused our attention on the average rotation of the system and ignored the rotation of individual atoms. Therefore, we decided to investigate the rotation of individual atoms in the ring and the track. The absolute and relative rotation of individual ring and track atoms is shown in Figure 6.15. The net relative rotation of the track atoms is exactly opposite to the net relative rotation of the ring. Therefore, it is not shown in the figures below.

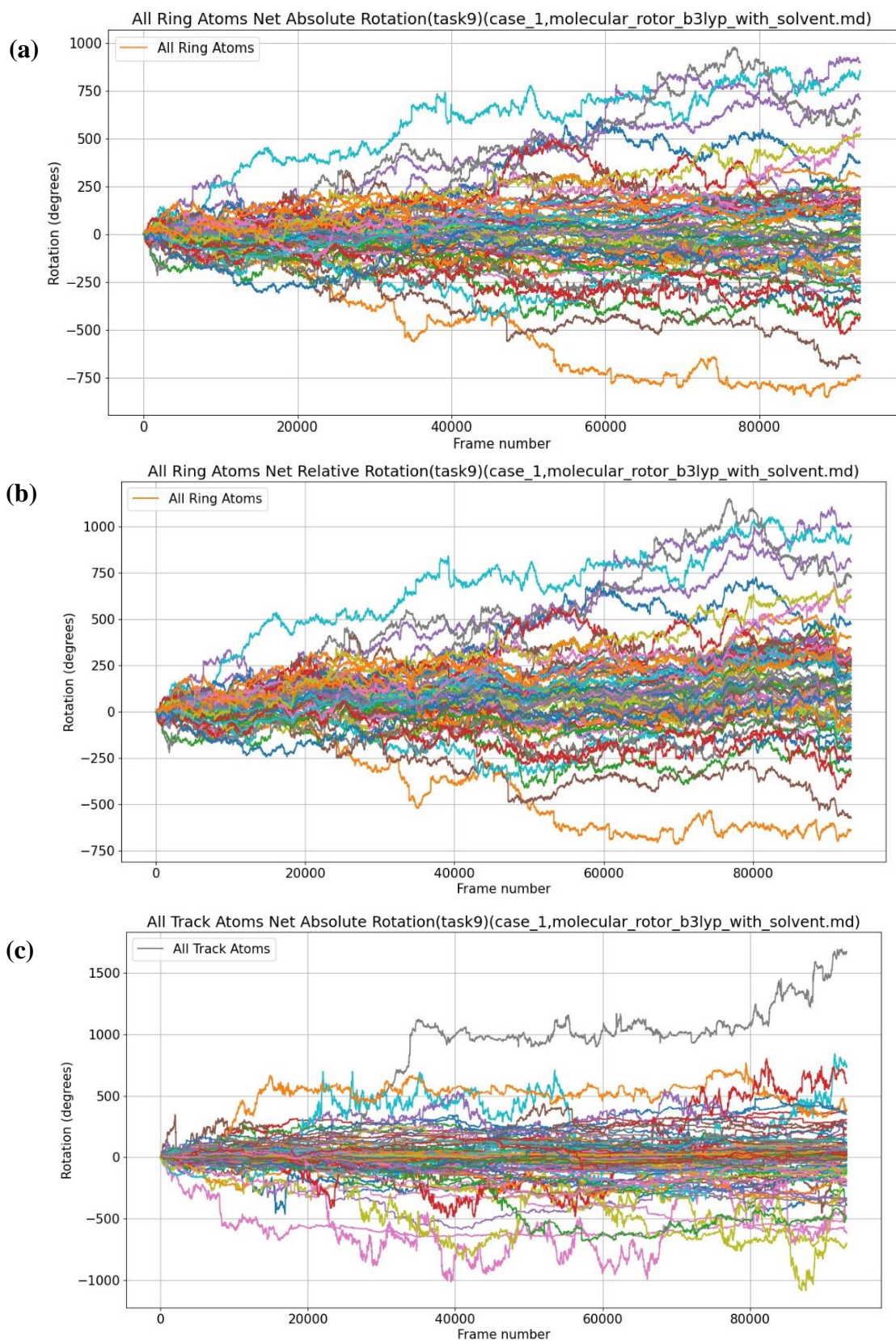


Figure 6.15. Rotation of ring and track atoms in rotaxane. (a) Net absolute rotation of ring atoms. (b) Net relative rotation of ring atoms. (c) Net absolute rotation of track atoms.

We identified a few issues with the algorithm during the analysis of atomic rotation. We observed that the rotation of the individual ring and track atoms is highly dispersed, ranging from -1000 to +1000 degrees. Thus, the algorithm fails to capture rotation at atomic scale. To gain further insight into the issue, we performed a detailed analysis of the rotation of individual ring atoms. First, we visualized the rotation axis at different time steps. A randomly selected time step from the dynamics is shown in Figure 6.16. In the figure, orange circles represent the ring atoms. The green arrow coming out of the figure (towards the viewer) is the rotation axis identified by the algorithm. The plane perpendicular to the rotation axis is shown in blue. According to the assumption, the rotation axis should approximately be perpendicular to the plane of the ring. However, for the time step shown in Figure 6.16, the rotation axis appears to be approximately parallel to the plane of the ring. Unfortunately, many other steps suffer from the same issue: the rotation axis is parallel instead of perpendicular to the ring.

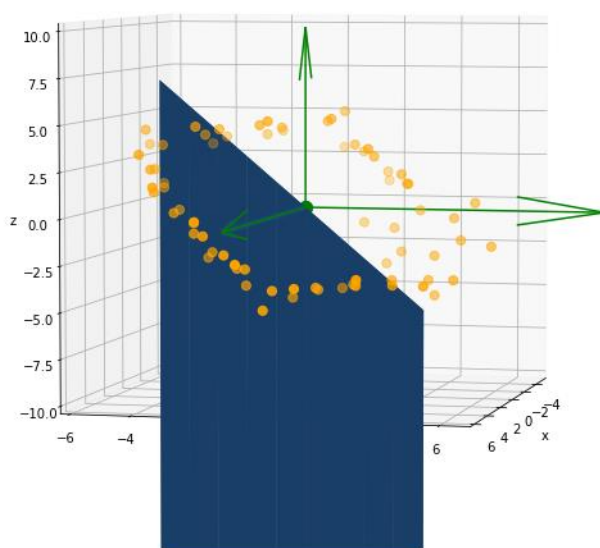


Figure 6.16. Visualization of the rotation axis in a randomly selected time step. Orange circles represent the ring atoms. The green arrow coming out of the figure (towards the viewer) is the rotation axis identified by the algorithm. The plane perpendicular to the rotation axis is shown in blue.

Next, we visualized the instantaneous rotation (i.e., rotation between two consecutive time steps) of ring atoms using the box plot (Figure 6.17). We can see that the instantaneous rotational values of individual atoms are spread across a broad range from -80 degrees to +80 degrees. Such large rotational values for the instantaneous absolute rotation are highly unlikely. We also visualized the maximum rotation of individual ring atoms. Figure 6.18 shows the time step at which a randomly selected ring atom has the maximum rotation. Red circles represent the rotation of ring atoms. The size of the red circle corresponds to the amount of rotation (i.e., a large circle represents high rotation and vice versa). We again observed that the rotation axis is approximately parallel to the plane of the ring. We also observed that the rotation of the atoms situated along the axis of rotation is higher than the atoms away from the axis.

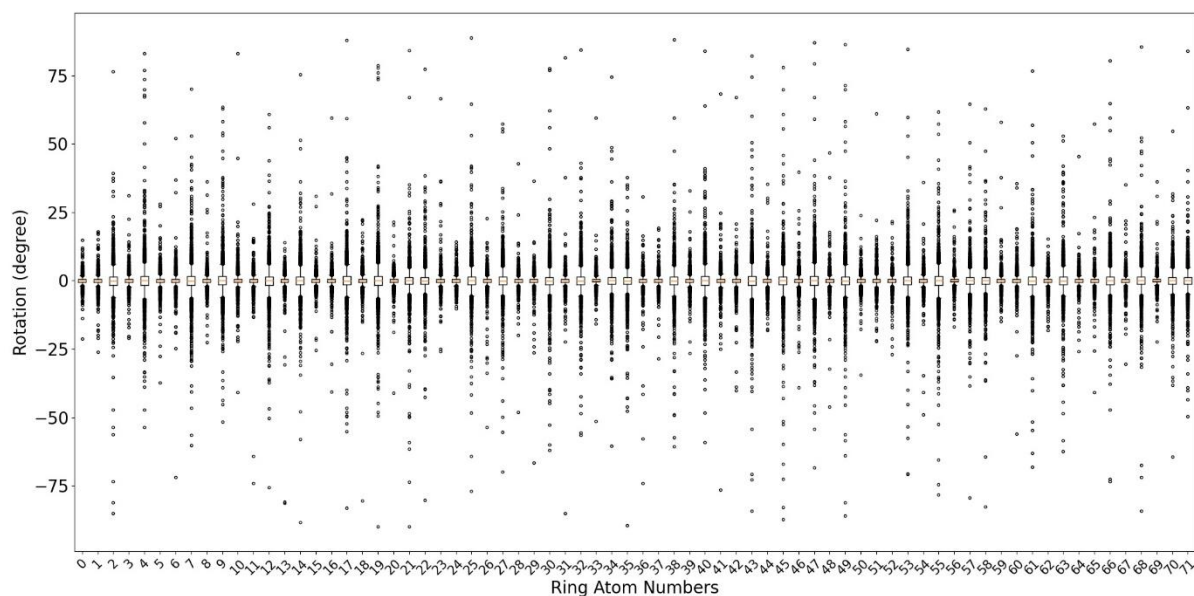


Figure 6.17. Distribution of instantaneous absolute rotation of ring atoms in the rotaxane system.

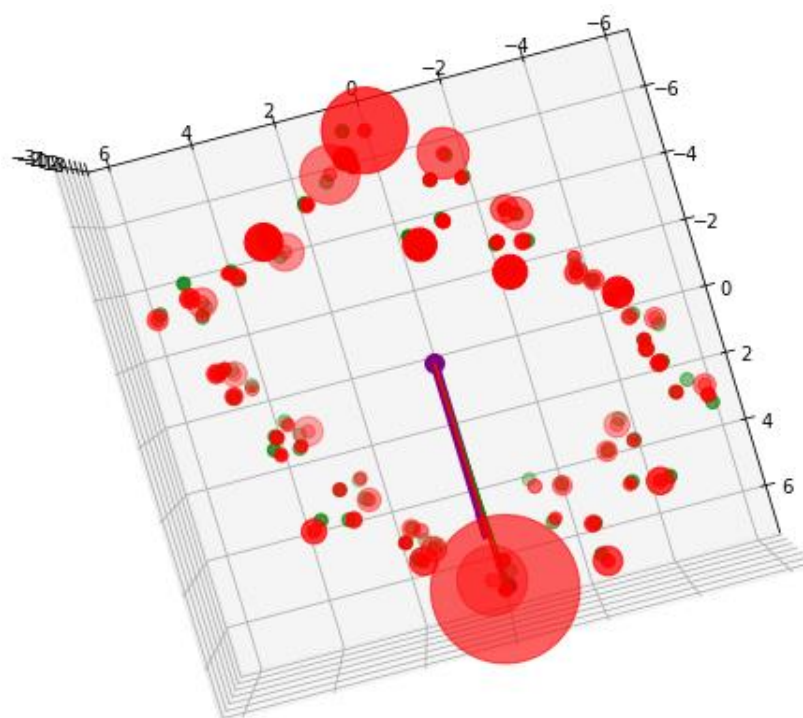


Figure 6.18. Visualization of a time step having a maximum value of instantaneous absolute rotation for a randomly selected ring atom. The rotation of ring atoms is represented by red circles. The size of the red circle corresponds to the amount of rotation.

Thus, the main cause of this issue was the axis of rotation which often comes close to the ring atoms and orients such that it is approximately parallel to the plane of the ring, giving rise to high rotational values. Therefore, we concluded that the high rotational values of the ring atoms are due to the wrong orientation of the rotation axis (i.e., parallel instead of perpendicular to the ring). In order to develop a reliable algorithm, we decided to pay attention to the rotation of individual atoms.

6.3.7 Resolving Issues Related to the Rotation of Ring Atoms

In order to address the issue with the absolute and relative rotation of ring atoms, we decided to correct the orientation of the rotation axis. We investigated two strategies that modify the method for calculating the rotation axis in step-2 (i.e., identification of the axis of rotation and the center of rotation) of the algorithm. We also added an extra step to skip invalid steps as described below.

Identification of the Rotation Axis using Linear Regression: Linear regression is a machine learning algorithm that accurately captures the linear relationship between the target and predictor variables, as shown in Figure 6.19 (a). It is one of the most efficient machine learning algorithms. In this development, we assumed that the axis of rotation is perpendicular (i.e., normal) to the plane of the ring. Thus, the rotation axis could be determined from the equation of the plane of the ring. In three dimensions, linear regression identifies a plane that captures the linear relationship in data, as shown in Figure 6.19 (b). Therefore, we employed a linear regression model to calculate the equation of the plane.

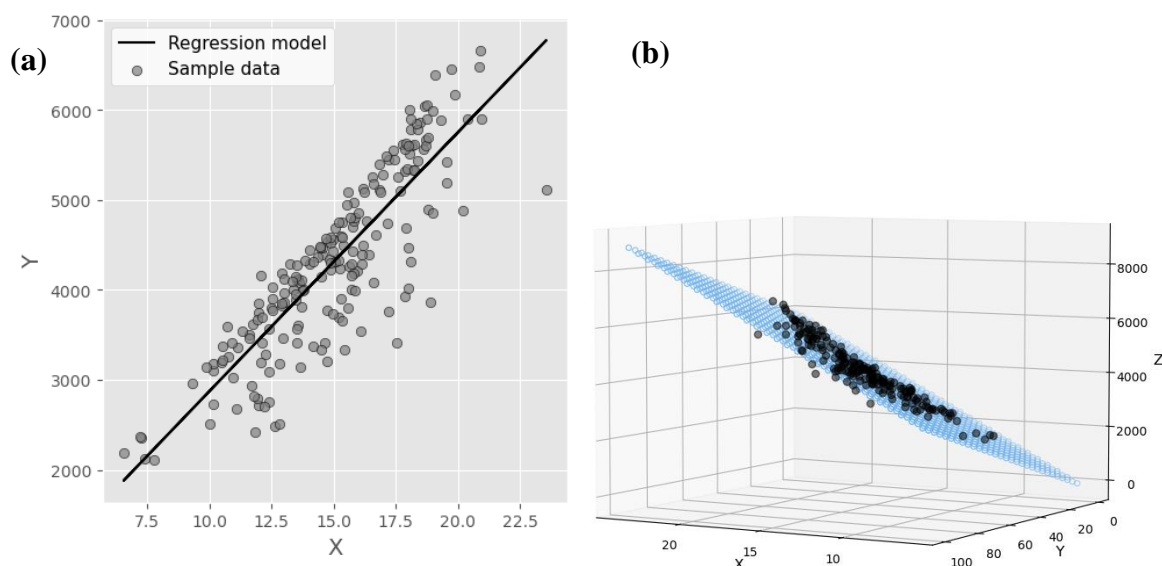


Figure 6.19. Visualizing linear regression in (i) one dimension and (ii) in three dimensions.

Another advantage of linear regression is that the parameters of the model are interpretable and could be used for calculating the normal of the hyperplane. In this strategy, we train a linear regression model on the coordinates of the ring atoms at every time step. During the training, x and y coordinates of the ring atoms represent the predictor or independent variables, whereas the z coordinate represents the target variable. We know that the plane with equation $Ax + By + Cz + D = 0$ has a normal vector $\vec{n} = A\hat{i} + B\hat{j} + C\hat{k}$. Similarly, a trained linear regression model represents the equation of the plane of the ring, $z = \beta_1x + \beta_2y + \beta_0 \Rightarrow \beta_1x + \beta_2y - z + \beta_0 = 0$ having a normal vector $\vec{n} = \beta_1\hat{i} + \beta_2\hat{j} - 1\hat{k}$. We use this normal vector obtained from the trained linear regression model as an axis of rotation. We analyzed the net absolute rotation of the ring in the rotaxane system using this strategy. The results are shown in Figure 6.20 and Figure 6.21.

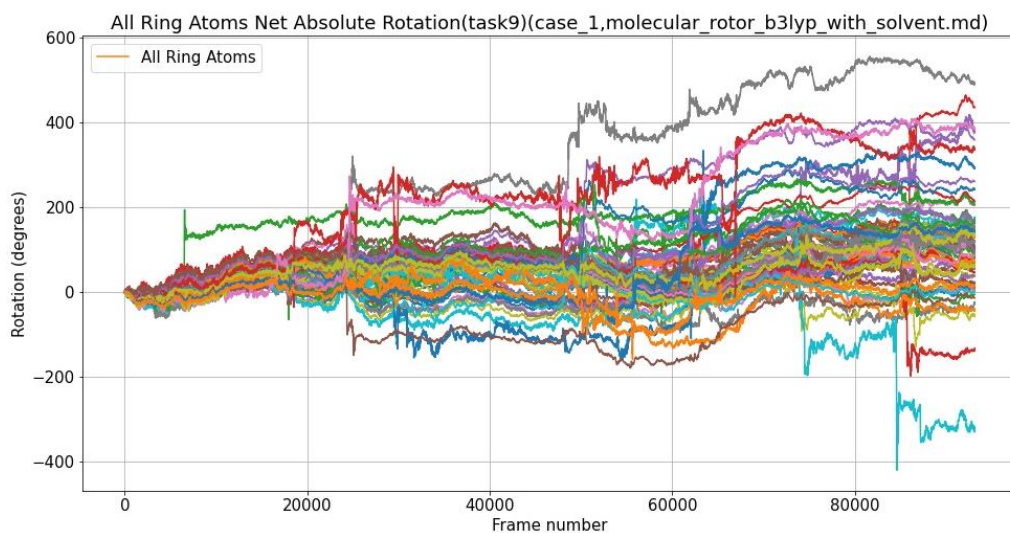


Figure 6.20. Net absolute rotation of the ring atoms obtained from linear regression strategy.

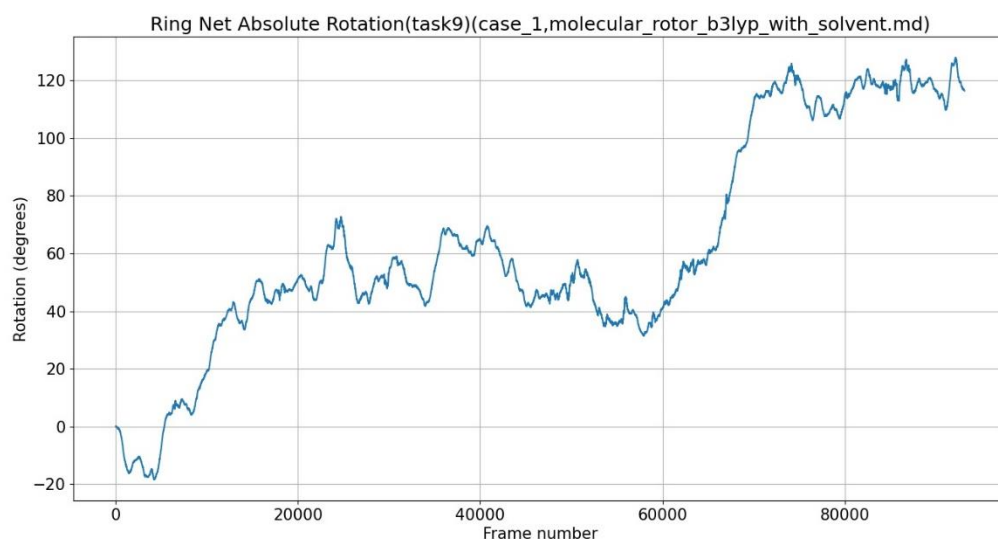


Figure 6.21. Net absolute rotation of the ring obtained from linear regression strategy.

We observed a considerable reduction in the dispersion of the net absolute rotation of the individual ring atoms. However, a few ring atoms still show relatively large positive and negative rotation. We also observed a sudden increase and decrease in the net absolute rotation of a few ring atoms, which is unexpected for the non-reacting system. Although this approach managed to resolve the issue associated with the rotation of ring and ring atoms to some extent, we still need a better strategy capable of accurately predicting the rotation of all atoms in the ring. We note that the plane obtained from linear regression depends on the orientation of the ring. Thus, for certain configurations, we may not get the correct plane.

Axis Optimisation Strategy: This strategy is inspired by linear regression. During training, linear regression minimizes a loss function with respect to the parameters of the model. Similarly, in this strategy, the axis of rotation is computed using an optimization problem as described below. A vector passing through the center of rotation located at a maximum distance from all the ring atoms is obtained by minimizing the objective function given below:

$$J = - \sum_i^N d_{i,\vec{V}}^2$$

Where,

\vec{V} is the axis of rotation

J is the negative sum of the squared distance between the axis of rotation \vec{V} and ring atoms

$d_{i,\vec{V}}$ is the perpendicular distance between i -th ring atom and the axis of rotation \vec{V}

N is the total number of ring atoms

We minimize J with respect to \vec{V} to obtain the rotation axis. Figure 6.22 shows the front and side view of the rotation axis obtained from the axis optimization strategy.

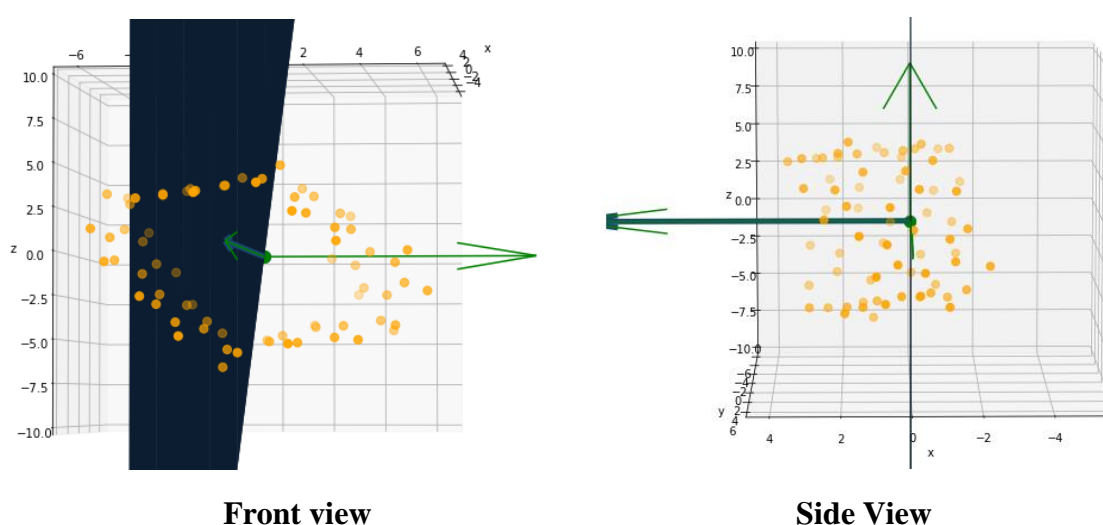


Figure 6.22. Visualization of the rotation axis obtained from the axis optimization strategy.

We can see that the rotation axis obtained from the axis optimization strategy is now approximately perpendicular to the plane of the ring (i.e., the y-z plane in figure). After analyzing many time steps, we conclude that the rotation axis obtained from this strategy is approximately perpendicular to the plane of the ring. Thus, the axis optimization strategy has resolved the issue related to the orientation of the rotation axis for most of the steps.

Addition of an Extra Step (Cylinder Test): Even after using the axis optimization strategy, we tend to get a rotation axis oriented approximately parallel to the ring in some steps. It is almost impossible to fix this issue. So we decided to ignore the time steps having an invalid rotation axis. In this step, the algorithm first decides whether a given time step is valid or not and skips the invalid time step. The time step in which the rotation axis is approximately parallel to the plane of the ring is classified as an invalid time step. We construct an infinitely long cylinder of radius 2 Å around the rotation axis to identify whether a time step is valid or invalid. Then, we find how many of the ring atoms are inside this cylinder. When the rotation axis is oriented approximately parallel to the ring, some of the ring atoms will fall inside the cylinder, as depicted in Figure 6.23. If at least one of the ring atoms is found inside this cylinder, then it is classified as an invalid time step. We carry out this procedure at every step and skip

the invalid steps. We call this procedure a “cylinder test”. Figure 6.23 depicts the valid and invalid time steps identified through this procedure. Thus, we can say that the cylinder test is a simple but powerful tool that can be used along with any base algorithm to improve its accuracy. The only problem with the cylinder test is that it can lead to severe data loss if an algorithm is not efficient at calculating the correct rotation axis.

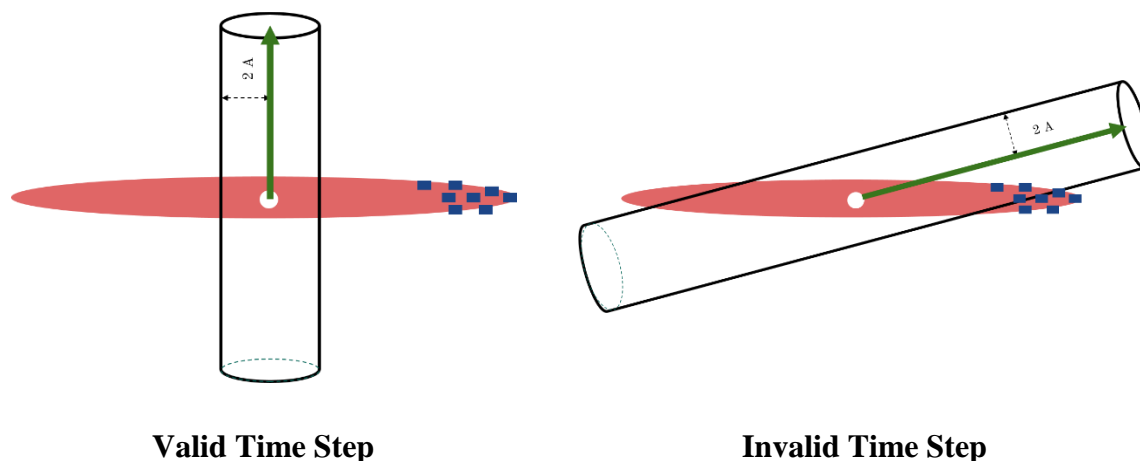


Figure 6.23. Examples showing valid and invalid time steps according to the cylinder test.

After incorporating the axis optimization strategy and the cylinder test in the algorithm, we re-analyzed the rotation of the ring in the rotaxane system. Figure 6.24 and Figure 6.25 show the net absolute rotation of the ring and individual ring atoms in a rotaxane system, respectively. Figure 6.26 and Figure 6.27, on the other hand, show the net relative rotation of the ring and ring atoms in a rotaxane system, respectively.

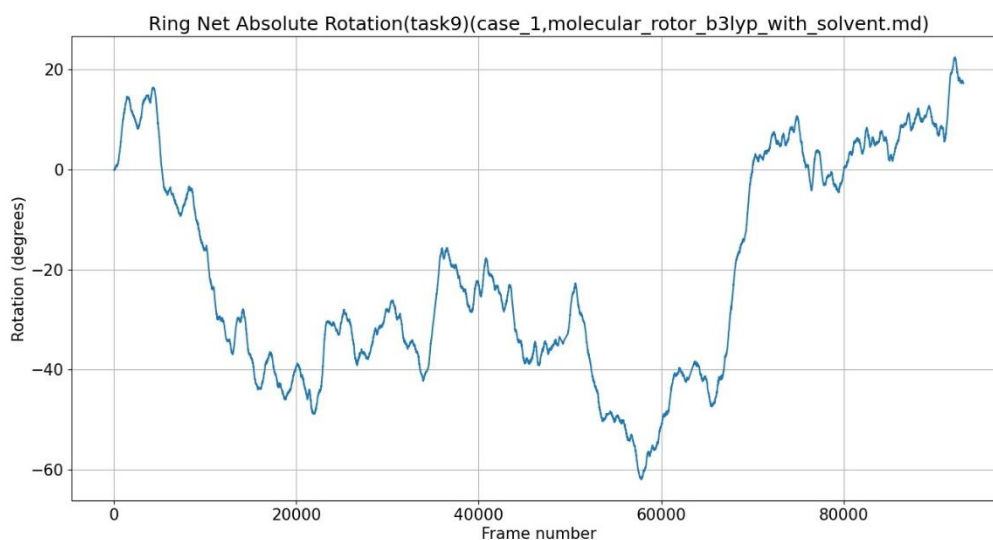


Figure 6.24. The net absolute rotation of the ring in a rotaxane system.

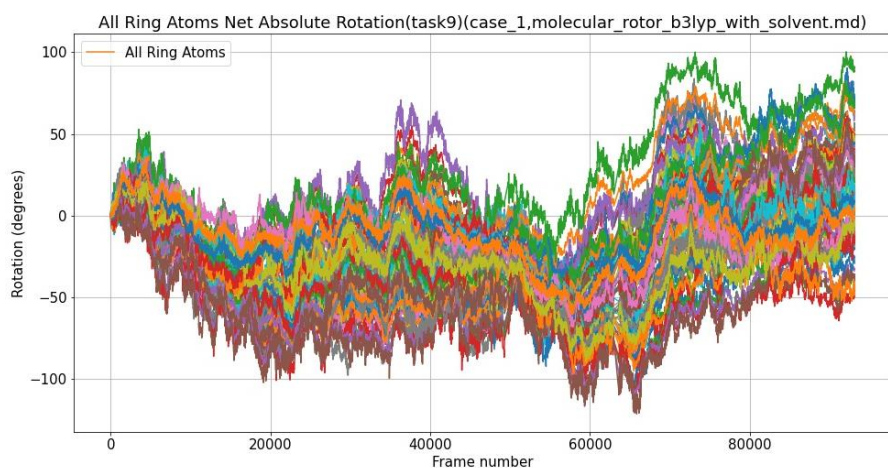


Figure 6.25. The net absolute rotation of ring atoms in the rotaxane system obtained after the incorporation of the axis optimization strategy and the cylinder test.

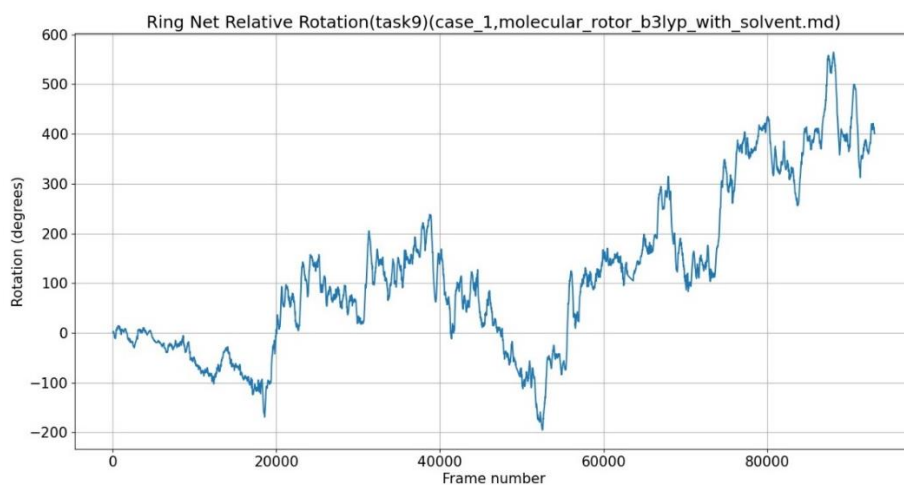


Figure 6.26. The net relative rotation of the ring in the rotaxane system obtained after the incorporation of the axis optimization strategy and the cylinder test.

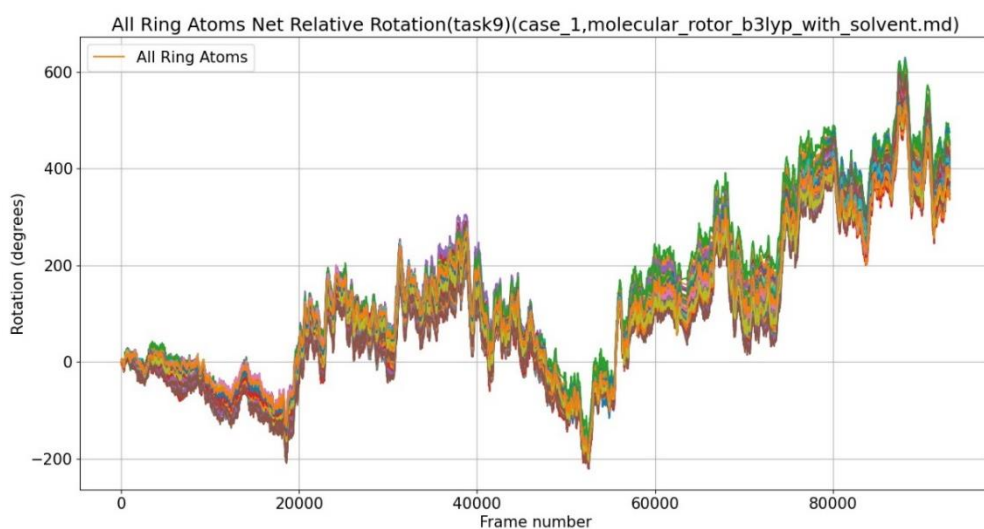


Figure 6.27. The net relative rotation of the ring atoms in the rotaxane system obtained after the incorporation of the axis optimization strategy and the cylinder test.

From the plots shown above, we can see that the absolute rotation of the ring and the individual ring atoms follow a similar trend. Similarly, the relative rotation of the ring and individual ring atoms agree in their trend. Next, we analyzed the effect of algorithmic improvements on the maximum and minimum rotation of the ring atoms. In Figure 6.28, we plot the maximum and minimum instantaneous absolute rotation of all ring atoms across the first 50,000 time steps obtained from the algorithm with and without improvements. Red and green colors correspond to the values obtained from the algorithm without improvements and with improvements, respectively.

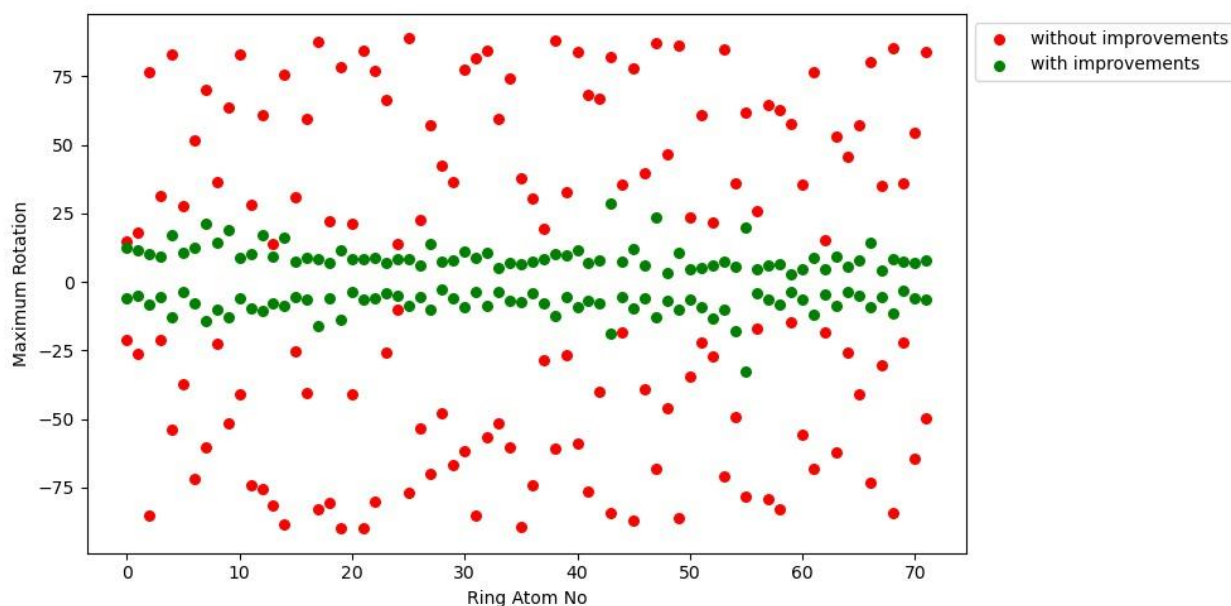


Figure 6.28. Visualizing the maximum and minimum rotation of all ring atoms obtained without algorithmic improvements and with algorithmic improvements. Algorithmic improvements include axis optimization strategy and cylinder test.

We observed a significant decrease in the maximum and minimum instantaneous rotational values of the ring atoms after improvements. Thus, we conclude that the improved algorithm decreases the atomic rotation of the ring in a rotaxane system by identifying the appropriate rotation axis that is approximately perpendicular to the ring.

6.3.8 Attempting to Resolve Issues Related to the Rotation of Track Atoms

In order to obtain correct values for the relative rotation of the ring, it is essential to obtain the correct values for the absolute rotation of the track. Therefore, we analyzed the absolute rotation of the track and individual track atoms after algorithmic improvement, which includes axis optimization strategy and cylinder test. Figure 6.29 and Figure 6.30 show the net absolute rotation of the track and individual track atoms obtained from the analysis.

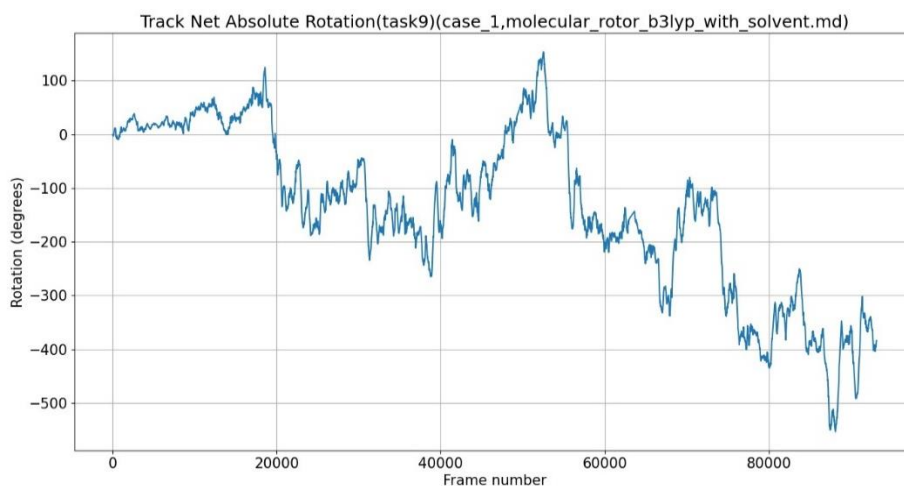


Figure 6.29. Net absolute rotation of the track in the rotaxane system obtained after algorithmic improvements. Algorithmic improvements include axis optimization strategy and cylinder test.

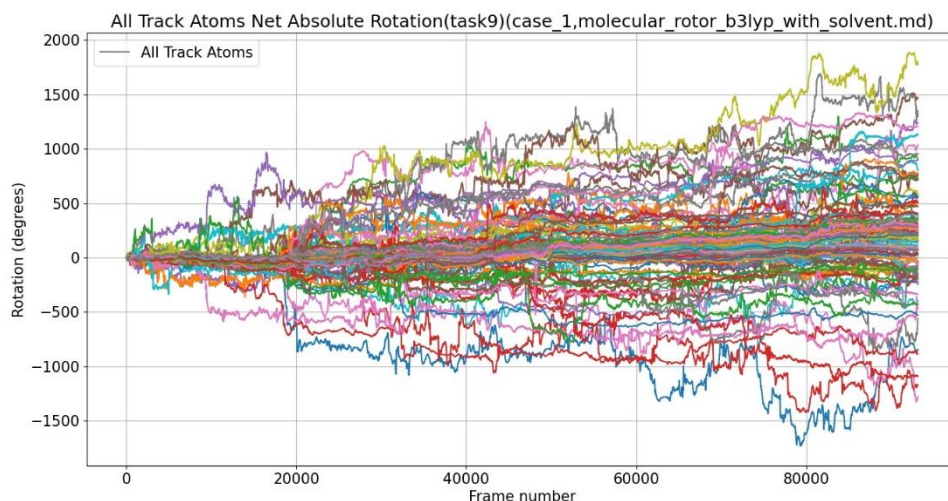


Figure 6.30. Net absolute rotation of the track atoms in the rotaxane system obtained after algorithmic improvements. Algorithmic improvements include axis optimization strategy and cylinder test.

We observed that the rotation of the track does not match the rotation of individual track atoms even after incorporating the axis optimization strategy and cylinder test. The rotation of track atoms is highly dispersed, ranging from -1000 degrees to +1500 degrees. This shows that the strategy developed for the ring does not work for the track. Hence we need to develop new strategies for addressing the rotation of the track. We modified our strategy for quantifying the rotation of the track. In step-5, we intended to use atoms of the track that are near the center of rotation. We identified such atoms by trimming the track around the center of rotation. The issue with this strategy is that it uses infinite planes to find the local track atoms. The use of infinite planes leads to the inclusion of distant track atoms due to curvatures in the track, as shown in Figure 6.31 (a). In order to overcome this, we replaced the infinite planes with a sphere around the center of geometry of the ring, as shown in Figure 6.31 (b).

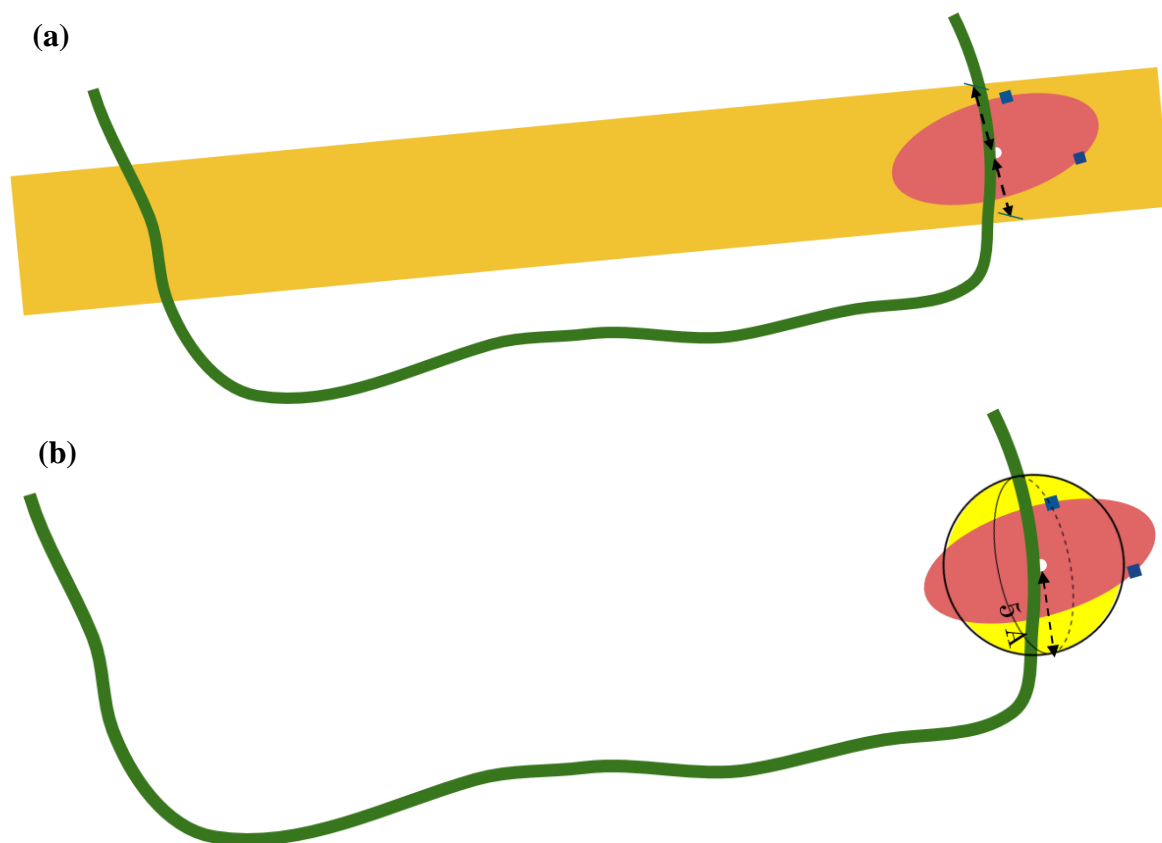


Figure 6.31. The strategy used for identifying track atoms. (a) The previous strategy employed two infinite planes, leading to the inclusion of distant track atoms. The yellow sheet represents the area between two planes (b) The new strategy uses a sphere around the center of rotation to identify the track atoms. The red circle represents the ring, and the green curve represents the track.

We re-analyzed the rotation of the track in rotaxane after incorporating the new procedure for identifying local track atoms. The results of the analysis are shown in the figures below. Figure 6.32 and Figure 6.33 show the net absolute rotation of the track and the individual track atoms, respectively. Unfortunately, the rotation of the track and track atoms obtained from the improved algorithm is still very high and unrealistic. Some track atoms appear to move in a positive direction while others in a negative direction, which is impossible.

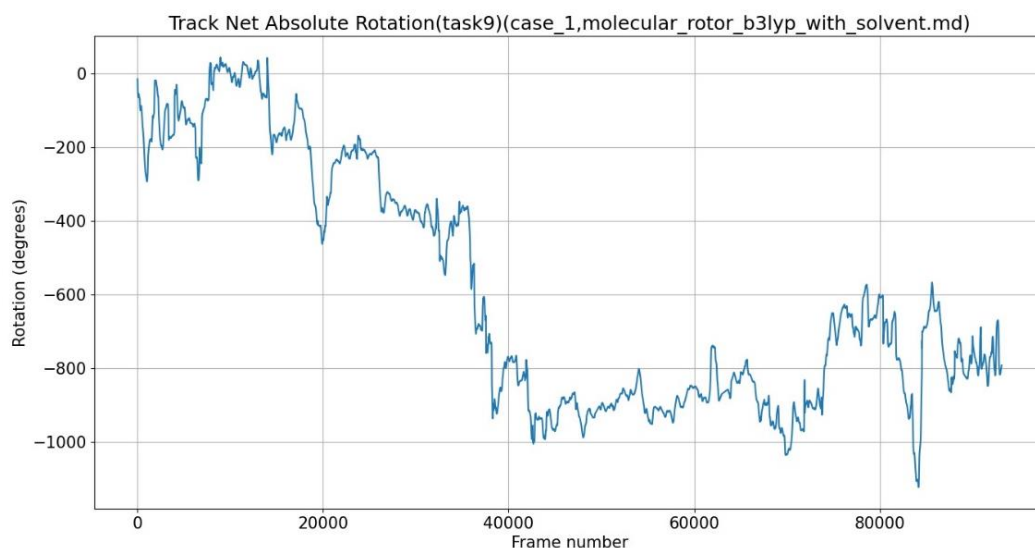


Figure 6.32. The net absolute rotation of the track in the rotaxane system obtained after incorporating the new procedure for identifying local track atoms.

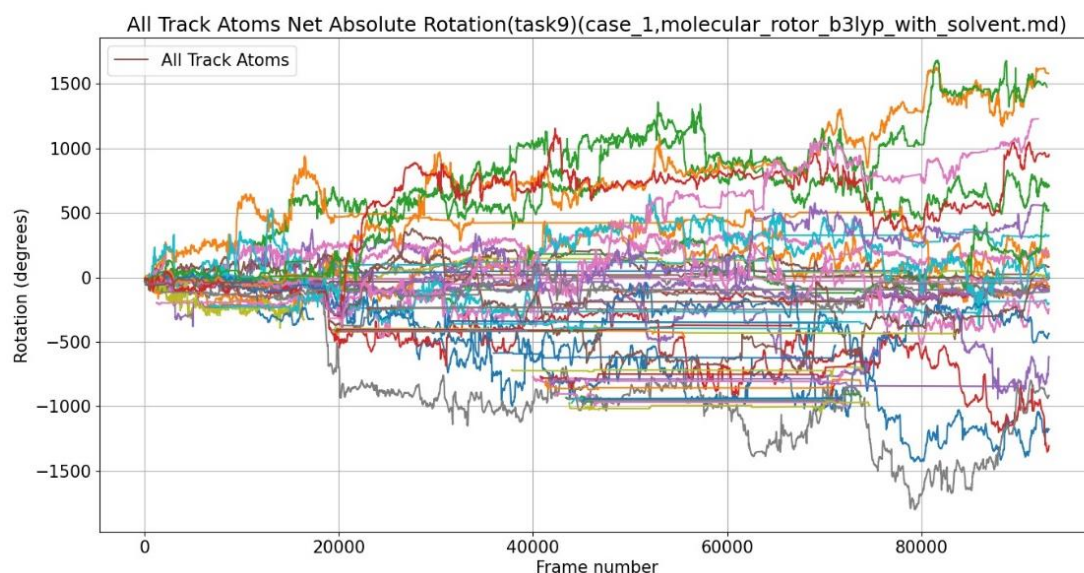


Figure 6.33. The net absolute rotation of track atoms in the rotaxane system obtained after incorporating the new procedure for identifying local track atoms.

It is possible that the issue related to the rotation of track atoms might be associated only with the system we are studying (i.e., rotaxane). Therefore, we also investigated the rotation of the track atoms in the catenane system. The rotation in the catenane system is shown in Figure 6.34. We observed that the issue persists across different systems. The rotation of track atoms in the catenane system is also highly dispersed and unrealistic, whereas the rotation of ring atoms is reasonable and matches the rotation of the ring. Thus, variation in different systems is not responsible for the unrealistic rotation of the track atoms.

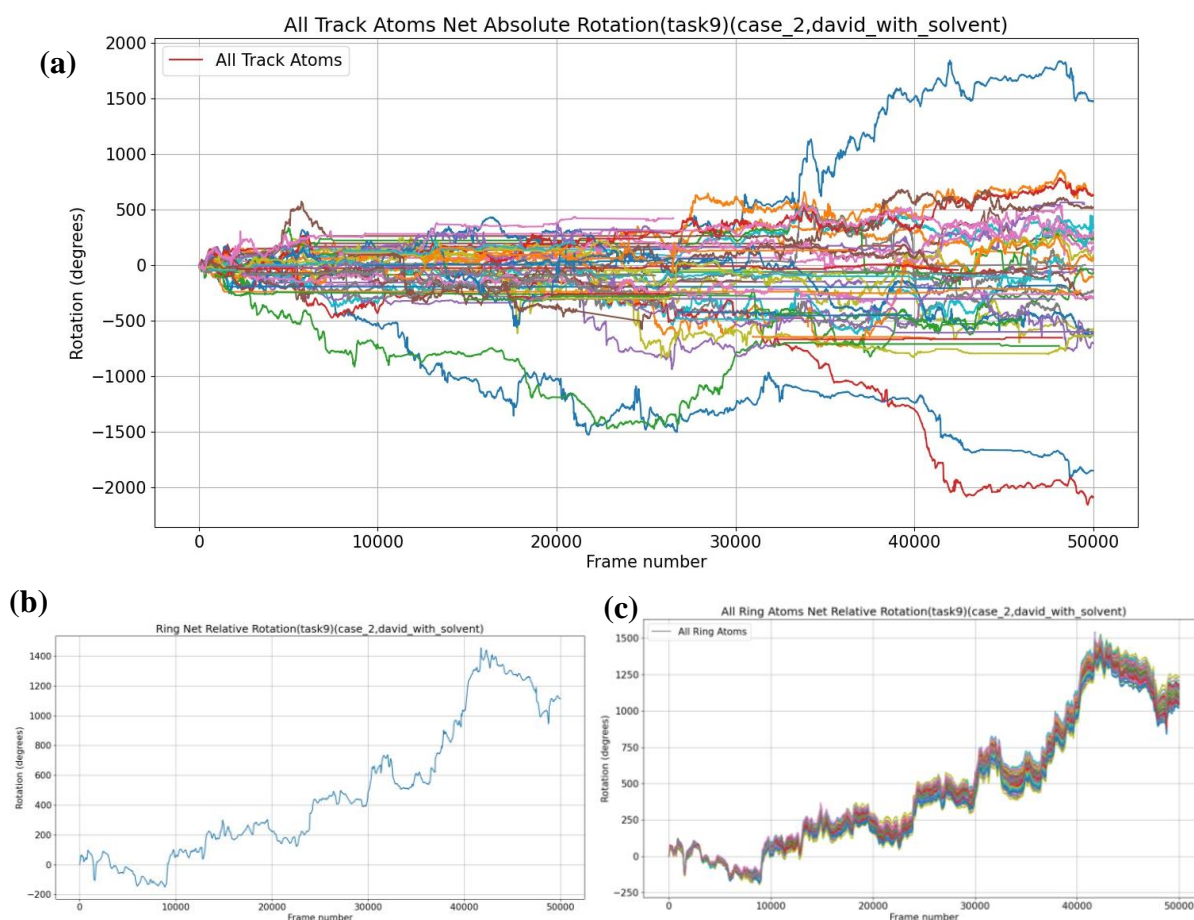


Figure 6.34. Investigating rotational motion in the catenane system simulated at 1500 K with solvent for 50,000 steps. (a) The net absolute rotation of track atoms. (b) The net relative rotation of the ring. (c) The net relative rotation of all ring atoms.

Next, we tried a few other strategies to improve our algorithm and fix the issue related to the rotation of the track atoms. Due to the lack of significant improvements, we only briefly mention these strategies below without their details.

Strategy-1: The results presented so far are based on the sphere of radius 5 Å. It effectively searches the track of length 10 Å, which is quite large for the ring having a thickness of ~4.3 Å. Therefore, we analyzed the rotation at smaller radii. We investigated spheres of radius 2 Å and 3 Å.

Strategy-2: In this strategy, we changed the center of rotation from the center of geometry of the ring to the center of geometry of the ring + track.

Strategy-3: Here, we modified the procedure for identifying the rotation axis. We decided to compute the rotation axis using two oxygen atoms (i.e., O_89 and O_106) present on the track. The new axis of rotation is defined as a unit vector from atom O_89 to O_106.

Unfortunately, all these strategies failed to address the issue related to the rotation of track atoms. Thus, this algorithm fails to correctly capture the rotation of individual track atoms.

Therefore, we cannot accurately calculate the rotation of the track and the relative rotation of the ring. Due to time constraints, we accepted this as a drawback of our algorithm.

6.3.9 Re-verification of the Improved Algorithm Developed for Quantifying Rotational Motion in Molecular Machines

As all the strategies tried so far failed to resolve the issue related to the rotation of the track atoms, we suspected that the code might have some bugs. So, we performed a detailed verification of the code using the following three different verification schemes:

Scheme-1: We performed several tests on the two test systems in this scheme. The tests include simultaneous and independent rotation and translation of the ring and the track. We performed these tests on the two test systems: (i) the artificial test system used previously and (ii) the rotaxane test system. The rotaxane test system consists of a ring and a track from the real rotaxane system. Below we show results only for an important test involving simultaneous rotation and translation of the ring and the track. The two plots corresponding to the ring and the track for each test system are shown in Figure 6.35. In this test, we manually rotated the ring and the track along the x-axis from -90 degrees to +90 degrees. We also translated the ring and the track by -10 \AA and $+10 \text{ \AA}$, respectively. The plots below show that the predicted rotation matches perfectly with the actual rotation (i.e., the $y = x$ line). We also observed that the predicted rotation matches the expected rotation in other tests not shown here. Thus, we conclude that the code passes all the standard tests and does not indicate bugs in the code.

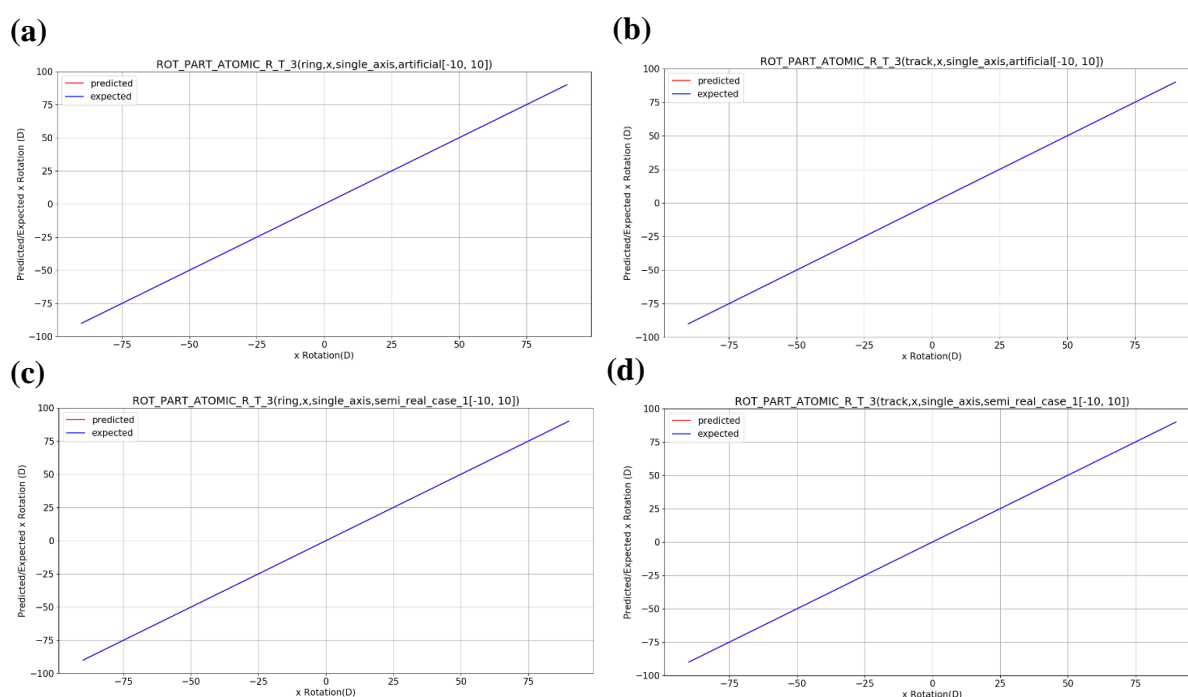


Figure 6.35. Plots showing expected and predicted rotation on the y-axis and manual rotation on the x-axis for the ring and the track corresponding to Scheme-1. The red line represents the predicted rotation from the algorithm, and the blue line denotes the expected rotation. (a) and (b) verification results on the ring and the rack from the artificial test system, respectively. (c) and (d) verification results on the ring and the track from the rotaxane test system, respectively.

Scheme-2: In this verification scheme, we simulated the rotational motion in artificial test systems in a pre-defined manner. Then, we compared the predicted rotation with the actual rotation of the system. The simulation profile of rotational motion is given in Table 6.2 below. The predicted rotation of all track atoms in this system is shown in Figure 6.36.

Table 6.2. Simulation profile of the artificial test system.

Frames	Ring Net Relative Rotation
0 to 100	0 to -200
100 to 200	-200 to 0
200 to 300	0 to -200

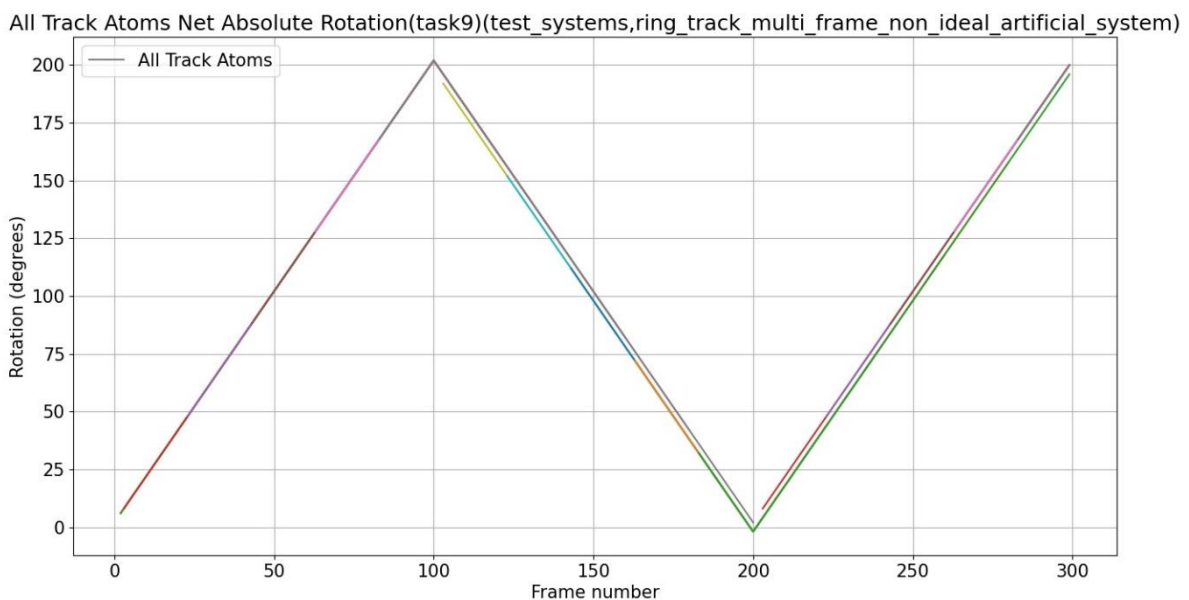


Figure 6.36. Net absolute rotation of track atoms in a simulated artificial test system.

From the plots in Figure 6.36, we can see that the predicted rotation matches very well with the actual simulated rotation of the artificial test system. Thus, we conclude that the code passes this verification test and that there are no indications of bugs in the code.

Scheme-3: In this scheme, we measured the angular deviation of the rotation axis from its initial position for the first 50,000 steps. If the algorithm is consistent in computing the direction of the rotation axis, we expect angular deviation to remain below 90 degrees. The sudden increase in the deviation above 90 degrees would indicate the reversal of the rotation axis. The result of this verification is shown in Figure 6.37. We observed that the angular deviation of the rotation axis is well below 90 degrees. Thus, we conclude that the direction obtained from the new algorithm is consistent.

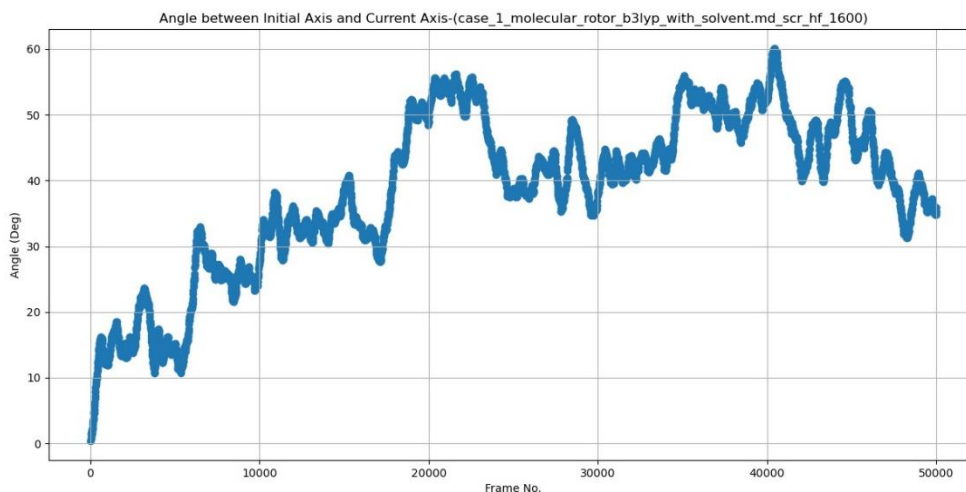


Figure 6.37. Scatter plot of the angular deviation of the rotation axis with respect to the initial rotation axis.

6.3.10 Investigating Rotational Motion of Only the Ring in the Molecular Machine

As the rotational dynamics of the ring obtained from the algorithm is very reasonable, we decided to focus our attention on the rotation of only the ring in the rotaxane system. One of the systems we are studying contains only a ring. Therefore, we verified the algorithm on the test system containing only a ring.

Verification on the System Containing only a Ring: We removed the track from the artificial test system to obtain the test system containing only a ring, as shown in Figure 6.38. Then we rotated the ring along the x-axis from -90 degrees to +90 degrees with and without translation. If the algorithm is working properly, we expect the predicted rotation to match the manual rotation (i.e., $y = x$ line).

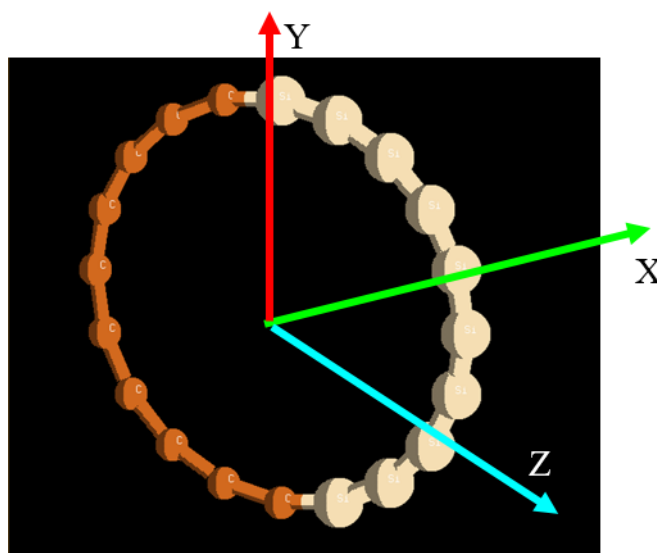
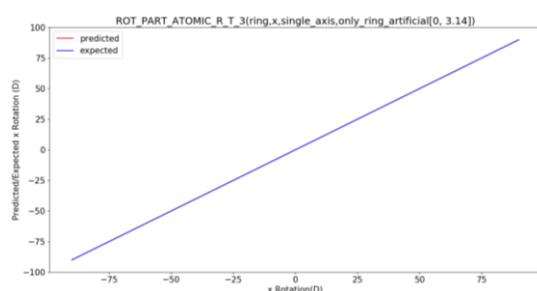


Figure 6.38. Artificial test system containing only the ring.

With Translation



Without Translation

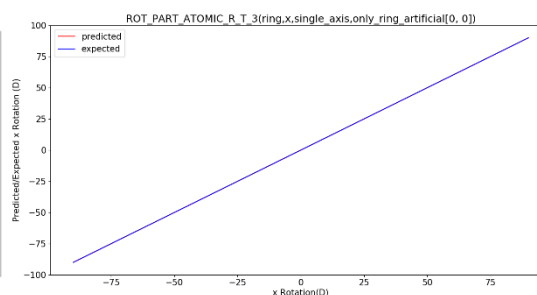


Figure 6.39. Plots showing expected and predicted rotation on the y-axis and manual rotation on the x-axis for the artificial test system containing only a ring. Verification was performed with and without translation of the ring. The red line represents predicted rotation from the algorithm, and the blue line denotes the expected rotation.

Figure 6.39 contains the verification results on the only ring test system. It can be seen that the predicted rotation matches perfectly with the expected rotation. Furthermore, we simulated the rotaxane test system containing only a ring. The rotational motion was simulated in a pre-defined manner. The simulation profile of the rotational motion is depicted in Table 6.3. The predicted rotation of all the ring atoms of this system is shown in Figure 6.40.

Table 6.3. Simulation profile of rotaxane test system containing only the ring.

Frames	Ring Net Absolute Rotation
0 to 100	0 to 100
100 to 300	100 to -100
300 to 500	-100 to 100

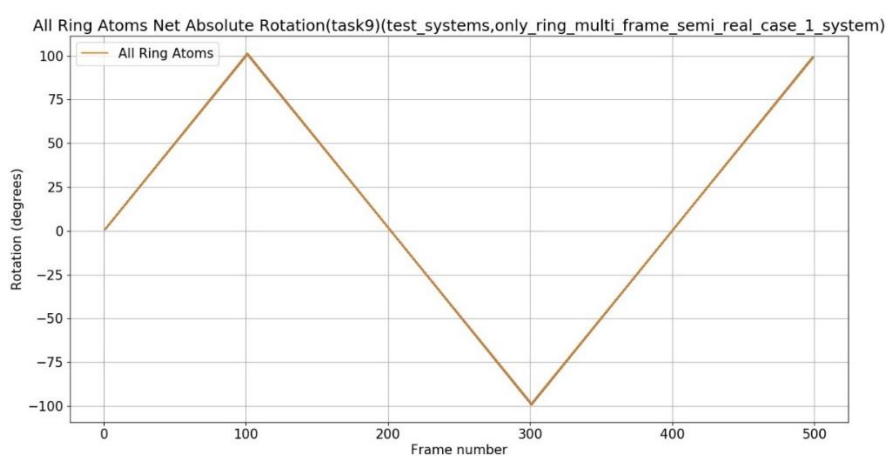


Figure 6.40. Net absolute rotation of the ring atoms in the simulated rotaxane test system containing only the ring.

From the plots in Figure 6.40, we observed that the predicted rotation matches very well with the simulated rotation of the ring. Thus, we conclude that the code is ready to investigate the absolute rotation of the ring in molecular machines. Next, we investigated the effect of the track, solvent, and counterions on the net absolute rotation of the ring in the rotaxane system.

Effect of the Track: We simulated the rotaxane system with and without the track under identical conditions (i.e., at 1300 K with solvent). Figure 6.41 (a) and Figure 6.41 (b) below show the net absolute rotation of the ring with and without the track, respectively. We observed long and straight lines in Figure 6.41 (b) corresponding to the time steps skipped during the computation of the net absolute rotation. We attribute the loss of time steps to the failure of the rotation axis to pass the cylinder test. To confirm this, we reduced the cylinder radius from 2 Å to 1 Å and recomputed the net absolute rotation of the ring without the track, which is shown in Figure 6.41 (c). We do not see the long and straight lines anymore in the plot. Thus, the rotation axis comes near the ring atoms in the absence of the track, resulting in failure to pass the cylinder test. In order to understand the factors responsible for the failure of the cylinder test, we analyzed the orientation of the rotation axis. The plot in Figure 6.41 (d) shows the angular deviation in the direction of the rotation axis with respect to the initial rotation axis in the rotaxane system without a track. If the orientation of the rotation axis was responsible, we expect to see the sudden appearance of large angular deviations. However, we observed a gradual change throughout the simulation. Thus, the orientation of the rotation axis cannot be the factor responsible for the failure of the cylinder test. Another factor could be the distortion of the ring. We investigated the distortion of the ring manually using visualization software.³² The undistorted ring at time step 0 and distorted ring at time step 20,000 are shown in Figure 6.42. We observed that the ring distorts significantly during the simulation. The six-membered rings present in the rotaxane ring distort and come inside the cylindrical region leading to the failure of the cylinder test. It was also observed that atomic rotation computed by the algorithm is highly dispersed for the ring without the track. One possible reason could be that the algorithm is struggling to find the correct rotation axis due to distortion in the ring.

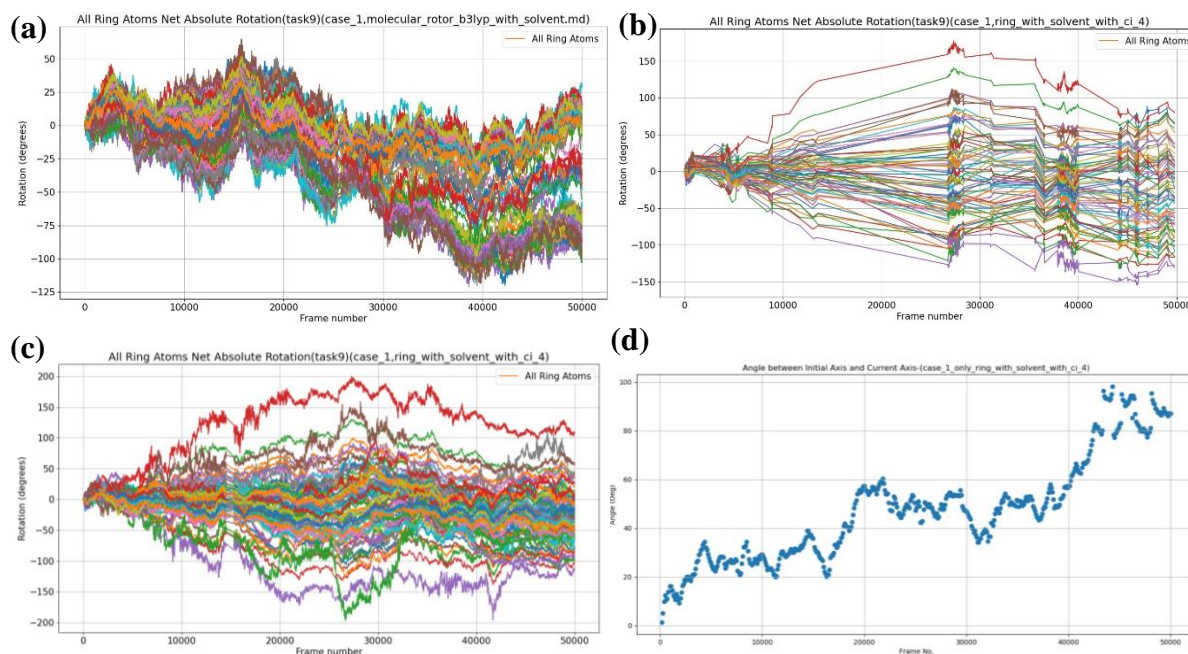


Figure 6.41. Effect of the track on the rotation of the ring in the rotaxane system simulated at 1300 K with solvent and counterions. (a) Net absolute rotation of ring atoms in the presence of the track. (b) Net absolute rotation of ring atoms without the track. (c) Net absolute rotation of the ring computed after decreasing the radius of the cylinder. (d) Angular deviation of the rotation axis with respect to the initial rotation axis.

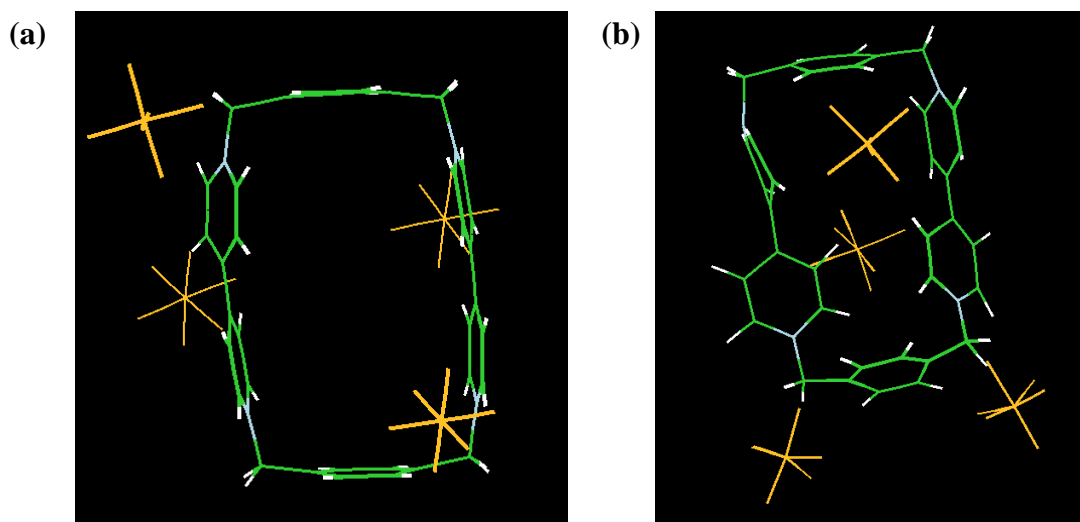


Figure 6.42. Visualizing the distortion of the ring in a rotaxane system without track (a) Ring at time step 0 (b) Ring at time step 20,000.

Effect of the Solvent: The rotaxane system containing both the ring and the track was simulated under identical conditions at 1300 K. The net absolute rotation of the ring with and without solvent is shown in Figure 6.43. We observed that the direction of rotational motion of the ring simulated with solvent is opposite to the ring simulated without the solvent. This trend is clearly visible between 30,000 to 50,000 time steps. For most of the steps, the net absolute

rotation of the ring with solvent is negative, whereas it is positive for the ring without the solvent. We did not observe a large difference in the magnitude of the net absolute rotation. However, it was observed that the maximum magnitude of the net absolute rotation of the ring with solvent is higher than the ring without solvent. Thus, we conclude that the solvent primarily affects the direction of rotation of the ring in the rotaxane system.



Figure 6.43. Plots showing net absolute rotation of the ring with and without solvent in the rotaxane system simulated at 1300 K in the presence of counterions.

Effect of Counterions: The rotaxane system is positively charged. Therefore, we added four PF_6^- counterions to balance the charge during the simulation. As electrostatic interaction is capable of producing torque, we assessed the effect of counterions on the rotational motion of the ring. Figure 6.44 shows the net absolute rotation of the ring in the rotaxane system with and without counterions. We observed that the net absolute rotation of the ring with and without counterions has a very similar trend except for the first ~8000 steps. We also observed that the value of net absolute rotation of the ring in the presence of counterions is higher than the ring without counterions at each step. To further understand the effect of counterions, we simulated the rotaxane system by removing one and two counterions. The box plots in Figure 6.45 show the distribution of net absolute rotation computed by varying the number of counterions in the system. It can be seen that the maximum value, 75th quartile, 50th quartile (i.e., median), 25th quartile, and minimum value of the net absolute rotation decrease systematically as we remove counterions from the system. We also analyzed the rotation of individual ring atoms. Figure 6.46 shows the rotation of ring atoms with and without counterions. We observed high dispersion in atomic rotation in the absence of counterions, possibly due to the distortion of the ring. Thus, counterions increase the value of the net absolute rotation and decrease the dispersion in atomic rotation computed by the algorithm in the rotaxane system.

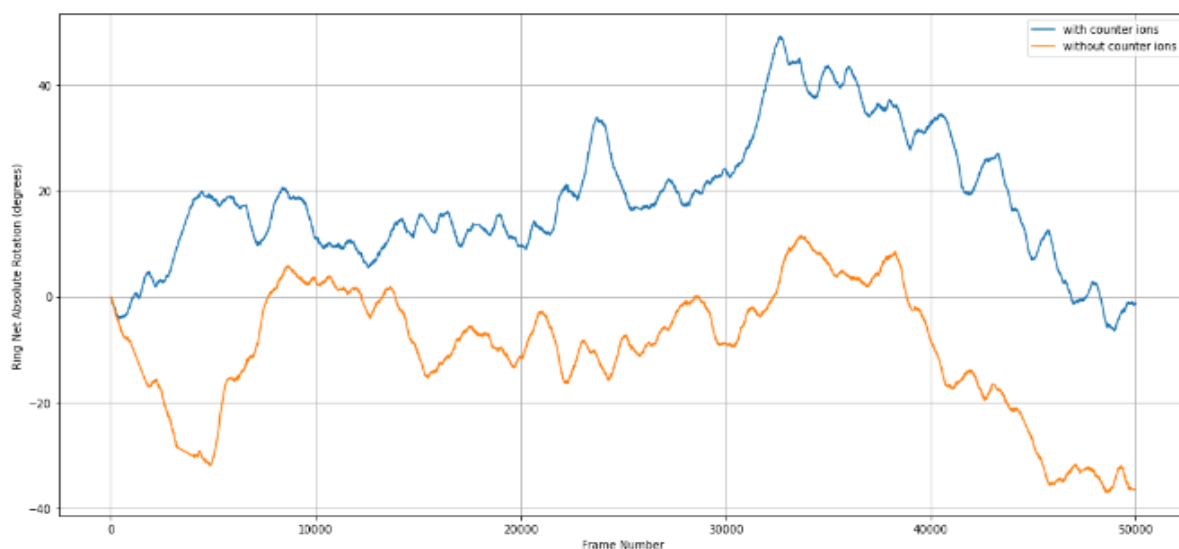


Figure 6.44. Net absolute rotation of the ring with and without counterions in the rotaxane system simulated at 1300 K without solvent.

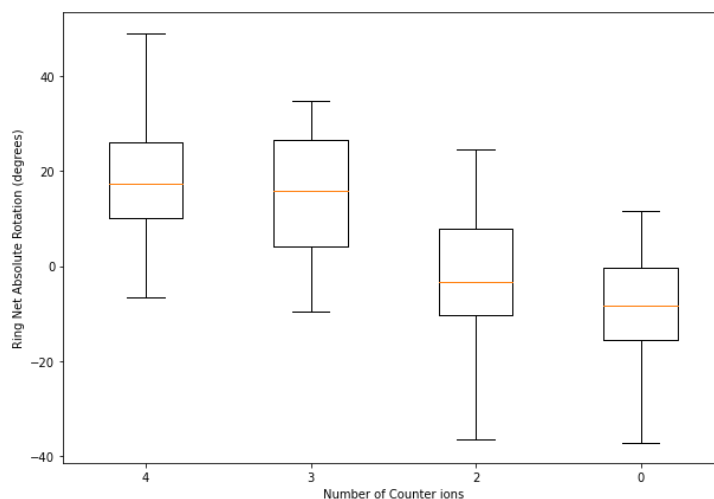


Figure 6.45. Box plots showing the distribution of net absolute rotation of the ring in rotaxane system simulated by varying the number of counterions at 1300 K without solvent.

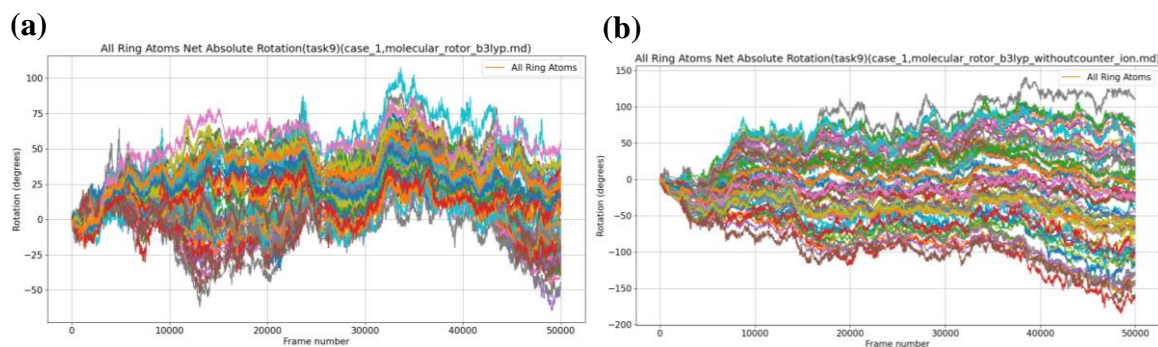


Figure 6.46. Net absolute rotation of the ring atoms in the rotaxane system simulated at 1300 K without solvent (a) with counterions. (b) without counterions.

Simultaneous Effect of the Solvent and Counterions: We also assessed the simultaneous effect of solvent and counterions on the rotation of the ring in the rotaxane system. We simulated the rotaxane system with and without solvent and counterions under identical conditions at 1300 K. Figure 6.47 depicts the net absolute rotation of the ring in the rotaxane system with and without solvent and counterions. We observed that the rotation of the ring simulated with solvent and counterions is opposite to that of the ring simulated without solvent and counterions. It can be seen that the ring with solvent and counterions has positive peaks between 0 and 20,000 time steps, whereas the ring without solvent and counterions has negative peaks in the same region. A similar but opposite trend was observed around 10,000 time step and between 30,000 and 50,000 time steps. We also observed that the magnitude of the net absolute rotation of the ring with solvent and counterions is higher than the ring without solvent and counterions for most of the time steps. From the earlier analyses, we know that the solvent reverses the direction of rotation, whereas counterions increase the magnitude of the rotation. The absence of counterions also increases the dispersion in atomic rotation computed by the algorithm (Figure 6.48). In this analysis, we saw both the effects in action. Thus, solvent and counterions appear to be exerting their effects somewhat independently on the ring in a rotaxane system.

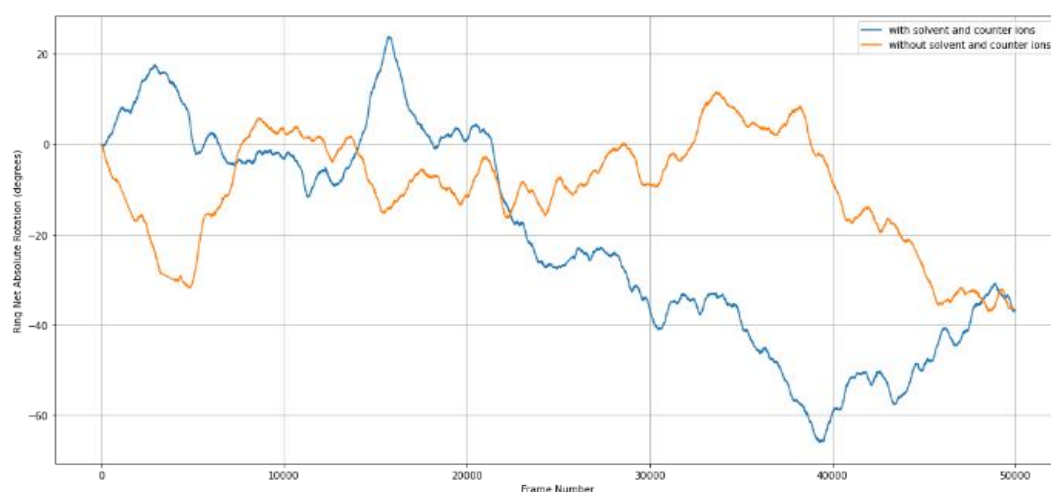


Figure 6.47. Net absolute rotation of the ring in the rotaxane system simulated with and without solvent and counterions at 1300 K.

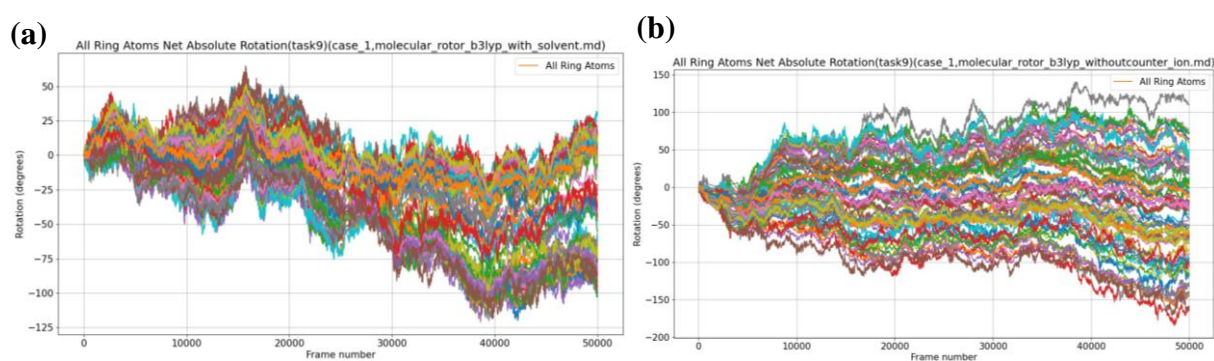


Figure 6.48. Net absolute rotation of the ring atoms in the rotaxane system simulated at 1300 K (a) with solvent and counterions. (b) without solvent and counterions.

6.4 Pseudocode of the algorithm developed for quantifying the net absolute rotation of the ring

The initial algorithm proposed in section 6.3.1 has been modified and improved several times in this study. Therefore, in this section, we outline the pseudocode of the final algorithm that has been developed through this study.

Let N be the total number of time steps (i.e., frames).

Let F be the file containing the trajectory of the molecular machine in terms of cartesian coordinates.

Let R be the net absolute rotation of the ring in degrees.

$R \leftarrow 0$

$i \leftarrow 0$

$step_size \leftarrow 10$

$ring_atom_number_list \leftarrow$ Identify the ring atom numbers from the initial time step

$track_atom_number_list \leftarrow$ Identify the track atom numbers from the initial time step

while $i < N - step_size$ **do**

$cord_1 \leftarrow$ obtain the cartesian coordinates of ring atoms at time step i

$cord_2 \leftarrow$ obtain the cartesian coordinates of ring atoms at time step $i + step_size$

 calculate the center of rotation, the axis of rotation, and the reference axis.

 align $cord_1$ and $cord_2$ such that the center of rotation lies at the origin, axis of rotation is oriented along the positive x-axis, and the reference axis is oriented along [0,1,1] direction.

if current time step is invalid **then**

 skip the current time step

else

 calculate the instantaneous rotation of the individual ring atoms.

$r \leftarrow$ calculate the instantaneous rotation of the ring by averaging the instantaneous rotation of individual ring atoms.

$R \leftarrow R + r$

end if

$i \leftarrow i + step_size$

end while

6.5 Conclusions

- In this study, we have developed an algorithm to quantify instantaneous and net absolute rotation of the ring in the molecular machines containing a ring and a track (i.e., mechanically interlocked systems). However, the algorithm can also quantify the rotation in the system containing only a ring.
- The trend observed in the absolute rotation of the ring atoms matches reasonably well with the trend observed in the absolute rotation of the ring. Thus, the algorithm also captures the rotation of atoms in the ring.
- We performed several tests to verify the algorithm using an artificial test system and a rotaxane test system.
- Unfortunately, the algorithm fails to correctly quantify the rotation of track and track atoms. Therefore, the results corresponding to the relative rotation of the ring in the rotaxane and catenane systems need further validation.
- We investigated linear regression, which is a machine learning algorithm for computing the rotation axis. Although linear regression was successful to some extent, it did not accurately predict the rotation of all ring atoms. Finally, the issue was resolved to a reasonable degree using the axis optimization strategy. We also employed a cylinder test to remove the steps containing an incorrectly orientated rotation axis when the axis optimization strategy fails. We also tried several strategies to resolve the issue related to the rotation of track atoms.
- As the algorithm can reasonably quantify the absolute rotation of the ring, we investigated the effect of various factors on the rotation of the ring in the rotaxane system.
- In the rotaxane system without a track, we often observed that some steps fail to clear the cylinder test due to distortion of the ring. Decreasing the radius of the cylinder solved this issue. It was also observed that the atomic rotation computed by the algorithm is highly dispersed in the rotaxane system without the track.
- For most of the time, we observed that the net absolute rotation of the ring with solvent was negative, whereas it was positive for the ring without solvent. We did not observe a large difference in the magnitude of the rotation. However, it was observed that the maximum magnitude of the net absolute rotation of the ring with solvent was higher than the ring without solvent. Thus, we conclude that the solvent primarily affects the direction of rotation of the ring in a rotaxane system.
- Counterions increase the value of the net absolute rotation and decrease the dispersion in atomic rotation computed by the algorithm.
- We also studied the simultaneous effect of the solvent and the counterions. It was observed that the solvent and the counterions appear to exert their effect somewhat independently on the rotation of the ring in the rotaxane system.
- We believe that the insights obtained from this study would help experimentalists develop novel molecular machines having desired rotational directionality.

6.6 References

- (1) There's Plenty of Room at the Bottom - Caltech Magazine <https://calteches.library.caltech.edu/1976/> (accessed Mar 22, 2022).
- (2) Mavroidis, C.; Dubey, A.; Yarmush, M. L. Molecular Machines. *Annual Review of Biomedical Engineering*. Annual Reviews July 15, 2004, pp 363–395. <https://doi.org/10.1146/annurev.bioeng.6.040803.140143>.
- (3) Wagoner, J. A.; Dill, K. A. Mechanisms for Achieving High Speed and Efficiency in Biomolecular Machines. *Proc. Natl. Acad. Sci. U. S. A.* **2019**, *116* (13), 5902–5907. https://doi.org/10.1073/PNAS.1812149116/SUPPL_FILE/PNAS.1812149116.SAPP.PDF.
- (4) Chambron, J. C.; Dietrich-Buchecker, C.; Hemmert, C.; Khemiss, A. K.; Mitchell, D.; Sauvage, J. P.; Weiss, J. Interlacing Molecular Threads on Transition Metals. *Pure Appl. Chem.* **1990**, *62* (6), 1027–1034. <https://doi.org/10.1351/PAC199062061027>.
- (5) Schill, G.; Henschel, R.; Neubauer, H.; Ziircher, C.; Vetter, W.; Beckmann, W.; Schweichert, N.; Fritz, H.; Harrison, I. T.; Harrison, S.; Ogino, H.; Ogina, H.; Ohata, K.; Yamanari, K.; Shimura, Y.; Rao, T. V. S.; Lawrence, D. S. A Molecular Shuttle. *J. Am. Chem. Soc.* **1991**, *113* (13), 5131–5133. <https://doi.org/10.1021/JA00013A096>.
- (6) Koumura, N.; Zijistra, R. W. J.; Van Delden, R. A.; Harada, N.; Feringa, B. L. Light-Driven Monodirectional Molecular Rotor. *Nat.* **1999**, *401* (6749), 152–155. <https://doi.org/10.1038/43646>.
- (7) Vives, G.; Tour, J. M. Synthesis of Single-Molecule Nanocars. *Acc. Chem. Res.* **2009**, *42* (3), 473–487. https://doi.org/10.1021/AR8002317/SUPPL_FILE/AR8002317_SI_002.WMV.
- (8) Nitoń, P.; Zywockiński, A.; Fiałkowski, M.; Hołyst, R. A “Nano-Windmill” Driven by a Flux of Water Vapour: A Comparison to the Rotating ATPase. *Nanoscale* **2013**, *5* (20), 9732–9738. <https://doi.org/10.1039/C3NR03496H>.
- (9) Bissell, R. A.; Córdova, E.; Kaifer, A. E.; Stoddart, J. F. A Chemically and Electrochemically Switchable Molecular Shuttle. *Nat.* **1994**, *369* (6476), 133–137. <https://doi.org/10.1038/369133a0>.
- (10) Wilson, M. R.; Solà, J.; Carlone, A.; Goldup, S. M.; Lebrasseur, N.; Leigh, D. A. An Autonomous Chemically Fuelled Small-Molecule Motor. *Nat.* **2016**, *534* (7606), 235–240. <https://doi.org/10.1038/nature18013>.
- (11) HARTKE, B. SIMULATION OF MOLECULAR MACHINES. **2021**, 261–265. https://doi.org/10.1142/9789811228216_0032.
- (12) Warshel, A. Multiscale Modeling of Biological Functions: From Enzymes to Molecular Machines (Nobel Lecture). *Angew. Chemie - Int. Ed.* **2014**, *53* (38), 10020–10031. <https://doi.org/10.1002/ANIE.201403689>.
- (13) Raeker, T.; Carstensen, N. O.; Hartke, B. Simulating a Molecular Machine in Action. *J. Phys. Chem. A* **2012**, *116* (46), 11241–11248. https://doi.org/10.1021/JP305258B/ASSET/IMAGES/JP305258B.SOCIAL.JPEG_V03.
- (14) Raeker, T.; Jansen, B.; Behrens, D.; Hartke, B. Simulations of Optically Switchable

- Molecular Machines for Particle Transport. *J. Comput. Chem.* **2018**, *39* (20), 1433–1443. <https://doi.org/10.1002/JCC.25212>.
- (15) Cnossen, A.; Kistemaker, J. C. M.; Kojima, T.; Feringa, B. L. Structural Dynamics of Overcrowded Alkene-Based Molecular Motors during Thermal Isomerization. *J. Org. Chem.* **2014**, *79* (3), 927–935. <https://doi.org/10.1021/JO402301J>.
- (16) Zhou, Y. H.; Yuan, L. Z.; Zheng, X. H. Ab Initio Study of the Transport Properties of a Light-Driven Switching Molecule Azobenzene Substituent. *Comput. Mater. Sci.* **2012**, *61*, 145–149. <https://doi.org/10.1016/J.COMMATSCI.2012.04.024>.
- (17) Bravo, J. A.; Raymo, F. M.; Stoddart, J. F.; White, A. J. P.; Williams, D. J. High Yielding Template-Directed Syntheses of [2]Rotaxanes. *European J. Org. Chem.* **1998**, *1998* (11), 2565–2571. [https://doi.org/10.1002/\(SICI\)1099-0690\(199811\)1998:11<2565::AID-EJOC2565>3.0.CO;2-8](https://doi.org/10.1002/(SICI)1099-0690(199811)1998:11<2565::AID-EJOC2565>3.0.CO;2-8).
- (18) Grabuleda, X.; Jaime, C. Molecular Shuttles. A Computational Study (MM and RID) on the Translational Isomerism in Some [2]Rotaxanes. *J. Org. Chem.* **1998**, *63* (26), 9635–9643. <https://doi.org/10.1021/JO980400T/ASSET/IMAGES/LARGE/JO980400TN00001.JPG>.
- (19) Li, H.; Li, X.; Wu, Y.; Ågren, H.; Qu, D. H. A Musclelike 2Rotaxane: Synthesis, Performance, and Molecular Dynamics Simulations. *J. Org. Chem.* **2014**, *79* (15), 6996–7004. <https://doi.org/10.1021/JO501127H>.
- (20) Grabuleda, X.; Ivanov, P.; Jaime, C. Shuttling Process in [2]Rotaxanes. Modeling by Molecular Dynamics and Free Energy Perturbation Simulations. *J. Phys. Chem. B* **2003**, *107* (31), 7582–7588. https://doi.org/10.1021/JP034658L/SUPPL_FILE/JP034658LSI20030314_111806.PDF.
- (21) Ivanov, P. Computational Study (MM and DFT) on the Conformations of Some Aromatic Crown Ether Rotaxane Macrocycles. *Comput. Theor. Chem.* **2021**, *1203*, 113266. <https://doi.org/10.1016/J.COMPTC.2021.113266>.
- (22) Liu, P.; Li, W.; Kan, Z.; Sun, H.; Ma, J. Factor Analysis of Conformations and NMR Signals of Rotaxanes: AIMD and Polarizable MD Simulations. *J. Phys. Chem. A* **2016**, *120* (4), 490–502. https://doi.org/10.1021/ACS.JPCA.5B10085/SUPPL_FILE/JP5B10085_SI_001.PDF.
- (23) Rauscher, P. M.; Rowan, S. J.; De Pablo, J. J. Topological Effects in Isolated Poly[n]Catenanes: Molecular Dynamics Simulations and Rouse Mode Analysis. *ACS Macro Lett.* **2018**, *7* (8), 938–943. https://doi.org/10.1021/ACSMACROLETT.8B00393/SUPPL_FILE/MZ8B00393_SI_001.PDF.
- (24) Vologodskii, A.; Rybenkov, V. V. Simulation of DNA Catenanes. *Phys. Chem. Chem. Phys.* **2009**, *11* (45), 10543–10552. <https://doi.org/10.1039/B910812B>.
- (25) Sohlberg, K.; Zheng, X. Computational Analysis of Switchable Rotaxanes. *Dekker Encycl. Nanosci. Nanotechnol.* **2004**. <https://doi.org/10.1201/NOE0849396397.ch82>.
- (26) Nishimura, D.; Oshikiri, T.; Takashima, Y.; Hashizume, A.; Yamaguchi, H.; Harada, A. Relative Rotational Motion between α -Cyclodextrin Derivatives and a Stiff Axle

- Molecule. *J. Org. Chem.* **2008**, 73 (7), 2496–2502. <https://doi.org/10.1021/JO702237Q>.
- (27) Gatti, F. G.; León, S.; Wong, J. K. Y.; Bottari, G.; Altieri, A.; Morales, M. A. F.; Teat, S. J.; Frochot, C.; Leigh, D. A.; Brouwer, A. M.; Zerbetto, F. Photoisomerization of a Rotaxane Hydrogen Bonding Template: Light-Induced Acceleration of a Large Amplitude Rotational Motion. *Proc. Natl. Acad. Sci. U. S. A.* **2003**, 100 (1), 10–14. <https://doi.org/10.1073/PNAS.0134757100>.
- (28) Ufimtsev, I. S.; Martinez, T. J. Quantum Chemistry on Graphical Processing Units. 3. Analytical Energy Gradients, Geometry Optimization, and First Principles Molecular Dynamics. *J. Chem. Theory Comput.* **2009**, 5 (10), 2619–2628. https://doi.org/10.1021/CT9003004/SUPPL_FILE/CT9003004_SI_001.PDF.
- (29) Ufimtsev, I. S.; Martinez, T. J. Quantum Chemistry on Graphical Processing Units. 2. Direct Self-Consistent-Field Implementation. *J. Chem. Theory Comput.* **2009**, 5 (4), 1004–1015. <https://doi.org/10.1021/CT800526S>.
- (30) Ufimtsev, I. S.; Martínez, T. J. Quantum Chemistry on Graphical Processing Units. 1. Strategies for Two-Electron Integral Evaluation. *J. Chem. Theory Comput.* **2008**, 4 (2), 222–231. <https://doi.org/10.1021/CT700268Q>.
- (31) Brown, A. I.; Sivak, D. A. Theory of Nonequilibrium Free Energy Transduction by Molecular Machines. *Chem. Rev.* **2020**, 120 (1), 434–459. https://doi.org/10.1021/ACS.CHEMREV.9B00254/ASSET/IMAGES/ACS.CHEMREV.9B00254.SOCIAL.JPEG_V03.
- (32) Gijs Schaftenaar. Molden.

Chapter 7
Summary and Future Outlook

Chapter 7

Summary and Future Outlook

7.1 Focus of this Thesis

The evolution of human civilization has been tied to the progress of science, so much so that the pre-historical periods are not named after social or economic advances but by the materials discovered during that age. This bias highlights the importance of scientific discovery for the progress of human society. Today, the discovery of antibiotics, painkillers, and anesthetics has increased our life span and revolutionized the field of medical care.¹ The discovery of alloys, semiconductors, ceramics, and polymers has transformed our society.²⁻⁵ The discovery of novel catalytic materials have advanced experimental research.⁶ The discovery of new materials has also led to scientific progress (i.e., the discovery of a new scientific phenomenon or insight into a known phenomenon). For example, the discovery of piezoelectric, ferroelectric, and superconducting effects was only possible due to the discovery of related materials.^{7,8} Molecules are building blocks of many materials. Therefore, new scientific discoveries are often associated with molecules. New discoveries require a guided exploration through the space of molecules (i.e., molecular space). The exploration is often guided by the property (i.e., desired property) that we wish to maximize or minimize. Unfortunately, molecular space is large and sparse.⁹ Thus, we need efficient strategies for its exploration. The conventional experimental and computational approaches are time-consuming and resource-intensive. Therefore, they are not suitable for the efficient exploration of large molecular space. Although approaches based on genetic and graph-theoretical algorithms have been developed, they are highly sensitive to initial conditions, parameters, and fitness functions. The inappropriate choice of any of these often leads to suboptimal solutions and meaningless molecules.^{10,11} Due to advancements in high throughput technologies, a large amount of experimental and computation data is available today. However, none of the conventional approaches use knowledge from these datasets. Furthermore, experimental, computational, and algorithmic approaches, to some extent, depend on the intuition and expertise of the researcher. Therefore, an efficient strategy capable of utilizing previous data to explore molecular space is required. Machine learning (ML) algorithms have gained a lot of attention in recent years. They have shown promising results in many fields of science and technology, surpassing the traditional approaches.¹²⁻¹⁶ ML algorithms can handle a large amount of data and find patterns in them. In this thesis work, we have demonstrated the development of strategies to explore different molecular spaces using machine learning algorithms and computational tools. However, the exploration strategy depends on the molecular space and the corresponding application. A single strategy may not work for every molecular space. Therefore, this work also demonstrates how one can develop strategies constrained to the requirements of the molecular space and the corresponding application. We have investigated three molecular spaces in this work: (i) battery materials based on phenazine molecules, (ii) biomolecules (DNA and proteins), and (ii) molecular machines. These molecular spaces were chosen due to

the need for an efficient exploration strategy in their corresponding applications. The results can be summarized as follows:

(i) Twenty linear and non-linear ML models have been investigated to predict the redox potential of phenazine derivatives in DME solvent. It was observed that models achieved excellent prediction accuracy on the test-set (i.e., $R^2 > 0.98$, $MSE < 0.008 \text{ V}^2$, and $MAE < 0.07 \text{ V}$). The molecular features used in this work do not require DFT calculations or experimental measurements, making our approach readily adaptable for similar studies. Model performance was assessed on four feature sets containing different types of molecular features (i.e., 2D, 3D, and molecular fingerprints). The analysis revealed an interesting order with respect to prediction accuracy: $2d > 2d+3d+fp > 3d > fp$. Average performance analysis also showed that 2D molecular features are better at predicting the redox potential of phenazine derivatives than 3D and molecular fingerprint features. Feature importance analysis also showed that 2D molecular features are more informative than 3D and molecular fingerprints. Due to the short prediction time compared to DFT and high accuracy, the ML models developed in this work could be employed for exploring a large molecular space of phenazine derivatives, containing a single functional group per molecule for better designed battery materials.

(ii) Next, four ML models were investigated to develop a hybrid DFT-ML strategy for the exploration of molecular space containing phenazine-based battery materials. High prediction accuracy on the unseen data was achieved using a small training set of 151 molecules. This work extends the range of molecular space that can be explored beyond the type of molecules used for training. We showed that despite being trained on the derivatives with a single type of functional group and only 2D molecular features, the ML models were able to predict the redox potentials of the derivatives containing multiple and different types of functional groups with good accuracies ($R^2 > 0.7$). We also investigated the effect of different structural features on the redox potential through feature importance analysis. Furthermore, we have identified promising phenazine derivatives for the anolyte in RFBs from the unseen dataset. This work has implications for the discovery of new green battery materials. The hybrid DFT-ML approach demonstrated in this work would help in accelerating the exploration of green battery materials such as phenazine derivatives.

(iii) In this work, we have investigated three unsupervised machine learning models (LDA, HDP, and NPLB) to explore the molecular space containing DNA regions obtained from the ChIP-seq and the DNase-seq data of the K562 cell line. This study aimed to identify the functional relationships between different DNA regions and proteins (i.e., transcription factors) rather than discover new molecules. The functional relationship between different DNA molecules and transcription factors (TFs) cannot be represented in a mathematical form. Therefore, strategies based on unsupervised methods are more suitable for this task. We clustered DNA regions into different groups using LDA, HDP, and NPLB. In most cases, it was observed that the discovered clusters (modules) represent a group of commonly interacting co-binding TFs and some regulatory modules found in the K562 cell line. LDA and HDP also identified some regulatory modules from the DNase-seq data without prior information on the binding patterns. NPLB showed more robust performance than topic models (LDA and HDP) due to its ability to ignore the noise in the data. The unsupervised machine learning approach

developed in this study has the potential to identify new regulatory modules and genetic determinants of the diseases.

(iv) Next, we developed an algorithmic strategy to investigate rotational motion in molecular machines containing a ring and a track. The molecular space in this work includes different configurations of the rotaxane and catenane. The main issue we faced in this work was a lack of methodology for quantifying rotational motion (desired property) in molecular machines directly from the MD trajectory. For the development, we analyzed the *ab initio* MD trajectory of rotaxane and catenane. We also investigated linear regression, a machine learning algorithm for identifying the rotational axis. The algorithm developed here can quantify instantaneous and net absolute rotation of the ring in rotaxane, catenane, and systems containing only a ring. Extensive verification of the algorithm was carried out using an artificial test system and a rotaxane test system. Furthermore, we analyzed the effect of the track, solvent, and counterions on the absolute rotation of the ring. It was observed that the ring distorts in the absence of the track; the solvent affects the direction of the rotation, and the counterions affect the magnitude of the rotation of the ring in a rotaxane system.

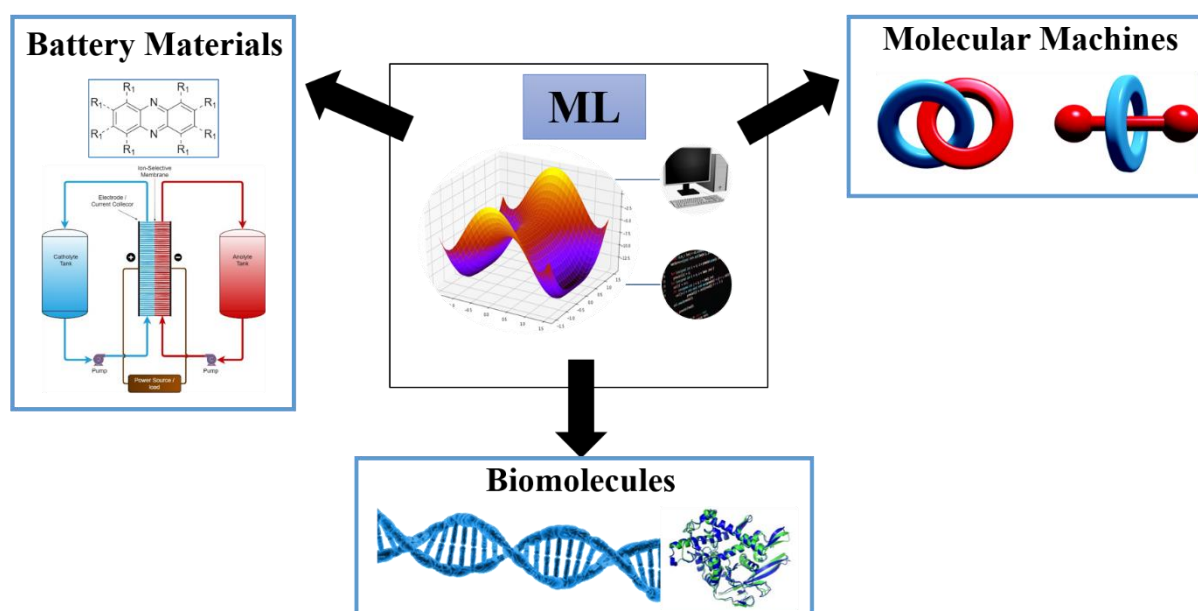


Figure 7.1. A representation of the research work presented in the thesis.

7.2 Future Outlook

Insights obtained from work presented in this thesis shed light on the critical areas of research and are likely to help experimentalists design and discover novel molecules. The developments demonstrated in this thesis work would help researchers from various fields, including experimentalists, develop machine learning based exploration strategies for their respective molecular spaces and applications for scientific discoveries. Machine learning models often need a large amount of data that may not be readily available in many scientific domains, particularly for new applications. This work demonstrates that it is possible to develop reasonably accurate ML models even with small datasets that generalize to unseen datasets. By investigating different molecular spaces, we have shown the applicability of both supervised and unsupervised learning models in the development of exploration strategies. The desired

property guides the exploration through molecular space. However, a methodology to compute the desired property is not always readily available. We encountered such a situation during the exploration of molecular machines and tried to develop an algorithm for quantifying the desired quantity. Although the development was not general, it demonstrates that some elements from machine learning models may be useful in algorithmic development. This thesis work also demonstrates an attempt to develop combined strategies such as computational-ML and algorithmic-ML.

Given the negative impact of energy generation from fossil fuels on the climate, green and efficient battery materials are urgently required for sustainable development. The ML models developed for predicting the redox potentials of phenazine derivatives could screen thousands of new phenazine derivatives in a short amount of time, which otherwise would require months using experimental and computational approaches. The ML models developed here showed good generalizability and do not require the computation of expensive molecular features. Therefore, one can screen thousands of phenazine derivatives to identify potential candidates that can be evaluated further using experimental or computational approaches. Using machine learning models as a first step in the screening would avoid the wastage of valuable resources on useless molecules. Thus, the hybrid DFT-ML approach has the potential to accelerate the discovery of novel green battery materials for RFBs. It is also possible to extend the range of applicable molecular spaces and prediction accuracy by adding molecules containing diverse and more functional groups in various solvents. Further reduction in error is possible by adding phenazine derivatives with positive redox potential. The hybrid DFT-ML approach demonstrated in this work could easily be adopted in other areas for the discovery of novel molecules.

Identifying genetic factors responsible for diseases requires insight into DNA-DNA, DNA-TF, and TF-TF interactions. Due to inherent stochasticity in these interactions, insight into them needs multiple sources of data or prior information. The unsupervised models (topic models and NPLB) developed in this thesis work would prove valuable in identifying regulatory modules directly from the DNase-seq data without prior information. The models also identified regulatory modules from ChIP-seq datasets, showing their applicability to other datasets. In particular, NPLB showed more robust performance on ChIP-seq datasets than topic models. However, the wider applicability of NPLB requires improvement in its speed. These models could also be applied to other cell types and species for identifying regulatory modules directly from the DNA regions. A comparative study using these models on diseased and normal cells would help identify regulatory elements correlated with disease.

The algorithm developed for quantifying the rotational motion of the ring would help researchers understand rotational dynamics in molecular machines. The insight obtained from analyzing the effect of different factors on the rotation of the ring in a rotaxane system would help experimentalists design and develop novel molecular machines with desired rotational directionality. It was observed that the solvent reverses the direction of rotation of the ring in a rotaxane system. So it might be possible to design a molecular machine in which the direction of rotation of the ring could be changed by adding or removing the solvent. Similarly, it would be possible to control the magnitude of the rotation of the ring by changing the number of counterions in the rotaxane system. The developed algorithm does not work for the track, so

future work might try to address this issue. It would help quantify the relative rotation of the ring with respect to the track, which might be more important in some applications.

7.3 References

- (1) Juffermans, N. P.; Radermacher, P.; Laffey, J. G. The Importance of Discovery Science in the Development of Therapies for the Critically Ill. *Intensive Care Med. Exp. 2020 81* **2020**, 8 (1), 1–7. <https://doi.org/10.1186/S40635-020-00304-4>.
- (2) Wood, J. The Top Ten Advances in Materials Science. *Mater. Today* **2008**, 11 (1–2), 40–45. [https://doi.org/10.1016/S1369-7021\(07\)70351-6](https://doi.org/10.1016/S1369-7021(07)70351-6).
- (3) Fortunato, E.; Barquinha, P.; Martins, R. Oxide Semiconductor Thin-Film Transistors: A Review of Recent Advances. *Adv. Mater.* **2012**, 24 (22), 2945–2986. <https://doi.org/10.1002/ADMA.201103228>.
- (4) Namazi, H. Polymers in Our Daily Life. *Bioimpacts* **2017**, 7 (2), 73. <https://doi.org/10.15171/BI.2017.09>.
- (5) NASA - Superhero Ceramics! https://www.nasa.gov/missions/science/spinoff9_nextel_f.html (accessed Jan 2, 2022).
- (6) Áejka, J.; Nachtigall, P.; Centi, G. New Catalytic Materials for Energy and Chemistry in Transition. *Chem. Soc. Rev.* **2018**, 47 (22), 8066–8071. <https://doi.org/10.1039/C8CS90119H>.
- (7) Haertling, G. H. Ferroelectric Ceramics: History and Technology. *J. Am. Ceram. Soc.* **1999**, 82 (4), 797–818. <https://doi.org/10.1111/J.1151-2916.1999.TB01840.X>.
- (8) Van Delft, D.; Kes, P. The Discovery of Superconductivity. *Phys. Today* **2010**, 63 (9), 38–43.
- (9) Drew, K. L. M.; Baiman, H.; Khwaounjoo, P.; Yu, B.; Reynisson, J. Size Estimation of Chemical Space: How Big Is It? *J. Pharm. Pharmacol.* **2012**, 64 (4), 490–495. <https://doi.org/10.1111/J.2042-7158.2011.01424.X>.
- (10) Yang, X.-S. Genetic Algorithms. *Nature-Inspired Optim. Algorithms* **2021**, 91–100. <https://doi.org/10.1016/B978-0-12-821986-7.00013-5>.
- (11) Katoch, S.; Chauhan, S. S.; Kumar, V. A Review on Genetic Algorithm: Past, Present, and Future. *Multimed. Tools Appl.* **2021**, 80 (5), 8091–8126. <https://doi.org/10.1007/S11042-020-10139-6/FIGURES/8>.
- (12) Fujiyoshi, H.; Hirakawa, T.; Yamashita, T. Deep Learning-Based Image Recognition for Autonomous Driving. *IATSS Res.* **2019**, 43 (4), 244–252. <https://doi.org/10.1016/J.IATSSR.2019.11.008>.
- (13) Raccuglia, P.; Elbert, K. C.; Adler, P. D. F.; Falk, C.; Wenny, M. B.; Mollo, A.; Zeller, M.; Friedler, S. A.; Schrier, J.; Norquist, A. J. Machine-Learning-Assisted Materials Discovery Using Failed Experiments. *Nat. 2016 5337601* **2016**, 533 (7601), 73–76. <https://doi.org/10.1038/nature17439>.
- (14) Nielsen, M. K.; Ahneman, D. T.; Riera, O.; Doyle, A. G. Deoxyfluorination with Sulfonyl Fluorides: Navigating Reaction Space with Machine Learning. *J. Am. Chem. Soc.* **2018**, 140 (15), 5004–5008.

https://doi.org/10.1021/JACS.8B01523/SUPPL_FILE/JA8B01523_SI_002.ZIP.

- (15) Lecun, Y.; Bengio, Y.; Hinton, G. ; Gregor K And Lecun, Y.; Icm1 ; F, K. D.; Philbin, J.; Cvpr ; Schuster, M.; Chen, Z. PERSPECTIVES Special Topic: Machine Learning Deep Learning for Natural Language Processing: Advantages and Challenges. *11*. Sprechmann P, Bronstein AM Sapiro G. *IEEE TPAMI* **2018**, 5 (1), 22–24. <https://doi.org/10.1093/nsr/nwx099>.
- (16) Hähnel, P.; Mareček, J.; Monteil, J.; O'Donncha, F. Using Deep Learning to Extend the Range of Air Pollution Monitoring and Forecasting. *J. Comput. Phys.* **2020**, 408, 109278. <https://doi.org/10.1016/J.JCP.2020.109278>.

ABSTRACT

Name of the Student: Ghule Siddharth Sambhaji **Registration No. :** 10CC17A26010
Faculty of Study: Chemical Science **Year of Submission:** 2022
AcSIR academic centre/CSIR Lab: CSIR-National **Name of the Supervisor(s):** Dr. Kumar
Chemical Laboratory Vanka

Title of the thesis: Computational Development of the Strategies to Explore Molecular Machines and the Molecular Space for Desired Properties using Machine Learning

For thousands of years, scientific discoveries have played a vital role in the progress of human civilization. The discovery of new materials or new scientific phenomena, or an improved understanding of the known phenomena requires exploration through the space available for a given class of molecules (the molecular space). The typical size of molecular space is estimated to be $\sim 10^{60}$, which is larger than the number of stars in the observable universe ($\sim 10^{24}$). Conventional experimental, computational, and algorithmic approaches are inefficient in exploring this vast molecular space. Furthermore, conventional exploration strategies do not take advantage of the large databases available today. On the other hand, machine learning (ML) algorithms can extract hidden knowledge from large datasets. They have shown excellent predictive accuracies in many fields, surpassing the traditional methods. Thus, ML algorithms are promising candidates for developing efficient exploration strategies for the vast molecular space. In this thesis work, we have demonstrated the development of exploration strategies using machine learning algorithms for three different molecular spaces. The first molecular space investigated in this thesis includes battery materials based on phenazine molecules. We have developed an accurate hybrid DFT-ML approach to explore this molecular space. We showed that 2D molecular features are most informative in predicting the redox potential of phenazine derivatives in DME. We also showed that it is possible to develop reasonably accurate machine learning models for complex quantities such as redox potential using small and simple datasets. Next, we investigated different unsupervised machine learning algorithms to explore the molecular space of DNA and proteins to uncover the interactions between them. We have shown that unsupervised machine learning models can discover commonly occurring regulatory modules containing interacting and co-binding transcription factors without prior information on binding activities. Sometimes, in fundamental research, one may encounter the desired property, which cannot be easily computed using existing methodologies. We faced this issue during the investigation of molecular machines. Therefore, we developed an algorithm for quantifying the desired property (i.e., rotational motion) of the ring in the molecular machines. We also investigated linear regression, a machine learning algorithm, during the development. The developed algorithm helped us get an insight into different factors responsible for the rotational directionality of the ring in the rotaxane system. Thus, this thesis work demonstrates the applicability of machine learning and computational tools to the development of efficient exploration strategies for molecular space. This work also shows how to address different issues one may encounter during the development. Furthermore, the specific strategies developed for three molecular spaces are valuable for discovering new molecules and new scientific phenomena. For example, the hybrid DFT-ML approach can help discover promising phenazine derivatives for green energy storage systems such as RFB. The unsupervised machine learning approach developed in this study has the potential to identify genetic determinants of diseases. The algorithm developed for quantifying rotation would help experimentalists develop novel molecular machines having rotational directionality.

Details of the publications emanating from the thesis work

1. List of publication(s) in SCI Journal(s) (published & accepted) emanating from the thesis work, with complete bibliographic details.

(i) **Siddharth Ghule***, Soumya Ranjan Dash, Sayan Bagchi, Kavita Joshi*, Kumar Vanka*. "Predicting the Redox Potentials of Phenazine Derivatives using DFT Assisted Machine Learning" *ACS Omega*, **2022**; DOI: 10.1021/acsomega.1c06856.

2. List of Papers with abstract presented (oral/poster) at national/international conferences/seminars with complete details.

(i) Poster Presentation:

Title: Directionality in Molecular Machines

Date: 25-02-2020 to 28-02-2020

Location: Science Day 2020 at CSIR NCL, Pune

Abstract:

Molecular Machines are ubiquitous in living organisms. They carry out tasks essential for survival. Some examples of complex molecular machines found in the living organisms are DNA polymerases which is responsible for the DNA replication task, ATP synthase, which generates energy, Kinesin which transports molecules inside the cell¹. There are many other molecular machines which are working tirelessly round the clock to keep our body alive. One of their essential characteristics is that these machines are generally more efficient than their macroscale counterparts². Molecular machines are capable of performing the complex task at a molecular level, and if we could make molecular machines to carry out the task that we desire, we could revolutionise the whole health care industry. Many diseases such as cancer could be cured. Owing to such a potential, many researchers around the world have been engaged in the synthesis of these molecular machines. However, there are many challenges before they could successfully synthesise world-changing artificial molecular machines. Nevertheless, researchers have found some success in the synthesis of a few simple molecular machines. In 2016 Nobel Prize in Chemistry was awarded to Jean-Pierre Sauvage, Sir J. Fraser Stoddart and Bernard L. Feringa for the design and synthesis of molecular machines¹. In the same year, David Leigh group also synthesised autonomous chemically fueled molecular motors³. Molecular machines could successfully perform these complex functions because their motion is directional. Kinesin could transport molecules inside the cell because its motion has a specific direction. Random motion cannot carry out this task efficiently. So, any molecular machine be it biological or artificial should be directional. In this work, we investigate this important property(directionality) of a few artificial molecular machines. We categorise directionality either as translational or rotational. Translational directionality has been shown to exist in these molecules^{1,3}. There have been no reports on rotational directionality. Here we discuss, a method which we have developed to calculate the rotational directionality in molecular machines. Furthermore, we use rotational directionality to calculate the percentage of total kinetic energy present in the system as rotational kinetic energy. We call this number Efficiency.

References:

1. https://en.wikipedia.org/wiki/Molecular_machine
2. (PDF) Molecular Machines. Available from: https://www.researchgate.net/publication/8453380_Molecular_Machines [accessed Dec 01 2018].
3. Leigh D.A. et al., Nature, 2016, 534 , 235

(ii) Oral Presentation:

Title: Predicting the Redox Potentials of Phenazine Derivatives using DFT Assisted Machine Learning

Date: 18-11-2021 to 19-11-2021

Location: 13th European Conference on Computational and Theoretical Chemistry organized by European Chemical Society

Abstract:

Here, four machine-learning models were employed to predict the redox potentials of phenazine derivatives in DME using DFT. A small dataset of 189 phenazine derivatives having only one type of functional group per molecule (20 unique groups) was used for the training. Models were validated on the external test-set containing new functional groups and diverse molecular structures and achieved reasonable accuracies (up to $R^2=0.77$). Despite being trained on the molecules with a single type of functional group, models were able to predict the redox potentials of derivatives containing multiple and different types of functional groups with reasonable accuracy ($R^2 > 0.6$). This type of performance for predicting redox potential from such a small and simple dataset of phenazine derivatives has never been reported before. Redox Flow Batteries (RFBs) are emerging as promising candidates for energy storage systems. However, new green and efficient materials are required for their widespread usage. We believe that the hybrid DFT-ML approach demonstrated in this report would help in accelerating the virtual screening of phenazine derivatives saving computational and experimental resources. This approach could potentially identify novel molecules for green energy storage systems such as RFB.

(iii) Poster Presentation:

Title: Machine Learning the Redox Potentials of Phenazine Derivatives: A Comparative Study on Molecular Features

Date: 11-12-2021 to 14-12-2021

Location: Theoretical Chemistry Symposium-2021 organized by IISER Kolkata, IACS Kolkata, Kalyani University and S.N. Bose National Centre for Basic Sciences Kolkata

Abstract:

Redox Flow Batteries (RFBs) are promising candidates for green and efficient energy storage systems. However, Their widespread adoption still needs further investigations into cheaper and greener alternative organic redox-active species^{1,2}. In this work, we have developed machine-learning models to predict the redox potential of phenazine derivatives in DME (dimethoxyethane) solvent using a small dataset of 185 molecules³. 2D, 3D, and molecular fingerprint features were computed using readily available and

easy-to-use Python libraries, making our approach easily adaptable to similar work⁴. Twenty linear and non-linear machine-learning models were investigated in this work. These models achieved excellent performance on the unseen data (i.e., $R^2 > 0.98$, $MSE < 0.008 \text{ V}^2$ and $MAE < 0.07 \text{ V}$). Model performance was assessed consistently using the training and evaluation “pipeline” method developed in this work. We showed that 2D molecular features were most informative and achieved the best prediction accuracy among four feature sets. We also showed that often less preferred but relatively faster linear models could perform better than non-linear models when the feature set contains different types of features (i.e., 2D, 3D, and molecular fingerprints). Further investigations revealed that it is possible to reduce the training and inference time without sacrificing prediction accuracy by using a small subset of features. Moreover, models were able to predict the previously reported promising redox-active compounds with high accuracy. Also, significantly low prediction errors were observed for most functional groups. Thus, we believe the results obtained in this report would help in the adoption of green energy by accelerating the field of materials discovery for energy storage applications.

Reference:

1. S. Shafiee, E. Topal, When will fossil fuel reserves be diminished?, *Energy Policy*. 37 (2009) 181–189.
2. T.M. Gür, Review of electrical energy storage technologies, materials and systems: Challenges and prospects for large-scale grid storage, *Energy Environ. Sci.* 11 (2018) 2696–2767.
3. C. De La Cruz, A. Molina, N. Patil, E. Ventosa, R. Marcilla, A. Mavrandonakis, New insights into phenazine-based organic redox flow batteries by using high-throughput DFT modelling, *Sustain. Energy Fuels*. 4 (2020) 5513–5521.
4. G. Landrum, RDKit: Open-source cheminformatics, (n.d.). <https://www.rdkit.org/>.

(iv) Poster Presentation:

Title: Machine Learning the Redox Potentials of Phenazine Derivatives: A Comparative Study on Molecular Features

Date: 15-12-2021 to 18-12-2021

Location: Symposium on "Recent Advances in Modelling Rare Events (RARE2021)" organized by IIT Kanpur

Abstract:

Redox Flow Batteries (RFBs) are promising candidates for green and efficient energy storage systems. However, Their widespread adoption still needs further investigations into cheaper and greener alternative organic redox-active species^{1,2}. In this work, we have developed machine-learning models to predict the redox potential of phenazine derivatives in DME (dimethoxyethane) solvent using a small dataset of 185 molecules³. 2D, 3D, and molecular fingerprint features were computed using readily available and easy-to-use Python libraries, making our approach easily adaptable to similar work⁴. Twenty linear and non-linear machine-learning models were investigated in this work. These models achieved excellent performance on the unseen data (i.e., $R^2 > 0.98$, $MSE < 0.008 \text{ V}^2$ and $MAE < 0.07 \text{ V}$). Model performance was assessed consistently using

the training and evaluation “pipeline” method developed in this work. We showed that 2D molecular features were most informative and achieved the best prediction accuracy among four feature sets. We also showed that often less preferred but relatively faster linear models could perform better than non-linear models when the feature set contains different types of features (i.e., 2D, 3D, and molecular fingerprints). Further investigations revealed that it is possible to reduce the training and inference time without sacrificing prediction accuracy by using a small subset of features. Moreover, models were able to predict the previously reported promising redox-active compounds with high accuracy. Also, significantly low prediction errors were observed for most functional groups. Thus, we believe the results obtained in this report would help in the adoption of green energy by accelerating the field of materials discovery for energy storage applications.

References:

1. S. Shafiee, E. Topal, When will fossil fuel reserves be diminished?, *Energy Policy*. 37 (2009) 181–189.
2. T.M. Gür, Review of electrical energy storage technologies, materials and systems: Challenges and prospects for large-scale grid storage, *Energy Environ. Sci.* 11 (2018) 2696–2767.
3. C. De La Cruz, A. Molina, N. Patil, E. Ventosa, R. Marcilla, A. Mavrandonakis, New insights into phenazine-based organic redox flow batteries by using high-throughput DFT modelling, *Sustain. Energy Fuels*. 4 (2020) 5513–5521.
4. G. Landrum, RDKit: Open-source cheminformatics, (n.d.). <https://www.rdkit.org/>.

3. A copy of all SCI publication(s), emanating from the thesis, to be bound at the end of the thesis.

Predicting the Redox Potentials of Phenazine Derivatives Using DFT-Assisted Machine Learning

Siddharth Ghule,* Soumya Ranjan Dash, Sayan Bagchi, Kavita Joshi,* and Kumar Vanka*

Cite This: *ACS Omega* 2022, 7, 11742–11755

Read Online

ACCESS |



Metrics & More

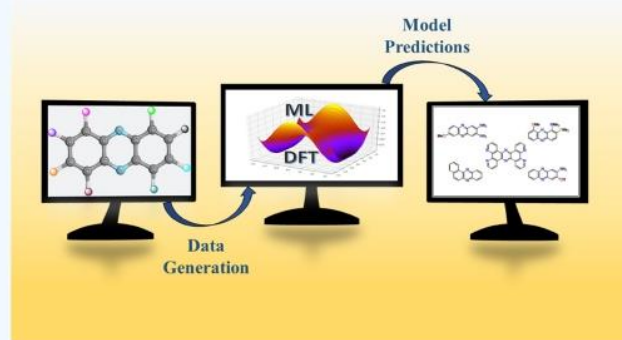


Article Recommendations



Supporting Information

ABSTRACT: This study investigates four machine-learning (ML) models to predict the redox potentials of phenazine derivatives in dimethoxyethane using density functional theory (DFT). A small data set of 151 phenazine derivatives having only one type of functional group per molecule (20 unique groups) was used for the training. Prediction accuracy was improved by a combined strategy of feature selection and hyperparameter optimization, using the external validation set. Models were evaluated on the external test set containing new functional groups and diverse molecular structures. High prediction accuracies of $R^2 > 0.74$ were obtained on the external test set. Despite being trained on the molecules with a single type of functional group, models were able to predict the redox potentials of derivatives containing multiple and different types of functional groups with good accuracies ($R^2 > 0.7$). This type of performance for predicting redox potential from such a small and simple data set of phenazine derivatives has never been reported before. Redox flow batteries (RFBs) are emerging as promising candidates for energy storage systems. However, new green and efficient materials are required for their widespread usage. We believe that the hybrid DFT-ML approach demonstrated in this report would help in accelerating the virtual screening of phenazine derivatives, thus saving computational and experimental costs. Using this approach, we have identified promising phenazine derivatives for green energy storage systems such as RFBs.



1. INTRODUCTION

Today, ~85% of the world's energy demand is being fulfilled by fossil fuels.^{1,2} The limited supply of fossil fuels and the ever-increasing population have raised concerns that we might run out of fossil fuels sooner than expected.^{1,3} Furthermore, electricity production from fossil fuels is one of the major factors responsible for greenhouse gas emissions.⁴ In this age, humanity faces two major challenges of balancing increased energy demand while reducing the environmental impact associated with energy production. In the past decades, investments and research efforts in the green technology have been increased to overcome these challenges.⁵ Significant progress has already been made to access renewable energy sources.^{6,7} Renewable energy sources, being intermittent, require efficient energy storage.⁴ Improvements in the energy storage technology would not only help in the adoption of renewable energy but also help in making efficient use of non-renewable energy sources. Historically, it has been more expensive to store energy than to expand energy generation for handling increased demand.⁸ Thus, grid systems employed today are likely to fail when additional energy cannot be generated during peak demand. The massive Texas Blackout in February 2021 is an example of such a failure.⁹ It suggests that an efficient energy storage technology is urgently required. Unfortunately, only 1.0% of the energy consumed worldwide

can be stored with the energy storage technology accessible today.¹⁰ Furthermore, the contribution of electrochemical batteries to energy storage capacity is less than 2.0%, even though most of the devices we use every day include batteries.^{8,10} Li-ion batteries are widely used today due to their high energy density, high specific energy, long cycle life, and fast charge–discharge cycle.^{4,8,11} Unfortunately, Li-ion batteries suffer from high production costs, safety issues, and high environmental impact.^{2,12} Redox flow batteries (RFBs) have the potential to overcome drawbacks of Li-ion batteries, owing to their high storage capacity, independent control over storage capacity and power, fast responsiveness, ease of scaling, room-temperature operation, cost-effectiveness, high round trip efficiency, safety, and lower environmental impact.^{13–26} RFBs are increasingly being used as energy storage devices in renewable energy systems, thereby helping in the adoption of green energy.^{15,22} A schematic diagram of the typical RFB is

Received: December 4, 2021

Accepted: February 7, 2022

Published: March 29, 2022



shown in Figure 1. The RFB consists of two storage tanks containing cathode and anode redox-active species dissolved in

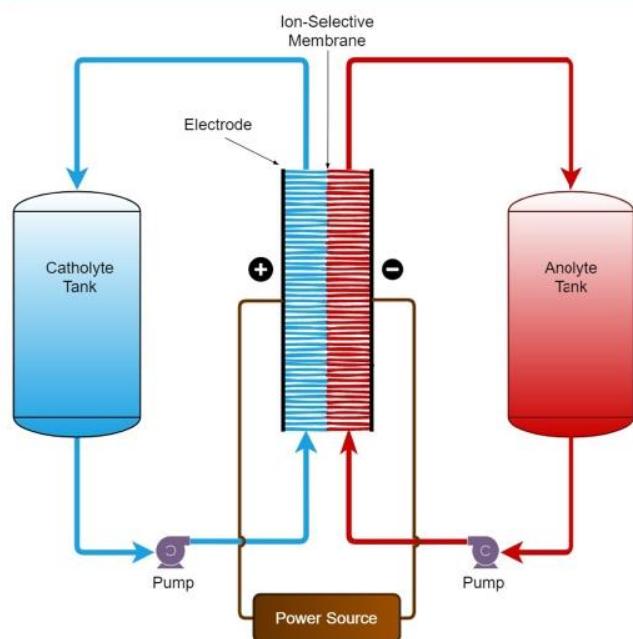


Figure 1. Schematic diagram of a typical RFB.

an electrolyte solution. The electrolyte solution in the positive and negative compartments is termed catholyte and anolyte, respectively. These storage tanks are connected to an electrochemical cell (or current collector) *via* pumps. The electrochemical cell consists of porous electrodes separated by an ion-selective membrane. During operation, electrolytes containing redox-active species are pumped to the electrochemical cell, where they undergo oxidation or reduction depending on the charge/discharge cycle. Then, electrolytes are circulated back to their storage tanks.^{13,24} So far, transition metal-based RFBs (such as vanadium, iron, and chromium) have found some commercial success. However, their widespread adoption has been limited mainly due to high production cost, toxicity, and cell component corrosion associated with the use of transition-metal salts.^{27,28} Therefore, RFBs containing organic redox-active species are being heavily investigated due to their low production cost, access to a massive space of electroactive compounds, and low environmental impact.^{28,29} Many organic compounds such as quinones, viologens, flavins, thiazines, imides, and their derivatives have been investigated for redox-active species in both aqueous and non-aqueous RFBs.^{27,30,31} However, non-aqueous RFBs offer large operating voltage.³⁰ Recently, phenazine derivatives have been shown to be promising redox-active candidates in non-aqueous RFBs. Recent reports have revealed why phenazine derivatives are promising redox-active candidates. Romadina et al. synthesized phenazine derivatives having significantly negative redox potential.³² RFBs require anolytes with high negative redox potential. They showed that the non-aqueous RFB based on the synthesized phenazine derivative is capable of achieving a potential of 2.3 V, high capacities, >95% Coulombic efficiency, and good charge–discharge cycling stability after the initial 20 cycles. Mavrandonakis and co-workers, in their computational investigation, reported the most negative redox-active

candidate based on phenazine for non-aqueous RFBs.²⁷ They showed that tetra-amino-phenazine has 140 mV more negative potential than *N*-methylphthalimide (MePht), which has one of the most negative redox potentials reported so far in RFBs.³³ They also proposed all-phenazine RFB reaching a high potential of 2.83 V. Furthermore, the redox potential of phenazine derivatives could be tuned easily with the addition of appropriate electron-donating or electron-withdrawing functional groups. The synthesis of phenazine derivatives is very economical than mining transition metals. Therefore, phenazine derivatives are currently being investigated as candidates for novel redox-active species.^{27,32}

These investigations remain primarily experimental. Unfortunately, the vast chemical space offered by organic compounds cannot be explored using experimental procedures. Quantum mechanical density functional theory (DFT) computations have been used heavily in materials science research due to high accuracy but are very slow and cannot screen millions of molecules in a reasonable amount of time. Therefore, a fast and reliable method to screen millions of compounds without compromising accuracy is required. In this regard, machine-learning (ML) algorithms have shown excellent predictive accuracies along with short development and prediction times.^{34–38} Therefore, ML models have been used extensively to screen millions of molecules in materials science and drug discovery.^{39–43} ML models generally require a large amount of data for accurate predictions. When the quantity of data is limited, feature engineering is employed to generate the most informative features. These features are expected to capture the appropriate molecular information necessary to predict the target quantity. Feature engineering requires domain knowledge, relying on having access to experts.^{44–46} In small data sets, DFT-based or experimentally determined features have been used due to their high accuracy. However, some reports also explore simple features based on the molecular structure.^{47–52}

In this work, we investigated four ML models to predict the redox potentials of phenazine derivatives in the dimethoxyethane (DME) solvent. The training-set containing 151 phenazine derivatives was obtained from the previously reported DFT study having 189 phenazine derivatives with only one type of functional group per molecule (20 unique functional groups).²⁷ Molecular features were computed from the optimized neutral structures using the RDKit python library.⁵³ Model accuracy was improved through feature selection and hyperparameter optimization using the external validation set. Then, the model performance was assessed on the external test-set compiled from the literature consisting of new functional groups, multiple functional groups, and diverse structures. Their redox potential was computed using the DFT. The trained models were employed to predict the redox potentials of randomly generated phenazine derivatives with multiple functional groups. We also carried out feature importance analysis and discussed the structure–functional relationship of phenazine derivatives. Finally, promising candidates were identified for the anolyte from the external test-set and multiple functional group test-sets.

2. MATERIALS AND METHODS

2.1. Computational Details. The redox potentials of phenazine derivatives were computed using the DFT workflow described in the paper by Mavrandonakis et al.²⁷ All the DFT calculations were performed with Gaussian 09 software.⁵⁴

Table 1. Representative Structures from Training-Set/Internal Test-Set^a

Mol ID: 1	Mol ID: 3	Mol ID: 5
Mol ID: 48	Mol ID: 52	Mol ID: 172

^aMol IDs were assigned to identify derivatives from the corresponding data set.

Table 2. Representative Structures from the External Test-Set^a

Mol ID: 1	Mol ID: 3	Mol ID: 5
Mol ID: 15	Mol ID: 17	Mol ID: 28

^aMol IDs were assigned to identify derivatives from the corresponding data set.

Geometry optimization of neutral and reduced forms of phenazine and its derivatives were carried out in the gas phase by employing B3LYP/6-31+G(d,p) level of theory.^{55–58} Harmonic frequency analysis was performed for all the structures to confirm them as minima. Solvation effects of DME were incorporated during the single-point calculations using the M06-2X functional,⁵⁹ by employing the SMD solvation model (details in the Supporting Information).^{60,61} The term “Redox Potential” in this report corresponds to the “Reduction Potential” with respect to unsubstituted phenazine molecule (i.e., the parent phenazine). The redox potentials of phenazine derivatives were computed using the following equations:



$$E_1^0 = -\frac{\Delta G_{(\text{rxn},\text{sol})}}{nF} + E_{1(\text{ref})}^0 \quad (2)$$

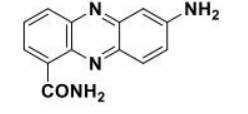
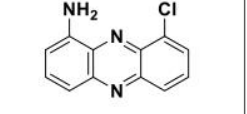
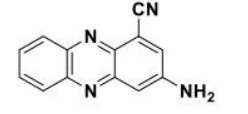
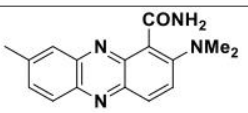
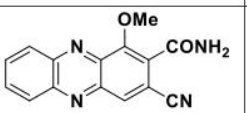
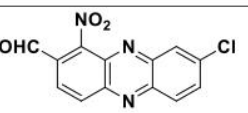
$$\Delta G_{(\text{rxn},\text{sol})} = G_{([\text{XPZ}]^-,\text{sol})}^0 + G_{([\text{PZ}],\text{sol})}^0 - G_{([\text{PZ}]^-,\text{sol})}^0 - G_{([\text{XPZ}],\text{sol})}^0 \quad (3)$$

$$G_{(\text{sol})}^0 = G_{(\text{them},\text{gas})}^{(\text{B3LYP})} + E_{(\text{sol})}^{\text{M06-2X}} \quad (4)$$

where PZ symbolizes the parent phenazine, XPZ represents the substituted phenazine molecules, $E_{1(\text{ref})}^0$ is the reported redox potential of parent PZ,²⁷ $\Delta G_{(\text{rxn},\text{sol})}$ corresponds to the free energy change of the reaction, F is the Faraday constant, n is number of electrons involved in the reduction, and $G_{(\text{sol})}^0$ represents the final composite free energy of individual species, which was calculated by adding the free energy contribution computed at the B3LYP level of theory, $G_{(\text{them},\text{gas})}^{(\text{B3LYP})}$, to the single-point energies calculated at the M06-2X level of theory: $E_{(\text{sol})}^{\text{M06-2X}}$.

2.2. Data Generation. 2.2.1. Training-Set and Internal Test–Test. These data sets were obtained from work reported by Mavrandonakis and co-workers.²⁷ In their report, the redox potentials of 189 phenazine derivatives were computed using DFT in the DME solvent. These DFT redox potentials were used as a target property in this work during training and testing. 20 unique electron-withdrawing and electron-donating functional groups were present in the data set [–N(CH₃)₂, –NH₂, –OH, –OCH₃, –P(CH₃)₂, –SCH₃, –SH, –CH₃, –C₆H₅, –CH=CH₂, –F, –Cl, –CHO, –COCH₃, –CONH₂,

Table 3. Representative Structures from Multiple Functional Group Test-Sets^a

 Mol ID: 19	 Mol ID: 9	 Mol ID: 7
 Mol ID: 12	 Mol ID: 14	 Mol ID: 5

^aMol IDs were assigned to identify derivatives from the corresponding data set.

–COOCH₃, –COOH, –CF₃, –CN, and –NO₂]. It should be noted that phenazine derivatives in this data set contain only one type of functional group per molecule. The optimized 3D structures of derivatives in neutral and in anionic states were also provided. However, only neutral structures were used in this study. Unfortunately, not all compounds were supplied with their neutral structure, those compounds were modeled, and their optimized structures were added to the data set. Next, 208 different types of features were generated using the RDKit python library.⁵³ The list of all features is given in Table S1 of Supporting Information. The features were scaled using the “StandardScaler” class of the scikit-learn library,⁶² removing the mean and scaling each feature to unit variance. Finally, the whole data set was shuffled and split randomly into a training-set and test-set in an 8:2 ratio (151 samples in the training-set and 38 samples in the test-set). A few phenazine derivatives from the training-set/internal test-set are shown in Table 1.

2.2.2. External Test-Set. This data set was compiled from different reports studying various properties of phenazine derivatives.^{63–67} Their redox potentials were computed using DFT and used as a target property during testing. We gathered a total of 30 phenazine derivatives. Derivatives containing five or more substituted rings were removed. Also, derivatives having drastically different neutral and anion structures were removed. In the end, 22 diverse phenazine derivatives with multiple types of functional groups remained in the external test-set. Table 2 shows some of the structures from this data set. It can be seen that this data set contains unique and different structures compared to the training-set.

2.2.3. Multiple Functional Group Test-Sets. This data set contains two test-sets: (i) two functional group test-set and (ii) three functional group test-set. These test-sets were generated by randomly choosing the position and the type of the functional group from this list [–N(CH₃)₂, –NH₂, –OH, –OCH₃, –P(CH₃)₂, –SCH₃, –SH, –CH₃, –C₆H₅, –CH=CH₂, –F, –Cl, –CHO, –COCH₃, –CONH₂, –COOCH₃, –COOH, –CF₃, –CN, and –NO₂]. 20 derivatives having two different types of functional groups per molecule were generated for two functional group test-set. Similarly, 20 derivatives having three different types of functional groups per molecule were generated for three functional group test-set. Their redox potentials were computed using DFT and used as a target property during testing. Five derivatives from two and three functional group test-sets were removed to form an external validation set. Thus, the final size of two and three

functional group test-sets was reduced from 20 to 15. In this report, the term “multiple” refers to the derivatives containing different types and more than one functional group. Similarly, the terms “two functional groups” and “three functional groups” refer to the derivatives containing two different types of functional groups and three different types of functional groups per molecule, respectively. A few representative structures from these test-sets are shown in Table 3.

2.2.4. External Validation Set. An external validation set of 10 phenazine derivatives was compiled from two and three functional group test-sets. Five derivatives from two functional group test-set and five derivatives from three functional group test-set were selected. Their redox potentials were computed using DFT and used as a target property. This validation set does not come from the training-set. Therefore, it is termed as an external validation set. It was used for feature selection and hyperparameter optimization. External validation set improves generalization by transferring knowledge from the test-set to models through hyperparameters.

2.3. Hyperparameter Optimization. Hyperparameters of the models were optimized using the “GridSearchCV” class of the scikit-learn library.⁶² During hyperparameter optimization, models were trained on the training-set and evaluated on the external validation set. Mean squared error (MSE) was used as an evaluation metric for hyperparameter optimization. The grid of hyperparameters for each model is given in Table S2 of Supporting Information. The parameter grid was adjusted manually.

2.4. ML Models. Following four ML models were investigated in this study. These models were chosen due to their ability to generalize from small data sets. Models were implemented with the scikit-learn python library.⁶² First, models were trained on the training-set containing all 208 features, followed by hyperparameter optimization. Then, the models were re-trained on different subsets of features to identify the set of features having the highest average performance on the external validation set. Once the optimum features were identified, hyperparameter optimization was performed with the selected features to improve the model performance further.

2.4.1. Automatic Relevance Determination Regression (ARDR). This is a probabilistic model related to the sparse Bayesian learning (SBL) framework. It assumes axis-parallel, elliptical Gaussian distribution for each coefficient. The precision of each Gaussian distribution is drawn from the

prior distribution (gamma distribution); therefore, it can lead to sparser coefficients. Thus, it is an effective tool to remove irrelevant features.^{68,69}

2.4.2. Gaussian Process Regression (GP). It is a non-parametric Bayesian model. The nonparametric Bayesian model provides the probability distribution of parameters over all possible functions that fit the data. The prior in a Gaussian process is specified on the function space. Gaussian process prior is a multivariate normal distribution whose mean is obtained from the data, and covariance is specified using the kernel function. The hyperparameters of the kernel are optimized during the training.^{70,71} We used a combination of *WhiteKernel* and *RBF* kernel. *WhiteKernel* is used for specifying the noise level, and *RBF* kernel is a very popular kernel used in many algorithms.

2.4.3. Kernel Ridge Regression (KRR). It is the extension of ridge regression with kernel trick. In ridge regression, a linear model is leaned with the L2-norm regularization. Using the kernel trick, KRR learns a linear function in the high dimensional non-linear space without actually transforming the data.⁷²

2.4.4. Support Vector Regression (SVR). This model is the regression form of the support vector machine (SVM), a popular algorithm for classification tasks. Analogous to the SVM, SVR depends on the subset of training data and ignores the points whose prediction is close to their true value. SVM also utilizes the kernel trick and learns a hyperplane in the high dimensional space.⁷³

2.5. Evaluation Metrics. The following metrics were used for evaluating the model performance. In the formulas below, N denotes the number of data points, \hat{y}_i denotes the predicted value of i th sample, and the y_i denotes the corresponding true value.

- Coefficient of determination (R^2):

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2}$$

where

$$\bar{y} = \frac{\sum_{i=1}^N y_i}{N}$$

- Mean Squared Error (MSE):

$$MSE = \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{N}$$

- Mean Absolute Error (MAE):

$$MAE = \frac{\sum_{i=1}^N |y_i - \hat{y}_i|}{N}$$

The use of terms “Accuracy” and “Performance” in this report is contextual and refers to one or more metrics defined above.

2.6. Feature Selection. As the number of features obtained from the RDKit library was more than the size of training-set, it was necessary to implement a feature selection strategy. It has been observed that the training-set containing more features than data points leads to overfitting.³⁰ Feature

selection was implemented using the “*SelectKBest*” class of the scikit-learn library.⁷⁴ The parameter “ k ” of the “*SelectKBest*” class was obtained by evaluating the average performance of models on the external validation set at different values of “ k .” First, models were trained on the training-set containing all features, followed by hyperparameter optimization. Then, the models were re-trained on the subsets of features selected using “*SelectKBest*” class at different values of “ k .” These values for “ k ” were tested: 50, 75, 100, 125, 150, and 208. The average model performance at different values of “ k ” on the external validation set is shown in Table 4. It can be seen that

Table 4. Average Model Performance on External Validation Set at Different Values of “ k ”

performance metric	values of “ k ”					
	50	75	100	125	150	208
R^2	0.45	0.42	0.57	0.55	0.54	0.54
MSE	0.02	0.02	0.02	0.02	0.02	0.02
MAE	0.12	0.12	0.10	0.10	0.10	0.10

the models trained on 100 selected features show the highest average performance in terms of R^2 . Therefore, these 100 features were selected for the subsequent analysis. The models trained on 100 selected features were further improved through hyperparameter optimization.

2.7. Feature Importance Analysis. Feature importance analysis was performed using the technique known as permutation importance. In this technique, values of the feature to be assessed are randomly shuffled (permuted). Then, prediction accuracy is computed on the shuffled data set. Shuffling feature values is equivalent to replacing the feature with noise, thereby removing its information from the data set. Therefore, the model is expected to perform poorly on the shuffled data set if the feature is important. The degree of importance depends on the amount of variation in the accuracy. This technique does not re-train the model; therefore, a trained model is required. The permutation importance was computed using “*permutation_importance*” class of the scikit-learn library and the training-set.⁷⁵ This procedure was repeated 100 times to obtain reliable estimates. The feature importance scores were rescaled between 0 to 1. The mean and standard deviation of the feature scores were reported. The mean feature score was used for the ranking of individual features. The terms “Feature” and “Descriptor” are used interchangeably in this report.

3. RESULTS AND DISCUSSION

3.1. Test-Set Performance. We assessed the generalizability of the trained models (i.e., performance on the unseen data) using internal and external test-sets. Please refer to Section 2 for the preparation of internal and external test-sets. As the internal test-set comes from the same source, it is very similar to the training-set and contains derivatives with only one type of functional group per molecule. However, the external test-set is compiled from multiple sources, therefore, it has very diverse phenazine derivatives with different types of functional groups. It also contains functional groups and structures not present in the training-set (e.g., $-NHPH$, $-Br$, and extended conjugation). Figure 2 shows the performance on the internal test-set, and Figure 3 shows performance on the external test-set. It can be seen that all models have excellent accuracy on the internal test-set ($R^2 > 0.98$) and high accuracy

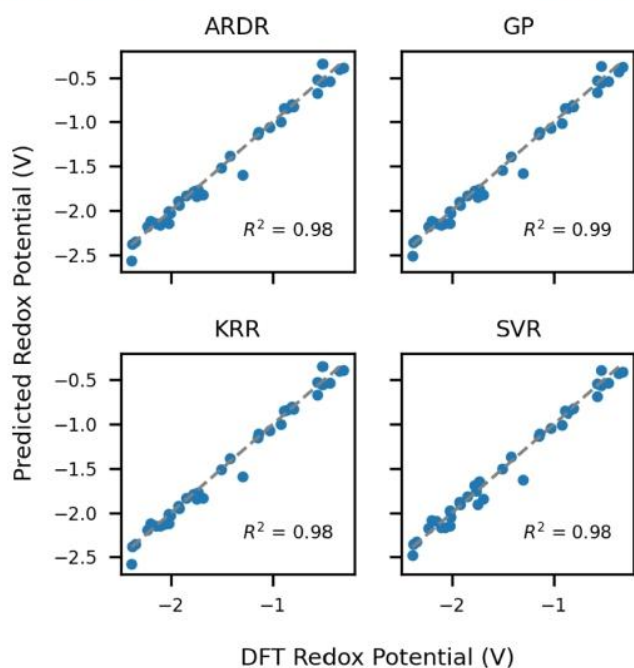


Figure 2. Plots showing ML predictions on internal test-set (*y*-axis) vs DFT redox potentials (*x*-axis). Gray dashed line corresponds to the perfect predictions.

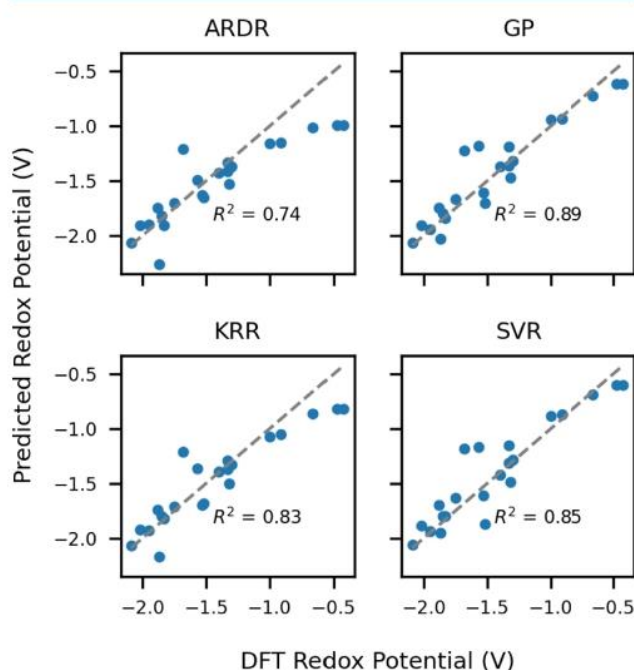


Figure 3. Plots showing ML predictions on external test-set (*y*-axis) vs DFT redox potentials (*x*-axis). Gray dashed line corresponds to the perfect predictions.

on the external test-set set ($R^2 > 0.74$). The GP model achieved the highest R^2 of 0.89 on the external test-set. After deep analysis in Section 3.3, it was revealed that GP is not a stable model, whereas relatively low-performing models KRR ($R^2 = 0.83$) and SVR ($R^2 = 0.85$) are more stable. Therefore, one should be careful while using the high-performing model, and the stability of the model should also be considered. The values of performance metrics on internal and external tests are

shown in Table 5. Such a performance on the external test-set is surprising as models were trained on the phenazine

Table 5. Values of Performance Metrics on Internal and External Test-Sets^a

Model name	Internal test-set			External test-set		
	R^2	MSE	MAE	R^2	MSE	MAE
ARDR	0.98	0.01	0.06	0.74	0.06	0.18
GP	0.99	0.01	0.05	0.89	0.03	0.11
KRR	0.98	0.01	0.05	0.83	0.04	0.14
SVR	0.98	0.01	0.07	0.85	0.03	0.13

^aNumbers were rounded upto two decimals.

derivatives having only one type of functional group. These results show that ML models are capable of generalizing from a very small and simple data set.

3.2. Prediction on Multiple Functional Group Test-Sets. Next, we assessed the model performance on the phenazine derivatives substituted with different types of functional groups per molecule. These test-sets were generated randomly; please refer to Section 2 for the generation of this data set. Figures 4 and 5 show the performance on the

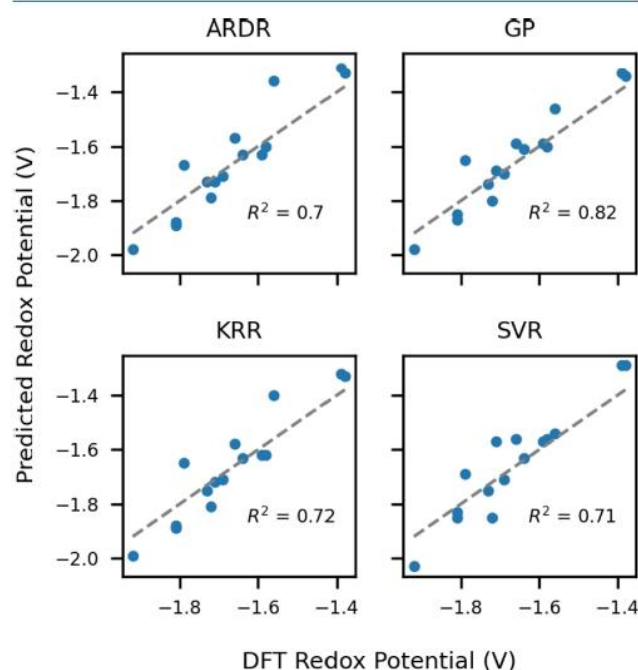


Figure 4. Plots showing ML predictions on two functional group test-set (*y*-axis) vs DFT redox potentials (*x*-axis). Gray dashed line corresponds to the perfect predictions.

derivatives containing two and three different functional groups, respectively. It can be seen that the models performed reasonably well ($R^2 > 0.7$) even though molecules used for the training had only one type of functional group per molecule. In particular, GP model achieved the highest performance of $R^2 = 0.82$ on two functional groups test-set. However, automatic relevance determination regression (ARDR) achieved the highest performance of $R^2 = 0.82$ on three functional groups test-set. A deeper analysis of GP and ARDR in Section 3.3 suggests that GP and ARDR are not very reliable models. Although KRR and SVR have relatively low performance, they

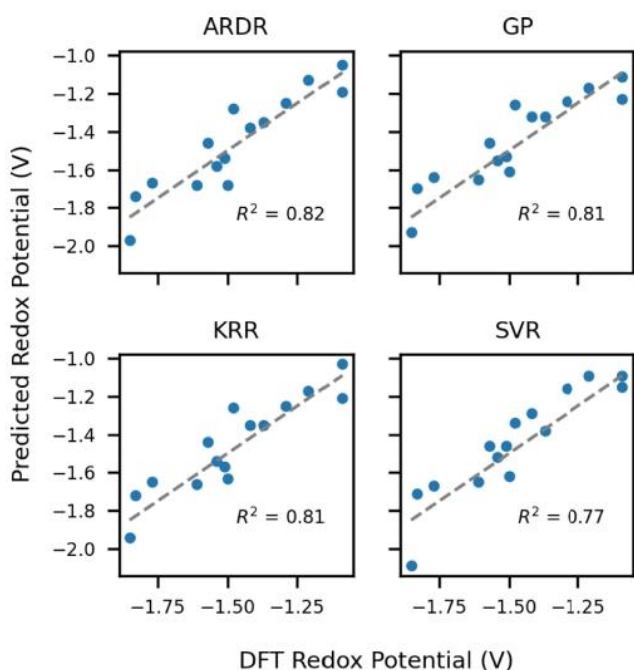


Figure 5. Plots showing ML predictions on three functional group test-set (*y*-axis) vs DFT redox potentials (*x*-axis). Gray dashed line corresponds to the perfect predictions.

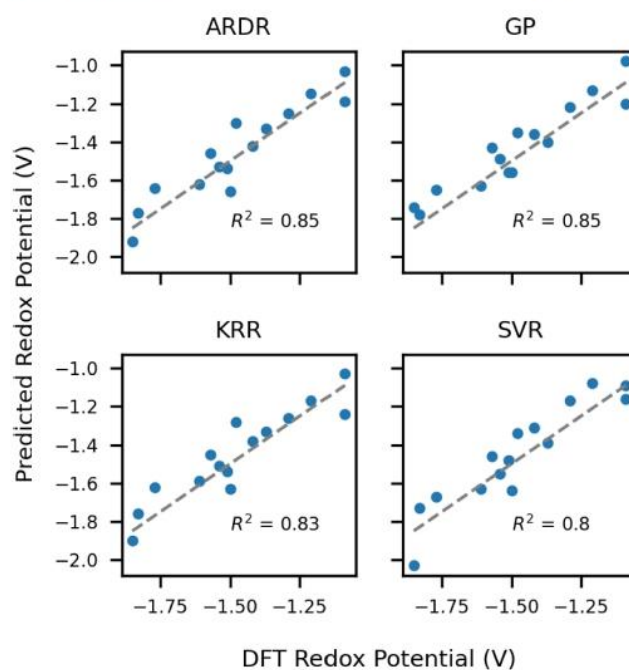


Figure 6. Plots showing ML predictions on three functional group test-set (*y*-axis) vs DFT redox potentials (*x*-axis). The combined data set (training-set + two functional group test-set) was used for the training. Gray dashed line corresponds to the perfect predictions.

are more reliable. Therefore, one should be careful while using a high-performing model, and the model's reliability and stability should also be considered. Nevertheless, these results again show the surprising generalization power of ML models.

Furthermore, we added all 15 derivatives from two functional group test-set to the training-set and re-trained the models on this new data set of 166 derivatives. The predictive performance of this combined data set was assessed on the same data set of 15 derivatives containing three different types of functional groups. The results of this analysis are shown in Figure 6. It can be seen that the model performance has improved with the addition of more data in the training-set.

3.3. Feature Importance Analysis. We carried out feature importance analysis using permutation importance. Please refer to Section 2 for the details on the technique. In order to understand how model performance changes with the number of descriptors, we re-trained the models on the subset of features and assessed their performance on the internal test-set. Top 50 features based on their permutation importance score were used. R^2 was used as a performance metric. The result of this analysis is shown in Figure 7. It can be seen that most of the models show a jump in the R^2 and have $R^2 > 0.9$ around the top 10 features. The unusual behavior of the GP model is attributed to the instability of the model for a small number of features. The plots in Figure 8 show the histograms of the top 10 important features from each model. Although models show variation in feature importance, they all agree in terms of the most important feature that is, "PEOE_VSA1." Interestingly, most of the features in ARDR have small weights as ARDR tries to prune the large number of irrelevant features, leading to a sparse model.^{69,76} Five out of 10 features—"MaxAbsPartialCharge," "PEOE_VSA1," "fr_ArN," "fr_NH0," and "fr_NH2" are common to all models. Variations in the feature importance scores could be attributed to the difference

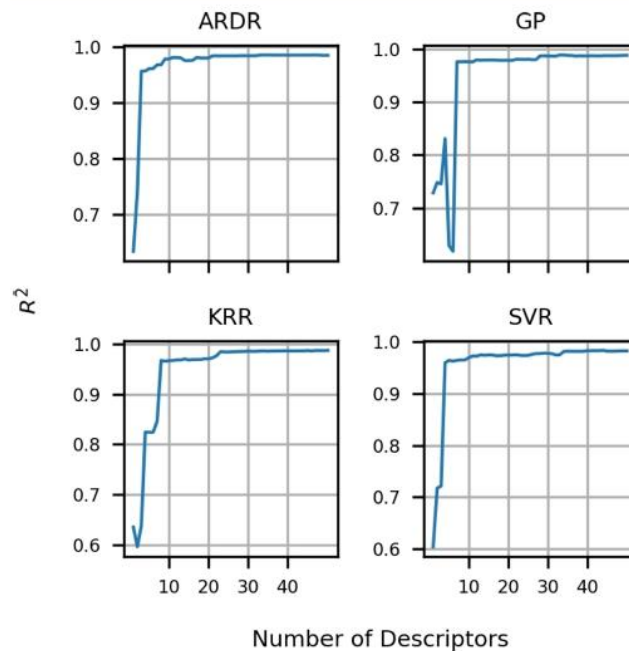


Figure 7. R^2 vs number of descriptors. R^2 was computed using the internal test-set. In this study, we identified a few issues with ARDR and GP. Despite high predictive performance, ARDR is not a reliable model as it places very high weight on one feature (i.e., "PEOE_VSA1"). Similarly, GP is not a reliable model as it becomes unstable when the small number of features are used. We encountered divided by zero errors in the kernel function during the analysis with the GP model.

in the internal structures of the models. Here, we discuss some of the common features from Figure 8.

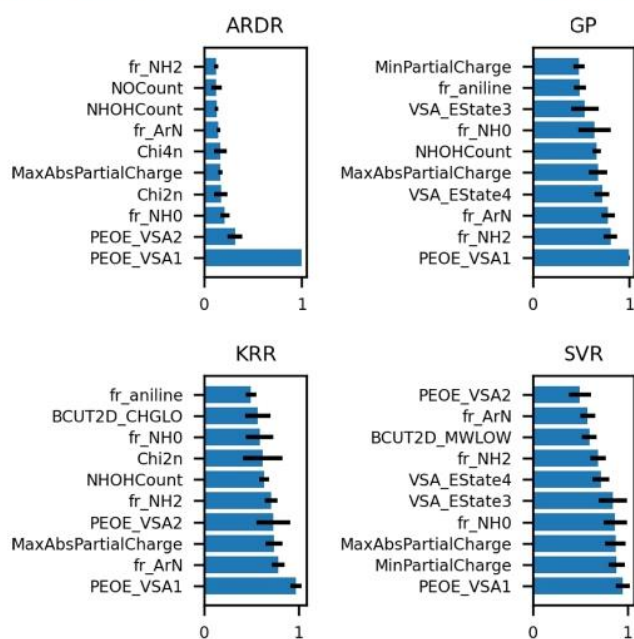


Figure 8. Top 10 features (y-axis) vs mean feature importance score (x-axis). Feature importance scores were rescaled between 0 to 1. Error bars represent standard deviation from 100 repetitions.

3.3.1. *PEOE_VSA1*. This is the sum of the approximate accessible van der Waals surface area (i.e., VSA in Å²) of the atoms having partial charge less than -0.30 .^{77–79} The partial charges are computed using the partial equalization of orbital electronegativities (PEOE) method developed by Gasteiger and Marsili in 1980. Please refer to the discussion of *MaxAbsPartialCharge* descriptor for the PEOE method. Thus, this descriptor captures the information related to molecular size and the number of electron-donating functional groups.

3.3.2. *MaxAbsPartialCharge*. This is the maximum value of the absolute Gasteiger partial charges present in the molecule. In 1980, Gasteiger and Marsili gave the procedure to calculate the partial charges in a molecule. That procedure is known as PEOE. In this method, the charge is transferred between

bonded atoms until equilibrium. Gasteiger partial charges depend on the connectivity and orbital electronegativity, thus capturing the electron-donating and electron-withdrawing power of the atoms.⁸⁰ Electronegativity is essential information as electron-donating groups decrease the redox potential, and electron-withdrawing groups increase the redox potential.²⁷

3.3.3. *MinPartialCharge*. This is the minimum value of the Gasteiger partial charges present in the molecule. Please refer to the discussion of *MaxAbsPartialCharge* descriptor for the properties of Gasteiger partial charges.

3.3.4. *fr_NH0*. It is the number of tertiary amines present in the molecule.

3.3.5. *fr_ArN*. It is the number of N functional groups attached to aromatic rings.

3.3.6. *fr_NH2*. It is the number of primary amines.

3.3.7. *NHOHCount*. It is the number of N–H and O–H bonds present in the molecule.

From the analysis in this section, we realized that there are some issues with the ARDR and GP which are outlined below. One should be very careful while using ARDR and GP models.

3.3.8. Issues with the ARDR Model. As ARDR is related to the SBL framework, it reduces the number of irrelevant features. Unfortunately, in this case, ARDR has put a lot of weight on only one feature, that is, “*PEOE_VSA1*” (Figure 8). Surprisingly, ARDR also archives an accuracy of more than $0.95R^2$ only with the two features (Figure 7). Although it has shown good performance on the data set used in this work, it may not work for the broad chemical space. This type of behavior reduces the reliability of the model.

3.3.9. Issues with the GP Model. From Figure 7, it can be seen that the model’s accuracy decreases with more features, and at around 10 features, there is a significant drop in the performance. We also encountered divided by zero errors in the kernel function during this analysis. This shows that GP may not be a very stable model in this case.

3.4. Structure–Functional Relationship. “*PEOE_VSA1*” is the most important descriptor common to all models. It is computed by summing over the approximate accessible VSA (i.e., in Å²) of the atoms having partial charge less than -0.30 .^{77–79} Thus, the “*PEOE_VSA1*” descriptor captures the information related to molecular size and the number of

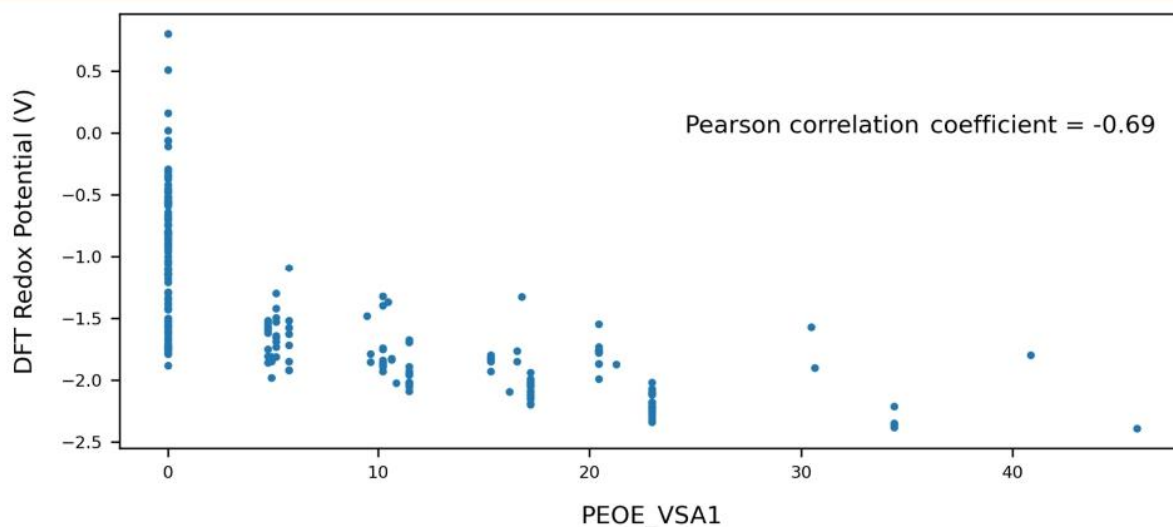


Figure 9. Redox potential vs “*PEOE_VSA1*.”

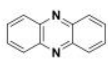
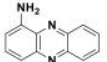
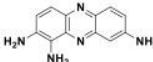
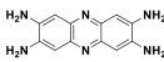
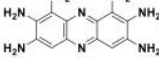
				
Mol ID: 0	Mol ID: 3	Mol ID: 105	Mol ID: 134	Mol ID: 136
PEOE_VSA1: 0	PEOE_VSA1: 5.73	PEOE_VSA1: 17.20	PEOE_VSA1: 22.93	PEOE_VSA1: 34.40
Potential: -1.74	Potential: -1.85	Potential: -2.09	Potential: -2.32	Potential: -2.36
PEOE_VSA1 increases		→		
Delocalization increases		→		
Redox Potential decreases		→		

Figure 10. Examples from the training-set showing the effect of charge delocalization on “PEOE_VSA1.” Values of “PEOE_VSA1” and DFT redox potentials in volts are also shown. Mol IDs were assigned to identify derivatives from the corresponding data set.

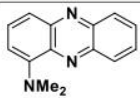
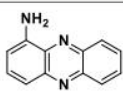
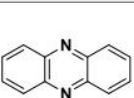
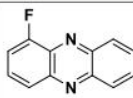
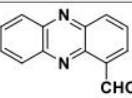
				
Mol ID: 1	Mol ID: 3	Mol ID: 0	Mol ID: 21	Mol ID: 26
Potential: -1.85	Potential: -1.85	Potential: -1.74	Potential: -1.63	Potential: -1.50
Shift: -0.11	Shift: -0.11	Shift: 0	Shift: 0.11	Shift: 0.24
Electron-donating groups		Electron-withdrawing groups		

Figure 11. Examples showing positive and negative shifts with respect to parent phenazine. DFT redox potentials and shifts in volts are also shown. Mol IDs were assigned to identify derivatives from the corresponding data set.

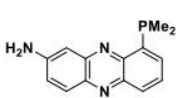
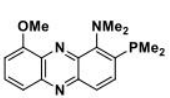
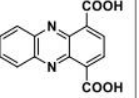
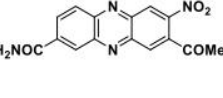
			
Mol ID: 9	Mol ID: 8	Mol ID: 7	Mol ID: 15
Potential: -1.92	Potential: -1.85	Potential: -1.40	Potential: -1.09
Shift: -0.18	Shift: -0.11	Shift: 0.34	Shift: 0.65
Electron-donating groups		Electron-withdrawing groups	

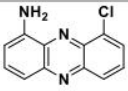
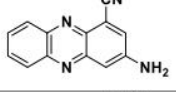
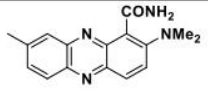
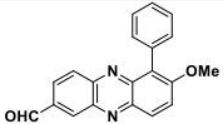
Figure 12. Examples showing the effect of similar type of functional groups on the redox potential. DFT redox potentials and shifts in volts are also shown. Mol IDs were assigned to identify derivatives from the corresponding data set.

electron-donating functional groups present in the molecule. From Figure 9, we can see that the redox potential of phenazine derivatives decreases with the increasing value of “PEOE_VSA1.” The Pearson correlation coefficient between “PEOE_VSA1” and redox potential is -0.69 , supporting the observation. We observed that the value of “PEOE_VSA1” is higher for the systems having delocalization of negative partial charge. The delocalized system contains more atoms with the negative partial charge than the corresponding localized system. Thus, the number of atoms contributing to “PEOE_VSA1” in delocalized systems is higher than localized ones. The effect of delocalization of partial charge on “PEOE_VSA1” is shown in Figure 10 with a few examples from the training-set. Thus, for designing better anolytes, it is suggested to increase the delocalization of negative partial charge in the phenazine derivatives.

The redox potential of phenazine derivative depends on the type of functional group, the position of attachment, and the number of functional groups. Two types of functional groups have been investigated in this study: (i) electron-donating, and

(ii) electron-withdrawing. The redox potential of parent phenazine without any functional group is -1.74 V. When the redox potential of the derivative decreases (i.e., less than -1.74 V) after the attachment of functional groups, then it is called a negative shift. Similarly, if it increases, it is called a positive shift. The shift is quantified as the difference between the redox potential of phenazine derivative and parent phenazine. After sorting phenazine derivatives based on the redox potential, it was observed that electron-donating groups show a negative shift, whereas electron-withdrawing groups show a positive shift. Thus, the shift corresponding to electron-donating groups is negative and electron-withdrawing groups is positive. The redox potentials of phenazine derivatives were computed using the approach discussed in Section 2.1. Equation 2 shows that functional groups stabilizing the anionic form of phenazine derivatives have high redox potential. In contrast, those that destabilize the anionic form have low redox potential. Therefore, electron-withdrawing groups show a positive shift as they stabilize the anionic form, and electron-donating groups show a negative shift as they destabilize the

Table 6. Examples Showing the Effect of Absolute Values of Single Functional Group Shift on the Redox Potential of Derivatives Containing Different Types of Functional Groups^a

	Phenazine Derivative	Details of the phenazine derivative	Redox potential of the corresponding single functional group derivative	Shift of the corresponding single functional group derivative
A.		Mol ID: 1 Potential: -1.71 Shift: 0.03	-NH ₂ : -1.85 -Cl: -1.61	-NH ₂ : -0.11 -Cl: 0.13
B.		Mol ID: 7 Potential: -1.58 Shift: 0.16	-NH ₂ : -1.92 -CN: -1.42	-NH ₂ : -0.18 -CN: 0.32
C.		Mol ID: 12 Potential: -1.83 Shift: -0.09	-N(CH ₃) ₂ : -1.98 -CH ₃ : -1.79 -CONH ₂ : -1.52	-N(CH ₃) ₂ : -0.24 -CH ₃ : -0.05 -CONH ₂ : 0.22
D.		Mol ID: 4 Potential: -1.54 Shift: 0.20	-OCH ₃ : -1.86 -C ₆ H ₅ : -1.76 -CHO: -1.51	-OCH ₃ : -0.12 -C ₆ H ₅ : -0.02 -CHO: 0.23

^aDFT redox potentials and shifts in volts are also shown. Mol IDs were assigned to identify derivatives from the corresponding data set.

anionic form. A few examples showing positive and negative shifts with respect to parent phenazine are shown in Figure 11.

In the case of derivatives with multiple functional groups, if all groups are similar, then shift also corresponds to their type. For example, when the derivative contains all electron-donating groups, it shows a negative shift. Similarly, the shift is positive when the derivative contains all electron-withdrawing groups. A few examples having similar types of functional groups are shown in Figure 12.

When derivatives contain more than one functional group that differ in their type, the shift is determined by the group showing the highest absolute shift in the corresponding single functional group derivative. For example, derivative A in Table 6 contains -NH₂, an electron-donating group which has a shift of -0.11 V and -Cl, an electron-withdrawing group which has a shift of 0.13 V. The absolute of the shift for -Cl is more than -NH₂; therefore, derivative A shows a positive shift of 0.03 V, supporting our claim. A similar analysis is applicable to derivative B, which also shows a positive shift. Derivative C contains -N(CH₃)₂ and -CH₃, two electron-donating groups, and -CO(NH₂), an electron-withdrawing group. An absolute shift of -N(CH₃)₂ is -0.24 V, which is the highest among all three groups. Therefore, derivative C shows a negative shift of -0.09 V. Derivative D contains -OCH₃ and -C₆H₅, two electron-donating groups, and -CHO, one electron-withdrawing group. However, derivative D shows a positive shift as the absolute shift of -CHO is more than both electron-donating groups. Thus, the redox potential of phenazine derivatives containing multiple functional groups is determined by the relative strength of electron-donating or electron-withdrawing power of the functional groups.

The effect of position on the redox potential of single functional group derivatives has been studied by Mavrandonakis and co-workers.²⁷ They showed that the position does not have a significant effect for electron-withdrawing groups. However, electron-donating groups which are capable of intramolecular hydrogen bonding show more negative shift when

attached at position 2 compared to position 1. The position numbers in phenazine derivatives are shown in Figure 13. They

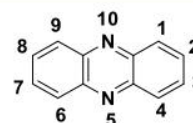
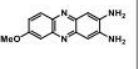
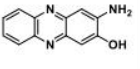
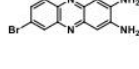
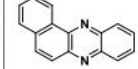
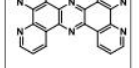
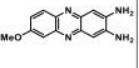
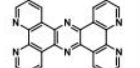
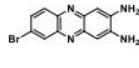
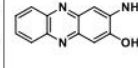
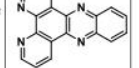
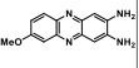
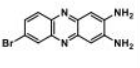
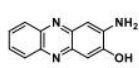
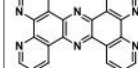
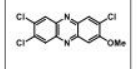


Figure 13. Numbering of the positions in phenazine derivatives.

also investigated the effect of the number of functional groups on redox potential. It was shown that the addition of more electron-withdrawing groups shifts the redox potential continuously toward positive values. However, this effect is less significant for electron-donating groups. The difference between the phenazine derivative with four amino groups and eight amino groups is very small (~0.05 V). The difference between phenazine derivative with four cyano groups and eight cyano groups is ~1.23 V.

3.5. Identification of Promising Phenazine Derivatives for the Analyte. In this section, we identify the top five promising candidates for the analyte using the trained ML models. Models developed in this study are based on features that do not require electronic structure calculations. Therefore, these models could screen millions of molecules in a significantly small amount of time. Then, experimentation or DFT calculations could be performed on the reduced number of molecules to identify the best redox-active molecules, saving computational and experimental costs. Using this hybrid DFT-ML approach, we have identified promising phenazine derivatives for the analyte in RFBs. These promising candidates would provide a good starting point for the experimentalists. Electron-donating molecules with negative redox potential are preferred candidates for the analyte. As KRR and SVR are stable models, the predictions here are based on them. The values of redox potentials are averaged over 100 independent iterations of data splitting and model training. Table 7 lists the top five phenazine derivatives from

Table 7. Top Five Anolyte Candidates Predicted Using DFT, KRR, and SVR from the External Test-Set^a

DFT	 Mol ID: 13 DFT: -2.09	 Mol ID: 29 DFT: -2.02	 Mol ID: 12 DFT: -1.95	 Mol ID: 1 DFT: -1.88	 Mol ID: 5 DFT: -1.87
KRR	 Mol ID: 13 ML: -2.09 DFT: -2.09	 Mol ID: 5 ML: -2.09 DFT: -1.87	 Mol ID: 12 ML: -1.98 DFT: -1.95	 Mol ID: 29 ML: -1.95 DFT: -2.02	 Mol ID: 4 ML: -1.78 DFT: -1.83
SVR	 Mol ID: 13 ML: -2.06 DFT: -2.09	 Mol ID: 12 ML: -1.96 DFT: -1.95	 Mol ID: 29 ML: -1.91 DFT: -2.02	 Mol ID: 5 ML: -1.89 DFT: -1.87	 Mol ID: 3 ML: -1.81 DFT: -1.52

^aSVR and KRR were trained on the phenazine derivatives containing single type of functional group per derivative. Mol IDs and redox potentials predicted from DFT and ML models are shown below the respective candidates. Mol IDs were assigned to identify derivatives from the corresponding test-set. Derivatives are arranged in increasing order of redox potential. Redox potentials are given in the unit of volts.

the external test-set with the most negative redox potentials obtained from DFT and two ML models. Four out of five predictions from KRR and SVR match with DFT predictions. The top five promising candidates from multiple functional groups test-sets are shown in Tables S3–S5 Supporting Information.

4. CONCLUSIONS

In this study, four ML models were employed to predict the redox potentials of phenazine derivatives in DME using DFT. Models were trained on a small data set of 151 phenazine derivatives having only one type of functional group per molecule (20 unique functional groups). The trained models achieved high accuracies ($R^2 > 0.74$) on internal and external test-sets containing diverse phenazine derivatives. We also showed that despite being trained on derivatives with a single type of functional groups, models were able to predict the redox potentials of the derivatives containing multiple and different types of functional groups with good accuracies ($R^2 > 0.7$). Feature selection and hyperparameter optimization using the validation set were critical strategies for performance improvement. Feature selection removed the unnecessary and noisy features. Hyperparameter optimization using an external validation set helped improve the generalizability of the models. The addition of 15 derivatives from two functional group test-sets in the training-set improved the accuracy on three functional group test-sets. It was observed that the “PEOE_VSA1” descriptor was the most important molecular feature as it contains information related to molecular size and the partial charges. Deeper analysis showed that one should not rely only on the model performance but also investigate the stability and reliability of the models. Through the structure–functional relationship, we observed that the redox potential of derivatives containing multiple functional groups is

influenced by the functional group having either strong electron-donating or strong electron-withdrawing power. Models developed in this study are based on features that do not require electronic structure calculations. Therefore, these models could screen millions of molecules in a significantly small amount of time. Then, experimentation or DFT calculations could be performed on the screened candidates to identify the best molecules, saving computational and experimental costs. Using this hybrid DFT-ML approach, we have identified promising phenazine derivatives for the anolyte in RFBs. These promising candidates would provide a good starting point for the experimentalists. This study shows that it is possible to develop reasonably accurate ML models for complex quantities such as redox potential using small and simple data sets.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at <https://pubs.acs.org/doi/10.1021/acsomega.1c06856>.

Additional Computation details; list of all features; and parameter grids used during hyperparameter optimization; top five candidates for the anolyte from multiple functional group test-sets (PDF)

■ AUTHOR INFORMATION

Corresponding Authors

Siddharth Ghule – Physical and Materials Chemistry Division, CSIR-National Chemical Laboratory (CSIR-NCL), Pune 411008, India; Academy of Scientific and Innovative Research (AcSIR), Ghaziabad 201002, India; orcid.org/0000-0003-0864-0777; Phone: +91-20-25903095; Email: ss.ghule@ncl.res.in

Kavita Joshi – Physical and Materials Chemistry Division, CSIR-National Chemical Laboratory (CSIR-NCL), Pune 411008, India; Academy of Scientific and Innovative Research (AcSIR), Ghaziabad 201002, India; Phone: +91-20-25902476; Email: k.joshi@ncl.res.in

Kumar Vanka – Physical and Materials Chemistry Division, CSIR-National Chemical Laboratory (CSIR-NCL), Pune 411008, India; Academy of Scientific and Innovative Research (AcSIR), Ghaziabad 201002, India; orcid.org/0000-0001-7301-7573; Phone: +91-20-25903095; Email: k.vanka@ncl.res.in

Authors

Soumya Ranjan Dash – Physical and Materials Chemistry Division, CSIR-National Chemical Laboratory (CSIR-NCL), Pune 411008, India; Academy of Scientific and Innovative Research (AcSIR), Ghaziabad 201002, India; orcid.org/0000-0001-7267-6104

Sayan Bagchi – Physical and Materials Chemistry Division, CSIR-National Chemical Laboratory (CSIR-NCL), Pune 411008, India; Academy of Scientific and Innovative Research (AcSIR), Ghaziabad 201002, India; orcid.org/0000-0001-6932-3113

Complete contact information is available at: <https://pubs.acs.org/10.1021/acsomega.1c06856>

Author Contributions

S.G. and S.D. carried out the research work and written the manuscript with the guidance and supervision of K.V., K.J., and S.B.

Notes

The authors declare no competing financial interest. The MOL files of all phenazine derivatives and compiled DFT data are available at https://github.com/siddharth-ncl-work/ml_redox_potential-DATA.git.

ACKNOWLEDGMENTS

K.V. is grateful to the Department of Science and Technology (DST) (EMR/2014/000013) for providing financial assistance. S.B. thanks CSIR-NCL and SERB, India (SR/S2/RJN-142/2012 and EMR/2016/000576), for financial support. S.G. acknowledges Council of Scientific and Industrial Research (CSIR) for providing Research Fellowship. S.D. acknowledges CSIR-NCL (MLP101026) for providing a fellowship. The support and the resources provided by the “PARAM Brahma Facility” under the National Supercomputing Mission, Government of India at the Indian Institute of Science Education and Research (IISER) Pune are gratefully acknowledged.

REFERENCES

- (1) Shafiee, S.; Topal, E. When Will Fossil Fuel Reserves Be Diminished? *Energy Policy* **2009**, *37*, 181–189.
- (2) Dehghani-Sanj, A. R.; Tharumalingam, E.; Dusseault, M. B.; Fraser, R. Study of Energy Storage Systems and Environmental Challenges of Batteries. *Renewable Sustainable Energy Rev.* **2019**, *104*, 192–208.
- (3) Höök, M.; Tang, X. Depletion of Fossil Fuels and Anthropogenic Climate Change—A Review. *Energy Policy* **2013**, *52*, 797–809.
- (4) Gür, T. M. Review of Electrical Energy Storage Technologies, Materials and Systems: Challenges and Prospects for Large-Scale Grid Storage. *Energy Environ. Sci.* **2018**, *11*, 2696–2767.
- (5) Chu, W.-S.; Chun, D.-M.; Ahn, S.-H. Research Advancement of Green Technologies. *Int. J. Precis. Eng. Manuf.* **2014**, *15*, 973–977.

(6) Balat, H. Green Power for a Sustainable Future. *Energy Explor. Exploit.* **2007**, *25*, 1–25.

(7) Demirbas, A. Electrical Power Production Facilities from Green Energy Sources. *Energy Sources, Part B* **2006**, *1*, 291–301.

(8) Dunn, B.; Kamath, H.; Tarascon, J.-M. Electrical Energy Storage for the Grid: A Battery of Choices. *Science* **2011**, *334*, 928–935.

(9) Chung, E. What Caused the Deadly Power Outages in Texas and How Canada's Grid Compares. CBC News. 2021. <https://www.cbc.ca/news/technology/power-outages-texas-canada-1.5920833> (accessed March 30, 2021).

(10) Larcher, D.; Tarascon, J.-M. Towards Greener and More Sustainable Batteries for Electrical Energy Storage. *Nat. Chem.* **2015**, *7*, 19–29.

(11) Koochi-Fayegh, S.; Rosen, M. A. A Review of Energy Storage Types, Applications and Recent Developments. *J. Energy Storage* **2020**, *27*, 101047.

(12) Deng, D. Li-ion Batteries: Basics, Progress, and Challenges. *Energy Sci. Eng.* **2015**, *3*, 385–418.

(13) Skyllas-Kazacos, M.; Chakrabarti, M. H.; Hajimolana, S. A.; Mjalli, F. S.; Saleem, M. Progress in Flow Battery Research and Development. *J. Electrochem. Soc.* **2011**, *158*, R55.

(14) Leung, P.; Li, X.; Ponce de León, C.; Berlouis, L.; Low, C. T. J.; Walsh, F. C. Progress in Redox Flow Batteries, Remaining Challenges and Their Applications in Energy Storage. *RSC Adv.* **2012**, *2*, 10125–10156.

(15) Sánchez-Diez, E.; Ventosa, E.; Guarnieri, M.; Trovò, A.; Flox, C.; Marcilla, R.; Soavi, F.; Mazur, P.; Aranzabe, E.; Ferret, R. Redox Flow Batteries: Status and Perspective towards Sustainable Stationary Energy Storage. *J. Power Sources* **2021**, *481*, 228804.

(16) Ha, S.; Gallagher, K. G. Estimating the System Price of Redox Flow Batteries for Grid Storage. *J. Power Sources* **2015**, *296*, 122–132.

(17) Whitehead, A. H.; Rabbow, T. J.; Trampert, M.; Pokorny, P. Critical Safety Features of the Vanadium Redox Flow Battery. *J. Power Sources* **2017**, *351*, 1–7.

(18) Chen, Y.; Kang, Y.; Zhao, Y.; Wang, L.; Liu, J.; Li, Y.; Liang, Z.; He, X.; Li, X.; Tavajohi, N.; Li, B. A Review of Lithium-Ion Battery Safety Concerns: The Issues, Strategies, and Testing Standards. *J. Energy Chem.* **2021**, *59*, 83–99.

(19) Díaz-Ramírez, M. C.; Ferreira, V. J.; García-Armingol, T.; López-Sabirón, A. M.; Ferreira, G. Environmental Assessment of Electrochemical Energy Storage Device Manufacturing to Identify Drivers for Attaining Goals of Sustainable Materials 4.0. *Sustain* **2020**, *12*, 342.

(20) da Silva Lima, L.; Quartier, M.; Buchmayr, A.; Sanjuan-Delmás, D.; Laget, H.; Corbisier, D.; Mertens, J.; Dewulf, J. Life Cycle Assessment of Lithium-Ion Batteries and Vanadium Redox Flow Batteries-Based Renewable Energy Storage Systems. *Sustain. Energy Technol. Assess.* **2021**, *46*, 101286.

(21) Kear, G.; Shah, A. A.; Walsh, F. C. Development of the All-Vanadium Redox Flow Battery for Energy Storage: A Review of Technological, Financial and Policy Aspects. *Int. J. Energy Res.* **2012**, *36*, 1105–1120.

(22) Alotto, P.; Guarnieri, M.; Moro, F. Redox Flow Batteries for the Storage of Renewable Energy: A Review. *Renewable Sustainable Energy Rev.* **2014**, *29*, 325–335.

(23) Weber, A. Z.; Mench, M. M.; Meyers, J. P.; Ross, P. N.; Gostick, J. T.; Liu, Q. Redox Flow Batteries: A Review. *J. Appl. Electrochem.* **2011**, *41*, 1137–1164.

(24) Qi, Z.; Koenig, G. M. Review Article: Flow Battery Systems with Solid Electroactive Materials. *J. Vac. Sci. Technol., B: Nanotechnol. Microelectron.: Mater., Process., Meas., Phenom.* **2017**, *35*, 040801.

(25) Sánchez-Diez, E.; Ventosa, E.; Guarnieri, M.; Trovò, A.; Flox, C.; Marcilla, R.; Soavi, F.; Mazur, P.; Aranzabe, E.; Ferret, R. Redox Flow Batteries: Status and Perspective towards Sustainable Stationary Energy Storage. *J. Power Sources* **2021**, *481*, 228804.

(26) Xi, J.; Xiao, S.; Yu, L.; Wu, L.; Liu, L.; Qiu, X. Broad Temperature Adaptability of Vanadium Redox Flow Battery—Part 2: Cell Research. *Electrochim. Acta* **2016**, *191*, 695–704.

- (27) De La Cruz, C.; Molina, A.; Patil, N.; Ventosa, E.; Marcilla, R.; Mavrandonakis, A. New Insights into Phenazine-Based Organic Redox Flow Batteries by Using High-Throughput DFT Modelling. *Sustainable Energy Fuels* **2020**, *4*, 5513–5521.
- (28) Gentil, S.; Reynard, D.; Girault, H. H. Aqueous Organic and Redox-Mediated Redox Flow Batteries: A Review. *Curr. Opin. Electrochem.* **2020**, *21*, 7–13.
- (29) Leung, P.; Shah, A. A.; Sanz, L.; Flox, C.; Morante, J. R.; Xu, Q.; Mohamed, M. R.; Ponce de León, C.; Walsh, F. C. Recent Developments in Organic Redox Flow Batteries: A Critical Review. *J. Power Sources* **2017**, *360*, 243–283.
- (30) Cao, J.; Tian, J.; Xu, J.; Wang, Y. Organic Flow Batteries: Recent Progress and Perspectives. *Energy Fuels* **2020**, *34*, 13384–13411.
- (31) Li, M.; Rhodes, Z.; Cabrera-Pardo, J. R.; Minter, S. D. Recent Advancements in Rational Design of Non-Aqueous Organic Redox Flow Batteries. *Sustainable Energy Fuels* **2020**, *4*, 4370–4389.
- (32) Romadina, E. I.; Komarov, D. S.; Stevenson, K. J.; Troshin, P. A.; Romadina, E. I.; Komarov, D. S.; Stevenson, K. J.; Troshin, P. A. New Phenazine Based Anolyte Material for High Voltage Organic Redox Flow Batteries. *Chem. Commun.* **2021**, *57*, 2986–2989.
- (33) Zhang, C.; Niu, Z.; Ding, Y.; Zhang, L.; Zhou, Y.; Guo, X.; Zhang, X.; Zhao, Y.; Yu, G. Highly Concentrated Phthalimide-Based Anolytes for Organic Redox Flow Batteries with Enhanced Reversibility. *Chem* **2018**, *4*, 2814–2825.
- (34) Butler, K. T.; Davies, D. W.; Cartwright, H.; Isayev, O.; Walsh, A. Machine Learning for Molecular and Materials Science. *Nature* **2018**, *559*, 547–555.
- (35) Schmidt, J.; Marques, M. R. G.; Botti, S.; Marques, M. A. L. Recent Advances and Applications of Machine Learning in Solid-State Materials Science. *npj Comput. Mater.* **2019**, *5*, 83.
- (36) Wei, J.; Chu, X.; Sun, X. Y.; Xu, K.; Deng, H. X.; Chen, J.; Wei, Z.; Lei, M. Machine Learning in Materials Science. *InfoMat* **2019**, *1*, 338–358.
- (37) Pilia, G.; Wang, C.; Jiang, X.; Rajasekaran, S.; Ramprasad, R. Accelerating Materials Property Predictions Using Machine Learning. *Sci. Rep.* **2013**, *3*, 2810.
- (38) Batra, R. Accurate Machine Learning in Materials Science Facilitated by Using Diverse Data Sources. *Nature* **2021**, *589*, 524–525.
- (39) Gómez-Bombarelli, R.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Duvenaud, D.; Maclaurin, D.; Blood-Forsythe, M. A.; Chae, H. S.; Eininger, M.; Ha, D. G.; Wu, T.; Markopoulos, G.; Jeon, S.; Kang, H.; Miyazaki, H.; Numata, M.; Kim, S.; Huang, W.; Hong, S. I.; Baldo, M.; Adams, R. P.; Aspuru-Guzik, A. Design of Efficient Molecular Organic Light-Emitting Diodes by a High-Throughput Virtual Screening and Experimental Approach. *Nat. Mater.* **2016**, *15*, 1120–1127.
- (40) Hautier, G.; Fischer, C. C.; Jain, A.; Mueller, T.; Ceder, G. Finding Natures Missing Ternary Oxide Compounds Using Machine Learning and Density Functional Theory. *Chem. Mater.* **2010**, *22*, 3762–3767.
- (41) Faber, F. A.; Lindmaa, A.; Von Lilienfeld, O. A.; Armiento, R. Machine Learning Energies of 2 Million Elpasolite (ABC2D6) Crystals. *Phys. Rev. Lett.* **2016**, *117*, 135502.
- (42) Carrasquilla, J.; Melko, R. G. Machine Learning Phases of Matter. *Nat. Phys.* **2017**, *13*, 431–434.
- (43) Cavasotto, C. N.; Di Filippo, J. I. Artificial Intelligence in the Early Stages of Drug Discovery. *Arch. Biochem. Biophys.* **2021**, *698*, 108730.
- (44) Peyton, B. G.; Briggs, C.; D’Cunha, R.; Margraf, J. T.; Crawford, T. D. Machine-Learning Coupled Cluster Properties through a Density Tensor Representation. *J. Phys. Chem. A* **2020**, *124*, 4861–4871.
- (45) Seko, A.; Hayashi, H.; Nakayama, K.; Takahashi, A.; Tanaka, I. Representation of Compounds for Machine-Learning Prediction of Physical Properties. *Phys. Rev. B* **2017**, *95*, 144110.
- (46) Sahoo, S.; Adhikari, C.; Kuanar, M.; Mishra, B. A Short Review of the Generation of Molecular Descriptors and Their Applications in Quantitative Structure Property/Activity Relationships. *Curr. Comput.-Aided Drug Des.* **2016**, *12*, 181–205.
- (47) Fisher, D.; Lukow, S. R.; Berezutskiy, G.; Gil, I.; Levy, T.; Zeiri, Y. Machine Learning Improves Trace Explosive Selectivity: Application to Nitrate-Based Explosives. *J. Phys. Chem. A* **2020**, *124*, 9656–9664.
- (48) Nayak, S.; Bhattacharjee, S.; Choi, J.-H.; Lee, S. C. Machine Learning and Scaling Laws for Prediction of Accurate Adsorption Energy. *J. Phys. Chem. A* **2019**, *124*, 247–254.
- (49) Wei, Y.; Chin, K.; Barge, L. M.; Hermis, N.; Wei, T.; Wei, T. Machine Learning Analysis of the Thermodynamic Responses of In Situ Dielectric Spectroscopy Data in Amino Acids and Inorganic Electrolytes. *J. Phys. Chem. B* **2020**, *124*, 11491–11500.
- (50) Nisbet, M. L.; Nolis, G. M.; Schrier, J.; Norquist, A. J.; Poepplmeier, K. R.; Cabana, J. Machine-Learning-Assisted Synthesis of Polar Racemates. *J. Am. Chem. Soc.* **2020**, *142*, 7555–7566.
- (51) Wexler, R. B.; Martirez, J. M. P.; Rappe, A. M. Chemical Pressure-Driven Enhancement of the Hydrogen Evolving Activity of Ni2P from Nonmetal Surface Doping Interpreted via Machine Learning. *J. Am. Chem. Soc.* **2018**, *140*, 4678–4683.
- (52) Lee, M.-H. Identification of Host-Guest Systems in Green TADF-Based OLEDs with Energy Level Matching Based on a Machine-Learning Study. *Phys. Chem. Chem. Phys.* **2020**, *22*, 16378–16386.
- (53) Landrum, G. RDKit: Open-source cheminformatics <https://www.rdkit.org/> (accessed Oct 23, 2021).
- (54) Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Scalmani, G.; Barone, V.; Petersson, G. A.; Nakatsuji, H.; Li, X.; Caricato, M.; Marenich, A.; Bloino, J.; Janesko, B. G.; Gomperts, R.; Mennucci, B.; Hratchian, H. P.; Ort, J. V.; Fox, D. J. *Gaussian 09*; Gaussian, Inc.: Wallingford CT 2016.
- (55) Lee, C.; Yang, W.; Parr, R. G. Development of the Colle-Salvetti Correlation-Energy Formula into a Functional of the Electron Density. *Phys. Rev. B: Condens. Matter Mater. Phys.* **1988**, *37*, 785.
- (56) Becke, A. D. Density-functional Thermochemistry. III. The Role of Exact Exchange. *J. Chem. Phys.* **1993**, *98*, 5648.
- (57) Tirado-Rives, J.; Jorgensen, W. L. Performance of B3LYP Density Functional Methods for a Large Set of Organic Molecules. *J. Chem. Theory Comput.* **2008**, *4*, 297–306.
- (58) Schuchardt, K. L.; Didier, B. T.; Elsethagen, T.; Sun, L.; Gurumothri, V.; Chase, J.; Li, J.; Windus, T. L. Basis Set Exchange: A Community Database for Computational Sciences. *J. Chem. Inf. Model.* **2007**, *47*, 1045–1052.
- (59) Zhao, Y.; Truhlar, D. G. Density Functionals with Broad Applicability in Chemistry. *Acc. Chem. Res.* **2008**, *41*, 157–167.
- (60) Papajak, E.; Truhlar, D. G. Efficient Diffuse Basis Sets for Density Functional Theory. *J. Chem. Theory Comput.* **2010**, *6*, 597–601.
- (61) Treitel, N.; Shenhar, R.; Aprahamian, I.; Sheradsky, T.; Rabinovitz, M. Calculations of PAH Anions: When Are Diffuse Functions Necessary? *Phys. Chem. Chem. Phys.* **2004**, *6*, 1113–1121.
- (62) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
- (63) Nakagawa, R.; Nishina, Y. Simulating the Redox Potentials of Unexplored Phenazine Derivatives as Electron Mediators for Biofuel Cells. *J. Phys. Energy* **2021**, *3*, 034008.
- (64) Miao, L.; Liu, L.; Zhang, K.; Chen, J. Molecular Design Strategy for High-Redox-Potential and Poorly Soluble n-Type Phenazine Derivatives as Cathode Materials for Lithium Batteries. *ChemSusChem* **2020**, *13*, 2337–2344.
- (65) Sousa, A. C.; Martins, L. O.; Robalo, M. P. Laccases: Versatile Biocatalysts for the Synthesis of Heterocyclic Cores. *Mol.* **2021**, *26*, 3719.
- (66) Castro, K. P.; Clikeman, T. T.; DeWeerd, N. J.; Bukovsky, E. V.; Rippey, K. C.; Kuvychko, I. V.; Hou, G. L.; Chen, Y. S.; Wang, X.

B.; Strauss, S. H.; Boltalina, O. V. Incremental Tuning Up of Fluorous Phenazine Acceptors. *Chem.—Eur. J.* **2016**, *22*, 3930–3936.

(67) Wang, C.; Li, X.; Yu, B.; Wang, Y.; Yang, Z.; Wang, H.; Lin, H.; Ma, J.; Li, G.; Jin, Z. Molecular Design of Fused-Ring Phenazine Derivatives for Long-Cycling Alkaline Redox Flow Batteries. *ACS Energy Lett.* **2020**, *5*, 411–417.

(68) 1.1. Linear Models—scikit-learn 1.0 documentation https://scikit-learn.org/stable/modules/linear_model.html#bayesian-regression (accessed Oct 23, 2021).

(69) Wipf, D.; Nagarajan, S. A New View of Automatic Relevance Determination. *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems*. 2007.

(70) 1.7. Gaussian Processes—scikit-learn 1.0 documentation https://scikit-learn.org/stable/modules/gaussian_process.html (accessed Oct 23, 2021).

(71) Sit, H. Quick Start to Gaussian Process Regression <https://towardsdatascience.com/quick-start-to-gaussian-process-regression-36d838810319> (accessed Oct 23, 2021).

(72) 1.3. Kernel ridge regression—scikit-learn 1.0 documentation https://scikit-learn.org/stable/modules/kernel_ridge.html (accessed Oct 23, 2021).

(73) 1.4. Support Vector Machines—scikit-learn 1.0 documentation <https://scikit-learn.org/stable/modules/svm.html#svm-regression> (accessed Oct 23, 2021).

(74) `sklearn.feature_selection.SelectKBest`—scikit-learn 1.0.2 documentation https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html (accessed Jan 7, 2022).

(75) 4.2. Permutation feature importance—scikit-learn 1.0 documentation https://scikit-learn.org/stable/modules/permutation_importance.html#permutation-importance (accessed Oct 23, 2021).

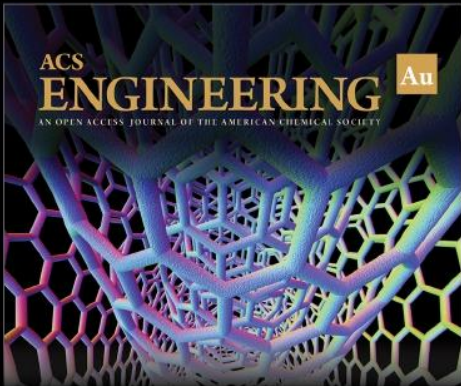
(76) 1.1. Linear Models—scikit-learn 1.0 documentation https://scikit-learn.org/stable/modules/linear_model.html (accessed Oct 21, 2021).

(77) Landrum, G. Getting Started with the RDKit in Python—The RDKit 2020.03.1 documentation <https://www.rdkit.org/docs/GettingStartedInPython.html#list-of-available-descriptors> (accessed March 31, 2021).

(78) QuaSAR-Descriptor <http://www.cadaster.eu/sites/cadaster.eu/files/challenge/descr.htm> (accessed Oct 22, 2021).

(79) `rdkit.Chem.MolSurf` module—The RDKit 2021.09.1 documentation <https://www.rdkit.org/docs/source/rdkit.Chem.MolSurf.html> (accessed Dec 29, 2021).


(80) Gasteiger, J.; Marsili, M. Iterative Partial Equalization of Orbital Electronegativity—a Rapid Access to Atomic Charges. *Tetrahedron* **1980**, *36*, 3219–3228.




ACS
ENGINEERING Au
AN OPEN ACCESS JOURNAL OF THE AMERICAN CHEMICAL SOCIETY

Editor-in-Chief: **Prof. Shelley D. Minter**, University of Utah, USA

Deputy Editor:
Prof. Vivek Ranade
University of Limerick, Ireland

Open for Submissions 

pubs.acs.org/engineeringau  ACS Publications
Most Trusted. Most Cited. Most Read.