

Design and application of scalable machine learning algorithms in molecular recognition, structure prediction and drug discovery

A thesis

Submitted in partial fulfilment of the requirements

Of the degree of

Doctor of Philosophy

By

Abhijit Gupta

20152021



INDIAN INSTITUTE OF SCIENCE EDUCATION AND RESEARCH PUNE

2021

*Dedicated to
my parents and teachers*

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.



Abhijit Gupta

Roll No.: **20152021**

Date: 31 August 2021

CERTIFICATE

Certified that the work incorporated in the thesis entitled “Design and application of scalable machine learning algorithms in molecular recognition, structure prediction and drug discovery” submitted by **Abhijit Gupta** was carried out by the candidate, under my supervision. The work presented here or any part of it has not been included in any other thesis submitted previously for the award of any degree or diploma from any other University or institution.

31 August 2021

Date:

Arnal Mukherjee
(Thesis Supervisor)

Acknowledgements

I acknowledge Indian Institute of Science Education and Research, Pune for providing a conducive research environment, institute fellowship and other amenities. I am grateful to my thesis supervisor, Dr Arnab Mukherjee for his mentoring, support, and intellectual freedom that he provided. His continuous encouragement, guidance and feedback have been immensely valuable. I am also thankful to my RAC members Dr M.S. Madhusudhan and Dr Leelavati Narlikar for their guidance, feedback, and helpful discussions. Dr Madhusudhan aroused my interest in programming and bioinformatics, and what followed is history. I am much obliged to Dr H. N. Gopi, who is our departmental chair. Much PhD work was done from home due to the Covid-19 pandemic and subsequent lockdowns. IISER Pune IT team provided excellent infrastructure to support and facilitate my work. Nisha Kurkure and Goldi Misra deserve a special mention for facilitating and maintaining GPU cluster access. I would also like to thank Dr Deepak Dhar, who taught me Statistical mechanics and engendered my interest in exploring statistics and probability theory in much greater detail. I am extremely grateful to my parents for their continuous encouragement and support in every possible form.

Last but not least, I would like to thank my friends at IISER Pune for the support and the fun we have had in the last five years. Although I cannot list all of them here, let me mention a few: Swati Deswal, Kanika Kohli, Rinku, Shubham Singh, Vineet Kumar Pandey, Unmesh, Naveen, and Shailendra Chaubey. The acknowledgement section would be incomplete without mentioning Intel corp and the exciting opportunities they provided me to explore the field of AI.

Contents

- 1) **Abstract (5)**
- 2) **Chapter 1: Introduction (6-14)**
- 3) **Chapter 2: Methodology (15-24)**
- 4) **Chapter 3: Accurate prediction of B-form/A-form DNA conformation propensity from the primary sequence: A machine learning and free energy handshake (25-48)**
- 5) **Chapter 4: Capturing Surface Complementarity in Proteins using Unsupervised Learning and Robust Curvature Measure (49-71)**
- 6) **Chapter 5: Prediction of good reaction coordinates and future evolution of MD trajectories using Regularized Sparse Autoencoders – A novel deep learning approach (72-89)**
- 7) **Chapter 6: Learning to learn – “What makes a molecule a prospective drug?” (90-101)**
- 8) **Appendix 1: (102-150)**
- 9) **Appendix 2: (151-156)**
- 10) **List of Publications (157)**

Abstract

Starting with the problem of structure prediction, we leveraged machine learning to predict DNA conformation from its sequence accurately. We developed an end-to-end data-driven approach using machine learning and free energy calculations to offer a fresh perspective on this long-standing problem. Besides accurately predicting the DNA conformation, our model also explains why certain sequences adopt a particular conformation.

Transitioning from the DNA to the world of proteins, we employed unsupervised learning (called hierarchical clustering) and our algebraic fitting algorithm to study the surface curvature of protein surfaces. We later used surface curvature to assess the shape complementarity among the interacting biomolecules, intending to devise a scoring algorithm for the fast selection of binders with complimentary curvature for a particular active site.

To find out the binding mechanism at the molecular level, one needs to identify the appropriate reaction coordinate. Therefore, our next endeavour was to devise a novel approach based on regularized sparse autoencoders – an energy-based model, to predict a useful and physically intuitive set of reaction coordinates.

Although finding strong binders is the first step towards finding a drug, it is not the most crucial step since all the binders to a receptor can not be characterized as drugs, which have to satisfy certain conditions called ADME condition. Therefore, finally, we tried to address this significant problem – “what makes a molecule a putative drug?”. We used representation learning in conjunction with modern graph neural network architectures to learn and predict crucial attributes behind the prospective drug-like activity. Overall, the goal of the studies carried out in the thesis is to find a fast selection of putative drugs.

Chapter 1: Introduction

1.1 Aim of this thesis

With the advent of big data and enormous computational power, machine learning continues to permeate every significant aspect of our life. There have been significant developments in this area over the last fifty years, particularly over the last two decades.

Machine learning's immediate applications are already quite wide-ranging, including image recognition, recommendation systems, fraud detection, and text and speech systems. With the rapid proliferation and advancement of AI, the technologies empowered by it have become invaluable tools in the various stages of the drug development process, such as identification and validation of drug targets, designing of new drugs, drug repurposing, improving the R&D efficiency, aggregating, and analysing biomedicine information and refining the decision-making process to recruit patients for clinical trials. It is expected that such a holistic AI approach can address the inefficiencies and uncertainties that arise in the classical drug development methods while minimising bias and human intervention in the process. The other uses of AI in drug development include the prediction of feasible synthetic routes for drug-like molecules¹, pharmacological properties², protein characteristics as well as efficacy³, drug combination and drug–target association⁴ and drug repurposing⁵. Additionally, machine learning techniques and predictive model software also contribute to the identification of target-specific virtual molecules and the association of the molecules with their respective target while optimising the safety and efficacy attributes.

In this thesis, I intend to touch upon the design and application of scalable machine learning algorithms in molecular recognition, drug design and drug discovery. The focus of the study has been on practically addressing the challenges of limited labelled data and designing fast and numerically robust algorithms at scale. When developing machine learning methods, we have to consider two things – "What is the learning goal?" and "What is the learning structure?". Although the answer to the first question depends on the problem of interest, the second question requires a substantial amount of work from the modeller's perspective. A priori, it is not easy to choose the model architecture or a specific machine learning algorithm, for that matter.

Several powerful machine learning algorithms with a high degree of expressivity have many hyperparameters, which require careful tuning to prevent overfitting. We also have to consider

the tradeoff between getting a good score on performance evaluation metrics and explainability. Essentially, if we want to extract meaningful inference from our model, then using a black-box approach is not the right choice. Therefore, in all the approaches developed in this thesis, we have focussed on interpretability.

Landscape and evolution of Machine learning-based research in Chemistry

An ongoing challenge in applied physical and chemical sciences has been to answer the question: how can one identify and make chemical compounds or materials with optimal properties for a given purpose? A substantial part of research concerns the discovery and characterisation of novel compounds that can benefit society, but most advances still are generally attributed to trial-and-error experimentation, and this requires significant time and cost. Current global challenges, especially the ones that manifested during the COVID-19 pandemic, create greater urgency for faster, robust, and less expensive research and development efforts. Computational chemistry methods have significantly improved over time, and they promise paradigm shifts in how compounds are fundamentally understood and designed for specific applications. Machine learning (ML) methods have witnessed an unprecedented and accelerated technological evolution, which have enabled a plethora of applications, some of which have become daily companions in our lives. This data-driven approach enables ML models to predict a wide range of material properties without requiring them to understand the chemistry or physics behind these properties.⁶ The past decade has seen a tremendous increase in the use of data-driven approaches for modelling molecules and materials. Atomistic and molecular dynamics simulation have been particularly fertile fields of use; applications range from the analysis and mining of large databases of materials properties⁷ to the design of molecules with the desired behavior for a given application⁸ (inverse design). Notably, deep generative models, which learn to generate the distribution of the data and also allows to sample from it, have been applied to numerous classes of materials: rational design of prospective drugs, finding synthetic routes to organic compounds, and optimisation of photovoltaics and redox flow batteries, as well as a variety of other solid-state materials⁸. Owing to the vast size of chemical space, which is estimated to be in the order of 10^{60} molecules, the task of successfully finding new drugs is daunting and predominantly the major hindrance in drug development. Simulation offers one way of probing this space without experimentation. Quantum mechanics govern the physics and chemistry of these molecules,

which can be solved via the Schrödinger equation to arrive at their exact properties. In practice, there is always a tradeoff between speed and accuracy, at least till the arrival of ML-based approaches. While state-of-the-art approximations to quantum problems impose severe computational bottlenecks, recent QML based developments indicate the possibility of substantial acceleration without sacrificing the predictive power of quantum mechanics⁹. While QML is still in its infancy, encouraging progress has been made. However, there is still a long way to go before we reach our goal of routinely designing and discovering new molecules and materials on computers. Some of the most fundamental problems, and also the most common tasks in quantum chemical computing, such as correctly predicting the ground-state energies and forces of new molecules or materials with high efficiency and accuracy, remain unsolved. These seemingly simple tasks are especially challenging when it comes to systems that are inherently highly distorted, charged or multi-referenced, or involve long-range non-bonded interactions. We believe these tasks are critical for subsequent, more challenging QML applications. Successful QML models can easily demonstrate their applicability by energy ranking of competing structures of real materials. In the context of material science, machine learning techniques are often used for property prediction, seeking to learn a function that maps a molecular material to the property of interest. To describe the general workflow of how machine learning can be incorporated in a field like organic chemistry, we generally follow the following steps: (1) Data set: By collecting published literature, databases, laboratories. Aggregate task-specific datasets by means of raw data, etc.; (2) Molecular description: Convert chemical molecules and reaction formulas into forms that can be recognised by algorithms; (3) Modular Model building: choose a model for a specific problem, choose an appropriate algorithm, Use the training set to train the model, and use the validation set to improve the model performance (4) Model application: use the trained model to predict unknown results; (5) Discussion and analysis: For prediction results (such as physical and chemical properties or reactivity properties, etc.) for attribution and interpretation. For this purpose, machine learning, deep learning, and AI have a potential role to play because their computational strategies automatically improve through experience.

In the world of open-access databases, there are also some open chemical databases, such as GDB-13¹⁰ and its sub-libraries QM7/QM7b, GDB-17¹¹ and its sub-libraries QM8, QM9, etc., have collected quantitative information on a large number of small molecule compounds. ZINC and ZINC-15 database¹² collected the 3D structures, suppliers, etc., of a large number of commercially available compounds, are recorded. After the data collection is completed,

another key question is how to transform it so that it is usable for ETL workflow (extract-transform-load). For transformation, the extracted data is converted into a form that the computer(our ML model) can recognise. One needs to consider - in what way to describe the molecule and whether the descriptor contains the implied meaning. The representation chosen will directly determine the predictive effect of the model. In the realm of semi-supervised learning, the feature engineering task is offloaded to the model itself. Mathematical representations of atomic configuration structures can not only be used as a starting point for supervised learning algorithms aimed at predicting their energies and properties¹³. It can also be used in conjunction with unsupervised learning schemes to compare structures to find repeating atomic patterns. We describe this approach in detail in chapter 6, where we use a graph as the data structure for representing molecules.

This method extracts various physicochemical parameters in molecules, such as log P, pKa and molecular weight, etc. These parameters are the final is aggregated into a set of feature vectors and used as input to train a machine learning model. The commonly used cheminformatics software, such as RDKit¹⁴ and CDK¹⁵, etc., can extract molecular features quickly and easily. In addition, high-precision molecular structure characterisation by semi-empirical or DFT theoretical calculations and physical and chemical parameters are also more commonly used descriptor generation methods.

The greatest value of machine learning here and in other related fields is the savings in time and resources. We envisage machine learning as an aid to guide experiments - rapidly evaluating many prospective drug-like molecules, assessment of very high numbers of new materials, which is not feasible with traditional experiments or ab initio models, and proposing candidates for further laboratory analysis. It is a tool that should be used in conjunction with the experiment, continually refining it and incorporating new data. The use of the two together can accelerate the progress of chemistry as a field.

1.2 Outline of the thesis:

1.2.1 Chapter 2: This chapter discusses the broad spectrum of different machine learning algorithms that we have used and designed. It starts with supervised learning, covering how we formulate the problem, discusses about model validation and regularization approaches, and then moves on to unsupervised learning, followed by semi-supervised learning algorithm at the end. It lays the basic foundations and motivation behind different approaches that we adopted.

1.2.1 Chapter 3: We begin with the application of machine learning in predicting DNA structure from its primary sequence. We show that *one can predict A- and B-DNA-forming sequences, the two prominent conformations of right-handed DNA helix, with ~93% accuracy.* While B-DNA is the ubiquitous and primary source of life form, the A-DNA conformation serves a crucial role in the formation of DNA complexes with polymerases, CAP binding, TBP-Binding and protection from DNA damage in various thermophilic and mesophilic bacteria. Moreover, an understanding of sequence specificity of B to A-DNA transition involved in the interaction of DNA with transposase, endonuclease and polymerase will unveil the possible hotspots of these biological processes. Therefore, prediction of the propensity of a given DNA sequence towards "A" or "B" form is an enticing problem and has several prospective applications.

Since the primary sequence encodes the structure of different forms of DNA, it should be possible to predict the structure from the sequence. The present study, which is the first of its kind, combines machine learning (ML) with thermodynamics to gain physical insights into why a specific sequence adopts a particular conformation. We incorporate the information obtained from *free energy values* for dinucleotide steps to explain the molecular and thermodynamic basis of the prediction made by our ML model. Our model provides a key insight into how the chemical nature of each dinucleotide step influences the final conformation attained by a given sequence. Surprisingly enough, we observed that the machine learning model discovered the intricate relation between each of the dinucleotide step's structural features and how it attributes towards its overall contribution in dictating a particular conformation.

When it comes to predictive modelling, robust statistical tests of performance are required to ensure that the model generalises well on unseen samples. We, therefore, rigorously tested our machine learning model using the nested stratified 5-fold cross-validation technique.

1.2.2 Chapter 4: In this chapter, we transition to the world of proteins. In this work, we developed a novel approach based on unsupervised machine learning and surface curvature to accurately measure the distribution of surface curvatures on a protein's surface. We used it to assess the degree of surface complementarity between two interacting partners. Proteins are functional elements in cellular machinery. They function through interactions with other proteins and biomolecules. Surface complementarity often governs these interactions, commonly known as analogous to the well-known "lock and key" mechanism.

However, due to several corrugations, a protein's surface poses significant challenges for its curvature-based characterisation. One of the key challenges is how to decide the appropriate size of a patch for curvature measurement. Previous approaches used a patch of fixed-sized radius, which failed to capture nuances in the surface topology of a protein. We overcame this difficulty by using an unsupervised machine learning-based approach that segments the protein surface effectively and automatically. Subsequently, we developed a fast, accurate, and numerically robust method based on algebraic fitting for measuring the surface curvature of a patch. We benchmarked our approach on known analytical surfaces and showed that our method is more accurate and faster than any previously known methods.

Once we establish the accuracy of curvature calculations and identification of surface topographies, we devised a scoring function based on curvature. We showed the existences of surface complementarity in various protein-protein and protein-ligand interactions based on our scoring function. It also detected subtle changes in proteins upon complexation with ligands that would not be otherwise detectible. This surface complementarity function will help detect a protein's active site's binding partners.

Our study can also be used to study the local curvature dynamics to understand the dynamical roles of protein surfaces in the presence and absence of binding partners.

1.2.3 Chapter 5: In this chapter, we tried to approach the problem of discovering an optimal set of reaction coordinates using self-supervised learning. Identifying reaction coordinates(RCs) is an active area of research, given the crucial role RCs play in determining

the progress of a chemical reaction^{16,17}. The choice of the reaction coordinate is often based on heuristic knowledge. However, an essential criterion for the choice is that the coordinate should capture both the reactant and product states unequivocally. Also, the coordinate should be the slowest one so that all the other degrees of freedom can easily equilibrate along the reaction coordinate¹⁸. Also, the coordinate should be the slowest one so that all the other degrees of freedom can easily equilibrate along the reaction coordinate. We used a regularised sparse autoencoder, an energy-based model, to discover a crucial set of reaction coordinates. Along with discovering reaction coordinates, our model also predicts the evolution of a molecular dynamics(MD) trajectory. We showcased that including sparsity enforcing regularisation helps in choosing a small but important set of reaction coordinates. We used two model systems to demonstrate our approach – alanine dipeptide system and proflavine and DNA system, which exhibited intercalation of proflavine into DNA minor groove in an aqueous environment. We model MD trajectory as a multivariate time series, and our latent variable model performs the task of multi-step time series prediction. This idea is inspired by the popular sparse coding approach - to represent each input sample as a linear combination of few elements taken from a set of representative patterns.

1.2.4 Chapter 6: This chapter takes self-supervised learning into the territory of graphs. SSL aims to learn "useful" representations of the input data without relying on human annotations. As we know, in the world of drug discovery, the known number of drugs(labelled data) are far fewer than the total number of molecules– there are only ~15,000 drugs, out of which ~4200 are approved ones. At the same time, the chemical space is combinatorically large. Owing to the vast size of chemical space, which is estimated to be in the order of 10^{60} molecules, the task of successfully finding new drugs is daunting and predominantly the major hindrance in drug development. This motivated us to adopt the strategy of learning apt representations of the drug-like molecules in the vast chemical space. SSL aims to learn "useful" representations of the input data without relying on human annotations. To apply this approach, we represent molecules as a graph. The graph data represents rich information, mainly the relation-based information, among the graph entities. We used Graph Neural Networks(GNNs) that offer an effective framework for representation learning on graph structures. We leveraged an attention-based mechanism with cardinality information for doing the aggregation¹⁹. The model was trained on a large unlabelled dataset comprising ZINC15²⁰, CheEMBL²¹, and QM9²² dataset using the SSL approach with a contrastive loss²³. We later used these representations for the downstream task of predicting drug-likeness via transfer learning.

1.2.5 Annexures: This section contains supplementary texts relevant to different chapters and details about mathematical notation used in the thesis.

References

- (1) Merk, D.; Friedrich, L.; Grisoni, F.; Schneider, G. De Novo Design of Bioactive Small Molecules by Artificial Intelligence. *Mol Inform* **2018**, *37* (1–2). <https://doi.org/10.1002/minf.201700153>.
- (2) Klopman, G.; Chakravarti, S. K.; Zhu, H.; Ivanov, J. M.; Saiakhov, R. D. ESP: A Method To Predict Toxicity and Pharmacological Properties of Chemicals Using Multiple MCASE Databases. *J. Chem. Inf. Comput. Sci.* **2004**, *44* (2), 704–715. <https://doi.org/10.1021/ci030298n>.
- (3) Menden, M. P.; Iorio, F.; Garnett, M.; McDermott, U.; Benes, C. H.; Ballester, P. J.; Saez-Rodriguez, J. Machine Learning Prediction of Cancer Cell Sensitivity to Drugs Based on Genomic and Chemical Properties. *PLOS ONE* **2013**, *8* (4), e61318. <https://doi.org/10.1371/journal.pone.0061318>.
- (4) Nascimento, A. C. A.; Prudêncio, R. B. C.; Costa, I. G. A Multiple Kernel Learning Algorithm for Drug-Target Interaction Prediction. *BMC Bioinformatics* **2016**, *17* (1), 46. <https://doi.org/10.1186/s12859-016-0890-3>.
- (5) Schneider, G. Automating Drug Discovery. *Nature Reviews Drug Discovery* **2018**, *17* (2), 97–113. <https://doi.org/10.1038/nrd.2017.232>.
- (6) Cao, B.; Adutwum, L. A.; Oliynyk, A. O.; Luber, E. J.; Olsen, B. C.; Mar, A.; Buriak, J. M. How to Optimize Materials and Devices via Design of Experiments and Machine Learning: Demonstration Using Organic Photovoltaics. *ACS nano* **2018**, *12* (8), 7434–7444.
- (7) Isayev, O.; Fourches, D.; Muratov, E. N.; Oses, C.; Rasch, K.; Tropsha, A.; Curtarolo, S. Materials Cartography: Representing and Mining Materials Space Using Structural and Electronic Fingerprints. *Chemistry of Materials* **2015**, *27* (3), 735–743.
- (8) Sanchez-Lengeling, B.; Aspuru-Guzik, A. Inverse Molecular Design Using Machine Learning: Generative Models for Matter Engineering. *Science* **2018**, *361* (6400), 360–365.
- (9) Pereira, F.; Aires-de-Sousa, J. Machine Learning for the Prediction of Molecular Dipole Moments Obtained by Density Functional Theory. *Journal of cheminformatics* **2018**, *10* (1), 1–11.
- (10) Blum, L. C.; Reymond, J.-L. 970 Million Druglike Small Molecules for Virtual Screening in the Chemical Universe Database GDB-13. *Journal of the American Chemical Society* **2009**, *131* (25), 8732–8733.
- (11) Ruddigkeit, L.; Van Deursen, R.; Blum, L. C.; Reymond, J.-L. Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17. *Journal of chemical information and modeling* **2012**, *52* (11), 2864–2875.
- (12) Sterling, T.; Irwin, J. J. ZINC 15 - Ligand Discovery for Everyone. *J. Chem. Inf. Model.* **2015**, *55*, 2324.
- (13) Jin, W.; Derr, T.; Liu, H.; Wang, Y.; Wang, S.; Liu, Z.; Tang, J. Self-Supervised Learning on Graphs: Deep Insights and New Direction. *arXiv preprint arXiv:2006.10141* **2020**.
- (14) Coley, C. W.; Green, W. H.; Jensen, K. F. RDChiral: An RDKit Wrapper for Handling Stereochemistry in Retrosynthetic Template Extraction and Application. *Journal of chemical information and modeling* **2019**, *59* (6), 2529–2537.

- (15) Beisken, S.; Meinl, T.; Wiswedel, B.; de Figueiredo, L. F.; Berthold, M.; Steinbeck, C. KNIME-CDK: Workflow-Driven Cheminformatics. *BMC bioinformatics* **2013**, *14* (1), 1–4.
- (16) Peters, B.; Trout, B. L. Obtaining Reaction Coordinates by Likelihood Maximization. *The Journal of chemical physics* **2006**, *125* (5), 054108.
- (17) Rohrdanz, M. A.; Zheng, W.; Maggioni, M.; Clementi, C. Determination of Reaction Coordinates via Locally Scaled Diffusion Map. *The Journal of chemical physics* **2011**, *134* (12), 03B624.
- (18) M. Sultan, M.; Pande, V. S. TICA-Metadynamics: Accelerating Metadynamics by Using Kinetically Selected Collective Variables. *Journal of chemical theory and computation* **2017**, *13* (6), 2440–2447.
- (19) Zhang, S.; Xie, L. Improving Attention Mechanism in Graph Neural Networks via Cardinality Preservation; NIH Public Access, 2020; Vol. 2020, p 1395.
- (20) Irwin, J. J.; Shoichet, B. K. ZINC - a Free Database of Commercially Available Compounds for Virtual Screening. *J. Chem. Inf. Model.* **2005**, *45*, 177.
- (21) Gaulton, A.; Bellis, L. J.; Bento, A. P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B.; Overington, J. P. ChEMBL: A Large-Scale Bioactivity Database for Drug Discovery. *Nucleic Acids Res.* **2012**, *40*, D1100.
- (22) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; Von Lilienfeld, O. A. Quantum Chemistry Structures and Properties of 134 Kilo Molecules. *Scientific data* **2014**, *1* (1), 1–7.
- (23) You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; Shen, Y. Graph Contrastive Learning with Augmentations. *Advances in Neural Information Processing Systems* **2020**, *33*, 5812–5823.

Chapter 2: Methodology

This chapter lays the foundation and motivation behind the different approaches we built and used in this thesis. Firstly, we look at the broad notion of using different types of Machine learning-based methods in Chemistry. We classify machine learning methods into four main classes based on the amount of supervision they receive during the training – supervised, unsupervised, semi-supervised, and reinforcement learning. The supervised learning approach is applicable when we have the labelled data. We have annotated data to train with, and the task involves predicting the label for a given sample.

In contrast, unsupervised learning involves capturing rich patterns in the data distribution. Self-supervised learning aims to learn good representation in the data for future downstream tasks. For the work presented here, we did not get the chance to explore the reinforcement learning approach. It involves getting feedback from the environment. The goal is to find a good behaviour, an action, or a label for each particular situation, if we will, to maximise the long-term rewards that the agent receives. Below we provide the overview and foundation of different machine learning algorithms that we used:

Supervised Learning:

The key tasks involved in supervised learning are classification and regression.

Classification: The model accepts the input data with labels and predicts the labels for the test data.

Regression: The model learns to understand the relationship between dependent and independent variables and predict a numerical value for the given input.

We describe the dataset D of M input/output pairs as follows:

$$D = \{(\mathbf{x}^{(i)}, y^{(i)})\}, i = 1, \dots, M$$

drawn from an underlying data distribution \mathbf{P} defined over $\mathbf{X} \times \mathbf{Y}$, where \mathbf{X} and \mathbf{Y} are respectively the data and label domain, where $x^{(i)} \in \mathbb{R}^d$. Note that for classification tasks, the output vector is discrete, i.e., $y \in \{0,1\}$ for a binary classification problem and a binary vector $\mathbf{y} \in \{0,1\}^k$ for multi-class classification and multi-label classification.

We express the result of running the machine learning algorithm as a learnt function f , which maps the input x to the output y . The precise form of the function f is determined during the training(learning) phase. If we assume that the labels y are generated from some unknown distribution f , the learning task reduces to estimating this function. To simplify it further, we assume that f can be estimated using a parametric family $\mathcal{F} = \{f_{\theta \in \Theta}\}$. This restriction has several benefits – it introduces an appropriate notion of regularity or inductive bias for f , and it also simplifies the task of estimating the unknown probability distribution of the data to estimating just parameters of some parametric probability distribution. For instance, if our data comes from a Normal distribution, we only need to estimate the mean and variance to encode the probability distribution during our learning phase. This model has a constant set of parameters, which is independent of the number of training samples (parametric model). After the learning phase, we have a ‘learned’ function \tilde{f} , which satisfies $\tilde{f}(x^{(i)}) \approx f(x^{(i)})$. We measure the performance of a learning algorithm on new samples drawn from \mathbf{P} , using some loss L ,

$$\mathcal{R}(\tilde{f}) = \mathbb{E}_{\mathbf{P}}[L(\tilde{f}(x), y)]$$

Where $\mathbb{E}_{\mathbf{P}}(\cdot)$ denotes the expected value over \mathbf{P} and \mathcal{R} is the expected value of loss function. Loss function tells us the loss we incur if we make a prediction u when the actual label is y .

Conversely, the number of parameters of a *non-parametric* approach(model) is a function of the training samples. For instance, the k-Nearest Neighbours classifier(a non-parametric algorithm) stores the entire training data - so the parameters that we learn are identical to the training set, and the number of parameters grows linearly with the training set size.¹ There are other challenges like the “curse of dimensionality” associated with high dimensional data. The basic problem is that the volume of space grows exponentially fast with dimension, so we might have to look quite far away in space to find our nearest neighbour. This leads to poor performance as the trouble with looking at neighbours that are so far away is that they may not be good predictors of the behaviour of the function at a given point.

Assessing Model accuracy

There is no free lunch in statistics – no single method dominates all others over all datasets.

Now, we give an overview of the different machine learning approaches we adopted in each chapter. We begin with the application of machine learning in predicting DNA structure from its primary sequence. Here, we demonstrated how a finely tuned lighGBM model (gradient

boosting) coupled with robust model validation and adjusting for class imbalance could be used to tackle a challenging problem. The `lightgbm`² is an ensemble model. The techniques in this class combine multiple machine learning models and aggregate their results to make predictions.

Ensembling is an effective way to improve performance and produce better predictions than any single model. No single machine learning model is perfect. To get a better understanding of where and how our model is wrong, we decompose the error of an ML model into the following parts: the irreducible error, the error due to bias, and the error due to variance. The irreducible error is the inherent error in the model resulting from a noisy dataset, the framing of the problem, or bad training examples, like measurement errors or confounding factors. We cannot do much about the *irreducible error*, just as the name implies. We call the error due to bias and variance *reducible error*, and here is where we can influence our model's performance. In short, bias is the inability of the model to learn enough about the relationship between the features and labels. At the same time, the variance captures the inability of a model to generalise on unseen examples. A model with high bias is said to be *underfitting*. A model with high variance has learned too much about the training data and is said to be *overfitting*. The goal of any machine learning model is to have low bias and low variance, but it is hard to achieve both in practice³. This phenomenon is known as the bias-variance trade-off. For example, increasing model complexity decreases bias but increases variance, while decreasing model complexity decreases variance but introduces more bias. The commonly used ensemble approaches are – bagging, boosting, and stacking.

Bagging (bootstrap aggregating) addresses high variance in machine learning models. This is a simple form of ensemble learning in which we fit M different base models to different randomly sampled versions of the data; this encourages the different models to make diverse predictions. The bootstrapping part of bagging refers to the datasets used for training the ensemble members. The datasets are sampled with replacement, so a given example may appear multiple times until we have a total of N examples per model (where N is the number of original data points). The disadvantage of bootstrap is that each base model only sees, on average, 63% of the unique input examples. The 37% of the training instances that a given base model does not use are called **out-of-bag instances** (oob). The predicted performance of the base model on these oob instances can be used as an estimate of test performance. This provides an alternative to cross validation^{4,5}. As seen in bagging, model averaging is a robust and reliable

method for reducing model variance. With random forest, the sub-models are all short decision trees (trees having limited depth).

Boosting

Boosting is another Ensemble technique. However, unlike bagging, boosting ultimately constructs an ensemble model with *more* learning capacity than the individual member models. For this reason, boosting provides a more effective means of reducing bias than a variance. An ensemble of trees can be represented as:

$$f(x; \theta) = \sum_{m=1}^M \beta_m F_m(x; \theta)$$

where F_m is one of the decision trees (weak learners). Each weak learner is marginally better than a random classifier, i.e., its accuracy is slightly better than 50%. For a binary classifier, we have $F_m \in \{-1, +1\}$. We first fit F_1 on the original data, assigning higher weights to the misclassified examples. Next, we fit F_2 to this weighted dataset. We keep iterating the process till we have fit M components. M or the number of weak learners is a hyperparameter, which controls the complexity of the ensemble model⁶. A hyperparameter is used to control the learning process, and unlike other model parameters, it is not derived during training. The hyperparameter can be chosen by different approaches like Bayesian optimisation, grid search, random search, or monitoring performance on the validation set in conjunction with early stopping². We discuss the strategies of choosing hyperparameters in the model training section of chapter 3, where we tried to address a common problem of having both class imbalance and less data. In such cases, the choice of hyperparameters becomes crucial for preventing overfitting. In the training procedure for boosting approach, successive classifiers are forced to emphasise samples that were misclassified by the previous classifiers. We also increase the weighting coefficients for the misclassified samples each iteration. The cost function, with an optional regulariser Ω is:

$$\mathcal{L}(f) = \sum_{i=1}^N \mathcal{I}(y_i, f(x_i)) + \Omega(f)$$

The loss l can be appropriately chosen. It is usually binary log loss for binary classification².

$$\hat{f} = \operatorname{argmin}_f \mathcal{L}(f)$$

imagine solving for $\hat{\mathbf{f}}$ by performing gradient descent in the space of functions. At a step m , let \mathbf{g}_m be the gradient of $\mathcal{L}(\mathbf{f})$ evaluated at $\mathbf{f} = \mathbf{f}_{m-1}$

$$g_{im} = \left[\frac{\partial l(y_i, f(\mathbf{x}_i))}{\partial f(\mathbf{x}_i)} \right]_{f=f_{m-1}}$$

Then, the update step is $\mathbf{f}_m = \mathbf{f}_{m-1} - \beta_m \mathbf{g}_m$, where β_m is the step length chosen as $\beta_m = \operatorname{argmin}_{\beta} \mathcal{L}(\mathbf{f}_{m-1} - \beta \mathbf{g}_m)$

The real-world performance of a model is severely affected if the model begins to overfit the data. In an ideal situation, we would have a large dataset to be able to train and validate our models (training samples) and have separate data for assessing the quality of our model (test samples). However, such data-rich situations are rare in the life sciences. In many practical applications, we seldom have the luxury of having a sufficiently large test set, which would provide an unbiased estimate of the generalization performance of our models. If we reserve too much data for training, it results in unreliable and biased estimates of the generalization performance; setting aside too much data for testing results in too little data for training, which hurts model performance. If the dataset is small and reserving data for independent test sets is not feasible, the nested cross-validation^{5,7} procedure offers a viable alternative. The above mentioned Ω term represents regularisation, which is any modification we make to a learning algorithm that is intended to reduce its generalisation error but not its training error. Generalisation error is the actual real-world performance on the test set. We have elaborately discussed model evaluation strategies in chapter 3.

Several strategies used in machine learning are explicitly designed to reduce the generalisation (test) error, possibly at the expense of increased training error. These strategies are known collectively as regularization.

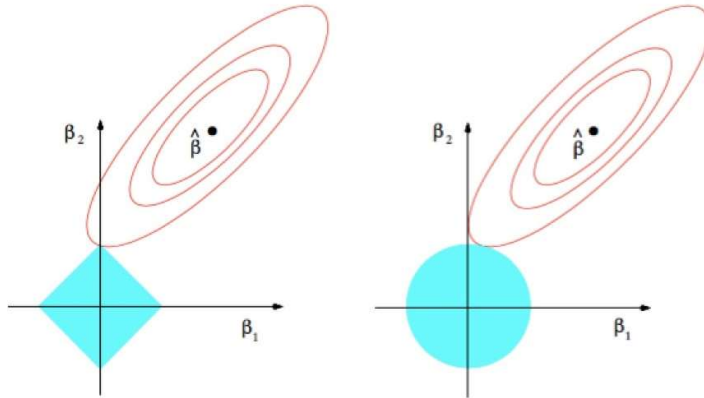


Figure 1: Illustration of L1 (left) vs L2 (right) regularisation

The differences between L1 and L2 regularisation:

- L1 regularisation penalises the sum of absolute values of the weights, whereas L2 regularisation penalises the sum of squares of the weights.
- The L1 regularisation solution is sparse. As illustrated in figure 1 above, the L1 regularization has sharp boundaries, and it is more probable for the regularised model to find a solution near corners, where we have only a few components non-zero, with others becoming zero. The L2 regularisation solution is non-sparse. The Euclidean norm does not encourage the sparsity constraint.
- L2 regularisation does not perform feature selection since weights are only reduced to values *near zero* instead of 0. L1 regularisation has built-in feature selection.
- L1 regularisation is robust to outliers; L2 regularisation is not. Since the difference between a wrongly predicted target value and the original target value will be quite large and squaring it will make it even larger when we have outliers.

Unsupervised Learning

In unsupervised learning, there is no target value. Here, the training set D consists of only input vectors:

$$D = \{\mathbf{x}^{(n)}\}_{n=1}^N$$

Each \mathbf{x} in D is a noisy observation of some unknown latent(hidden) variable \mathbf{h} such that, we have

$$\mathbf{x} = f(\mathbf{h})$$

Unlike in supervised learning, our aim here is to find both the unknown function f and hidden variables $\mathbf{h} \in \mathbb{R}^q$. This leads to latent variable models¹. Such models can be used to answer questions like – discovering subgroups among the variables or among the observations and dimensionality reduction. Often unsupervised learning is used as part of exploratory data analysis(EDA)⁸. There are, however, some challenges of unsupervised learning – there is no way to check our work because we do not know the true answer – the problem is hence called “unsupervised”. Despite this problem, unsupervised learning is still very useful for practical purposes. *Clustering* is one of the ways covered under very broad techniques for finding *subgroups*, or clusters, in a dataset. This procedure seek to partition a dataset into groups so that observations in each group are similar to each other, while observations in different groups are quite different from each other. In data mining and statistics, hierarchical clustering is a method of cluster analysis which seeks to build a hierarchy of clusters. There are other routinely used strategies like k-means, DBSCAN, Gaussian mixture models. Deep learning based clustering techniques iteratively group the features with a standard clustering algorithm, k-means or use a subnetwork like Graph neural networks, and use the subsequent assignments as supervision to update the weights of the network^{9,10}. In chapter 4, for the problem of partitioning the protein surface, we employed hierarchical clustering⁹ with farthest neighbour approach (complete-linkage clustering)¹¹ that would work on both convex and concave datasets. We provide below an overview of different types of hierarchical clustering.

Agglomerative: This is a "bottom up" approach: It recursively merges the pair of clusters that minimally increases a given linkage distance(metric for measuring distance). The linkage criterion determines which distance to use between sets of observation. The algorithm will merge the pairs of cluster that minimize this criterion. The most commonly used linkage criteria are – ward, farthest neighbour or complete-linkage clustering, single-linkage clustering, and average-linkage clustering.

Divisive: This is a "top down" approach: all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy

Farthest-neighbour or Complete-linkage clustering:

In the first step, every observation is perceived as a separate cluster. We then merge the two most similar/closest clusters together. The distance between the two clusters equals the distance between those two elements that are farthest away from each other. This process iteratively merges clusters and observations until we end up with one huge cluster containing all the observations. We are free to choose from several distance metrics (this needs to work in general for comparison of an observation with another observation, an observation with a cluster as well as for comparison of two clusters).

Next we look at **self-supervised learning** –

Supervised learning, using deep neural network, has achieved great success in the last decade. However, its heavy dependence on having a large, labelled dataset is its main limitation. The efficacy of machine learning techniques heavily relies on not only the design of the algorithms themselves, but also a good representation of data. As an alternative, self-supervised learning (SSL) attracts many researchers for its soaring performance on representation learning in the last several years. SSL leverages input data itself as supervision and benefits almost all types of downstream tasks. The intuition of SSL is to leverage the data's inherent co-occurrence relationships as the self-supervision, which could be versatile. From a statistical perspective, machine learning models are categorized into generative and discriminative models. Given the joint distribution $P(X, Y)$ of the input X and target Y , the generative model calculates the conditional probability $p(X|Y = y)$ while the discriminative model tries to model the $P(Y |X = x)$. Because most of the representation learning tasks hope to model relationships between x , for a long time, people believed that the generative model is the only choice for representation learning. However, recent breakthroughs in contrastive learning, such as Deep InfoMax, MoCo and SimCLR, shed light on the potential of discriminative models for representation. Contrastive learning aims at “learn to compare” through a Noise Contrastive Estimation (NCE)^{12,13} objective. We use contrastive learning approach on graphs for our problem in chapter 6. The question we wanted to ask was “Can we leverage huge unlabelled dataset of small molecules?”.

Graph representation learning aims at assigning nodes in a graph to low dimensional representations and effectively preserving the graph structures. To efficiently process graph data, the first key challenge is to find an efficient representation of graph data, i.e. how to represent graphs concisely so that advanced analytical tasks such as pattern discovery, analysis, and prediction can be efficiently performed in time and space. To better support network

inference, modern graph embeddings consider richer information in the graph. According to the type of information preserved in graph representation learning, existing methods can be divided into three types: (1) preserve graph structure and properties of graph embeddings, (2) graph representing learning with auxiliary information and (3) advanced information retention graph representation learning. In terms of technology, the different models are used to combine different types of information or address different goals. This commonly used models include matrix factorization, random walks, deep neural networks, and their variants¹⁴.

The basic idea of graph neural networks is to iteratively update node representations by combining representations of neighbors with their own representations. Starting from the initial node representation $H^0 = X$, in each layer we have two important functions:

AGGREGATE: which attempts to aggregate information from neighbors of each node;

COMBINE: update the node representations by combining the aggregated information from neighbors with the current node representations.

The aggregate and combine operations vary depending on the design choice. We used cardinal attention mechanism in our approach. Chapter 6 lists detail about specific use case for our application.

To conclude, these neural networks can usually divided into two categories, including supervised and unsupervised methods. The main differences between the different architectures are how messages are propagated between nodes, how messages from neighbors are aggregated, and how aggregated messages from neighbors are combined with the node representation itself. In the future, promising directions for graph neural networks include theoretical analysis to understand the behavior of graph neural networks, and apply them to various domains and domains such as recommender systems, knowledge graphs, drug and material discovery, computer vision and nature language comprehension.

References:

- (1) Bishop, C. M. *Pattern Recognition and Machine Learning*; Springer Science+ Business Media, 2006.
- (2) Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.-Y. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems 30*; Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc., 2017; pp 3146–3154.
- (3) Cawley, G. C.; Talbot, N. L. On Over-Fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. *J. Mach. Learn. Res.* **2010**, *11*, 2079–2107.
- (4) Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5.
- (5) Kohavi, R. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection; Montreal, Canada, 1995; Vol. 14, pp 1137–1145.
- (6) Bergstra, J. S.; Bardenet, R.; Bengio, Y.; Kégl, B. Algorithms for Hyper-Parameter Optimization; 2011; pp 2546–2554.
- (7) Varma, S.; Simon, R. Bias in Error Estimation When Using Cross-Validation for Model Selection. *BMC Bioinformatics* **2006**, *7* (1), 91.
- (8) Friedman, J.; Hastie, T.; Tibshirani, R. *The Elements of Statistical Learning*; Springer series in statistics New York, 2001; Vol. 1.
- (9) Murtagh, F.; Contreras, P. Algorithms for Hierarchical Clustering: An Overview. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2012**, *2* (1), 86–97.
- (10) Bo, D.; Wang, X.; Shi, C.; Zhu, M.; Lu, E.; Cui, P. Structural Deep Clustering Network. In *Proceedings of The Web Conference 2020*; WWW '20; Association for Computing Machinery: New York, NY, USA, 2020; pp 1400–1410. <https://doi.org/10.1145/3366423.3380214>.
- (11) Dawyndt, P.; De Meyer, H.; De Baets, B. The Complete Linkage Clustering Algorithm Revisited. *Soft Comput.* **2005**, *9* (5), 385–392.
- (12) Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A Simple Framework for Contrastive Learning of Visual Representations; PMLR, 2020; pp 1597–1607.
- (13) Oord, A. van den; Li, Y.; Vinyals, O. Representation Learning with Contrastive Predictive Coding. *ArXiv Prepr. ArXiv180703748* **2018**.
- (14) Jin, W.; Derr, T.; Liu, H.; Wang, Y.; Wang, S.; Liu, Z.; Tang, J. Self-Supervised Learning on Graphs: Deep Insights and New Direction. *ArXiv Prepr. ArXiv200610141* **2020**.

Chapter 3: Accurate Prediction of B-form/A-form DNA Conformation Propensity from Primary Sequence: A Machine Learning and Free energy Handshake

Introduction:

The prediction of a DNA conformation from the mere knowledge of its sequence presents an opportunity to presume its role in specific biological processes. The biological processes, such as direct and indirect readout mechanisms in protein-DNA interactions, exploit the conformational flexibility exhibited by DNA. The reduction in relative humidity around DNA due to the presence of other solvents like ethanol¹ or the presence of protein molecules² causes B-DNA to A-DNA transition. The A-DNA conformation is shorter and more compact compared to B-DNA. During B → A transition, the phosphate groups protrude out, and minor groove becomes broad and shallow, forming more water bridges in accordance with the theory of economy of hydration proposed by Saenger *et al.*³

The protein molecules such as transposase, endonuclease, and polymerase interact with B-DNA locally and convert a few dinucleotide steps to A-form in a whole DNA.² A-philic DNA segments exhibit low energy cost for deformation, and thus proteins bind to such hotspots during indirect recognition mechanism to commence the transcription process.² This mechanism is different from the sequence-specific mechanism where protein interacts with a specific nucleotide sequence at the binding site. The A-form also participates in the protection of bacterial cells under extreme UV exposure.⁴ Whelan and coworkers have shown fully reversible B→A-DNA transition in living bacterial cells on desiccation and rehydration using FTIR spectroscopy.⁵ Extremophiles like SIRV2 virus (*Sulfolobus islandicus rod-shaped virus* 2) survives at extreme temperatures of 80°C and acidity of pH three by adopting complete DNA in A-form, and aids protein to encapsidate DNA.⁶ The motors that drive double-stranded DNA (dsDNA) genomes into viral capsids are among the strongest of all biological motors for which forces have been measured. DNA plays an active role in force generation.

The "scrunchworm hypothesis" holds that the motor proteins repeatedly dehydrate and rehydrate the DNA, which then undergoes cyclic shortening and lengthening motions. The protein components of the motor dehydrate a section of the DNA, converting it from the B- to

A-form and shortening it by about 23%. The proteins then rehydrate the DNA, which converts back to the B form.⁷

Thus, it has become clear of late that A-DNA is merely not a non-functional conformation of DNA; it is an essential adaptation of DNA to survive harsh conditions. It is, therefore, intriguing to predict the sequence-structure relationship in DNA. Moreover, understanding sequence specificity of B-form (A-form transition) and an a priori detection of the A-philic segment in the genome will unveil the possible hotspots of certain biological processes in specific genes of organisms. We have developed a method based on machine learning to realize this apriori prediction of conformational preference of a given DNA sequence towards A-form or B-form with high accuracy. We also relate this conformational preference to the free energy cost of a dinucleotide step to be converted to A-form. We can employ this approach to design primers that are conformationally biased towards either A or B form and use them to study their impact in different biological processes.

The polymorphic nature of DNA makes the DNA conformation's prediction a challenging task. The local or partial B-form to A-form transition of a small segment of DNA sequence always possesses the penalty of B-form/A-form junction formation on both 5' and 3' ends of a newly formed A-DNA segment in a whole sequence⁸. Considering this aspect, we had previously performed rigorous umbrella sampling simulations to calculate this junction free energy values and characteristic local B-form to A-form free energy values for all ten unique dinucleotide steps.⁹ The free energy values obtained therein are termed as “absolute free energy” values (ΔG_a) as they are devoid of any effects from flanking base pairs. We have used these absolute free energy values in our inference model for explaining the effect and relative contribution of each dinucleotide step towards the conformational preference of a DNA sequence.

Previous studies:

There are only a few studies that attempted the prediction of DNA conformation from its sequence. Basham and coworkers derived A-DNA propensity energy (APE)¹⁰ based on the solvation free energy of trinucleotide steps to determine DNA structural preferences. However, APEs are unavailable for specific trinucleotide steps, thereby making this method inapplicable in general across a genomic DNA sequence. In a different approach, Tolstorukov and coworkers¹¹ formulated free energy models for all ten unique dinucleotide steps (D-12 model) and 32 individual trinucleotide steps (T-32 model) from experimental data of midpoints in

B→A-DNA transition studied earlier by others.^{12,13} The T-32 model was found to be more accurate than the D-12 model. It inherently considers stereochemical effects present along the B → A transition as it is based on three consecutive DNA base steps. However, the absence of the TAA/TTA free energy values limits the application of this dataset for a DNA structure prediction.

Schneider and coworkers developed an automated workflow¹⁴⁻¹⁶ to analyze DNA local conformations. They classified DNA dinucleotide steps based on local backbone conformations. It was observed that DNA structure exhibit mixed A-form/B-form traits in the backbone torsional space, even though the overall structure appears as either A-form or B-form. Their work demonstrated a high-resolution atlas of local DNA conformations.

In our approach, we have focused on the development of a general and more accurate method based on a machine learning (ML) approach that considers occurrences of all ten unique dinucleotide steps to predict the conformational preference of a given DNA sequence. In an ML-based approach, the inference is drawn based on observation alone. Therefore, although ML methods are suitable for prediction, the molecular or thermodynamic origin behind the prediction remains unknown. We have also built an explanatory model based on SHAP values for interpreting and explaining our model output to address this issue. This method also incorporates the information obtained from free energy values that we obtained earlier to explain the molecular and thermodynamic basis of the prediction made by our ML model.

Materials and Methods:

(a) Data Curation: The first step in an ML approach is data curation. Since we use a supervised learning approach, we collected A and B-DNA structures from the Nucleic Acid Database (NDB repository)^{18,19}. The corresponding sequences were retrieved from the RCSB PDB²⁰ database by a parser we wrote. We filtered out all redundant sequences along with all those sequences which had anything in addition to A, C, G, and T. Further, we have considered only the unbound double-stranded DNA structures. We removed all DNA sequences less than five base pairs long from our analysis as they are too short to be deciding a particular conformation. While selecting sequences for our study, we looked at the different experimental conditions under which different DNA structures were crystallized/labelled, namely - "Crystallization Method", "Temperature (K)", "pH", "Crystal Growth Procedure", "R-free values", "Percent Solvent Content". Notably, for X-Ray structures, we selected those sequences that

corresponded to structures with high R free values and resolution. For NMR-based structures, we considered the “Sample Temperature”, “Sample pH values”, “solvent system”, “Ionic strength”, and other relevant parameters. We have presented the distribution of different experimental conditions under which different structures were obtained in Section A, Supporting Information (Appendix 1). To minimize the influence of varying experimental conditions, we tried to select the sequences obtained under similar conditions. We also checked for outlier samples using the skewness adjusted Interquartile Range(IQR) method²¹(See section A, Appendix 1). It takes into consideration the skewness in the distribution for robust outlier detection. This helped us in getting those sequences for which the experimental conditions were similar, irrespective of the class label. Section A of the Appendix 1 shows kernel density estimation plots of each experimental condition for both A and B DNA samples that we included in our dataset.

We also performed the sequence similarity analysis across all sequences in a given class. We used the alignment-free sequence comparison approach that is based on the frequencies of k-mers (subsequences or words of length k)²². It considers the “Euclidean distance” between k-mers frequency profiles of two sequences as a measure of the dissimilarity between them. The pairwise distance matrix hence obtained is normalized between 0 and 1. Unlike alignment-based methods, the alignment-free method does not assume the contiguity of homologous regions. They are also less dependent on substitution/evolutionary models and are comparatively computationally inexpensive. The choice of k depends on the nature and the length of the sequences. Smaller k-mers should be used when sequences are obviously different (e.g., they are not related), whereas longer k-mers can be used for very similar sequences^{23,24}. For nucleotide sequences, k is usually set to 4-10 for smaller sequences, and $k = 8$ or 10 is typically used for comparing longer sequences^{23,25}. We considered $k = \{4,5,6\}$ for comparing sequence similarity. The mean sequence similarity is 31.9% for A-DNA samples and 28.7% for B-DNA samples in our curated dataset. In our dataset, the smallest sequences are of length six, and hence it is the upper bound on the choice of k.

Our curated dataset contained 192 samples, out of which 61 are A-DNA sequences, and 131 are B-DNA sequences. The list of curated DNA sequences along with resolution (Å), R-value, R-free (for crystallographic structures) and other relevant experimental conditions are mentioned in Appendix 1, section F (Dataset S1).

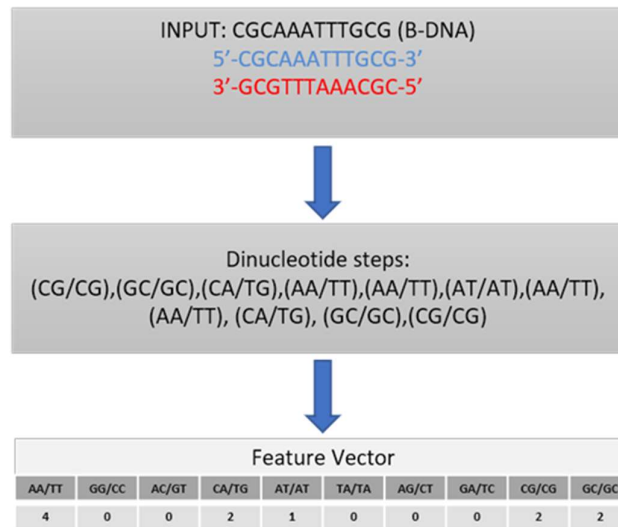


Figure 1: Schematic illustration of feature extraction.

(b) Feature extraction: Feature extraction or “feature design” is an essential step in any ML approach. The characteristics of any object are called features. In a DNA sequence, relevant features could be the length of the DNA, the number and types of dinucleotide steps, the number, and types of tetranucleotide steps. In this study, we have considered the count of all ten unique dinucleotide steps in the given DNA sequence as our feature vectors (Figure 1). There are two main rationales behind our choice: (i) first, the dinucleotide step represents the smallest possible building block for DNA conformation^{26,27}. (ii) second, we have used the absolute free energy values for each dinucleotide step⁹ in the model interpretation part, explaining how a particular conformation can be attributed to structural and chemical aspects associated with each dinucleotide step.

We want to mention that the lack of enough data precludes us from building a model that considers the relative positions of the different dinucleotide steps in a sequence. Such a model, although desirable, would require a large number of training samples for training. Our approach, on the other hand, offers a viable compromise.

(c) Pre-processing and adjusting the class imbalance: Data pre-processing involves the transformations that are applied to the data before feeding it to our machine learning models. For this classification problem, we have encoded the A-DNA samples as the positive class with the label ‘1’ and the B-DNA samples as the negative class with the label ‘0’. Some ML models like Support Vector Machines with radial basis function as the kernel²⁸ and models that use L1 and L2 regularisation assume that all features are centred around 0 and have variance in the

same order²⁹. We, therefore, standardized the features by removing the mean and scaling to unit variance. The standard score of a sample x is calculated as:

$$z = (x - u)/s,$$

where u is the mean of the training samples, and s is the standard deviation of the training samples. Centring and scaling happen independently on each feature by computing the relevant statistics on the samples in the training set.

We also observed a significant class imbalance (32% A-DNA vs 68% B-DNA curated, non-redundant sequences) that became apparent during the preliminary analysis. To address class imbalance issue, in which training data belonging to one class outnumber the examples in the other, we tried two different strategies during the training stage. First, adjusting the class weight - due to the imbalanced number of positive(A-DNA) and negative samples(B-DNA), the class weight option imposes a heavier penalty for errors in the minority class. Class weights are inversely proportional to class frequencies in the training data. The second strategy employed the SMOTE+TOMEK method³⁰, which is a combination of oversampling and undersampling. SMOTE is an oversampling method that synthesizes new plausible examples in the majority class. Tomek-Links refers to a method for identifying pairs of nearest neighbours in a dataset that have different classes. Removing one or both examples in these pairs makes the decision boundary in the training dataset less noisy or ambiguous. Despite the differences between the two approaches, they give similar improvements.

(d) Model Building. In this stage, we considered different machine learning algorithms for our problem. Classification of a sequence into A/B DNA is a binary classification problem. We tried LightGBM³¹ (based on gradient boosting decision tree), Support Vector Machine(SVM) classifier with “RBF” and linear kernel²⁸, Random Forest Classifier, Naïve Bayes Classifier, and Logistic Regression²⁸. Each model outputs the probability $p(C_k|x)$ of a class, $C_k = \{0, 1\}$, given a sequence x (0 represents B-DNA and 1 represents A-DNA). We then used an optimal threshold for converting this probability into class labels. When selecting a classification algorithm for a particular problem, one has to simultaneously select the best algorithm for that dataset and the best set of hyperparameters for the chosen model. These hyperparameters are intrinsic to each algorithm, and they define the model architecture. The accuracy of a model on unseen data is critically dependent on the choice of suitable values for the hyperparameters. The search for optimal values for the hyperparameters is a process known as model selection.

Machine Learning models like LightGBM have several hyperparameters. These are threshold parameter – *scale_pos_weight* for adjusting the threshold for an imbalanced dataset, regularization parameters – *L1* and *L2*, number of leaves(for controlling the complexity of the model), number of iterations, learning rate, bagging fraction, and bagging frequency. Even fairly simple generalized additive models like Logistic regression have hyperparameters like regularization, *class_weight* or threshold. Most of these models would perform poorly on the unseen data if one were to use the default set of hyperparameters. Hyperparameter optimization can be accomplished in several ways – one can exhaustively consider all parameter combinations using grid search, use randomized search strategy to sample a given number of candidates from a parameter space with a specified distribution or optimize the criterion of Expected Improvement (EI) using Gaussian Process(GP)/Tree-structured Parzen Estimator Approach(TPE). We chose to use the optimization of the EI criterion because it is intuitive and has been shown to work well in a wide variety of settings³². For tuning hyperparameters of our models, we used TPE approach implemented in Optuna framework³³.

We have used Intel Distribution for Python and Python API for Intel Data Analytics Acceleration Library (Intel DAAL) - named PyDAAL³⁴ — to boost machine-learning and data analytics performance. Using the advantage of optimized scikit-learn (Scikit-learn with Intel DAAL) that comes with it, we achieved faster training time and accurate results for the prediction problem.

(e) Training and evaluation. In an ideal situation, we would have a large dataset to be able to train and validate our models (training samples) and have separate data for assessing the quality of our model (test samples). However, such data-rich situations are rare in the life sciences. In many practical applications, we seldom have the luxury of having a sufficiently large test set, which would provide an unbiased estimate of the generalization performance of our models. If we reserve too much data for training, it results in unreliable and biased estimates of the generalization performance; setting aside too much data for testing results in too little data for training, which hurts model performance. If the dataset is small and reserving data for independent test sets is not feasible, the nested cross-validation^{35,36} procedure offers a viable alternative. Nested cross-validation(CV) can be used for choosing an appropriate classifier (model) and optimizing its hyperparameters to get a reliable and unbiased estimate of generalization performance.^{35,37} Model selection without nested CV uses the same data to tune

model parameters and evaluate model performance. Information may thus “leak” into the model and overfit the data, leading to a phenomenon called “overfitting in model selection”³⁷. We compared the performance of the machine learning algorithms, referred to as ML algorithms hereafter, by performing nested 5-fold stratified nested cross-validation. This process consists of two nested cross-validation loops, which are often referred to as inner(internal) and outer(external) cross-validation loops. We perform the model selection in the inner loop, and in the outer loop, we estimate the generalization performance (See Figure 2 for a schematic overview of nested CV). In the outer loop, our dataset is randomly split into five non-overlapping groups. Stratification is used to preserve the percentage of samples for each class. In each group, these two disjoint subsets are referred to as the training and the test set. In each group, the test set is exclusively used for model assessment. In the inner loop, the training set is used for model building and model selection. In each iteration of the inner loop, the incoming training set is repeatedly split into inner training and validation data sets by a stratified three-fold cross-validation approach. The inner training folds are used to derive different models by varying the hyperparameters (tuning parameters) of the model family at hand, whereas the validation sets are used to estimate the models’ performance. The hyperparameters corresponding to the model with the lowest cross-validation error across the inner folds are chosen for training the outer loop model. Along with tuning of hyperparameters, we also choose the optimal threshold via threshold-moving technique on the validation data. This involves choosing the threshold that corresponds to the maximum score on a chosen evaluation metric. For this purpose, we have chosen the F1 score metric. It tries to find the balance between precision and recall, which is extremely useful in scenarios when we are working with imbalanced datasets. Finally, in each iteration of the outer loop, we initialize the model with the tuned hyperparameters and threshold and use the test set to get an unbiased estimate of the selected model. We present below the pseudocode for the nested cross-validation algorithm:

For $i = 1$ to K_1 splits **do**: *//(outer loop)*

Split \mathcal{D} into \mathcal{D}_i^{train} , \mathcal{D}_i^{test} for the i' th split

For $j = 1$ to K_2 splits **do**: *//(inner loop)*

Split \mathcal{D}_i^{train} into $\mathcal{D}_j^{inner\ train}$, $\mathcal{D}_j^{validation}$ for the j' th split

sample parameter space (\mathcal{P}_{sets}) using random search and TPE to get P_j

Initialize and train model \mathcal{M} on $\mathcal{D}_j^{inner\ train}$ with hyperparameter set \mathcal{P}_j

Tune hyperparameters to get P_j^* and compute validation error $E_j^{validation}$ for \mathcal{M} with $\mathcal{D}_j^{validation}$

Select optimal hyperparameter set P^* from \mathcal{P}_{sets} , where $E_j^{validation}$ is the least

Train \mathcal{M} with \mathcal{D}_i^{train} , using P^* as hyperparameters

Compute test error metrics E_i^{test} for \mathcal{M} with \mathcal{D}_i^{test}

For assessment of the performance of our classification model, we have chosen accuracy, F1-score, Matthews correlation coefficient (MCC), ROC (Receiver Operating Characteristics) curve, and Precision-Recall(PR) curves as our primary evaluation metrics. When there is a class imbalance, the accuracy alone cannot give an accurate assessment of the performance of a classification model. A classifier may proclaim all data points as belonging to the majority class and obtain a high accuracy score while performing poorly on the prediction of minority class samples. Therefore, just using accuracy as the sole criterion for model evaluation can lead to over-optimistic inflated results, especially on imbalanced datasets. ROC represents a probability curve, and the area under the curve (AUC) of the ROC curve represents the measure of separability between the two classes. The higher the AUC-ROC score, the better the model is at distinguishing between A and B DNA samples. Precision is defined as the ratio of true positives and the sum of true positives and false positives. False positives are outcomes the model incorrectly labels as positive that are actually negative. In our example, false positives are B-DNA that the model classifies as A-DNA. In contrast, recall expresses the number of true positives divided by the sum of true positives and false negatives. In most problems pertaining to classification, one could give a higher priority to maximizing precision, or recall, depending upon the problem one is trying to solve. However, in general, there exists a more straightforward metric that takes into consideration both precision and recall. This metric is known as F1-score. It is the harmonic mean of precision and recall. Notably, the MCC coefficient considers true and false positives and negatives and is generally regarded as a balanced measure that can be used when there is a class imbalance.³⁸ It produces a more informative and truthful score in evaluating binary classifications than accuracy and F1 score.

The formulae of these metrics are mentioned in Supplementary Information, Section D. Section F of the Appendix 1 contains the list of all samples used for training and testing for each iteration of the outer loop.

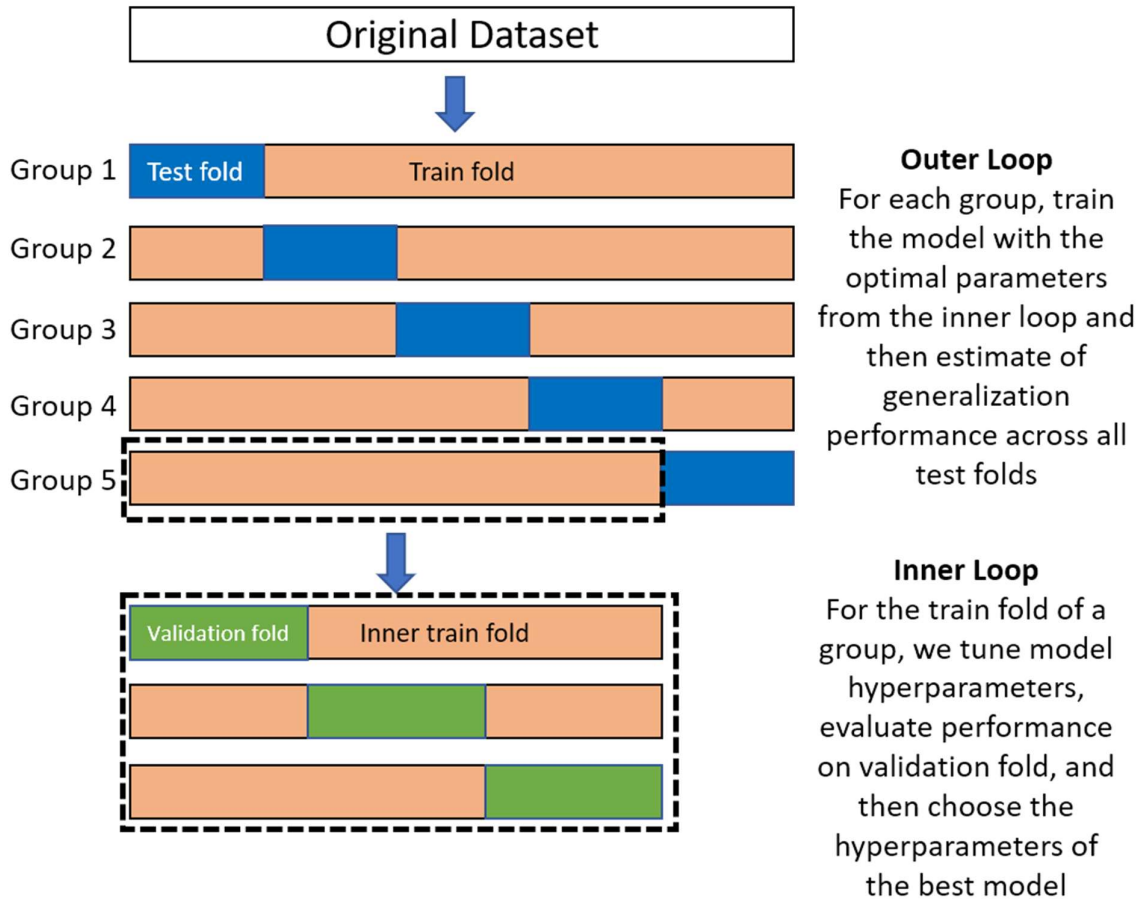


Figure 2: A schematic display of nested 5-fold stratified cross-validation. A set of n observations is randomly split into five non-overlapping groups in the outer loop. Each group contains approximately the same percentage of samples of each target class as the complete set (stratification). In the inner loop, each training fold is divided again for another round of cross-validation ($k=3$) to determine optimal hyperparameters for the classifier.

RESULTS

We describe here the results of the nested cross-validation performance of LightGBM algorithm across different metrics used for model assessment. We observed that LightGBM algorithm gave the best overall classification results across all five test sets in the nested CV. Figure 3 shows ROC Curves and Precision-recall Curves plotted across all five different test sets(folds). Table 1 shows performance metrics across test sets. We obtained a mean ROC AUC score of 0.97 ± 0.03 , a mean MCC score of 0.83, a mean accuracy score of 92.7%, a mean F1 score of 0.881, a mean AUC-PR of 0.956, and a mean average precision(average PR) of 0.957 on the test sets. The overall performance of our classifier summarized across different thresholds is given by the area under the ROC curve (AUC). Similar to the ROC curve, the *precision-recall (PR) curve* can be used to test all the possible positive predictive values and sensitivities obtained through a binary classification. They are especially valuable for assessing how well a machine learning model performs on the positive class (A DNA samples). A high area under the precision-recall curve represents both high recall and high precision. Table 2 displays the per-class performance across all test sets. We observe both high precision and high recall values for each class label. The weighted average returns the average score considering the proportion for each label in the dataset. In contrast, the macro average returns the average without considering the proportion for each label in the dataset. Furthermore, to ensure reproducibility, we also provide the values of tuned hyperparameters for each model and all datasets in section C of the Appendix 1. We also note that the detailed results of other approaches are deferred to the Section B of Appendix 1 (See Random Forest(Figure S4, Table S2), SVM Classifier(Figure S5, Table S3), Logistic Regression(Figure S6, Table S4), Naïve Bayes classifier(Figure S7, Table S5)).

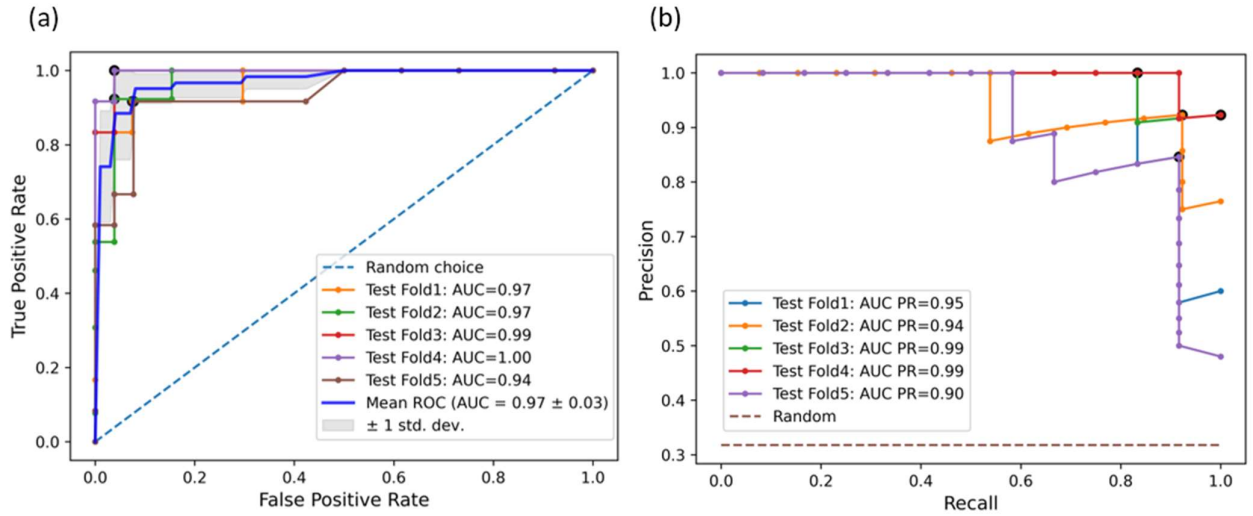


Figure 3: Nested Stratified five-fold cross-validation performance of the LightGBM model (a) ROC-AUC curves (b) Precision-Recall curves

Table 1: Classification performance of LightGBM algorithm with tuned hyperparameters (see section C, Appendix 1) across different test folds

	Average PR	AUC PR	ROC AUC	Accuracy	F1	MCC
Test Fold 1	0.954	0.952	0.969	0.923	0.857	0.822
Test Fold 2	0.946	0.944	0.973	0.923	0.880	0.825
Test Fold 3	0.987	0.986	0.994	0.947	0.917	0.878
Test Fold 4	0.994	0.993	0.997	0.947	0.917	0.878
Test Fold 5	0.906	0.904	0.939	0.895	0.833	0.756
Mean	0.957	0.956	0.974	0.927	0.881	0.832

Table 2: Detailed model evaluation report across different test folds

Test Fold 1		Precision	Recall	F1 score	Support
	B-DNA	0.90	1.00	0.90	27
	A-DNA	1.00	0.75	0.86	12
	macro average	0.95	0.88	0.90	39
	weighted average	0.93	0.92	0.92	39
Test Fold 2		Precision	Recall	F1 score	Support
	B-DNA	0.89	0.96	0.93	26
	A-DNA	0.91	0.77	0.83	13
	macro average	0.90	0.87	0.88	39
	weighted average	0.90	0.90	0.90	39
Test Fold 3		Precision	Recall	F1 score	Support
	B-DNA	0.96	0.96	0.96	26
	A-DNA	0.92	0.92	0.92	12
	macro average	0.94	0.94	0.94	38
	weighted average	0.95	0.95	0.95	38
		Precision	Recall	F1 score	Support

Test Fold 4	B-DNA	0.96	0.96	0.96	26
	A-DNA	0.92	0.92	0.92	12
	macro average	0.94	0.94	0.94	38
	weighted average	0.95	0.95	0.95	38
Test Fold 5		Precision	Recall	F1 score	Support
	B-DNA	0.96	0.92	0.94	26
	A-DNA	0.85	0.92	0.88	12
	macro average	0.90	0.92	0.91	38
	weighted average	0.92	0.92	0.92	38

In LightGBM, boosting helps in reducing bias and variance in ensemble-based models, which is particularly useful for controlling overfitting. It builds trees in a stage-wise forward manner, where weak-learners(trees) are added to address the shortcomings of existing weak-learners. As the end result, the model is able to achieve high accuracy by increasing the importance of “difficult” observations (samples which have complex non-linear decision boundary). As more trees are added, they rectify the misclassification error committed by existing learners. To control overfitting, we use the optimal value of regularization parameters – L1, L2 regularization, bagging fraction and frequency, number of leaves and feature fraction (Section C, Appendix 1). Another benefit of using gradient boosting is that after the boosted trees are constructed, it is relatively straightforward to retrieve importance scores for each attribute³⁹. To understand how individual dinucleotide steps affect the propensity of a sequence to assume a given conformation, we have used SHAP¹⁷ (SHapley Additive exPlanations). SHAP is a unified approach for explaining the output of any machine learning model. It connects game theory with local explanations, uniting several previous methods, and representing the only possible consistent and locally accurate additive feature attribution method based on

expectations¹⁷. This explanation model uses simplified inputs, which are toggling features on and off rather than raw inputs to the original model. Figure 4 shows the schematic models of SHAP, where data is processed using the original model and using the SHAP criteria as mentioned above. $g(z')$ is a linear function of binary variables (ON or OFF), which determines the role of individual inputs of features in the prediction. SHAP builds model explanations by asking the same question for every prediction and feature: “How does prediction i change when feature j is removed from the model?”, as mentioned above.

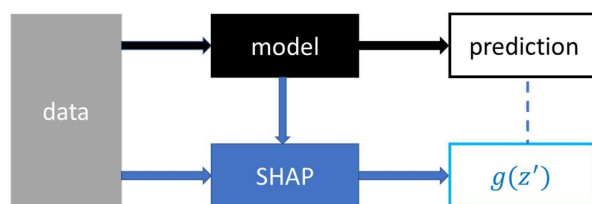


Figure 4: Schematics of SHAP model

To interpret and relate these SHAP values with thermodynamics, we describe the concept of the absolute free energy values (see section E, Supplementary Text of Appendix 1 and ref.⁹ for further details). Thermodynamically, the conformation of a particular structure depends on the free energetic stability. Therefore, the propensity of a sequence to adopt a particular conformation should depend on the overall free energy of the sequence in that conformation. Keeping that in mind, we had earlier calculated the free energy cost (Table 3) for the formation of A-form of each of the ten dinucleotide steps, as discussed below.⁹

The simulations were started from B-DNA conformations and modelled using the AMBER99/parmbsc0 force field. The structures were solvated using the TIP3P water model, and simulations were performed at physiological ion concentration (150 mM NaCl). We used umbrella sampling simulations along a new reaction coordinate Z'_p and average Z'_p ($\overline{Z'_p}$) for ten unique dinucleotide steps and a few trinucleotide steps embedded in the 13-mer DNA structure.⁸ These sequences, in general, can be presented as d(CGCGXXYYCGCG)₂, where X/Y can be either A, T, C, or G. The presence of CG sequences on both termini reduces the

possibility of base pairs fraying at the ends.¹⁸ We showed earlier that creating an A-form in a B-DNA creates two B/A junctions.

Table 3. List of absolute energy values (ΔG_a) for all ten possible dinucleotide steps.

Dinucleotide Steps	ΔG_a (kcal/mol)
AA/TT	2.34
GG/CC	0.86
AC/GT	1.91
CA/TG	2.40
AT/AT	2.29
TA/TA	1.59
AG/CT	0.67
GA/TC	0.84
CG/CG	3.06
GC/GC	1.33

* Please note, ΔG_j values were calculated only for homonucleotide steps and not heteronucleotide steps. ΔG_j is 1.59 kcal/mol for AA/TT and 0.52 kcal/mol for GG/CC.

Therefore, the free energy obtained for the dinucleotide step XY (underlined in 13-mer sequence) from simulation can be written as,

$$\Delta G_{sim}(XY) = \Delta G_j(XX) + \Delta G_a(XY) + \Delta G_j(YY). \quad \text{Eq. 1}$$

At this stage, we are only aware of $\Delta G_{sim}(XY)$ value. We then performed simulations on di- and tri- homonucleotide sequences $d(\text{CGCGXXXXXCGCG})_2$ to find the junction and absolute free energy values for homo-dinucleotide steps. The free energy cost to convert XX step along Z'_p in sequence $d(\text{CGCGXXXXXCGCG})_2$ can be decomposed as,

$$\Delta G_{sim}(XX) = \Delta G_j(XX) + \Delta G_a(XX) + \Delta G_j(XX). \quad \text{Eq. 2}$$

Also, the free energy cost to convert XXX step in the same sequence $d(\text{CGCGXXXXXCGCG})_2$ using an average Z'_p ($\overline{Z'_p}$) can be decomposed as,

$$\Delta G_{sim}(XXX) = \Delta G_j(XX) + 2\Delta G_a(XX) + \Delta G_j(XX) \quad \text{Eq. 3}$$

Subtracting Eq. 2 from Eq. 3, when a part of the DNA is converted from B-form to A-form. The full conversion of a B-DNA to A-DNA will depend only on the absolute free energy cost. That is the primary reason to calculate absolute free energy.

To get an idea about which features are most important for our model, we have plotted the SHAP values of each dinucleotide step(feature) for every sample. Figure 5 shows the SHAP summary plot, which sorts features by the sum of SHAP value magnitudes over all samples and uses these SHAP values to show the distribution of the impacts each feature has on the model output. The summary plot combines feature importance with feature effects. Each point on the summary plot is a Shapley value for a feature and an instance. The position on the y-axis is determined by the feature and on the x-axis by the Shapley value. The colour represents the value of the feature from low to high (red means high impact, blue means low impact). Overlapping points are jittered in the y-axis direction, so we get a sense of the distribution of the Shapley values per feature. The features are ordered according to their importance. We see that (AA/TT), a B-promoting dinucleotide step, and GG/CC, an A-promoting dinucleotide step, have the highest impact on our model prediction. The AA/TT step has the highest negative SHAP value, which corresponds to its highest contribution in predicting B-promoting DNA sequence. Similarly, the GG/CC and GC/GC have the highest positive SHAP value, which corresponds to their highest contribution in predicting A-promoting DNA sequences.

It is interesting to note that there is a strong concordance between these inferences drawn from our ML model with the absolute free energy values (Table 3). Figure 6 shows the standard bar plot obtained by taking the mean absolute value of the SHAP values for each feature. This plot

shows how each dinucleotide step(feature) contributes to the prediction of the propensity of A/B promoting DNA sequence.

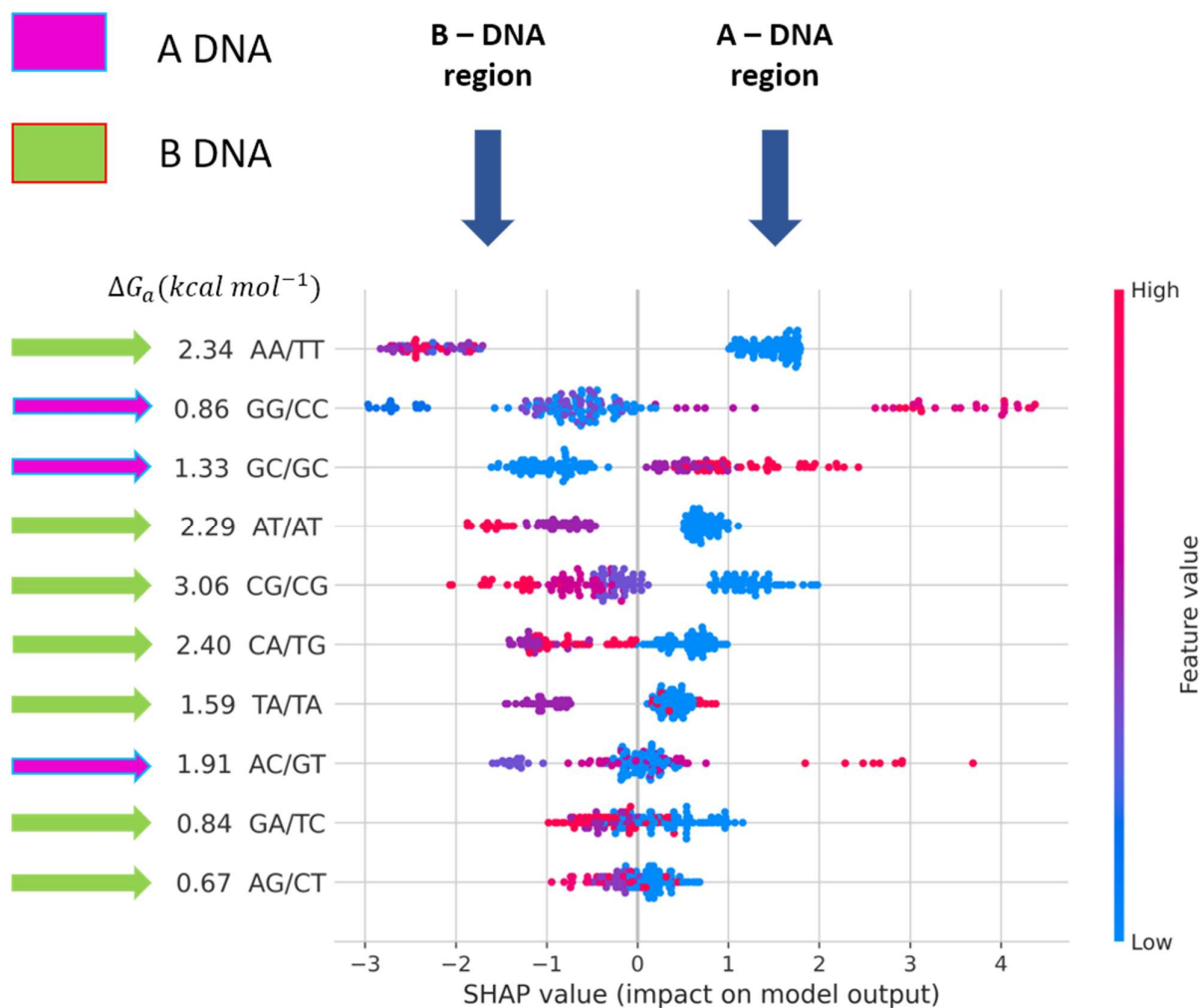


Figure 5: SHAP Summary. The plot above sorts features by the sum of SHAP value magnitudes over all samples and uses SHAP values to show the distribution of the impacts each feature has on the model output. The colour represents the feature value (red high, blue low). The horizontal scale represents the SHAP values, with the left-side indicating B-DNA region (negative values) and the right-side indicating A-DNA region (positive values). The absolute free energy value of each dinucleotide step is mentioned adjacent to its label.

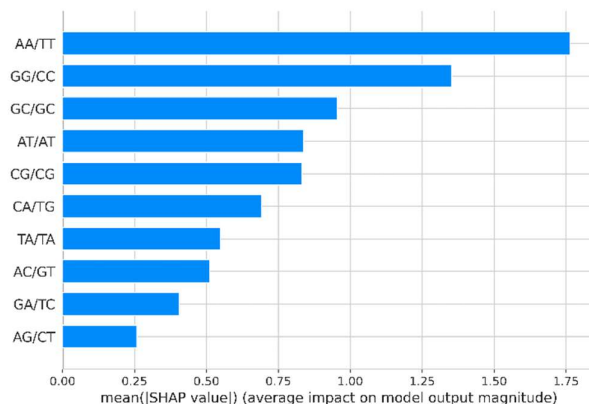


Figure 6: Mean of Absolute SHAP values show the average impact of each dinucleotide step in predicting whether a given sequence will attain A or B conformation.

DISCUSSION

In our approach, we have trained different machine-learning (ML) algorithms using a set of known A-DNA / B-DNA sequences. The best ML approach (LightGBM) provides prediction with a correctness of $\sim 93\%$ and an MCC score of 0.832. As it turns out, our model is able to capture the complex relationship between the feature vectors (dinucleotide steps) that attribute to the final conformation assumed by a DNA sequence. Understanding why a model makes a specific prediction can be as important as the prediction's accuracy in many applications. It is crucial when we want to understand how each fundamental dinucleotide step contributes towards the conformation attained by a sequence. The highest accuracy for large modern datasets is often achieved by complex models that are difficult to interpret, such as an ensemble of several models or deep learning models. LightGBM³¹ is an implementation of gradient boosting decision tree technique that offers a balanced tradeoff between accuracy and interpretability. For gaining further insight into the interpretability of our model, SHAP analysis was employed with which we could come up with a consistent and locally accurate additive feature attribution method based on expectations. Our study thus indicates that the conformational preference of a DNA lies in the fundamental free energetic driving force at a local dinucleotide level. Most of the DNA sequences used here, however, are short. Therefore, the cooperative effect may play a role in the case of longer DNA sequences, and an effort is underway to understand this. Our training set contains some hexamer or octamer A-DNA

sequences. Such short oligonucleotides are affected by crystal packing forces.^{40,41} At the same time, we also find NMR structures with A-form. We would also like to point out that crystal packing may have a role in producing A-form for certain sequences. However, this influence is limited to only a specific set of sequences. We believe that this has to do with *the inherent propensities of these sequences to adopt A-form*. Because of this, we do not find all short sequences adopting A-form. There are some short AT-rich sequences (PDB ID: 4U9M, 2G1Z) that are crystallized as B-DNA, as opposed to crystal packing derived A-DNA conformation. There are many 8 or 10-mer sequences that are in B form, just like not all A-DNA sequences are 8 or 10-mers. Therefore, it is not obvious that the length of the DNA sequence would dictate a particular conformation. The high predictive performance of our model indicates that there must be some inherent tendencies of these sequences to adopt A-form, and the objective of the present work is to capture that.

AA steps are highly B-philic due to the steric hindrance of their antisense counterpart TT step. A severe steric hindrance between protruding methyl groups of thymine base exists if it undergoes B→A transition and, thus, enhances the free energetic cost of the process. It is surprising that ML models can predict AA step as most B-philic step without the knowledge of the structure and interactions between the stacking base steps.

GG step is well-known to adopt or induce A-form in DNA sequences. Again, it is encouraging to note that the ML model can predict GG and GC as most A-philic steps without any structural information.

The DNA structures considered in the present study are assigned as B-DNA or A-DNA because these structures do not contain mixed A-form/B-form dinucleotide steps. We assume that even with mixed A/B traits at the local level, based on the definition of recent studies, the whole DNA structure appears as B / A due to the prominent conformational preference of each dinucleotide step of DNA. The cooperative effects of these dinucleotide steps contribute to the overall conformational preference in DNA oligonucleotides.

Finally, the classification of a sequence to A or B is based on the NDB data. Therefore, our goal was to apply the method to a given sequence and predict the A/B classification in conformity with the NDB (global) structural classification. As mentioned earlier, in our curated dataset, we tried to include sequences whose structures were obtained under similar experimental conditions to minimize the effect of varying experimental conditions.

At the moment, we are restricted by the paucity of a sufficient number of labelled DNA sequences. Out of 192 curated DNA sequences in the NDB dataset, 61 are A-DNA sequences, and 131 are B-DNA sequences (Appendix 1, section F [dataset S1]). Lack of enough data is one of the significant challenges in any machine learning model. Furthermore, the severe class imbalance between A and B DNA is another limitation, although we have adopted several measures to overcome these limitations.

The present study focuses only on canonical A-DNA or B-DNA conformation, with the objective of developing a method to understand the tendency of short DNA segments in long oligonucleotides to adopt these conformations. Thus, we have not considered non-canonical DNA structures. Furthermore, we acknowledge that there are subclasses of this broad classification⁴² - different A-form conformers, conformers bridging A to B form and vice versa, a separate Z form, subdivision of B-conformations into BI and BII form⁴³, which we could not categorize owing to the paucity of data in the NDB database. The eukaryotic and prokaryotic genomes contain DNA segments that can be easily converted to A-form (A-DNA promoter sequences, APS). Such APS allows transcription factor binding and could play a role in protein-DNA binding mechanisms.

Whitley and coworkers⁴⁴ used Basham's trinucleotide solvation free energy method of A/B DNA structure prediction¹⁰ to find out A-DNA promoter in *Xenopus tropicalis* genome. Owing to the limited applicability of the above method, we believe that our proposed machine learning model can be implemented on other genomes such as to find unknown A-DNA promoter DNA steps *a priori*. Further study is underway to explore eukaryotic genome analysis and the genome of organisms that survive under stringent conditions using A-form of DNA.

Summary and Conclusion:

In this study, we attempted to solve an important problem of finding out DNA conformation based on the primary nucleotide sequence, and we could attain, in a statistically rigorous way, a high accuracy (93%) of prediction and superior performance across classification metrics, despite the limitation of a small dataset. This was achieved through a nested cross-validation strategy that simultaneously provides an unbiased estimate of the generalization performance of a machine learning algorithm and allows one to tune the hyperparameters optimally. Also,

this process is free from selection bias, which may result by chance from using an arbitrary split of the data into a single test and train set or from opportunistically choosing a “favourable” test set. Besides being free from selection bias, nested CV also addresses the subtle but critical issue of “overfitting in model selection. Moreover, we employed various forms of regularization techniques which are available as hyperparameters in each algorithm. These regularization techniques are used in controlling overfitting. Furthermore, we also built a secondary model based on SHAP that showed the inference of the machine learning could be correlated with a completely different approach of thermodynamic propensities of a dinucleotide step in adopting a particular conformation. Our aim for the future is to apply this predictive approach to map segments of genomic sequence to their conformational preference and functions.

- (1) Franklin, R. E.; Gosling, R. G. Molecular Configuration in Sodium Thymonucleate. *Nature* **1953**, *171* (4356), 740–741. <https://doi.org/10.1038/171740a0>.
- (2) Lu, X.-J.; Shakked, Z.; Olson, W. K. A-Form Conformational Motifs in Ligand-Bound DNA Structures. *Journal of Molecular Biology* **2000**, *300* (4), 819–840. <https://doi.org/10.1006/jmbi.2000.3690>.
- (3) Saenger, W.; Hunter, W. N.; Kennard, O. DNA Conformation Is Determined by Economics in the Hydration of Phosphate Groups. *Nature* **1986**, *324* (6095), 385–388. <https://doi.org/10.1038/324385a0>.
- (4) Mohr, S. C.; Sokolov, N. V.; He, C. M.; Setlow, P. Binding of Small Acid-Soluble Spore Proteins from *Bacillus Subtilis* Changes the Conformation of DNA from B to A. *Proceedings of the National Academy of Sciences* **1991**, *88* (1), 77–81. <https://doi.org/10.1073/pnas.88.1.77>.
- (5) Whelan, D. R.; Hiscox, T. J.; Rood, J. I.; Bambery, K. R.; McNaughton, D.; Wood, B. R. Detection of an En Masse and Reversible B- to A-DNA Conformational Transition in Prokaryotes in Response to Desiccation. *Journal of The Royal Society Interface* **2014**, *11* (97), 20140454. <https://doi.org/10.1098/rsif.2014.0454>.
- (6) DiMaio, F.; Yu, X.; Rensen, E.; Krupovic, M.; Prangishvili, D.; Egelman, E. H. A Virus That Infects a Hyperthermophile Encapsidates A-Form DNA. *Science* **2015**, *348* (6237), 914–917. <https://doi.org/10.1126/science.aaa4181>.
- (7) Harvey, S. C. The Scrunchworm Hypothesis: Transitions between A-DNA and B-DNA Provide the Driving Force for Genome Packaging in Double-Stranded DNA Bacteriophages. *Journal of Structural Biology* **2015**, *189* (1), 1–8. <https://doi.org/10.1016/j.jsb.2014.11.012>.
- (8) Jacobo-Molina, A.; Ding, J.; Nanni, R. G.; Clark, A. D.; Lu, X.; Tantillo, C.; Williams, R. L.; Kamer, G.; Ferris, A. L.; Clark, P.; Hizi, A.; Hughes, S. H.; Arnold, E. Crystal Structure of Human Immunodeficiency Virus Type 1 Reverse Transcriptase Complexed with Double-Stranded DNA at 3.0 Å Resolution Shows Bent DNA. *Proceedings of the National Academy of Sciences of the United States of America* **1993**, *90* (13), 6320–6324.
- (9) Kulkarni, M.; Mukherjee, A. Computational Approach to Explore the B/A Junction Free Energy in DNA. *ChemPhysChem* **2016**, *17* (1), 147–154. <https://doi.org/10.1002/cphc.201500690>.
- (10) Basham, B.; Schroth, G. P.; Ho, P. S. An A-DNA Triplet Code: Thermodynamic Rules for Predicting A- and B-DNA. *Proceedings of the National Academy of Sciences of the United States of America* **1995**, *92* (14), 6464–6468.
- (11) Tolstorukov, M. Y.; Ivanov, V. I.; Malenkov, G. G.; Jernigan, R. L.; Zhurkin, V. B. Sequence-Dependent B \leftrightarrow A Transition in DNA Evaluated with Dimeric and Trimeric Scales. *Biophysical Journal* **2001**, *81* (6), 3409–3421. [https://doi.org/10.1016/S0006-3495\(01\)75973-5](https://doi.org/10.1016/S0006-3495(01)75973-5).

- (12) Minchenkova, L. E.; Schyolkina, A. K.; Chernov, B. K.; Ivanov, V. I. CC/GG Contacts Facilitate the B to A Transition of DMA in Solution. *Journal of Biomolecular Structure and Dynamics* **1986**, *4* (3), 463–476. <https://doi.org/10.1080/07391102.1986.10506362>.
- (13) Ivanov, V. I.; Minchenkova, L. E.; Minyat, E. E.; Schyolkina, A. K. Cooperative Transitions in DNA with No Separation of Strands. *Cold Spring Harb Symp Quant Biol* **1983**, *47*, 243–250. <https://doi.org/10.1101/SQB.1983.047.01.029>.
- (14) Svozil, D.; Kalina, J.; Omelka, M.; Schneider, B. DNA Conformations and Their Sequence Preferences. *Nucleic Acids Research* **2008**, *36* (11), 3690–3706.
- (15) Čech, P.; Kukul, J.; Černý, J.; Schneider, B.; Svozil, D. Automatic Workflow for the Classification of Local DNA Conformations. *BMC bioinformatics* **2013**, *14* (1), 205.
- (16) Schneider, B.; Božíková, P.; Čech, P.; Svozil, D.; Černý, J. A DNA Structural Alphabet Distinguishes Structural Features of DNA Bound to Regulatory Proteins and in the Nucleosome Core Particle. *Genes* **2017**, *8* (10), 278.
- (17) Lundberg, S. M.; Lee, S.-I. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30*; Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc., 2017; pp 4765–4774.
- (18) Berman, H. M.; Olson, W. K.; Beveridge, D. L.; Westbrook, J.; Gelbin, A.; Demeny, T.; Hsieh, S. H.; Srinivasan, A. R.; Schneider, B. The Nucleic Acid Database. A Comprehensive Relational Database of Three-Dimensional Structures of Nucleic Acids. *Biophys J* **1992**, *63* (3), 751–759.
- (19) Coimbatore Narayanan, B.; Westbrook, J.; Ghosh, S.; Petrov, A. I.; Sweeney, B.; Zirbel, C. L.; Leontis, N. B.; Berman, H. M. The Nucleic Acid Database: New Features and Capabilities. *Nucleic Acids Res* **2014**, *42* (D1), D114–D122. <https://doi.org/10.1093/nar/gkt980>.
- (20) Deshpande, N.; Address, K. J.; Bluhm, W. F.; Merino-Ott, J. C.; Townsend-Merino, W.; Zhang, Q.; Knezevich, C.; Xie, L.; Chen, L.; Feng, Z.; Kramer Green, R.; Flippen-Anderson, J. L.; Westbrook, J.; Berman, H. M.; Bourne, P. E. The RCSB Protein Data Bank: A Redesigned Query System and Relational Database Based on the MmCIF Schema. *Nucleic Acids Res* **2005**, *33* (suppl_1), D233–D237. <https://doi.org/10.1093/nar/gki057>.
- (21) Hubert, M.; Vandervieren, E. An Adjusted Boxplot for Skewed Distributions. *Computational statistics & data analysis* **2008**, *52* (12), 5186–5201.
- (22) Zielezinski, A.; Vinga, S.; Almeida, J.; Karlowski, W. M. Alignment-Free Sequence Comparison: Benefits, Applications, and Tools. *Genome biology* **2017**, *18* (1), 186.
- (23) Sims, G. E.; Jun, S.-R.; Wu, G. A.; Kim, S.-H. Alignment-Free Genome Comparison with Feature Frequency Profiles (FFP) and Optimal Resolutions. *Proceedings of the National Academy of Sciences* **2009**, *106* (8), 2677–2682.
- (24) Wu, T.-J.; Huang, Y.-H.; Li, L.-A. Optimal Word Sizes for Dissimilarity Measures and Estimation of the Degree of Dissimilarity between DNA Sequences. *Bioinformatics* **2005**, *21* (22), 4125–4132.
- (25) Chan, C. X.; Bernard, G.; Poirion, O.; Hogan, J. M.; Ragan, M. A. Inferring Phylogenies of Evolving Sequences without Multiple Sequence Alignment. *Scientific reports* **2014**, *4*, 6504.
- (26) Dickerson, R. E. Definitions and Nomenclature of Nucleic Acid Structure Components. *Nucleic acids research* **1989**, *17* (5), 1797–1803.
- (27) El Hassan, M.; Calladine, C. Conformational Characteristics of DNA: Empirical Classifications and a Hypothesis for the Conformational Behaviour of Dinucleotide Steps. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* **1997**, *355* (1722), 43–100.
- (28) Bishop, C. M. *Pattern Recognition and Machine Learning*; Information science and statistics; Springer: New York, NY, 2006.
- (29) Friedman, J.; Hastie, T.; Tibshirani, R. *The Elements of Statistical Learning*; Springer series in statistics New York, 2001; Vol. 1.

- (30) Batista, G. E. A. P. A.; Prati, R. C.; Monard, M. C. A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. *SIGKDD Explor. Newsl.* **2004**, *6* (1), 20–29. <https://doi.org/10.1145/1007730.1007735>.
- (31) Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.-Y. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems 30*; Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc., 2017; pp 3146–3154.
- (32) Bergstra, J. S.; Bardenet, R.; Bengio, Y.; Kégl, B. Algorithms for Hyper-Parameter Optimization; 2011; pp 2546–2554.
- (33) Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A Next-Generation Hyperparameter Optimization Framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*; KDD '19; Association for Computing Machinery: New York, NY, USA, 2019; pp 2623–2631. <https://doi.org/10.1145/3292500.3330701>.
- (34) Malakhov, A.; Liu, D.; Gorshkov, A.; Wilmarth, T. Composable Multi-Threading and Multi-Processing for Numeric Libraries. *PROC. OF THE 17th PYTHON IN SCIENCE CONF. (SCIPY 2018)* **2018**.
- (35) Varma, S.; Simon, R. Bias in Error Estimation When Using Cross-Validation for Model Selection. *BMC bioinformatics* **2006**, *7* (1), 91.
- (36) Kohavi, R. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection; Montreal, Canada, 1995; Vol. 14, pp 1137–1145.
- (37) Cawley, G. C.; Talbot, N. L. C. On Over-Fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. *Journal of Machine Learning Research* **2010**, *11* (70), 2079–2107.
- (38) Baldi, P.; Brunak, S.; Chauvin, Y.; Andersen, C. A. F.; Nielsen, H. Assessing the Accuracy of Prediction Algorithms for Classification: An Overview. *Bioinformatics* **2000**, *16* (5), 412–424. <https://doi.org/10.1093/bioinformatics/16.5.412>.
- (39) Mayr, A.; Binder, H.; Gefeller, O.; Schmid, M. The Evolution of Boosting Algorithms—from Machine Learning to Statistical Modelling. *arXiv preprint arXiv:1403.1452* **2014**.
- (40) Ramakrishnan, B.; Sundaralingam, M. Evidence for Crystal Environment Dominating Base Sequence Effects on DNA Conformation: Crystal Structures of the Orthorhombic and Hexagonal Polymorphs of the A-DNA Decamer d (GCGGGCCCGC) and Comparison with Their Isomorphous Crystal Structures. *Biochemistry* **1993**, *32* (42), 11458–11468.
- (41) Shakked, Z.; Guenstein-Guzikevich, G.; Eisenstein, M.; Frolow, F.; Rabinovich, D. The Conformation of the DNA Double Helix in the Crystal Is Dependent on Its Environment. *Nature* **1989**, *342* (6248), 456–460.
- (42) Schneider, B.; Božíková, P.; Nečasová, I.; Čech, P.; Svozil, D.; Černý, J. A DNA Structural Alphabet Provides New Insight into DNA Flexibility. *Acta Crystallographica Section D: Structural Biology* **2018**, *74* (1), 52–64.
- (43) Hartmann, B.; Piazzola, D.; Lavery, R. B I-B II Transitions in B-DNA. *Nucleic acids research* **1993**, *21* (3), 561–568.
- (44) Whitley, D. C.; Runfola, V.; Cary, P.; Nazlamova, L.; Guille, M.; Scarlett, G. APTE: Identification of Indirect Read-out A-DNA Promoter Elements in Genomes. *BMC bioinformatics* **2014**, *15* (1), 288.

Chapter 4: Capturing Surface Complementarity in Proteins using Unsupervised Learning and Robust Curvature Measure

Introduction

Protein performs its function through interaction with other molecules such as ligands, DNA, and other proteins. The three-dimensional structure of a protein provides the necessary shape and physicochemical texture to facilitate many of these interactions. The comparison of protein structures may identify functional relationships between proteins, even when no apparent sequence similarity is detected¹. The protein's molecular surface (MS) is a higher-level representation of its structure that models a protein as a continuous shape with geometric and chemical features. One of the molecular surface's important characteristics is its curvature, which measures how much a surface deviates from being flat.² Surface curvature is invariant under transformations like translation and rotation; it is an intrinsic property of a stable structure. The intuitive description of surface curvature is a major player in the molecular stereospecificity³, characterization of protein-protein, protein-nucleic acid interaction hotspots, membrane-protein interactions⁴, drug binding pockets⁵⁻⁷, and analysis of molecular solvation.⁸ Moreover, protein surface curvature may influence the hydrophobic effect, which is essential in understanding protein folding⁹⁻¹³. Local surface curvature can be used as a key descriptor for surface shape complementarity between proteins and their interacting partners.

Currently, there are a few methods to measure surface curvature. One of the classic and well-known methods is Connolly's solid-angle approach¹⁴. In this method, the centre of a sphere is placed at the molecular surface (Connolly surface¹⁵) (Fig. 1a). The solid angle, measured as the ratio of the sphere's surface area lying inside the protein surface to the sphere's total surface area, provides us with an estimate of the surface curvature. However, this method cannot discriminate between surfaces with the same solid angle but actually different curvatures¹⁶. As illustrated in Fig. 1a, it ignores the protein surface's topology that lies inside the sphere¹⁶. It only considers the points where the placed sphere and the protein surface intersect for surface curvature calculation.

The second class of methods employs the differential geometry approach, where the greatest and smallest curvatures, known as the principal curvatures of the surface, are calculated. The principal curvatures are then averaged to yield the mean curvature or multiplied together to yield the Gaussian curvature of the protein surface. Differential geometry-based approaches have been used to study why biomolecules assume complex structures and why biomolecular complexes admit convoluted interfaces between different parts.¹⁷ Depending on the nature of the representation used for molecular surface, its smoothness varies. These methods assume a continuous and differential representation of the surface, which is dependent on the nature of representation used for the protein's molecular surface. Some surface representations of a protein are rugged, with torus cusps and creases resulting from the intersection of molecular surface elements.¹⁸ To model the protein surface, Duncan and Olson used a Gaussian representation of protein atoms in part to overcome this problem¹⁹. An alternative approach, formulated by Tsodikov and co-workers, involves partitioning the surface into the continuous section and then calculating the average of each section's curvatures (*FastSurf* and *SurfRace*)²⁰. Several approaches use a functional based representation of molecular surface and then use iterative optimization to improve it. Bates et al.²¹ defined a hypersurface function with atomic constraints from biomolecular structural information and minimized the mean curvature of the hypersurface function through an iterative procedure. After minimization, a level surface is extracted from the steady-state hypersurface function to obtain the minimal molecular surface (MMS). A yet another approach, namely alpha shapes by Albou et al.²², classifies a protein's surface into knobs and clefts. They describe a novel conception of a surface patch (composed of 20 residues) by travelling along the surface from a central residue or atom. Recently, Gainza et al.²³ used a mesh-based representation of solvent excluded surface (SES) generated by the MSMS²⁴ program and used a distant-dependent curvature as one of the features in their geometric deep learning approach. Here, the protein mesh was decomposed into a set of overlapping patches of fixed geodesic radius ($r=9\text{\AA}$ or $r=12\text{\AA}$). Notably, they regularize the mesh after computing the MSMS surface, which is an expensive operation and one of the bottlenecks in their data preparation pipelines.

The third class of methods uses least-squares fitting (LSF) to fit an object with a known curvature to a given surface. The LSF class of methods has the differential geometry method's

advantages while also providing a quantitative curvature measure that is straightforward to apply and has a direct physical interpretation. Coleman *et al.* generated the least-squares fitted sphere to a surface patch and used the reciprocal of the sphere's radius as the curvature measurement¹⁶. Notably, the advantage of fitting a sphere to a surface patch is that a sphere can be fitted to any surface. It hence avoids the issues caused by differential geometry requirements for a smooth, differentiable surface¹⁹. Moreover, the surface of a sphere has the same curvature everywhere and hence offers a straightforward way to compare curvature values of different patches on a protein surface. Coleman *et al.* transformed the sphere-fitting problem into a solvable plane-fitting problem using a geometric transformation known as inversion.¹⁶

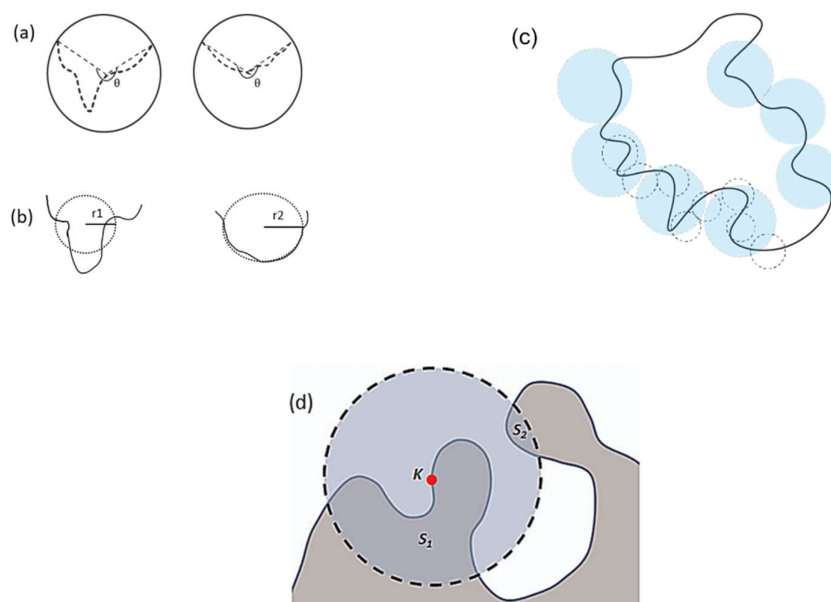


Figure 1: (a) Schematic illustration of the solid-angle curvature calculation method, (b) Schematic illustration of the LSF sphere method (Adapted from Coleman *et al.* 2005), (c) Two instances of the non-optimal division of a protein surface into patches. The blue coloured circles highlight sampling of the surface with a fixed patch of larger radius. The small, dotted circles highlight sampling of the surface with a fixed patch of smaller radius. A patch of a larger

radius fails to capture smaller cavities, and a patch of a smaller radius fails to capture larger cavities. **(d)** Schematic figure showing the importance of geodesic distance. Here, both S_1 and S_2 surface parts are enclosed within the sphere of a particular radius centred at K when Euclidean distance is used. However, the points of S_2 have greater geodesic distances than the predefined threshold G_{max} ; thus they are discarded.²⁵

However, Coleman's approach requires a fixed size radius to partition the protein surface, which results in a non-optimal division of the protein's surface, leading to inaccurate curvature measurement, as shown in Fig. 1c. Moreover, this approach further relied on an additional filtering criterion based on geodesic distance (shortest path on the surface connecting two points) from the centre to discard small unconnected surface parts enclosed within the sphere²⁶. Surface points with a distance greater than a predefined threshold were excluded from the surface²⁶.

In the present work, we have focussed on developing a fast, robust method for calculating the surface curvature of a given surface representation, which in our case is the SES surface generated by MSMS. We obviate the need for using such ad-hoc filtering by employing hierarchical clustering, a form of unsupervised learning, with the farthest neighbour approach²⁷, to segment the protein surface into contiguous patches efficiently. It gives us patches of varying size and ensures that each patch retains its intrinsic nature without discontinuities and retaining the nuances of surface topographies, i.e., an entire cavity or an entire protrusion will belong to a particular patch. Subsequently, we devised a fast, accurate, and numerically robust least square fitting method by extending the 'Hyperaccurate Algebraic fitting'²⁸ method for circle fitting to fit spheres on arbitrary surface patches. Note that the term "hyperaccurate" is used in the context of sphere fitting algorithm, popularized by the authors of "Error analysis of circle fitting algorithms"²⁸ We developed a scoring function based on the curvature and showed the existences of surface complementarity in various protein-protein and protein-ligand interactions, along with subtle changes in local curvature in proteins upon complexation with ligands that would not be otherwise detectible. This surface complementarity function can be helpful for detecting a protein's active site's binding partners.

METHODS

A schematic representation of our methodology is presented in Fig. 2, followed by a detailed description.

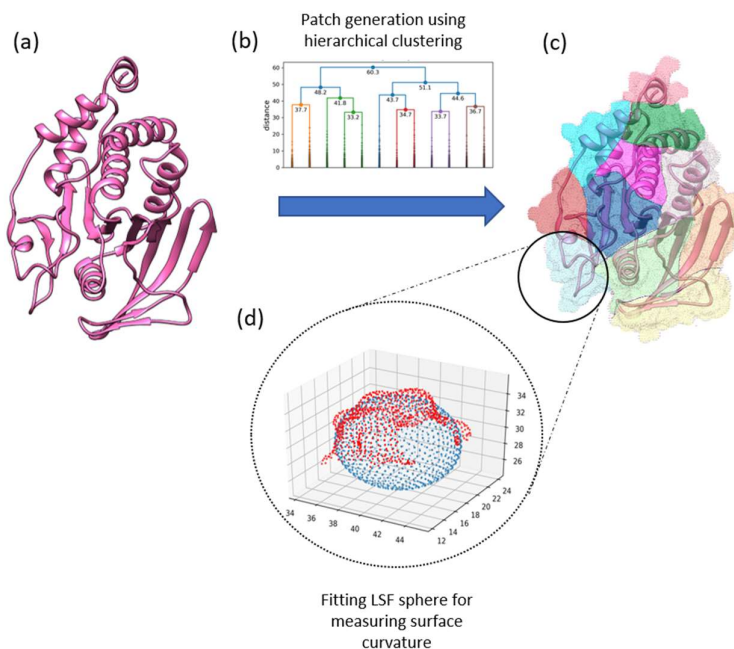


Figure 2: Illustration of our approach: (a) Ribbon representation of 2HNP protein, (b) illustration of the hierarchical clustering with the farthest neighbor approach for segregating the molecular surface of the protein's (2HNP). (c) Representative demonstration of final segregation of protein's surface into contiguous patches. (d) A representative surface patch (red dots) fitted a sphere (blue dots) using the 'Hyperaccurate algebraic fit' approach.

The steps involved are described in detail below.

a. Surface representation

For surface representation, we have used *solvent-excluded* molecular surface^{24,29}. The solvent-excluded molecular surface (SES) is defined as the boundary of the solvent-excluded molecular volume. The earliest methods for calculating the SES used the 'rolling ball' numerical integration method, in which a spherical probe of diameter as the size of water is rolled over the exposed contact surface of each atom²⁹. Currently, numerous algorithms exist for SES and solvent-accessible surface (SAS) calculations³⁰. We have used the MSMS program that provides a fast, analytical approach for calculating molecular surface²⁴. There are several programs for calculating SES. PQMS by Connolly et al. computes SES surfaces, but the

surfaces generated by it had self-intersecting elements¹⁸. MSMS program uses the reduced surfaces and attempts to address the singularities present in the computed SES. MSMS consists of four algorithms. The first computes the reduced surface of a molecule from which the second algorithm builds an analytical representation of the solvent-excluded surface that may be self-intersecting. The third algorithm removes all self-intersecting parts. The last algorithm produces a triangulation of the SES.²⁴ The SES generated by MSMS of Sanner et al. resolves all singularity issues associated with SES. The SES program by Connolly (PQMS), on the other hand, suffers from non-radial singularities, which cannot be differentiated²⁴. This program inputs a PDB file containing atom coordinates and radii and produces a triangulated solvent-excluded surface. The surface hence obtained is ‘dot molecular surface’ (DMS surface), whose triangulation density (number of vertices per Å²) could be adjusted for the desired accuracy. In our approach, we have used a density value of 3.0 (points/Å²) and a water probe radius of 1.5 Å.

b. Construction of surface patches using Complete-linkage clustering

A protein surface representation is non-uniform; it has cavities and protrusions of varying shapes and sizes. Previous approaches picked points within a local radius to define the patch for curvature measurement^{14,16}. However, this approach may only work for surfaces with relatively simple topology³¹ as it would always provide convex-shaped patches, which poorly represent the local topology of the protein surface [Fig. 1c].

Therefore, to capture the nuances of surface topography, we employed the unsupervised clustering approach, which attempts to combine “similar” elements into a particular group. This similarity criterion depends on the problem of interest. Here we assume that surface points belonging to a particular topography will be closer.

Among various unsupervised clustering methods such as k-means, hierarchical clustering, DBSCAN, mixture modelling, etc.³², we have employed hierarchical clustering²⁷ with farthest neighbour approach (complete-linkage clustering)³³ that would work on both convex and concave datasets. *Francetič et al.*³⁴ assessed the performance of different clustering methods when using concave sets of data and found that complete-linkage clustering (farthest neighbour clustering) gave the highest percentage (87.8%) of correctly assigned group membership with the lowest degree of data separation. It performs equally well in the case of the highest degree of data separation. Another advantage of this hierarchical approach is the automatic selection of the appropriate number of clusters, unlike k-means which requires the user to define the

number of clusters beforehand³². Therefore, hierarchical clustering naturally segregates a protein's surface into the relevant number of patches.

Moreover, it has a threshold parameter, i.e., 'resolution', that can be adjusted to select the desired features. The threshold parameter corresponds to the minimum geodesic distance connecting two points on the surface. Figure S1 and S2 in Appendix 2 illustrate how the threshold is used to partition the protein surface. Clustering tends to capture micro features, such as small ligand-binding pockets, at the lower threshold values; the generated patches are smaller in size and more significant in number. On the other hand, a larger threshold captures more prominent features like shape complementarity between proteins at the protein-protein interface. As the threshold increases, the patches grow larger in size and correspondingly fewer in number. An optimal resolution is essential to get meaningful values in protein surface patches. We varied these threshold parameters 10 to 20 (Figure 3) and observed that the curvature values and patch sizes do not vary significantly within the range between 10 to 15. However, at the threshold value of 20, the effect of an increase in patch size on curvature values is more pronounced. Therefore, we have used 15 as an optimum value for subsequent curvature calculations.

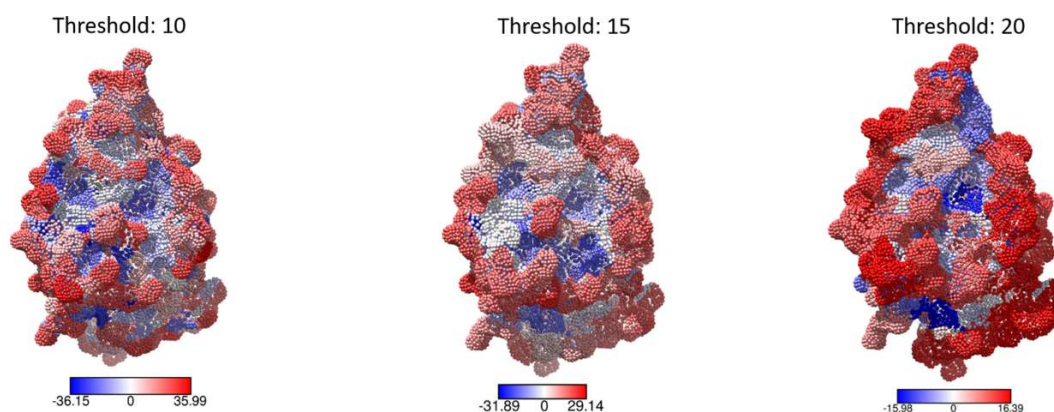


Figure 3: Surface curvature measurement at different levels of granularity for human protein tyrosine phosphatase (PDB: 2HNP). The figure shows the proteins with colour-coded curvature values where blue represents cavities and red represents protrusions. The colour intensity highlights the curvature of the surface. At the threshold value of 10, we observe many small regions of large positive and negative curvatures. These correspond to micro-level features like small pockets on a protein surface. With the increase in the threshold value, the clustering

procedure tends to form larger patches. These are features like cleft and more prominent protrusions.

c. Measurement of Surface Curvature of a patch by ‘Hyperaccurate’ algebraic sphere fitting:

Once we identify and segregate the local surface patches of the protein, we needed to calculate the curvature of these patches. Typically, if we could fit the surface patches to a sphere, we could obtain the curvature from the sphere's radius. There are geometric fit algorithms for fitting²⁸, but they are computationally expensive and suffer from issues of local minima, divergence, and strong dependency on initialization³⁵. Moreover, they can only be implemented iteratively, and their convergence rate is non-deterministic.³⁵ To address the above issues, we decided to use an algebraic fit algorithm that we developed by extending the “hyper-accurate algebraic” fit for circles. This approach has the least mean square error (MSE) and nearly zero bias.¹⁶ Here, we have extended that circle fitting algorithm into 3D to fit a sphere to a surface patch by modifying the objective function and the constraints. The extension maintains the non-iterative nature of the calculation with little MSE and bias, resulting in faster and more accurate curvature calculations, as shown in detail below.

We have adopted a standard functional model in which data points are noisy observations of some true points, $(\tilde{x}_1, \tilde{y}_1, \tilde{z}_1), \dots, (\tilde{x}_n, \tilde{y}_n, \tilde{z}_n)$; i.e.,

$$(x_1, y_1, z_1), \dots, (x_n, y_n, z_n)$$

$$x_i = \tilde{x}_i + \delta_i, \quad y_i = \tilde{y}_i + \epsilon_i, \quad z_i = \tilde{z}_i + \gamma_i,$$

where $(\delta_i, \epsilon_i, \gamma_i)$ represent isotropic Gaussian noise and are independent identically distributed normal random variables with mean zero and variance σ^2 . We could describe a sphere by the general equation,

$$A(x^2 + y^2 + z^2) + Bx + Cy + Dz + E = 0.$$

Therefore, our parameter vector is $\mathbf{A} = (A, B, C, D, E)$ and the corresponding data matrix can be written as,

$$Z \stackrel{\text{def}}{=} \begin{bmatrix} \hat{z}_1 & x_1 & y_1 & z_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \hat{z}_n & x_n & y_n & z_n & 1 \end{bmatrix},$$

where $\hat{z}_i = x_i^2 + y_i^2 + z_i^2$.

We defined the ‘matrix of moments’ as, $\mathbf{M} \stackrel{\text{def}}{=} \frac{1}{n} \mathbf{Z}^T \mathbf{Z}$. \mathbf{M} is a positive semi-definite matrix. The objective function is defined as, $F(\mathbf{A}) = \mathbf{A}^T \mathbf{M} \mathbf{A}$. To fit the data points to a sphere, we need to minimize the objective function $F(\mathbf{A}) = \mathbf{A}^T \mathbf{M} \mathbf{A}$, subject to a constraint $\mathbf{A}^T \mathbf{N} \mathbf{A} = 1$, where the matrix \mathbf{N} corresponds to the ‘Hyper-accurate’ fit.

$$\mathbf{N} = \begin{bmatrix} 8\bar{z} & 4\bar{x} & 4\bar{y} & 4\bar{z} & 2 \\ 4\bar{x} & 1 & 0 & 0 & 0 \\ 4\bar{y} & 0 & 1 & 0 & 0 \\ 4\bar{z} & 0 & 0 & 1 & 0 \\ 2 & 0 & 0 & 0 & 0 \end{bmatrix}$$

For solving the constrained minimization problem, we used Lagrange multiplier η and reduced the problem to an unconstrained minimization of the function,

$$\mathcal{G}(\mathbf{A}, \eta) = \mathbf{A}^T \mathbf{M} \mathbf{A} - \eta(\mathbf{A}^T \mathbf{N} \mathbf{A} - 1).$$

Differentiating with respect to \mathbf{A} and η gives

$$\mathbf{M} \mathbf{A} = \eta \mathbf{N} \mathbf{A}$$

and

$$\mathbf{A}^T \mathbf{N} \mathbf{A} = 1.$$

Thus, \mathbf{A} must be a generalized eigenvector for the matrix pair (\mathbf{M}, \mathbf{N}) , which also satisfies $\mathbf{A}^T \mathbf{N} \mathbf{A} = 1$. The above two equations may have several solutions. However, the right solution (η, \mathbf{A}) will satisfy the following condition,

$$\mathbf{A}^T \mathbf{M} \mathbf{A} = \eta \mathbf{A}^T \mathbf{N} \mathbf{A} = \eta.$$

Thus, for the purpose of minimizing $\mathbf{A}^T \mathbf{M} \mathbf{A}$ we should choose the solution with the smallest η (see the expression of $\mathcal{G}(\mathbf{A}, \eta)$ above). This objective function is convex, and hence we can use efficient convex optimization techniques³⁶ such as the singular value decomposition (SVD) approach. The SVD of \mathbf{Z} is written as $\mathbf{Z} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T$. If its smallest singular value, represented by σ_5 ($\mathbf{\Sigma}$ is a 5x5 matrix), is less than a predefined tolerance ϵ (chosen here as 10^{-12}), then \mathbf{A} (our parameter vector) is the corresponding right singular vector, i.e., the last column of \mathbf{V} . Otherwise, in regular cases when $\sigma_5 \geq \epsilon$, we form $\mathbf{Y} = \mathbf{V} \mathbf{\Sigma} \mathbf{V}^T$ and find the eigenpairs of the symmetric matrix $\mathbf{Y} \mathbf{N}^{-1} \mathbf{Y}$. Then, we select the eigenpair (η, \mathbf{A}^*) with the smallest positive eigenvalue and computing $\mathbf{Y}^{-1} \mathbf{A}^*$ completes the solution. The components of the parameter

vector hence calculated is then used to obtain radius and the centre of the fitted sphere for a patch.

To decide whether a patch is predominantly a protrusion or a cavity, we measured the distance from the centroid of the protein to a point in a patch $dist_{cp}$ and compared this distance with the distance between the centre of LSF sphere and the centroid $dist_{cc}$. If $dist_{cc} < dist_{cp}$ for more than half of the points in a patch, we classified that patch as a protrusion. For a cavity, $dist_{cc} > dist_{cp}$.

e. Representation of Protein Surface Curvature Graphics:

To generate protein surface coloured by curvature, the surface representation, i.e., dot molecular surface (DMS surface), was written to the PDB file. The B-factors for each entity was replaced with corresponding curvature value. The curvature values were scaled up to a relevant factor (chosen as $100 * \kappa$) to facilitate visualization and comparison.

RESULTS

a. Validation using analytical dataset and comparison of the runtime of our algorithm with the previous approach

To establish our method's accuracy and speed, we used a synthetic dataset containing surface points of five hundred spheres of randomly varying radii (within the range between 0.1 Å and 10 Å, in increments of 0.02 Å), with surface points density ranging between 10 and 5000. Figure 4(a) shows some representative spheres of random size and point density. To add the effect of corrugations and uncertainty, the points on the sphere were perturbed by the addition of zero-mean Gaussian noise. Our method's speed and accuracy are shown in Fig. 4 b,c. We have also compared our sphere fitting algorithm's runtime with the least square fitting (LSF) approach using inversion geometry proposed by Coleman et al.¹², which has shown to be the best method till now. Figure 4b shows the time required to calculate the curvature for each sphere arranged according to the number of surface points. Therefore, more surface points require more time. Our approach takes far less time than that by Coleman's method. Also, the fluctuation/variations in the time are also far less, indicating that our method's calculations'

runtime is much more predictable. The reason for high fluctuation in Coleman’s method is because the geometric fitting algorithms involve iterative approximations, which are computationally intensive and subject to occasional divergence. Our approach is numerically robust, fast, and non-iterative. Figure 4c shows the fitting errors (calculated from the known mean radius), which are lesser with lesser fluctuation than Coleman’s method, indicating the approach's high accuracy and numerical stability.

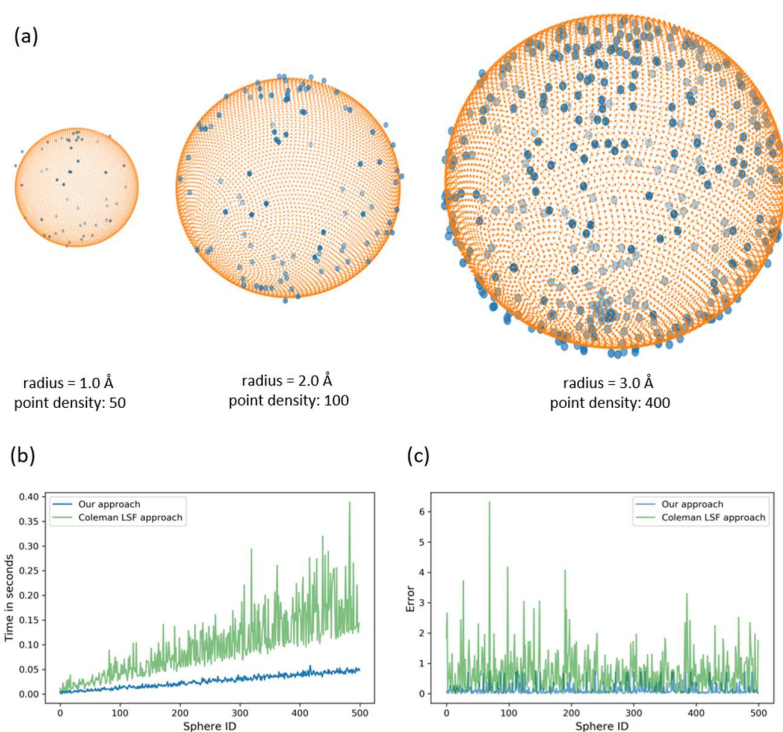


Figure 4: (a) Fitting sphere to surface points (b) Comparison of time taken by our approach to fit a given array of points perturbed slightly from an Ideal sphere with inversion geometry approach proposed by Coleman et al. (c) Comparison of error measured as the deviation from the actual radius of each sphere by our approach with Coleman *et al.* approach.

b. Measurement of the surface curvature of patches in a protein

Once we validated our approach on a known analytical dataset, we wanted to test it on real systems, i.e., protein surfaces. For that, we have chosen human protein tyrosine phosphatase in

both unbound (PDB ID: 2HNP) and bound (complexed with two phosphotyrosine molecules; PDB ID: 1PTY) states. This protein binds with tyrosine phosphatases (PTPs). It constitutes a family of receptor-like and cytoplasmic signal transducing enzymes that catalyze the dephosphorylation of phosphotyrosine residues and are characterized by homologous catalytic domains³⁷. We have calculated the curvatures of the protein's surface patches. Figure 5 shows the protein patches with different curvatures for the unbound tyrosine phosphatase (Fig. 5a) and the same in complexation with two phosphotyrosine molecules (Fig. 5b). The curvatures are colour-coded, with blue being the deepest cavity while red is the protrusion. Figure 5c, d show the distribution of curvatures of the patches for the uncomplexed and complexed proteins, respectively. This distribution captures the overall structure of the surface of the protein. We can see that the curvature distribution changes even for the binding of a small ligand. This is clearer in Figure 5e, which shows the curvature values of the residues where the ligand binds. There is a noticeable change in the curvature values when we go from the unbound (2HNP) state to the bound state (1PTY), indicating that our approach can capture the change in the protein's curvature upon complexation with the small molecules. (Note that we did not consider the ligand molecules while calculating the curvature of the bound structure. Therefore, although the two phosphotyrosine molecules bind in two deep clefts present in 2HNP, the small change in the curvature is captured through our calculations. This indicates that the present method could be used as a model to study how local binding of a ligand induces a global change in the structure of a protein using curvature as the metric, in conjunction with network analysis, which uses parameters such as cliques, clusters, and communities to study the effect of local ligand binding on the global structure of a protein³⁸.

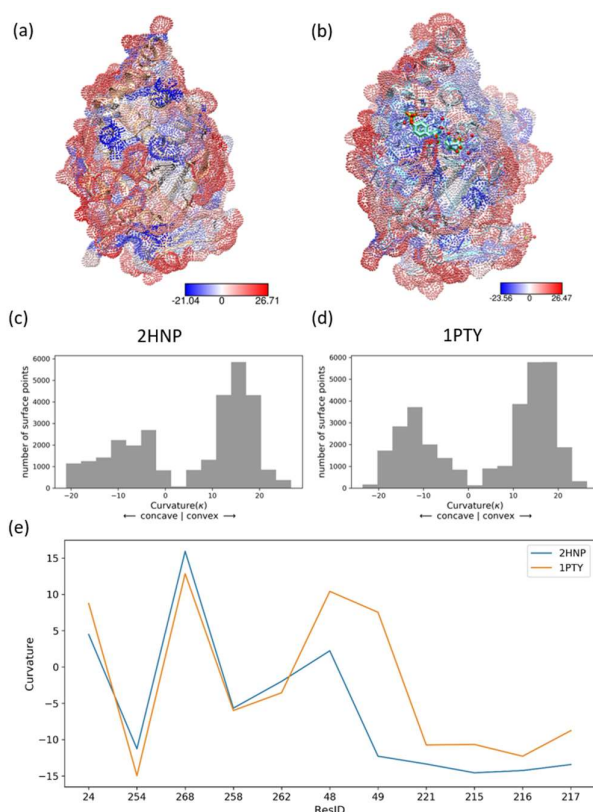


Figure 5: (a) Curvature colored molecular surface of human protein tyrosine phosphatase 1B (PDB ID:2HNP), (b) the same receptor complexed with two phosphotyrosine molecules (PDB ID: 1PTY). We have used BWR as the colouring gradation - blue represents cavities, and red represents protrusions. The colour intensity highlights the curvature of the surface. The near planar surface is represented with white colour. (c) Histogram of curvature values for human protein Tyrosine phosphate 1B in the unbound state – PDB ID 2HNP (d) PDB ID 1PTY shows the bound state (e) Distribution of curvature values for the exposed residues in the unbound (2HNP) and bound state(1PTY)

c. Using Curvature complementarity to quantify shape complementarity between different interacting systems

In most biological processes, proteins interact with other molecules to perform their functions. These interactions include both electrostatic and dispersive nature.³⁹ However, shape complementarity has long been recognized as a significant factor in interactions involving protein aggregation and complex formation with small ligands³⁹⁻⁴¹. Biological complexes typically exhibit intermolecular interfaces of high shape complementarity, and it is one of the

most fundamental ingredients of the scoring functions for protein-protein docking⁴². Consequently, shape complementarity has been used as a prime consideration in docking approaches that consider entire molecular surfaces rather than strictly active site regions⁴³. Recently, Gainza et al.²³ employed the classical “shape index” and “curvedness”², defined in terms of principal curvatures (κ_1 and κ_2) to measure shape complementarity. Like the previous approaches, they use patches of fixed size. Here we use our surface curvature estimation to quantify the degree of shape complementarity between the interacting partners. We take two ‘bound’ complexes and generate their respective molecular surfaces. The surface curvatures of patches in each of the two systems A and B are calculated by our methodology described above. Then, we take the cartesian product of curvature values of each point on the dot molecular surface of the two systems, as shown in Fig. 6. We then define a Gaussian fall-off function as shown below,

$$f_{ij} = \exp\left(-\frac{[r_{ij} - \mu_i]^2}{2\sigma_i^2}\right).$$

Here r_{ij} is the Euclidean distance between points p_i and p_j on A and B, respectively.

μ_i is the average distance between a point p_i on surface A and all points on surface B. We used f_{ij} as the penalty function for a pair of points in the two systems' non-interface region. It is scaled to lie in between 0.0 and 1.0. The points that are close to the interface will have values close to 1, and the far-off points that are not in the interface region will have smaller values. Using the above approach, we can quickly rule out the pair of points that do not lie in the two systems' interface and do not contribute towards the interaction at the interfacial region.

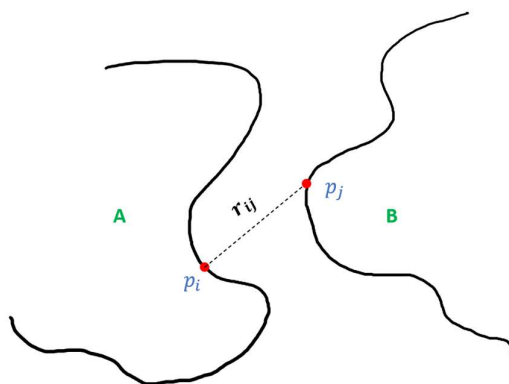


Figure 6: Schematic illustration of shape complementarity measured in terms of surface curvature complementarity. A and B represent the two interacting partners. r_{ij} is the distance between two points - p_i and p_j that lie on the surface of A and B, respectively.

We add their respective curvature values to measure curvature compatibility between any two points on A and B's molecular surfaces. Intuitively, if a point p_i lies on a cavity patch on A then it has negative signed curvature. Similarly, if a point p_j lies in a protrusion patch on B, then it has positive signed curvature. For two points on A and B to be '*shape compatible*', the sum of their respective signed curvature values should be minimum. We take the cartesian product of all points on the surface of A and B and store this metric κ_{ij} in a 2D array. Next, we scale all values with our above defined Gaussian fall-off function f_{ij} as, $s_{ij} = f_{ij} * \kappa_{ij}$, where s_{ij} is the shape complementarity measure and κ_{ij} is the sum of respective curvature values. As mentioned above, if two interacting surfaces have a high degree of curvature complementarity, then the weighted pairwise sum (s_{ij}) of curvature values should be near zero. We calculated s_{ij} for the two types of interacting systems – binary complexation between two proteins and protein-ligand system (Fig. 7) and ternary complexation (antigen-antibody interaction) Fig. 8)

Figure 7a shows the molecular structure of human Fibroblast stromelysin-1(Red) (PDB ID: 1SLN) and its inhibitor (Blue), along with a figure showing the weighted curvature values(Fig 7b). For surfaces that share a high degree of shape complementarity, we expect the distribution of s_{ij} to be concentrated in the left region [Figure 7b]. As mentioned above, for surfaces having compatible surface curvatures, the pairwise sum of weighted curvature values near the interfacial region is close to zero. The greater the degree of surface curvature compatibility, the more is the number density of s_{ij} with close to zero values. To quantify the degree of surface complementarity, we compute the measure of positive skewness. A positively skewed distribution has the most density concentrated in the left, with a long and fat tail on the right side (See section E below). Consequently, if interacting surfaces were rugged and irregular, then the distribution of s_{ij} would shift towards the right.

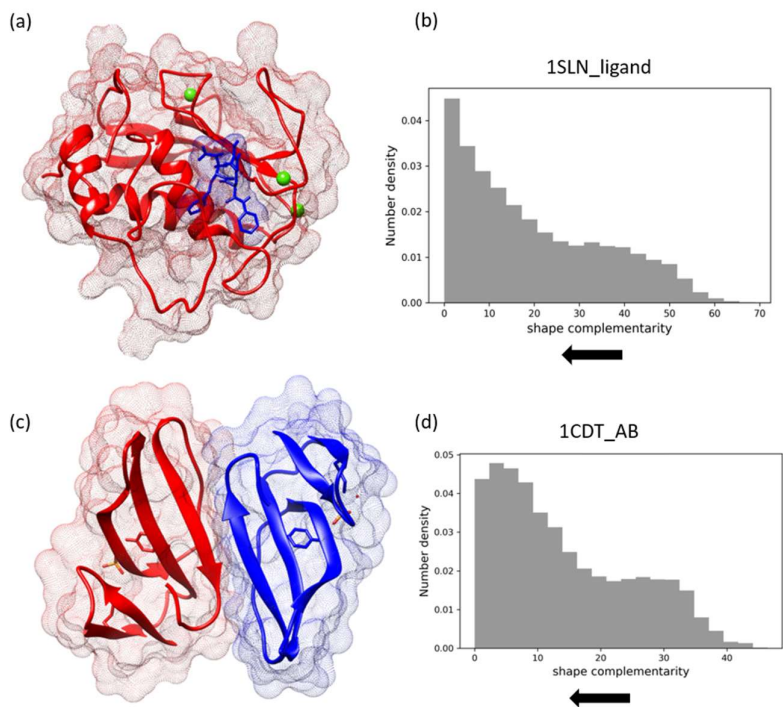


Figure 7: (a) Molecular surface of the catalytic domain of human Fibroblast stromelysin-1 (Red) (PDB ID: 1SLN) and its inhibitor (Blue). (b) Shape complementarity, as measured by our metric. The histogram is normalized. The peak in the left half indicates high shape complementarity at the interfacial region of the protein surface and its inhibitor (c) Molecular surface of homologous protein dimer [PDB ID: 1CDT] with chain A (Red) and chain B (Blue). (d) Shape complementarity, as measured by our metric. The histogram is normalized. The peak in the left half indicates high shape complementarity at the interfacial region of the protein dimers.

Similarly, Fig. 7c shows the molecular structure of a homologous dimer of cardiotoxin VII4 (PDB ID: 1CDT). Interestingly, the interface here is complementary - accordingly, our measurement of curvature, as shown in Fig 8d.

Figure 8 shows the results for a ternary complex of antigen-antibody interaction. Unlike protein-protein and protein-inhibitor interactions, the antigen-antibody interactions have lesser

pronounced shape compatibility⁴⁴. We show below the antibody-antigen system - 1A2Y, the hen egg-white lysozyme (D18A mutant), in complex with mouse monoclonal antibody D1.3. The shape complementarity is observed even in this ternary complex.

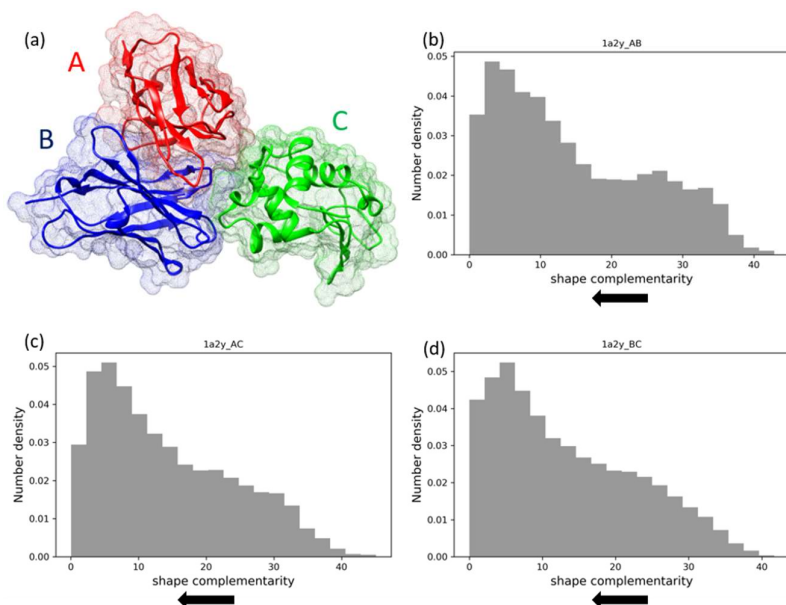


Figure 8: (a) Molecular surface of antibody (A and B)-antigen(C) protein-protein complex [PDB ID: 1A2Y]. Chain A is the light chain, and chain B is the heavy chain. (b, c, and d) Shape complementarity, as measured by our metric. The histogram is normalized. The peak in the left half indicates high shape complementarity at the interfacial region of the complex.

D. Runtime Complexity of our approach

We now briefly discuss the overall computational complexity of our approach: (1) The molecular surface generation by MSMS is $O(N \log N)$. (2) For hierarchical clustering, we use a fast implementation of agglomerative clustering using k-NN graphs⁴⁵. It uses an approximate nearest neighbor graph for reducing the number of distance calculations. It significantly speeds up the naïve implementation from $O(N^3)$ to $O(\tau N \log N)$; here, τ denotes the number of nearest neighbor updates required at each iteration. For 3D dataset, like our point cloud data, τ is found to be less than 8⁴⁵. For k-NN graph creation, we use a k-d tree that has a time complexity of $O(N \log N)$. (3) For fitting, per patch we use truncated SVD in our hyperaccurate

sphere fitting algorithm. Its complexity is $O(\min\{mn^2, m^2n\})$. Here, $n=3$ is fixed for 3D data, and m represents the number of points per patch.

E. *Quantifying the shape complementarity at the interface*

Once we obtain the distribution of s_{ij} , we measure the skewness of the distribution to quantify the degree of shape complementarity. The more the number of curvature-compatible points on the two interacting systems' interface, the more is the degree of right-skewness (positive skewness). A positive-skewed distribution is characterized by the long and fat tail on the right side. The skewness of a random variable X is the third standardized moment:

$$\tilde{\mu} = E \left[\left(\frac{X - \mu}{\sigma} \right)^3 \right],$$

where μ is the mean, and σ is the standard deviation.

Here we consider the interfacial points on the molecular surface of the interacting system. We have chosen the interfacial distance cut-off as 1.5 Å. In Table S1 of Appendix 2, we used skewness to quantify the change in shape complementarity with the different orientations of interacting systems (a protein-protein complex). This highlights the method's ability to distinguish different docking orientations for a given pair of interacting molecules. We illustrate below (Figure 9) two different orientations of a homodimer 1CDT, one at the native state (0 degrees) and one rotated (along the Z-axis along the plane of the paper) by 120 degrees. At the native state, we observe a high value of skewness, and hence high shape complementarity at the Interfacial region. However, rotating chain B (blue) lowers the skewness value. A detailed comparison of skewness at different orientations is given in section B, table S1 of Appendix 2. We also employed our approach to testing the relationship between geometric shape complementarity as measured by our skewness function and the binding constant for two realistic systems (Figure S3, Appendix 2).

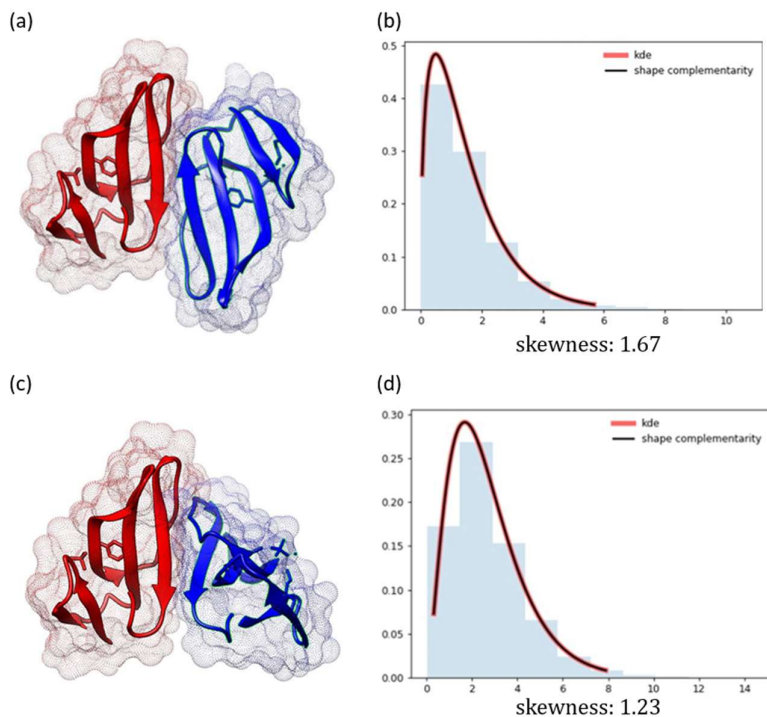


Figure 9: Skewness measured at the (a,b) native conformation (0 degree) and at (c,d) rotated (120 degrees) conformation for the homologous protein dimer [PDB ID: 1CDT] .

Discussion

In summary, we have designed a fast, robust non-iterative algorithm for surface curvature calculation and used it to reinterpret “shape complementarity”. Our approach of employing surface curvature as the measure of shape complementarity provides a more straightforward and more intuitive way to interpret how different surfaces in a binding system interact with each other. Our approach does not have shortcomings of the previous approaches that used patches of a fixed size radius, leading to the non-optimal division of a protein surface. In our approach, however, the size of patches is automatically inferred by hierarchical clustering, and points are clustered together automatically using the farthest neighbour approach. We used the geodesic distance as the distance criteria and quickly implemented hierarchical clustering using k-NN graphs (elaborated in methods section and runtime complexity section). This addresses several issues related to using a fixed-size patch, particularly radial patches using Euclidean distance or an ad-hoc fixed distance cut-off(Figure 1). For protein surface representation, we

used SES generated by the MSMS program²⁴. We designed an effective method for dividing this protein surface into patches using the farthest neighbour hierarchical clustering, an unsupervised machine learning approach. The patches hence obtained in protrusions, cavities, and saddle surfaces of varying shape and sizes, carry vital structural information about the protein. We can vary the granularity of the surface patch by adjusting the threshold to apply when forming flat clusters. This allows measurement of curvature at different “resolutions”. For instance, a larger threshold value ($t > 20$) could highlight macroscopic features such as deep clefts for side-chain recognition. In comparison, a smaller threshold value ($t < 10$) allows the identification of a more nuanced atomic-level feature [Figure 3]. This implementation of varying patch size has a significant advantage over previous approaches. When using patches made by sampling points within a local radius, the true size that would capture desired features like cavities and protrusions accurately is unknown, as such features are present in varying size and shape, and one must rely on their intuition or visual inspection to get the ‘right’ size [Figure 1c].

The surface of each patch is rugged and non-uniform. A pragmatic and straightforward way to measure the surface curvature is fitting a sphere to the surface patch of interest using least square fitting. We used a non-iterative method for fitting sphere to a surface patch, which is faster and more stable than the previous approaches. Further, unlike the solid-angle approach¹¹, our method is sensitive to nuances in surface topology.

Comparison with the current state-of-art approach

Recently, Gainza et al.²³ used a mesh-based representation of solvent excluded surface (SES) generated by the MSMS²⁴ program and used a distant-dependent curvature as one of the features in their geometric deep learning approach. Here, the protein mesh was decomposed into a set of overlapping patches of fixed geodesic radius ($r = 9\text{Å}$ or $r = 12\text{Å}$). Notably, they regularize the mesh after computing the MSMS surface, which is an expensive operation and one of the bottlenecks in their data preparation pipelines.

We want to point out that Gainza used an overlapping set of radial patches. It means that in their approach, protein surface was not partitioned per se – the neighbouring patch of a fixed radial geodesic distance of 12Å was extracted around each point. This was done to implement geometric convolution, where we have a kernel sliding on overlapping patches. Different features, including charges, distant dependent curvature, hydrophobicity, were calculated and then these features were used in their geometric deep learning approach. *Their use of*

overlapping patches makes it difficult to draw out a direct comparison. Moreover, from their current approach, we cannot infer how significant the geometric shape complementarity is for a particular system.

We adopt a more direct approach where we partition the protein surface into a set of non-overlapping patches using hierarchical clustering. Our approach uses an approximate nearest neighbour graph for reducing the number of distance calculations. It significantly speeds up the naïve implementation, which is $O(N^3)$, to $O(\tau N \log N)$; followed by fitting sphere to each patch. The run time complexity is now discussed in the Results section. We would like to point out that our approach is significantly faster than SiteEngine^{46,47}, which Ganzia et al. found to be the closest competing method. It uses both physiochemical and geometric features for searching similarities among molecular surfaces⁴⁷. It is based on explicit alignments of pockets using pseudo-representations of the molecular surface, which results in a much higher runtime. Our approach uses just surface curvature to predict surface complementarity. Therefore, a direct comparison with Ganzia et al. is not feasible at this moment.

The results above show that we can use this method to calculate protein's local curvatures quickly and accurately. Thus, it can be employed to select ligands with complimentary curvature for a known receptor quickly. Moreover, it can be employed to understand curvature variation during the dynamical motions of proteins, which may help open up newer possibilities of interaction with the environment of both solvent and other molecules.

References:

- (1) Holm, L.; Sander, C. Mapping the Protein Universe. *Science* **1996**, *273* (5275), 595–602.
- (2) Koenderink, J. J.; Van Doorn, A. J. Surface Shape and Curvature Scales. *Image Vis. Comput.* **1992**, *10* (8), 557–564.
- (3) Cipriano, G.; Phillips Jr, G. N.; Gleicher, M. Multi-Scale Surface Descriptors. *IEEE Trans. Vis. Comput. Graph.* **2009**, *15* (6), 1201–1208.
- (4) Alimohamadi, H.; Rangamani, P. Modeling Membrane Curvature Generation Due to Membrane–Protein Interactions. *Biomolecules* **2018**, *8* (4), 120.
- (5) Feng, X.; Xia, K.; Tong, Y.; Wei, G. Geometric Modeling of Subcellular Structures, Organelles, and Multiprotein Complexes. *Int. J. Numer. Methods Biomed. Eng.* **2012**, *28* (12), 1198–1223.
- (6) Feng, X.; Xia, K.; Chen, Z.; Tong, Y.; Wei, G. Multiscale Geometric Modeling of Macromolecules II: Lagrangian Representation. *J. Comput. Chem.* **2013**, *34* (24), 2100–2120.
- (7) Xia, K.; Feng, X.; Chen, Z.; Tong, Y.; Wei, G.-W. Multiscale Geometric Modeling of Macromolecules I: Cartesian Representation. *J. Comput. Phys.* **2014**, *257*, 912–936.

- (8) Dzubiella, J.; Swanson, J. M.; McCammon, J. Coupling Hydrophobicity, Dispersion, and Electrostatics in Continuum Solvent Models. *Phys. Rev. Lett.* **2006**, *96* (8), 087802.
- (9) Sharp, K. A.; Nicholls, A.; Fine, R. F.; Honig, B. Reconciling the Magnitude of the Microscopic and Macroscopic Hydrophobic Effects. *Science* **1991**, *252* (5002), 106–109.
- (10) Chan, H. S.; Dill, K. A. Solvation: Effects of Molecular Size and Shape. *J. Chem. Phys.* **1994**, *101* (8), 7007–7026.
- (11) Southall, N. T.; Dill, K. A. The Mechanism of Hydrophobic Solvation Depends on Solute Radius. *J. Phys. Chem. B* **2000**, *104* (6), 1326–1331.
- (12) Kobe, B.; Kajava, A. V. When Protein Folding Is Simplified to Protein Coiling: The Continuum of Solenoid Protein Structures. *Trends Biochem. Sci.* **2000**, *25* (10), 509–515.
- (13) Mamatkulov, S. I.; Khabibullaev, P. K.; Netz, R. R. Water at Hydrophobic Substrates: Curvature, Pressure, and Temperature Effects. *Langmuir* **2004**, *20* (11), 4756–4763.
- (14) M. L. Connolly. Measurement of Protein Surface Shape by Solid Angles. *J Mol Graph Model* **1986**, *4* (1), 3–6.
- (15) Connolly, M. L. Analytical Molecular Surface Calculation. *J. Appl. Crystallogr.* **1983**, *16* (5), 548–558.
- (16) Coleman, R. G.; Burr, M. A.; Souvaine, D. L.; Cheng, A. C. An Intuitive Approach to Measuring Protein Surface Curvature. *Proteins Struct. Funct. Bioinforma.* **2005**, *61* (4), 1068–1074. <https://doi.org/10.1002/prot.20680>.
- (17) Wei, G.-W. Differential Geometry Based Multiscale Models. *Bull. Math. Biol.* **2010**, *72* (6), 1562–1622.
- (18) Connolly, M. L. The Molecular Surface Package. *J. Mol. Graph.* **1993**, *11* (2), 139–141.
- (19) Duncan, B. S.; Olson, A. J. Shape Analysis of Molecular Surfaces. *Biopolymers* **1993**, *33* (2), 231–238. <https://doi.org/10.1002/bip.360330205>.
- (20) Tsodikov, O. V.; Record Mt Jr Fau - Sergeev, Y. V.; Sergeev, Y. V. Novel Computer Program for Fast Exact Calculation of Accessible and Molecular Surface Areas and Average Surface Curvature. No. 0192-8651 (Print).
- (21) Bates, P. W.; Wei, G.-W.; Zhao, S. Minimal Molecular Surfaces and Their Applications. *J. Comput. Chem.* **2008**, *29* (3), 380–391.
- (22) Albou, L.; Schwarz, B.; Poch, O.; Wurtz, J. M.; Moras, D. Defining and Characterizing Protein Surface Using Alpha Shapes. *Proteins Struct. Funct. Bioinforma.* **2009**, *76* (1), 1–12.
- (23) Gainza, P.; Sverrisson, F.; Monti, F.; Rodola, E.; Boscaini, D.; Bronstein, M.; Correia, B. Deciphering Interaction Fingerprints from Protein Molecular Surfaces Using Geometric Deep Learning. *Nat. Methods* **2020**, *17* (2), 184–192.
- (24) Sanner, M. F.; Olson, A. J.; Spehner, J. Reduced Surface: An Efficient Way to Compute Molecular Surfaces. *Biopolymers* **1996**, *38* (3), 305–320.
- (25) Axenopoulos, A.; Daras, P.; Papadopoulos, G.; Houstis, E. A Shape Descriptor for Fast Complementarity Matching in Molecular Docking. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2011**, *8* (6), 1441–1457.
- (26) Yin, S.; Proctor, E. A.; Lugovskoy, A. A.; Dokholyan, N. V. Fast Screening of Protein Surfaces Using Geometric Invariant Fingerprints. *Proc. Natl. Acad. Sci.* **2009**, *106* (39), 16622–16626.
- (27) Murtagh, F.; Contreras, P. Algorithms for Hierarchical Clustering: An Overview. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2012**, *2* (1), 86–97.
- (28) Al-Sharadqah, A. and C. Error Analysis for Circle Fitting Algorithms. *Electron J Stat.* **2009**, *Volume 3*, 886–911. <https://doi.org/doi:10.1214/09-EJS419>.

- (29) Richards, F. M. Areas, Volumes, Packing, and Protein Structure. *Annu. Rev. Biophys. Bioeng.* **1977**, *6* (1), 151–176.
- (30) Bystroff, C. MASKER: Improved Solvent-Excluded Molecular Surface Area Estimations Using Boolean Masks. *Protein Eng. Des. Sel.* **2002**, *15* (12), 959–965. <https://doi.org/10.1093/protein/15.12.959>.
- (31) Bayley, M. J.; Gardiner, E. J.; Willett, P.; Artymiuk, P. J. A Fourier Fingerprint-Based Method for Protein Surface Representation. *J. Chem. Inf. Model.* **2005**, *45* (3), 696–707.
- (32) Saxena, A.; Prasad, M.; Gupta, A.; Bharill, N.; Patel, O. P.; Tiwari, A.; Er, M. J.; Ding, W.; Lin, C.-T. A Review of Clustering Techniques and Developments. *Neurocomputing* **2017**, *267*, 664–681.
- (33) Dawyndt, P.; De Meyer, H.; De Baets, B. The Complete Linkage Clustering Algorithm Revisited. *Soft Comput.* **2005**, *9* (5), 385–392.
- (34) Francetič, M.; Nagode, M.; Nastav, B. Hierarchical Clustering with Concave Data Sets.
- (35) Chernov, N.; Sapirstein, P. Fitting Circles to Data with Correlated Noise. *Comput. Stat. Data Anal.* **2008**, *52* (12), 5328–5337.
- (36) Boyd, S.; Boyd, S. P.; Vandenberghe, L. *Convex Optimization*; Cambridge university press, 2004.
- (37) Puius, Y. A.; Zhao, Y.; Sullivan, M.; Lawrence, D. S.; Almo, S. C.; Zhang, Z. Y. Identification of a Second Aryl Phosphate-Binding Site in Protein-Tyrosine Phosphatase 1B: A Paradigm for Inhibitor Design. *Proc Natl Acad Sci U S A* **1997**, *94* (25), 13420–13425.
- (38) Sukhwal, A.; Bhattacharyya, M.; Vishveshwara, S. Network Approach for Capturing Ligand-Induced Subtle Global Changes in Protein Structures. *Acta Crystallogr Biol Crystallogr* **2011**, *67* (Pt 5), 429–439. <https://doi.org/10.1107/s0907444911007062>.
- (39) Li, Y.; Zhang, X.; Cao, D. The Role of Shape Complementarity in the Protein-Protein Interactions. *Sci. Rep.* **2013**, *3*, 3271. <https://doi.org/10.1038/srep03271>.
- (40) Jones, S.; Thornton, J. M. Principles of Protein-Protein Interactions. *Proc. Natl. Acad. Sci.* **1996**, *93* (1), 13–20.
- (41) Katchalski-Katzir, E.; Shariv, I.; Eisenstein, M.; Friesem, A. A.; Aflalo, C.; Vakser, I. A. Molecular Surface Recognition: Determination of Geometric Fit between Proteins and Their Ligands by Correlation Techniques. *Proc. Natl. Acad. Sci.* **1992**, *89* (6), 2195–2199.
- (42) Chen, R.; Weng, Z. A Novel Shape Complementarity Scoring Function for Protein-Protein Docking. *Proteins* **2003**, *51* (3), 397–408. <https://doi.org/10.1002/prot.10334>.
- (43) Norel Raquel; Petrey Donald; Wolfson Haim J; Nussinov Ruth. Examination of Shape Complementarity in Docking of Unbound Proteins. *Proteins Struct. Funct. Bioinforma.* **1999**, *36* (3), 307–317. [https://doi.org/doi:10.1002/\(SICI\)1097-0134\(19990815\)36:3<307::AID-PROT5>3.0.CO;2-R](https://doi.org/doi:10.1002/(SICI)1097-0134(19990815)36:3<307::AID-PROT5>3.0.CO;2-R).
- (44) Lawrence, M. C.; Colman, P. M. Shape Complementarity at Protein/Protein Interfaces. Elsevier 1993.
- (45) Franti, P.; Virmajoki, O.; Hautamaki, V. Fast Agglomerative Clustering Using a K-Nearest Neighbor Graph. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28* (11), 1875–1881.
- (46) Shulman-Peleg, A.; Nussinov, R.; Wolfson, H. J. SiteEngines: Recognition and Comparison of Binding Sites and Protein-Protein Interfaces. *Nucleic Acids Res.* **2005**, *33* (suppl_2), W337–W341.
- (47) Shulman-Peleg, A.; Nussinov, R.; Wolfson, H. J. Recognition of Functional Sites in Protein Structures. *J. Mol. Biol.* **2004**, *339* (3), 607–633.

Chapter 5: Prediction of good reaction coordinates and future evolution of MD trajectories using Regularized Sparse Autoencoders – A novel deep learning approach

Introduction:

We widely use reaction coordinates throughout chemical physics to model and understand complex chemical transformations. Often simple chemical reactions can be described in terms of one-dimensional reaction-coordinate, which differs from the Cartesian coordinates, and is a generalized coordinate of the system $q = q(r_1, r_2, \dots, r_N)$, a function of cartesian coordinate. For describing a complex dynamical process, it is often necessary to use a set of reaction coordinates. The set of such reaction coordinates themselves comprise a combination of simple reaction coordinates. When generalized coordinates describe a reaction profile, they are typically referred to as reaction coordinates, collective variables (CVs), or order parameters, depending on the context and type of system. Reaction coordinates play a pivotal functional role in understanding the dynamics of a chemical reaction. A good set of reaction coordinates is required to estimate kinetically significant energy barriers or elucidating reaction mechanisms.¹ The natural reaction coordinate is the most informative about the system's future evolution among all different one-dimensional measurements of the state of some high-dimensional dynamical system. While reaction coordinates or collective variables are potentially helpful and intuitively appealing, we must be careful while using them. For example, molecular dynamics (MD) simulation allows us to study molecular processes, but the sampling problem constrains its usefulness. A solution to this long-standing problem is enhanced sampling approaches. However, when applied to poorly chosen reaction coordinates, they can bias the system in misleading ways and generate erroneous predictions of free energy barriers, transition states, and mechanisms. Furthermore, reactions in condensed phase systems occur in a very high dimensional space that includes many uninvolved solutes, solvent coordinates that are not intrinsic to identifying reaction coordinates. Thus, it often leads to several difficulties in deciphering correct reaction coordinates, which renders the use of "physical intuition", or ad-hoc methods routinely employed infeasible and inaccurate.

Even though the idea of reaction coordinate is so widely used in chemical kinetics, the community has not reached a consensus regarding its precise definition.²⁻⁵ In our approach, we

wish to define the natural reaction coordinate to not depend on a particular "reaction" or "product" conformations or subsets of phase space.⁶ A natural reaction coordinate should be a function that maps any point in the phase space to a single real number $q: \Omega \rightarrow \mathbb{R}$, where q is the reaction coordinate, and Ω denotes the phase space. The reaction coordinate of this form includes geometric or physical observable properties. Other definitions, mainly the path-based ones such as MEP or MAP, do not take this form. Instead, they define a path through phase space, a mapping from \mathbb{R} to Ω . These paths map an arc length to phase space coordinate. The reaction coordinate's value is undefined for all conformations in Ω that are not on the path.

In our approach, we jointly predict the optimal set of physically interpretable reaction coordinates and the future evolution of the dynamical system. We model the MD trajectories, which are input in our machine learning(ML) model as a collection of multivariate time series(MTS).

Also, the coordinate should be the slowest one so that all the other degrees of freedom can easily equilibrate along the reaction coordinate⁷. Previous work involved using Principal component analysis(PCA), a technique used for dimensionality reduction, to approximate reaction coordinates. The problem with PCA is that it does not consider the time aspect involved in MD trajectory data. It chooses the direction of maximum variance, which is usually not what we are looking for when searching for slow coordinates⁸. Another factor that limits the applicability of PCA is that different low dimensional representations constructed by PCA are not comparable with each other. We might choose different sets of internal coordinates, like contact distances, bond dihedrals, and each yields a different solution. Sultan et al. used time-structured based independent component analysis(tiCA) for identifying RCs. tiCA aims to find projections of the MD data that minimise the loss of kinetic information. Unlike PCA, tiCA does not assume that high variance modes are associated with slow degrees of freedom. It does so by maximising the autocorrelation function. However, tiCA is a linear model, and this limits its ability. Kernel trick can be used to extend tiCA and yield non-linear solutions. However, it is computationally expensive and is dependent on tuning and choice of kernel⁹. There have been several deep learning-based approaches for choosing or discovering an optimal set of reaction coordinates in recent times. VAMPnets employ the variational approach for Markov processes (VAMP) to develop a deep learning framework for molecular kinetics using neural networks. It encodes the entire mapping from molecular coordinates to Markov

states, thus combining the whole data processing pipeline in a single end-to-end framework¹⁰. Wehmeyer et al. used a variant of autoencoder, namely time-lagged autoencoder, to find low dimensional embeddings for the high dimensional molecular dynamics data¹¹. They highlighted the importance of using an appropriate set of collective variables(CVs) in Markov state modelling(MSM) and employed their approach on different analytical systems and alanine dipeptide systems. The Variation approach for conformation dynamics(VAC) forms the basis of many methods that are currently used for identifying slow CVs¹². It searches for d orthogonal directions r_i , such that the projection $r_i^T z_t$ is maximal. The eigenvalues of the propagator bound these autocorrelations from above. Nuske et al. emphasized that the eigenvalues and eigenvectors of the MD propagator, also called the transfer operator, contain the key information about thermodynamics and kinetics. They presented a variational approach for the calculation of the dominant eigenvectors and eigenvalues of the propagator. In the Markovian model-based approach, there is an implicit assumption that the future evolution of the system, $x_{t+\tau}$ depends only on the present state x_t , where t is the time step and τ is the lag time. There are many physical processes, both deterministic and stochastic, which are Markovian.

Dynamic mode decomposition(DMD) is another approach for finding RCs. It tries to minimise the regression error: $\sum_t \|z_{t+\tau} - \mathbf{K}^T z_t\|^2$, where \mathbf{K} is a linear model, and compute its d eigenvectors r_i with largest eigenvalues^{13,14}. All these models use a linear model of the form:

$$\mathbb{E}[g(x_{t+\tau})] = \mathbf{K}^T \mathbb{E}[f(x_t)]$$

The $f(\cdot)$ and $g(\cdot)$ are feature transformations that act on x and transform it into the feature space where dynamics are approximately linear. The expectation value over the time accounts for the stochasticity. For DMD, the feature transformation is an identity transformation \mathbb{I} .

Dimension reduction can be facilitated when working feature space instead of directly using the cartesian coordinates¹⁵.

Methods

We denote a matrix of multivariate time series by \mathbf{X} and its component column vectors by \mathbf{x} . For a vector \mathbf{x} , its i -th element is denoted by x_i . For a matrix \mathbf{X} , we use x_i as the i -th column

and $x_{i,j}$ is the (i,j) -th entry of \mathbf{X} . In our model, we denote a collection of high dimensional multivariate time series by $(\mathbf{x}_{T+1}, \dots, \mathbf{x}_{T+k})$, where each \mathbf{x}_i is a vector of dimension n (features) at time point i . Here, T , denotes an arbitrary time point. We consider the problem of forecasting l time future values, given the information(history) about $(k - l)$ time steps. The \bar{y} denotes the reconstructed future output (windowed trajectory). We now outline below the architecture of a simple autoencoder and highlight the difference between it and our modified sparse autoencoder that uses additional regularization terms.

A simple autoencoder is trained to reconstruct the input fed to it. It consists of an encoder and a decoder function [Figure 1].

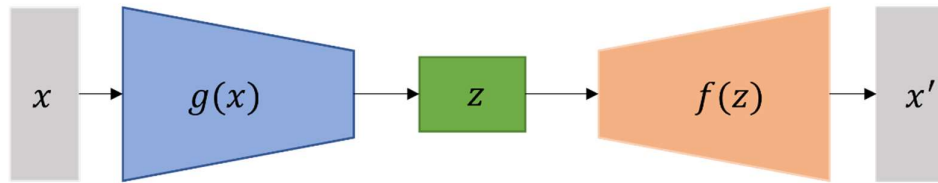


Figure 1: A schematic representation of a simple autoencoder. The code or the latent vector z is not regularized, and the auto-encoder can be made over-complete or under-complete by tuning the dimension of z .

The encoder function $g(\cdot)$ takes the input x and learns the mapping $x \rightarrow z$, where z denotes the latent space representation of x . The z is also called the latent vector since it consists of latent or "hidden" values that are not observed in the data. The decoder function $f(\cdot)$ learns the mapping $z \rightarrow x$ and outputs x' , which is called the reconstructed input. The loss function for such an architecture is the reconstruction error measured as the mean squared error(MSE) between the original input x and the reconstructed input x' .

Our regularized sparse autoencoder architecture accepts as an input a multivariate time series x_t and instead of simply reconstructing the input, it predicts the $x_{t+\tau}$, i.e. the evolution of the trajectory x_t after lag time τ . As mentioned earlier, we denote the prediction by \bar{y} .

The loss function for our model is the sum of three terms. \mathbf{C} measures the reconstruction error between the output and the model prediction. \mathbf{R} is the sparsity regularisation term for the latent variable \mathbf{z} . Here, we impose the L_1 regularizer on the latent variable. L_2 is the ridge penalty $\lambda \sum_{j=1}^d \mathbf{z}_j^2$ and L_1 is the lasso penalty $\lambda \sum_{j=1}^d |\mathbf{z}_j|$ that we have used in our regularization function \mathbf{R} on the latent vector for the loss function.

The motivation behind using L_1 regularization is that in a high dimensional space, many of the weight parameters will equal zero simultaneously. Intuitively, it helps in choosing those latent variables which contribute significantly towards the prediction of the evolution of an MD trajectory. This scenario is quite unlike the L_2 regularization, which does not impose a sparsity constraint, i.e., it encourages the weight values towards zero (but not exactly zero). D is the error associated with Encoder prediction of latent variable \bar{z} and z :

$$Loss = C(y, Dec(z, h)) + D(z, Enc(y, h)) + \lambda R(z)$$

The addition of an additional sparsity regularizer forces the autoencoder to cut down the number of active neurons in the coding layer. This results in representation generated by combination of a small number of active neurons. An alternative strategy is to actually measure the sparsity of the coding layer and penalize the model when this exceeds the target value of sparsity. If we want to measure the divergence between the target threshold(probability) p that a neuron in the coding layer will activate and the actual probability q , we can measure the KL divergence.

$$D_{KL}(p || q) = p \log \frac{p}{q} + (1 - p) \log \frac{1 - p}{1 - q}$$

In the current approach, though, we have chosen L_1 regularization.

By varying the latent variable in the latent space, the output generated varies over the manifold of possible predictions. This provides the model with the ability to make multimodal prediction. The model finds the optimal z that minimizes the reconstruction error. The regularization constraints on z limit its information capacity and forces the model to learn non-trivial latent space representation of the inputs. In an energy-based modelling terminology, this limits the volume of space that has low energy¹⁶.

To update the parameters of the model, we first predict z that minimizes

$$C(y, Dec(z, h)) + \lambda R(z),$$

and then use this z as a feedback signal (target) that predicts \bar{z} from x and y (Figure 2, greyed box) by feeding h and y into the *Encoder* with $D(z, \bar{z})$ as the loss. This strategy helps in learning optimal z , and we do not have to make the latent variable inference again.

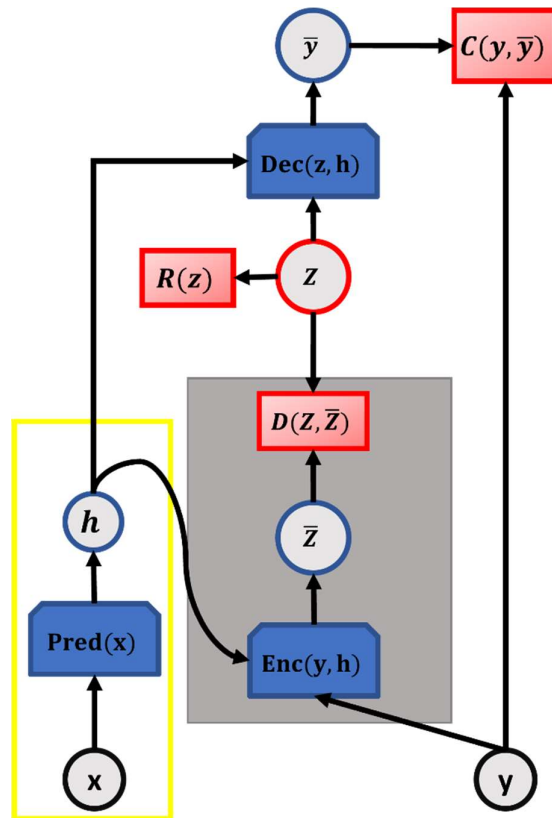


Figure 2: A schematic representation of our architecture. The greyed box contains the Encoder that predicts \bar{z}

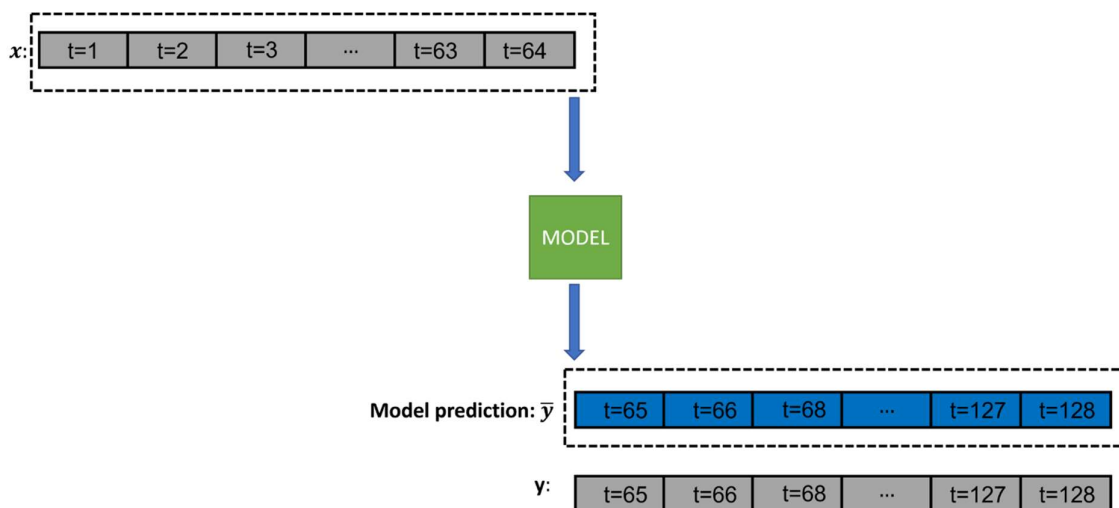


Figure 3: A schematic representation of data flow – our multi-step prediction model predicts \bar{y} vector of length l .

We train our model end-to-end to predict optimal sparse latent variable z (Reaction coordinate(s)), along with future values $y := x_{T+k-l+1:T+k} = \{x_{T+k-l+1}, \dots, x_{T+k}\}$ based on the past $k - l$ steps $\{x_{T+1}, x_{T+2}, \dots, x_{T+k-l}\}$. T denotes the starting point in the data (Figure 3)

We applied our model to the two systems –

- 1) Study of different metastable states of alanine dipeptide
- 2) Intercalation of Proflavine into DNA minor groove in an aqueous environment¹⁷

For the alanine dipeptide system, we retrieved data from two sources:

1. We used MDSHARE¹⁸ to obtain MD simulation data, consisting of 250ns trajectories spanning all 6 metastable states. The details of the trajectories are given below in Table 1.

Table 1: MD trajectory details

Property	Value
Code	ACEMD
Forcefield	AMBER ff-99SB-ILDN
Integrator	Langevin

Integrator time step	2 fs
Simulation time	250 ns
Frame spacing	1 ps
Temperature	300 K
Volume	(2.3222 nm) ³ periodic box
Solvation	651 TIP3P waters
Electrostatics	PME
PME real-space cutoff	0.9 nm
PME grid spacing	0.1 nm
PME updates	every two-time step
Constraints	all bonds between hydrogens and heavy atoms

2. **shoot-302K-100ps**: This dataset contains 5000 x 100 ps shooting trajectories out of each of 6 manually-identified states. Hamiltonian trajectories (velocity Verlet without thermostat) were initiated from a canonical (NVT) distribution at 302 K from within each state¹⁹.

We used PyEMMA¹⁸ and extracted backbone torsions, backbone atom positions, and backbone atom distances for the featurization of the data. This results in three feature matrices of dimensions (T, 11), (T, 4), and (T, 18), respectively for a single trajectory. The time series data is represented as a tensor, with batch dimension as the first axis (Figure 4)

We adopt the following data windowing strategy:

Our model involves multi-output and multi-time step prediction. Figure 3 shows a schematic representation of single output, multi-output prediction. We used a window size of 128. The output size is also 128 in length.

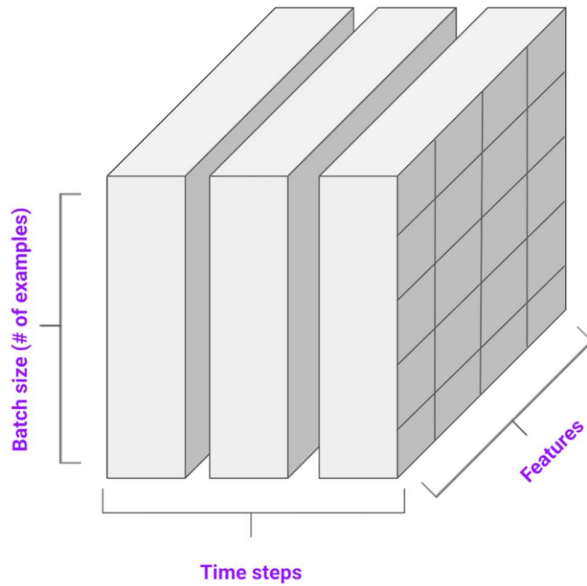


Figure 4: representing multivariate time series data – a tensor of shape : (Number of examples, time steps, features). For a single sample, the batch axis can be ignored.

We have used stacked 1D convolutional layers, doubling the dilation rate at every layer. The receptive field doubles at every layer. This architecture is similar to wavenet.²⁰ The lower layers in the encoder learn short-term patterns, and the higher layers learn long term patterns. Doubling of dilation rate at each layer gives the network the ability to handle very long sequences.

The decoder architecture is symmetric with the encoder and uses dilated deconvolutions and is defined by transposed operations. The loss used for \mathbf{D} and \mathbf{Z} is MSE.

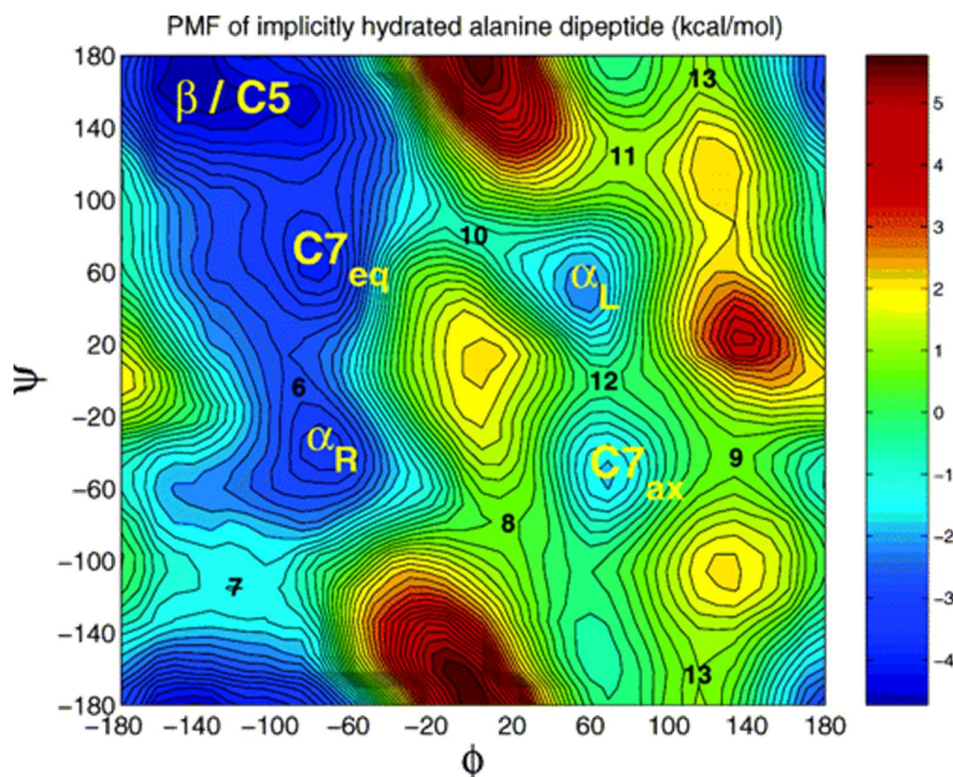


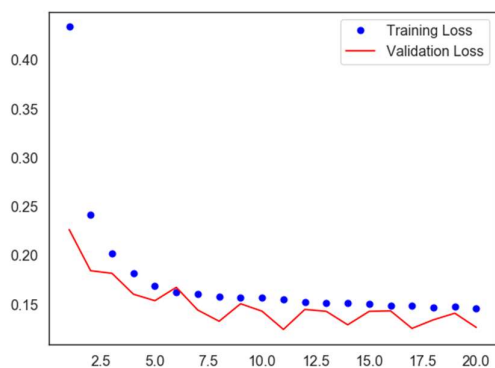
Figure 5: Contour map of the interpolated PMF of the implicitly hydrated alanine dipeptide. Adapted with permission from Long-Time Conformational Transitions of Alanine Dipeptide in Aqueous Solution: Continuous and Discrete-State Kinetic Models; Dmitriy S. Chekmarev, Tateki Ishida, and Ronald M. Levy; *The Journal of Physical Chemistry B* **2004** 108 (50), 19487-19495; DOI: 10.1021/jp048540w. Copyright 2004 American Chemical Society

The different metastable states of alanine dipeptide are shown in Figure 5.

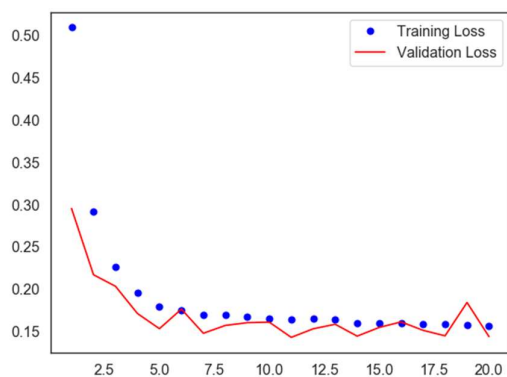
The main conformers of the hydrated alanine dipeptide molecule can be arranged in the following order according to the effective free energy difference ΔW with respect to the lowest energy structure $\beta/C5:\beta/C5$ (taken as zero energy) $< C7_{eq}$ ($\Delta W \approx 0.9$ kcal/mol) $< \alpha_R$ ($\Delta W \approx 1.5$ kcal/mol) $< \alpha_L$ ($\Delta W \approx 2.7$ kcal/mol) $< C7_{ax}$ ($\Delta W \approx 3.2$ kcal/mol)²¹.

In Figure 6, we show the results obtained for the prediction of the future evolution of trajectory for the three features backbone atomic positions, torsions, and distances between atoms. The training and validation loss decreases with each epoch.

(a)



(b)



(c)

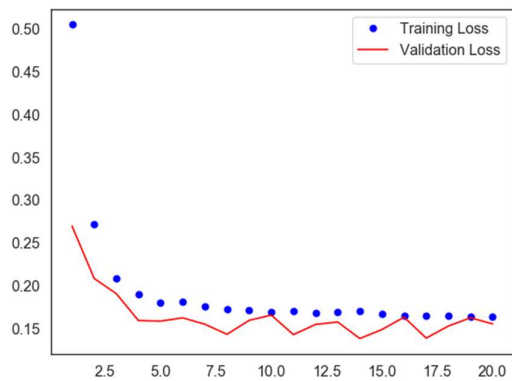


Figure 6: (a) Training and validation loss for prediction of the average of backbone atom positions (b) Training and validation loss for prediction of the average of torsions (c) Training and validation loss for prediction of the average of distances over all atoms

For alanine dipeptide, the torsion angles ϕ and ψ aptly describe its dynamics as it transitions into different metastable states. We used the latent variable representation from our model and compared it with the actual values of the torsions. We observed close agreement between the two (Figure 7).

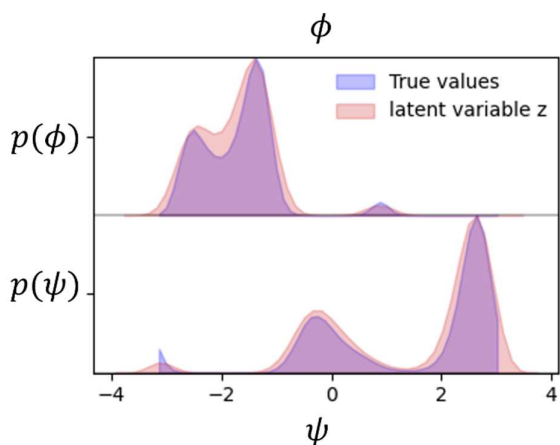


Figure 7: The model predicts the optimal set of latent variables ϕ and ψ as the true reaction coordinates. We have projected the latent variable z found by the model onto the known values of ϕ and ψ . The model's prediction matches closely with the experimentally known reaction coordinates.

For the second system of DNA-proflavine, which describes the phenomenon of recrossing behaviour of MD trajectories near the transition state, the proflavine drug intercalates into the minor groove of DNA in an aqueous environment¹⁷. Recrossing occurs due to the coupling of environment degrees of freedom with the RCs. In this model, we wanted to understand what the key reaction coordinates besides X and ϕ are near the transition state region for understanding recrossing behaviour. Previously, in our group, we studied the recrossing behaviour by using X (separation) and ϕ as the reaction coordinates²². While X defines the position of the drug with respect to the intercalation base pairs, the collective variable ϕ denotes the position of the drug along the helical axis of the DNA. The RR trajectory(described below) are the trajectories that show recrossing behaviour. The different features in DNA are listed in figure 8. The other features are the number of hydrogen bonds between water molecules and proflavine(hbnum1), the number of water around the drug(wat_0.34_flv_heavy) and around the intercalating base-pair IBP (wat_0.34_ibp_heavy), each within 0.34 nm distance, and the separation coordinate¹⁷.

The system details are as follows:

- Tensor of shape (Samples, Timesteps, Features)
1. Reactant – Reactant trajectories (RR) shape: (1050, 6001, 19)
 2. Reactant – Product trajectories (RP) shape: (810, 6001, 19)

For training and prediction, we choose the window size of 128 steps. The training phase involves using the windowed trajectory and predicting whether the trajectory will show recrossing behaviour. Figure 9 shows the training and validation accuracy of the model.

Figure 10 shows the SHAP summary plot²³ of each feature and its weightage in its contribution towards the RCs set. The features were inferred by doing canonical correlation analysis(CCA) of encoded time series and univariate time series of each feature. We observed that base pair rise and role parameters contribute the most.

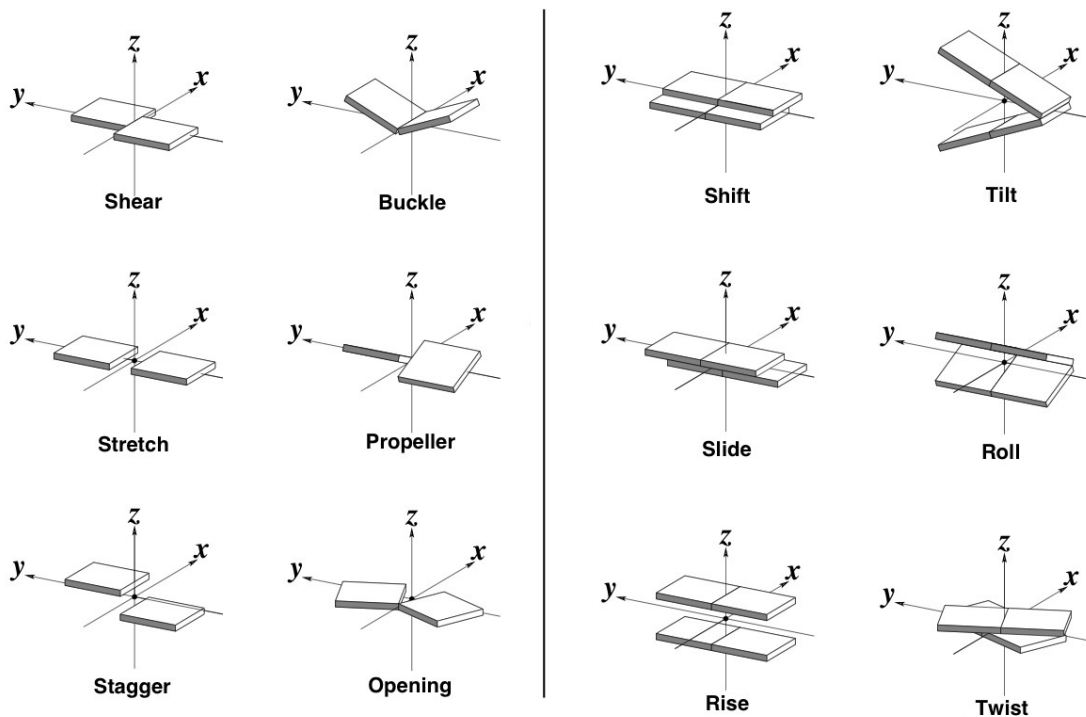


Figure 8: The different translational base pair parameters: Rise, Shift and Slide. The rotational base pair parameters: Twist and Roll. Buckle is a rotational base step parameter. The figure is adapted from article ²⁴ [Lu XJ, Olson WK. 3DNA: a software package for the analysis, rebuilding and visualization of three-dimensional nucleic acid structures. *Nucleic Acids Res.* 2003 Sep 1;31(17):5108-21. doi: 10.1093/nar/gkg680. PMID: 12930962; PMCID: PMC212791.]

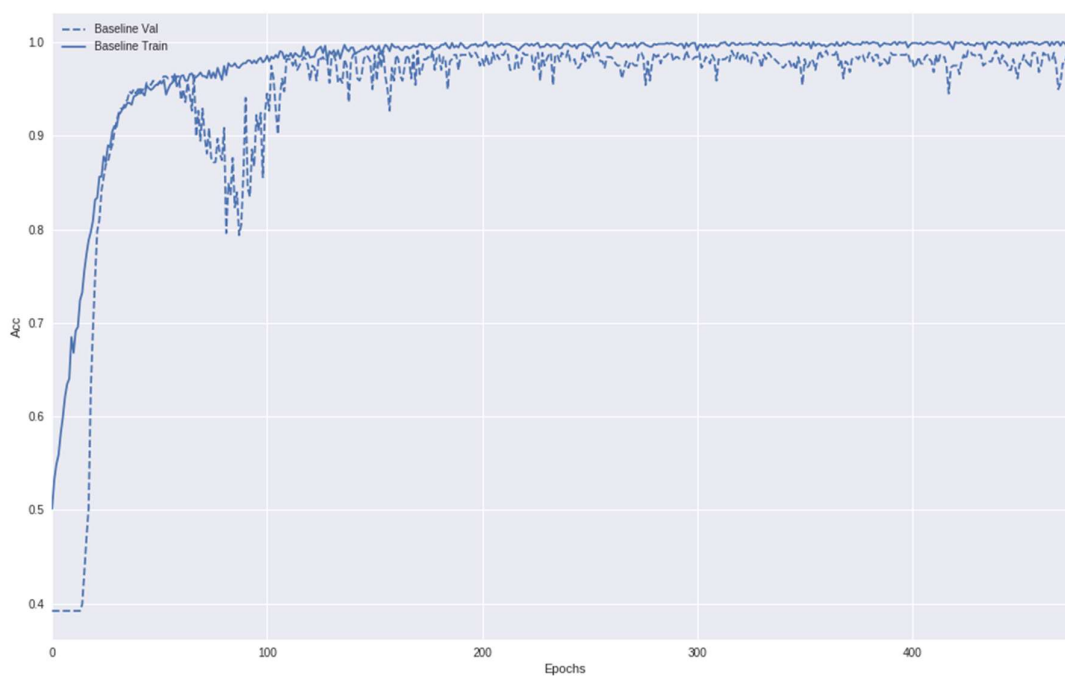


Figure 9: Training and validation accuracy for the DNA-proflavine system

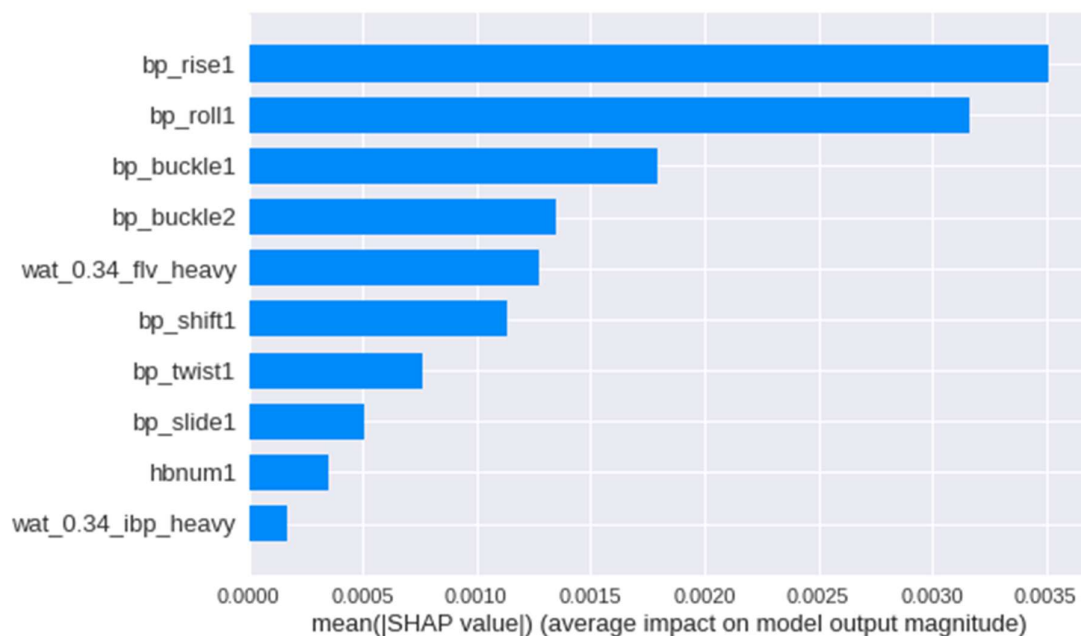


Figure 10: Feature vector importance near the transition state region

In the present study, we intended to study what other factors intrinsic to DNA – rise, roll, shift, buckle, twist, and the number of hydrogen-bonded pairs of water in the vicinity contribute to optimal CV. This was one of the unanswered but crucial questions in the previous study¹⁷.

Discussion and conclusion

Comparison with contemporary approaches

Our method, illustrated in Figure 2, differs from the “time-lagged autoencoder” by Noe et al. in several ways. Firstly, our approach takes inspiration from “Energy-based models” introduced by Yann LeCun et al. and his colleagues¹⁶ and its application in fast sparse coding using autoencoders²⁵. The energy-based model framework was quite revolutionary. For a given input, sparse coding minimizes a quadratic reconstruction error with an L_1 penalty term on the code.

The sparse encoding approach aims to find the **fixed point** of our parametric estimate of the optimal latent vector (reaction coordinate/CVs) $z(t + \tau) = \Theta(\text{Enc}(y, h)_t)$, where Θ is some function that is learned as we train the model by feeding in the widowed trajectories from the

input. $Enc(y, h)_t$ is the encoder output at time t . The idea of finding the fixed point comes from the fact that CVs usually represent the slowest relaxing degrees of freedom⁷, so as the model gets better at each epoch (loss declines, figure 6 and figure 9), the CVs that are learnt get better.

Notably, Noe. et.al. optimize a different objective function (Equation 2)¹¹. Although they have used the same letter \mathbf{z} , the \mathbf{z} in their paper is the input trajectory, whereas in our notation z represents the latent vector. They have tried to minimize the regression error of reconstruction without sparsity constraints.

In the present study of DNA-drug recrossing for the second system, we intended to study what other factors intrinsic to DNA – rise, roll, shift, buckle, twist, and the number of hydrogen-bonded pairs of water in the vicinity contribute to optimal CV for the recrossing phenomenon. The free energy surface for the DNA-proflavin system is represented in the cited paper from our group¹⁷ in collaboration with Hynes – “Dynamical Recrossing in the Intercalation Process of the Anticancer Agent Proflavine into DNA”. The reaction coordinate for this system was established in²⁶. Previously, in our group, we studied the recrossing behaviour by using X (separation) and ϕ as the reaction coordinates²². In the present study of DNA-drug recrossing, we intended to study what other factors intrinsic to DNA – rise, roll, shift, buckle, twist, and the number of hydrogen-bonded pairs of water in the vicinity contribute to optimal CV for the recrossing phenomenon. While X defines the position of the drug with respect to the intercalation base pairs, the collective variable ϕ denotes the position of the drug along the helical axis of the DNA. This was one of the unanswered but crucial questions in the previous study. For the first specimen in our study (alanine-dipeptide), we already showed the derived CVs and compared it with well-established CVs for that system.

The sparsity constrain in our model limits the model to learn only crucial features ranked by their importance in SHAP graph (Figure 10).

Our strategy of using the latent vector representations can be used to assess and infer the reaction coordinates. As a side effect, the system is also able to predict the future evolution of MD trajectories, which is analogous to time series prediction.

References

- (1) McGibbon, R. T.; Pande, V. S. *J Chem Phys* **2015**, *142*, 124105.

- (2) Fukui, K. Formulation of the Reaction Coordinate. *J. Phys. Chem.* **1970**, *74* (23), 4161–4163.
- (3) Tachibana, A.; Fukui, K. *Theor Chim Acta* **1980**, *57*, 81.
- (4) Quapp, W.; Heidrich, D. *Theor Chim Acta* **1984**, *66*, 245.
- (5) Yamashita, K.; Yamabe, T.; Fukui, K. *Chem Phys Lett* **1981**, *84*, 123.
- (6) McGibbon, R. T.; Husic, B. E.; Pande, V. S. Identification of Simple Reaction Coordinates from Complex Dynamics. *J. Chem. Phys.* **2017**, *146* (4), 044109. <https://doi.org/10.1063/1.4974306>.
- (7) M. Sultan, M.; Pande, V. S. TICA-Metadynamics: Accelerating Metadynamics by Using Kinetically Selected Collective Variables. *J. Chem. Theory Comput.* **2017**, *13* (6), 2440–2447.
- (8) Li, W.; Ma, A. Recent Developments in Methods for Identifying Reaction Coordinates. *Mol. Simul.* **2014**, *40* (10–11), 784–793.
- (9) Schwantes, C. R.; Pande, V. S. Modeling Molecular Kinetics with TICA and the Kernel Trick. *J. Chem. Theory Comput.* **2015**, *11* (2), 600–608.
- (10) Mardt, A.; Pasquali, L.; Wu, H.; Noé, F. VAMPnets for Deep Learning of Molecular Kinetics. *Nat. Commun.* **2018**, *9* (1), 5. <https://doi.org/10.1038/s41467-017-02388-1>.
- (11) Wehmeyer, C.; Noé, F. Time-Lagged Autoencoders: Deep Learning of Slow Collective Variables for Molecular Kinetics. *J. Chem. Phys.* **2018**, *148* (24), 241703. <https://doi.org/10.1063/1.5011399>.
- (12) Nuske, F.; Keller, B. G.; Pérez-Hernández, G.; Mey, A. S.; Noé, F. Variational Approach to Molecular Kinetics. *J. Chem. Theory Comput.* **2014**, *10* (4), 1739–1752.
- (13) Mezić, I. Spectral Properties of Dynamical Systems, Model Reduction and Decompositions. *Nonlinear Dyn.* **2005**, *41* (1), 309–325.
- (14) Tu, J. H. Dynamic Mode Decomposition: Theory and Applications. **2013**.
- (15) Harrigan, M. P.; Sultan, M. M.; Hernández, C. X.; Husic, B. E.; Eastman, P.; Schwantes, C. R.; Beauchamp, K. A.; McGibbon, R. T.; Pande, V. S. MSMBuilder: Statistical Models for Biomolecular Dynamics. *Biophys. J.* **2017**, *112* (1), 10–15.
- (16) LeCun, Y.; Chopra, S.; Hadsell, R.; Ranzato, M.; Huang, F. A Tutorial on Energy-Based Learning. *Predict. Struct. Data* **2006**, *1* (0).
- (17) Hynes, J. T.; MUKHERJEE, A.; HRIDYA, V. Dynamical Recrossing in the Intercalation Process of the Anticancer Agent Proflavine into DNA. **2019**.
- (18) Scherer, M. K.; Trendelkamp-Schroer, B.; Paul, F.; Pérez-Hernández, G.; Hoffmann, M.; Plattner, N.; Wehmeyer, C.; Prinz, J.-H.; Noé, F. PyEMMA 2: A Software Package for Estimation, Validation, and Analysis of Markov Models. *J. Chem. Theory Comput.* **2015**, *11* (11), 5525–5542.
- (19) Chodera, J. D.; Swope, W. C.; Pitera, J. W.; Dill, K. A. Long-Time Protein Folding Dynamics from Short-Time Molecular Dynamics Simulations. *Multiscale Model. Simul.* **2006**, *5* (4), 1214–1226.
- (20) Rethage, D.; Pons, J.; Serra, X. A Wavenet for Speech Denoising; IEEE, 2018; pp 5069–5073.
- (21) Chekmarev, D. S.; Ishida, T.; Levy, R. M. Long-Time Conformational Transitions of Alanine Dipeptide in Aqueous Solution: Continuous and Discrete-State Kinetic Models. *J. Phys. Chem. B* **2004**, *108* (50), 19487–19495. <https://doi.org/10.1021/jp048540w>.
- (22) Sasikala, W. D.; Mukherjee, A. Molecular Mechanism of Direct Proflavine–DNA Intercalation: Evidence for Drug-Induced Minimum Base-Stacking Penalty Pathway. *J. Phys. Chem. B* **2012**, *116* (40), 12208–12212. <https://doi.org/10.1021/jp307911r>.
- (23) Lundberg, S. M.; Lee, S.-I. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30*; Guyon, I., Luxburg, U. V.,

- Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc., 2017; pp 4765–4774.
- (24) Lu, X.; Olson, W. K. 3DNA: A Software Package for the Analysis, Rebuilding and Visualization of Three-dimensional Nucleic Acid Structures. *Nucleic Acids Res.* **2003**, *31* (17), 5108–5121.
- (25) Gregor, K.; LeCun, Y. Learning Fast Approximations of Sparse Coding; 2010; pp 399–406.
- (26) Sasikala, W. D.; Mukherjee, A. Molecular Mechanism of Direct Proflavine–DNA Intercalation: Evidence for Drug-Induced Minimum Base-Stacking Penalty Pathway. *J Phys Chem B* **2012**, *116*, 12208.

Chapter 6: Learning to learn – “What makes a molecule a prospective drug?”

Introduction

In the world of drug discovery, the task-specific labels are scarce – there are only ~15,000 drugs, out of which ~4200 are approved ones. At the same time, the chemical space is combinatorically large. Owing to the vast size of chemical space, which is estimated to be in the order of 10^{60} molecules, the task of successfully finding new drugs is daunting and predominantly the major hindrance in drug development. With the rapid proliferation and advancement of AI, the technologies empowered by it have become invaluable tools in the various stages of the drug development process, such as identification and validation of drug targets, designing of new drugs, drug repurposing, improving the R&D efficiency, aggregating, and analysing biomedicine information and refining the decision-making process to recruit patients for clinical trials. It is expected that such a holistic AI approach will address the inefficiencies and uncertainties that arise in the classical drug development methods while minimising bias and human intervention in the process. The other uses of AI in drug development include the prediction of feasible synthetic routes for drug-like molecules¹, pharmacological properties², protein characteristics as well as efficacy³, drug combination and drug–target association⁴ and drug repurposing⁵. Deep learning has demonstrated outstanding success in proposing potent drug candidates and accurately predicting their properties and the possible toxicity risks⁶. Circumventing past problems in drug development – such as analysis of large datasets, laborious screening of compounds while minimising standard error, requiring large amounts of R&D cost and time of over US\$2.5 billion and more than a decade – are now possible using AI methods. With AI technology, new studies can be carried out in assisting the identification of new drug targets, rational drug designing and drug repurposing^{7,8}. Additionally, ML techniques and predictive model software also contribute to the identification of target-specific virtual molecules and the association of the molecules with their respective target while optimising the safety and efficacy attributes.

In this chapter, we leverage the power of self-supervised learning (SSL) to learn good representations of molecules. SSL has profoundly impacted Natural Language Processing(NLP), allowing the language models to be trained on large unlabelled text datasets and then use these models for downstream tasks⁹. After pre-training, transfer learning is used to repurpose the model for a different but related task. Pre-training involves training a model on related tasks with abundant data and then fine-tuning it on a downstream task of interest. Transfer Learning is a technique where we use a pre-trained model to solve a problem similar to the problem the model was initially trained to solve.

SSL leverages the underlying structure in the data and obtains the supervisory signals from the data itself. The learning approach involves predicting the hidden(masked) input part from any unhidden part of the output. To apply this approach, we represent molecules as a graph. The graph data represents rich information, mainly the relation-based information, among the graph entities. These entities are called nodes or vertices, and edges connect different nodes. In the world of molecules, a node represents an atom, and a node is connected to other nodes(atoms) through edges(bonds). Intuitively, we would like to build neural networks that, on the input, takes a graph and, on the output, makes predictions. These predictions can be at the different levels - nodes, pairs of nodes, at the subgraph(community) level, or at the graph-level - prediction of a property of a given molecule that can be represented as a graph on the input. Each of these molecules/atoms has different features, such as the associated charge, bond type and other relevant information.

Graph Neural Networks (GNNs) provide an effective solution to representation learning on graph data. Their operating principle involves a neighbourhood aggregation scheme. We iteratively update the representation vector of a given node by aggregating and transforming representation vectors of its neighbours at each stage. Previously, GNNs have been used to extract molecular fingerprints, which encode the structure of molecules. These fingerprints offer better predictive performance on downstream tasks, better interpretability, and reduced downstream computation time¹⁰.

In traditional ML approaches, much effort goes into designing useful features, and devising proper ways to capture data structure so machine learning models can take advantage of it. In representation learning approach that we have incorporated here; this feature engineering step

is not required. Once we have the graph data, we can learn “good representation” of the graph to be used for the downstream machine learning algorithm. Representation learning is all about automatically extracting or learning features in the chemical graph. SSL has also been used as a pre-training strategy for Graph Neural Networks(GNNs)¹¹.

Motivation and background for using GNNs – The widely used multi-layer perceptron (MLPs) are very flexible function approximators. Even an MLP with just a single hidden layer can approximate any possible function, assuming that layer is wide enough. However, the MLP doesn’t scale well with the input dimensionality. For instance, for representing a megapixel image, the number of parameters in the model quickly explodes. Consequently, the model overfits. Convolutional neural networks can address this issue for structured signals that live on a grid 1D – time grid or 2D grid such as an image. However, the problem with CNN is that they work for such regular grid structured data like above. Most data cannot be described in such a regular format, for instance, molecules, which have a graph structure that cannot be easily brought into a regular grid structure format. We seek a model class that scales better than MLPs and is more flexible than a convolutional neural network. The idea is to generalize CNN to be more flexible and is scalable. This provides us with the motivation for using neural nets for general graph-structured inputs – Graph Neural Networks.

We want to exploit the local structure of the graph. The local structure is the local connectivity in the graph is the prior information that we want to exploit to build the model that generalizes well. Graphs are descriptors of the signal structure where the signals are stored at the nodes, and the edges express the similarity between the signal components.

In the 2D convolutional grid – the image grid also expresses closeness. However, the grid does not need to be regular in the general graph formulation, and the edges can even have different weights. The convolutions we define on the graph are polynomials conditioned on the graph structure encoded in a matrix derived from the graph. Intuitively, we are applying a filter; as we apply a convolutional filter on the 2d grid structure, we are applying a convolutional filter on a graph. The size of the filter on a graph structure depends on how far a target node is from its k-hop neighbours. The neighbourhood size depends on the value of k; the larger value of k, the larger the neighbourhood. $K = 1$ represents the immediate neighbours of a node.

The graph neural network paradigm allows us to model various tasks ranging from NLP, where we have parsed trees, which are essentially graphs, to modelling everyday scenes where we model the compositional structure of objects.

There have been several use cases for using graph neural networks in drug discovery and drug interactions. For instance, drug interaction was modelled by representing drugs and proteins as nodes and the drug-protein and protein-protein interactions as edges. In literature, the known side effects of drugs, when taken together, is sparse. A good use case is designing models to predict the edges(links) between drugs. This methodology was used to discover new side effects that were not known earlier in the FDI database. At the graph-level machine learning tasks, one of the impactful applications is drug discovery. Recently, Stokes et al. used a graph-based deep learning approach for discovering new antibiotics. The GNN was used to classify different molecules and predict promising molecules from a large pool of candidates, followed by experimental validation. A sub-task of drug discovery involves generating novel molecules with therapeutic activity.

We map nodes in a graph to d -dimensional embeddings such that similar nodes in the graph are embedded close together in this embedding space. The model learns the function $f: u \rightarrow R^d$.

Methods

Notation: We denote graph G defined by vertices(nodes) V , edges E , adjacency matrix A . The graph features include node features h_i for a node i , edge features e_{ij} for an edge connecting node i and node j , and graph features g . The graph features specification varies depending on the application.

Representation:

- Node Features $\mathbf{H} = \{h_1, h_2, \dots, h_N\}; h_i \in \mathbb{R}^F$
- Edge features $e_{ij} \in \mathbb{R}^{F'}$, $\mathbf{E} = \{e_1, e_2, \dots, e_{N_e}\}$, where N_e is the total number of edges
- Adjacency matrix: $\mathbf{A} \in \mathbb{R}^{N \times N}$
- Neighbourhood of a node $\mathcal{N}_i = \{j \mid i = j \text{ or } A_{ij} \neq 0\}$

The general paradigm used for training graph neural networks is message passing, which is briefly discussed below:

There are two key phases involved in the forward pass, that is, the calculation of output values from the input during training – the message passing phase and the readout phase. Message

passing phase is run for T steps, and we define it using message functions M_t and vertex(node) update functions U_t . We update the node features at each node based on the messages:

$$m_i^{t+1} = \sum_{w \in \mathcal{N}_i} M_t(h_i^t, h_j^t, e_{ij})$$

$$h_i^{t+1} = U_t(h_i^t, m_i^{t+1})$$

Here, $w \in \mathcal{N}_i$ denotes the nodes in neighbours of node i . During the readout phase, we compute a feature vector for G using a readout function R

$$\hat{y} = R(\{h_i^T \mid i \in V\})$$

M_t , U_t , and R are differentiable and are learned during the training phase. We note that R is permutation invariant with respect to node states. This is an important constraint; permutation invariance helps us in exploiting the molecule symmetry. Note that we could also learn edge features by using an equation similar to the one for node features update. At each stage, the features for the nodes are updated iteratively. The receptive field at each stage of iteration is expanded and the information flows across different nodes when we are updating a given node. This results in learning a richer representation of the entire molecule. Finally, we could use \hat{y} as the entire graph representation.

The aggregation function we have used is cardinality preserving attention mechanism¹². The presence of cardinality information improves on the previous vanilla attention-based mechanism.

$$e_{ij}^l = \text{Att}(h_i^l, h_j^l),$$

$$\alpha_{ij}^l = \frac{\exp(e_{ij}^l)}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik}^l)},$$

$$h_i^{l+1} = f^{l+1} \left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^l h_j^l + w^{l+1} \odot \sum_{j \in \mathcal{N}_i} h_j^l \right),$$

The value of w can be set to 1 and in that case the $h_i^{l+1} = f^{l+1}(\sum_{j \in \mathcal{N}_i} (\alpha_{ij}^l + 1) h_j^l)$

Att is the attention coefficient usually calculated as $LeakyRELU(a^{(l)T} \cdot concat(z_i^{(l)}, z_j^{(l)}))$, where a is a learnable weight vector and z_i and z_j are linear transformation of $h_i^{(l)}$ and $h_j^{(l)}$ using $W^{(l)}$ as a learnable weight matrix. f is non-linear function (σ).

In the SSL framework (Figure 1), we have used a data augmentation module that we call **T**. It generates different views of molecules using attribute masking, where node/edge attributes are randomly masked^{11,13,14}. Based on the neighbouring structure, the model learns to predict these masked attributes. For masking, we have used masked token for the atom(node) attribute that is masked. We have used NT-Xent loss¹⁴, and extension of InfoNCE loss as the contrastive loss in our approach¹⁵. The loss function L is given below

$$L_{i,j} = \log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{\{k \neq i\}} \exp(\text{sim}(z_i, z_k)/\tau)}$$

The z_i and z_j denote the positive pair (Figure 1) generated by the MLP projection head, τ is the temperature parameter, and sim represents the cosine similarity. We note that in SimCLR¹⁴ the authors note that several different data augmentations techniques can be composed together to yield better results. We have chosen to use only attribute masking as it gave the best results for downstream tasks when used with attention-based approach mentioned above.

We use the following attributes of atoms and bonds to encode molecular graph:

Attributes name	Description
Atomic type	H, C, O, N, F (encoded as one-hot vector)
Chirality	R or S or NULL (encoded as one-hot vector)
Acceptor	Checks whether an atom is an electron acceptor (binary attribute)
Donor	Checks whether an atom is an electron donor (binary attribute)
Atomic number	Atomic number of the atom
Aromatic	Checks whether an atom is a member of an aromatic ring (binary attribute)

Hybridization	sp, sp^2, sp^3 or NULL (encoded as one-hot vector)
Ring size	If an atom belongs to aromatic rings, this tells us the number of rings that include this atom (Integer)
Hydrogens	Number of hydrogens attached to this atom (Integer)

Bond features:

Bond type	It tells if a bond is single, double, triple or an aromatic type (one-hot vector)
Same ring edge	It tells if the atoms on this edge are on the same ring (binary or NULL)

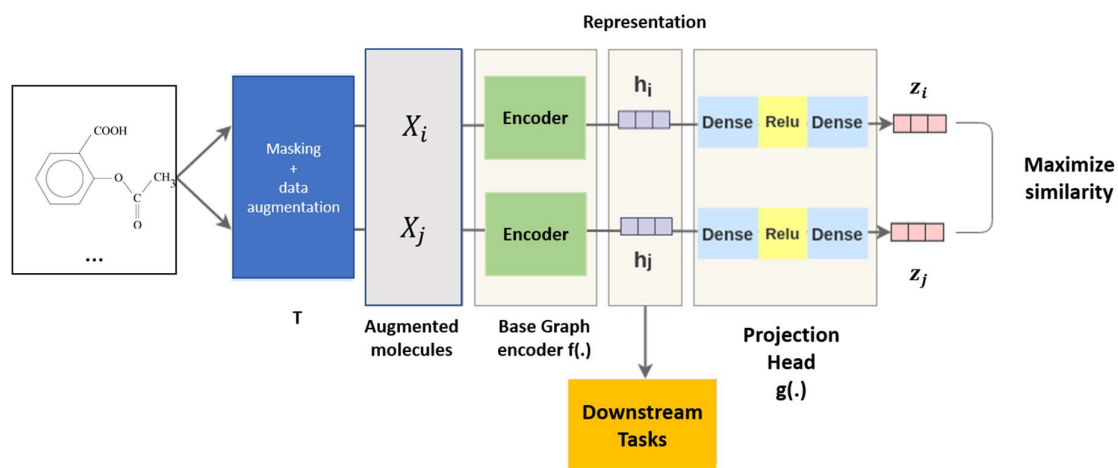


Figure 1: Schematic representation of our model architecture

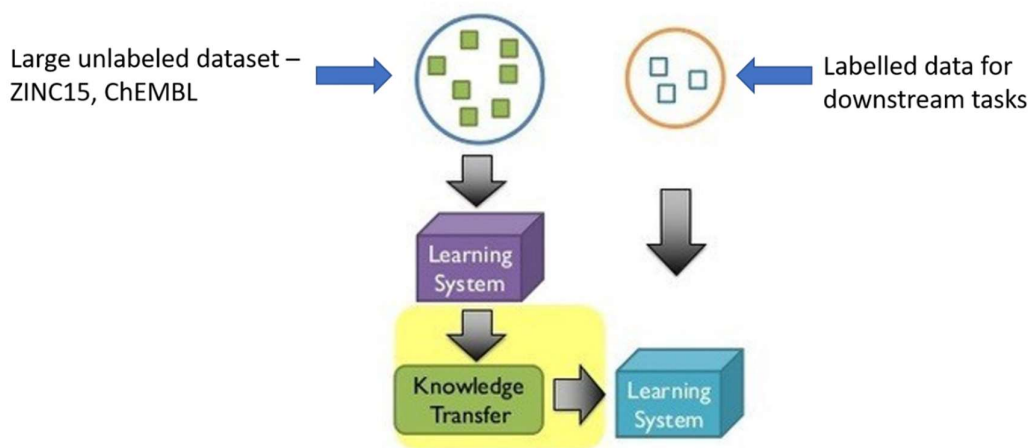


Figure 2: Schematic representation of Transfer learning approach for downstream tasks.

Dataset details:

For pre-training stage, we used QM9¹⁶, ZINC15¹⁷, ChEMBL¹⁸ datasets. The QM9 has ~134K molecules and was used first for training, followed by using ChEMBL and ZINC. From the ZINC15 database, we used a sample of 2 million compounds, and from ChEMBL we used a curated sample¹⁹ of ~456K compounds.

For the downstream task of molecular property prediction, we used CHEMBL_Caco-2, CHEMBL_hMC, CHEMBL_mMC datasets that we curated from ChEMBL database¹⁸. Public data sets for metabolic clearance and passive permeability in Caco-2 cells were extracted from ChEMBLv23. Raw data were obtained by keyword search in the assay description field. The resulting assay list was manually refined. Passive permeability was collected from apparent permeability (P_{app}) values. Clearance data was standardized in units of $\text{mL} \cdot \text{min}^{-1} \cdot \text{g}^{-1}$ and split by species. For each species, the data set was merged using canonical SMILES; the standard deviation was used to keep data following $\text{stddev}(\text{CL}) < 20 \text{ mL} \cdot \text{min}^{-1} \cdot \text{g}^{-1}$. The hERG dataset was obtained from DDH²⁰.

Training details:

For downstream tasks of molecule property prediction, we add a 2-layer MLP with ReLU as the activation function. For the classification task on hERG dataset, the final layer was replaced with the sigmoid layer.

Results:

Table 1: R^2 score based on five fold cross validation compared with the previous approaches

Dataset	R^2 (5-fold CV score)	R^2 (previous SOT)
CHEMBL_Caco-2 ²¹	0.875 ± 0.04	0.77
CHEMBL_hMC ²²	0.765 ± 0.03	0.624
CHEMBL_rMC ²²	0.812 ± 0.02	0.722
CHEMBL_mMC ²²	0.714 ± 0.04	0.575

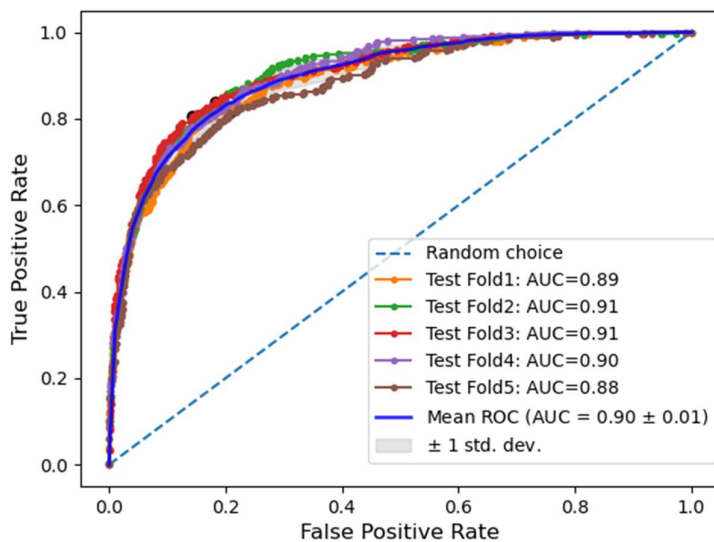


Figure 3: ROC-AUC for 5 fold cross-validation on hERG inhibitory activity dataset (1556 non-blocker, 7551 blocker compounds)

Table 2: results obtained on hERG inhibitory activity dataset

Metric	Result
Accuracy	0.89 \pm 0.03
MCC	0.72 \pm 0.03

The Matthews correlation coefficient(MCC) considers true and false positives and negatives and is generally regarded as a balanced measure that can be used when there is a class imbalance.²³ It produces a more informative and truthful score in evaluating binary classifications than accuracy and F1 score.

Summary and conclusion

The representations learnt by the model are transferable to the downstream tasks where the data size is limited. We tried a powerful cardinality based attention mechanism architecture that captures the molecule structure encoded in graph effectively. We used just one data augmentation strategy of attribute masking in this work, but we expect that other augmentation strategies like context prediction, deletion might be beneficial for improving the model further. The SSL approach is a powerful paradigm, especially under the limited data constraint. Further investigations could be helpful in learning better representations that work well for shared downstream tasks.

References

- (1) Merk, D.; Friedrich, L.; Grisoni, F.; Schneider, G. De Novo Design of Bioactive Small Molecules by Artificial Intelligence. *Mol Inform* **2018**, *37* (1–2). <https://doi.org/10.1002/minf.201700153>.
- (2) Klopman, G.; Chakravarti, S. K.; Zhu, H.; Ivanov, J. M.; Saiakhov, R. D. ESP: A Method To Predict Toxicity and Pharmacological Properties of Chemicals Using Multiple MCASE Databases. *J. Chem. Inf. Comput. Sci.* **2004**, *44* (2), 704–715. <https://doi.org/10.1021/ci030298n>.
- (3) Menden, M. P.; Iorio, F.; Garnett, M.; McDermott, U.; Benes, C. H.; Ballester, P. J.; Saez-Rodriguez, J. Machine Learning Prediction of Cancer Cell Sensitivity to Drugs Based on Genomic and Chemical Properties. *PLOS ONE* **2013**, *8* (4), e61318. <https://doi.org/10.1371/journal.pone.0061318>.

- (4) Nascimento, A. C. A.; Prudêncio, R. B. C.; Costa, I. G. A Multiple Kernel Learning Algorithm for Drug-Target Interaction Prediction. *BMC Bioinformatics* **2016**, *17* (1), 46. <https://doi.org/10.1186/s12859-016-0890-3>.
- (5) Schneider, G. Automating Drug Discovery. *Nature Reviews Drug Discovery* **2018**, *17* (2), 97–113. <https://doi.org/10.1038/nrd.2017.232>.
- (6) Hughes, J. P.; Rees, S.; Kalindjian, S. B.; Philpott, K. L. Principles of Early Drug Discovery. *Br. J. Pharmacol.* **2011**, *162* (6), 1239–1249. <https://doi.org/10.1111/j.1476-5381.2010.01127.x>.
- (7) Katsila, T.; Spyroulias, G. A.; Patrinos, G. P.; Matsoukas, M.-T. Computational Approaches in Target Identification and Drug Discovery. *Comput Struct Biotechnol J* **2016**, *14*, 177–184. <https://doi.org/10.1016/j.csbj.2016.04.004>.
- (8) Drug Target Prediction and Repositioning Using an Integrated Network-Based Approach <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0060618> (accessed 2019 -08 -24).
- (9) Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805* **2018**.
- (10) Duvenaud, D. K.; Maclaurin, D.; Iparraguirre, J.; Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. Convolutional Networks on Graphs for Learning Molecular Fingerprints. In *Advances in Neural Information Processing Systems*; Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R., Eds.; Curran Associates, Inc., 2015; Vol. 28.
- (11) Hu, W.; Liu, B.; Gomes, J.; Zitnik, M.; Liang, P.; Pande, V.; Leskovec, J. Strategies for Pre-Training Graph Neural Networks. *arXiv preprint arXiv:1905.12265* **2019**.
- (12) Zhang, S.; Xie, L. Improving Attention Mechanism in Graph Neural Networks via Cardinality Preservation; NIH Public Access, 2020; Vol. 2020, p 1395.
- (13) Jin, W.; Derr, T.; Liu, H.; Wang, Y.; Wang, S.; Liu, Z.; Tang, J. Self-Supervised Learning on Graphs: Deep Insights and New Direction. *arXiv preprint arXiv:2006.10141* **2020**.
- (14) Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A Simple Framework for Contrastive Learning of Visual Representations; PMLR, 2020; pp 1597–1607.
- (15) Oord, A. van den; Li, Y.; Vinyals, O. Representation Learning with Contrastive Predictive Coding. *arXiv preprint arXiv:1807.03748* **2018**.
- (16) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; Von Lilienfeld, O. A. Quantum Chemistry Structures and Properties of 134 Kilo Molecules. *Scientific data* **2014**, *1* (1), 1–7.
- (17) Sterling, T.; Irwin, J. J. ZINC 15 - Ligand Discovery for Everyone. *J. Chem. Inf. Model.* **2015**, *55*, 2324.
- (18) Gaulton, A.; Bellis, L. J.; Bento, A. P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B.; Overington, J. P. ChEMBL: A Large-Scale Bioactivity Database for Drug Discovery. *Nucleic Acids Res.* **2012**, *40*, D1100.
- (19) Mayr, A.; Klambauer, G.; Unterthiner, T.; Steijaert, M.; Wegner, J. K.; Ceulemans, H.; Clevert, D.-A.; Hochreiter, S. Large-Scale Comparison of Machine Learning Methods for Drug Target Prediction on ChEMBL. *Chemical science* **2018**, *9* (24), 5441–5451.
- (20) DDH. Drug Discovery Hackathon, https://static.mygov.in/rest/s3fs-public/mygov_159353119651307401.pdf.
- (21) Wang, Y.; Chen, X. QSPR Model for Caco-2 Cell Permeability Prediction Using a Combination of HQPSO and Dual-RBF Neural Network. *RSC Advances* **2020**, *10* (70), 42938–42952.

- (22) Wenzel, J.; Matter, H.; Schmidt, F. Predictive Multitask Deep Neural Network Models for ADME-Tox Properties: Learning from Large Data Sets. *Journal of chemical information and modeling* **2019**, *59* (3), 1253–1268.
- (23) Baldi, P.; Brunak, S.; Chauvin, Y.; Andersen, C. A. F.; Nielsen, H. Assessing the Accuracy of Prediction Algorithms for Classification: An Overview. *Bioinformatics* **2000**, *16* (5), 412–424. <https://doi.org/10.1093/bioinformatics/16.5.412>.

Appendix 1

Section A

Experimental conditions

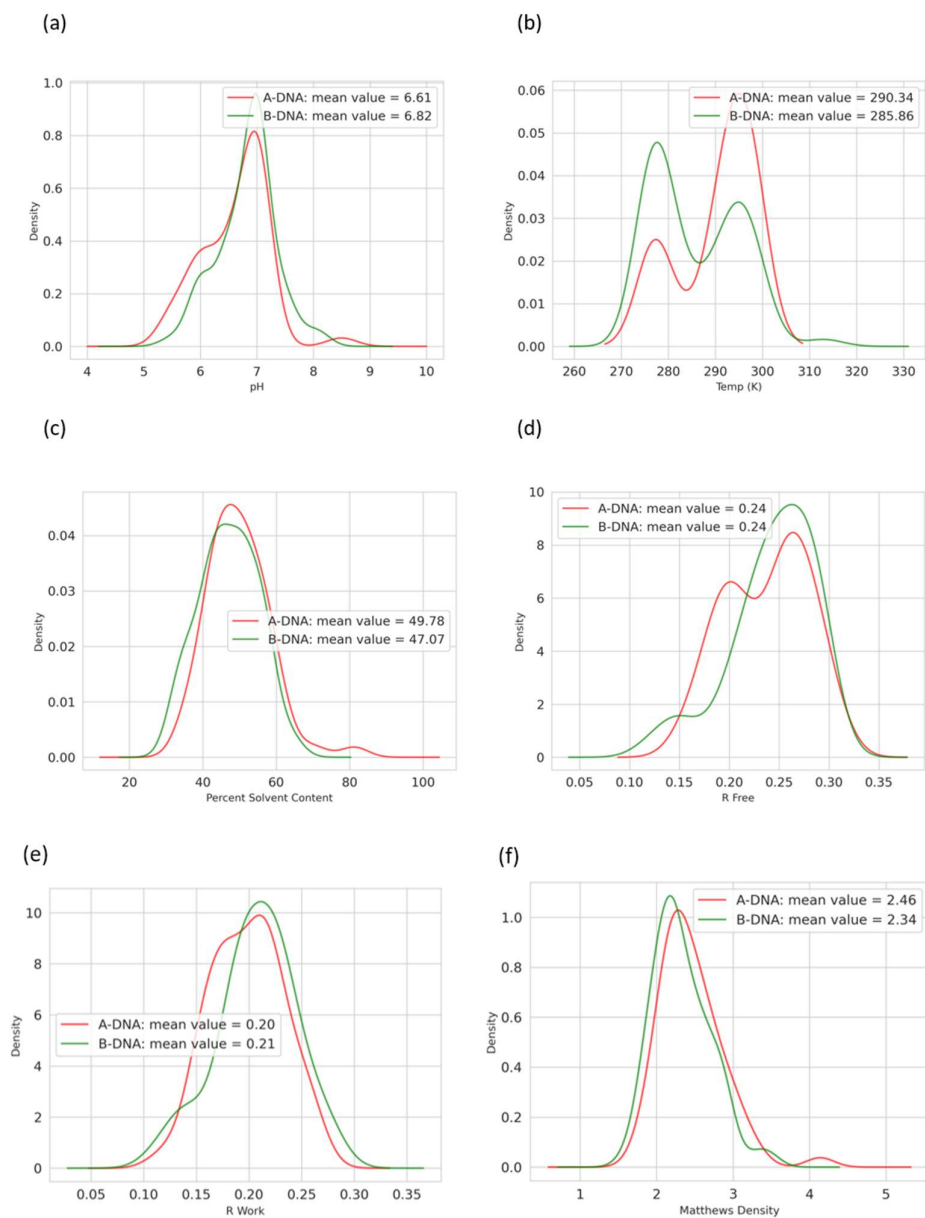


Figure S1: Kernel density estimation distribution of different experimental conditions under which samples were selected: (a) pH, (b) Temperature(K), (c) Percentage solvent content, (d) R-free values, (e) R work values, (f) Matthews Density

Temperature: Temperature in kelvins(K) at which the crystal was grown. If more than one temperature was employed during the crystallization process, the final temperature is reported here.

Matthews Density: It represents the density of the crystal, expressed as the ratio of the volume of the asymmetric unit to the molecular mass of a monomer of the structure. It is expressed in $\text{\AA}^3/Da$

Percentage solvent content: It is the density value calculated from the crystal cell and contents, expressed as percent solvent.

pH: The pH at which the crystal was grown. If more than one pH was employed during the crystallization process, the final pH is reported.

The R value is used to assess progress in the refinement of a model from X-ray crystallographic data and can be used as one factor in evaluating the quality of a model. It measures how well the simulated diffraction pattern matches the experimentally observed diffraction pattern.

Free R (R_{free}) is a statistical quantity that provides a better estimate of model-to-data agreement. Unlike R values, free R is free of any bias that may have been introduced during refinement¹.

The above graphs indicate that the various different experimental conditions for both "A" and "B" DNA follow a similar distribution, except for temperature, which shows some interesting bimodality. However, the mean and standard deviations are similar. Note that crystallization techniques often use very different temperature range.

For NMR structures, the mean sample temperature was 291K, mean sample pH value is 6.77, and mean sample pressure is 1 atm. The values of Ionic strength and solvent system used in the

experiment are provided in the separate excel sheet named "*Experimental conditions.xls*". This sheet provides detailed information about other experimental conditions and crystal properties, which we cannot include here owing to space constraints.

Detection of outliers using skewness adjusted Interquartile Range(IQR) method²

$$[Q_1 - h_l(MC) * IQR, Q_3 + h_u(MC) * IQR]$$

The points that lie outside this above range are classified as "potential outliers".

MC is the value of medcouple. It measures the skewness of a univariate distribution. Q_1 represents the first quartile and Q_3 represents the third quartile. The h_l and h_u are given by:

$$h_l(MC) = 1.5^{a*MC}$$

$$h_u(MC) = 1.5^{b*MC}$$

The values of a and b depend on the sign of MC . If $MC < 0$, $a = -4$ and $b = 3$ and if $MC \geq 0$, $a = -3$ and $b = 4$. We considered the distribution of different experimental/crystallization conditions to check for outlier samples. We have presented below the code for implementing this procedure.

```

def Skewness_IQR(*, data, name, MC=True):
    temp = data[name][data[name].notna()]
    q25, q75 = np.percentile(temp, 25), np.percentile(temp, 75)
    iqr = q75 - q25
    cut_off = iqr * 1.5
    if MC:
        mc = statsmodels.stats.stattools.medcouple(temp)
        if mc > 0:
            lower = q25 - cut_off*np.exp(-4*mc)
            upper = q75 + cut_off*np.exp(3*mc)
        else:
            lower = q25 - cut_off*np.exp(-3*mc)
            upper = q75 + cut_off*np.exp(4*mc)
    return lower, upper

```

Exploratory Data Analysis:

The figure below shows the class-conditional correlation between the features in our curated dataset.

(a)

	AA/TT	GG/CC	AC/GT	CA/TG	AT/AT	TA/TA	AG/CT	GA/TC	CG/CG	GC/GC
AA/TT	1.00	-0.17	-0.09	-0.07	-0.05	-0.07	0.14	0.25	-0.13	-0.02
GG/CC	-0.17	1.00	-0.43	-0.13	0.12	-0.40	-0.14	-0.12	-0.17	-0.26
AC/GT	-0.09	-0.43	1.00	0.19	-0.20	0.53	-0.26	-0.12	0.19	-0.32
CA/TG	-0.07	-0.13	0.19	1.00	0.28	-0.24	-0.19	-0.18	-0.24	0.08
AT/AT	-0.05	0.12	-0.20	0.28	1.00	-0.09	-0.07	0.14	-0.29	-0.13
TA/TA	-0.07	-0.40	0.53	-0.24	-0.09	1.00	0.34	-0.26	0.02	-0.22
AG/CT	0.14	-0.14	-0.26	-0.19	-0.07	0.34	1.00	0.24	-0.41	-0.08
GA/TC	0.25	-0.12	-0.12	-0.18	0.14	-0.26	0.24	1.00	-0.19	-0.32
CG/CG	-0.13	-0.17	0.19	-0.24	-0.29	0.02	-0.41	-0.19	1.00	0.45
GC/GC	-0.02	-0.26	-0.32	0.08	-0.13	-0.22	-0.08	-0.32	0.45	1.00

(b)

	AA/TT	GG/CC	AC/GT	CA/TG	AT/AT	TA/TA	AG/CT	GA/TC	CG/CG	GC/GC
AA/TT	1.00	-0.22	-0.15	-0.08	-0.08	-0.10	-0.19	-0.36	-0.09	-0.05
GG/CC	-0.22	1.00	0.03	0.04	-0.29	-0.18	0.10	-0.09	-0.17	-0.16
AC/GT	-0.15	0.03	1.00	0.18	-0.38	-0.01	-0.07	-0.02	0.07	-0.26
CA/TG	-0.08	0.04	0.18	1.00	0.07	-0.22	0.00	-0.29	-0.48	0.10
AT/AT	-0.08	-0.29	-0.38	0.07	1.00	0.50	-0.36	0.12	-0.01	-0.02
TA/TA	-0.10	-0.18	-0.01	-0.22	0.50	1.00	0.07	-0.20	-0.02	-0.08
AG/CT	-0.19	0.10	-0.07	0.00	-0.36	0.07	1.00	0.18	-0.39	0.00
GA/TC	-0.36	-0.09	-0.02	-0.29	0.12	-0.20	0.18	1.00	0.28	-0.16
CG/CG	-0.09	-0.17	0.07	-0.48	-0.01	-0.02	-0.39	0.28	1.00	0.36
GC/GC	-0.05	-0.16	-0.26	0.10	-0.02	-0.08	0.00	-0.16	0.36	1.00

Figure S2: Class-conditional sample correlation heatmap between different features for (a) A-DNA (b) B-DNA

We observe moderate correlation (both positive and negative) between different features in our dataset. It provides us with an insight that Machine learning algorithms that rely on class-conditional independence between different features might not be the right choice for our problem of classifying sequences.

Section B:

Results Obtained for different ML algorithms. We tried different ML algorithms for classification, the results of which are given below.

LightGBM: We have shown here the results of LightGBM, which was presented in the main text, for quick comparison.

Table S1: Classification performance of LightGBM algorithm with tuned hyperparameters across different test folds

	Average PR	AUC PR	ROC AUC	Accuracy	F1	MCC
Test Fold 1	0.954	0.952	0.969	0.923	0.857	0.822
Test Fold 2	0.946	0.944	0.973	0.923	0.880	0.825
Test Fold 3	0.987	0.986	0.994	0.947	0.917	0.878
Test Fold 4	0.994	0.993	0.997	0.947	0.917	0.878
Test Fold 5	0.906	0.904	0.939	0.895	0.833	0.756
Mean	0.957	0.956	0.974	0.927	0.881	0.832

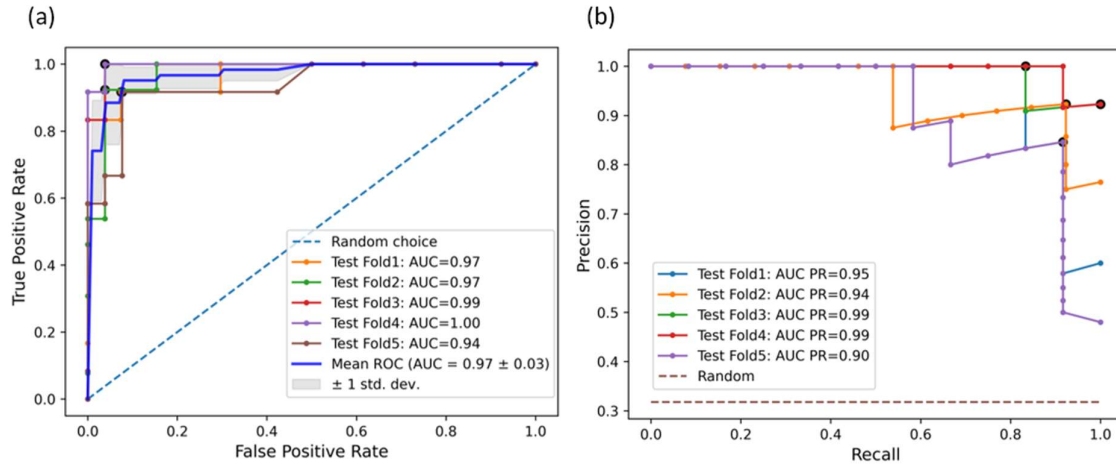


Figure S3: Results obtained for Nested Stratified five-fold cross-validation (a) ROC-AUC curves of tuned LightGBM model (b) Precision-Recall curves of tuned LightGBM model

Random Forest:

Table S2: Classification performance of the Random Forest algorithm with tuned hyperparameters across different test folds

	Average PR	AUC PR	ROC AUC	Accuracy	F1	MCC
Test Fold 1	0.839	0.834	0.934	0.846	0.786	0.686
Test Fold 2	0.826	0.816	0.925	0.846	0.786	0.671
Test Fold 3	0.903	0.910	0.952	0.842	0.786	0.682
Test Fold 4	1.000	1.000	1.000	0.974	0.957	0.940
Test Fold 5	0.870	0.865	0.913	0.895	0.833	0.756
Mean	0.888	0.885	0.945	0.881	0.829	0.747

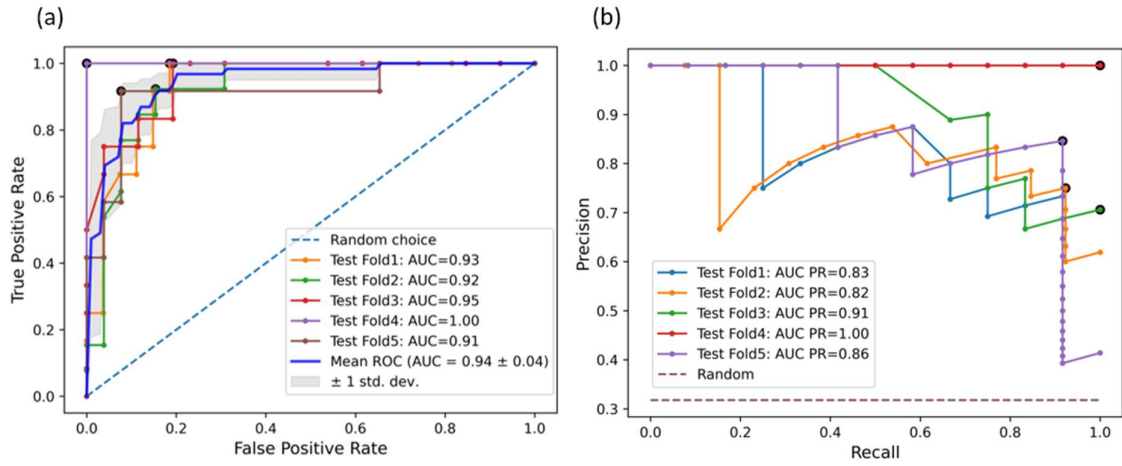


Figure S4: Results obtained for Nested Stratified five-fold cross-validation (a) ROC-AUC curves of tuned Random Forest model (b) Precision-Recall curves of tuned Random Forest model

Support Vector Machine Classifier:

Table S3: Classification performance of SVM Classifier algorithm with tuned hyperparameters across different test folds

	Average PR	AUC PR	ROC AUC	Accuracy	F1	MCC
Test Fold 1	0.925	0.922	0.960	0.897	0.833	0.759
Test Fold 2	0.793	0.783	0.876	0.795	0.714	0.559
Test Fold 3	0.988	0.988	0.994	0.947	0.909	0.880
Test Fold 4	0.924	0.922	0.946	0.895	0.818	0.751
Test Fold 5	0.927	0.925	0.939	0.895	0.833	0.756
Mean	0.912	0.908	0.943	0.886	0.822	0.741

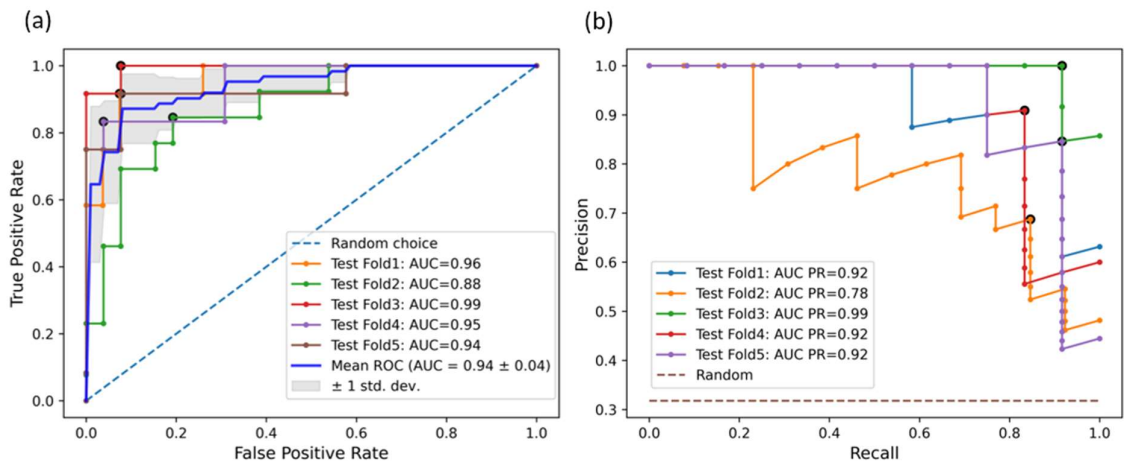


Figure S5: Results obtained for Nested Stratified five-fold cross-validation (a) ROC-AUC curves of tuned Support Vector Machine classifier model (b) Precision-Recall curves of tuned Support Vector Machine classifier model

Logistic Regression:

Table S4: Classification performance of Logistic Regression algorithm with tuned hyperparameters across different test folds

	Average PR	AUC PR	ROC AUC	Accuracy	F1	MCC
Test Fold 1	0.911	0.907	0.951	0.897	0.818	0.754
Test Fold 2	0.816	0.807	0.888	0.846	0.727	0.645
Test Fold 3	0.961	0.960	0.978	0.921	0.870	0.815
Test Fold 4	0.966	0.965	0.981	0.921	0.857	0.820
Test Fold 5	0.897	0.893	0.929	0.895	0.833	0.756
Mean	0.910	0.906	0.945	0.896	0.821	0.758

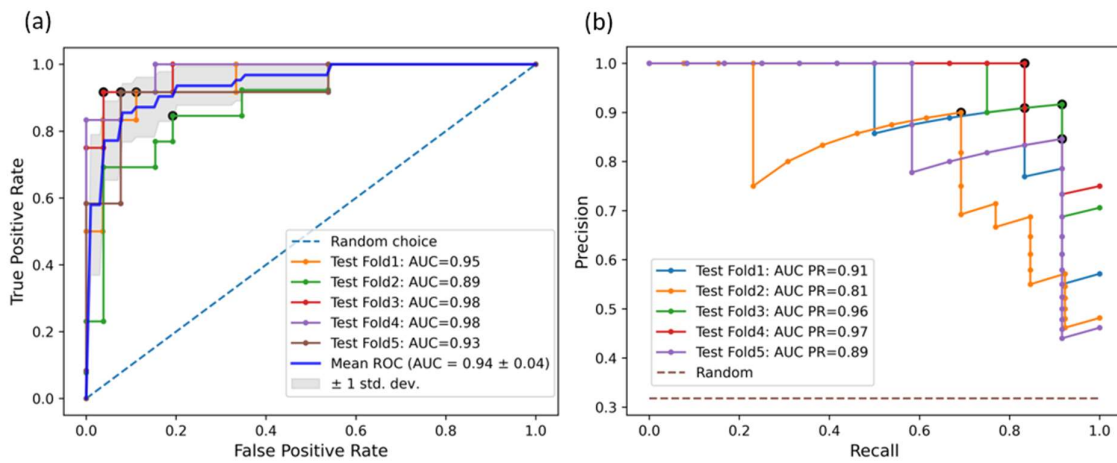


Figure S6: Results obtained for Nested Stratified five-fold cross-validation (a) ROC-AUC curves of tuned Logistic Regression model (b) Precision-Recall curves tuned Logistic Regression model

Naïve Bayes Classifier:

Table S5: Classification performance of Naïve Bayes algorithm with tuned hyperparameters across different test folds

	Average PR	AUC PR	ROC AUC	Accuracy	F1	MCC
Test Fold 1	0.865	0.859	0.935	0.846	0.786	0.686
Test Fold 2	0.909	0.906	0.944	0.821	0.759	0.627
Test Fold 3	0.964	0.963	0.971	0.947	0.909	0.880
Test Fold 4	0.961	0.959	0.984	0.947	0.917	0.878
Test Fold 5	0.895	0.902	0.925	0.895	0.833	0.756
Mean	0.919	0.918	0.952	0.891	0.841	0.765

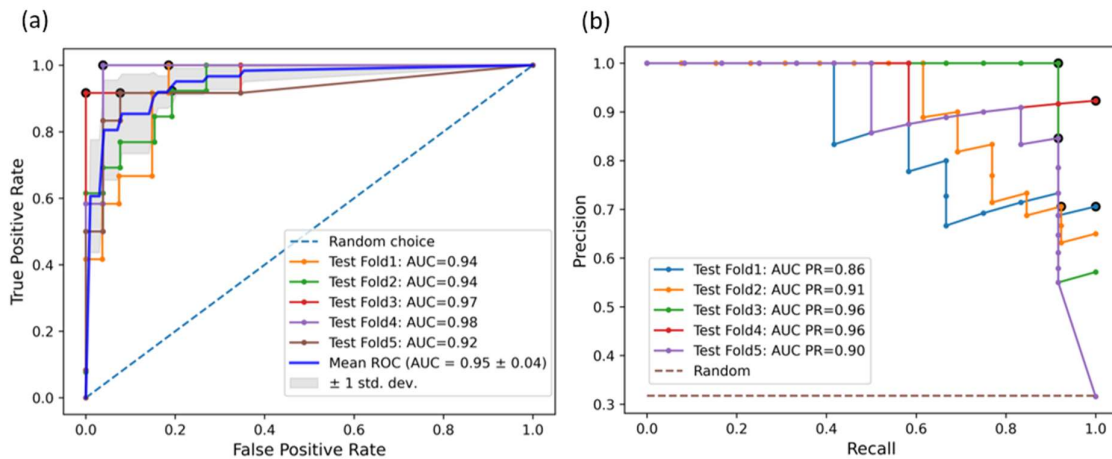


Figure S7: Results obtained for Nested Stratified five-fold cross-validation (a) ROC-AUC curves of tuned Naïve Bayes model (b) Precision-Recall curves tuned Naïve Bayes model

Section C:

Tuned Hyperparameters for LightGBM algorithm

'Model 1':

```
{'objective': 'binary',  
'seed': 42,  
'metric': 'auc',  
'verbosity': -1,  
'boosting_type': 'gbdt',  
'learning_rate': 0.1,  
'scale_pos_weight': 2.1475409836065578,  
'lambda_11': 1.1076661600828544e-05,  
'lambda_12': 7.17127731301496e-07,  
'num_leaves': 65,  
'feature_fraction': 0.847751242221898,  
'bagging_fraction': 0.7769441567151428,  
'bagging_freq': 1,  
'min_child_samples': 2,  
'num_iterations': 150,  
'early_stopping_round': None},
```

'Model 2':

```
{'objective': 'binary',  
'seed': 42,  
'metric': 'auc',  
'verbosity': -1,  
'boosting_type': 'gbdt',  
'learning_rate': 0.1,
```

'scale_pos_weight': 2.1475409836065578,
'lambda_11': 0.0002473760927419952,
'lambda_12': 2.300122453672787e-05,
'num_leaves': 244,
'feature_fraction': 0.5113852454880337,
'bagging_fraction': 0.6141970987056998,
'bagging_freq': 1,
'min_child_samples': 4,
'num_iterations': 150,
'early_stopping_round': None},

'Model_3':

{'objective': 'binary',
'seed': 42,
'metric': 'auc',
'verbosity': -1,
'boosting_type': 'gbdt',
'learning_rate': 0.1,
'scale_pos_weight': 2.1475409836065578,
'lambda_11': 0.0024514743127548003,
'lambda_12': 4.813012088689529e-07,
'num_leaves': 63,
'feature_fraction': 0.49408739562012993,
'bagging_fraction': 0.9049935235703056,
'bagging_freq': 1,
'min_child_samples': 9,
'num_iterations': 150,
'early_stopping_round': None},

'Model_4':

{'objective': 'binary',
'seed': 42,

```
'metric': 'auc',  
'verbosity': -1,  
'boosting_type': 'gbdt',  
'learning_rate': 0.1,  
'scale_pos_weight': 2.1475409836065578,  
'lambda_11': 7.402504100760827e-06,  
'lambda_12': 0.009939804805916733,  
'num_leaves': 201,  
'feature_fraction': 0.8771670627761629,  
'bagging_fraction': 0.5895571461784969,  
'bagging_freq': 1,  
'min_child_samples': 12,  
'num_iterations': 150,  
'early_stopping_round': None},
```

'Model_5':

```
{'objective': 'binary',  
'seed': 42,  
'metric': 'auc',  
'verbosity': -1,  
'boosting_type': 'gbdt',  
'learning_rate': 0.1,  
'scale_pos_weight': 2.1475409836065578,  
'lambda_11': 0.00012694834278974962,  
'lambda_12': 0.1677126712803063,  
'num_leaves': 101,  
'feature_fraction': 0.48023107028252104,  
'bagging_fraction': 0.9206698663841568,  
'bagging_freq': 7,  
'min_child_samples': 3,  
'num_iterations': 150,  
'early_stopping_round': None}}
```

Tuned Hyperparameters of other ML algorithms are presented in tabular form in the excel workbook – *Models.xlsx* inside sheet Tuned Parameters(hosted on https://github.com/abhijitmjj/DNA-structure-prediction_). We have also provided the hyperparameters for each model in the code repository.

Section D

Metrics used for evaluation of the classification model

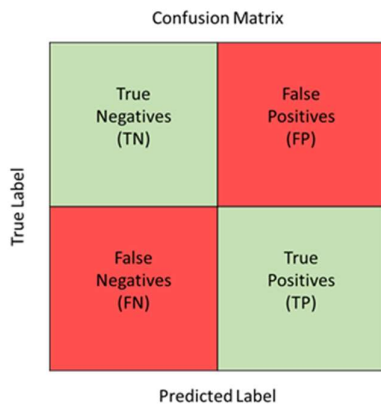


Figure S8: Confusion matrix description- it shows the actual and predicted labels from a classification problem.

Outcomes of binary classification (A vs B)

True positives: data points labelled as positive(A-DNA) that are positive(A-DNA)

False positives: data points labelled as positive(A-DNA) that are negative(B-DNA)

True negatives: data points labelled as negative(B-DNA) that are negative(B-DNA)

False negatives: data points labelled as negative(B-DNA) that are positive(A-DNA)

$$\text{Accuracy: } \frac{TP+TN}{TP+TN+FP+FN}$$

MCC (*Matthews Correlation Coefficient*)³ :

$$\frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Precision: It is the number of True Positives divided by the number of True Positives and False Positives.

Recall: It is the number of True Positives divided by the number of True Positives and the number of False Negatives.

F1-score: It conveys the balance between precision and recall and is measured by the following formula:

$$2 * \frac{precision * recall}{precision + recall}$$

Section E:

Supplementary Text

Calculation of change in absolute free energy values for $A \rightarrow B$ transition

The polymorphic nature of DNA makes the DNA conformation prediction a challenging task. The local or partial B-form to A-form transition of a small segment of DNA sequence always possesses the penalty of B-form/A-form junction formation on both 5' and 3' ends of newly formed A-form DNA segment in a whole sequence. Considering this fact, we performed rigorous umbrella sampling simulations to calculate this junction free energy values and characteristic local B-form to A-form free energy values for all ten unique dinucleotide steps⁴. The free energy values obtained therein are termed as "absolute free energy" values (ΔG_a) as they are devoid of any effects from flanking base pairs. We used umbrella sampling simulations along a new reaction coordinate Z'_p and average Z'_p ($\overline{Z'_p}$) for 10 unique dinucleotide steps and a few trinucleotide steps embedded in the 13-mer DNA sequence.⁵ These sequences, in general, can be presented as d(CGCGXXYYCGCG)₂, where X/Y can be either A, T, C, or G. The presence of CG sequences on both termini reduces the possibility of base pairs fraying at the ends.⁶ We showed earlier that creating an A-form in a B-DNA creates two B/A junctions. Therefore, the free energy obtained for the dinucleotide step XY (underlined in 13-mer sequence) from simulation can be written as,

$$\Delta G_{sim}(XY) = \Delta G_j(XX) + \Delta G_a(XY) + \Delta G_j(YY). \quad \text{Eq. 1}$$

At this stage, we are only aware of $\Delta G_{sim}(XY)$ value. We then performed simulations on di- and tri- homonucleotide sequences $d(\text{CGCGXXXXXXCGCG})_2$ to find the junction and absolute free energy values for homo-dinucleotide steps. The free energy cost to convert XX step along Z'_p in sequence $d(\text{CGCGXXXXXXCGCG})_2$ can be decomposed as,

$$\Delta G_{sim}(XX) = \Delta G_j(XX) + \Delta G_a(XX) + \Delta G_j(XX). \quad \text{Eq. 2}$$

Also, the free energy cost to convert XXX step in the same sequence $d(\text{CGCGXXXXXXCGCG})_2$ using an average Z'_p ($\overline{Z'_p}$) can be decomposed as,

$$\Delta G_{sim}(XXX) = \Delta G_j(XX) + 2\Delta G_a(XX) + \Delta G_j(XX) \quad \text{Eq. 3}$$

Subtracting Eq. 2 from Eq. 3, one can obtain absolute free energy value $\Delta G_a(XX)$ (which is devoid of any junction effect) for creating an A-form dinucleotide step within a B-DNA and eventually junction free energy values for homo dinucleotide steps AA, TT, GG, and CC. Table 1 lists these absolute and junction free energies. Using this junction free energy values ($\Delta G_j(XX)$ or $\Delta G_j(YY)$) one can calculate these absolute free energy values ($\Delta G_a(XY)$) for the rest of the hetero-dinucleotide steps. These values are also listed in Table 1. Note that, the junction effect comes only when a part of the DNA is converted from B-form to A-form. The full conversion of a B-DNA to A-DNA will depend only on the absolute free energy cost. That is the primary reason to calculate absolute free energy.

With these absolute free energy values, we constructed a model to predict the B- and A-DNA conformations from the sequence. This is similar to the earlier approach of Basham, who used the solvation free energy-based approach for trinucleotide step.⁷ However, in Basham's results, all the trinucleotide steps were not considered. In our approach, we use the dinucleotide step and thereby can consider all possible sequence variations. Moreover, we believe that this is a direct approach where the free energetic stability dictates the propensity for a particular conformation. However, translating this free energy cost from a dinucleotide step to a full DNA can be accomplished in multiple ways. We adopted a simple approach where we calculated the average free energy cost (ΔG_{avFE}) for the conformational transition of a DNA sequence between B- and A-form defined as,

$$\Delta G_{avFE} = \frac{1}{N} \sum_{i=1}^N \Delta G_a(XY),$$

where N is the number of dinucleotide steps in a given sequence. This number is equal to one less than the length of DNA sequence and $\Delta G_a(XY)$ is absolute free energy value for a particular dinucleotide step, where $X, Y = A, C, G, \text{ or } T$.

Section F:

Dataset S1: Filtered dataset and details of different splits of data used for nested 5-fold stratified cross-validation. The data was procured from NDB database^{6,8} and was then filtered to create this dataset.

	Experimental Method	Sequence	type	R Free	Resolution (Å)	R Observed	R Work
6RSO	X-RAY DIFFRACTION	TCGGCGCCGA	B	0.2474	1.97	0.209	0.2068
6QJR	X-RAY DIFFRACTION	CGCAAAAAGCG	B	0.2435	2.9	0.19145	0.18814
6F3C	X-RAY DIFFRACTION	CGTACG	B	0.2923	2.3	0.261	0.2573
6GIM	X-RAY DIFFRACTION	AAATTT	B	0.19286	1.43	0.14585	0.14342
6ASF	SOLUTION NMR	CCAAGATAG	B				
5M68	X-RAY DIFFRACTION	CGAATTAATTCG	B	0.2469	2.64	0.22824	0.22731
5GUN	X-RAY DIFFRACTION	GTGGAATGGAAC	B	0.2875	2.588	0.2449	0.2429
5UZF	SOLUTION NMR	CGATTTTTGGC	B				
5UZD	SOLUTION NMR	GCATCGATTGGC	B				
5J3F	SOLUTION NMR	CGGCCGCCGA	B				
2N5P	SOLUTION NMR	ATGGAGCTC	B				
2N9H	SOLUTION NMR	GATGACTGCTAG	B				
2N9F	SOLUTION NMR	CTAGCGGCATC	B				
5K14	SOLUTION NMR	ATCCGGTAG	B				
5K15	SOLUTION NMR	TTAGGCCTG	B				
4R6M	X-RAY DIFFRACTION	GGACTTCGCG	B	0.2405	2.357	0.2167	0.2155
2MNE	SOLUTION NMR	CGACTAGTCG	B				
2MH6	SOLUTION NMR	CAGTTCCA	B				
4OCD	X-RAY DIFFRACTION	AAAATTTT	B	0.25103	2.1	0.23682	0.23614
2RT8	SOLUTION NMR	CGCGTTGTCC	B				
4J2I	X-RAY DIFFRACTION	AATAAATTATT	B	0.28729	2.98	0.27031	0.26963

2MCI	SOLUTION NMR	GTCGGCTG	B				
2LZW	SOLUTION NMR	CGAAAGTTTCG	B				
4BZT	SOLUTION NMR	ATGCAT	B				
4BZU	SOLUTION NMR	TATGCATA	B				
2LWH	SOLUTION NMR	GGATATATCC	B				
3TOK	X-RAY DIFFRACTION	CCGATACCGG	B	0.2913	1.74	0.22042	0.2339
2LIB	SOLUTION NMR	GTCCAGGACG	B				
2LG3	SOLUTION NMR	GCTAGCGAGTCC	B				
4E1U	X-RAY DIFFRACTION	CGGAAATTACCG	B	0.149	0.92	0.1401	0.1397
2LGM	SOLUTION NMR	GCATGTGTACG	B				
1VTJ	X-RAY DIFFRACTION	CGCGATATCGCG	B		2.4	0.202	
3L1Q	X-RAY DIFFRACTION	TGGCCTTAAGG	B		2.5	0.22325	0.22325
3OMJ	X-RAY DIFFRACTION	CCAGTACTGG	B	0.1237	0.95	0.1127	0.1121
3N4N	X-RAY DIFFRACTION	CGCGAA	B	0.263	1.92		0.258
3LPV	X-RAY DIFFRACTION	CCTCTGGTCTCC	B	0.19788	1.77	0.17342	0.17212
2KY7	SOLUTION NMR	AACAATTGTT	B				
2KH5	SOLUTION NMR	GTGCGTGTGTTGT	B				
3FT6	X-RAY DIFFRACTION	CGATCG	B	0.21929	1.12	0.18811	0.18433
3EY0	X-RAY DIFFRACTION	ATATATATAT	B	0.27715	2.52	0.22193	0.21572
2KNK	SOLUTION NMR	AGGCGCCT	B				
2KNL	SOLUTION NMR	TCCGCGGA	B				
3IGT	X-RAY DIFFRACTION	CCGAGTCCTA	B	0.265	1.9	0.224	0.224
3EIL	X-RAY DIFFRACTION	CGTTAATTAACG	B	0.28471	2.6	0.23679	0.23465
2KAL	SOLUTION NMR	GCGAGATCTGCG	B				
3C2J	X-RAY DIFFRACTION	AACCGGTT	B	0.265	1.78	0.222	0.22
2Z2H	SOLUTION NMR	CTCGGCGCCATC	B				
2GOT	X-RAY DIFFRACTION	GCGAACGC	B	0.269	2.602	0.254	0.253
2OKS	X-RAY DIFFRACTION	CCAACGTTGG	B	0.208	1.65		0.182
2O1I	X-RAY DIFFRACTION	CGGAAATTCCCG	B	0.204	1.1		
2B2B	X-RAY DIFFRACTION	CCGCTAGCGG	B	0.26162	1.5	0.21549	0.20995
1X2O	SOLUTION NMR	GACTGTACAGTC	B				
2GE2	SOLUTION NMR	CGTACGCATGC	B				
1ZYF	SOLUTION NMR	CAACCATGGTTG	B				
1ZYG	SOLUTION NMR	CAACCGGGTTG	B				
1ZYH	SOLUTION NMR	CAACCAGGGTTG	B				
1X26	SOLUTION NMR	CTAACAGAATG	B				
2B1D	X-RAY DIFFRACTION	GCAGACGTCTGC	B	0.284	2.5	0.239	0.235

1XCI	SOLUTION NMR	CGAAATTTTCG	B				
1SK5	X-RAY DIFFRACTION	CTTTTAAAAG	B	0.144	0.89	0.12631	0.1263
1ZF7	X-RAY DIFFRACTION	CCGTCGACGG	B	0.288	1.05	0.276	0.276
1ZF3	X-RAY DIFFRACTION	CCGATATCGG	B	0.231	1.84	0.248	0.202
1ZF0	X-RAY DIFFRACTION	CCGTTAACGG	B	0.258	1.5	0.253	0.253
1Y9H	SOLUTION NMR	CCATCGCTACC	B				
1SY8	SOLUTION NMR	TGATCA	B				
1TUQ	SOLUTION NMR	CTCCACGTGGAG	B				
1S9B	X-RAY DIFFRACTION	GAATTCG	B	0.26428	2.81	0.281	0.281
1U6O	SOLUTION NMR	CGGACAAGAAG	B				
1PQQ	SOLUTION NMR	CGCTAACAGGC	B				
1S74	SOLUTION NMR	GTCCACGACG	B				
1S23	X-RAY DIFFRACTION	CGCAATTGCG	B	0.28152	1.6	0.21601	0.20969
1RVI	SOLUTION NMR	CGTTTTAAAACG	B				
1RVH	SOLUTION NMR	GCAAAATTTTGC	B				
1LP7	X-RAY DIFFRACTION	CGCTTATATGCG	B	0.293	2.4		0.229
1ONM	SOLUTION NMR	GCTTCAGTCGT	B				
1N1N	SOLUTION NMR	AGATCAATGT	B				
1OSR	SOLUTION NMR	AGGACCACG	B				
1HQ7	X-RAY DIFFRACTION	GCAAACGTTTGC	B	0.266	2.1	0.237	0.237
1NEV	SOLUTION NMR	GGCAAAACGG	B				
1N4E	X-RAY DIFFRACTION	GCTTAATTCG	B	0.255	2.5		0.197
1N37	SOLUTION NMR	AGACGTCT	B				
1G5K	SOLUTION NMR	CCAAAG	B				
1N0O	SOLUTION NMR	CCAAGG	B				
1N2W	SOLUTION NMR	CGCGAATTGGCG	B				
1MXK	SOLUTION NMR	GGAAGCTTCC	B				
1KVH	SOLUTION NMR	CCCGATGC	B				
1ILC	X-RAY DIFFRACTION	ACCGAATTCGGT	B	0.292	2.2	0.224	0.222
1K9G	X-RAY DIFFRACTION	CCTAGG	B	0.227	1.4	0.209	0.208
1IKK	X-RAY DIFFRACTION	CCTTTAAAGG	B	0.236	1.6	0.184	0.177
1ENN	X-RAY DIFFRACTION	GCGAATTCG	B	0.161	0.89	0.135	0.135
1QSX	SOLUTION NMR	CTTTTGCAAAAG	B				
1CVY	X-RAY DIFFRACTION	CCAGATCTGG	B	0.229	2.15	0.229	0.229
456D	X-RAY DIFFRACTION	CGCGAATCCGCG	B	0.231	1.6	0.196	0.196
1D8X	X-RAY DIFFRACTION	CCGAATGAGG	B	0.246	1.2	0.188	
424D	X-RAY DIFFRACTION	ACCGACGTCCGT	B	0.283	2.7	0.211	0.211

1QMS	SOLUTION NMR	GCACCTTCCTGC	B				
477D	X-RAY DIFFRACTION	GGCGAATTCGCG	B	0.235	1.7	0.194	0.194
476D	X-RAY DIFFRACTION	GCGAATTCGCG	B	0.22	1.3		0.182
1DSM	SOLUTION NMR	GACTAATTGAC	B				
335D	X-RAY DIFFRACTION	GGCAATTGCG	B		2.4	0.203	0.203
334D	X-RAY DIFFRACTION	CATGGCCATG	B		1.8	0.2	0.2
307D	X-RAY DIFFRACTION	CAAAGAAAAG	B		1.85	0.233	
309D	X-RAY DIFFRACTION	CGACGATCGT	B		2.6	0.214	
251D	X-RAY DIFFRACTION	CTCGAG	B		1.9	0.186	0.186
206D	X-RAY DIFFRACTION	CGGTGG	B		2.5	0.221	0.221
249D	X-RAY DIFFRACTION	CGCTCTAGAGCG	B		2.25	0.197	0.197
253D	X-RAY DIFFRACTION	GCGTACGCG	B	0.2137	2.2	0.1931	0.1931
226D	SOLUTION NMR	CGTTTTTACG	B				
1DXA	SOLUTION NMR	GGTCACGAG	B				
218D	X-RAY DIFFRACTION	CGTGAATTCGCG	B		2.25	0.167	0.167
202D	SOLUTION NMR	GACATGTC	B				
107D	SOLUTION NMR	CCTTTC	B				
194D	X-RAY DIFFRACTION	CGCGTTAACGCG	B		2.3	0.148	0.148
178D	X-RAY DIFFRACTION	CGCAAATTGGCG	B		2.5	0.168	
175D	SOLUTION NMR	GCGAATGAGC	B				
150D	X-RAY DIFFRACTION	CGCGAATTAGCG	B		2.25	0.182	0.182
158D	X-RAY DIFFRACTION	CCAAGCTTGG	B		1.9	0.179	
132D	SOLUTION NMR	GCCGTTAACGCG	B				
153D	X-RAY DIFFRACTION	CGAGAATTCGCG	B		2.9	0.169	
119D	X-RAY DIFFRACTION	CGTAGATCTACG	B		2.25	0.138	
1D89	X-RAY DIFFRACTION	CGCGAAAAAACG	B		2.3	0.232	0.232
1D83	SOLUTION NMR	AAGGCCTT	B				
1D69	SOLUTION NMR	ATGAGCGAATA	B				
1DA3	X-RAY DIFFRACTION	CGATCGATCG	B		2	0.172	
1D56	X-RAY DIFFRACTION	CGATATATCG	B		1.7	0.178	
1D49	X-RAY DIFFRACTION	CGATTAATCG	B		1.5	0.157	
1D20	SOLUTION NMR	TCTATCACCG	B				
1D18	SOLUTION NMR	CATGCATG	B				
1BDN	X-RAY DIFFRACTION	CGCAAAAATGCG	B		2.6	0.201	
1DN9	X-RAY DIFFRACTION	CGCATATATGCG	B		2.2	0.189	0.189
3DNB	X-RAY DIFFRACTION	CCAAGATTGG	B		1.3	0.164	
6GN2	X-RAY DIFFRACTION	CCCGGG	A	0.2723	2.48	0.2601	0.2586

5ZAS	X-RAY DIFFRACTION	CCAGCGCTGG	A	0.1621	1.56	0.14736	0.14666
6DXJ	X-RAY DIFFRACTION	GAGGCCTC	A	0.25535	1.65	0.21607	0.21395
6DY5	X-RAY DIFFRACTION	AGGGATCCCT	A	0.25374	1.26	0.21649	0.2143
6DY9	X-RAY DIFFRACTION	GGGATCCC	A	0.27582	2.3	0.2545	0.25346
5XK0	X-RAY DIFFRACTION	GCCCCGAGC	A	0.1956	1.451	0.181	0.1803
5MVQ	X-RAY DIFFRACTION	CTACGGCCGTAG	A	0.1938	1.604	0.1795	0.1785
5MVT	X-RAY DIFFRACTION	CTACGTACGTAG	A	0.2245	1.896	0.2031	0.2015
5MVP	X-RAY DIFFRACTION	CTAGGGCCCTAG	A	0.2089	1.606	0.1557	0.153
5WSS	X-RAY DIFFRACTION	GGTCGTCC	A	0.1934	1.45	0.15017	0.14799
5JVW	X-RAY DIFFRACTION	AGAGGCCTCT	A	0.26713	2	0.22118	0.21869
5JW0	X-RAY DIFFRACTION	AGGGTACCCT	A	0.28905	2.4	0.26353	0.2619
4YS5	X-RAY DIFFRACTION	GTGGCCAC	A	0.26294	1.65	0.22695	0.22517
4IZQ	X-RAY DIFFRACTION	GGGCATGCC	A	0.25227	2.04	0.1997	0.19703
4F4N	X-RAY DIFFRACTION	GTGTACAC	A	0.18111	1.3	0.15543	0.15416
1VT7	X-RAY DIFFRACTION	GGGTGCC	A		2.5	0.152	0.152
1VT5	X-RAY DIFFRACTION	CCCCGGGG	A		2.25	0.24	0.24
1VT9	X-RAY DIFFRACTION	GGGTACCC	A		2.5	0.119	0.119
3IFF	X-RAY DIFFRACTION	GTACGCGTAC	A	0.27831	1.75	0.20609	0.20226
2PLO	X-RAY DIFFRACTION	GGTATACC	A		1.4	0.1773	0.17
2B1B	X-RAY DIFFRACTION	GCGTGGGCAC	A	0.255	1.9	0.212	0.212
2B1C	X-RAY DIFFRACTION	GCGTGGGACC	A	0.286	2.2	0.234	0.234
2A7E	X-RAY DIFFRACTION	CCCTAGGG	A	0.197	1.66	0.184	0.183
1ZF6	X-RAY DIFFRACTION	CCCCATGGGG	A	0.306	1.5		0.256
1ZJE	X-RAY DIFFRACTION	AGGGGCGGGGCT	A	0.2561	2.1	0.2117	0.2168
1ZF8	X-RAY DIFFRACTION	CCACCGGTGG	A	0.263	1.48	0.22	0.22
1ZF9	X-RAY DIFFRACTION	CCCCCGGGGG	A	0.259	1.38	0.237	0.237
1ZF1	X-RAY DIFFRACTION	CCGGGCCCGG	A	0.245	1.35	0.222	0.222
1ZFA	X-RAY DIFFRACTION	CCTCCGGAGG	A	0.3	1.56	0.241	0.241
1R3Z	X-RAY DIFFRACTION	GCGCGCGC	A	0.203	1.7	0.193	0.17
1M77	X-RAY DIFFRACTION	CCCGATCGGG	A	0.185	1.25		0.163
382D	X-RAY DIFFRACTION	CCGCCGGCGG	A		2.2	0.197	0.197
414D	X-RAY DIFFRACTION	GGGGCGCCCC	A		1.9		
1QPH	X-RAY DIFFRACTION	GACCACGTGGTC	A		2.5	0.225	0.225
399D	X-RAY DIFFRACTION	CGCCCGGGGCG	A	0.203	1.9	0.165	0.165
440D	X-RAY DIFFRACTION	AGGGGCCCT	A	0.251	1.1	0.214	0.213
369D	X-RAY DIFFRACTION	CCCGCGGG	A	0.213	1.9	0.176	0.172
401D	X-RAY DIFFRACTION	GACCGGGTC	A	0.285	2.2	0.219	0.219

343D	X-RAY DIFFRACTION	GCTAGC	A	0.286	2.1	0.204	0.204
327D	X-RAY DIFFRACTION	GCGCGCGCGC	A	0.227	1.94	0.191	0.191
281D	X-RAY DIFFRACTION	GGCATGCC	A		2.38	0.171	0.171
260D	X-RAY DIFFRACTION	GCACGCGTGC	A		1.9	0.186	0.186
257D	X-RAY DIFFRACTION	GCCGGC	A	0.161	2.3	0.164	0.164
254D	X-RAY DIFFRACTION	GCGCGC	A	0.186	1.9	0.197	0.197
232D	X-RAY DIFFRACTION	AGGCATGCCT	A	0.216	1.3	0.1385	
243D	X-RAY DIFFRACTION	ACGTACGT	A		1.9	0.177	0.177
220D	X-RAY DIFFRACTION	ACCCGCGGGT	A		2	0.206	
207D	SOLUTION NMR	TAGCTAGCTA	A				
146D	SOLUTION NMR	TCGCGA	A				
197D	X-RAY DIFFRACTION	GTACGTAC	A		2.19	0.161	
172D	X-RAY DIFFRACTION	GAAGCTTC	A		3	0.212	0.212
138D	X-RAY DIFFRACTION	GCGGGCCCGC	A		1.8	0.183	0.183
1D93	X-RAY DIFFRACTION	CTCTAGAG	A		2.15	0.147	
1D91	X-RAY DIFFRACTION	GGGGTCCC	A		2.1	0.145	
1D92	X-RAY DIFFRACTION	GGGGCTCC	A		2.25	0.136	
116D	X-RAY DIFFRACTION	CCGTACGTACGG	A		2.5	0.15	
117D	X-RAY DIFFRACTION	GCGTACGTACGC	A		2.55	0.142	
118D	X-RAY DIFFRACTION	GTGCGCAC	A		1.64	0.154	
1D26	X-RAY DIFFRACTION	GCCCCGGC	A		2.12	0.16	
2D47	X-RAY DIFFRACTION	CCCCCGGGGG	A		2	0.177	
2ANA	X-RAY DIFFRACTION	GGGGCCCC	A		2.5	0.14	

Details of splits of the dataset into Train and Test set for nested 5-fold stratified cross-validation

Train Fold 1:

	Experimental Method	Sequence	type	R Free	Resolution (Å)	R Observed	R Work
6RSO	X-RAY DIFFRACTION	TCGGCGCCGA	B	0.2474	1.97	0.209	0.2068
6QJR	X-RAY DIFFRACTION	CGCAAAAAAGCG	B	0.2435	2.9	0.19145	0.18814
6GIM	X-RAY DIFFRACTION	AAATTT	B	0.19286	1.43	0.14585	0.14342
6ASF	SOLUTION NMR	CCAAGATAG	B				
5M68	X-RAY DIFFRACTION	CGAATTAATTCG	B	0.2469	2.64	0.22824	0.22731

5GUN	X-RAY DIFFRACTION	GTGGAATGGAAC	B	0.2875	2.588	0.2449	0.2429
5UZF	SOLUTION NMR	CGATTTTTTGGC	B				
5UZD	SOLUTION NMR	GCATCGATTGGC	B				
2N5P	SOLUTION NMR	ATGGAGCTC	B				
2N9H	SOLUTION NMR	GATGACTGCTAG	B				
2N9F	SOLUTION NMR	CTAGCGGTCATC	B				
5K15	SOLUTION NMR	TTAGGCCTG	B				
4R6M	X-RAY DIFFRACTION	GGACTTCGCG	B	0.2405	2.357	0.2167	0.2155
2MNE	SOLUTION NMR	CGACTAGTCG	B				
2MH6	SOLUTION NMR	CAGTTCCA	B				
4OCD	X-RAY DIFFRACTION	AAAATTTT	B	0.25103	2.1	0.23682	0.23614
2RT8	SOLUTION NMR	CGCGTTGTCC	B				
2LZW	SOLUTION NMR	CGAAAGTTTCG	B				
4BZU	SOLUTION NMR	TATGCATA	B				
2LWH	SOLUTION NMR	GGATATATCC	B				
3TOK	X-RAY DIFFRACTION	CCGATACCGG	B	0.2913	1.74	0.22042	0.2339
2LIB	SOLUTION NMR	GTCCAGGACG	B				
2LG3	SOLUTION NMR	GCTAGCGAGTCC	B				
1VTJ	X-RAY DIFFRACTION	CGCGATATCGCG	B		2.4	0.202	
3LIQ	X-RAY DIFFRACTION	TGGCCTTAAGG	B		2.5	0.22325	0.22325
3N4N	X-RAY DIFFRACTION	CGCGAA	B	0.263	1.92		0.258
3LPV	X-RAY DIFFRACTION	CCTCTGGTCTCC	B	0.19788	1.77	0.17342	0.17212
2KY7	SOLUTION NMR	AACAATTGTT	B				
3FT6	X-RAY DIFFRACTION	CGATCG	B	0.21929	1.12	0.18811	0.18433
2KNK	SOLUTION NMR	AGGCGCCT	B				
2KNL	SOLUTION NMR	TCCGCGGA	B				
3IGT	X-RAY DIFFRACTION	CCGAGTCCTA	B	0.265	1.9	0.224	0.224
3EIL	X-RAY DIFFRACTION	CGTTAATTAACG	B	0.28471	2.6	0.23679	0.23465
2KAL	SOLUTION NMR	GCGAGATCTGCG	B				
3C2J	X-RAY DIFFRACTION	AACCGGTT	B	0.265	1.78	0.222	0.22
2GOT	X-RAY DIFFRACTION	GCGAACGC	B	0.269	2.602	0.254	0.253
2OKS	X-RAY DIFFRACTION	CCAACGTTGG	B	0.208	1.65		0.182
2B2B	X-RAY DIFFRACTION	CCGCTAGCGG	B	0.26162	1.5	0.21549	0.20995
1X2O	SOLUTION NMR	GACTGTACAGTC	B				
2GE2	SOLUTION NMR	CGTACGCATGC	B				
1ZYF	SOLUTION NMR	CAACCATGGTTG	B				
1ZYG	SOLUTION NMR	CAACCCGGGTTG	B				

1X26	SOLUTION NMR	CTAACAGAATG	B				
2B1D	X-RAY DIFFRACTION	GCAGACGTCTGC	B	0.284	2.5	0.239	0.235
1XCI	SOLUTION NMR	CGAAATTTTCG	B				
1SK5	X-RAY DIFFRACTION	CTTTTAAAAG	B	0.144	0.89	0.12631	0.1263
1ZF7	X-RAY DIFFRACTION	CCGTCGACGG	B	0.288	1.05	0.276	0.276
1ZF0	X-RAY DIFFRACTION	CCGTTAACGG	B	0.258	1.5	0.253	0.253
1Y9H	SOLUTION NMR	CCATCGCTACC	B				
1SY8	SOLUTION NMR	TGATCA	B				
1S9B	X-RAY DIFFRACTION	GAATTCG	B	0.26428	2.81	0.281	0.281
1U6O	SOLUTION NMR	CGGACAAGAAG	B				
1PQQ	SOLUTION NMR	CGCTAACAGGC	B				
1S74	SOLUTION NMR	GTCCACGACG	B				
1RVH	SOLUTION NMR	GCAAATTTTGC	B				
1N1N	SOLUTION NMR	AGATCAATGT	B				
1OSR	SOLUTION NMR	AGGACCACG	B				
1HQ7	X-RAY DIFFRACTION	GCAAACGTTTGC	B	0.266	2.1	0.237	0.237
1N4E	X-RAY DIFFRACTION	GCTTAATTCG	B	0.255	2.5		0.197
1N37	SOLUTION NMR	AGACGTCT	B				
1G5K	SOLUTION NMR	CCAAAG	B				
1N0O	SOLUTION NMR	CCAAGG	B				
1N2W	SOLUTION NMR	CGCGAATTGGCG	B				
1MXK	SOLUTION NMR	GGAAGCTTCC	B				
1KVH	SOLUTION NMR	CCCGATGC	B				
1ILC	X-RAY DIFFRACTION	ACCGAATTCGGT	B	0.292	2.2	0.224	0.222
1K9G	X-RAY DIFFRACTION	CCTAGG	B	0.227	1.4	0.209	0.208
1IKK	X-RAY DIFFRACTION	CCTTTAAAGG	B	0.236	1.6	0.184	0.177
1ENN	X-RAY DIFFRACTION	GCGAATTCG	B	0.161	0.89	0.135	0.135
1QSX	SOLUTION NMR	CTTTTGCAAAAAG	B				
1CVY	X-RAY DIFFRACTION	CCAGATCTGG	B	0.229	2.15	0.229	0.229
456D	X-RAY DIFFRACTION	CGCGAATCCGCG	B	0.231	1.6	0.196	0.196
1D8X	X-RAY DIFFRACTION	CCGAATGAGG	B	0.246	1.2	0.188	
424D	X-RAY DIFFRACTION	ACCGACGTCCGGT	B	0.283	2.7	0.211	0.211
1QMS	SOLUTION NMR	GCACCTTCTGC	B				
477D	X-RAY DIFFRACTION	GGCGAATTCGCG	B	0.235	1.7	0.194	0.194
476D	X-RAY DIFFRACTION	GCGAATTCGCG	B	0.22	1.3		0.182
1DSM	SOLUTION NMR	GACTAATTGAC	B				
335D	X-RAY DIFFRACTION	GGCAATTGCG	B		2.4	0.203	0.203

334D	X-RAY DIFFRACTION	CATGGCCATG	B		1.8	0.2	0.2
307D	X-RAY DIFFRACTION	CAAAGAAAAG	B		1.85	0.233	
309D	X-RAY DIFFRACTION	CGACGATCGT	B		2.6	0.214	
251D	X-RAY DIFFRACTION	CTCGAG	B		1.9	0.186	0.186
206D	X-RAY DIFFRACTION	CGGTGG	B		2.5	0.221	0.221
249D	X-RAY DIFFRACTION	CGCTCTAGAGCG	B		2.25	0.197	0.197
253D	X-RAY DIFFRACTION	GCGTACGCG	B	0.2137	2.2	0.1931	0.1931
226D	SOLUTION NMR	CGTTTTTACG	B				
1DXA	SOLUTION NMR	GGTCACGAG	B				
202D	SOLUTION NMR	GACATGTC	B				
175D	SOLUTION NMR	GCGAATGAGC	B				
150D	X-RAY DIFFRACTION	CGCGAATTAGCG	B		2.25	0.182	0.182
158D	X-RAY DIFFRACTION	CCAAGCTTGG	B		1.9	0.179	
132D	SOLUTION NMR	GCCGTTAACGGC	B				
153D	X-RAY DIFFRACTION	CGAGAATTCGCG	B		2.9	0.169	
119D	X-RAY DIFFRACTION	CGTAGATCTACG	B		2.25	0.138	
1D89	X-RAY DIFFRACTION	CGCGAAAAAACG	B		2.3	0.232	0.232
1D83	SOLUTION NMR	AAGGCCTT	B				
1D69	SOLUTION NMR	ATGAGCGAATA	B				
1DA3	X-RAY DIFFRACTION	CGATCGATCG	B		2	0.172	
1D49	X-RAY DIFFRACTION	CGATTAATCG	B		1.5	0.157	
1D20	SOLUTION NMR	TCTATCACCG	B				
1D18	SOLUTION NMR	CATGCATG	B				
1DN9	X-RAY DIFFRACTION	CGCATATATGCG	B		2.2	0.189	0.189
3DNB	X-RAY DIFFRACTION	CCAAGATTGG	B		1.3	0.164	
6GN2	X-RAY DIFFRACTION	CCCGGG	A	0.2723	2.48	0.2601	0.2586
5ZAS	X-RAY DIFFRACTION	CCAGCGCTGG	A	0.1621	1.56	0.14736	0.14666
6DXJ	X-RAY DIFFRACTION	GAGGCCTC	A	0.25535	1.65	0.21607	0.21395
6DY5	X-RAY DIFFRACTION	AGGGATCCCT	A	0.25374	1.26	0.21649	0.2143
5XK0	X-RAY DIFFRACTION	GCCCGAGC	A	0.1956	1.451	0.181	0.1803
5MVQ	X-RAY DIFFRACTION	CTACGGCCGTAG	A	0.1938	1.604	0.1795	0.1785
5MVT	X-RAY DIFFRACTION	CTACGTACGTAG	A	0.2245	1.896	0.2031	0.2015
5MVP	X-RAY DIFFRACTION	CTAGGGCCCTAG	A	0.2089	1.606	0.1557	0.153
5WSS	X-RAY DIFFRACTION	GGTCGTCC	A	0.1934	1.45	0.15017	0.14799
5JVW	X-RAY DIFFRACTION	AGAGGCCTCT	A	0.26713	2	0.22118	0.21869
5JW0	X-RAY DIFFRACTION	AGGGTACCCT	A	0.28905	2.4	0.26353	0.2619
4YS5	X-RAY DIFFRACTION	GTGGCCAC	A	0.26294	1.65	0.22695	0.22517

41ZQ	X-RAY DIFFRACTION	GGGCATGCC	A	0.25227	2.04	0.1997	0.19703
4F4N	X-RAY DIFFRACTION	GTGTACAC	A	0.18111	1.3	0.15543	0.15416
1VT7	X-RAY DIFFRACTION	GGGTGCC	A		2.5	0.152	0.152
3IFF	X-RAY DIFFRACTION	GTACGCGTAC	A	0.27831	1.75	0.20609	0.20226
2B1C	X-RAY DIFFRACTION	GCGTGGGACC	A	0.286	2.2	0.234	0.234
2A7E	X-RAY DIFFRACTION	CCCTAGGG	A	0.197	1.66	0.184	0.183
1ZF6	X-RAY DIFFRACTION	CCCCATGGGG	A	0.306	1.5		0.256
1ZJE	X-RAY DIFFRACTION	AGGGGCGGGGCT	A	0.2561	2.1	0.2117	0.2168
1ZF8	X-RAY DIFFRACTION	CCACCGGTGG	A	0.263	1.48	0.22	0.22
1ZF1	X-RAY DIFFRACTION	CCGGGCCCGG	A	0.245	1.35	0.222	0.222
1ZFA	X-RAY DIFFRACTION	CCTCCGGAGG	A	0.3	1.56	0.241	0.241
1M77	X-RAY DIFFRACTION	CCCGATCGGG	A	0.185	1.25		0.163
382D	X-RAY DIFFRACTION	CCGCCGGCGG	A		2.2	0.197	0.197
414D	X-RAY DIFFRACTION	GGGGCGCCCC	A		1.9		
399D	X-RAY DIFFRACTION	CGCCCGGGGCG	A	0.203	1.9	0.165	0.165
440D	X-RAY DIFFRACTION	AGGGGCCCT	A	0.251	1.1	0.214	0.213
369D	X-RAY DIFFRACTION	CCCGCGGG	A	0.213	1.9	0.176	0.172
401D	X-RAY DIFFRACTION	GACCGCGGTC	A	0.285	2.2	0.219	0.219
343D	X-RAY DIFFRACTION	GCTAGC	A	0.286	2.1	0.204	0.204
327D	X-RAY DIFFRACTION	GCGCGCGCGC	A	0.227	1.94	0.191	0.191
260D	X-RAY DIFFRACTION	GCACGCGTGC	A		1.9	0.186	0.186
257D	X-RAY DIFFRACTION	GCCGGC	A	0.161	2.3	0.164	0.164
254D	X-RAY DIFFRACTION	GCGCGC	A	0.186	1.9	0.197	0.197
232D	X-RAY DIFFRACTION	AGGCATGCCT	A	0.216	1.3	0.1385	
220D	X-RAY DIFFRACTION	ACCCGCGGGT	A		2	0.206	
146D	SOLUTION NMR	TCGCGA	A				
197D	X-RAY DIFFRACTION	GTACGTAC	A		2.19	0.161	
172D	X-RAY DIFFRACTION	GAAGCTTC	A		3	0.212	0.212
138D	X-RAY DIFFRACTION	GCGGGCCCCGC	A		1.8	0.183	0.183
1D91	X-RAY DIFFRACTION	GGGGTCCC	A		2.1	0.145	
1D92	X-RAY DIFFRACTION	GGGGCTCC	A		2.25	0.136	
116D	X-RAY DIFFRACTION	CCGTACGTACGG	A		2.5	0.15	
117D	X-RAY DIFFRACTION	GCGTACGTACGC	A		2.55	0.142	
118D	X-RAY DIFFRACTION	GTGCGCAC	A		1.64	0.154	
1D26	X-RAY DIFFRACTION	GCCCCGGG	A		2.12	0.16	
2D47	X-RAY DIFFRACTION	CCCCCGGGGG	A		2	0.177	
2ANA	X-RAY DIFFRACTION	GGGGCCCC	A		2.5	0.14	

Test Fold 1:

	Experimental Method	Sequence	type	R Free	Resolution (Å)	R Observed	R Work
6F3C	X-RAY DIFFRACTION	CGTACG	B	0.2923	2.3	0.261	0.2573
5J3F	SOLUTION NMR	CGGCCGCCGA	B				
5K14	SOLUTION NMR	ATCCGGTAG	B				
4J2I	X-RAY DIFFRACTION	AATAAATTTATT	B	0.28729	2.98	0.27031	0.26963
2MCI	SOLUTION NMR	GTCGGCTG	B				
4BZT	SOLUTION NMR	ATGCAT	B				
4E1U	X-RAY DIFFRACTION	CGGAAATTACCG	B	0.149	0.92	0.1401	0.1397
2LGM	SOLUTION NMR	GCATGTGTACG	B				
3OMJ	X-RAY DIFFRACTION	CCAGTACTGG	B	0.1237	0.95	0.1127	0.1121
2KH5	SOLUTION NMR	GTGCGTGTTTGT	B				
3EY0	X-RAY DIFFRACTION	ATATATATAT	B	0.27715	2.52	0.22193	0.21572
2Z2H	SOLUTION NMR	CTCGGCGCCATC	B				
2O1I	X-RAY DIFFRACTION	CGGAAATTCCCG	B	0.204	1.1		
1ZYH	SOLUTION NMR	CAACCAGGGTTG	B				
1ZF3	X-RAY DIFFRACTION	CCGATATCGG	B	0.231	1.84	0.248	0.202
1TUQ	SOLUTION NMR	CTCCACGTGGAG	B				
1S23	X-RAY DIFFRACTION	CGCAATTGCG	B	0.28152	1.6	0.21601	0.20969
1RVI	SOLUTION NMR	CGTTTTAAAACG	B				
1LP7	X-RAY DIFFRACTION	CGCTTATATGCG	B	0.293	2.4		0.229
1ONM	SOLUTION NMR	GCTTCAGTCGT	B				
1NEV	SOLUTION NMR	GGCAAAAACGG	B				
218D	X-RAY DIFFRACTION	CGTGAATTCGCG	B		2.25	0.167	0.167
107D	SOLUTION NMR	CCTTTTC	B				
194D	X-RAY DIFFRACTION	CGCGTTAACGCG	B		2.3	0.148	0.148
178D	X-RAY DIFFRACTION	CGCAAATGGCG	B		2.5	0.168	
1D56	X-RAY DIFFRACTION	CGATATATCG	B		1.7	0.178	
1BDN	X-RAY DIFFRACTION	CGCAAAAATGCG	B		2.6	0.201	
6DY9	X-RAY DIFFRACTION	GGGATCCC	A	0.27582	2.3	0.2545	0.25346
1VT5	X-RAY DIFFRACTION	CCCCGGGG	A		2.25	0.24	0.24
1VT9	X-RAY DIFFRACTION	GGGTACCC	A		2.5	0.119	0.119
2PLO	X-RAY DIFFRACTION	GGTATACC	A		1.4	0.1773	0.17
2B1B	X-RAY DIFFRACTION	GCGTGGGCAC	A	0.255	1.9	0.212	0.212
1ZF9	X-RAY DIFFRACTION	CCCCCGGGGG	A	0.259	1.38	0.237	0.237
1R3Z	X-RAY DIFFRACTION	GCGCGCGC	A	0.203	1.7	0.193	0.17

1QPH	X-RAY DIFFRACTION	GACCACGTGGTC	A		2.5	0.225	0.225
281D	X-RAY DIFFRACTION	GGCATGCC	A		2.38	0.171	0.171
243D	X-RAY DIFFRACTION	ACGTACGT	A		1.9	0.177	0.177
207D	SOLUTION NMR	TAGCTAGCTA	A				
1D93	X-RAY DIFFRACTION	CTCTAGAG	A		2.15	0.147	

Train Fold 2:

	Experimental Method	Sequence	type	R Free	Resolution (Å)	R Observed	R Work
6RSO	X-RAY DIFFRACTION	TCGGCGCCGA	B	0.2474	1.97	0.209	0.2068
6F3C	X-RAY DIFFRACTION	CGTACG	B	0.2923	2.3	0.261	0.2573
6ASF	SOLUTION NMR	CCAAGATAG	B				
5M68	X-RAY DIFFRACTION	CGAATTAATTCG	B	0.2469	2.64	0.22824	0.22731
5GUN	X-RAY DIFFRACTION	GTGGAATGGAAC	B	0.2875	2.588	0.2449	0.2429
5UZF	SOLUTION NMR	CGATTTTTGGC	B				
5UZD	SOLUTION NMR	GCATCGATTGGC	B				
5J3F	SOLUTION NMR	CGGCCGCCGA	B				
2N5P	SOLUTION NMR	ATGGAGCTC	B				
2N9F	SOLUTION NMR	CTAGCGGTCATC	B				
5KI4	SOLUTION NMR	ATCCGGTAG	B				
5KI5	SOLUTION NMR	TTAGGCCTG	B				
4R6M	X-RAY DIFFRACTION	GGACTTCGCG	B	0.2405	2.357	0.2167	0.2155
2MH6	SOLUTION NMR	CAGTTCCA	B				
4OCD	X-RAY DIFFRACTION	AAAATTTT	B	0.25103	2.1	0.23682	0.23614
2RT8	SOLUTION NMR	CGCGTTGTCC	B				
4J2I	X-RAY DIFFRACTION	AATAAATTTATT	B	0.28729	2.98	0.27031	0.26963
2MCI	SOLUTION NMR	GTCGGCTG	B				
2LZW	SOLUTION NMR	CGAAAGTTTCG	B				
4BZT	SOLUTION NMR	ATGCAT	B				
4BZU	SOLUTION NMR	TATGCATA	B				
3TOK	X-RAY DIFFRACTION	CCGATACCGG	B	0.2913	1.74	0.22042	0.2339
2LIB	SOLUTION NMR	GTCCAGGACG	B				
4E1U	X-RAY DIFFRACTION	CGGAAATTACCG	B	0.149	0.92	0.1401	0.1397
2LGM	SOLUTION NMR	GCATGTGTACG	B				
3OMJ	X-RAY DIFFRACTION	CCAGTACTGG	B	0.1237	0.95	0.1127	0.1121
3N4N	X-RAY DIFFRACTION	CGCGAA	B	0.263	1.92		0.258

3LPV	X-RAY DIFFRACTION	CCTCTGGTCTCC	B	0.19788	1.77	0.17342	0.17212
2KY7	SOLUTION NMR	AACAATTGTT	B				
2KH5	SOLUTION NMR	GTGCGTGTTTGT	B				
3FT6	X-RAY DIFFRACTION	CGATCG	B	0.21929	1.12	0.18811	0.18433
3EY0	X-RAY DIFFRACTION	ATATATATAT	B	0.27715	2.52	0.22193	0.21572
2KNL	SOLUTION NMR	TCCGCGGA	B				
3IGT	X-RAY DIFFRACTION	CCGAGTCCTA	B	0.265	1.9	0.224	0.224
2KAL	SOLUTION NMR	GCGAGATCTGCG	B				
3C2J	X-RAY DIFFRACTION	AACCGGTT	B	0.265	1.78	0.222	0.22
2Z2H	SOLUTION NMR	CTCGGCGCCATC	B				
2OKS	X-RAY DIFFRACTION	CCAACGTTGG	B	0.208	1.65		0.182
2O1I	X-RAY DIFFRACTION	CGGAAATTCCTG	B	0.204	1.1		
2B2B	X-RAY DIFFRACTION	CCGCTAGCGG	B	0.26162	1.5	0.21549	0.20995
2GE2	SOLUTION NMR	CGTACGCATGC	B				
1ZYF	SOLUTION NMR	CAACCATGGTTG	B				
1ZYG	SOLUTION NMR	CAACCCGGGTTG	B				
1ZYH	SOLUTION NMR	CAACCAGGGTTG	B				
1X26	SOLUTION NMR	CTAACAGAATG	B				
2B1D	X-RAY DIFFRACTION	GCAGACGTCTGC	B	0.284	2.5	0.239	0.235
1ZF3	X-RAY DIFFRACTION	CCGATATCGG	B	0.231	1.84	0.248	0.202
1ZF0	X-RAY DIFFRACTION	CCGTTAACGG	B	0.258	1.5	0.253	0.253
1SY8	SOLUTION NMR	TGATCA	B				
1TUQ	SOLUTION NMR	CTCCACGTGGAG	B				
1U6O	SOLUTION NMR	CGGACAAGAAG	B				
1PQQ	SOLUTION NMR	CGCTAACAGGC	B				
1S74	SOLUTION NMR	GTCCACGACG	B				
1S23	X-RAY DIFFRACTION	CGCAATTGCG	B	0.28152	1.6	0.21601	0.20969
1RVI	SOLUTION NMR	CGTTTTAAAACG	B				
1RVH	SOLUTION NMR	GCAAAATTTTGC	B				
1LP7	X-RAY DIFFRACTION	CGCTTATATGCG	B	0.293	2.4		0.229
1ONM	SOLUTION NMR	GCTTCAGTCGT	B				
1OSR	SOLUTION NMR	AGGACCACG	B				
1HQ7	X-RAY DIFFRACTION	GCAAACGTTTGC	B	0.266	2.1	0.237	0.237
1NEV	SOLUTION NMR	GGCAAAACGG	B				
1N4E	X-RAY DIFFRACTION	GCTTAATTCG	B	0.255	2.5		0.197
1N37	SOLUTION NMR	AGACGTCT	B				
1G5K	SOLUTION NMR	CCAAAG	B				

1N00	SOLUTION NMR	CCAAGG	B				
1MXK	SOLUTION NMR	GGAAGCTTCC	B				
1ILC	X-RAY DIFFRACTION	ACCGAATTCGGT	B	0.292	2.2	0.224	0.222
1K9G	X-RAY DIFFRACTION	CCTAGG	B	0.227	1.4	0.209	0.208
1IKK	X-RAY DIFFRACTION	CCTTTAAAGG	B	0.236	1.6	0.184	0.177
1ENN	X-RAY DIFFRACTION	GCGAATTCG	B	0.161	0.89	0.135	0.135
1QSX	SOLUTION NMR	CTTTTGCAAAAG	B				
1CVY	X-RAY DIFFRACTION	CCAGATCTGG	B	0.229	2.15	0.229	0.229
1D8X	X-RAY DIFFRACTION	CCGAATGAGG	B	0.246	1.2	0.188	
424D	X-RAY DIFFRACTION	ACCGACGTCGGT	B	0.283	2.7	0.211	0.211
477D	X-RAY DIFFRACTION	GGCGAATTCGCG	B	0.235	1.7	0.194	0.194
476D	X-RAY DIFFRACTION	GCGAATTCGCG	B	0.22	1.3		0.182
1DSM	SOLUTION NMR	GACTAATTGAC	B				
335D	X-RAY DIFFRACTION	GGCAATGCG	B		2.4	0.203	0.203
307D	X-RAY DIFFRACTION	CAAAGAAAAG	B		1.85	0.233	
309D	X-RAY DIFFRACTION	CGACGATCGT	B		2.6	0.214	
251D	X-RAY DIFFRACTION	CTCGAG	B		1.9	0.186	0.186
206D	X-RAY DIFFRACTION	CGGTGG	B		2.5	0.221	0.221
226D	SOLUTION NMR	CGTTTTTACG	B				
1DXA	SOLUTION NMR	GGTCACGAG	B				
218D	X-RAY DIFFRACTION	CGTGAATTCGCG	B		2.25	0.167	0.167
107D	SOLUTION NMR	CCTTTC	B				
194D	X-RAY DIFFRACTION	CGCGTTAACGCG	B		2.3	0.148	0.148
178D	X-RAY DIFFRACTION	CGCAAATTGGCG	B		2.5	0.168	
175D	SOLUTION NMR	GCGAATGAGC	B				
150D	X-RAY DIFFRACTION	CGCGAATTAGCG	B		2.25	0.182	0.182
158D	X-RAY DIFFRACTION	CCAAGCTTGG	B		1.9	0.179	
132D	SOLUTION NMR	GCCGTTAACGGC	B				
153D	X-RAY DIFFRACTION	CGAGAATTCGCG	B		2.9	0.169	
119D	X-RAY DIFFRACTION	CGTAGATCTACG	B		2.25	0.138	
1D89	X-RAY DIFFRACTION	CGCGAAAAACG	B		2.3	0.232	0.232
1D83	SOLUTION NMR	AAGGCCTT	B				
1D69	SOLUTION NMR	ATGAGCGAATA	B				
1DA3	X-RAY DIFFRACTION	CGATCGATCG	B		2	0.172	
1D56	X-RAY DIFFRACTION	CGATATATCG	B		1.7	0.178	
1D49	X-RAY DIFFRACTION	CGATTAATCG	B		1.5	0.157	
1D20	SOLUTION NMR	TCTATCACCG	B				

1D18	SOLUTION NMR	CATGCATG	B				
1BDN	X-RAY DIFFRACTION	CGCAAAAATGCG	B		2.6	0.201	
1DN9	X-RAY DIFFRACTION	CGCATATATGCG	B		2.2	0.189	0.189
3DNB	X-RAY DIFFRACTION	CCAAGATTGG	B		1.3	0.164	
6GN2	X-RAY DIFFRACTION	CCCGGG	A	0.2723	2.48	0.2601	0.2586
5ZAS	X-RAY DIFFRACTION	CCAGCGCTGG	A	0.1621	1.56	0.14736	0.14666
6DY5	X-RAY DIFFRACTION	AGGGATCCCT	A	0.25374	1.26	0.21649	0.2143
6DY9	X-RAY DIFFRACTION	GGGATCCC	A	0.27582	2.3	0.2545	0.25346
5MVQ	X-RAY DIFFRACTION	CTACGGCCGTAG	A	0.1938	1.604	0.1795	0.1785
5MVP	X-RAY DIFFRACTION	CTAGGGCCCTAG	A	0.2089	1.606	0.1557	0.153
5WSS	X-RAY DIFFRACTION	GGTCGTCC	A	0.1934	1.45	0.15017	0.14799
5JVW	X-RAY DIFFRACTION	AGAGGCCTCT	A	0.26713	2	0.22118	0.21869
4YS5	X-RAY DIFFRACTION	GTGGCCAC	A	0.26294	1.65	0.22695	0.22517
4IZQ	X-RAY DIFFRACTION	GGGCATGCCC	A	0.25227	2.04	0.1997	0.19703
4F4N	X-RAY DIFFRACTION	GTGTACAC	A	0.18111	1.3	0.15543	0.15416
1VT5	X-RAY DIFFRACTION	CCCGGGG	A		2.25	0.24	0.24
1VT9	X-RAY DIFFRACTION	GGGTACCC	A		2.5	0.119	0.119
2PLO	X-RAY DIFFRACTION	GGTATACC	A		1.4	0.1773	0.17
2B1B	X-RAY DIFFRACTION	GCGTGGGCAC	A	0.255	1.9	0.212	0.212
2B1C	X-RAY DIFFRACTION	GCGTGGGACC	A	0.286	2.2	0.234	0.234
1ZF6	X-RAY DIFFRACTION	CCCCATGGGG	A	0.306	1.5		0.256
1ZJE	X-RAY DIFFRACTION	AGGGGCGGGGCT	A	0.2561	2.1	0.2117	0.2168
1ZF8	X-RAY DIFFRACTION	CCACCGGTGG	A	0.263	1.48	0.22	0.22
1ZF9	X-RAY DIFFRACTION	CCCCCGGGG	A	0.259	1.38	0.237	0.237
1ZF1	X-RAY DIFFRACTION	CCGGGCCCGG	A	0.245	1.35	0.222	0.222
1ZFA	X-RAY DIFFRACTION	CCTCCGAGG	A	0.3	1.56	0.241	0.241
1R3Z	X-RAY DIFFRACTION	GCGCGCGC	A	0.203	1.7	0.193	0.17
382D	X-RAY DIFFRACTION	CCGCCGGCGG	A		2.2	0.197	0.197
414D	X-RAY DIFFRACTION	GGGGCGCCCC	A		1.9		
1QPH	X-RAY DIFFRACTION	GACCACGTGGTC	A		2.5	0.225	0.225
399D	X-RAY DIFFRACTION	CGCCCGGGGCG	A	0.203	1.9	0.165	0.165
440D	X-RAY DIFFRACTION	AGGGGCCCT	A	0.251	1.1	0.214	0.213
401D	X-RAY DIFFRACTION	GACCGGGTC	A	0.285	2.2	0.219	0.219
343D	X-RAY DIFFRACTION	GCTAGC	A	0.286	2.1	0.204	0.204
281D	X-RAY DIFFRACTION	GGCATGCC	A		2.38	0.171	0.171
260D	X-RAY DIFFRACTION	GCACGCGTGC	A		1.9	0.186	0.186
257D	X-RAY DIFFRACTION	GCCGGC	A	0.161	2.3	0.164	0.164

254D	X-RAY DIFFRACTION	GCGCGC	A	0.186	1.9	0.197	0.197
232D	X-RAY DIFFRACTION	AGGCATGCCT	A	0.216	1.3	0.1385	
243D	X-RAY DIFFRACTION	ACGTACGT	A		1.9	0.177	0.177
220D	X-RAY DIFFRACTION	ACCCGCGGGT	A		2	0.206	
207D	SOLUTION NMR	TAGCTAGCTA	A				
197D	X-RAY DIFFRACTION	GTACGTAC	A		2.19	0.161	
172D	X-RAY DIFFRACTION	GAAGCTTC	A		3	0.212	0.212
138D	X-RAY DIFFRACTION	GCGGGCCCGC	A		1.8	0.183	0.183
1D93	X-RAY DIFFRACTION	CTCTAGAG	A		2.15	0.147	
1D91	X-RAY DIFFRACTION	GGGGTCCC	A		2.1	0.145	
1D92	X-RAY DIFFRACTION	GGGGCTCC	A		2.25	0.136	
117D	X-RAY DIFFRACTION	GCGTACGTACGC	A		2.55	0.142	
118D	X-RAY DIFFRACTION	GTGCGCAC	A		1.64	0.154	
1D26	X-RAY DIFFRACTION	GCCCCGGC	A		2.12	0.16	
2D47	X-RAY DIFFRACTION	CCCCCGCGGGGG	A		2	0.177	

Test Fold 2:

	Experimental Method	Sequence	type	R Free	Resolution (Å)	R Observed	R Work
6QJR	X-RAY DIFFRACTION	CGCAAAAAGCG	B	0.2435	2.9	0.19145	0.18814
6GIM	X-RAY DIFFRACTION	AAATTT	B	0.19286	1.43	0.14585	0.14342
2N9H	SOLUTION NMR	GATGACTGCTAG	B				
2MNE	SOLUTION NMR	CGACTAGTCG	B				
2LWH	SOLUTION NMR	GGATATATCC	B				
2LG3	SOLUTION NMR	GCTAGCGAGTCC	B				
1VTJ	X-RAY DIFFRACTION	CGCGATATCGCG	B		2.4	0.202	
3L1Q	X-RAY DIFFRACTION	TGGCCTTAAGG	B		2.5	0.22325	0.22325
2KNK	SOLUTION NMR	AGGCGCCT	B				
3EIL	X-RAY DIFFRACTION	CGTTAATTAACG	B	0.28471	2.6	0.23679	0.23465
2GOT	X-RAY DIFFRACTION	GCGAACGC	B	0.269	2.602	0.254	0.253
1X2O	SOLUTION NMR	GACTGTACAGTC	B				
1XCI	SOLUTION NMR	CGAAATTTTCG	B				
1SK5	X-RAY DIFFRACTION	CTTTAAAAG	B	0.144	0.89	0.12631	0.1263
1ZF7	X-RAY DIFFRACTION	CCGTCGACGG	B	0.288	1.05	0.276	0.276
1Y9H	SOLUTION NMR	CCATCGCTACC	B				
1S9B	X-RAY DIFFRACTION	GAATTCG	B	0.26428	2.81	0.281	0.281
1N1N	SOLUTION NMR	AGATCAATGT	B				
1N2W	SOLUTION NMR	CGCGAATTGGCG	B				
1KVH	SOLUTION NMR	CCCGATGC	B				
456D	X-RAY DIFFRACTION	CGCGAATCCGCG	B	0.231	1.6	0.196	0.196
1QMS	SOLUTION NMR	GCACCTTCTGTC	B				
334D	X-RAY DIFFRACTION	CATGGCCATG	B		1.8	0.2	0.2
249D	X-RAY DIFFRACTION	CGCTCTAGAGCG	B		2.25	0.197	0.197

253D	X-RAY DIFFRACTION	GCGTACGCG	B	0.2137	2.2	0.1931	0.1931
202D	SOLUTION NMR	GACATGTC	B				
6DXJ	X-RAY DIFFRACTION	GAGGCCTC	A	0.25535	1.65	0.21607	0.21395
5XK0	X-RAY DIFFRACTION	GCCCGAGC	A	0.1956	1.451	0.181	0.1803
5MVT	X-RAY DIFFRACTION	CTACGTACGTAG	A	0.2245	1.896	0.2031	0.2015
5JW0	X-RAY DIFFRACTION	AGGGTACCCT	A	0.28905	2.4	0.26353	0.2619
1VT7	X-RAY DIFFRACTION	GGGTGCC	A		2.5	0.152	0.152
3IFF	X-RAY DIFFRACTION	GTACGCGTAC	A	0.27831	1.75	0.20609	0.20226
2A7E	X-RAY DIFFRACTION	CCCTAGGG	A	0.197	1.66	0.184	0.183
1M77	X-RAY DIFFRACTION	CCCGATCGGG	A	0.185	1.25		0.163
369D	X-RAY DIFFRACTION	CCCGCGGG	A	0.213	1.9	0.176	0.172
327D	X-RAY DIFFRACTION	GCGCGCGCGC	A	0.227	1.94	0.191	0.191
146D	SOLUTION NMR	TCGCGA	A				
116D	X-RAY DIFFRACTION	CCGTACGTACGG	A		2.5	0.15	
2ANA	X-RAY DIFFRACTION	GGGGCCCC	A		2.5	0.14	

Train Fold 3:

	Experimental Method	Sequence	type	R Free	Resolution (Å)	R Observed	R Work
6QJR	X-RAY DIFFRACTION	CGCAAAAAGCG	B	0.2435	2.9	0.19145	0.18814
6F3C	X-RAY DIFFRACTION	CGTACG	B	0.2923	2.3	0.261	0.2573
6GIM	X-RAY DIFFRACTION	AAATTT	B	0.19286	1.43	0.14585	0.14342
6ASF	SOLUTION NMR	CCAAGATAG	B				
5UZF	SOLUTION NMR	CGATTTTTGGC	B				
5UZD	SOLUTION NMR	GCATCGATTGGC	B				
5J3F	SOLUTION NMR	CGGCCGCCGA	B				
2N5P	SOLUTION NMR	ATGGAGCTC	B				
2N9H	SOLUTION NMR	GATGACTGCTAG	B				
5KI4	SOLUTION NMR	ATCCGGTAG	B				
5KI5	SOLUTION NMR	TTAGGCCTG	B				
4R6M	X-RAY DIFFRACTION	GGACTTCGCG	B	0.2405	2.357	0.2167	0.2155
2MNE	SOLUTION NMR	CGACTAGTCG	B				
2MH6	SOLUTION NMR	CAGTCCA	B				
4OCD	X-RAY DIFFRACTION	AAAATTTT	B	0.25103	2.1	0.23682	0.23614
2RT8	SOLUTION NMR	CGCGTTGTCC	B				
4J2I	X-RAY DIFFRACTION	AATAAATTTATT	B	0.28729	2.98	0.27031	0.26963
2MCI	SOLUTION NMR	GTCGGCTG	B				

2LZW	SOLUTION NMR	CGAAAGTTTCG	B				
4BZT	SOLUTION NMR	ATGCAT	B				
4BZU	SOLUTION NMR	TATGCATA	B				
2LWH	SOLUTION NMR	GGATATATCC	B				
2LG3	SOLUTION NMR	GCTAGCGAGTCC	B				
4E1U	X-RAY DIFFRACTION	CGGAAATTACCG	B	0.149	0.92	0.1401	0.1397
2LGM	SOLUTION NMR	GCATGTGTACG	B				
1VTJ	X-RAY DIFFRACTION	CGCGATATCGCG	B		2.4	0.202	
3L1Q	X-RAY DIFFRACTION	TGGCCTTAAGG	B		2.5	0.22325	0.22325
30MJ	X-RAY DIFFRACTION	CCAGTACTGG	B	0.1237	0.95	0.1127	0.1121
3LPV	X-RAY DIFFRACTION	CCTCTGGTCTCC	B	0.19788	1.77	0.17342	0.17212
2KY7	SOLUTION NMR	AACAATTGTT	B				
2KH5	SOLUTION NMR	GTGCGTGTGTTGT	B				
3EY0	X-RAY DIFFRACTION	ATATATATAT	B	0.27715	2.52	0.22193	0.21572
2KNK	SOLUTION NMR	AGGCGCCT	B				
3IGT	X-RAY DIFFRACTION	CCGAGTCCTA	B	0.265	1.9	0.224	0.224
3EIL	X-RAY DIFFRACTION	CGTTAATTAACG	B	0.28471	2.6	0.23679	0.23465
3C2J	X-RAY DIFFRACTION	AACCGGTT	B	0.265	1.78	0.222	0.22
2Z2H	SOLUTION NMR	CTCGCGCCATC	B				
2GOT	X-RAY DIFFRACTION	GCGAACGC	B	0.269	2.602	0.254	0.253
2O1I	X-RAY DIFFRACTION	CGGAAATTCGG	B	0.204	1.1		
2B2B	X-RAY DIFFRACTION	CCGCTAGCGG	B	0.26162	1.5	0.21549	0.20995
1X2O	SOLUTION NMR	GACTGTACAGTC	B				
2GE2	SOLUTION NMR	CGTACGCATGC	B				
1ZYF	SOLUTION NMR	CAACCATGGTTG	B				
1ZYH	SOLUTION NMR	CAACCAGGGTTG	B				
1XCI	SOLUTION NMR	CGAAATTTTCG	B				
1SK5	X-RAY DIFFRACTION	CTTTTAAAAG	B	0.144	0.89	0.12631	0.1263
1ZF7	X-RAY DIFFRACTION	CCGTCGACGG	B	0.288	1.05	0.276	0.276
1ZF3	X-RAY DIFFRACTION	CCGATATCGG	B	0.231	1.84	0.248	0.202
1ZF0	X-RAY DIFFRACTION	CCGTTAACGG	B	0.258	1.5	0.253	0.253
1Y9H	SOLUTION NMR	CCATCGCTACC	B				
1SY8	SOLUTION NMR	TGATCA	B				
1TUQ	SOLUTION NMR	CTCCACGTGGAG	B				
1S9B	X-RAY DIFFRACTION	GAATTCG	B	0.26428	2.81	0.281	0.281
1PQQ	SOLUTION NMR	CGCTAACAGGC	B				
1S74	SOLUTION NMR	GTCCACGACG	B				
1S23	X-RAY DIFFRACTION	CGCAATTGCG	B	0.28152	1.6	0.21601	0.20969
1RVI	SOLUTION NMR	CGTTTTAAAACG	B				
1RVH	SOLUTION NMR	GCAAAATTTTGC	B				
1LP7	X-RAY DIFFRACTION	CGCTTATATGCG	B	0.293	2.4		0.229
1ONM	SOLUTION NMR	GCTTCAGTCGT	B				
1N1N	SOLUTION NMR	AGATCAATGT	B				
1HQ7	X-RAY DIFFRACTION	GCAAACGTTTGC	B	0.266	2.1	0.237	0.237
1NEV	SOLUTION NMR	GGCAAACGG	B				
1N4E	X-RAY DIFFRACTION	GCTTAATTCG	B	0.255	2.5		0.197
1N37	SOLUTION NMR	AGACGTCT	B				
1G5K	SOLUTION NMR	CCAAAG	B				
1N2W	SOLUTION NMR	CGCGAATTGGCG	B				
1MXK	SOLUTION NMR	GGAAGCTTCC	B				
1KVH	SOLUTION NMR	CCCGATGC	B				
1ILC	X-RAY DIFFRACTION	ACCGAATTCGGT	B	0.292	2.2	0.224	0.222

11KK	X-RAY DIFFRACTION	CCTTTAAAGG	B	0.236	1.6	0.184	0.177
1ENN	X-RAY DIFFRACTION	GCGAATTCG	B	0.161	0.89	0.135	0.135
1QSX	SOLUTION NMR	CTTTTGCAAAAG	B				
1CVY	X-RAY DIFFRACTION	CCAGATCTGG	B	0.229	2.15	0.229	0.229
456D	X-RAY DIFFRACTION	CGCGAATCCGCG	B	0.231	1.6	0.196	0.196
424D	X-RAY DIFFRACTION	ACCGACGTCGGT	B	0.283	2.7	0.211	0.211
1QMS	SOLUTION NMR	GCACCTTCTGC	B				
477D	X-RAY DIFFRACTION	GGCGAATTCGCG	B	0.235	1.7	0.194	0.194
476D	X-RAY DIFFRACTION	GCGAATTCGCG	B	0.22	1.3		0.182
334D	X-RAY DIFFRACTION	CATGGCCATG	B		1.8	0.2	0.2
307D	X-RAY DIFFRACTION	CAAAGAAAAG	B		1.85	0.233	
206D	X-RAY DIFFRACTION	CGGTGG	B		2.5	0.221	0.221
249D	X-RAY DIFFRACTION	CGCTCTAGAGCG	B		2.25	0.197	0.197
253D	X-RAY DIFFRACTION	GCGTACGCG	B	0.2137	2.2	0.1931	0.1931
226D	SOLUTION NMR	CGTTTTTACG	B				
1DXA	SOLUTION NMR	GGTCACGAG	B				
218D	X-RAY DIFFRACTION	CGTGAATTCGCG	B		2.25	0.167	0.167
202D	SOLUTION NMR	GACATGTC	B				
107D	SOLUTION NMR	CCTTTTC	B				
194D	X-RAY DIFFRACTION	CGCGTAAACGCG	B		2.3	0.148	0.148
178D	X-RAY DIFFRACTION	CGCAAATTGGCG	B		2.5	0.168	
175D	SOLUTION NMR	GCGAATGAGC	B				
150D	X-RAY DIFFRACTION	CGCGAATTAGCG	B		2.25	0.182	0.182
158D	X-RAY DIFFRACTION	CCAAGCTTGG	B		1.9	0.179	
132D	SOLUTION NMR	GCCGTAAACGGC	B				
153D	X-RAY DIFFRACTION	CGAGAATTCGCG	B		2.9	0.169	
1D89	X-RAY DIFFRACTION	CGCGAAAAAACG	B		2.3	0.232	0.232
1D83	SOLUTION NMR	AAGGCCTT	B				
1D69	SOLUTION NMR	ATGAGCGAATA	B				
1DA3	X-RAY DIFFRACTION	CGATCGATCG	B		2	0.172	
1D56	X-RAY DIFFRACTION	CGATATATCG	B		1.7	0.178	
1D18	SOLUTION NMR	CATGCATG	B				
1BDN	X-RAY DIFFRACTION	CGCAAAAATGCG	B		2.6	0.201	
1DN9	X-RAY DIFFRACTION	CGCATATATGCG	B		2.2	0.189	0.189
3DNB	X-RAY DIFFRACTION	CCAAGATTGG	B		1.3	0.164	
5ZAS	X-RAY DIFFRACTION	CCAGCGCTGG	A	0.1621	1.56	0.14736	0.14666
6DXJ	X-RAY DIFFRACTION	GAGGCCTC	A	0.25535	1.65	0.21607	0.21395
6DY5	X-RAY DIFFRACTION	AGGGATCCCT	A	0.25374	1.26	0.21649	0.2143
6DY9	X-RAY DIFFRACTION	GGGATCCC	A	0.27582	2.3	0.2545	0.25346
5XK0	X-RAY DIFFRACTION	GCCCCGAGC	A	0.1956	1.451	0.181	0.1803
5MVQ	X-RAY DIFFRACTION	CTACGGCCGTAG	A	0.1938	1.604	0.1795	0.1785
5MVT	X-RAY DIFFRACTION	CTACGTACGTAG	A	0.2245	1.896	0.2031	0.2015
5WSS	X-RAY DIFFRACTION	GGTCGTCC	A	0.1934	1.45	0.15017	0.14799
5JVW	X-RAY DIFFRACTION	AGAGGCCTCT	A	0.26713	2	0.22118	0.21869
5JW0	X-RAY DIFFRACTION	AGGGTACCCT	A	0.28905	2.4	0.26353	0.2619
4YS5	X-RAY DIFFRACTION	GTGGCCAC	A	0.26294	1.65	0.22695	0.22517
4IZQ	X-RAY DIFFRACTION	GGGCATGCC	A	0.25227	2.04	0.1997	0.19703
4F4N	X-RAY DIFFRACTION	GTGTACAC	A	0.18111	1.3	0.15543	0.15416
1VT7	X-RAY DIFFRACTION	GGGTGCC	A		2.5	0.152	0.152
1VT5	X-RAY DIFFRACTION	CCCCGGGG	A		2.25	0.24	0.24
1VT9	X-RAY DIFFRACTION	GGGTACCC	A		2.5	0.119	0.119
3IFF	X-RAY DIFFRACTION	GTACGCGTAC	A	0.27831	1.75	0.20609	0.20226

2PLO	X-RAY DIFFRACTION	GGTATACC	A		1.4	0.1773	0.17
2B1B	X-RAY DIFFRACTION	GCGTGGGCAC	A	0.255	1.9	0.212	0.212
2B1C	X-RAY DIFFRACTION	GCGTGGGACC	A	0.286	2.2	0.234	0.234
2A7E	X-RAY DIFFRACTION	CCCTAGGG	A	0.197	1.66	0.184	0.183
1ZF9	X-RAY DIFFRACTION	CCCCCGGGG	A	0.259	1.38	0.237	0.237
1ZF1	X-RAY DIFFRACTION	CCGGGCCCGG	A	0.245	1.35	0.222	0.222
1ZF6	X-RAY DIFFRACTION	CCTCCGAGG	A	0.3	1.56	0.241	0.241
1R3Z	X-RAY DIFFRACTION	GCGCGCGC	A	0.203	1.7	0.193	0.17
1M77	X-RAY DIFFRACTION	CCCGATCGGG	A	0.185	1.25		0.163
382D	X-RAY DIFFRACTION	CCGCCGGCGG	A		2.2	0.197	0.197
1QPH	X-RAY DIFFRACTION	GACCACGTGGTC	A		2.5	0.225	0.225
369D	X-RAY DIFFRACTION	CCCGCGGG	A	0.213	1.9	0.176	0.172
401D	X-RAY DIFFRACTION	GACCGCGGTC	A	0.285	2.2	0.219	0.219
343D	X-RAY DIFFRACTION	GCTAGC	A	0.286	2.1	0.204	0.204
327D	X-RAY DIFFRACTION	GCGCGCGCGC	A	0.227	1.94	0.191	0.191
281D	X-RAY DIFFRACTION	GGCATGCC	A		2.38	0.171	0.171
260D	X-RAY DIFFRACTION	GCACGCGTGC	A		1.9	0.186	0.186
257D	X-RAY DIFFRACTION	GCCGGC	A	0.161	2.3	0.164	0.164
254D	X-RAY DIFFRACTION	GCGCGC	A	0.186	1.9	0.197	0.197
243D	X-RAY DIFFRACTION	ACGTACGT	A		1.9	0.177	0.177
220D	X-RAY DIFFRACTION	ACCCGCGGGT	A		2	0.206	
207D	SOLUTION NMR	TAGCTAGCTA	A				
146D	SOLUTION NMR	TCGCGA	A				
197D	X-RAY DIFFRACTION	GTACGTAC	A		2.19	0.161	
172D	X-RAY DIFFRACTION	GAAGCTTC	A		3	0.212	0.212
138D	X-RAY DIFFRACTION	GCGGGCCCGC	A		1.8	0.183	0.183
1D93	X-RAY DIFFRACTION	CTCTAGAG	A		2.15	0.147	
1D91	X-RAY DIFFRACTION	GGGGTCCC	A		2.1	0.145	
116D	X-RAY DIFFRACTION	CCGTACGTACGG	A		2.5	0.15	
117D	X-RAY DIFFRACTION	GCGTACGTACGC	A		2.55	0.142	
118D	X-RAY DIFFRACTION	GTGCGCAC	A		1.64	0.154	
2ANA	X-RAY DIFFRACTION	GGGGCCCC	A		2.5	0.14	

Test Fold 3:

	Experimental Method	Sequence	type	R Free	Resolution (Å)	R Observed	R Work
6RSO	X-RAY DIFFRACTION	TCGGCGCCGA	B	0.2474	1.97	0.209	0.2068
5M68	X-RAY DIFFRACTION	CGAATTAATTCG	B	0.2469	2.64	0.22824	0.22731
5GUN	X-RAY DIFFRACTION	GTGGAATGGAAC	B	0.2875	2.588	0.2449	0.2429
2N9F	SOLUTION NMR	CTAGCGGTCATC	B				
3TOK	X-RAY DIFFRACTION	CCGATACCGG	B	0.2913	1.74	0.22042	0.2339
2LIB	SOLUTION NMR	GTCCAGGACG	B				
3N4N	X-RAY DIFFRACTION	CGCGAA	B	0.263	1.92		0.258
3FT6	X-RAY DIFFRACTION	CGATCG	B	0.21929	1.12	0.18811	0.18433
2KNL	SOLUTION NMR	TCCGCGGA	B				
2KAL	SOLUTION NMR	GCGAGATCTGCG	B				
2OKS	X-RAY DIFFRACTION	CCAACGTTGG	B	0.208	1.65		0.182

1ZYG	SOLUTION NMR	CAACCCGGGTTG	B				
1X26	SOLUTION NMR	CTAACAGAATG	B				
2B1D	X-RAY DIFFRACTION	GCAGACGTCTGC	B	0.284	2.5	0.239	0.235
1U6O	SOLUTION NMR	CGGACAAGAAG	B				
1OSR	SOLUTION NMR	AGGACCACG	B				
1N0O	SOLUTION NMR	CCAAGG	B				
1K9G	X-RAY DIFFRACTION	CCTAGG	B	0.227	1.4	0.209	0.208
1D8X	X-RAY DIFFRACTION	CCGAATGAGG	B	0.246	1.2	0.188	
1DSM	SOLUTION NMR	GACTAATTGAC	B				
335D	X-RAY DIFFRACTION	GGCAATTGCG	B		2.4	0.203	0.203
309D	X-RAY DIFFRACTION	CGACGATCGT	B		2.6	0.214	
251D	X-RAY DIFFRACTION	CTCGAG	B		1.9	0.186	0.186
119D	X-RAY DIFFRACTION	CGTAGATCTACG	B		2.25	0.138	
1D49	X-RAY DIFFRACTION	CGATTAATCG	B		1.5	0.157	
1D20	SOLUTION NMR	TCTATCACCG	B				
6GN2	X-RAY DIFFRACTION	CCCGGG	A	0.2723	2.48	0.2601	0.2586
5MVP	X-RAY DIFFRACTION	CTAGGGCCCTAG	A	0.2089	1.606	0.1557	0.153
1ZF6	X-RAY DIFFRACTION	CCCATGGGG	A	0.306	1.5		0.256
1ZJE	X-RAY DIFFRACTION	AGGGGCGGGGCT	A	0.2561	2.1	0.2117	0.2168
1ZF8	X-RAY DIFFRACTION	CCACCGGTGG	A	0.263	1.48	0.22	0.22
414D	X-RAY DIFFRACTION	GGGGCGCCCC	A		1.9		
399D	X-RAY DIFFRACTION	CGCCCGGGGCG	A	0.203	1.9	0.165	0.165
440D	X-RAY DIFFRACTION	AGGGGCCCT	A	0.251	1.1	0.214	0.213
232D	X-RAY DIFFRACTION	AGGCATGCCT	A	0.216	1.3	0.1385	
1D92	X-RAY DIFFRACTION	GGGGCTCC	A		2.25	0.136	
1D26	X-RAY DIFFRACTION	GCCCGGGC	A		2.12	0.16	
2D47	X-RAY DIFFRACTION	CCCCCGGGGGG	A		2	0.177	

Train Fold 4:

	Experimental Method	Sequence	type	R Free	Resolution (Å)	R Observed	R Work
6RSO	X-RAY DIFFRACTION	TCGGCGCCGA	B	0.2474	1.97	0.209	0.2068
6QJR	X-RAY DIFFRACTION	CGAAAAAAGCG	B	0.2435	2.9	0.19145	0.18814
6F3C	X-RAY DIFFRACTION	CGTACG	B	0.2923	2.3	0.261	0.2573
6GIM	X-RAY DIFFRACTION	AAATTT	B	0.19286	1.43	0.14585	0.14342
6ASF	SOLUTION NMR	CCAAGATAG	B				
5M68	X-RAY DIFFRACTION	CGAATTAATTCG	B	0.2469	2.64	0.22824	0.22731
5GUN	X-RAY DIFFRACTION	GTGGAATGGAAC	B	0.2875	2.588	0.2449	0.2429
5UZF	SOLUTION NMR	CGATTTTTGGC	B				
5J3F	SOLUTION NMR	CGCCCGCCGA	B				

2N9H	SOLUTION NMR	GATGACTGCTAG	B				
2N9F	SOLUTION NMR	CTAGCGGTCATC	B				
5KI4	SOLUTION NMR	ATCCGGTAG	B				
2MNE	SOLUTION NMR	CGACTAGTCG	B				
4J2I	X-RAY DIFFRACTION	AATAAATTTATT	B	0.28729	2.98	0.27031	0.26963
2MCI	SOLUTION NMR	GTCGGCTG	B				
2LZW	SOLUTION NMR	CGAAAGTTTCG	B				
4BZT	SOLUTION NMR	ATGCAT	B				
4BZU	SOLUTION NMR	TATGCATA	B				
2LWH	SOLUTION NMR	GGATATATCC	B				
3TOK	X-RAY DIFFRACTION	CCGATACCGG	B	0.2913	1.74	0.22042	0.2339
2LIB	SOLUTION NMR	GTCCAGGACG	B				
2LG3	SOLUTION NMR	GCTAGCGAGTCC	B				
4E1U	X-RAY DIFFRACTION	CGGAAATTACCG	B	0.149	0.92	0.1401	0.1397
2LGM	SOLUTION NMR	GCATGTGTACG	B				
1VTJ	X-RAY DIFFRACTION	CGCGATATCGCG	B		2.4	0.202	
3LIQ	X-RAY DIFFRACTION	TGGCCTTAAGG	B		2.5	0.22325	0.22325
30MJ	X-RAY DIFFRACTION	CCAGTACTGG	B	0.1237	0.95	0.1127	0.1121
3N4N	X-RAY DIFFRACTION	CGCGAA	B	0.263	1.92		0.258
3LPV	X-RAY DIFFRACTION	CCTCTGGTCTCC	B	0.19788	1.77	0.17342	0.17212
2KH5	SOLUTION NMR	GTGCGTGTGTTGT	B				
3FT6	X-RAY DIFFRACTION	CGATCG	B	0.21929	1.12	0.18811	0.18433
3EY0	X-RAY DIFFRACTION	ATATATATAT	B	0.27715	2.52	0.22193	0.21572
2KNK	SOLUTION NMR	AGGCGCCT	B				
2KNL	SOLUTION NMR	TCCGCGGA	B				
3EIL	X-RAY DIFFRACTION	CGTTAATTAACG	B	0.28471	2.6	0.23679	0.23465
2KAL	SOLUTION NMR	GCGAGATCTGCG	B				
3C2J	X-RAY DIFFRACTION	AACCGGTT	B	0.265	1.78	0.222	0.22
2Z2H	SOLUTION NMR	CTCGGCGCCATC	B				
2GOT	X-RAY DIFFRACTION	GCGAACGC	B	0.269	2.602	0.254	0.253
2OKS	X-RAY DIFFRACTION	CCAACGTTGG	B	0.208	1.65		0.182
2O1I	X-RAY DIFFRACTION	CGGAAATTCGG	B	0.204	1.1		
2B2B	X-RAY DIFFRACTION	CCGCTAGCGG	B	0.26162	1.5	0.21549	0.20995
1X2O	SOLUTION NMR	GACTGTACAGTC	B				
1ZYF	SOLUTION NMR	CAACCATGGTTG	B				
1ZYG	SOLUTION NMR	CAACCCGGGTTG	B				
1ZYH	SOLUTION NMR	CAACCAGGGTTG	B				
1X26	SOLUTION NMR	CTAACAGAATG	B				
2B1D	X-RAY DIFFRACTION	GCAGACGTCTGC	B	0.284	2.5	0.239	0.235
1XCI	SOLUTION NMR	CGAAATTTTCG	B				
1SK5	X-RAY DIFFRACTION	CTTTTAAAAG	B	0.144	0.89	0.12631	0.1263
1ZF7	X-RAY DIFFRACTION	CCGTCGACGG	B	0.288	1.05	0.276	0.276
1ZF3	X-RAY DIFFRACTION	CCGATATCGG	B	0.231	1.84	0.248	0.202
1ZF0	X-RAY DIFFRACTION	CCGTTAACGG	B	0.258	1.5	0.253	0.253
1Y9H	SOLUTION NMR	CCATCGCTACC	B				
1TUQ	SOLUTION NMR	CTCCACGTGGAG	B				
1S9B	X-RAY DIFFRACTION	GAATTCG	B	0.26428	2.81	0.281	0.281
1U6O	SOLUTION NMR	CGGACAAGAAG	B				
1S74	SOLUTION NMR	GTCCACGACG	B				
1S23	X-RAY DIFFRACTION	CGCAATTGCG	B	0.28152	1.6	0.21601	0.20969
1RVI	SOLUTION NMR	CGTTTTAAAACG	B				
1LP7	X-RAY DIFFRACTION	CGTTTATATGCG	B	0.293	2.4		0.229

1ONM	SOLUTION NMR	GCTTCAGTCGT	B				
1N1N	SOLUTION NMR	AGATCAATGT	B				
1OSR	SOLUTION NMR	AGGACCACG	B				
1NEV	SOLUTION NMR	GGCAAAACGG	B				
1N37	SOLUTION NMR	AGACGTCT	B				
1G5K	SOLUTION NMR	CCAAAG	B				
1N0O	SOLUTION NMR	CCAAGG	B				
1N2W	SOLUTION NMR	CGCGAATTGGCG	B				
1MXK	SOLUTION NMR	GGAAGCTTCC	B				
1KVH	SOLUTION NMR	CCCGATGC	B				
1ILC	X-RAY DIFFRACTION	ACCGAATTCGGT	B	0.292	2.2	0.224	0.222
1K9G	X-RAY DIFFRACTION	CCTAGG	B	0.227	1.4	0.209	0.208
1IKK	X-RAY DIFFRACTION	CCTTTAAAGG	B	0.236	1.6	0.184	0.177
1QSX	SOLUTION NMR	CTTTTGCAAAAAG	B				
456D	X-RAY DIFFRACTION	CGCGAATCCGCG	B	0.231	1.6	0.196	0.196
1D8X	X-RAY DIFFRACTION	CCGAATGAGG	B	0.246	1.2	0.188	
1QMS	SOLUTION NMR	GCACCTTCCTGC	B				
476D	X-RAY DIFFRACTION	GCGAATTCGCG	B	0.22	1.3		0.182
1DSM	SOLUTION NMR	GACTAATTGAC	B				
335D	X-RAY DIFFRACTION	GGCAATTGCG	B		2.4	0.203	0.203
334D	X-RAY DIFFRACTION	CATGGCCATG	B		1.8	0.2	0.2
307D	X-RAY DIFFRACTION	CAAAGAAAAG	B		1.85	0.233	
309D	X-RAY DIFFRACTION	CGACGATCGT	B		2.6	0.214	
251D	X-RAY DIFFRACTION	CTCGAG	B		1.9	0.186	0.186
206D	X-RAY DIFFRACTION	CGGTGG	B		2.5	0.221	0.221
249D	X-RAY DIFFRACTION	CGCTCTAGAGCG	B		2.25	0.197	0.197
253D	X-RAY DIFFRACTION	GCGTACGCG	B	0.2137	2.2	0.1931	0.1931
226D	SOLUTION NMR	CGTTTTTACG	B				
1DXA	SOLUTION NMR	GGTCACGAG	B				
218D	X-RAY DIFFRACTION	CGTGAATTCGCG	B		2.25	0.167	0.167
202D	SOLUTION NMR	GACATGTC	B				
107D	SOLUTION NMR	CCTTTTC	B				
194D	X-RAY DIFFRACTION	CGCGTAAACGCG	B		2.3	0.148	0.148
178D	X-RAY DIFFRACTION	CGCAAATTGGCG	B		2.5	0.168	
158D	X-RAY DIFFRACTION	CCAAGCTTGG	B		1.9	0.179	
153D	X-RAY DIFFRACTION	CGAGAATTCGCG	B		2.9	0.169	
119D	X-RAY DIFFRACTION	CGTAGATCTACG	B		2.25	0.138	
1D69	SOLUTION NMR	ATGAGCGAATA	B				
1DA3	X-RAY DIFFRACTION	CGATCGATCG	B		2	0.172	
1D56	X-RAY DIFFRACTION	CGATATATCG	B		1.7	0.178	
1D49	X-RAY DIFFRACTION	CGATTAATCG	B		1.5	0.157	
1D20	SOLUTION NMR	TCTATCACCG	B				
1D18	SOLUTION NMR	CATGCATG	B				
1BDN	X-RAY DIFFRACTION	CGCAAAAATGCG	B		2.6	0.201	
6GN2	X-RAY DIFFRACTION	CCCCGGG	A	0.2723	2.48	0.2601	0.2586
5ZAS	X-RAY DIFFRACTION	CCAGCGCTGG	A	0.1621	1.56	0.14736	0.14666
6DXJ	X-RAY DIFFRACTION	GAGGCCTC	A	0.25535	1.65	0.21607	0.21395
6DY9	X-RAY DIFFRACTION	GGGATCCC	A	0.27582	2.3	0.2545	0.25346
5XK0	X-RAY DIFFRACTION	GCCCCGAGC	A	0.1956	1.451	0.181	0.1803
5MVQ	X-RAY DIFFRACTION	CTACGGCCGTAG	A	0.1938	1.604	0.1795	0.1785
5MVT	X-RAY DIFFRACTION	CTACGTACGTAG	A	0.2245	1.896	0.2031	0.2015
5MVP	X-RAY DIFFRACTION	CTAGGGCCCTAG	A	0.2089	1.606	0.1557	0.153

5JVW	X-RAY DIFFRACTION	AGAGGCCTCT	A	0.26713	2	0.22118	0.21869
5JW0	X-RAY DIFFRACTION	AGGGTACCCT	A	0.28905	2.4	0.26353	0.2619
4IZQ	X-RAY DIFFRACTION	GGGCATGCC	A	0.25227	2.04	0.1997	0.19703
1VT7	X-RAY DIFFRACTION	GGGTGCC	A		2.5	0.152	0.152
1VT5	X-RAY DIFFRACTION	CCCCGGGG	A		2.25	0.24	0.24
1VT9	X-RAY DIFFRACTION	GGGTACCC	A		2.5	0.119	0.119
3IFF	X-RAY DIFFRACTION	GTACGCTAC	A	0.27831	1.75	0.20609	0.20226
2PLO	X-RAY DIFFRACTION	GGTATACC	A		1.4	0.1773	0.17
2B1B	X-RAY DIFFRACTION	GCGTGGGCAC	A	0.255	1.9	0.212	0.212
2B1C	X-RAY DIFFRACTION	GCGTGGGACC	A	0.286	2.2	0.234	0.234
2A7E	X-RAY DIFFRACTION	CCCTAGGG	A	0.197	1.66	0.184	0.183
1ZF6	X-RAY DIFFRACTION	CCCCATGGGG	A	0.306	1.5		0.256
1ZJE	X-RAY DIFFRACTION	AGGGGCGGGCT	A	0.2561	2.1	0.2117	0.2168
1ZF8	X-RAY DIFFRACTION	CCACCGGTGG	A	0.263	1.48	0.22	0.22
1ZF9	X-RAY DIFFRACTION	CCCCCGGGG	A	0.259	1.38	0.237	0.237
1ZFA	X-RAY DIFFRACTION	CCTCCGAGG	A	0.3	1.56	0.241	0.241
1R3Z	X-RAY DIFFRACTION	GCGCGCGC	A	0.203	1.7	0.193	0.17
1M77	X-RAY DIFFRACTION	CCCGATCGGG	A	0.185	1.25		0.163
414D	X-RAY DIFFRACTION	GGGGCGCCCC	A		1.9		
1QPH	X-RAY DIFFRACTION	GACCACGTGGTC	A		2.5	0.225	0.225
399D	X-RAY DIFFRACTION	CGCCCGGGGCG	A	0.203	1.9	0.165	0.165
440D	X-RAY DIFFRACTION	AGGGGCCCT	A	0.251	1.1	0.214	0.213
369D	X-RAY DIFFRACTION	CCCGGGG	A	0.213	1.9	0.176	0.172
343D	X-RAY DIFFRACTION	GCTAGC	A	0.286	2.1	0.204	0.204
327D	X-RAY DIFFRACTION	GCGCGCGCGC	A	0.227	1.94	0.191	0.191
281D	X-RAY DIFFRACTION	GGCATGCC	A		2.38	0.171	0.171
260D	X-RAY DIFFRACTION	GCACGCGTGC	A		1.9	0.186	0.186
232D	X-RAY DIFFRACTION	AGGCATGCCT	A	0.216	1.3	0.1385	
243D	X-RAY DIFFRACTION	ACGTACGT	A		1.9	0.177	0.177
220D	X-RAY DIFFRACTION	ACCCGCGGGT	A		2	0.206	
207D	SOLUTION NMR	TAGCTAGCTA	A				
146D	SOLUTION NMR	TCGCGA	A				
172D	X-RAY DIFFRACTION	GAAGCTTC	A		3	0.212	0.212
138D	X-RAY DIFFRACTION	GCGGGCCCGC	A		1.8	0.183	0.183
1D93	X-RAY DIFFRACTION	CTCTAGAG	A		2.15	0.147	
1D91	X-RAY DIFFRACTION	GGGGTCCC	A		2.1	0.145	
1D92	X-RAY DIFFRACTION	GGGGCTCC	A		2.25	0.136	
116D	X-RAY DIFFRACTION	CCGTACGTACGG	A		2.5	0.15	
1D26	X-RAY DIFFRACTION	GCCCCGGC	A		2.12	0.16	
2D47	X-RAY DIFFRACTION	CCCCCGGGGG	A		2	0.177	
2ANA	X-RAY DIFFRACTION	GGGGCCCC	A		2.5	0.14	

Test Fold 4:

	Experimental Method	Sequence	type	R Free	Resolution (Å)	R Observed	R Work
--	---------------------	----------	------	--------	----------------	------------	--------

5UZD	SOLUTION NMR	GCATCGATTGGC	B				
2N5P	SOLUTION NMR	ATGGAGCTC	B				
5KI5	SOLUTION NMR	TTAGGCCTG	B				
4R6M	X-RAY DIFFRACTION	GGACTTCGCG	B	0.2405	2.357	0.2167	0.2155
2MH6	SOLUTION NMR	CAGTTCCA	B				
4OCD	X-RAY DIFFRACTION	AAAATTTT	B	0.25103	2.1	0.23682	0.23614
2RT8	SOLUTION NMR	GCGTTGTCC	B				
2KY7	SOLUTION NMR	AACAATTGTT	B				
3IGT	X-RAY DIFFRACTION	CCGAGTCCTA	B	0.265	1.9	0.224	0.224
2GE2	SOLUTION NMR	CGTACGCATGC	B				
1SY8	SOLUTION NMR	TGATCA	B				
1PQQ	SOLUTION NMR	CGCTAACAGGC	B				
1RVH	SOLUTION NMR	GCAAAATTTTGC	B				
1HQ7	X-RAY DIFFRACTION	GCAAACGTTTGC	B	0.266	2.1	0.237	0.237
1N4E	X-RAY DIFFRACTION	GCTTAATTCG	B	0.255	2.5		0.197
1ENN	X-RAY DIFFRACTION	GCGAATTCG	B	0.161	0.89	0.135	0.135
1CVY	X-RAY DIFFRACTION	CCAGATCTGG	B	0.229	2.15	0.229	0.229
424D	X-RAY DIFFRACTION	ACCAGCGTCGGT	B	0.283	2.7	0.211	0.211
477D	X-RAY DIFFRACTION	GCGAATTCGCG	B	0.235	1.7	0.194	0.194
175D	SOLUTION NMR	GCGAATGAGC	B				
150D	X-RAY DIFFRACTION	GCGAATTAGCG	B		2.25	0.182	0.182
132D	SOLUTION NMR	GCCGTTAACGGC	B				
1D89	X-RAY DIFFRACTION	CGCGAAAAACG	B		2.3	0.232	0.232
1D83	SOLUTION NMR	AAGGCCTT	B				
1DN9	X-RAY DIFFRACTION	CGCATATATGCG	B		2.2	0.189	0.189
3DNB	X-RAY DIFFRACTION	CCAAGATTGG	B		1.3	0.164	
6DY5	X-RAY DIFFRACTION	AGGGATCCCT	A	0.25374	1.26	0.21649	0.2143
5WSS	X-RAY DIFFRACTION	GGTCGTCC	A	0.1934	1.45	0.15017	0.14799
4YS5	X-RAY DIFFRACTION	GTGGCCAC	A	0.26294	1.65	0.22695	0.22517
4F4N	X-RAY DIFFRACTION	GTGTACAC	A	0.18111	1.3	0.15543	0.15416
1ZF1	X-RAY DIFFRACTION	CCGGGCCCGG	A	0.245	1.35	0.222	0.222
382D	X-RAY DIFFRACTION	CCGCCGGCGG	A		2.2	0.197	0.197
401D	X-RAY DIFFRACTION	GACCGCGGTC	A	0.285	2.2	0.219	0.219
257D	X-RAY DIFFRACTION	GCCGGC	A	0.161	2.3	0.164	0.164
254D	X-RAY DIFFRACTION	GCGCGC	A	0.186	1.9	0.197	0.197
197D	X-RAY DIFFRACTION	GTACGTAC	A		2.19	0.161	
117D	X-RAY DIFFRACTION	GCGTACGTACGC	A		2.55	0.142	
118D	X-RAY DIFFRACTION	GTGCGCAC	A		1.64	0.154	

Train Fold 5:

	Experimental Method	Sequence	type	R Free	Resolution (Å)	R Observed	R Work
6RSO	X-RAY DIFFRACTION	TCGGCGCCGA	B	0.2474	1.97	0.209	0.2068
6QJR	X-RAY DIFFRACTION	CGCAAAAAGCG	B	0.2435	2.9	0.19145	0.18814
6F3C	X-RAY DIFFRACTION	CGTACG	B	0.2923	2.3	0.261	0.2573
6GIM	X-RAY DIFFRACTION	AAATTT	B	0.19286	1.43	0.14585	0.14342
5M68	X-RAY DIFFRACTION	CGAATTAATTCG	B	0.2469	2.64	0.22824	0.22731
5GUN	X-RAY DIFFRACTION	GTGGAATGGAAC	B	0.2875	2.588	0.2449	0.2429
5UZD	SOLUTION NMR	GCATCGATTGGC	B				

5J3F	SOLUTION NMR	CGGCCGCCGA	B				
2N5P	SOLUTION NMR	ATGGAGCTC	B				
2N9H	SOLUTION NMR	GATGACTGCTAG	B				
2N9F	SOLUTION NMR	CTAGCGGTCATC	B				
5K14	SOLUTION NMR	ATCCGGTAG	B				
5K15	SOLUTION NMR	TTAGGCCTG	B				
4R6M	X-RAY DIFFRACTION	GGACTTCGCG	B	0.2405	2.357	0.2167	0.2155
2MNE	SOLUTION NMR	CGACTAGTCG	B				
2MH6	SOLUTION NMR	CAGTTCCA	B				
4OCD	X-RAY DIFFRACTION	AAAATTTT	B	0.25103	2.1	0.23682	0.23614
2RT8	SOLUTION NMR	CGCGTTGTCC	B				
4J2I	X-RAY DIFFRACTION	AATAAATTTATT	B	0.28729	2.98	0.27031	0.26963
2MCI	SOLUTION NMR	GTCGGCTG	B				
4BZT	SOLUTION NMR	ATGCAT	B				
2LWH	SOLUTION NMR	GGATATATCC	B				
3TOK	X-RAY DIFFRACTION	CCGATACCGG	B	0.2913	1.74	0.22042	0.2339
2LIB	SOLUTION NMR	GTCCAGGACG	B				
2LG3	SOLUTION NMR	GCTAGCGAGTCC	B				
4E1U	X-RAY DIFFRACTION	CGGAAATTACCG	B	0.149	0.92	0.1401	0.1397
2LGM	SOLUTION NMR	GCATGTGTACG	B				
1VTJ	X-RAY DIFFRACTION	CGCGATATCGCG	B		2.4	0.202	
3L1Q	X-RAY DIFFRACTION	TGGCCTTAAGG	B		2.5	0.22325	0.22325
3OMJ	X-RAY DIFFRACTION	CCAGTACTGG	B	0.1237	0.95	0.1127	0.1121
3N4N	X-RAY DIFFRACTION	CGCGAA	B	0.263	1.92		0.258
2KY7	SOLUTION NMR	AACAATTGTT	B				
2KH5	SOLUTION NMR	GTGCGTGTGTTGT	B				
3FT6	X-RAY DIFFRACTION	CGATCG	B	0.21929	1.12	0.18811	0.18433
3EY0	X-RAY DIFFRACTION	ATATATATAT	B	0.27715	2.52	0.22193	0.21572
2KNK	SOLUTION NMR	AGGCGCCT	B				
2KNL	SOLUTION NMR	TCCGCGGA	B				
3IGT	X-RAY DIFFRACTION	CCGAGTCCTA	B	0.265	1.9	0.224	0.224
3EIL	X-RAY DIFFRACTION	CGTTAATTAACG	B	0.28471	2.6	0.23679	0.23465
2KAL	SOLUTION NMR	GCGAGATCTGCG	B				
2Z2H	SOLUTION NMR	CTCGGCGCCATC	B				
2GOT	X-RAY DIFFRACTION	GCGAACGC	B	0.269	2.602	0.254	0.253
2OKS	X-RAY DIFFRACTION	CCAACGTTGG	B	0.208	1.65		0.182
2O1I	X-RAY DIFFRACTION	CGGAAATTCCCG	B	0.204	1.1		
1X2O	SOLUTION NMR	GACTGTACAGTC	B				
2GE2	SOLUTION NMR	CGTACGCATGC	B				
1ZYG	SOLUTION NMR	CAACCCGGGTTG	B				
1ZYH	SOLUTION NMR	CAACCAGGGTTG	B				
1X26	SOLUTION NMR	CTAACAGAATG	B				
2B1D	X-RAY DIFFRACTION	GCAGACGTCTGC	B	0.284	2.5	0.239	0.235
1XC1	SOLUTION NMR	CGAAATTTTCG	B				
1SK5	X-RAY DIFFRACTION	CTTTTAAAAG	B	0.144	0.89	0.12631	0.1263
1ZF7	X-RAY DIFFRACTION	CCGTCGACGG	B	0.288	1.05	0.276	0.276
1ZF3	X-RAY DIFFRACTION	CCGATATCGG	B	0.231	1.84	0.248	0.202
1Y9H	SOLUTION NMR	CCATCGCTACC	B				
1SY8	SOLUTION NMR	TGATCA	B				
1TUQ	SOLUTION NMR	CTCCACGTGGAG	B				
1S9B	X-RAY DIFFRACTION	GAATTCG	B	0.26428	2.81	0.281	0.281
1U6O	SOLUTION NMR	CGGACAAGAAG	B				

1PQQ	SOLUTION NMR	CGCTAACAGGC	B				
1S23	X-RAY DIFFRACTION	CGCAATTGCG	B	0.28152	1.6	0.21601	0.20969
1RVI	SOLUTION NMR	CGTTTTAAAACG	B				
1RVH	SOLUTION NMR	GCAAAATTTTGC	B				
1LP7	X-RAY DIFFRACTION	CGCTTATATGCG	B	0.293	2.4		0.229
1ONM	SOLUTION NMR	GCTTCAGTCGT	B				
1N1N	SOLUTION NMR	AGATCAATGT	B				
1OSR	SOLUTION NMR	AGGACCACG	B				
1HQ7	X-RAY DIFFRACTION	GCAAACGTTTGC	B	0.266	2.1	0.237	0.237
1NEV	SOLUTION NMR	GGCAAAACGG	B				
1N4E	X-RAY DIFFRACTION	GCTTAATTCG	B	0.255	2.5		0.197
1N0O	SOLUTION NMR	CCAAGG	B				
1N2W	SOLUTION NMR	CGCGAATTGGCG	B				
1KVH	SOLUTION NMR	CCCGATGC	B				
1K9G	X-RAY DIFFRACTION	CCTAGG	B	0.227	1.4	0.209	0.208
1ENN	X-RAY DIFFRACTION	GCGAATTCG	B	0.161	0.89	0.135	0.135
1CVY	X-RAY DIFFRACTION	CCAGATCTGG	B	0.229	2.15	0.229	0.229
456D	X-RAY DIFFRACTION	CGCGAATCCGCG	B	0.231	1.6	0.196	0.196
1D8X	X-RAY DIFFRACTION	CCGAATGAGG	B	0.246	1.2	0.188	
424D	X-RAY DIFFRACTION	ACCGACGTCGGT	B	0.283	2.7	0.211	0.211
1QMS	SOLUTION NMR	GCACCTTCTGC	B				
477D	X-RAY DIFFRACTION	GGCGAATTCGCG	B	0.235	1.7	0.194	0.194
1DSM	SOLUTION NMR	GACTAATTGAC	B				
335D	X-RAY DIFFRACTION	GGCAATTGCG	B		2.4	0.203	0.203
334D	X-RAY DIFFRACTION	CATGGCCATG	B		1.8	0.2	0.2
309D	X-RAY DIFFRACTION	CGACGATCGT	B		2.6	0.214	
251D	X-RAY DIFFRACTION	CTCGAG	B		1.9	0.186	0.186
249D	X-RAY DIFFRACTION	CGCTCTAGAGCG	B		2.25	0.197	0.197
253D	X-RAY DIFFRACTION	GCGTACGCG	B	0.2137	2.2	0.1931	0.1931
218D	X-RAY DIFFRACTION	CGTGAATTCGCG	B		2.25	0.167	0.167
202D	SOLUTION NMR	GACATGTC	B				
107D	SOLUTION NMR	CCTTTTC	B				
194D	X-RAY DIFFRACTION	CGCGTAAACGCG	B		2.3	0.148	0.148
178D	X-RAY DIFFRACTION	CGCAAATTGGCG	B		2.5	0.168	
175D	SOLUTION NMR	GCGAATGAGC	B				
150D	X-RAY DIFFRACTION	CGCGAATTAGCG	B		2.25	0.182	0.182
132D	SOLUTION NMR	GCCGTAAACGGC	B				
119D	X-RAY DIFFRACTION	CGTAGATCTACG	B		2.25	0.138	
1D89	X-RAY DIFFRACTION	CGCGAAAAAACG	B		2.3	0.232	0.232
1D83	SOLUTION NMR	AAGGCCTT	B				
1D56	X-RAY DIFFRACTION	CGATATATCG	B		1.7	0.178	
1D49	X-RAY DIFFRACTION	CGATTAATCG	B		1.5	0.157	
1D20	SOLUTION NMR	TCTATCACCG	B				
1BDN	X-RAY DIFFRACTION	CGCAAAAATGCG	B		2.6	0.201	
1DN9	X-RAY DIFFRACTION	CGCATATATGCG	B		2.2	0.189	0.189
3DNB	X-RAY DIFFRACTION	CCAAGATTGG	B		1.3	0.164	
6GN2	X-RAY DIFFRACTION	CCCGGG	A	0.2723	2.48	0.2601	0.2586
6DXJ	X-RAY DIFFRACTION	GAGGCCTC	A	0.25535	1.65	0.21607	0.21395
6DY5	X-RAY DIFFRACTION	AGGGATCCCT	A	0.25374	1.26	0.21649	0.2143
6DY9	X-RAY DIFFRACTION	GGGATCCC	A	0.27582	2.3	0.2545	0.25346
5XK0	X-RAY DIFFRACTION	GCCCCGAGC	A	0.1956	1.451	0.181	0.1803
5MVT	X-RAY DIFFRACTION	CTACGTACGTAG	A	0.2245	1.896	0.2031	0.2015

5MVP	X-RAY DIFFRACTION	CTAGGGCCCTAG	A	0.2089	1.606	0.1557	0.153
5WSS	X-RAY DIFFRACTION	GGTCGTCC	A	0.1934	1.45	0.15017	0.14799
5JW0	X-RAY DIFFRACTION	AGGGTACCCT	A	0.28905	2.4	0.26353	0.2619
4YS5	X-RAY DIFFRACTION	GTGGCCAC	A	0.26294	1.65	0.22695	0.22517
4F4N	X-RAY DIFFRACTION	GTGTACAC	A	0.18111	1.3	0.15543	0.15416
1VT7	X-RAY DIFFRACTION	GGGTGCC	A		2.5	0.152	0.152
1VT5	X-RAY DIFFRACTION	CCCCGGGG	A		2.25	0.24	0.24
1VT9	X-RAY DIFFRACTION	GGGTACCC	A		2.5	0.119	0.119
3IFF	X-RAY DIFFRACTION	GTACGCGTAC	A	0.27831	1.75	0.20609	0.20226
2PLO	X-RAY DIFFRACTION	GGTATACC	A		1.4	0.1773	0.17
2B1B	X-RAY DIFFRACTION	GCGTGGGCAC	A	0.255	1.9	0.212	0.212
2A7E	X-RAY DIFFRACTION	CCCTAGGG	A	0.197	1.66	0.184	0.183
1ZF6	X-RAY DIFFRACTION	CCCCATGGGG	A	0.306	1.5		0.256
1ZJE	X-RAY DIFFRACTION	AGGGGCGGGGCT	A	0.2561	2.1	0.2117	0.2168
1ZF8	X-RAY DIFFRACTION	CCACCGGTGG	A	0.263	1.48	0.22	0.22
1ZF9	X-RAY DIFFRACTION	CCCCCGGGGG	A	0.259	1.38	0.237	0.237
1ZF1	X-RAY DIFFRACTION	CCGGGCCCGG	A	0.245	1.35	0.222	0.222
1R3Z	X-RAY DIFFRACTION	GCGCGCGC	A	0.203	1.7	0.193	0.17
1M77	X-RAY DIFFRACTION	CCCGATCGGG	A	0.185	1.25		0.163
382D	X-RAY DIFFRACTION	CCGCCGCGG	A		2.2	0.197	0.197
414D	X-RAY DIFFRACTION	GGGGCGCCCC	A		1.9		
1QPH	X-RAY DIFFRACTION	GACCACGTGGTC	A		2.5	0.225	0.225
399D	X-RAY DIFFRACTION	CGCCCGGGGCG	A	0.203	1.9	0.165	0.165
440D	X-RAY DIFFRACTION	AGGGGCCCT	A	0.251	1.1	0.214	0.213
369D	X-RAY DIFFRACTION	CCCGCGGG	A	0.213	1.9	0.176	0.172
401D	X-RAY DIFFRACTION	GACCGGGTC	A	0.285	2.2	0.219	0.219
327D	X-RAY DIFFRACTION	GCGCGCGCGC	A	0.227	1.94	0.191	0.191
281D	X-RAY DIFFRACTION	GGCATGCC	A		2.38	0.171	0.171
257D	X-RAY DIFFRACTION	GCCGGC	A	0.161	2.3	0.164	0.164
254D	X-RAY DIFFRACTION	GCGCGC	A	0.186	1.9	0.197	0.197
232D	X-RAY DIFFRACTION	AGGCATGCCT	A	0.216	1.3	0.1385	
243D	X-RAY DIFFRACTION	ACGTACGT	A		1.9	0.177	0.177
207D	SOLUTION NMR	TAGCTAGCTA	A				
146D	SOLUTION NMR	TCGCGA	A				
197D	X-RAY DIFFRACTION	GTACGTAC	A		2.19	0.161	
1D93	X-RAY DIFFRACTION	CTCTAGAG	A		2.15	0.147	
1D92	X-RAY DIFFRACTION	GGGGCTCC	A		2.25	0.136	
116D	X-RAY DIFFRACTION	CCGTACGTACGG	A		2.5	0.15	
117D	X-RAY DIFFRACTION	GCGTACGTACGC	A		2.55	0.142	
118D	X-RAY DIFFRACTION	GTGCGCAC	A		1.64	0.154	
1D26	X-RAY DIFFRACTION	GCCCCGGC	A		2.12	0.16	
2D47	X-RAY DIFFRACTION	CCCCCGGGGG	A		2	0.177	
2ANA	X-RAY DIFFRACTION	GGGGCCCC	A		2.5	0.14	

Test Fold 5:

	Experimental Method	Sequence	type	R Free	Resolution (Å)	R Observed	R Work
6ASF	SOLUTION NMR	CCAAGATAG	B				
5UZF	SOLUTION NMR	CGATTTTTTGCC	B				

2LZW	SOLUTION NMR	CGAAAGTTTCG	B				
4BZU	SOLUTION NMR	TATGCATA	B				
3LPV	X-RAY DIFFRACTION	CCTCTGGTCTCC	B	0.19788	1.77	0.17342	0.17212
3C2J	X-RAY DIFFRACTION	AACCCGTT	B	0.265	1.78	0.222	0.22
2B2B	X-RAY DIFFRACTION	CCGCTAGCGG	B	0.26162	1.5	0.21549	0.20995
1ZYF	SOLUTION NMR	CAACCATGGTTG	B				
1ZF0	X-RAY DIFFRACTION	CCGTAAACGG	B	0.258	1.5	0.253	0.253
1S74	SOLUTION NMR	GTCCACGACG	B				
1N37	SOLUTION NMR	AGACGTCT	B				
1G5K	SOLUTION NMR	CCAAAG	B				
1MXK	SOLUTION NMR	GGAAGCTTCC	B				
1ILC	X-RAY DIFFRACTION	ACCGAATTCGGT	B	0.292	2.2	0.224	0.222
1IKK	X-RAY DIFFRACTION	CCTTTAAAGG	B	0.236	1.6	0.184	0.177
1QSX	SOLUTION NMR	CTTTTGCAAAAAG	B				
476D	X-RAY DIFFRACTION	GCGAATTCGCG	B	0.22	1.3		0.182
307D	X-RAY DIFFRACTION	CAAAGAAAAG	B		1.85	0.233	
206D	X-RAY DIFFRACTION	CGGTGG	B		2.5	0.221	0.221
226D	SOLUTION NMR	CGTTTTTACG	B				
1DXA	SOLUTION NMR	GGTCACGAG	B				
158D	X-RAY DIFFRACTION	CCAAGCTTGG	B		1.9	0.179	
153D	X-RAY DIFFRACTION	CGAGAATTCGCG	B		2.9	0.169	
1D69	SOLUTION NMR	ATGAGCGAATA	B				
1DA3	X-RAY DIFFRACTION	CGATCGATCG	B		2	0.172	
1D18	SOLUTION NMR	CATGCATG	B				
5ZAS	X-RAY DIFFRACTION	CCAGCGCTGG	A	0.1621	1.56	0.14736	0.14666
5MVQ	X-RAY DIFFRACTION	CTACGGCCGTAG	A	0.1938	1.604	0.1795	0.1785
5JVW	X-RAY DIFFRACTION	AGAGGCCTCT	A	0.26713	2	0.22118	0.21869
4IZQ	X-RAY DIFFRACTION	GGGCATGCC	A	0.25227	2.04	0.1997	0.19703
2B1C	X-RAY DIFFRACTION	GCGTGGGACC	A	0.286	2.2	0.234	0.234
1ZF0	X-RAY DIFFRACTION	CCTCCGGAGG	A	0.3	1.56	0.241	0.241
343D	X-RAY DIFFRACTION	GCTAGC	A	0.286	2.1	0.204	0.204
260D	X-RAY DIFFRACTION	GCACGCGTGC	A		1.9	0.186	0.186
220D	X-RAY DIFFRACTION	ACCCGCGGGT	A		2	0.206	
172D	X-RAY DIFFRACTION	GAAGCTTC	A		3	0.212	0.212
138D	X-RAY DIFFRACTION	GCGGGCCCGC	A		1.8	0.183	0.183
1D91	X-RAY DIFFRACTION	GGGGTCCC	A		2.1	0.145	

SECTION G:

Repeated Stratified(k=2) nested cross validation(k=5)

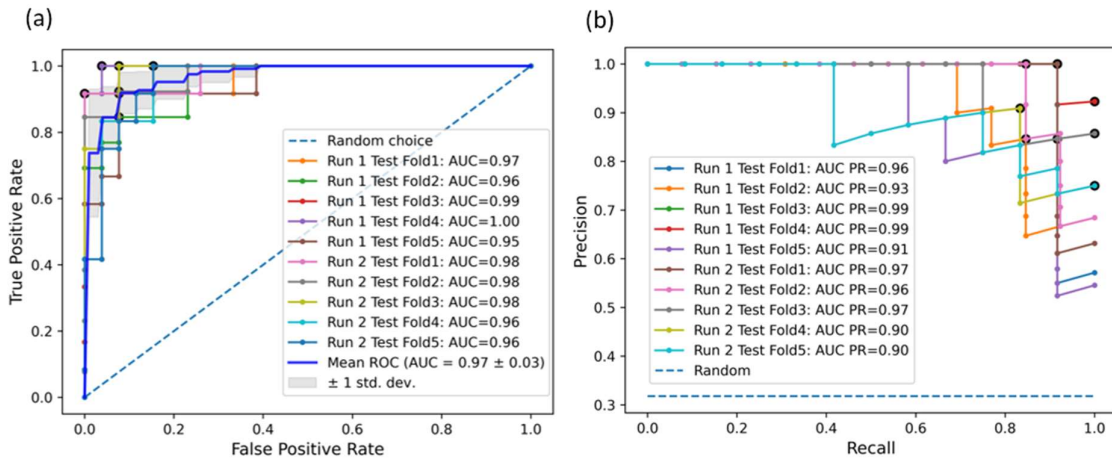


Figure S9: Results obtained for Repeated k-fold(k=2) Nested Stratified five-fold cross-validation (a) ROC-AUC curves of tuned LightGBM model (b) Precision-Recall curves of tuned LightGBM model

Table S6:

(a) Results obtained when using SMOTE+TOMEK for adjusting class imbalance

	Average PR	AUC PR	ROC AUC	Accuracy	F1	MCC	cohen_kappa_score
0	0.956313	0.954461	0.978395	0.923077	0.869565	0.816717	0.815166
1	0.907577	0.902992	0.955621	0.871795	0.814815	0.718132	0.716981
2	0.922562	0.919515	0.958333	0.868421	0.814815	0.725421	0.714715

	Average PR	AUC PR	ROC AUC	Accuracy	F1	MCC	cohen_kappa_score
3	0.981151	0.980377	0.990385	0.921053	0.880000	0.822777	0.821317
4	0.941178	0.938529	0.971154	0.894737	0.833333	0.756410	0.756410
Mean	0.941756	0.939175	0.970778	0.895816	0.842506	0.767892	0.764918

(b) Results obtained when using class weiging(scale_pos_weight) for adjusting class imbalance

	Average PR	AUC PR	ROC AUC	Accuracy	F1	MCC	cohen_kappa_score
0	0.957633	0.956180	0.975309	0.923077	0.857143	0.821584	0.805970
1	0.930681	0.927855	0.961538	0.897436	0.833333	0.765532	0.760000
2	0.986645	0.986063	0.993590	0.947368	0.916667	0.878205	0.878205
3	0.979070	0.978108	0.990385	0.947368	0.916667	0.878205	0.878205
4	0.959366	0.957532	0.980769	0.921053	0.880000	0.822777	0.821317
Mean	0.962679	0.961147	0.980318	0.927260	0.880762	0.833261	0.828739

References:

- (1) Read, R. J.; Adams, P. D.; Arendall III, W. B.; Brunger, A. T.; Emsley, P.; Joosten, R. P.; Kleywegt, G. J.; Krissinel, E. B.; Lütke, T.; Otwinowski, Z. A New Generation of Crystallographic Validation Tools for the Protein Data Bank. *Structure* **2011**, *19* (10), 1395–1412.
- (2) Hubert, M.; Vandervieren, E. An Adjusted Boxplot for Skewed Distributions. *Computational statistics & data analysis* **2008**, *52* (12), 5186–5201.

- (3) Baldi, P.; Brunak, S.; Chauvin, Y.; Andersen, C. A. F.; Nielsen, H. Assessing the Accuracy of Prediction Algorithms for Classification: An Overview. *Bioinformatics* **2000**, *16* (5), 412–424. <https://doi.org/10.1093/bioinformatics/16.5.412>.
- (4) Kulkarni, M.; Mukherjee, A. Computational Approach to Explore the B/A Junction Free Energy in DNA. *ChemPhysChem* **2016**, *17* (1), 147–154. <https://doi.org/10.1002/cphc.201500690>.
- (5) Jacobo-Molina, A.; Ding, J.; Nanni, R. G.; Clark, A. D.; Lu, X.; Tantillo, C.; Williams, R. L.; Kamer, G.; Ferris, A. L.; Clark, P.; Hizi, A.; Hughes, S. H.; Arnold, E. Crystal Structure of Human Immunodeficiency Virus Type 1 Reverse Transcriptase Complexed with Double-Stranded DNA at 3.0 Å Resolution Shows Bent DNA. *Proceedings of the National Academy of Sciences of the United States of America* **1993**, *90* (13), 6320–6324.
- (6) Berman, H. M.; Olson, W. K.; Beveridge, D. L.; Westbrook, J.; Gelbin, A.; Demeny, T.; Hsieh, S. H.; Srinivasan, A. R.; Schneider, B. The Nucleic Acid Database. A Comprehensive Relational Database of Three-Dimensional Structures of Nucleic Acids. *Biophys J* **1992**, *63* (3), 751–759.
- (7) Basham, B.; Schroth, G. P.; Ho, P. S. An A-DNA Triplet Code: Thermodynamic Rules for Predicting A- and B-DNA. *Proceedings of the National Academy of Sciences of the United States of America* **1995**, *92* (14), 6464–6468.
- (8) Coimbatore Narayanan, B.; Westbrook, J.; Ghosh, S.; Petrov, A. I.; Sweeney, B.; Zirbel, C. L.; Leontis, N. B.; Berman, H. M. The Nucleic Acid Database: New Features and Capabilities. *Nucleic Acids Res* **2014**, *42* (D1), D114–D122. <https://doi.org/10.1093/nar/gkt980>.

Appendix 2

Section A

Hierarchical clustering for partitioning molecular surface

A common interpretation made of hierarchical clustering^{1,2} is to derive a partition. In the initial stage, all points on the molecular surface are in their respective clusters. Next, the two clusters separated by shortest “distance” are combined. We used complete linkage distance³ criteria, where the distance between clusters equals the distance between those two elements (one in each cluster) that are farthest away from each other.

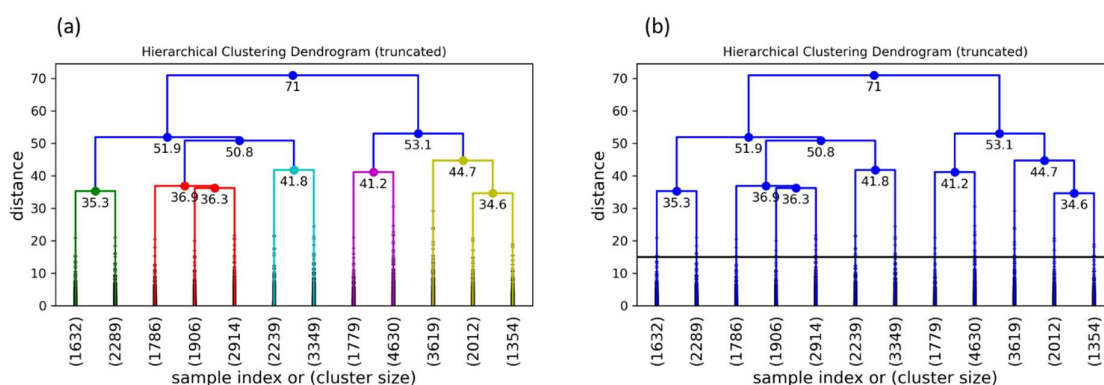


Figure S1: Illustration of Complete linkage clustering on the molecular surface of 2HNP protein with a threshold of 15. (a) Truncated clustering dendrogram for clustering procedure. (b) Once the threshold criterion is met, the clustering process stops, and we have partitions as the number of clusters.

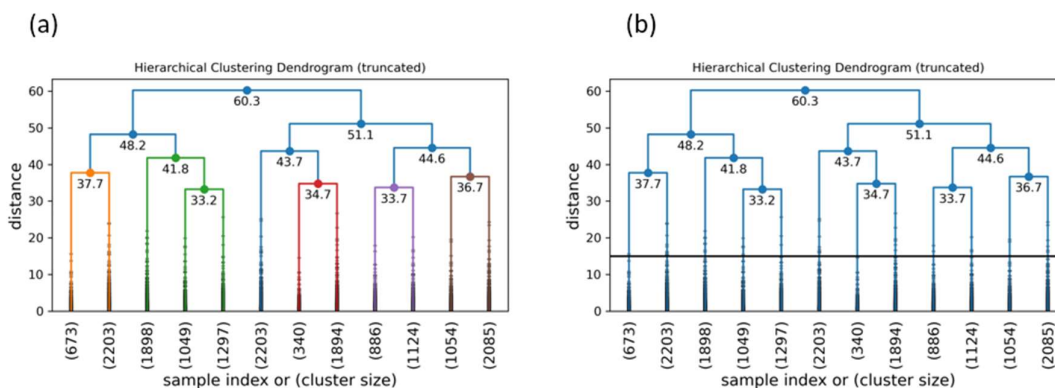


Figure S2: Illustration of Complete linkage clustering on the molecular surface of 1A42 protein with a threshold chosen as 15. (a) Truncated clustering dendrogram for clustering procedure. (b) Once the threshold criterion is met, the clustering process stops, and we have partitions as the number of clusters at that point.

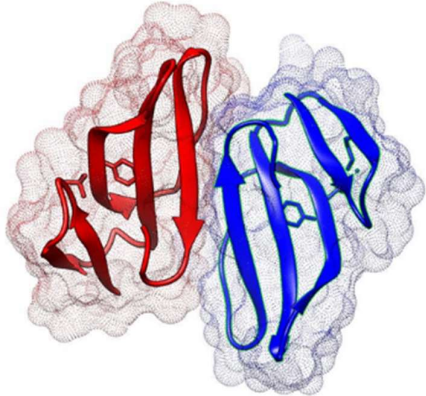
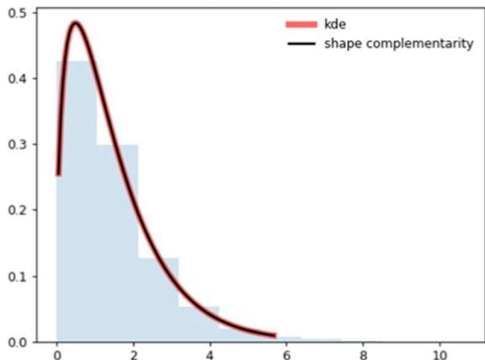
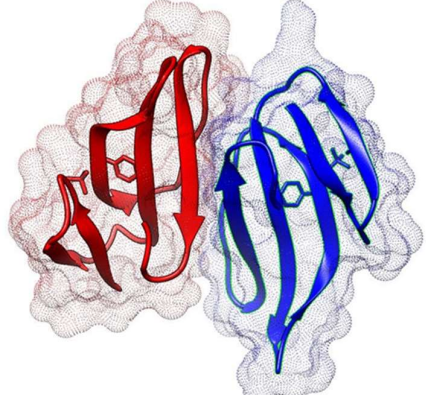
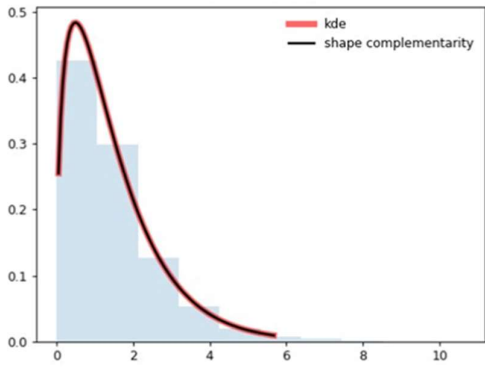
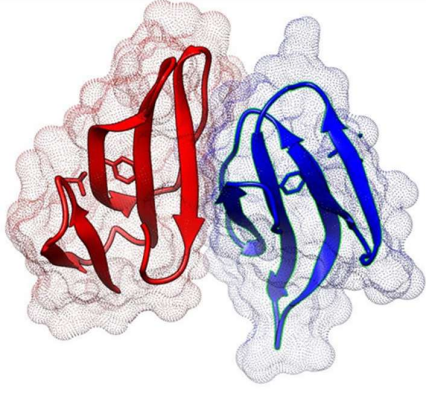
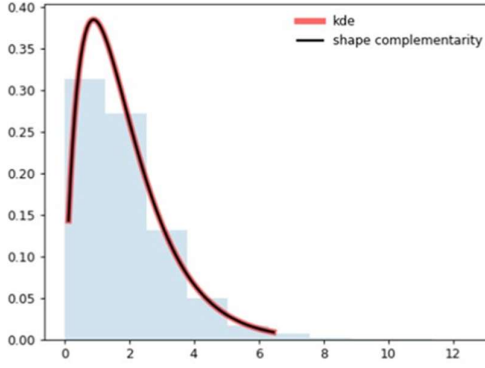
Section B

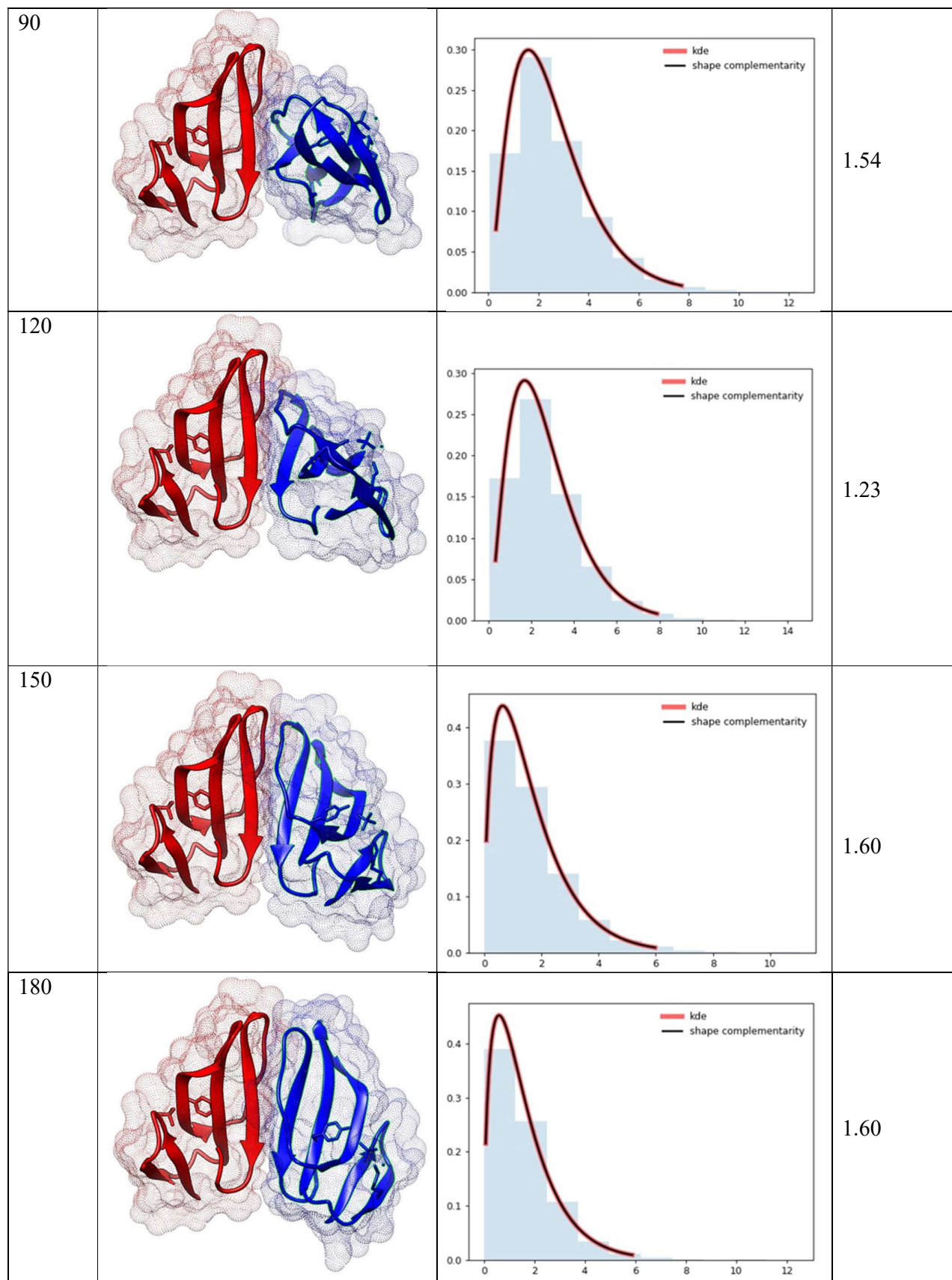
Quantifying shape complementarity

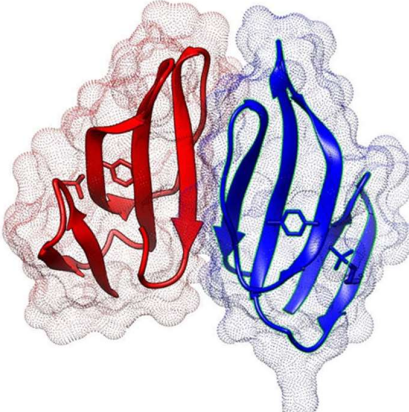
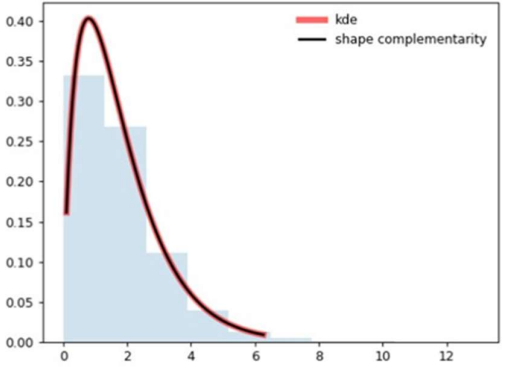
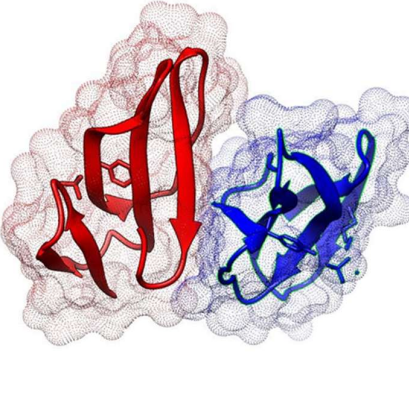
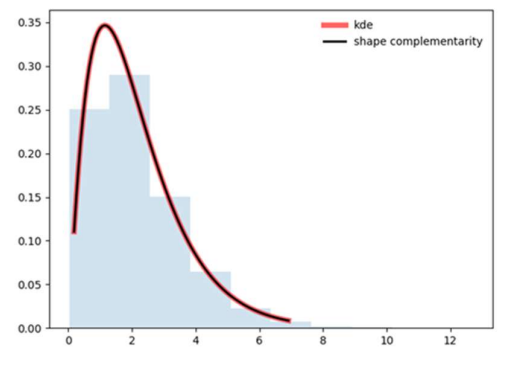
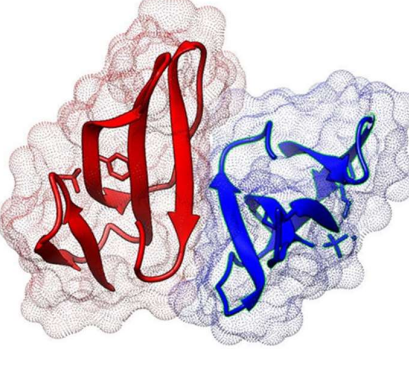
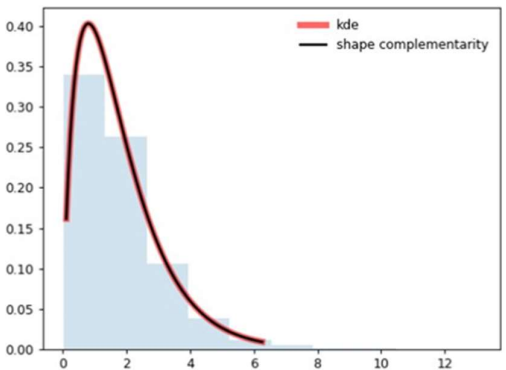
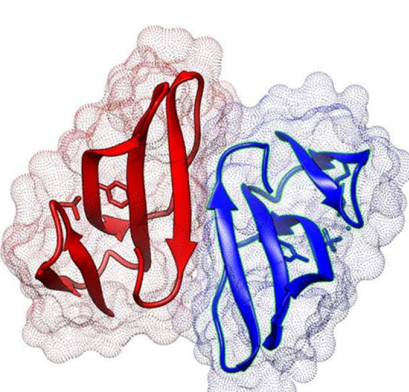
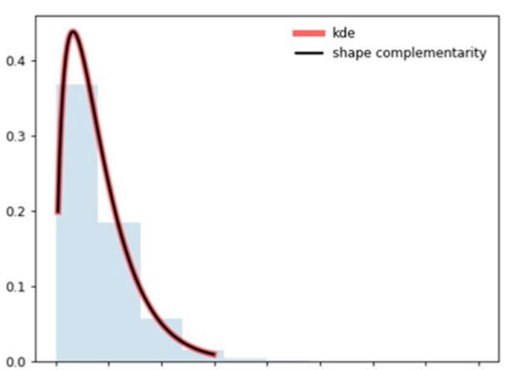
Shape Complementarity measured in terms of skewness for a homodimer - 1CDT

We fixed the chain A of the homodimer 1CDT and then rotated the chain B at several angles along Z-axis, each separated by 30 degrees. The 0 degree represents the starting orientation of the crystal structure. Our measure of shape complementarity can distinguish these pairs.

Table S1:

Degree	Structure	Crvature Distribution	Skewness
0			1.67
30			1.47
60			1.54



210			1.52
270			1.37
300			1.52
330			1.60

Comparison with experimental data:

We present below two sets of PDBs, which represent different dimeric systems (protein-protein Interaction) retrieved from PDBbind database 2019 version. The top panel figure (a,b) represents the more strongly binding complex 1AVX ($pK_d = 13.22$), and the below panel represents a comparatively less strongly binding complex 1A22 ($pK_d = 9.47$). Based solely on geometric shape complementarity using surface curvature as the measure, our approach can distinguish them at the Interfacial region.

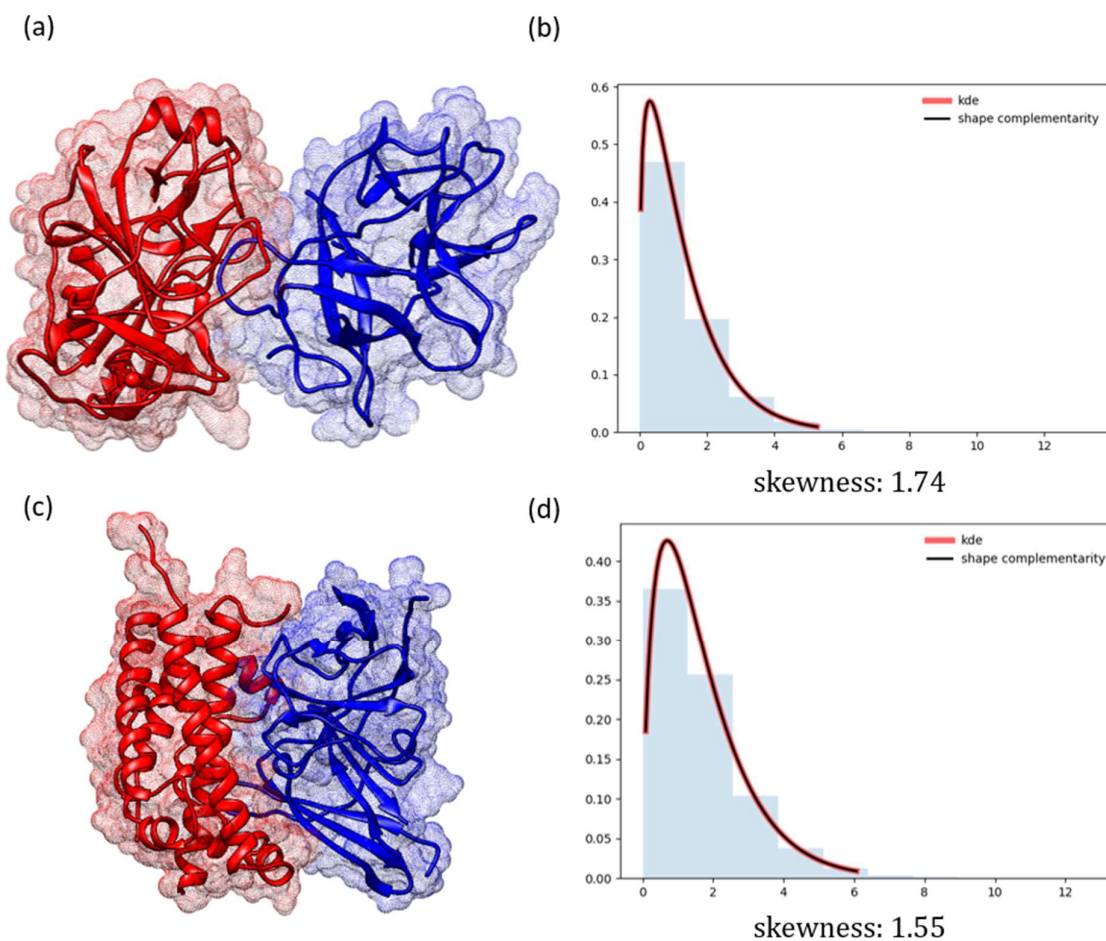


Figure S3: (a,b) 1AVX with ($pK_d = 13.22$), a Kunitz-type soybean trypsin inhibitor complex with porcine trypsin. (c,d) 1A22 with ($pK_d = 9.47$), human growth hormone bound to a single receptor

References

- (1) Murtagh, F.; Contreras, P. Algorithms for Hierarchical Clustering: An Overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **2012**, 2 (1), 86–97.
- (2) Dawyndt, P.; De Meyer, H.; De Baets, B. The Complete Linkage Clustering Algorithm Revisited. *Soft Computing* **2005**, 9 (5), 385–392.
- (3) Saraçlı, S.; Doğan, N.; Doğan, İ. Comparison of Hierarchical Cluster Analysis Methods by Cophenetic Correlation. *J Inequal Appl* **2013**, 2013 (1), 203.
<https://doi.org/10.1186/1029-242X-2013-203>.

List of Publications

- 1) Gupta et al., Accurate prediction of B-form/A-form DNA conformation propensity from primary sequence: A machine learning and free energy handshake, *Patterns* (2021), <https://doi.org/10.1016/j.patter.2021.100329>
- 2) Gupta, Abhijit and Mukherjee, Arnab, Capturing Surface Complementarity in Proteins Using Unsupervised Learning and Robust Curvature Measure. Available at SSRN: <https://ssrn.com/abstract=3784950> or <http://dx.doi.org/10.2139/ssrn.3784950>