THE UNIVERSITY
of EDINBURGH

Applications of Artificial Intelligence to Alchemical Free Energy Calculations in Contemporary Drug Design

Jenke Scheen

A thesis presented for the degree of

Doctor of Philosophy

EaStCHEM School of Chemistry

University of Edinburgh

United Kingdom

April 3, 2023

Declaration of sole authorship

I hereby declare that I, Jenke Scheen, have composed this thesis, "*Applications of Artificial Intelligence to Alchemical Free Energy Calculations in Contemporary Drug Design*" of my own and that the research presented in this work is of my making. I confirm that:

- the thesis has been composed by me while in candidature for a doctorate degree at the University of Edinburgh.
- the research entities presented in this thesis are my own and that where collaborative efforts were involved this has been clearly stated.
- the work has not been submitted for any other degree or professional qualification except as specified.
- where the published work of peers has been consulted to support the current work this has been clearly stated.
- where I have quoted from the work of others, the source is always given and applicable software licenses are referred to. With the exception of such quotations, this thesis is entirely my own work.

April 3, 2023

....

Jenke Scheen Date

Abstract

Applications of Artificial Intelligence to Alchemical Free Energy Calculations in Contemporary Drug Design

The work presented in this thesis resides at the interface of alchemical free energy methods (AFE) and machine-learning (ML) in the context of computer-aided drug discovery (CADD). The majority of the work consists of explorations into regions of synergy between the individual parts. The overarching hypothesis behind this work is that although areas of high potential exist for standalone ML and AFE in CADD, an additional source of value can be found in areas where ML and AFE are combined in such a way that the new methodology profits from key strengths in either part.

Physics-based AFE calculations have - over several decades - grown into precise and accurate sub-kcal·mol^{-1} (in terms of mean absolute error versus experimental measures) methods of predicting ligand-protein binding affinities which is the main driver of its popularity in project support in drug design workflows. Data-driven ML methods have seen a similar rapid development spurred by the exponential growth in computational hardware capabilities, but are generally still lacking in accuracy versus experimental measures of binding affinities to support drug design work. Contrastingly, however, the first relies mainly on physical rules in the form of statistical mechanics and the latter profits from interpolating signals within large training domains of data.

After a historical and theoretical introduction into drug discovery, AFE calculations and ML methods, the thesis will highlight several studies that reflect the above hypothesis along multiple key points in the AFE workflow.

Firstly, a methodology that combines AFE with ML has been developed to compute accurate absolute hydration free energies. The hybrid AFE/ML methodology was trained on a subset of the FreeSolv database, and retrospectively shown to outperform most submissions from the SAMPL4 competition. Compared to pure machine-learning approaches, AFE/ML yields more precise estimates of free energies of hydration, and requires a fraction of the training set size to outperform standalone AFE calculations. The ML-derived correction terms are further shown to be transferable to a range of related AFE simulation protocols. The approach may be used to inexpensively improve the accuracy of AFE calculations, and to flag molecules which will benefit the most from bespoke force field parameterisation efforts.

Secondly, early investigations into data-driven AFE network generators has been performed. Because AFE calculations make use of alchemical transformations between ligands in congeneric series, practitioners are required to estimate an optimal combination of transformations for each series. AFE networks constitute the collection of edges chosen such that all ligands (nodes) are included in the network and where each edge is a AFE calculation. As there are a vast number of possible configurations for such networks this step in AFE setup suffers from several shortcomings such as scalability and transferability between AFE softwares.

Although AFE network generation has been automated in the past, the algorithm depends mostly on expert-driven estimation of AFE transformation reliabilities. This work presents a first iteration of a data-driven alternative to the state-of-the-art using a graph siamese neural network architecture. A novel dataset, RBFE-Space, is presented as a representative and transferable training domain for AFE ML research. The workflow presented in this thesis matches state-of-the-art AFE network generation performance with several key benefits. The workflow provides full transferability of the network generator because RBFE-Space is open-sourced and ready to be applied to other AFE softwares. Additionally, the deep learning model represents the first robust ML predictor of transformation reliabilities in AFE calculations.

Finally, one major shortcoming of AFE calculations is its decreased reliability for transformations that are larger than ∼5 heavy atoms. The work reported in this thesis describes investigations into whether running charge, Van der Waals and bond parameter transformations individually (with variable λ allocation per step) offers an advantage to transforming all parameters in a single step, as is the current standard in most AFE workflows. Initial results in this work qualitatively suggest that the bound leg benefits from a MultiStep protocol over a onestep ("SoftCore") protocol, whereas the free leg does not show benefit. Further work was performed by Cresset that showed no observable benefit of the MultiStep approach over the Softcore approach. Several key findings are reported in this work that illustrate the benefits of dissecting an FEP approach and comparing the two approaches side-by-side.

Lay summary

Drug discovery is a challenging, time-consuming and expensive process. Although estimations vary significantly, the whole process generally takes 10-15 years and costs £0.3-0.8B, although other estimates often extend beyond the £1B mark. In reality, this process takes even longer because diseases are researched in academic laboratories for decades leading up to the point that a drug discovery campaign can be launched for the disease in question.

Despite being slow and expensive, it stands without reason that pharmaceutical discoveries are pivotal in advancing medicine and therapeutics towards the clinic, eventually aiding in patients' qualities of life and in other cases eradicating diseases altogether. Enormous amounts of global research are being done to investigate whether the process can be sped up or whether there are aspects in the drug discovery pipeline that can be made less expensive (often, these two aspects mean the same thing).

Computational chemistry is a major field of research that sprung up in the 1970's-1980's and it promised exactly this. The research in this thesis is in computational chemistry and is ultimately aimed at aiding drug discovery: the general name for the research field is computer-aided drug design (CADD). The main idea behind CADD approaches is that we are able (or rather, starting to be able) to replace costly laboratory ('wet lab') experiments with calculations done using computers ('in silico'). Although there exist many techniques in CADD, not all have proven succesful and even today the vast majority of pharmaceutical research is still done in a wet lab.

Alchemical free energy (AFE) calculations are one of the most popular CADD techniques. These work using molecular simulations (*i.e.* simulated using computers) and are able to accurately predict the binding strength of a drug candidate to a therapeutic target. For drug discovery this is extremely helpful as in early stages of drug discovery there is the need for candidate molecules that can bind the therapeutic target protein to inhibit its function: for example, ibuprofen binds the COX-1 and COX-2 proteins in the human body which normally cause inflammation; by binding these, ibuprofen *inhibits* their function and therefore reduces inflammation. Traditionally, medicinal chemists would have had to synthesise hundreds (if not more) trial candidate molecules that were small variations of the ibuprofen molecule (and testing how well they bind to COX1/2) before coming to ibuprofen - this process typically requires months of work done by large teams of scientists. With AFE, all of this can be done using computers in a matter of days given enough computer hardware.

AFE is a fairly established technique in early-stage drug discovery campaigns, but it is nowhere near a silver bullet and synthesis of candidate molecules is still required; AFE is currently used to support projects and guide medicinal chemists toward low-hanging fruit. This thesis concerns itself with the question of how AFE can be improved using a variety of approaches. Machine learning (ML) - or artificial intelligence - approaches have become very popular in CADD over the last decade and the work presented in this thesis aims to profit from these recent advances. The main research question for this work is thus:

Although areas of high potential exist for standalone ML and AFE in CADD, can additional value be found in areas where ML and AFE are combined in such a way that the new methodology profits from key strengths in either part?

Within this research framework I have performed several investigations which I describe in chapters 1-4 of this thesis. 1) First, I have composed a broad introduction to the theory behind the work in this thesis 2) I present work on a hybridisation of AFE and ML, where I have trained ML models to predict the *mistake* that AFE predictions make versus experimental measures. The aim of this work was to correct future AFE predictions using these mistake predictions: in scientific terms this is called *applying a correction term.* 3) Because ML models require large amounts of example data to learn from and because no such dataset exists within the field of AFE, I set out to generate such a dataset. Using this novel dataset, I have trained ML models to learn how precise certain AFE predictions are. This is helpful in planning large AFE campaigns, as running AFE simulations takes some time and it is valuable to be able to pick out 'easy to predict' candidates from a large series of candidate molecules. 4) finally, I have detailed investigations done during my research placement at Cresset, the CADD software company that has co-sponsored my studentship. This last chapter does not involve any ML science, but rather focuses on the bowels of AFE methodologies.

In summary, this thesis describes a rather unique approach in AFE science. By finding ways to loosely connect AFE and ML we have come to multiple additions to the scientific body of CADD; I am confident that future research in this area will further advance drug discovery efforts and ultimately bring medicine to patients more effectively.

Acknowledgements

I would like to express my deep gratitude for the excellent mentorship of Dr Julien Michel - working with you over the last four years has been inspiring and has helped me grow significantly towards scientific excellence. I am grateful that we will have further opportunities to interact professionally in the (near) future. I would also like to thank Dr Antonia Mey and Dr Mark Mackey for their unwavering support - I never knew scientific creativity could be lifted to such levels!

This thesis would not have been possible without the support of my family and friends, both in the United Kingdom and in the Netherlands. Claire, Cosmo, Frans, Gretha and Jeemijn and the Relevant folks as well as all our friends in Edinburgh who have made the city feel like home over the years.

Many colleagues have come and gone over the years within the Michel group and I owe my thanks to all of them - from small technical help up to inspiring me to push bleeding edge research in the lab. Especially Sofia, Jordi, Michalis, Adele: hopefully we will see each other soon!

Contents

List of Figures

List of Tables

List of acronyms

ABFE	Absolute Binding Free Energy
AFE	Alchemical Free Energy
AMBER	Assisted Model Building and Energy Refinement
AMOEBA	Atomic Multipole Optimized Energetics for Biomolecular Applications
APFP	Atom-Pair FingerPrint
API	Application Programming Interface
BAR	Benett Acceptance Ratio
BCC	Bond Charge Correction
BFGS	Broyden–Fletcher–Goldfarb–Shanno [algorithm]
BHO	Bayesian Hyper-Parameter Optimisation
CADD	Computer-Aided Drug Design
CDF	Cumulative Distribution Function
CDK	Cyclin-dependent Kinase
CG	Coarse-Grain
CHARMM	Chemistry at Harvard Macromolecular Mechanics
CHEMBL	Chemistry European Molecular Biology Laboratory
COVID	Coronavirus Disease
COX	CycloOxygenase
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
CYP	Cytochrome P
DNA	DeOxyribonucleic Acid

DNN	Deep Neural Network
ECFP	Extended-Connectivity FingerPrint
EI	Expected Improvement
ESDD	Early-Stage Drug Discovery
ESOL	Estimated SOLubility
FC	Fully-Connected
FE	Free Energy
FEP	Free Energy Perturbation
FF	Force Field
FXA	Factor Xa
FXR	Farnesoid X-Activated Receptor
GAFF	general AMBER force field
GBSA	Generalised Born and Surface Area [solvation]
GNN	Graph Neural Network
GPU	Graphical Processing Unit
GROMACS	GROningen MAchine for Chemical Simulation
GROMOS	GROningen MOlecular Simulation
HIF	Hypoxia Inducible Factor
HIV	Human Immunodeficiency Virus
HSP	Heat Shock Protein
HTS	High-Throughput Screening
JNK	c-Jun N-terminal Kinase
LIE	Linear Interaction Energy
LJ	Lennard-Jones
LOMAP	Lead-Optimization MAPper
MAE	Mean Absolute Error
MAPK	Mitogen-Activated Protein Kinase
MBAR	Multistate Bennett Acceptance Ratio
MCL	Induced Myeloid Leukemia [cell differentiation protein]

MCS	Maximum Common Scaffold
MD	Molecular Dynamics
ML	Machine Learning
MLR	Multiple Linear Regression
MM	Molecular Mechanics
MMPBSA	Molecular Mechanics Poisson-Boltzmann Surface Area
MPNN	Message-Passing Neural Network
MSE	Mean Squared Error
MSM	Markov State Model
MUE	Mean Unsigned [absolute] Error
NN	Neural Network
NPT	Number Pressure Temperature
NVT	Number Volume Temperature
OFF	Open Force Field
PBC	Periodic Boundary Conditions
PCA	Principal Component Analysis
PCM	ProteoChemometric Model
PDB	Protein Data Bank
PFKFB	6-PhosphoFructo-2-Kinase/Fructose-2,6-Biphosphatase 2
PL	Protein-Ligand
PME	Particle-Mesh Ewald
QM	Quantum Mechanics
QSAR	Quantitative Structure-Activity Relationships
QUBE	QUantum mechanical BEspoke
RBFE	Relative Binding Free Energy
RBFENN	Relative Binding Free Energy Neural Network
RF	Random Forest
RMSD	Root Mean Squared Devation
RMSE	Root Mean Squared Error

SAR	Structure-Activity Relationships
SB	Structure-Based
SEM	Standard Error of the Mean
SF	Statistical Fluctuation
SGD	Stochastic Gradient Descent
SKOPT	SciKit OPTimize
SMARTS	SMILES Arbitrary Target Specification
SMILES	Simplified Molecular Input Line Entry System
SOMD	Sire-OpenMM-MD
SPC	Simple Point Charge
SVM	Support Vector Machine
SVR	Support Vector [Machine] Regression
SYK	Spleen Tyrosine Kinase
TI	Thermodynamic Integration
TIP3P	Transferable Intermolecular Potential with 3 Points
TNKS2	TaNKyrase 2

Publications supporting the thesis

The research presented in this thesis has lead to the following publications:

- **Best Practices for Alchemical Free Energy Calculations**

  A. S. J. S. Mey, B. K. Allen, H. E. Bruce McDonald, J. D. Chodera, D. F. Hahn, M. Kuhn, J. Michel, D. L. Mobley, L. N. Naden, S. Prasad, A. Rizzi, J. Scheen, M. R. Shirts, G. Tresadern and H. Xu

  Living Journal of Computational Molecular Science, 2020, 2, 18378

- **Hybrid Alchemical Free Energy/Machine-Learning Methodology for the Computation of Hydration Free Energies**

  J. Scheen, W. Wu, A. S. J. S. Mey, P. Tosco, M. Mackey and J. Michel

  Journal of Chemical Information and Modeling, 2020, 60, 5331–5339

- **Data-driven Generation of Perturbation Networks for Relative Binding Free Energy Calculations**

  J. Scheen, M. Mackey and J. Michel

  Digital Discovery, 2022, 1, 870-885

Chapter 1

Introduction

Introduction

1.1 Foreword

The introduction to this thesis serves to provide the reader with the necessary background information to effectively parse the research outlined in the following chapters. Although an attempt has been made to span as much supporting theory as possible while remaining concise, further reading is recommended and is referenced where appropriate.

The structure of this introduction is constructed in a *top-down* design: initial subsections will outline the general context of the thesis research (the pharmaceutical drug discovery pipeline and the role of computer-aided drug design therein) after which the basic foundation theory of the binding of ligands to proteins is reflected upon. Then, an introduction to molecular simulation is provided, after which this is related to alchemical free energy calculations. Finally, introductory theory to machine learning in computer-aided drug design is presented.

1.2 The modern drug discovery landscape

The discovery of novel medicines is pivotal in advancing global healthcare towards curing disease. In most cases, the main objective of a drug discovery campaign is to find a therapeutic agent that interacts with a therapeutic target such that it modulates its biological function in a manner that leads to an alleviated disease phenotype. The global research body for drug development is vast in scope in both academic and commercial settings, resulting in a wide variety of therapeutic agent categories such as small-molecule inhibitors,[1] monoclonal antibodies[2] and vaccines.[3] Because the entirety of this thesis operates in the context of small-molecule drug discovery, the remainder of this introduction will omit other therapeutic agent categories. From here on, small-molecule drug discovery will be referred to as drug discovery.

1.2.1 The pharmaceutical pipeline

The financial and time expenses of drug discovery are notoriously high: although highly variable, the cost of bringing a drug to the market is estimated to range from ∼£0.3-0.8B,[4] although other estimates often extend beyond the £1B mark. Perhaps more critical however is the slow pace of development, resulting in an estimated 10-15 years[5] for most pharmaceutical campaigns. This period starts at the launch of a preclinical program and ends with a commercial, marketed drug. In the estimates described above, basic research to untangle the foundations of the disease phenotype are not taken into account; often these run for multiple decades across multiple parties (primarily in the academic domain) and financial expenses for these processes are challenging to quantify.

There exist a multitude of reasons for why this process is so intensive. There is one core concept that underlies these, which is that designing an effective drug (*i.e.* a drug that achieves its therapeutic goal while being synthesisable at scale) that is physiologically 'safe' (acceptable pharmacokinetics/dynamics and toxicity profiles, etc.) is extremely challenging. This results in high failure rates (highly dependent on disease context) with recent systematic estimates as high as 86.2% drawn across 5764 drug discovery companies.[4]

The pharmaceutical pipeline classically consists of sequential segments that operate as a workflow that inputs molecular candidates and outputs marketable drugs (figure 1.1). Given a defined therapeutic target, a large number of molecular candidates (in the order of 10^3 to 10^9) is screened both experimentally and computationally to produce a lead compound (**early-stage drug discovery**). A lead (*i.e.* 'leading') compound is defined as a compound that has reached pharmacological or biological goal effects but requires further adjustments (in terms of structure) to either bind the therapeutic target better or increase metabolism/toxicity profiles (see 1.2.2). Then, the lead compound is subjected to *in vivo* animal experimentation to determine physiological response prior to human testing (**pre-clinical trials**); subsequently there exist three **clinical trials** that sequentially investigate safety (**1**), efficacy of the indication (**2**) and efficacy in large populations (**3**) concluding with the regulatory process of allowing the developed drug to be marketed (**approval**). As clinical trial phases 1-3 involve increasingly larger infrastructures of experimentation these are associated with the bulk of the cost of developing drugs; unfortunately these phases are also associated with the highest estimated attrition rates: 86.2%, 79% and 41%, respectively. For this reason it is paramount that clinical candidates are of high quality to improve the chance that they pass clinical phases. It is clear then that any methodology that allows drug developers to save financial and time expenses while improving drug candidate quality can have a substantial impact in the development of novel medicinal agents.[6,7]

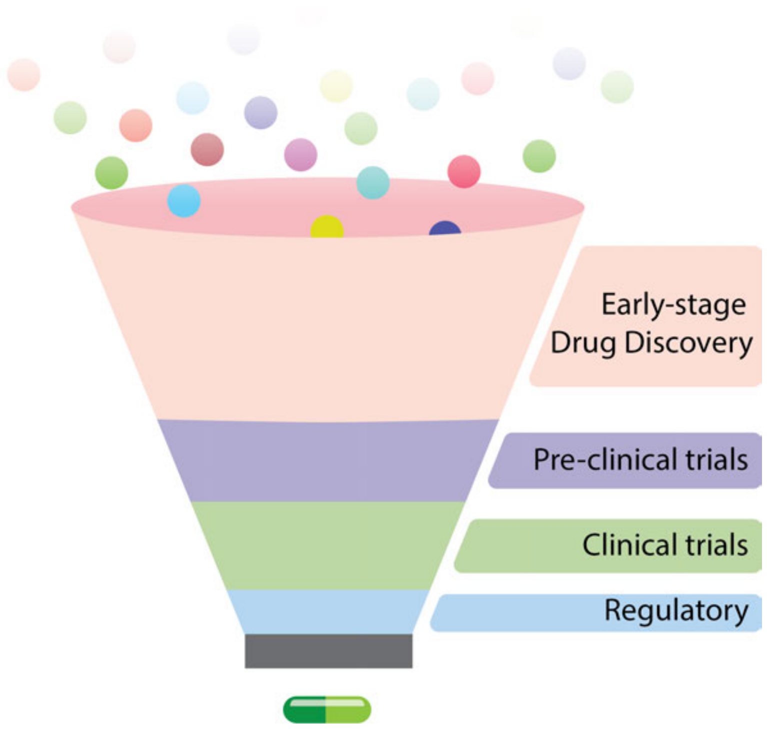


Figure 1.1: Schematic overview of the drug discovery pipeline funnel, a commonly used concept in visualising the end-to-end process of creating a marketed pharmaceutical agent. Each coloured circle represents a different drug candidate; at the bottom of the funnel a single marketed drug is created.

1.2.2 Early-stage drug discovery

The purpose of early-stage drug discovery (ESDD) phase is to produce compound leads with a high chance of progressing through the following stages of the pharmaceutical pipeline. The ESDD phase typically starts with a therapeutic target (in some unfortunate cases, no known target is defined which forces ESDD to be restrained to phenotypic screening[8]) for which a large number of molecular candidates are screened for hit molecules. From here on a 'target' is assumed to be a protein target rather than any other macromolecule. Additionally, candidate molecules are assumed to be small-molecules.

Hit discovery stage

Historically, one of the most successful implementations of hit discovery is high-throughput screening (HTS) for which (hardly any or) no prior knowledge is assumed except for some initial *in vitro* assay with a sensible readout related to the hypothesised mode of action;[9] HTS involves screening a large (n=10^3 to 10^6) compound library to the therapeutic target *in vitro* which requires considerable infrastructure to allow execution in an automated manner. The hit discovery stage of ESDD is intended to produce multiple hits that achieve some level of chemical diversity in order to spread the chances of succeeding in the next stages of ESDD. Subsequent stages involve only several hundreds of compounds.[5]

Hit-to-lead stage

The hit-to-lead (HtL) stage serves to further investigate all individual (series of) hit molecules obtained from the hit discovery stage. The main purpose of the HtL stage is to establish a robust structure-activity relationship (SAR) investigation to improve hit binding affinity to the therapeutic target as well as ensuring that selectivity is retained. At this point, it is common practice to develop a variety of assays that allow assessment of the degrees of selectivity and other undesired effects (CYP450 inhibitory activity, e.g.) that (optimised) hits might develop during SAR studies. Depending on the pharmacokinetic intention of the drug in development, steering of SAR towards certain physicochemical properties is applied in this stage as well. Early *in vivo* work can be performed on promising candidates in this stage.

Lead optimisation stage

The lead optimisation (LO) stage involves further fine-tuning of lead compounds that succesfully passed the HtL stage. In essence the main activity in LO is fine-tuning lead compound structures while optimising their performance in the assays mentioned in the HtL stage paragraph as well as general toxicity work. Successfully optimised leads are presented as pre-clinical candidates to further *in vivo* work. Typically only one or two pre-clinical candidates are advanced past the LO stage.

1.2.3 Computer-aided drug discovery

As outlined in 1.2.1, the drug discovery pipeline is highly complex and contains a vast number of (hidden) pitfalls, contributing to its costly and time-consuming nature. The dawn of computer-aided drug discovery (CADD) in the late 20^{th} century promised a virtual form of supporting ESDD in an effort to reduce its cost and speed up the process where possible. Although the integration of CADD has not been as rapid as previously estimated,[10] there exist many domains in ESDD (and other phases in the pharmaceutical pipeline) that benefit from CADD approaches at the time of writing.[11–13] This section will cover a selection of major CADD approaches that have proven successful in supporting ESDD.

Discovery of therapeutic targets

Therapeutic target discovery is positioned at the onset of the pharmaceutical pipeline. Although this stage is predominantly being executed in wet labs, some computational efforts are gaining traction. One major paradigm shift is the community's move away from the traditional reductionist view towards -omics approaches, where the aim is to assess a set of molecules in a specific system, for example metabolomics. Many disease-specific databases are under construction to support computational approaches to screen these, for example using expression data analyses stored within.[14,15]

More recently, with the rise of machine learning (ML) approaches combined with the increased availability of digitalised experimental biological data, new methods for target discovery have seen an increase in development and application, with several commercial start-ups applying this technique as a result. The core principle in this approach is that given the low computational expense of ML predictors it is now possible to virtually screen and select novel therapeutic targets for a disease indication by referencing it to the target and ligand scaffold of interest. This screening is done using a variety of descriptors and target labels. There are numerous examples of research using these techniques that have shown to aggregate therapeutic targets across protein classes for the same disease indication, hopefully paving the way towards combined therapies.[16,17]

In the last few years, considerable advances in protein structure prediction have been made, with the most recent leap made by AlphaFold2 developed by DeepMind.[18] This most recent predictor surpasses homology modelling accuracy in most cases and in some cases even matches experimental structure with very high accuracy (to within experimental uncertainties). AlphaFold2 offers an attractive technique for target identification: researchers now have the option of accurately predicting protein structure from sequence data alone. As crystal structure determination is a major bottle-neck in target identification and computational chemistry as a whole (with *e.g.* membrane-bound protein crystallisation only becoming a possibility in recent years[19]), this leap forward is set to advance the field of computational chemistry for years to come.[20,21]

Ligand-based approaches

As with target discovery, the ongoing growth of virtual databases (often containing bioactivity information) has proven to be fertile ground for ligand-based (LB) approaches in CADD. Primary examples of these are ChEMBL[22], a large database with over 2.1M compounds with bioactivity data at the time of writing, as well as ultra-large readily synthesisable virtual libraries such as Enamine REAL[23] and ZINC20.[24]

Although a wide variety of LB approaches exist in CADD (*e.g.* solubility/stability prediction, similarity searching), this section will highlight only LB approaches related to quantitative structure-activity relationship (QSAR) modelling as the thesis theme operates in this context. LB approaches have seen a rise in popularity due to a marked improvement in bioactivity databases such as ChEMBL; especially with novel descriptors and more accurate endpoint representations and scoring functions machine-learning models (section 1.6) have gained traction in QSAR modelling, leading to numerous QSAR models that are able to support virtual screening of large chemical databases *in silico*.[25,26]

Paired with LB virtual screening, *de novo*/generative modelling of candidate molecules has seen an increase in popularity in both the academic and commercial pharmaceutical landscape. Although the technique has seen applications since the early 1990s,[27,28] more recently with the rise of ML approaches its application has become more widespread and has been shown to be beneficial in several drug discovery campaigns. An attractive feature of this reinvigorated technique is the possibility of exploring chemical space while steering for specific physicochemical properties to arrive at drug candidates with desired attributes for the drug discovery campaign in question.[29–31]

Structure-based approaches

With the dawn of computational chemistry and CADD in the 1980s, the expected key outcome was the paradigm shift of experimental QSAR studies to rational design approaches. Although to this day SAR approaches are still the dominant method in HtL and LO stages of the pharmaceutical pipeline, structure-based (SB) approaches have grown to be invaluable tools in support of the ligand optimisation problem. In tandem with high-resolution crystal structures, docking approaches are pivotal in prioritising candidate molecules for synthesis by medicinal chemists. High-throughput docking algorithms allow rapid screening of large virtual databases to sizes of 10^{11} molecules, effectively allowing rapid exploration of a large part of drug-like chemical space in search of novel scaffolds for protein targets. A common first step when predicting binding affinities for novel ligands is to use docking to find reasonable ligand poses (in case the novel ligand can not be properly aligned to a crystallised ligand pose). It should be noted that in this step docking scoring methods are occasionally used as indicators of binding affinity but extensive literature exists that shows that these methods are insufficient in their current form. Docking scores should thus solely serve as indicators of how well the ligand fits in the binding pocket, and is most useful for discarding ligands/poses that will not fit at all.[32,33]

As with LB approaches, SB CADD has benefited greatly from the increase in virtual databases that create a data-rich context for further developement of these techniques. Primary examples of these are the protein data bank (PDB;[34] 180K 3D structures of which <70K are unique) which is a public repository of molecular crystal structures and the PDBBind database[35] which is a collection of ligand-protein crystal structures. More recently, the AlphaFold Protein Structure Database was generated that builds on successes of AlphaFold 2 (section 1.2.3) which contains over 360K predicted structures and has recently been expanded to cover all of UNIPROT.[36]

This rapid expansion of SB virtual libraries has significantly contributed to the increased development of SB physicochemical property prediction. For example, a variety of data-driven methods now exist that are competitive with docking approaches for ligand-protein pose prediction and combinatorial approaches have been shown to offer benefits compared to standalone techniques.[37,38]

Predictions of ligand binding affinity

Whereas the previously described SB approaches are informed by static images of (typically) crystal structures of ligand-protein systems, in reality these systems are dynamic which reduces the applicability and accuracy of naive ”snapshot-style” approaches such as docking. Approaches that describe molecular structures across a timescale using molecular simulation offer more elaborate representations of the thermodynamic landscape of a ligand-protein context, allowing for more accurate estimation of binding affinities. Although more detailed introductions to molecular dynamics (MD) and free energy perturbation (FEP) are presented later in this introduction (sections 1.4 and 1.5), this section will outline some alternatives to these methodologies.

Although highly accurate potential energy calculation techniques such as high-level *ab initio* methods are theoretically applicable to protein-ligand systems, in practice these are far too computationally expensive to handle molecular systems of this size. Sampling potential energy landscapes of protein-ligand systems is thus currently out of reach for these techniques. Instead, the most common physical models used are empirical force fields that describe the physical system as a collection of charged points connected by springs (section 1.4.2). Because this method of describing physical systems is considerably less expensive, protein-ligand system dynamics can be simulated over timescales up to the order of milliseconds in recent works.[39,40] MD opens up the possibility of free energy calculations in a time-dependent manner, where end-point methods such as the Molecular Mechanics Poisson-Boltzmann Surface Area (MMPBSA), Molecular Mechanics/Generalized Born Surface Area (MM/GBSA) and Linear Interaction Energy (LIE) have become increasingly popular. Although not strictly time-dependent (*i.e.* the same result should hold when frames are scrambled), these techniques benefit from conformational sampling of the molecular system and are able to compute statistics over the ensemble.[41–43]

More advanced alternatives to the aforementioned MD-based binding affinity estimation such as absolute FEP[44] and funnel metadynamics[45,46] are set to further advance the field of accurate computational ligand-protein binding affinity estimation. Markov state modelling (MSM) is rapidly gaining popularity and offers attractive opportunities for the development of novel algorithms that explore ligand binding modes, protein folding and allostery.[47,48]

1.3 A brief introduction to ligand-protein binding

To understand the work presented in this thesis a basic understanding of the principles of ligand-protein binding is required. This section will outline these concepts concisely which will serve as supporting theory for further sections that explore computational ligand-protein binding affinity predictions.

1.3.1 Pharmacological foundations

In the classical sense, small-molecule drugs are designed to bind protein receptors in some way to induce a therapeutic effect by increasing or decreasing the receptor's function. This can be accomplished by binding the receptor's catalytic site (binding pocket) directly or partially (*orthosteric binding*) or by binding a non-active site of the receptor (*allosteric binding*). Beyond this distinction, several drug-target relationships are defined: antagonists (partially) disable function of the receptor target, whereas agonists (partially) enable function of the receptor target (figure 1.2). Within the field of pharmacology there exist more distinctions (inverse/partial agonists; competitiveness etc.) but these are considered out of scope for this introduction. The binding of a drug A with its therapeutic target receptor R can be defined as

$$A + R \underset{k_{-1}}{\overset{k_{+1}}{\rightleftharpoons}} AR, \tag{1.1}$$

where k_{+1} and k_{-1} are the forward and reverse binding rate constants and AR is the drug-receptor complex. At equilibrium, the strength of binding (or rather, the tendency of the dissociation) of ligand A to receptor R can be expressed as

$$K_d = \frac{k_{-1}}{k_{+1}} = \frac{[A][R]}{[AR]}, \tag{1.2}$$

where K_d is the dissociation constant which has the dimension of concentration, $[A]$, $[R]$ and $[AR]$ are concentrations of free ligand A, unbound receptor R and complex AR, respectively. In practical terms, the K_d is used to express numerically the concentration of ligand A required to occupy 50% of the receptor A population at equilibrium.

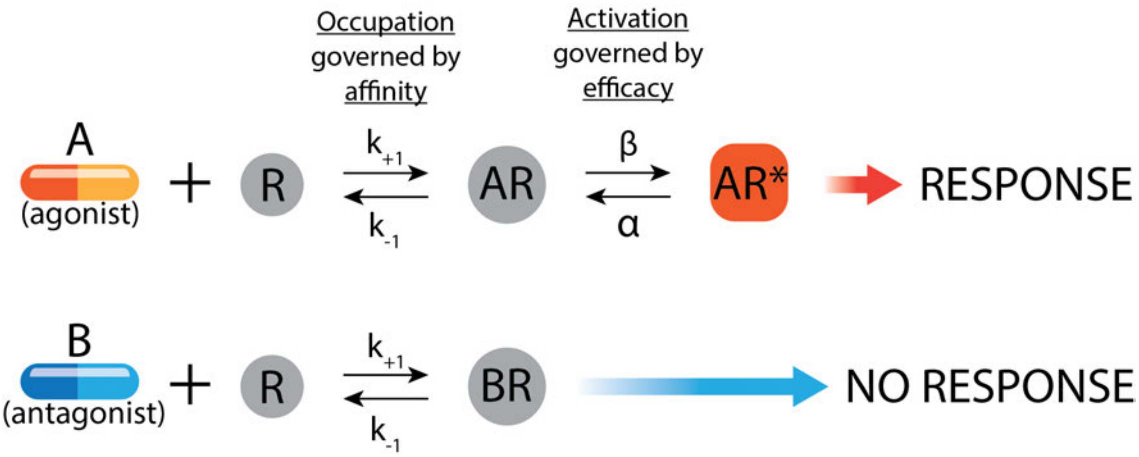


Figure 1.2: The distinction between drug agonism and antagonism in the context of protein receptor activation. Shown are drugs A and B (red/blue, agonist and antagonist, resp.) binding to target receptor **R** with binding rate **k**. In the case of agonism, the drug-target complex is activated with rate β, resulting in activated complex **AR*** which results in a biological response. Complex **BR** has $\beta = 0$ and thus no biological response follows after binding. Adapted from Rang *et al.*.[49]

1.3.2 Thermodynamic contributions of ligand-protein binding

Maximising ligand binding affinity to therapeutic targets is a key focus of early-stage drug discovery campaigns and is the ultimate goal of most of the major themes discussed in this thesis. By optimising ligand structure to decrease K_d, the therapeutic objective (*e.g.*, target inhibition; activation, see figure 1.1) is reached while accounting for other factors discussed in section 1.2.2. It is thus instructive to examine the thermodynamic foundations of what factors increase ligand binding affinities in ligand-protein binding. The standard state Gibbs free energy of binding of a ligand to a protein is expressed as

$$\Delta G^{\circ}_{bind} = -k_B T \ln K^{\circ}_d, \tag{1.3}$$

where k_B is the Boltzmann constant and T is the temperature in Kelvin and the K°_d is K_d divided by the standard state concentration. When $\Delta G < 0$, there is a thermodynamic favourability of the binding reaction in the forward direction, *i.e.* the ligand will tend to associate. An alternative thermodynamic description of the Gibbs free energy is

$$\Delta G_{bind} = \Delta H - T\Delta S^{\circ}, \tag{1.4}$$

where ΔH is the change in enthalpy and ΔS° is the change in standard state entropy of the system on ligand binding. Classically speaking, the entropic term describes the change in disorder of the whole system resulting from ligand binding or the change in translational, rotational and conformational entropy. The enthalpic term describes the change in heat resulting from ligand binding at thermodynamic equilibrium. Both terms are in kcal·mol^{-1}. Although Eq. 1.3 is more relevant to techniques discussed later in this chapter (section 1.5), the emphasis in this theoretical introduction will be on entropy and enthalpy as described in Eq. 1.4.

Entropic contributions to ligand binding

From a systemic perspective, entropy is the dispersal of energy from a localised form to a spread out form. On a molecular level, entropy relates to a system's number of accessible microstates. It follows that the standard behaviour of a system is to *maximise up to its limit* its entropy over time.[50] Binding of ligands to proteins decreases conformational entropy due to an increase in rigidity in both the ligand and the protein's binding site, thus resulting in an entropic penalty in Eq. 1.4. Complexation thus results in translational and rotational entropy loss which ultimately results in a penalty to binding affinity which can be as high as 25 kcal·mol^{-1}.[51,52] Beside the unfavourable change in conformational entropy, there exists the favourable effect of the change in solvation entropy on ligand binding resulting from (partial) desolvation of the ligand to the binding pocket. Both hydrophobic and polarisation entropic rewards are gained upon desolvation as well. This is due to an increase in accessible states of water in bulk versus water at the protein-ligand interface.[53] Water organises around hydrophobic molecules (in the form of clathrate cages), so burying hydrophobic molecules/regions contributes to disorganizing water (entropy gain). This is one of the key phenomena driving protein folding. The higher the degree of 'buriedness' of the binding pocket (and the more buried the ligand is) the higher the entropic gain upon ligand binding.

There have been some attempts at quantifying entropic contributions to ligand-protein binding computationally,[52] which revealed high variability in entropic penalties that are highly dependent on the molecular system in question. For host-guest systems, estimations are in the order of ∼20 kcal·mol^{-1},[54] whereas for example the entropic loss of amprenavir binding HIV protease is around 25 kcal·mol^{-1}. For the latter example it was found that the conformation contributed minimally to the entropic loss, whereas the vibrational entropy - a measure of the degree to which atoms are able to vibrate in their environment - contributed 93.2%.

Enthalpic contributions to ligand binding

A system's enthalpy is a measure of the total energy associated with the system which represents thermodynamically the sum of energies of all non-covalent intermolecular interactions contained in the system. If energy could be decomposed unambiguously a secondary definition would be that it is the energy 'stored' in all degrees of freedom in the system. On binding, enthalpy thus also involves internal interactions. ΔH is negative in exothermic reactions and positive in endothermic reactions.

A common strategy in ligand optimisation is to design an enthalpy-driven binding process by minimising the entropic contribution and maximising the enthalpic contribution (Eq. 1.4). An example of minimising the entropic contribution is to design a *preorganised* ligand that undergoes minimal conformational change upon binding (*i.e.* the ligand pose solvation is similar to that in the bound phase).[55,56] Optimising the enthalpic term (*i.e.* making ligand binding as exothermic as possible) is generally performed by introducing additional non-covalent interactions between the ligand and binding pocket amino acids. Any newly introduced interaction will further decrease the enthalpic term in favour of increased binding affinity; however the picture is not as clear-cut as ligand binding (*i.e.* gaining the collection of intended inter-molecular interactions in the ligand-protein complex) is also associated with loss of non-covalent interactions in the protein-solvent and ligand-solvent interphases.[57] Ultimately the collection of lost and newly-gained interactions constitute the net enthalpy change.

Key intermolecular interactions in ligand-protein binding

There exists a wide variety of non-covalent interactions. Rather than involving the sharing of orbitals (as is the case with covalent bonds), this type of interaction relies on electromagnetism in various forms. This section will briefly outline several non-covalent interactions that are commonly pursued in the ligand optimisation problem;

a visual guide is provided in figure 1.3. An attempt will be made to highlight the degree of contribution to enthalpy of each interaction mentioned, but it should be noted that the strength is highly dependent on the structures of the molecular participants as well as the solvent of the system in question.

Electrostatic interactions *Hydrogen (H-) bonds* are one of the most sought after interactions in ligand optimisation as their enthalpy contribution can be rather large at 1-4 kcal·mol^{-1} in aqueous solution (even ranging up to 20 kcal·mol^{-1} in some non-aqeous solutions) although desolvation costs can negate this effect.[58] H-bonds involve a dipole electrostatic attraction between a partially positively charged hydrogen atom and a partially negatively charged electronegative element such as oxygen, fluorine or nitrogen. *Halogen bonds* involve a similar electrostatic interaction as H-bonds, however in this case the electron acceptor is a halogen atom instead of a hydrogen. *Ionic interactions* (also referred to as salt-bridges) which involve a direct interaction between a negatively charged atom (*e.g.*, Cl^-) and a positively charged atom (*e.g.*, Na^+) also play an important role in ligand binding as physiological pH often results in ionised functional groups (*e.g.* acids or amines). All amino acids can be involved in H-bonding via their backbone; several also allow H-bonding on their side-chains such as tyrosine and glutamate. An important role for H-bonds in ligand-protein binding is in water networks, where a buried water molecule (or multiple) can act as an intermediate for H-bonding between the ligand and protein.[59]

Van der Waals Non-covalent interactions that involve a dipole-dipole interaction which are dependent on small fluctuation in electron densities are referred to as *Van der Waals* interactions, which is an umbrella term for several types of weak interactions (Keesom force, Debye force and London dispersion forces). Although these forces are not very strong by themselves (0.5-1 kcal·mol^{-1}), in practice they are essential in ligand-protein binding as their abundance leads to additive effects in binding affinity and they are regarded as the main driving force in ligand binding.[60]

π-π interactions The third most abundant interaction in protein-ligand environment are π effects, with π-π stacking the most commonly sought-after in rational design. These effects involve the interactions of π orbitals between two molecular entities, of which the most classical example is between two benzene rings. There exist multiple forms of π-π interactions: stacked/sandwiched, edge-to-face and displaced/slip-stacked which all vary in the range of 1-2 kcal·mol^{-1}. Slip-stacked π interactions are the least prominent of the three types. The importance of π interactions is highlighted by the abundance of aromatic moieties in drug-like molecular scaffolds; typically π-π stacking plays a pivotal role in directing the ligand pose to fit in the binding pocket geometry. Almost 50% of all π-π stacking interactions of ligands are with phenylalanine; followed by tyrosine, tryptophan and histidine.[59]

A comparison of intermolecular interactions for drug discovery Across the intermolecular interactions outlined in the above paragraphs there are several relationships to be drawn that influence which interactions are sought after during lead optimisation in early-stage drug discovery. Their strengths are one key factor (as outlined in the above paragraphs) - interaction span (or: length) is a second. Hydrogen bonds typically span across 2-3Å, depending on what chemical moieties make up the acceptor/donor parties. Hydrophobic interactions usually span longer distances, between 3.3-4Å. Although it may be attractive to favour high-strength intermolecular interactions during drug design, it is often more favourable to aim for a larger number of weaker interactions that span the protein binding site, for example targeting both the catalytic centre as well as the protein backbone with hydrogen bonding. Besides, over-emphasising certain moieties can lead to unwanted ADMET outcomes, such as with hydrogen donors/acceptors (skewed lipophilicity) and π-π benzene rings (metabolite toxicity).[61,62]

The hydrophobic effect and desolvation The tendency of non-polar chemical moieties to cluster with other non-polar chemical moieties (as with water-oil separation) is likely one of the main drivers in ligand-protein binding.[59] Coincidentally, the hydrophobic effect is also under heavy investigation because it is the main driver in protein folding.[63] Upon ligand binding, non-polar sections of both molecules approach each other while H-bonded networks of water molecules dissociate from both molecules while escaping into the bulk solution. As displaced water molecules are able to engage in more H-bonding in the bulk solvent (often up to four), an enthalpy gain is often achieved on desolvation. Additionally, entropy is affected favourably by releasing water molecules from the constrictive binding pocket. However, this entropic effect is partially compensated by the unfavourable decrease in entropy on ligand binding.[64,65] Water displacement research is in progress as there are examples of the free energy gain being as high as $\sim$2kcal$\cdot$mol^{-1} per water molecule[66] whereas other works have observed no benefit from water displacement which suggests that this principle is highly context-specific.[67,68]

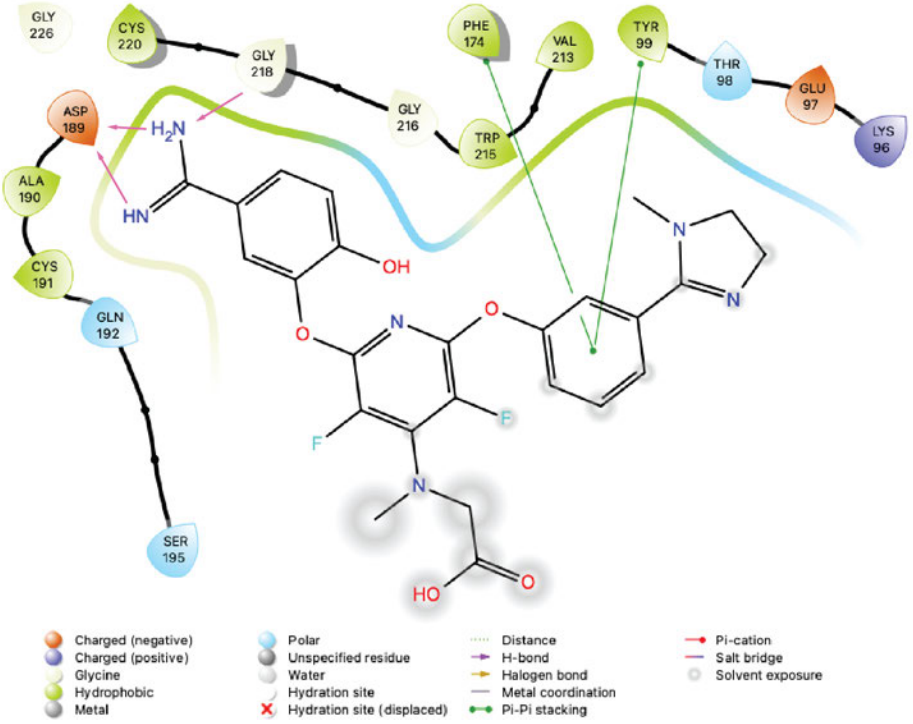


Figure 1.3: A 2D interaction diagram of PDB ID 1FJS (Factor Xa with inhibitor ZK-807834) generated using Maestro (academic). The three interactions depicted are hydrogen bonds (magenta, top left), π-π stacking (green, top right) and solvent exposure (grey areas, bottom). The protein backbone is shown as black thick lines between amino acids which are numbered according to index in the original crystal structure. The green/blue curved line indicates the binding pocket surface as a measure of lipophilicity of its residues (blue is hydrophilic; green is lipophilic).

1.4 Molecular dynamics simulations

Molecular dynamics (MD) simulations act as computational techniques that allow accurate depiction of the conformation of molecular systems over time. With advances in both computer hardware and MD algorithms, longer time scales can be simulated up to the point of milliseconds (given extensive hardware) in macromolecular systems at the time of writing. This order of simulation time allows sampling of biologically relevant effects. MD is extremely valuable to the scientific community in that it offers a fully quantifiable body of data on otherwise challenging to quantify real-world experiments. Beside allowing analysis of spontaneous events such as protein folding, MD is also widely used in the prediction/quantification of physicochemical properties of (macro) molecular systems.[69,70]

1.4.1 Foundations of molecular dynamics

The aim of MD is to simulate (bio-)physical processes accurately to enable quantification of otherwise difficult – or even impossible – measurements. To enable this, the fundamental physics that govern motion of bodies must be taken into account. Newton's second law of motion defines:

$$F_i = m_i a_i, \tag{1.5}$$

where for a given body i, the force F_i is a function of the body's mass m_i and acceleration a_i. Alternatively, the force F on body i can be expressed as the negative gradient of the overall potential energy with respect to the change in the body's position:

$$F_i = -\nabla U_i = -\frac{\mathrm{d}U}{\mathrm{d}r_i} \tag{1.6}$$

where U_i is the potential energy of body i (*e.g.* computed by an empirical force field, see 1.4.2) and r_i is the change in position of body i, typically in Cartesian coordinates. Finally, the body acceleration a_i is calculated as

$$a_i = \frac{\mathrm{d}v_i}{\mathrm{d}t} \tag{1.7}$$

where the velocity v_i is given as

$$v_i = \frac{\mathrm{d}r_i}{\mathrm{d}t}. \tag{1.8}$$

Instead of using the infinitesimally small change in time $\mathrm{d}t$, a timestep Δt is used in MD simulations to estimate derivatives via the finite difference approximation during simulation: this timestep is required to be of sufficiently small size to allow efficient sampling of molecular bodies that contain three or more atoms. Because it is not possible to reach an analytical solution to the equations of motion (Eq. 1.7-1.8) these must be solved in a discrete manner using numerical integration. Too large values of $\mathrm{d}t$ would result in large fluctuations of a_i leading to the frequently observed and unintentional *blowing up* of molecular systems in MD. For this reason, $\mathrm{d}t$ is typically defined in the order of femtoseconds - as one of the fastest molecular motions (C-H bond stretching) takes $\sim$ 10 fs, a recommended value of $\mathrm{d}t$ is 1 fs.[71,72] Timestep adjustments are frequently investigated in MD because increasing the timestep decreases the computational expense of running simulations: for example increasing $\mathrm{d}t$ from 2 to 4 fs can reduce simulation walltime by $\sim$50%. Although as previously mentioned higher values of $\mathrm{d}t$ (towards 10 fs) are more likely to result in unstable simulations, techniques have recently been explored that enable stable $\mathrm{d}t$ = 4fs MD simulations by using for instance hydrogen mass repartitioning and novel integrators such as the LangevinMiddleIntegrator.[73,74]

Controlling thermodynamic properties during molecular dynamics

Thermostats and barostats aim to ensure that the average temperature and pressure of a system remain at a desired level (resp.), rather than fixing them at the level. Thermostat algorithms work by adjusting the Newton equation of motion

(Eq. 1.5) or by rescaling particle velocities after the timestep. For example, the Andersen thermostat randomly selects particles and lets them collide with particles in an implicit external heat bath.[75] Barostats work similarly to thermostats in the sense that they typically have a loose connection with Eq. 1.5; through these, pressure is maintained during molecular simulations as if a piston would be working on the system. The Andersen barostat (used throughout chapters in this thesis) works similarly to its thermostat counterpart by introducing an implicit pressure bath which ultimately behaves as if the system is being acted upon by an isobaric piston.[76] There exist a wide variety of thermostats and barostats, however a detailed overview of these is considered out of scope for this theoretical section.

Electrostatic interactions

In MD, long-range electrostatic interactions that decay with separation in space are considered. To deal with the added complexity of these long-range interactions, all possible interactions (*i.e.* all particles interacting with all particles) up to a certain *cutoff* are considered. Periodic boundary conditions (PBCs) are introduced by creating identical copies of the system in a tiled manner. PBCs alleviate issues that come with the simulation of finite-size systems and allow more accurate descriptions of bulk properties during these simulations. Electrostatic interaction potentials can be numerically computed using a variety of algorithms, but the most classically used are Ewald-based methods such as Particle-Mesh Ewald methods (PME) in which a distinction is made between a direct and reciprocal space computation of interactions using a cutoff of a set distance in Å. The reciprocal interactions are typically the rate-limiting step of PME. PME is implemented in the majority of MD engines currently available.[77,78]

1.4.2 Force fields

Because a quantum-mechanics approach to compute a system's potential energy U is generally too computationally expensive for MD, a simplified quantification in the form of empirical force fields (FFs) is used which is expressed as:

$$U_{FF} = U_{bonds} + U_{angles} + U_{torsions} + U_{VdW} + U_{elec}, \tag{1.9}$$

where U_{FF} describes the total potential energy as computed by the force field by summation of the potential energy contributions of bonds, angles, torsions, Van der Waals and electrostatic energies. It is deemed out of scope for this thesis to describe the broad range of available FFs extensively, but a concise description of the FFs used in the work body of this thesis will be given together with how they relate to similar alternatives.

Ligand force fields Accurately parameterised small-molecules in ligand-protein simulations are pivotal in MD. Because the chemical diversity of ligands is large (as opposed to proteins which are built up of semi-consistent building blocks), a large amount of development is required to develop a ligand FF that accurately describes all small-molecule patterns (*i.e.* moieties or functional groups), as well as all combinations of them. This complexity has resulted in a wide variety of ligand FFs, some examples of which are the General Amber FF (GAFF; the dominant ligand FF in this thesis work due to its integration into Sire,[79] and its robust handling of varied chemical matter),[80] Optimized Potentials for Liquid Simulations 3 (OPLS-3)[81] and the CHARMM General FF (CGenFF).[82] More recently, ligand FF development has seen novel directions in the form of open-sourced FF development by OpenFF[83] and bespoke QM FFs such as QUBE-Kit.[84]

Protein force fields To describe large (biological) molecular systems such as proteins and DNA, protein FFs are used to compute U. Typically these force fields describe backbone, residue and tertiary interactions additively. Because of the complexity of macromolecular systems protein FFs experience a lot of incremental development: for example, the Amber force fields have seen incremental improvements in the handling of dihedral angles over the last 20 years. There exist a wide range of protein FFs such as Amber-type protein FFs (*e.g.*, ff19SB; the dominant protein FF in this thesis work)[85] and the OPLS-All Atom FF (OPLS-AA).[86]

Water models As water molecules play a crucial role in ligand-protein binding (see 1.3.2) the accurate modelling of these bodies in MD in this context is of substantial importance. Waters can be modelled implicitly (as a bulk solvent presence) and explicitly (as individual water molecules): the latter is used in the work presented in this thesis. Water FFs (more often referred to as water models) model water to varying detail; simpler, less expensive descriptions are 3-point (H-O-H) and 4-point (H-OM-H; M=oxygen partial charge) geometries; more elaborate models are also available but not often applied in protein-ligand simulation (*e.g.*, 5-point geometry and polarisable).[87] Commonly-used water models are TIP3P (used mostly in this thesis work),[88] and the Simple Point Charge (SPC) water model.[89]

1.4.3 State-of-the-art methodologies in molecular dynamics

The application of MD is not a single contained simulation - rather, practitionerspracticioners are required to execute a number of procedures sequentially in order to obtain reliable and accurate trajectories. There exist many MD methodologies owing to its diversity in applications but also to its algorithmic complexity and the requirement of balancing shortcomings in each approach. This subsection will give a brief overview of operations in MD that are commonly applied prior to the main MD production simulation; further reading is recommended to gain a more detailed knowledge of the theory involved.[69,90,91]

Preparation of ligands and proteins is required to set up input molecules for simulation. This step involves correction of structures as for instance with protein crystal structure PDB files occasionally there are missing or mis-represented residues in the system. This step also involves charging structures using some form of pKa estimator. Atomistic overlaps ('clashes') are resolved in this step, either algorithmically or by visual inspection and adjustments. Most importantly, this step involves the application of the desired FFs for the simulation ('parameterisation').

Solvation of protein-ligand systems is performed using some form of water box. Traditionally these water boxes used to be orthorhombic (orthogonal cube); however, because this geometry involves potentially a large number of explicit water molecules sufficiently distanced from the biomolecule (especially toward the eight corners of the box) such that they do not influence ligand-protein energetics, it is worth excluding these waters in some form. For this reason triclinic (non-orthogonal cube) and even shapes with > 6 sides are employed. In all cases, the system is considered in periodic boundary conditions that allow diffusion of particles across unit cells between neighbouring translated copies of the unit cell.[92]

Energy minimisation is performed on solvated, parameterised protein-ligand systems as a crude and computationally inexpensive measure to prevent large velocities in the initial steps of the MD simulation that result from high energies in the system. These large velocities can result from the initial system coordinates of the setup system that are abstracted from the crystallised protein structure. The purpose of this step is to find a local energy minimum.[93]

Equilibration is a necessary technique that aims to stabilise certain properties of the minimised protein-ligand system to obtain a thermodynamic ensemble. Especially for the work discussed in this thesis this step is important because the aim of the handled techniques is to run MD at equilibrium, *i.e.* with the protein-ligand system in a 'stable' or 'relaxed' state. Practically, this involves short simulations that aim to bring the system to a desired temperature or density. The work discussed in this thesis typically involves performing an NVT (constant temperature and volume) simulation and then an NPT (constant temperature and pressure) simulation - the production ensemble in this particular case is defined as an NPT ensemble (see 1.4.1).

1.4.4 Hardware

Graphical Processing Units (GPUs) are responsible for the leap in performance in MD methodologies. Although the first GPU was introduced in 1999, when NVIDIA presented the high-level programming interface CUDA in 2007 scientific programming for GPU hardware was democratised enormously.[94] Using GPUs and CUDA (and alternative interfaces), scientists were able to program massively parallel calculations - this created an advantage over Central Processing Unit (CPU) hardware as these are lacking in parallelisability as even multi-core CPU systems share central memory. From 2007 on, GPUs were no longer used solely for handling of video graphics but also for MD as well as for other applications that require large numbers of small calculations. MD is especially suited for this kind of hardware as the integration calculations required for running MD (see 1.4.1) can be aggressively parallelised. At the time of writing, the scale of protein-ligand systems simulated in regular MD campaigns has far exceeded the applicability domain of commodity CPU hardware.

1.5 Relative binding free energy calculations

Alchemical free energy (AFE) calculations are a group of free energy calculations that model *alchemical* processes using MD simulations, *i.e.* processes that are not chemically feasible (*alchemy*). The purpose of these calculations is to compute the change in Gibbs free energy for a given (alchemical) process in NPT ensembles. Example AFE calculations are estimating the membrane-binding free energy of small-molecules, computing the change in free energy due to a conformational change across a high free energy barrier and the binding free energy change on protein residue mutation.[95] The remainder of this theoretical introduction will describe relative binding free energy (RBFE) calculations. Although this introduction will use the terminology RBFE calculations, the same technique is also sometimes described as (relative) free energy perturbation (FEP) or simply binding free energy calculations.

While simulating binding events has been used to estimate binding affinities[96,97] or to get insights into the binding pathways and kinetics of receptor-ligand systems[98–102], the computational cost of these calculations is usually dominated by the rate of dissociation, which can be on the microsecond timescale even for millimolar binders[97] and reaches the microsecond to second timescale for a typical drug[103,104]. Depending on system size and simulation settings, common molecular dynamics software packages can reach a few hundreds of ns/day using currently available high-end GPUs[105,106], making these type of calculations unappealing and irrelevant on a pharmaceutical drug discovery timescale. Other methods compute the free energy of binding by building potential of mean force profiles along a reaction coordinate[107–110], but these methods require prior knowledge of a high-probability binding pathway, which is not easily available, especially in the prospective scenarios typical of the drug development process.

After early development starting in the early 1980s, around a decade into the 21^{st} century, RBFE calculations emerged as the first popular method that is able to robustly predict ligand binding affinities with a level of accuracy that is high enough to be able to support hit-to-lead and lead-optimisation campaigns in medicinal chemistry in commercial settings.[111] Since then, the field has progressed further up to a point where large numbers of calculations (in the order of hundreds of compounds) can be run in the course of only a few days (given sufficient hardware) which enables computational chemists to provide medicinal chemists with accurate predictions in aid of SAR studies at a considerably faster pace than synthesising each compound.[112]

Setup stage RBFE calculations simulate alchemical transformations between ligands (see 1.5.1 for rationale) which in practice means that for a protein target with a series of ligands a collection of ligand 'pairs' must be selected. Depending on the project, these series can be small (5-15) for *e.g.* benchmarking purposes, or large (15-100) for larger lead-optimisation projects. Several methods exist to propose which transformations to calculate and are implemented in RBFE softwares (figure 1.4A).[113,114] *Perturbation networks* are commonly used to depict the collection of edges proposed for a series of ligands and allow users to include/exclude transformations based on user experience (as some transformations are more likely to be reliable than others). For larger projects with many ligands, a star-shaped network is typically deployed with the reference ligand being the lead molecule that is being optimised. Force field assignment (see section 1.4.2) to both ligands and proteins is performed at this stage of the pipeline as well (figure 1.4B).

Production stage After defining the molecular transformation between the members of each ligand transformation (*i.e.* edge in the perturbation network) a λ decoupling parameter is typically used to divide the transformation into a number of bins, where parameters are adjusted in a bin-wise manner, each containing increasingly perturbed parameters. Both λ endpoints contain the atomistic parameters of both ligand endpoints and each λ intermediate system contains incrementally transformed atomistic parameters. These λ *windows* are then simulated individually using a given molecular dynamics engine (figure 1.4C): this process typically consumes the majority of RBFE walltime. Note that at this step both the bound and free legs are simulated (see 1.5.1). Additionally, edges are ideally simulated in both directions (*i.e.* both A → B and B → A) such that hysteresis in both directions can be analysed on-the-fly. In some RBFE implementations the bidirectional differences are used to re-balance $\Delta\Delta G_{bound}$ predictions by pushing edge hystereses closer to 0 kcal·mol^{-1}. There exist several methods of representing the atomistic change between two ligand enpoints, notably single (which involves transforming atoms to new atom types directly as much as possible) and dual (which involves changing atom types only to and from non-interacting dummy atoms) topology algorithms.[95]

Free energy estimation stage Upon completion of simulations, the relative free energy across the λ decoupling parameter is then computed using estimators such as Thermodynamic Integration (TI) or more recently Multistate Bennett Acceptance

Ratio (MBAR) for the perturbation in both the bound and free phase. See 1.5.2 for a more detailed theory outline of this estimation. In this stage a pairwise relative binding free energy ($\Delta\Delta G_{bound}$) estimation is produced for all planned transformations of the RBFE campaign.

Analysis stage Finally, $\Delta\Delta G_{bound}$ values per ligand are estimated using the original perturbation network, where it is common to correct for cycle closures (as a cycle of ligands' free energies should have a net energy of 0 kcal·mol^{-1} due to the law of conservation of energy). A common strategy in correcting this is to run calculations for both directions of an edge, then shifting the forward and reverse free energy predictions involved in the cycle such that the cycle net energy is brought to 0 kcal·mol^{-1}. Typically $\Delta\Delta G_{bind}$ values are estimated through *e.g.* a weighted-least squares method where edge predictions are weighted by some form of uncertainty quantification such as the standard error of the mean free energy prediction across replicates or an uncertainty estimate using bootstrapped subsampling of the simulation data. Using one of the ligands as reference, per-ligand $\Delta\Delta G_{bound}$ values are estimated. These values can be compared to experimental binding measures (that have been scaled to the same reference ligand) to allow benchmarking of the RBFE workflow.

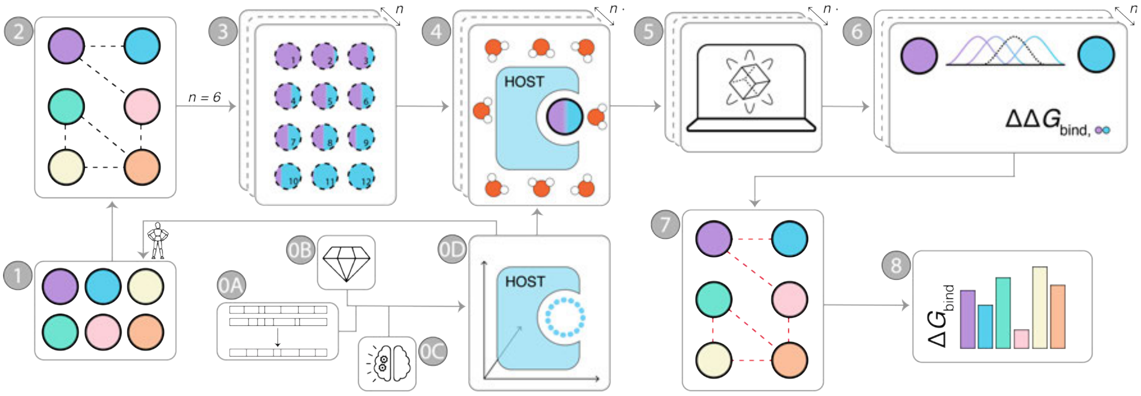


Figure 1.4: Workflow of a typical relative binding free energy (RBFE) calculation campaign. **0A-D**: a reference three-dimensional protein structure is procured (D) from homology modelling (A), crystallography (B) or machine learning (C) **1**: n ligands are manually positioned in the binding pocket of the protein structure through *e.g.* docking algorithms. **2**: a perturbation network is generated that specifies which perturbations will be performed between the ligands in the series. **3**: transformations are set up for each edge's λ windows **4**: ligand transformations are placed in a solvated box with the reference protein **5**: simulations are run on GPU hardware. **6**: given the completed λ window simulations, the relative free energy of binding can be estimated across the transformation **7**: with the perturbation network's edges completed, analysis is done on pairwise $\Delta\Delta G_{bind}$ values to infer ΔG_{bind} for each ligand compared to a reference ligand **8**: ΔG_{bind} predictions per ligand can be compared to experimental values for benchmarking or can be used directly to guide lead optimisation.

1.5.1 Theoretical foundations

In many cases, the quantity of interest is the change in binding affinity between a compound A and a related compound B (*e.g.*, by modifying one of the drug scaffold's substituents) can be considered as the difference between two standard state binding free energies (Eq. 1.3) which is given by

$$\begin{aligned}\Delta\Delta G_{\text{bind},AB} &= \Delta G^{\circ}_{\text{bind},B} - \Delta G^{\circ}_{\text{bind},A} \\ &\approx -k_B T \left(\ln \left[\frac{Z(RB)}{Z(R+B)} \right]^{\circ} - \ln \left[\frac{Z(RA)}{Z(R+A)} \right]^{\circ} \right) ,\end{aligned} \tag{1.10}$$

where $-k_BT$ is the Boltzmann constant times temperature in K, R is the target receptor and $Z(..)$ is a configurational partition function; these terms can be considered parallel to a likelihood variant of K_i as in Eq. 1.3. Note that the terms involving the standard concentration cancel out when we assume that the volume is identical for A and B. Predictions of $\Delta\Delta G_{\text{bind},AB}$ with non-alchemical methods generally require long simulations of both ligands, possibly through different binding pathways. Alchemical relative free energy calculations avoid the need to simulate binding and unbinding events by making use of the fact that the free energy is a state function and exploiting the thermodynamic cycle illustrated in Fig. 1.5. This is apparent after rewriting Eq. 1.10 as

$$\begin{aligned}\Delta\Delta G_{\text{bind},AB} &\approx -k_B T \left(\ln \left[\frac{Z(RB)}{Z(RA)} \right]^{\circ} - \ln \left[\frac{Z(R+B)}{Z(R+A)} \right]^{\circ} \right) \\ &= -k_B T \left(\ln \left[\frac{Z(RB)}{Z(RA)} \right]^{\circ} - \ln \left[\frac{Z(B)}{Z(A)} \right]^{\circ} \right) \\ &= \Delta G^{\circ}_{\text{bound}} - \Delta G^{\circ}_{\text{unbound}} ,\end{aligned} \tag{1.11}$$

where $\Delta G_{\text{bound/unbound}}$ is the free energy of mutating A to B in the bound/unbound state. Eq. 1.11 and Fig. 1.5 tell us that the difference in free energy of binding between toluene (A) and benzyl alcohol (B) can be computed by running two independent calculations estimating the free energy cost of mutating A into B in the binding pocket (ΔG_{bound}) and in solvent ($\Delta G_{\text{unbound}}$), saving us the need to simulate the physical binding process of the two compounds. In particular, the second line of Eq. 1.11 is a consequence of $\Delta G_{\text{unbound}}$ being independent of the presence of the receptor in the simulation box as the definition of the unbound state assumes receptor and ligand to be at a sufficient distance for them to have no energetic interactions. Note that, when A and B have different numbers of atoms, the factors $\ln \frac{Z(RB)}{Z(RA)}$ and $\ln \frac{Z(B)}{Z(A)}$ in Eq. 1.11 appear both to have factors with units of volume in the logarithms, but these factors exactly cancel between the terms.

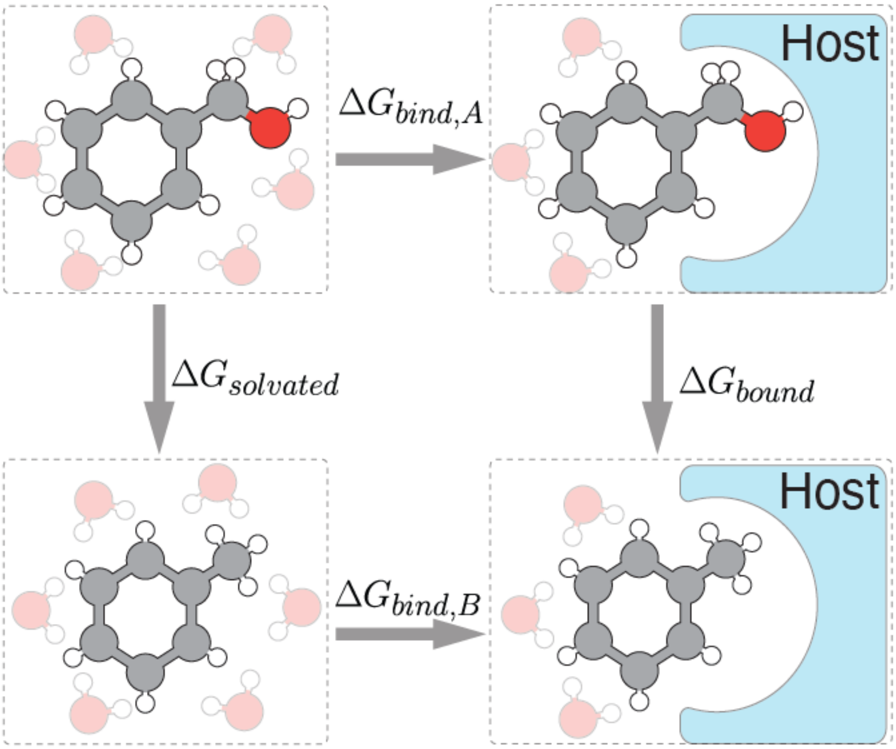


Figure 1.5: Thermodynamic cycle for computing the relative free energy of binding ($\Delta\Delta G$) between two related small molecules to a supramolecular host or a rigid receptor. The relative binding free energy difference between two small molecules, $\Delta\Delta G_{\mathrm{bind},A\to B} \equiv \Delta G_{\mathrm{bind},B} - \Delta G_{\mathrm{bind},A}$—here benzyl alcohol (top) to toluene (bottom)—can be computed as a difference between two alchemical transformations, $\Delta G_{\mathrm{bound}} - \Delta G_{\mathrm{solvated}}$, where $\Delta G_{\mathrm{bound}}$ represents the free energy change of transforming $A \to B$ in complex, *i.e.* bound to a host molecule, and $\Delta G_{\mathrm{solvated}}$ the free energy change of transforming $A \to B$ in solvent, typically water. Figure was adapted from Mey *et al.*[95] under the CC-BY 4.0 license.

1.5.2 Estimation of free energies

Because it is challenging to design molecular simulations that involve alchemical atomistic transformations, RBFE workflows use a λ decoupling parameter to divide the transformation into a number of bins, where parameters are adjusted in a bin-wise manner, each containing increasingly perturbed parameters (see 1.5 and figure 1.4). Typically these λ windows are spaced equidistantly but other spacings exist.[115–119] The key consideration for choosing alchemical pathways is that the intermediate states that a given pathway produces should sample configurational ensembles that change as slowly as possible as λ changes, while still managing to go from the initial state to the final state as λ goes from 0 to 1.

Common algorithms

There exists a range of functions to estimate relative energies across pathways in alchemical free energy calculations. For the sake of pedagogy, this theoretical section will first outline thermodynamic integration (TI) and then expand onto the more complex (but more commonly applied) approaches Bennet Acceptance Ratio (BAR) and Multistate-BAR (MBAR).

TI is one of the most simple estimation functions of free energy differences. In essence, the objective is to obtain the free energy derivatives with respect to λ. Formally, this derivative can be expressed as

$$\frac{\mathrm{d}G}{\mathrm{d}\lambda} = \left\langle \frac{\mathrm{d}U(\lambda, \vec{q})}{\mathrm{d}\lambda} \right\rangle_{\lambda}, \tag{1.12}$$

where $\vec{q}$ is the collective variable for coordinates and momentum of the system for the given λ state. When all λ simulations are obtained, the relative free energy across the pathway λ can be computed as the integral

$$\Delta G = \int_0^1 \left\langle \frac{\mathrm{d}U(\lambda, \vec{q})}{\mathrm{d}\lambda} \right\rangle_{\lambda} \mathrm{d}\lambda. \tag{1.13}$$

Although TI is attractive due to its ease of use, it suffers from limitations due to its usage of singular ensemble states rather than coming to an iterative solution. TI has the problem that the integral must be estimated via numerical integration which requires a finite number of data points and depending on where the integrand are taken there will be a different systematic bias in the resulting free energy estimation- this results in TI requiring the user to have to resort to increased sampling of transformations which can be detrimental to computational expense.[120,121] **BAR** offers the advantage of a decrease in bias due to the inclusion of both forward ($\mathrm{d}U_{ij}$) and reverse ($\mathrm{d}U_{ji}$) potential energy differences in its analysis. The free energy difference between two neighbouring states i and j is found by numerically solving

$$\frac{1}{\langle 1+\exp[+\beta(\mathrm{d}U_{ij}-C)]\rangle_i} = \frac{1}{\langle 1+\exp[-\beta(\mathrm{d}U_{ji}-C)]\rangle_j} \tag{1.14}$$

where $C = \Delta G_{ij} + \frac{1}{\beta}\ln(\frac{N_j}{N_i})$ and $\beta = (k_b T)^{-1}$. Finally, **MBAR** is a direct extension of BAR as it allows data assessment from all states in λ instead of just adjacent ones:

$$\hat{f}_i = -\ln\left\langle \frac{\exp\left[-u_i(x_n)\right]}{\sum_{k=1}^{K} \frac{N_k}{N}\exp\left[\hat{f}_k - u_k(x_n)\right]} \right\rangle, \tag{1.15}$$

where u_i and u_k are the reduced potentials of thermodynamic states i and k, x_n is the nth observable configuration and K is the collection of states across λ. $\hat{f}_k$ is a single free energy - another free energy must be taken as reference which will result in restoration of the estimation of *relative* free energies. MBAR has been shown to have the lowest variance estimator of all free energy estimators and it allows direct computation of prediction uncertainties which has lead to the algorithm's widespread use.[122,123] For a concise description see *e.g.* `alchemistry.org/wiki/Multistate_Bennett_Acceptance_Ratio`.

Phase space overlap

Another way of stating this is that intermediate states should sample molecular configurations that have similar likelihoods to be observed in the sampled intermediate states. The more similar the configurations are between intermediate states, the lower the statistical uncertainty is in the estimate of free energy between intervals. This can be proven directly from the BAR and MBAR formulae[123,124], though the exact same principles apply for TI. For a 'good' path to work and give a sequence of states with maximally similar configurations, sufficient similarity in potential energy distributions is required. Figure 1.6A and B illustrate this. Figure 1.6A shows in a pictorial way a soft-core potential can be applied across different λs. Figure 1.6B illustrates the potential energy distributions at the different λ intermediates, with sufficient overlap between neighboring λ states to ensure that reweighting estimators such as MBAR can be used for analysis. The actual transformation is best handled with soft-core potentials of the form shown in figure 1.6C and B.

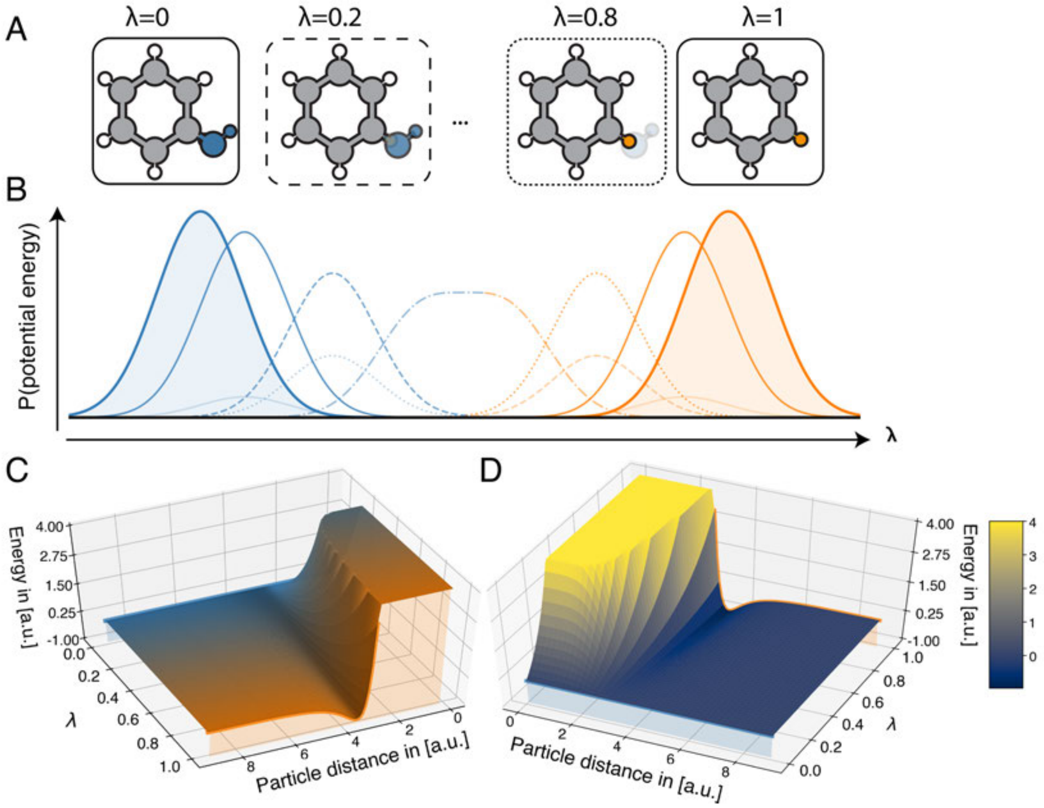


Figure 1.6: Alchemical intermediates are created by making the potential energy depend on an additional variable λ that interpolates between the chemical endpoints. In (**A**), at $\vec{\lambda} = 0$ the molecule is a fully interacting phenol and at $\vec{\lambda} = 1$, a fully interacting benzene. (**B**) shows an illustration of the probability distribution of the potential energies as the switching function takes values of $\vec{\lambda} = 0$ to $\vec{\lambda} = 1$. Intermediates states are required for a sufficient overlap in potential energies to estimate a free energy difference between $\vec{\lambda} = 0$ and $\vec{\lambda} = 1$. Soft-core potentials provide one of the most efficient families of intermediate pathways, with a λ dependence. In (**C**) the potential energy surface is coloured according to λ with blue being $\vec{\lambda} = 0$ and $\vec{\lambda} = 1$ orange. In (**D**) the potential is coloured according to the potential energy. Note how as λ approaches 0, the energy smoothly approaches zero at all r, a necessary requirement for efficient and stable calculations. Particle distance in these plots is a an arbitrary distance between the atoms, one in either chemical endpoint, that are being transformed between across λ. Figure was adapted from Mey *et al.*[95] under the CC-BY 4.0 license.

1.5.3 Applicability domain of relative binding free energy calculations

Although RBFE is a relatively robust method to predict ligand-protein binding affinities, it is not a silver bullet that can be applied to any ligand optimisation problem. The applicability domain of FEP is limited by a number of factors. State-of-the-art commercial RBFE software implementations are ideally deployed on small-to-medium-sized water-soluble protein systems (1) in ligand-bound conformations that do not have overly flexible regions (2). Protein crystal structures are typically required to have a resolution of <2Å. Preferably, ligand binding pockets are buried (or at least, not overly hydrophobic) and do not contain water molecules (beside the ligand) or metal ions (3). The investigated series of ligands must be well-aligned to an accurate (preferably crystallised) binding pose (4), and ligands must be topologically similar (5) and have the same net charge for consistently reliable results (6) (note that this is a major caveat for drug discovery project support and research on this front is highly active[95]). Furthermore, the series' inter-ligand binding affinities (dynamic range) should be larger than $\sim$ 4 kcal·mol^{-1} to allow reasonable statistical analysis for benchmarking purposes; in prospective work this point is less relevant (7).

1.5.4 Success stories

There exists a rich collection of success stories that involve some form of RBFE. For the purpose of this subsection, only recent highlights of the field will be discussed. Unfortunately, as most RBFE-supported hit-to-lead and lead-optimisation is performed commercially, the majority of this work exists outside of the literature domain.

Merck has performed a large-scale benchmarking study on FEP+ software[111] performance and has published a large portion of this dataset for public benchmarking purposes.[125] It was found that although there exist caveats in the applicability domain of the software, the majority of projects benefit from FEP+ support.

Schrödinger as well as its practitioners have published a wide range of publications that outline improvements to its FEP+ engine. Victories here include protein FEP, integration with generative ML workflows, macrocycle FEP and enhanced sampling of water molecules using Grand Canonical Monte Carlo algorithms all of which have further expanded the domain of applicability of FEP applications for common practice.[126–129] Recent work done by Janssen highlights how FEP can support hit-to-lead phases by accurately predicting a stereochemical SAR switch for NIK-kinase inhibitors in multiple myeloma.[130] Additionally, ligand binding was modelled accurately on membrane-bound Adenosin 2A and orexin-2 receptors in a collaboration with Sosei Heptares and the university of Leiden.[131] More recently, Jorgensen *et al.* have shown how RBFE calculations are able to support drug discovery by optimising triarylpridinone inhibitors of the main protease of the SARS-CoV-2 virus.[132] Here, RBFE calculations were primarily used to explore binding affinities of alternative heterocycles on the ligand scaffold. Also in the context of SARS-CoV-2 drug discovery was the global *COVID-Moonshot* consortium which in 2021 used the Folding@home infrastructure to run RBFE calculations on up to 10,000 compounds per week. This scale was accomplished due to the immense community effort of distributed computing contributions by the community through Folding@home which resulted in the first reported exascale computing infrastructure for this purpose.[133] Finally, a recent study has shown that using a novel form of RBFE calculations (non-equilibrium switching[134]) and given sufficient hardware hundreds of compounds can be predicted on in a matter of days.[112]

1.6 Machine-learning in drug discovery

1.6.1 Historical overview

Data-driven models such as machine-learning (ML) models owe their recent popularity mainly due to the rapid expansion of available data over the last few decades. These large datasets are typically open-source (*e.g.* bioactivity databases such as

ChEMBL,[22] a large molecular activity database with over 2.1M compounds at the time of writing) and are essential when attempting to train ML models to predict ligand-protein binding affinities when practicioners do not possess readily available in-house datasets.

A wide variety of ML models exist, and a paradigm shift can clearly be observed: whereas initial models such as support vector machines (SVMs) and random forests (RFs) took the stage in the 90s as affinity predictors, deep neural networks (NNs) have become increasingly popular over the last decade because of their ability to handle large amounts of data while being supported by increasingly powerful GPUs as well as perceived successes in other fields such as computer vision and natural language processing. The growth of NNs for drug discovery has resulted in a large pool of available algorithms and platforms such as DeepChem.[135] More recently, convolutional neural networks (developed primarily for computer vision algorithms) have been used to train on ligand-protein systems,[136,137] *de novo* generation of molecules[138–140] and even proteins[18,141] is set to further advance the field.

Despite their diversity, data-driven models for early-stage drug discovery ligand optimisation support ultimately share the same aim as physics-based models: to predict binding affinity of ligands to protein binding pockets. In this case however they do this through some form of featurisation (*e.g.* molecular properties, molecular fingerprints, or structural information) that, given a training set, is used to fit molecular properties such as binding affinity as either a regression (*i.e.* continuous) or classification (*i.e.* categorical, *e.g.* active vs non-active). Although a promising alternative to physics-based modelling which can be computationally expensive, occurrences where data-driven models were able to predict ligand-protein binding affinity with mean squared error (MSE) under 1 kcal·mol^{-1} on prospective tests have been rare, rendering the technique mostly insufficient for lead-optimisation campaigns. An additional downside to data-driven models is the general lack of explainability commonly referred to as the black-box problem.[142,143] Pure data-driven modelling is at this time mostly employed at hit discovery and hit-to-lead stages of the computational drug discovery pipeline.[144]

Because the main requirement for training effective ML models typically is a large training domain, these models often suffer from inadequate predictivity in cases where data is sparse and in cases where there is insufficient contextual data to describe energy fluctuations such as with activity cliffs.[145,146] Additionally, it is often prohibitively challenging to predict binding affinities accurately for completely novel therapeutic targets for which no/ hardly any data exists. This particular situation is common in early drug discovery campaigns.

1.6.2 State-of-the-art

The current landscape of ML research is diverse and fast-paced. There exists a wide variety of ML algorithms that are in active development, mainly in the fields of image recognition, natural language processing and robotics. The last few years have seen many of the models investigated in these fields translated into algorithms designed to handle chemical information in novel ways. Examples of these are convolutional neural networks and graph neural networks.[147] Because of the diversity of ML algorithms in the field, this thesis chapter has been constructed to only highlight the essential theory behind the three main techniques presented in the thesis chapters, namely NNs, RFs and SVMs (figure 1.7). Although dataset handling protocols can be different per ML implementation, division into training and test (and in the case of NNs, validation) sets is critical to ensure that models are trained independently of the test data - this allows scientists to accurately depict statistical performance of ML models that is a realistic portrayal of how these trained models would perform in settings where they are applied to practical test sets.

Neural Networks

In neural network (NN) nomenclature, a complete pass of the training set to the network is referred to as an epoch. In general, a NN algorithm learns by iterating over epochs, checking its accuracy compared to true values, and tweaking parameters to perform better in the next epoch. NNs are made up of j neurons arranged in l layers that each express an activation value a_j^l:

$$a_j^l = \sum_k w_{ij}^l a_i^{l-1} + b_j^l \tag{1.16}$$

where w_{ij}^l is a continuous weight variable between 0-1 that originated from the previous neuron i, b_j^l is the bias variable between [0-1] in cases where normalisation is applied that is associated with neuron j and $a_i^{l-1}$is the activation output between [0-1] that originated from the previous neuron k (figure 1.7A). In the case of a regression problem, NN architectures are typically designed to converge to a single neuron of which the activation a_j^l will be a value between [0-1]. There exist many types of activation fuctions, each designed to pass the activation in different patterns. Commonly used examples are linear activation function which outputs a linear form of a_j^l between [0-1] and a sigmoidal activation function that biases a_j^l towards either 0 or 1. In regression problems the output neuron typically contains a linear activation function to allow extrapolation of a realistic prediction label. The weights and bias variables are referred to as parameters; all other (user-set) parameters such as the type of activation function and the number of neurons per layer are referred to as hyperparameters. Within the NN architecture, a vector (an array of feature values, *i.e.* a data point with a set of features) is supplied as the input layer, where each neuron occupies a value of the vector. Layer-wise multiplication as dictated by Eq. 1.16 leads to a single predicted value of a_j^l for the output neuron by gradually decreasing the dimensionality of l. Because the training set contains true values (labels), these can be cross-checked with the prediction to compute some form of error. This cross-checking is referred to as the cost function, and the mean squared error is one of the most commonly used types:

$$E(X, \Theta) = \frac{1}{n} \sum_{d=1}^{n} (\hat{y}_d - y_d)^2, \tag{1.17}$$

where $E(X, \Theta)$ denotes the error for dataset X with parameters Θ (the weights and biases for each epoch). The error is computed over n datapoints (*i.e.* vectors) that are present in the dataset comparing the predicted output $\hat{y}_i$ with the true value y_i. In a given epoch, when all neurons have forwarded their activation and $E(X, \Theta)$ has been computed, an algorithm called back-propagation ("backward propagation of errors") creates a gradient of errors by calculating the error per neuron, per layer in the inverse direction. For these errors, a partial derivative of the cost function with respect to a weight from neuron i from previous node j in layer l is computed as

$$\frac{\partial E(X, \Theta)}{\partial w_{ij}^{l}} = \frac{1}{n} \sum_{d=1}^{n} \frac{\partial}{\partial w_{ij}^{l}} (\frac{1}{2}(\hat{y}_d - y_d)^2) = \frac{1}{n} \sum_{d=1}^{n} \frac{\partial E_d}{\partial w_{ij}^{l}}. \tag{1.18}$$

This gradient is then used to adjust the weights in the NN using a gradient descent protocol on the partial derivatives as supplied by back-propagation to update the weights in the NN denoted as Θ:

$$\Theta^{t+1} = \Theta^{t} - \alpha \frac{\partial E(X, \Theta^{t})}{\partial \Theta}, \tag{1.19}$$

where Θ^{t+1} is the updated set of weights as opposed to the original set of weights set in Θ^{t}. The learning rate α is a hyperparameter between [0-1] set by the user that, when set to $\alpha < 1$, can reduce the impact back-propagation has on the adjustment of the weights per epoch. The gradient descent function, often referred to as the optimiser function in NN nomenclature, attempts to find an optimal parameter setting (Θ) to minimise the cost function. A wide range of optimisers exist and although it remains challenging to determine which types are fit for which optimisation problems, the most widely used are stochastic gradient descent (SGD) or derivatives thereof such as adaptive moment estimation (Adam). Briefly, SGD descents the gradient efficiently by only sampling random subsets of training data. The main bottleneck with SGD is that it can get stuck in local error minima; Adam has been developed to counter this issue. Adam is adaptive in the sense that it scales α (Eq. 1.19) using the squared gradients of the cost function; it also descends on the moving average of the gradient rather than the gradient itself as is the case with SGD.

Specifically with NNs, overfitting is a common issue. This occurs when a model under training is starting to exactly fit its training data, thereby reducing its predictivity on external test sets (which is after all its main purpose). Care must be taken to avoid overfitting NNs, this can be done by for example *early stopping* algorithms that monitor the error on the validation set and halt training once this error starts to increase along epochs.

Random Forests

The work presented in this thesis makes use of 'extremely randomised trees' which is based on the original RF algorithm by Breiman.[148] In general, RFs make use of an ensemble of decision trees which are constructed from random, independent feature subsets of the training data - a process called bagging, short for bootstrap aggregating. The objective is to make all trees in the ensemble as uncorrelated as possible; the more random the bagging the lower the ensemble prediction error (figure 1.7B). For each tree, the dataset (*i.e.* the subset of the training set) is split m times along its rows from the top node A into two daughter nodes based on a random threshold for a random feature. The split m that results in the highest mean squared error between the first and second daughter nodes' average values Y_1^i and Y_2^i, respectively, is found by calculating

$$Err(A,m) = \frac{1}{n} \sum_{\substack{i=1 \\ Y_1,Y_2 \in m}}^{n} (Y_1^i - Y_2^i)^2 \tag{1.20}$$

for each split where i is the number of rows in each split. Computing the error for both the top node A and the two daughter nodes $A^{'}$ and $A^{''}$ allows for computing the splitting error for the candidate split m:

$$I(m) = Err(A, m) - Err(A', m) - Err(A'', m), \tag{1.21}$$

where *I(m)* is the compiled error of the split candidate. For each node split, the candidate with the highest value of *I(m)* is chosen, and then each daughter node is in turn considered as separate top node and so forth. More intuitively, the chosen candidate split term for A will divide the datapoints (or rows) in A based on a condition for a picked feature. A' will contain the rows for which the condition is true, A'' will contain the rows for which the condition is false, or vice versa. The growing of the tree (*i.e.* the downwards splitting of nodes) is continued until the user-set parameter max depth is reached, which commonly ranges from 10-150 in regression problems or when $Err(A', m) = Err(A'', m)$ or when only one row is present in a node. Unless the max depth is set to the number of points i in the training set, $i-maxdepth > 0$ and thus leaves (*i.e.* final nodes) that contain multiple data points will exist. In RFs, the labels associated with these nodes are averaged to obtain a single label for each leaf in the tree. When training (*i.e.* growing all the decision trees in the ensemble) has been completed, a test set row can be predicted on by simply allowing the row to be passed through the splits in each decision tree; because the features are equal to the training set's features, the decision will be sent into the leaf that best describes the row's feature values. Across decision trees in the ensemble, the prediction for the test row is averaged to produce the model prediction.

Support Vector Machines

The support vector machine (SVM) algorithm consists of an interplay between two components: a linear regression and a kernel trick.[149] Given a training set, the algorithm attempts to linearly separate it across a one-dimensional plane (the 'hyperplane'). Because datasets are rarely linearly separable in their original state, a kernel function is used to map the dataset to a space with *d+1* dimensions, and separation is attempted again (figure 1.7C). This procedure can be repeated to infinity up to the point where the data is linearly separable; this process is called the kernel trick. Although more commonly used in classification, SVMs can also perform regression where instead of a linear separation, a suitable linear regression is sought using the kernel trick mapping procedure to map across dimensions $\mathbb{R}$. Suppose a training set of structure $(x_1, y_1), .., (x_l, y_l) \subset X\mathbb{R}$, where X represents the space of input patterns, consider a linear function f:

$$f(x) = \langle w, x \rangle + b \; with \; w \in X, b \in \mathbb{R}, \tag{1.22}$$

in which $\langle w, x \rangle$ represents the dot product of the flatness w and vector x in a given space $\mathbb{R}^d$. The aim is to find a function $f(x)$ that has at most ε deviation (a hyper-parameter) from the vector y_i in the training data while maximising flatness, *i.e.* minimising w across X such that $f(x) \to b$. By minimising the norm $\|w\|^2 = \langle w, x \rangle$ to ensure flatness across X, the problem can be written as a convex optimisation problem as originally formulated by Vapnik:

$$\begin{aligned} &\text{minimise } \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \\ &\text{subject to } \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \qquad \geq \quad 0 \end{cases} \end{aligned} \tag{1.23}$$

where ξ_i and ξ_i^* are variables that represent the distance per point within and outside the ε margin, respectively and thus the user-set constant C determines the trade-off between the flatness term and the slack term. The slack term creates a soft margin that forgives the algorithm if the points are not perfectly separable; this is added because the optimisation problem cannot always be solved and the only method to solve the problem is to relax the constraints (figure 1.7D). As previously stated, $(x_1, y_1), .., (x_l, y_l)$ is mapped across $\mathbb{R}$ using a kernel function. The choice of kernel function to use can be quite challenging, but the most frequently used in modern machine learning problems is the radial basis function (RBF) kernel - this kernel is used in work throughout this thesis because of its ease of hyperparameter tuning and its documented effectiveness 'out-of-the-box'. Consider a two-dimensional dataset with vectors x and $x^{'}$. A third dimension could be added by simply adding $\langle x, x^{'} \rangle$ as a third vector; however, this becomes increasingly expensive in higher dimensions. The RBF kernel function cheaply computes a Euclidian norm for all dimensions contained by

$$K(x, x^{'}) = \exp(-\gamma \| c - c^{'} \|^2), \tag{1.24}$$

where γ is a hyperparameter between [0-1]. A fourth dimension is computed by solving the Euclidian norm for $x - x^{'} - x''$, *et cetera.*

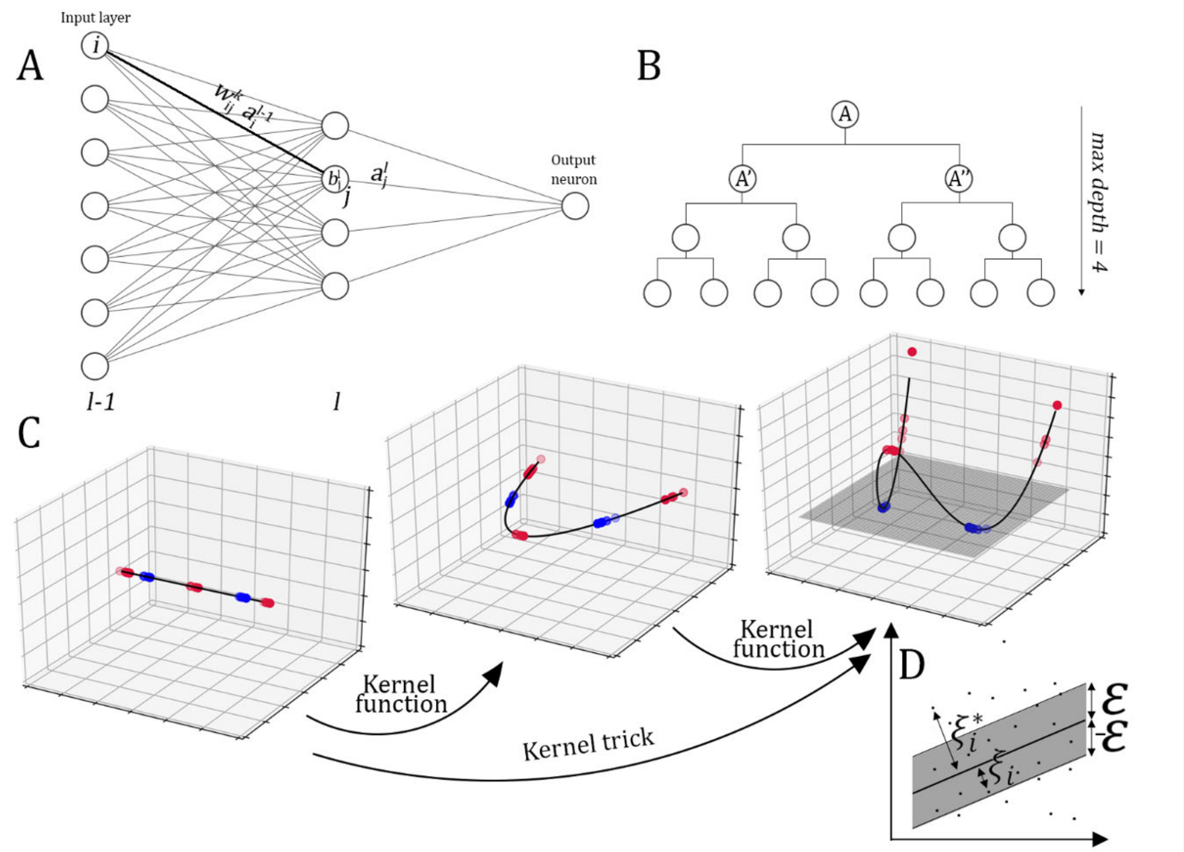


Figure 1.7: Visual representations of the core concepts behind the machine learning algorithms discussed in this chapter. All depicted parameters are explained in the theory section. **A**: a simplified representation of a neural network where each circle represents a neuron (*i.e.* node). **B**: a simplified representation of a decision tree where each circle represents a split point. **C**: mapping the data to higher dimensions allows linear separation of blue and red datapoints (right-hand side plot) as used by support vector machines. **D**: regression performed by support vector machines.

Hyperparameter optimisation

Hyperparameter configurations (e.g. α in NNs, *maxdepth* in RFs, C in SVMs) are pivotal in machine learning model performance. In model optimisation it is common to tune these hyperparameters to search for a seemingly optimal configuration in hyperparameter space (*i.e.* all possible configurations) that minimises the machine learning model (*i.e.* validation error). As with other sampling approaches, the main challenge when sampling hyperparameter space is the trade-off between computational expense and the ability to reach a global minimum. There exist a variety of optimisation techniques to sample hyperparameter space such as manual tuning to fully automated but expensive methods such as grid search, which entails sampling all possible configurations in hyperparameter space.

Neither manual nor grid search is optimal, because the first is unlikely to converge to a global minimum and because the second involves sampling a vast number of hyperparameter configurations for modern ML methods. Random search - sampling n random points in hyperparameter space - greatly reduces the number of explored configurations (and thus expense) but it is still unlikely to converge to a global minimum. More recently, sequential optimisation methods have emerged that sample hyperparameter space more efficiently. One of these methods is Bayesian hyperparameter optimisation (BHO), which attempts to predict which next sample in hyperparameter space would decrease the function (*i.e.* the model's validation error) most and thereby converges to a global minimum with minimal sampling. BHO is summarised as a sequential function in

$$x^* = \arg\min_{x \in X} f_m(\Theta), \tag{1.25}$$

where $f_m(\Theta)$ is the validation error of a given machine learning model given a configuration of hyperparameters Θ, x^* is the specific configuration of hyperparameters that produces the global minimum of the validation error $f_m(\Theta)$ and X is the complete hyperparameter space. Each dimension in X is assumed to be a Gaussian process and is assumed random initially which means the uncertainty $\sigma(\Theta)$ is constant. Conforming to Bayesian statistics, BHO sequentially observes a sample and updates its prior to be its new posterior. Because there is a choice to be made between exploitation (maintain sampling in a low area of $f_m(\Theta)$ to see if there is any further decrease to be gained) and exploration (sampling in areas of high uncertainty, *i.e.* high $\sigma(\Theta)$), acquisition functions are used to balance between these decisions. The most commonly applied acquisition function is expected improvement:

$$EI(\Theta) = \begin{cases} \left(\mu(\Theta) - f_{(m)}(\hat{\Theta})\right) \emptyset(Z) + \sigma(\Theta)\o(Z), & \sigma(\Theta) > 0, \\ 0, & \sigma(\Theta) = 0 \end{cases} \tag{1.26}$$

where

$$Z = \frac{\mu(\Theta) - f_m(\Theta)}{\sigma(\hat{\Theta})}. \tag{1.27}$$

Here, $\mu(\Theta)$ and $\sigma(\Theta)$ are the mean and variance of the posterior $f_m(\Theta)$, $f_m(\hat{\Theta})$ is the validation error of the best model so far and $\emptyset$ and $\o$ are the cumulative distribution function (CDF) and the probability density function (PDF) of the standard normal distribution of $f_m(\Theta)$ values of all hyperparameters present in x at each Θ, respectively. More intuitively, the first summation term in Eq. 1.26 represents exploitation, whereas the second represents exploration. Choosing the next sampling point in X is then a matter of finding the maximum of the expected improvement function:

$$\Theta_{new} = \arg\max_{\Theta} EI(\Theta). \tag{1.28}$$

Sampling is terminated when a user-set limit of samplings is reached. The proposed global minimum in $f_m(\Theta)$ is then proposed as the terminal $f_m(\hat{\Theta}))$.

Chapter 2

A Hybrid Alchemical Free Energy/Machine Learning Methodology for the Computation of Hydration Free Energies

2.1 Introduction

Alchemical free energy calculations (or Free Energy Perturbation -FEP-) are increasingly used in academia and industry to support ligand optimisation problems in the early stage of drug discovery.[150–153] The domain of applicability of current alchemical methodologies has to date mainly been restricted to hit-to-lead and lead optimisation scenarios owing to limitations in computing cost, conformational sampling, and the accuracy of the potential energy functions used to compute protein-ligand energetics[154–159]. There is continued interest in the development of more accurate potential energy functions to benchmark FEP workflows on diverse well-curated protein-ligand datasets,[111,160–162] and for applications to blinded challenges or methodological studies.[163–168]

The calculation of hydration free energies has historically been an important stepping stone towards more accurate forcefields for protein-ligand binding free energy calculations[169–171]. Blinded competitions such as SAMPL (where contenders predict on datasets without experimental reference and predictions are compiled and benchmarked by the organising committee) have also focused on hydration free energy calculations[172]. Forcefield parameterization is a painstaking challenge that requires meticulous and laborious efforts to yield steady gains in accuracy. Recent parameterization efforts from the Open Force Field, AMBER, CHARMM communities have involved multiple groups[173–175]. Recent work has sought to simplify the parameterization process by direct chemical perception of hierarchical parameter types[176]. Nevertheless it can be difficult to identify what modifications to introduce to improve the accuracy of parameter sets. Ultimately fundamental limits in accuracy cannot be overcome due to an incomplete description of the physics of the process, for instance due to use of fixed-charge forcefields that neglect polarisation effects.[177] Notably this realisation has prompted the development of post-processing methodologies based on quantum mechanical (QM) calculations to introduce correction terms for hydration and binding free energies computed by FEP methods using a classical force field.[173–175,178,179]

Data-driven machine-learning (ML) methods have witnessed a resurgence of interest in drug discovery in recent years. Impressive advances have been made in the area of machine learning of quantum chemical calculations,[180,181] virtual screening,[182,183] and free energies of hydration.[184–187] Efforts such as DeepChem[188] and MoleculeNet[189] have popularised the use of ML methods for molecular property predictions. Recent efforts have made use of 3D convolutional neural networks or other graph convolutional neural networks to predict binding affinities from the spatial structure of protein-ligand systems.[190,191] While impressive results have been demonstrated, the performance of ML methods is limited by the requirements of often substantial training sets, and a rapid decrease in accuracy when applying the models to molecules that are dissimilar to those that were included in the training set.

In previous work undertaken by our group as part of the SAMPL6 competition[168] we observed that empirically correcting FEP-derived host-guest binding free energies by a linear regression model calibrated on preceding SAMPL5 submissions,[192] led to significant decrease in mean unsigned error (MUE) of the predicted binding affinities. The present study extends this approach with machine-learning regression models that act as empirical correction terms to the FEP results. That is, the ML models are trained to predict the *mistake* compared to experimental values in Gibbs free energy that alchemical calculations make, referred to from here on as the ΔG_{offset} (figure 2.1).

For any given alchemical prediction ΔG_{FEP} and associated experimental free energy ΔG_{EXP}, ΔG_{offset} is defined as the difference between the two; it also constitutes the training label for the given perturbation. This method relies on the assumption that given a training set of sufficient size, an empirical model trained on this set will be able to estimate accurately ΔG_{offset} values for a new set of alchemical predictions, thereby compensating for systematic errors in the underlying alchemical methodology.

As a proof-of-concept, we explore absolute alchemical calculations of hydration free energies performed with GROMACS.[193] Our results show that the proposed hybrid FEP/ML methodology leads to significant improvements in the accuracy of calculated hydration free energies, whilst only requiring modest training sets compared to a pure machine learning approach, *i.e.* one using ML to directly predict hydration free energies.

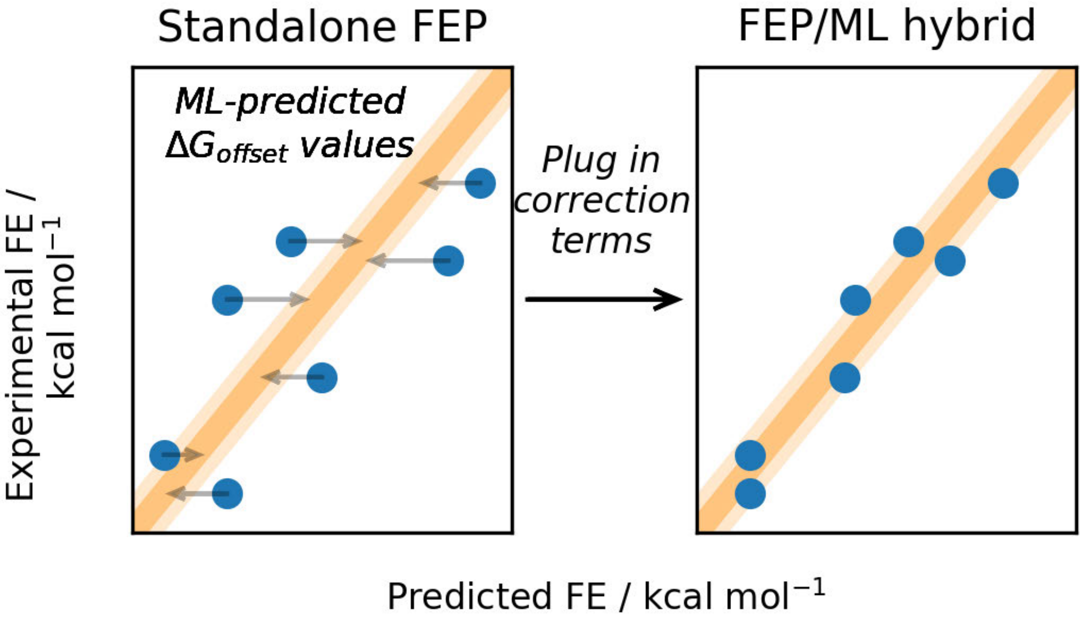

Standalone FEP
FEP/ML hybrid
ML-predicted
ΔG_{offset} values
Plug in
correction
terms
Experimental FE / kcal mol^{-1}
Predicted FE / kcal mol^{-1}

Figure 2.1: Schematic of the work presented in this chapter. With a given absolute Free Energy Perturbation (FEP) prediction and the correlating experimental Free Energy (FE) values per ligand, this project aims to predict the 'mistake' of the FEP versus experimental value per prediction using a machine learning (ML) approach. Using these predicted ΔG_{offset} values, FEP can be hybridised with ML by using the correction terms to correct the standalone FEP predictions, improving the prediction accuracy versus experimental values.

2.2 Theory & methods

FEP/ML model generation

The present methodology describes a regression model that fits the *mistake* that an alchemical calculation makes for a given molecule A, where the mistake is defined by:

$$\Delta G_{offset}(A) = \Delta G_{EXP}(A) - \Delta G_{FEP}(A), \tag{2.1}$$

where $\Delta \mathrm{G}_{FEP}(A)$ is the hydration free energy of molecule A calculated by the alchemical method, and $\Delta \mathrm{G}_{EXP}(A)$ is the experimentally determined hydration free energy for the same molecule. For a given training set with defined descriptors, machine-learning models were used to fit the training domain using five-fold cross-validation over 10 replicates, resulting in a total population N_{pop} of 50 trained models (see methods section below). All individual models in N_{pop} are regression models predicting their own $\Delta\hat{G}_{offset}$ value. We define our offset estimator as the arithmetic mean of these offset values, and use the standard deviation of the mean as a measure of the precision of the calculated offset. Thus we define a corrected hydration free energy as:

$$\Delta G_{FEP/ML}(A) = \Delta G_{FEP}(A) + \langle \Delta\hat{G}_{offset}(A)\rangle_{N_{pop}}. \tag{2.2}$$

and the precision of the $\Delta \mathrm{G}_{FEP/ML}(A)$ estimate is determined by propagating statistical errors of the alchemical and ML terms.

Dataset acquisition

Version 0.52 of the FreeSolv database[194] was downloaded from `https://github.com/MobleyLab/FreeSolv`. This version contains 642 small neutral molecules. Aside from experimentally-determined values, the database contains absolute free energies of hydration computed from alchemical simulations using GROMACS.[193] A detailed description of the particular FEP methodology used can be found in Ramos Matos et al. [195] FreeSolv calculations were performed using the GAFF[196] force field, AM1-BCC[197] partial charges and the TIP3P water model.[198,199]

A dataset split was performed by excluding the FreeSolvSAMPL4 set which contains all the compounds (n=47) that were used in the SAMPL4 blinded competition (and had been subsequently appended to the FreeSolv database after this challenge).[200] Compounds belonging to this set were extracted by filtering for the keyword 'SAMPL4_Guthrie' in the experimental reference column of the database's overview textfile. Six molecules (mobley_6309289, mobley_3395921, mobley_6739648, mobley_2607611, mobley_637522 and mobley_172879) were added manually to the test set because even though they were present in the SAMPL4 challenge they were not tagged with this keyword in v0.52 of the FreeSolv database. This resulted in a training set of 595 molecules. From here on only the training set will be described, but all treatment of data can be considered equal between the training and test set unless otherwise indicated. All data-handling was done in Python 3.7.4.

Feature generation & pre-processing

Features (descriptors) were generated for all compounds present in FreeSolv. The ML models in this study were generated using RDKit 2019.03.4.0.[201] Molecules were loaded using the provided SDF files, and featurized using the following classes on standard settings unless indicated otherwise:

- *APFP*: Atom-pair fingerprints were generated using `rdkit.Chem.rdMolDescriptors.GetHashedAtomPairFingerprint()`; length was set to 256.

- *ECFP*: Extended-connectivity fingerprints were generated using `rdkit.Chem.AllChem.GetMorganFingerprintAsBitVect()`; length was set to 1024. In order to generate fingerprints with diameters ECFP2/4/6/8, the radius was set to 1, 2, 3 and 4, respectively.

- *TOPOL*: Topological fingerprints were generated using `rdkit.Chem.RDKFingerprint()`; length was set to 1024.

- *MolProps*: Molecular properties were generated using the Mordred python

API[202] with inclusion of 3D properties. Although the total number of descriptors that this API generates is 1825, non-numeric columns were excluded resulting in 1113 properties that constitute the features per compound. This particular molecular properties generator was chosen owing to the large number of molecular properties readily computed via its API.

- *X-NOISE*: Noise 'fingerprints' were generated using `NumPy.random.randint()`; length was set to 100 and random integers ranged between 0-100.

The 'X-NOISE' feature was added to act as a negative control as random values should not be able to produce a predictive model. Additionally, all fingerprints were appended individually to MolProps features (resulting in for instance a feature set called 'MolPropsAPFP' which was obtained by appending 'APFP' to 'MolProps') resulting in fingerprints with a length of the sum of both feature sets (in the case of MolPropsAPFP, 1113 + 256 = 1369). Every feature set was subsequently Z-normalized to zero mean and `sklearn.decomposition.PCA` was used to reduce dimensionality using a principal component analysis, and retaining principal components contributing up to 95% of the variance. Through this, the resulting dimensions were the principal vectors rather than original features exhibiting large enough variance.

After data pre-processing, the corresponding label (G_{offset}, see Eq. 2.1) was appended to each data point in order to build the final training set (named 'FEP/ML'). Additionally, a second training set (named 'ML') was generated by using as labels (output variables) the experimentally-determined ΔG_{exp} value for each data point.

A 5-fold cross-validation approach was chosen to reduce the risks of overfitting the training set. The training set was thus randomly split into five equally-sized folds (of sizes 595/5=119). Training was repeated five times, rotating the folds so that each fold acted as the validation set once for the other four training set folds. Additionally, training was performed with 10 replicates per feature set, resulting in a total of 50 trained models per feature set-ML model combination.

Machine-learning models

Scikit-Learn 0.11.1[203] was used to generate all ML models. The models were generated on a machine running Ubuntu 18.04.3 LTS containing 10 Inter i9-7900X CPU cores. For Support vector machines (SVMs), random forests (RFs), deep neural networks (DNNs) and multiple linear regressions (MLRs), the classes `sklearn.svm.SVR`, `sklearn.ensemble.RandomForestRegressor`, `sklearn.neural_network.MLPRegressor` and `sklearn.linear_model.LinearRegression` were used on standard settings except for DNN which used `max_iter=5000`.

In order to choose optimal hyperparameter configurations for each ML model, a Bayesian hyperparameter optimization routine was adopted using SciKit-Optimize 0.5.2 (SKOPT)[204], which makes use of an *expected improvement* acquisition function to search hyperparameter space more efficiently than a random or grid search. The number of steps (*calls* in SKOPT nomenclature) was set to 50 because convergence was observed before this point in most cases. After training a call, the cost function (mean absolute error of predicting on the validation set) across folds is returned to the SKOPT decorator which in turn chooses a new hyperparameter configuration for the next call using its acquisition function to attempt to further decrease the model's cost function. A more detailed description of the algorithm can be found in the online SKOPT documentation. Note that this means that for any ML model, each of the 10 replicates had its own configuration of hyperparameters, but within each replicate all five folds would have the same hyperparameter configuration. The complete hyperparameter space is described in table 2.1. Approximate runtimes for the complete training protocols were, for SVM, MLR, DNN and RF, 10h, 25h, 104h and 134h, respectively.

The code to reproduce all key results and figures presented in this manuscript is available at `https://github.com/michellab/hybrid_FEP-ML`.

2.3 Results & discussion

Protocol optimization on training set

For all the ML models derived in this study it was observed that hyperparameters played an important role in model validation accuracy. This is likely due to the relatively small size of the training set (595 datapoints). Thus a hyperparameter optimization algorithm was adopted in which hyperparameters were tuned with the help of Bayesian optimization based on Gaussian process regression (see table 2.1). This algorithm searches through hyperparameter space by wrapping around noisy, expensive ML functions; after 50 calls (configuration attempts), the hyperparameter configuration returning the lowest validation error is saved together with the corresponding trained model. For SVM, RF and DNN models convergence was observed from around 30 calls. MLR in this case does not have any hyperparameters to tune which means that in every SKOPT call the same model is trained which results in an equal validation error along calls.

Table 2.1: Hyperparameter descriptions of all machine-learning algorithms used in the study. SVM, RF, DNN and MLR are support-vector machines, random forests, deep neural networks and multiple-linear regressions, respectively. Total configurations are computed by multiplying the number of values per hyperparameter for each ML model.

ML model	**Hyperparameter**	**Range**	**Total configurations**
SVM	C	1e-3, 1e-2, ..., 1e+2	216
	ϵ	1e-3, 1e-2, ..., 1e+2	
	γ	1e-3, 1e-2, ..., 1e+2	
RF	NumEstimators	1, 2, ..., 1000	9e+4
	MaxDepth	1, 2, ..., 5	
	MinSamplesSplit	2, 3, ..., 10	
	Bootstrap	True, False	
DNN	ActivationFn	logistic, tanh, relu	3.1e+6
	Solver	lbfgs, sgd, adam	
	Layers*	(100,50),(50,20), (100,100,50), (100,50,20), (50,20,5)	
	Adam-β1	0.1, 0.2, ..., 0.99	
	Adam-β2	0.1, 0.2, ..., 0.99	
	Adam-ϵ	10e-8, 10e-7, ..., 10e-1	
MLR	No hyperparameters to tune.		1

**For the 'Layers' hyperparameter, the standard SKLearn tuple-input form is given where the i-th element represents the number of neurons in the i-th hidden layer.*

Based on the training protocol it can be observed that random forests (RF) and multiple linear regressions (MLR) do not fit the training set as well as support vector machines (SVM) and deep neural networks (DNN) protocols (see figure 2.2). For MLR this is to be expected because of the relative simplicity of the model. Although the RF algorithm is more complex, it is primarily designed for classification problems rather than regression problems due to its dependence on decision trees, which may explain its underfitting. The algorithm is included as a control in the current study.

A range of different feature sets was used to identify efficient encodings for describing ΔG_{offset}. A general trend in feature set performance can be observed across ML models. MolProps and combinatorial feature sets (fingerprints appended to MolProps) fit the training set better than standalone fingerprints (APFP, TOPOL and ECFP6), and X-NOISE performs worst as expected since this feature set is generated from random data.

Because standalone MolProps generally outperform standalone fingerprints, it is likely that the combined feature sets benefit mainly from the more predictive MolProps component. The observation that MolProps appears to outperform other feature sets suggests some of the descriptors (*e.g.*, molecular weight and polar surface area) included in MolProps correlate well with free energies of hydration. This is reinforced by our observation that the MolProps feature set outperforms generally other feature sets when predicting ΔG of hydration directly in our pure ML models (see figure 2.2).

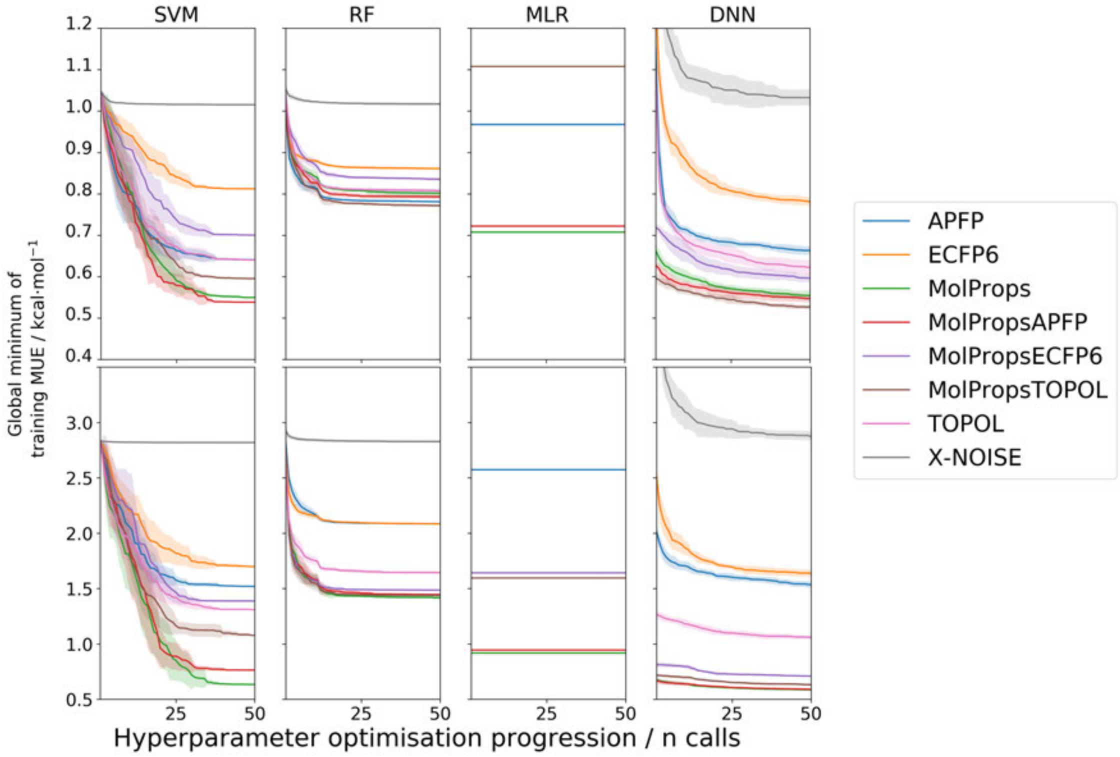


Figure 2.2: Hyperparameter optimisation of machine-learning models fitting on ΔG_{offset} (top row) and ΔG (bottom row) for different feature types computed for compounds in the FreeSolv database. These trained models are subsequently named FEP/ML and pure-ML (ML) models in the main text body. Depicted are the number of hyperparameter calls versus global minima of training validation mean unsigned error in kcal·mol^{-1}. The shaded regions indicate the standard deviation across ten replicates. Note that in the case of MLR several lines fall above the depicted error range.

Although the Extended-connectivity fingerprint (ECFP)[205] is used extensively in QSAR regression problems, our training protocol suggests underfitting of the training set for this feature type. This is likely because the used diameter of six bonds is too large to accurately discriminate between the relatively small compounds in the FreeSolv database (see figure 2.3); testing with smaller diameters suggests an increase in fitting ability, however these models still underperform with respect to other feature types (see figure 2.4).

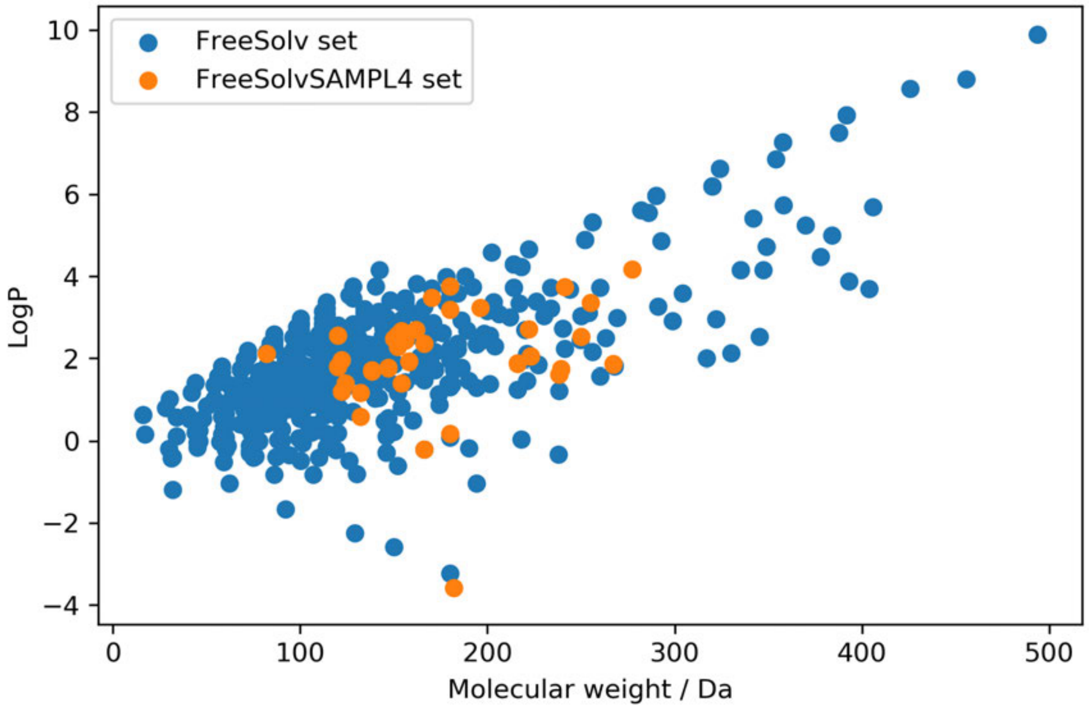


Figure 2.3: Molecular characteristics of the FreeSolv database (blue) with the contained FreeSolvSAMPL4 set (orange). Depicted is molecular weight in daltons versus log-partition coefficient per molecule in the database. Both properties were calculated using RDKit 2019.03.4.0.

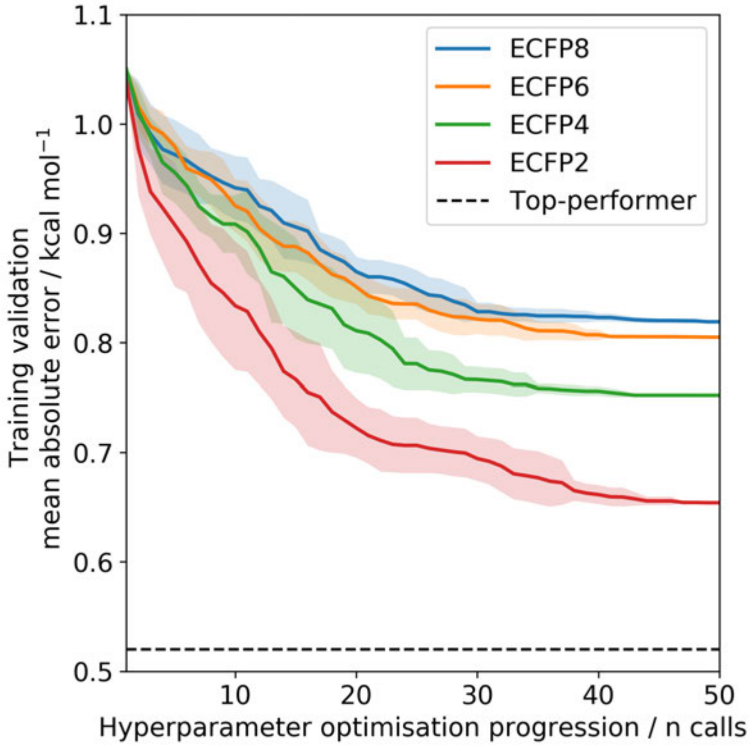


Figure 2.4: Hyperparameter optimisation of support-vector machine models fitting on ΔG_{offset} for different ECFP diameter sizes computed for compounds in the FreeSolv database. Depicted are the number of hyperparameter calls versus cumulative minima of training validation mean unsigned error in kcal·mol^{-1} for extended-connectivity fingerprint diameters 2, 4, 6 and 8. Error regions are computed as standard deviation across ten replicates. The black dashed line indicates the converged validation error for the top-performing model in the main text body (SVM-MolPropsAPFP).

Hybrid FEP/ML models outperform standalone FEP and ML models in SAMPL4

The trained models were used to predict on the Freesolv-SAMPL4 test set. Because low errors in training validation do not necessarily translate into low errors in testing validation, all trained models were tested (see figure 2.5 and table 2.2). Top-performing models per ML model (see figure 2.6) were based primarily on the MolProps feature set for SVM, RF and MLR, but not for DNN. It is likely that the latter suffers from a degree of overfitting causing individual models to differ widely in predicted offset values. This is apparent in the much larger uncertainties in dataset metrics for DNN. It could also be that there exist several models with comparable performance (*i.e.*, local minima). Nevertheless the accuracy of the predictions obtained by averaging over the 50 DNN models is competitive. Overall SVM appeared to give more consistently accurate and precise estimates of ΔG_{offset} values.

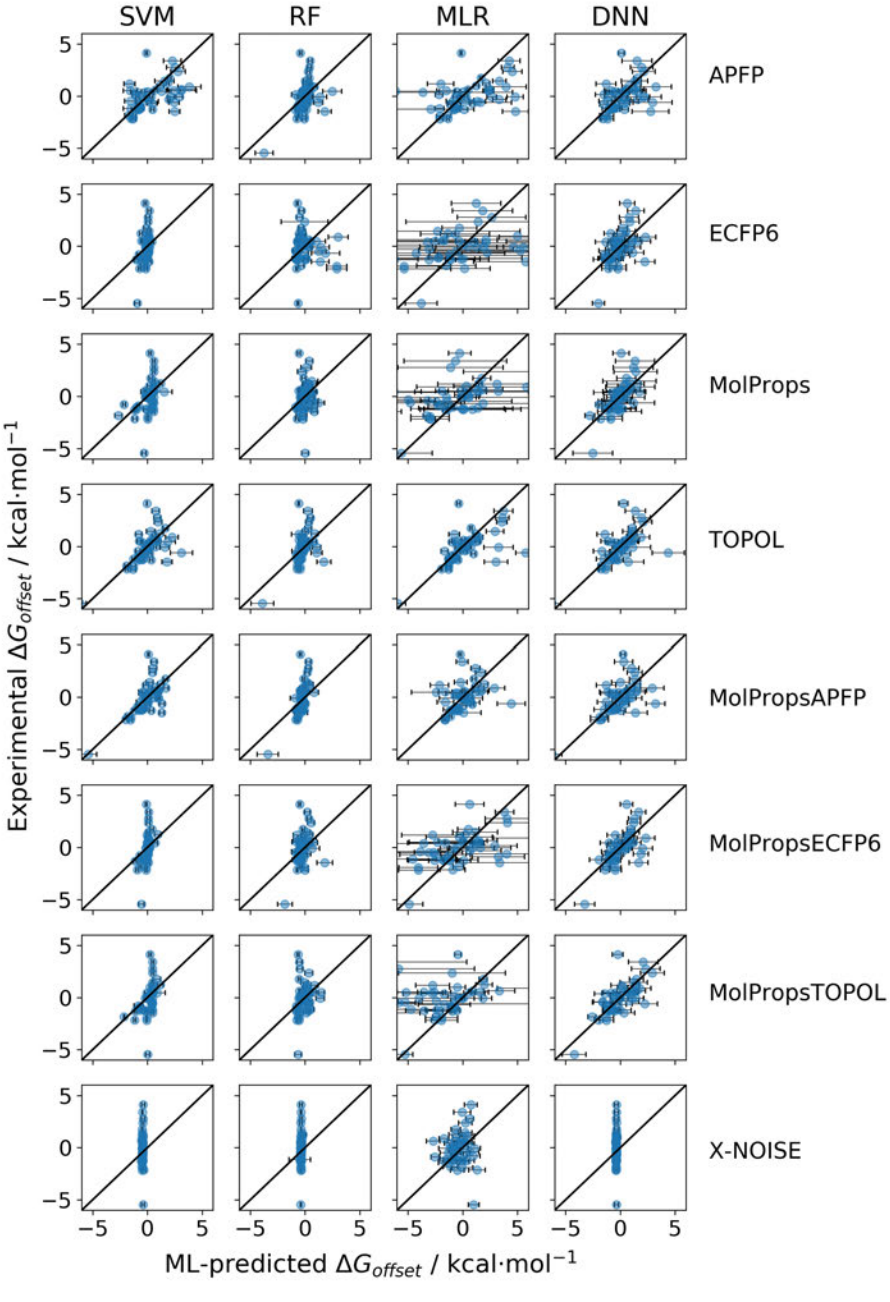


Figure 2.5: Machine-learning ΔG_{offset} predictions versus experimental ΔG_{offset} values for all feature types on the FreeSolvSAMPL4 test set. Depicated are mean predictions across 10 replicates. SVM, RF, MLR and DNN are support-vector machines, random forests, multiple-linear regressions and deep neural networks, respectively. Standard deviations are depicted as error bars.

Table 2.2: Key statistics for machine-learning models combined with FEP (FEP/ML entries) and pure machine-learning models predicting ΔG directly (ML entries), sorted by MUE in ascending order. Uncertainties per statistic are given as plus-minus entries. For these results, the mannitol outlier (mobley_4587267) has been removed. Uncertainties per statistic are given as plus-minus entries. MUE (mean unsigned error) and RMSE (root mean-squared error) are shown in kcal·mol^{-1}.

Model	Featureset	Type	Pearson r	MUE	RMSE	Spearman rho	Kendall tau
FEP			0.92±0.0	1.07±0.04	1.9±-0.49	0.86±0.0	0.71±0.0
DNN	TOPOL	FEP/ML	0.94±0.12	0.75±1.09	1.07±1.03	0.92±0.17	0.79±0.21
SVM	MolPropsAPFP	FEP/ML	0.94±0.01	0.78±0.22	1.31±0.0	0.95±0.02	0.86±0.05
DNN	MolPropsTOPOL	FEP/ML	0.93±0.08	0.79±0.7	1.21±0.54	0.94±0.09	0.81±0.14
DNN	MolProps	FEP/ML	0.91±0.07	0.8±0.56	1.58±0.22	0.95±0.06	0.83±0.11
SVM	MolPropsTOPOL	FEP/ML	0.94±0.01	0.83±0.13	1.38±-0.11	0.92±0.02	0.78±0.03
DNN	MolPropsECFP6	FEP/ML	0.93±0.06	0.83±0.58	1.31±0.34	0.94±0.12	0.81±0.15
SVM	MolProps	FEP/ML	0.91±0.03	0.87±0.23	1.68±-0.15	0.94±0.01	0.82±0.03
SVM	MolPropsECFP6	FEP/ML	0.94±0.01	0.9±0.14	1.54±-0.2	0.9±0.02	0.76±0.02
SVM	TOPOL	FEP/ML	0.93±0.01	0.9±0.16	1.53±-0.16	0.9±0.03	0.74±0.04
RF	MolPropsAPFP	FEP/ML	0.93±0.01	0.93±0.19	1.53±-0.15	0.88±0.02	0.74±0.04
DNN	MolPropsAPFP	FEP/ML	0.91±0.06	0.94±0.58	1.76±0.1	0.94±0.09	0.81±0.13
MLR	MolProps	FEP/ML	0.85±0.07	0.94±0.31	2.52±-0.59	0.89±0.06	0.76±0.07
DNN	ECFP6	FEP/ML	0.91±0.08	0.96±0.76	1.58±0.33	0.88±0.13	0.73±0.14
SVM	ECFP6	FEP/ML	0.93±0.01	0.96±0.17	1.71±-0.28	0.86±0.02	0.71±0.03
RF	MolProps	FEP/ML	0.91±0.02	0.98±0.22	1.81±-0.27	0.91±0.03	0.76±0.03
RF	MolPropsECFP6	FEP/ML	0.91±0.03	1.02±0.27	1.91±-0.3	0.87±0.04	0.73±0.04
RF	MolPropsTOPOL	FEP/ML	0.91±0.02	1.03±0.2	2.0±-0.42	0.87±0.03	0.73±0.05
DNN	APFP	FEP/ML	0.89±0.09	1.03±0.72	2.07±0.09	0.91±0.14	0.75±0.15
RF	TOPOL	FEP/ML	0.91±0.03	1.05±0.32	1.95±-0.3	0.85±0.04	0.7±0.04
RF	APFP	FEP/ML	0.91±0.02	1.06±0.19	1.93±-0.35	0.9±0.04	0.75±0.05
RF	X-NOISE	FEP/ML	0.92±0.01	1.07±0.12	2.02±-0.51	0.86±0.02	0.71±0.02
DNN	X-NOISE	FEP/ML	0.92±0.01	1.08±0.17	2.03±-0.47	0.86±0.03	0.71±0.03
SVM	X-NOISE	FEP/ML	0.92±0.01	1.09±0.1	2.09±-0.57	0.86±0.02	0.71±0.01
MLR	MolPropsAPFP	FEP/ML	0.89±0.06	1.11±0.49	2.42±-0.33	0.91±0.05	0.78±0.06
SVM	APFP	FEP/ML	0.88±0.04	1.14±0.44	2.51±-0.51	0.89±0.05	0.72±0.07
MLR	X-NOISE	FEP/ML	0.9±0.07	1.22±0.6	2.35±-0.3	0.81±0.1	0.65±0.1
RF	ECFP6	FEP/ML	0.86±0.04	1.34±0.38	3.45±-1.18	0.81±0.08	0.65±0.09
MLR	APFP	FEP/ML	0.74±0.21	1.61±0.82	5.09±-1.55	0.72±0.12	0.56±0.09
MLR	MolPropsECFP6	FEP/ML	0.62±0.27	1.98±1.93	7.88±-3.18	0.65±0.24	0.51±0.21
MLR	TOPOL	FEP/ML	0.73±0.2	2.35±2.84	13.61±-6.66	0.64±0.36	0.48±0.27
MLR	ECFP6	FEP/ML	0.63±0.27	2.73±4.53	12.87±-3.97	0.6±0.19	0.44±0.16
MLR	MolPropsTOPOL	FEP/ML	0.78±0.01	2.82±1.97	19.44±-12.34	0.75±0.05	0.6±0.04
SVM	MolProps	ML	0.88±0.06	0.9±0.35	1.89±-0.19	0.89±0.04	0.73±0.06
SVM	MolPropsAPFP	ML	0.8±0.05	1.01±0.28	3.04±-1.11	0.87±0.06	0.71±0.07
MLR	MolProps	ML	0.88±0.04	1.06±0.31	2.88±-0.86	0.89±0.05	0.74±0.06
DNN	MolProps	ML	0.87±0.1	1.1±0.67	2.32±-0.16	0.81±0.11	0.66±0.12
DNN	MolPropsAPFP	ML	0.82±0.1	1.3±0.7	3.53±-0.98	0.79±0.16	0.63±0.17
DNN	MolPropsECFP6	ML	0.79±0.26	1.33±0.88	3.25±-0.61	0.81±0.26	0.64±0.24
RF	MolPropsAPFP	ML	0.72±0.09	1.33±0.53	4.15±-1.64	0.71±0.07	0.56±0.07
RF	MolPropsECFP6	ML	0.71±0.15	1.38±0.49	4.26±-1.82	0.71±0.16	0.54±0.15
RF	MolProps	ML	0.76±0.17	1.42±0.51	4.01±-1.59	0.79±0.15	0.6±0.17
DNN	MolPropsTOPOL	ML	0.78±0.21	1.46±1.08	4.43±-1.3	0.72±0.16	0.57±0.16
SVM	APFP	ML	0.76±0.19	1.48±1.18	4.05±-0.8	0.71±0.18	0.53±0.16
SVM	MolPropsTOPOL	ML	0.61±0.07	1.55±0.2	5.23±-2.83	0.76±0.25	0.54±0.19
MLR	MolPropsAPFP	ML	0.88±0.06	1.57±0.71	6.31±-2.95	0.9±0.03	0.76±0.06
DNN	TOPOL	ML	0.67±0.42	1.67±1.72	4.98±-1.09	0.64±0.43	0.47±0.32
SVM	MolPropsECFP6	ML	0.6±0.06	1.69±0.13	5.98±-3.45	0.78±0.08	0.57±0.08
RF	TOPOL	ML	0.48±0.13	1.74±0.3	6.84±-3.99	0.54±0.14	0.39±0.13
RF	APFP	ML	0.7±0.18	1.75±0.37	4.93±-2.35	0.52±0.14	0.38±0.1
DNN	APFP	ML	0.77±0.18	1.75±1.71	6.07±-1.94	0.66±0.21	0.5±0.18
SVM	ECFP6	ML	0.58±0.11	1.76±0.23	5.72±-3.17	0.64±0.18	0.45±0.14
RF	MolPropsTOPOL	ML	0.56±0.19	1.88±0.51	6.13±-3.09	0.55±0.16	0.41±0.14
DNN	ECFP6	ML	0.59±0.59	1.88±2.1	5.63±-1.15	0.62±0.56	0.46±0.38
SVM	TOPOL	ML	0.57±0.12	1.89±0.42	5.82±-3.06	0.7±0.34	0.49±0.26
RF	ECFP6	ML	0.4±0.18	2.17±0.49	8.12±-4.74	0.44±0.13	0.34±0.11
DNN	X-NOISE	ML	0.1±0.64	2.49±0.63	11.52±-7.64	-0.01±0.56	-0.0±0.4
RF	X-NOISE	ML	-0.22±0.38	2.5±0.44	11.97±-8.2	-0.29±0.35	-0.19±0.26
SVM	X-NOISE	ML	0.08±0.71	2.56±0.13	12.12±-8.54	0.11±0.72	0.08±0.49
MLR	APFP	ML	0.73±0.12	2.94±2.26	17.08±-9.15	0.65±0.13	0.48±0.1
MLR	MolPropsECFP6	ML	0.44±0.21	2.99±2.5	19.11±-12.01	0.48±0.21	0.34±0.16
MLR	X-NOISE	ML	-0.01±0.19	3.1±1.42	15.66±-10.52	-0.08±0.19	-0.06±0.09
MLR	MolPropsTOPOL	ML	0.75±0.02	5.6±2.12	78.19±-67.0	0.69±0.05	0.54±0.05
MLR	ECFP6	ML	0.35±0.17	7.42±10.29	116.45±-95.16	0.35±0.31	0.24±0.23
MLR	TOPOL	ML	0.21±0.07	8.81±5.83	178.3±-157.46	0.33±0.06	0.24±0.05

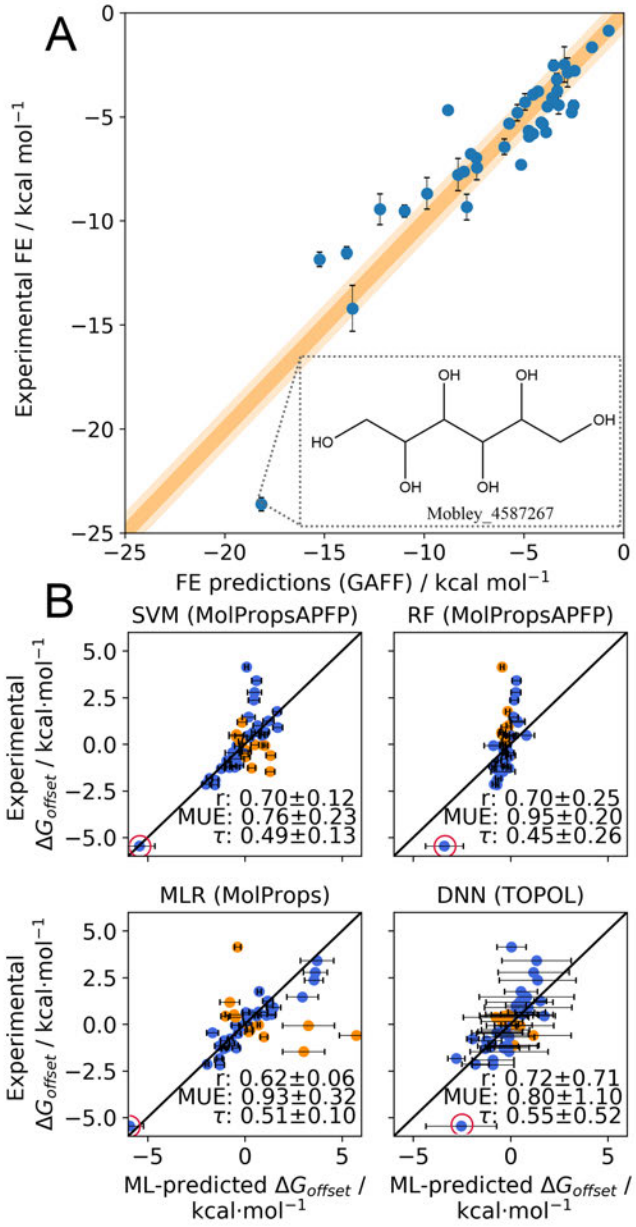


Figure 2.6: Overview of prediction results on the SAMPL4-Freesolv test set. **A:** FEP-predicted versus experimentally-determined free energies of hydration in kcal·mol^{-1}. The orange and light-orange areas are confidence regions for 1 and 2 kcal·mol^{-1}, respectively. Statistical uncertainties as supplied by the authors are shown as errorbars. **B:** Scatter plots of top-performing ML models predicting ΔG_{offset} for the FreeSolvSAMPL4 set with respective statistical intervals. Corrections with correct directionality (*i.e.* when $\langle \Delta \hat{G}_{offset} \rangle_{N_{pop}}$ and ΔG_{offset} values are both positive or both negative) are shown in blue; Corrections with incorrect directionality are shown in orange. The error bars on x-axis values denote the standard error of the mean offset value from ensembles of 50 ML models. Black diagonal lines show the $x = y$ diagonals. Red circles annotate the outlier discussed in the text body.

One compound in the test set (mobley_4587267, (2R,3R,4R,5R)-hexane-1,2,3,4,5,6-hexol, referred to as mannitol from hereon) stands out with a free energy of hydration significantly more negative than other compounds in the test set ($\sim$ -24 kcal·mol^{-1}). This compound has a large associated ΔG_{offset} value of $\sim$-5 kcal·mol^{-1} (figures 2.6A and 2.6B, resp.). SVM and MLR models appear to correct this outlier better than RF and DNN models do, and it is likely that this outlier correction skews the statistical performances of the four models to a degree (see table 2.2 for model performances excluding the outlier); indeed, when plugging in the correction terms (figure 2.7), FEP/ML FE predictions for mannitol appear to be close to experimental hydration free energy measures, especially for SVM and MLR models.

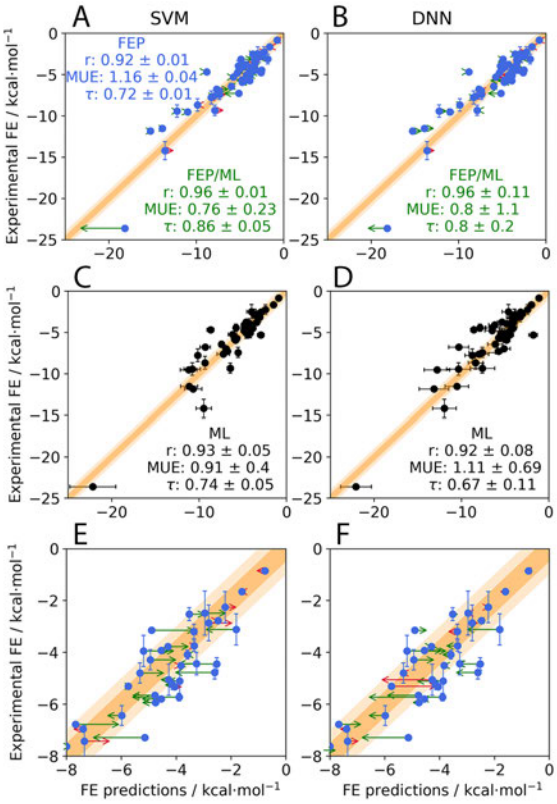


Figure 2.7: Machine-learned correction terms applied to FEP predictions. Results are shown for both support vector machine (left column and deep neural network (right column) ensembles. **A/B:** The FreeSolvSAMPL4 set FEP predictions (figure 2.6) with corrections as predicted by ML models shown with arrows. Green/red arrows depict corrections that improve/worsen agreement with experiment. Statistics for standalone FEP (blue) and hybrid FEP/ML (green) are shown. **C/D:** pure machine-learning (ML) models directly predicting ΔG of hydration with statistics in black text. **E/F:** contains the same data as A/B, but with a smaller range on both axes. Model uncertainties are shown as error bars. For all statistics the uncertainties are shown with a plus-minus sign.

The top-performing FEP/ML model (SVM; MolPropsAPFP, Figure 2.7A) outperformed standalone FEP in Pearson r, MUE and RMSE statistics and had higher ranking statistics (Spearman ρ and Kendall τ) than standalone FEP (see table 2.2). The top-performing FEP/ML DNN model achieves similar accuracy, but introduces significant uncertainties compared to FEP (Figure 2.7B). This reflects the larger uncertainties of the DNN-derived offset values in comparison with other ML protocols (see Figure 2.6B). Even when offset predictions for a given model are of modest accuracy, plugging in the correction term results in a FEP/ML model free energy prediction that performs equally well than the standalone FEP component. It seems that instead of predicting increasingly random values, the worse ΔG_{offset} predictor models converge towards predicting the training set mean offset value (-0.32 kcal·mol^{-1}) for all compounds (see table 2.2 X-NOISE entries). This is significant because it implies that, given that a properly-trained model is used, the correction term can be applied confidently to FEP datasets with minimal risk of worsening the model performance. The exception to this observation is MLR, which appears to occasionally predict high ΔG_{offset} values. This was confirmed by high training validation values in figure 2.2 and bottom-level FEP/ML entries in table 2.2.

The top-performing ML model (SVM; MolProps, Figure 2.7C) achieves accuracy similar to FEP, but with larger uncertainties. This trend worsens for the top-performing DNN model (Figure 2.7D). As noted before, mannitol contributes substantially to model performance: a second table with statistical performances excluding mannitol can be found in table 2.2. Indeed, excluding this compound slightly diminishes the gain in performance when comparing FEP/ML models to standalone FEP, although ranking statistics seem to benefit equally well from correction compared to when mannitol is included. This suggests that the small corrections (figure 2.7E and F) introduce primarily a correct reordering of compound ΔG values.

The top-performing FEP/ML (SVM; MolPropsAPFP)and ML (SVM; MolProps) models were introduced in the SAMPL4 challenge retrospectively (figure 2.8) to correct the results of SAMPL4 submission 004 that featured a FEP protocol most similar to the one used to generate calculated FEP values in FreeSolv. In line with the results obtained on the FreeSolvSAMPL4 test set, FEP/ML SVM models trained with MolPropsAPFP outperformed standalone FEP for all SAMPL4 statistics. For all metrics the gains are significant, moving the FEP/ML prediction to 1st or 2nd rank as judged by MUE, r or Kendall tau metrics, and from 28th to 4th position as judged by RMSE. Many of the top-performing methods have very similar performance within statistical uncertainties, so care must be taken not to overinterpet changes in rankings. Nevertheless it is clear that the ML-derived correction terms improve the accuracy of the FEP methodology.

ML performed broadly similarly to FEP, but the uncertainty of the metrics is again remarkably large. This indicates that there is significant variability in the predicted free energies of hydration of the same compound by the ensemble of ML models. By contrast the FEP/ML predictions are of similar precision to the FEP predictions as the uncertainties in the offset terms is comparable or smaller to the uncertainties in the alchemical estimates.

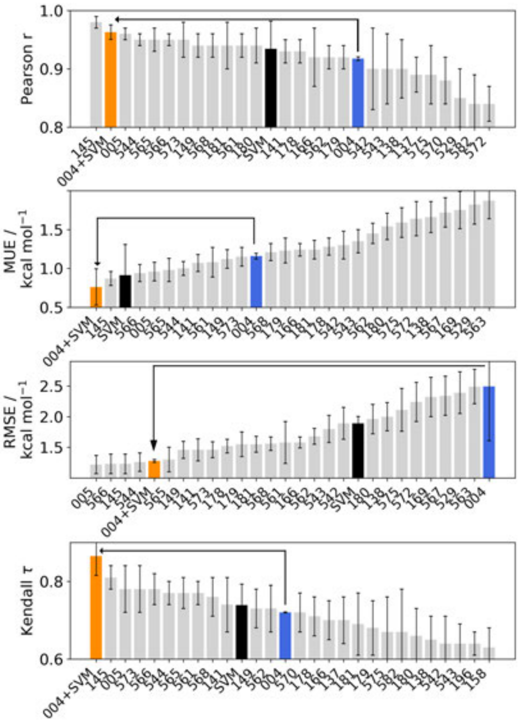


Figure 2.8: SAMPL4 statistical performances of top-ranked entries with inserted pure ML and FEP/ML predictions as depicted in the original challenge. Entry 004 (standalone FEP) is shown in blue. The FEP/ML model is shown in orange, and rank gains between standalone FEP and FEP/ML are depicted as black arrows. Pure ML models (ML) are shown as black bars. Error bars show model uncertainties as depicted in the SAMPL4 overview publication.[200]

Influence of training set size on accuracy of correction terms

We also evaluated the impact of training set size on the accuracy of the correction terms (figure 2.9). Hyperparameter configurations were taken from top performers in the training phase of this study (see table 2.1), and increasingly large, randomly sampled subsections of FreeSolv (exluding the test set) were used were used as training sets. For simplicity only SVM (trained using MolPropsAPFP) results are shown as this model consistently outperformed all others.

It was observed that with training sets of increasing size the cost function (in this case, MUE of FEP/ML prediction on SAMPL4 in kcal·mol^{-1}) decreases monotonically. FEP/ML models appear to outperform standalone FEP after being trained on *ca.* 20 compounds in FreeSolv (figure 2.9A), and converge with training sets of *ca.* 400 compounds. Strikingly, standalone ML models require much larger training sets of *ca.* 450 compounds to outperform standalone FEP. In both cases the gradual decrease in uncertainty with increase in training set size is due to higher overlap in training sets composition between replicates as the full training set size (n=595) is approached. Whereas the FEP/ML model seems to converge at *ca.* 400 compounds, the ML model does not appear to have converged and could likely benefit from a larger training set. This indicates that, given a sufficiently large dataset, a pure ML model may not require any prior FEP calculations.

To put these results in perspective in the context of SAMPL4, the changes in ranks of the FEP/ML entry was plotted as a function of training set size (figure 2.9B). FEP/ML models outperform standalone FEP for all statistical measures, although some variability is observed. Whereas MUE and Kendall τ already show clear improvements from small training set sizes (*ca.* 100 and 50, resp.), Pearson r and RMSE appear to require models trained on a larger number of compounds to reach placement in the top five ranks of the SAMPL4 challenge (250 and 500, resp.).

A top-ranked result by Pearson r is not achieved even with a full training set of 595 compounds. This is also apparent in figure 2.8, where entry 145 is shown to outperform the FEP/ML model. This entry consists of a quantum-mechanical-based method with implicit solvent and applies an empirical correction term to alcohol, ether, ester, amines and aromatic nitrogen groups which were derived from experimental data.[206] It is difficult to compare correction terms in this case because these corrections are generated from experimental measures versus Poisson-Boltzmann-based free energy calculations.

Although FEP/ML hybridisation does not appear to benefit RMSE scores in figure 2.8, the RMSE ranking for FEP/ML models appear to approach first place in the SAMPL4 challenge when trained on the full training set (595 compounds). The working model in figure 2.8 is trained using a cross-validation approach which effectively limits training set sizes to $0.8 * 595 = 476$ compounds which suggests that when generating a definitive ML correction term it would be preferable to use all 595 compounds as a training set.

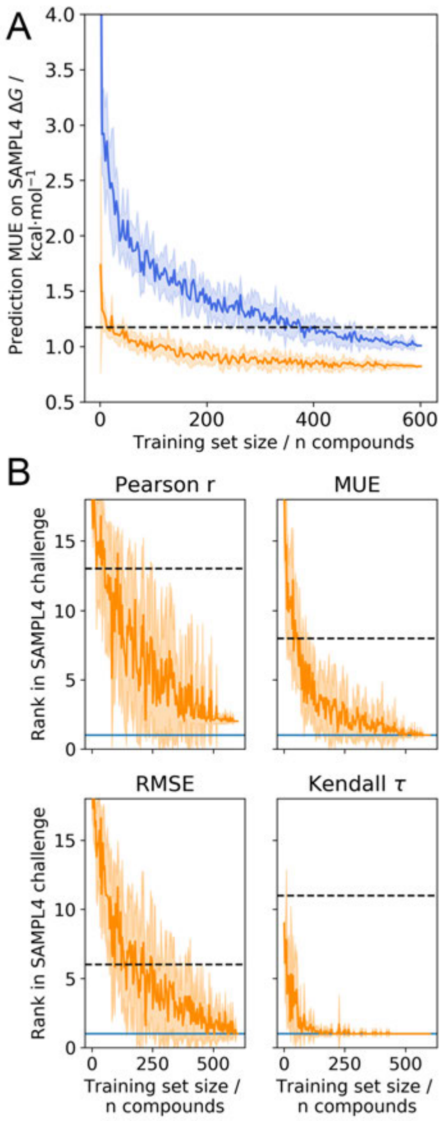


Figure 2.9: Effect of increasing training set size on machine-learned correction models. Results depicted are produced by support vector machines trained using MolPropsAPFP and MolProps for FEP/ML and pure ML models, respectively. **A:** FEP/ML model mean unsigned errors in the SAMPL4 challenge are shown with increasingly large (randomly sampled) subsets of the FreeSolv database as training sets with uncertainties across replicates (n=10) shown as lighter-shaded regions. Orange and blue lines are FEP/ML (FEP+ML, trained on ΔG_{offset}) and pure ML (trained on ΔG) predictors, respectively. Horizontal dashed line indicates the standalone FEP MUE of the FreeSolvSAMPL4 set in the SAMPL4 challenge. **B:** results for the same experiment as *A* but with ranking position of the FEP/ML model in the SAMPL4 challenge on the y axis per statistical measure. Horizontal dashed lines indicates the standalone FEP statistical measures of the FreeSolvSAMPL4 set in the SAMPL4 challenge and solid blue lines indicate first place in the challenge (*i.e.* $y = 1$).

The offsets are transferable to a number of related SAMPL4 submissions

The transferability of the ML-derived offsets to related simulation protocols was also assessed to evaluate the general applicability of the methodology. Figure 2.10 summarises changes in metric ranks for all complete submissions that featured an FEP methodology (n=19). Overall the offsets improved/maintained/worsen the rankings of 12/5/2 submissions for Pearson r; 10/3/6 submissions for MUE and RMSE; 9/6/4 submissions for Kendall Tau. Importantly with one exception (see below) the offsets do not worsen the ranks of the top-performing submissions.

As expected, SAMPL4 submission 004 is among the entries that benefit the most from the correction terms. Several entries that used a similar forcefield (GAFF and AM1-BCC charges, gromacs simulation engine) but a different simulation engine or different free energy estimation protocols (*e.g.* 137, 168, 544, 575) also show improvements in metrics. This is reasonable as it has been shown that, when properly implemented, hydration free energies computed with the same forcefield by different simulation engines will broadly agree to within 0.2 $kcal{\cdot}\mathrm{mol}^{-1}$.[159]

The charge model used significantly influences the transferability of the offsets. Submission 542, 543, 545 only differ from submission 544 in the charge model used (RESP/HF-631G*, RESP/MP2/aug-cc-pVDZ/PCM, vCHARGE, AM1-BCC respectively). The offsets worsen the accuracy of the RESP methods but improve slightly the vCHARGE results. Other RESP-based submissions (166, 167, 169) see marginal changes in ranks. Submissions based on OPLS forcefields (562, 563, 564) benefit somewhat from the offsets, but not a GROMOS (529) or an AMOEBA (582) submission. This may be explained by the higher correlation of the AM1-BCC/GAFF hydration free energies with the OPLS hydration free energies (Pearson r 0.95, mean absolute deviation 1.1 kcal $\cdot\mathrm{mol}^{-1}$) than the GROMOS hydration free energies (Pearson r 0.84, MUE 1.9 kcal $\cdot\mathrm{mol}^{-1}$) or AMOEBA hydration free energies (Pearson r 0.86, MUE 3.5 kcal$\cdot\mathrm{mol}^{-1}$).

A number of submissions made use of empirical correction terms that account for known deficiencies of the GAFF force field. For instance submission 005 corrects the tendency of the GAFF forcefield to underhydrate hydroxyls.

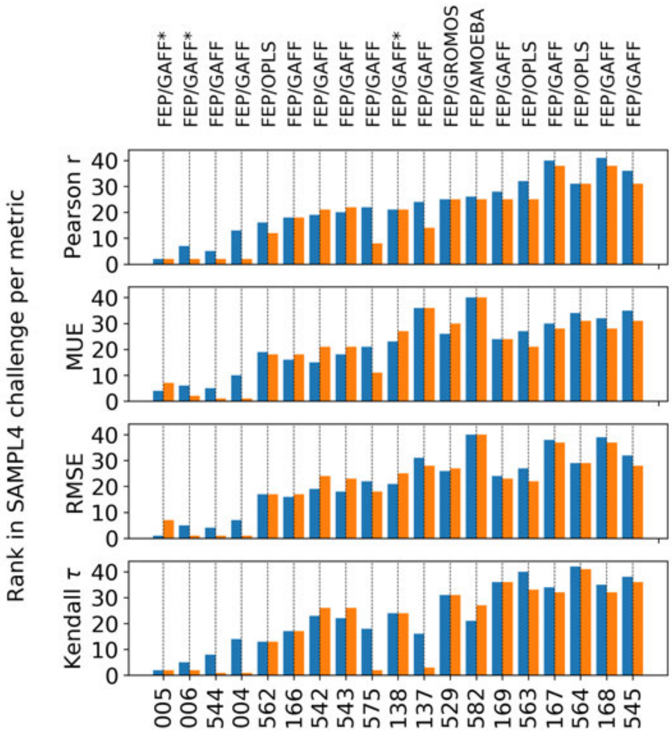


Figure 2.10: Changes in ranks of SAMPL4 submissions after application of offsets to predicted hydration free energies. Depicted are SAMPL4 FEP entries before (blue) and after (orange) hybridisation with the SVM-MolPropsAPFP correction term. The version of this plot with non-FEP entries can be found in figure 2.11. Entries were sorted by total ranks gained in ascending order. The FreeSolvSAMPL4 set corresponds to entry 004.

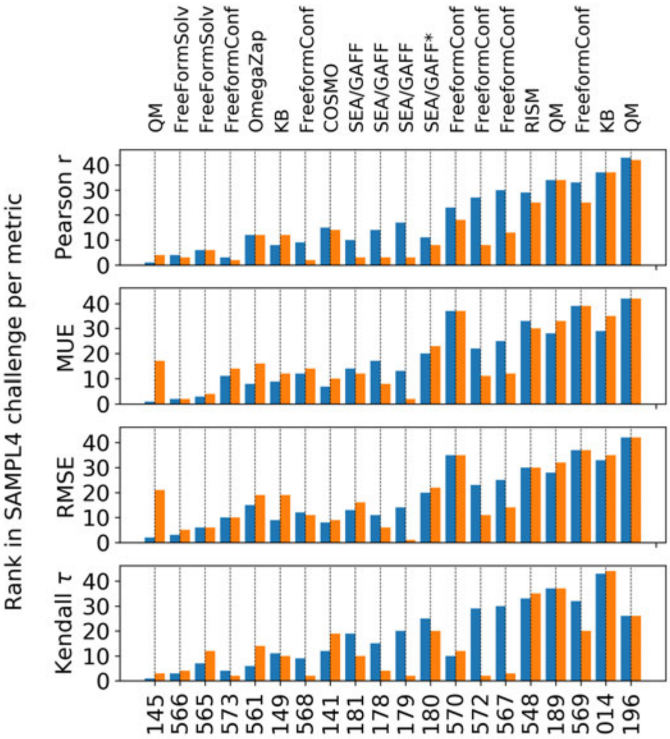


Figure 2.11: Changes in ranks of SAMPL4 submissions after application of offsets to predicted hydration free energies. Depicted are the top 20 SAMPL4 non-FEP entries before (blue) and after (orange) hybridisation with the SVM-MolPropsAPFP correction term. Entries were sorted by total ranks gained in ascending order.

Table 2.3: Effects on challenge rankings for a selection of entries in the SAMPL4 challenge when applying the ML-predicted correction term. For these entries the SVM MolPropsAPFP correction term was applied. For a given statistical measure, the standalone rank as well as the FEP/ML rank are shown separated by a tilde. Entries were sorted in ascending order by total number of ranks gained.

Entry	Method	Pearson r	MUE	Kendall tau
567	ZAP	35∼15	30∼15	25∼9
572	ZAP	31∼13	26∼15	26∼9
581	OPLS	17∼2	17∼1	30∼20
179	GAFF	15∼2	10∼1	16∼1
004	GAFF	16∼2	11∼1	13∼1
575	GAFF	25∼12	24∼15	21∼10
178	GAFF	14∼3	15∼7	11∼5
563	OPLS	37∼28	32∼24	40∼34
569	ZAP	40∼28	46∼45	35∼25
137	GAFF	27∼17	42∼41	20∼10
562	OPLS	19∼13	21∼15	17∼14
544	GAFF	6∼2	6∼1	6∼1
545	GAFF	42∼37	41∼35	43∼41
006	GAFF	7∼2	5∼2	5∼2
169	GAFF	34∼28	31∼27	39∼39
564	OPLS	38∼36	39∼33	45∼44
168	GAFF	47∼45	37∼33	37∼34
531	CHARMM	36∼31	36∼36	48∼48
015	KB	44∼44	40∼36	49∼49
548	OPLS	32∼29	34∼33	34∼34
167	GAFF	46∼44	35∼33	36∼36
138	GAFF	24∼24	28∼31	32∼26
570	ZAP	26∼21	43∼43	14∼16
530	CHARMM	28∼26	25∼27	46∼45
568	ZAP	11∼3	14∼20	4∼6
196	QM	49∼48	48∼48	22∼24
181	GAFF	3∼3	8∼13	10∼7
582	AMOEBA	33∼28	45∼45	24∼32
014	KB	43∼43	38∼41	50∼50
152	QM	45∼44	47∼47	44∼48
153	QM	39∼41	44∼45	38∼39
529	GROMOS	30∼30	29∼32	31∼32
166	GAFF	20∼20	19∼21	19∼21
197	QM	50∼50	50∼50	27∼32
158	QM	48∼48	49∼49	23∼28
180	GAFF	9∼8	22∼27	18∼20
534	QM	29∼30	27∼31	47∼48
005	GAFF	2∼2	3∼10	2∼2
189	QM	41∼41	33∼40	42∼43
149	KB	12∼16	12∼17	12∼12
573	ZAP	4∼3	13∼20	3∼6
566	ZAP	5∼7	1∼4	7∼12
141	MISC	18∼17	9∼12	15∼23
543	GAFF	22∼26	20∼24	29∼32
565	ZAP	8∼11	4∼6	9∼18
542	GAFF	21∼25	16∼24	28∼30
532	QM	23∼27	23∼31	41∼47
533	QM	10∼20	18∼24	33∼41
561	ZAP	13∼17	7∼20	8∼22
145	QM	1∼11	2∼20	1∼9

2.4 Early investigations into protein-ligand FEP/ML hybridisation

This section outlines preliminary work on applying the research theme presented in sections 2.1-2.3 to protein-ligand systems in relative Free Energy Perturbation (FEP) calculations. A similar approach was taken, however in the current form instead of modelling ΔG_{offset}, ML models were trained to fit $\Delta\Delta G_{offset}$ because FEP calculations involve a *relative* binding free energy rather than an absolute one.

2.4.1 Methods

FEP dataset generation and simulation protocol

FEP datasets were created both using both retrospective data and prospective data (see table 2.4). For retrospective FEP datasets, only the ligand poses, protein files, $\Delta\Delta G_{bind}$ predictions and experimental $\Delta\Delta G_{bind}$ values were collected. For ACK1, only the structures in protocol D of the original publication were used.[207] For FXR,[166] only the D3R stage 1 dataset's first binding modes were used because no observable difference in prediction error could be found between the two binding modes. Ligands **10**, **12**, **74**, **76-79**, **81-83**, **85**, **88** and **89** were excluded because they constituted a change in net charge compared to the other ligands in the series. Water molecules were retained in all protein structures. Prospectively, BACE, CDK2, JNK1, MCL1, PTP1B, Thrombin (PS) and TYK2 input files were adopted from Wang et al.[208] ROS1 structure files were adopted from Pérez-Benito et al.[209] using only structures associated with protein 1WHTS. DPP4 ligand structures were extracted from a Cresset in-house benchmarking set with ligands (n=73) aligned to the ligand pose in 1X70[210] using Cresset's Forge.[211] A final subset was extracted using only ligands that included a variation on the phenyl ring (n=25).

For the prospective (PS) collections, molecular simulations were run using SOMD[212] (v2018.2.0) enabling simulations to be run on a cluster of GPUs, here consisting of 16 NVIDIA GeForce GTX 980 Ti GPU cards. Prior to simulations, all systems were equilibrated using an NVT ensemble for 200ps at 298K and subsequently an NPT ensemble for another 200ps at 1atm. All simulations were run for 4 ns with 9 λ windows; in cases of hysteresis or insufficient ($< 3\%$) adjacent λ window phase space overlap they were run again with 17 λ windows (initially) or 26 λ windows (finally).

Training models to predict $\Delta\Delta\mathrm{G}offset$ values

Handling of molecular data was done using RDKit v2017.09.143 in python v3.5.6 unless mentioned otherwise. For each featurised perturbation A→B, the final features were computed by subtracting the feature values of A from the feature values of B such that each feature for the perturbation describes the change for the given feature between the two members of the perturbation. For example, given a binary feature describing the presence of a fluorine and a perturbation containing a defluorination, the feature would be set to -1. For all training and test sets, three feature sets were generated to describe atom changes, molecular property changes and changes in ligand-protein contacts, respectively:

- ΔPerturbationFingerprints were computed similarly to reaction fingerprints using Atom-pair fingerprints set to hash down to 256 bits generated with RDKit. Fingerprint subtraction between members of a given perturbation thus described the change in atom pairs for the perturbation.
- ΔMolecularProperties were computed using the Mordred[213] v1.1.145 python library using a selection of descriptors (n=71). The features contained in this set mostly describe presence of atom types, bond types, ring types, classical molecular properties (*e.g.* weight; lipophilicity) and surface area descriptors. The source-code's VdwVolumeABC.py script lacked Van der Waals radii data for iodine, so the correct value (189pm) had to be inserted. Additionally, electrostatic complementarity scores, Pearson's R and Spearman's Rho rank correlation coefficients (correlating the ligand's and protein's electrostatic surfaces) were computed using the python implementation of Cresset's Flare.[214]

Molecular property subtraction between members of a given perturbation thus described the change in a given property for the perturbation.

- ΔCloseContacts were computed using the Protein–Ligand Extended Connectivity (PLEC) function of the Open Drug Discovery Toolkit[215] v0.648 with ligand-depth = 1Å, protein-depth= 5Å, distance cut-off= 3.5Å, include waters and fingerprint length set to hash down to 16384 bits. PLEC fingerprint subtraction between members of a given perturbation thus described the change in close contacts for the perturbation.

After generation of each feature set, they were combined into separate training/ test sets comprised of all possible combinations of the feature sets (*i.e.* individual, paired or triple; n=7). As a negative control, an eighth 'noise' feature set of 256 bits was constructed where each bit was set to a random integer between 0-100.
For a given FEP calculation, the offset (*i.e.* error versus its experimental measure) can be described as:

$$\Delta\Delta G_{offset} = \Delta\Delta G_{FEP} - \Delta\Delta G_{Experimental}. \tag{2.3}$$

For each perturbation, the $\Delta\Delta G_{offset}$ value was used as the label to train/ predict on, except for the null model where $\Delta\Delta G_{Experimental}$ values were used as labels. Each of the eight feature sets was handled individually. Each set was normalised to a standard score and a SciKit-Learn principal component analysis (PCA) was used to reduce features up to 95% of variance explained. PCA loadings for each feature were computed by inversely transforming the PCA set which produces a vector of covariances per feature across PCA dimensions. The cumulative covariance per feature was considered the PCA loading for that feature. Because of the sparse nature of the training data (perturbation sets per protein target), a cross-validation-type learning procedure was adopted where the training set was split into folds and an individual model was generated for each fold. Suppose a training set is built from 3 perturbation sets for protein targets A, B and C. The training algorithm would split the collection into three folds:

- Fold “A”: B & C as training set, A as validation set
- Fold “B”: A & C as training set, B as validation set
- Fold “C”: A & B as training set, C as validation set

Following the above example, the number of models generated was thus equal to perturbation sets contained in the training set (n=11). After splitting, each model was generated using a Bayesian hyperparameter optimisation (BHO) scheme (see section 2.2). For each optimisation routine, the number of calls was set to 40 because more calls became too computationally expensive because of the scalability of the acquisition function. Hyperparameters set to be tuned by SKOPT differed per machine learning algorithm and will be outlined below. Per fold, the optimisation scheme was repeated over 30 replicates of which the top performing 10 were retained resulting in an ensemble of models per fold of size 10. Note that each of these 10 models will have their own configuration of hyperparameters. For support vector machines (SVMs), during BHO hyperparameters C, ϵ and γ were set to logarithmic scales of -3.0 to 2.0 with 6 steps. For Random Forests (RFs), the number of trees and max depth hyperparameters were set to range from 8 to 128 with seven steps. Training procedures were run on twenty Intel i9-7900X CPU cards using Scikit-Learn v0.20.0. TensorFlow-GPU v1.8.0 was used to generate feed-forward densely connected neural networks using Keras with a rectified linear activation function. An Adam optimiser was used in conjunction with an early stopping routine set to monitor validation loss and halt training with a patience of 20 epochs. During BHO, the following hyperparameters in ranges [start-end, n steps] were set: Adam parameters (to control descent and momentum) $\beta 1$, $\beta 2$ in [0.8 - 0.99, 11] and ϵ in [0.0001 - 0.5, 11]; batch size in [32 - 128, 7]; number of deep layers in [1 - 2] with number of nodes in [5 - 261, 10]. The number of layers and amount of nodes ranges were set twice to allow sampling of versatility in layer size per architecture. Training was performed using three NVIDIA GeForce GTX 1080 cards.

Table 2.4: Overview of the relative binding free energy (FEP) data sets used in constructing the training and test sets for the $\Delta\Delta G_{bind}$ correction model. Used information for each collection would consist of ligand poses, a protein file, predicted $\Delta\Delta G_{bind}$ values and experimental $\Delta\Delta G_{bind}$ values. Collections were mined both retrospectively from earlier publications (RS) or prospectively (PS) using SOMD/OpenMM. Methodologies for each retrospective dataset can be found in their respective publications (see DOI). For PS collections, structures provided by earlier publications were used where available. Note that although thrombin appears in both the retrospective and prospective collections these are two distinct congeneric series.

Type	Collection	FEP dataset	Number of perturbations	DOI
Training set	RS	ACK1	28	10.1101/333120
		FXR	76	10.1007/s10822-017-0083-9
		HSP90	24	10.1016/j.bmc.2016.07.044
		Thrombin	27	10.1021/acs.jpcb.6b03296
	PS	BACE	85	10.1021/ja512751q
		CDK2	42	
		JNK1	64	
		MCL1	114	
		PTP1B	37	
		Thrombin	12	
		TYK2	33	
Test set	PS	ROS1	51	10.1021/acs.jctc.8b01290
		DPP4	72	NA
		Cathepsin S	TBA	NA
		FXA	28	10.1021/jm0111346

2.4.2 Machine learning models fitting $\Delta\Delta G_{offset}$ values

During BHO, the dimension-less MAE/MAD was used as a validation metric. Here, the MAE is the mean absolute error of predicted $\Delta\Delta \mathrm{G}_{offset}$ versus experimental $\Delta\Delta \mathrm{G}_{offset}$ and MAD is the mean absolute variance in the dynamic range of the congeneric series in question. This metric is useful because it shows the statistical performance of a model normalised by the dynamic range of the congeneric series, meaning that any metric value lower than 1.0 suggests improvement compared to the noise associated with the intrinsic variance of the congeneric series.

It was observed that during the training phase of this protocol not all congeneric series (both RS and PS) were fit effectively. Especially TYK2, FXR, MCL1 and PTP1B did not show significant training validation. JNK1, ACK1 and Thrombin (primarily PS) did show model fitting with 0.8 <MAE/MAD< 1.0. Overall it was observed that ΔMolecularProperties outperformed other feature sets in terms of fitting this training domain for SVMs, DNNs and RFs.

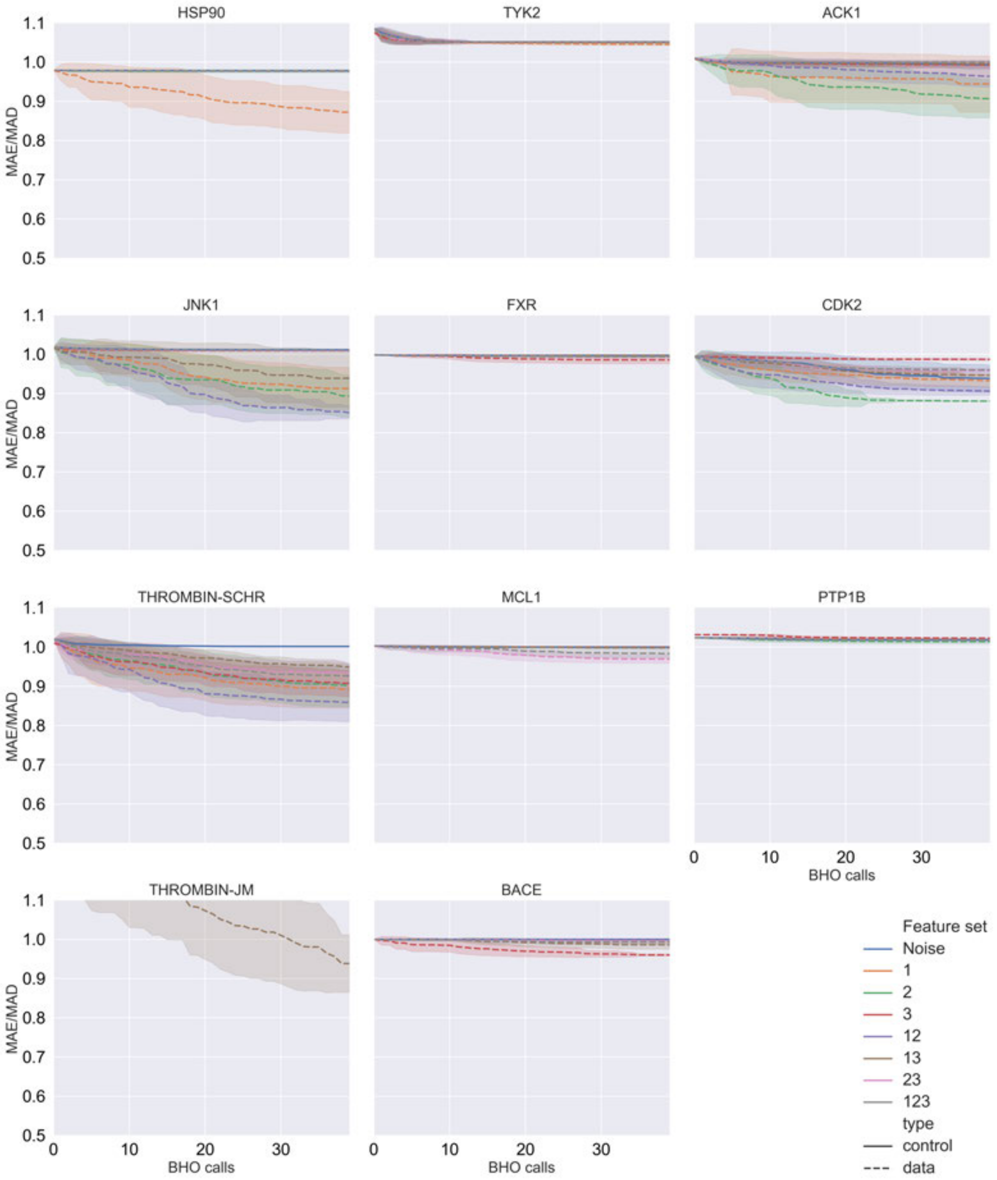


Figure 2.12: Convergence plot for the support vector machine Bayesian hyperparameter optimisation protocol showing the number of calls against the glocal minimum validation absolute error divided by the variance of the dynamic range of each dataset. For each subject fold, all different feature sets are trained on; each feature set is an ensemble of 10 models, *i.e.* per subplot each line depicts the mean validation error and the transparent region depicts the standard deviation across the ensemble. Dotted lines depict the true feature sets whereas the continuous blue line depicts the 'noise' dataset, *i.e.* the negative control.

2.4.3 Application of FEP/ML hybrid to ROS1K

$\Delta\Delta G_{offset}$ predictions for the ROS1 perturbations correlated with Pearson r values of 0.46 and 0.43 for the DNN and SVM ensembles, respectively, whereas the RF ensemble predicted significantly less accurately with a Pearson r value of 0.21 (data shown for SVM in 2.13A). When correcting the $\Delta\Delta G_{bind}$ predictions with their respective predicted $\Delta\Delta G_{offset}$ the increase in accuracy was consistent albeit modest (figure 2.13B). Across hybrids, the FEP/SVM hybrid showed the largest benefits compared to naive FEP. Hybrid models showed, on average, a ~0.1 kcal·mol^{-1} decrease in mean unsigned error (MUE) relative to experimental measures. Per-ligand ΔG_{bind} estimations are not depicted because the modest corrections did not have an effect on binding estimation accuracies likely because the weighted least squares regression algorithm used in this estimation is not sensitive enough to handle the small corrections suggested by the ML models.

One encouraging result is that the majority of corrections (~75%) were corrections in the correct 'direction', *i.e.* positive or negative corrections when they *should be* positive or negative, respectively (figure 2.13A). This suggests that with a more rigourous training protocol the correction model could be viable. However, it appears that much larger datasets are required to effectively train ML models on this type of data. The approaches taken in this work introduce large amounts of noise by collecting retrospective data and prospective data where there is a large variety in FEP methodologies (*i.e.* force fields, amount of sampling, versioning, etc..). Care must be taken to stratify future datasets in a more robust way.

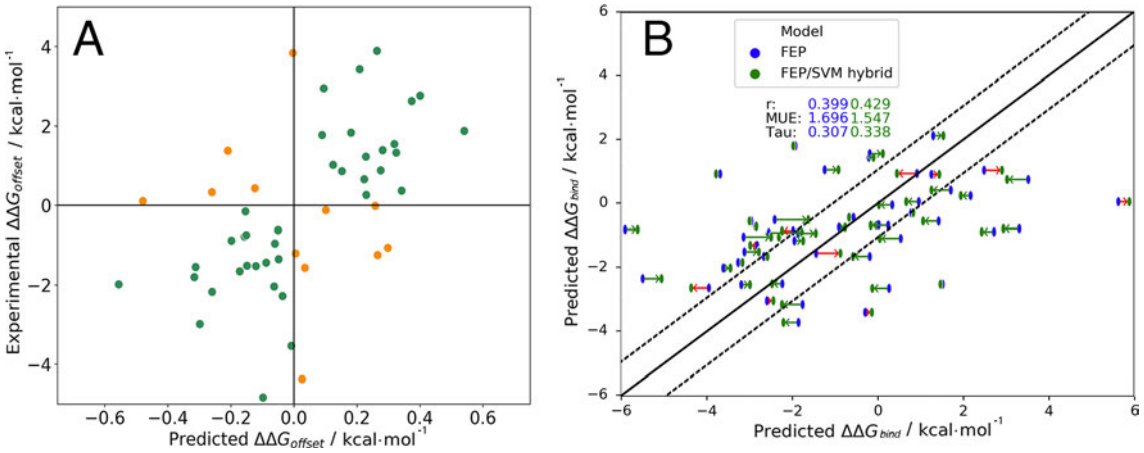


Figure 2.13: The FEP/ML hybridisation scheme applied to the protein-ligand system C-ros oncogene 1 receptor tyrosine kinase (ROS1). The correction term was learned by a support vector machine (SVM) on a variety of protein-ligand systems (table 2.4) using molecular properties as descriptors. **A**: predicted ΔΔG$_{offset}$ values versus 'experimental' (*i.e.* FEP prediction mistake). Green points are perturbations that were corrected in the correct direction (*i.e.* the correction is beneficial) and orange points are perturbations that were corrected in the wrong direction (*i.e.* the correction makes the FEP/ML hybrid prediction worse than standalone FEP). **B**: ML correction terms (arrows) plugged in to FEP predictions (blue) to produce FEP/ML hybrid predictions (green). Shown are Pearson r, MUE (in kcal/mol) and Kendall τ for the standalone FEP (blue) and FEP/ML hybrid (green) predictions versus experiment. Dashed diagonal lines indicate the 1 kcal/mol confidence bounds. For this series only a single replicate was run so no standard errors could be estimated - MBAR subsampling errors are not shown for clarity but averaged at $\sim$0.1 kcal/mol.

2.5 Conclusions

This work has demonstrated that it is possible to combine 'physics-driven' FEP methods with 'data-driven' machine learning methods to predict absolute hydration free energies of small molecules. The chief advantage over FEP is that improvements in the accuracy of the predictions are achieved without having to embark in cumbersome forcefield parameterization efforts. When compared with ML, the FEP/ML approach outperforms FEP with a much smaller training set size. This is significant as it indicates that for a new dataset it is possible to make predictions without any available experimental data initially, and switch to an FEP/ML approach once a sufficient number of data points have been experimentally determined. This advantage stems from the fact that in the FEP/ML approach the ML models only need to learn to correct errors in the FEP calculations, whereas in a pure ML approach the models must learn the physics of hydration. Another advantage of FEP/ML is that the hydration free energies of individual compounds are predicted with precision similar to that of the FEP calculations, whereas ML-based predictions by ensembles of identical models show more significant variability. In a retrospective analysis of all SAMPL4 submissions, the accuracy gains obtained in FEP/ML are sufficient to propel a mid-ranked FEP protocol among the top-ranked submissions. Further, the accuracy improvements are not limited to a single simulation protocol, and a number of related FEP approaches benefit from the correction terms. This likely stems from the fact that the hydration free energies predicted by a number of forcefields and software show correlations in their outliers.[159,216] However the performance of the correction terms is expected to decrease the more the simulation protocol diverges from that used to generate the training set.

There would be of course no need for such correction terms if more accurate forcefields were available. Thus beyond empirically correcting forcefield errors, the ML correction terms are useful to flag at essentially no computing cost molecules for which predictions are likely to deviate significantly from experimental data. This should be useful to help focus time-consuming forcefield parameterization efforts, or as part of automated workflows to decide whether to embark in bespoke forcefield parameterization for a given compound. The methodology presented here could be applied to other scenarios where FEP is used extensively, for instance relative or absolute protein-ligand binding free energy calculations. This will likely require further methodological developments to handle non negligible statistical sampling errors in the FEP results; as well as learning of a diverse set of physical interactions present in the more heterogeneous environment found in protein binding sites. Generating a training domain suitable for such a machine learning problem will be challenging as it is likely that large amounts of data are required for fitting a chemical space this large and information-rich. Nevertheless the current growth in size and diversity of protein-ligand datasets with associated FEP data should render FEP/ML an increasingly appealing option to improve the effectiveness of FEP methods in drug discovery.[111,125,162]

Chapter 3

Data-driven Generation of Perturbation Networks for Relative Binding Free Energy Calculations

3.1 Introduction

Alchemical Free Energy (AFE) calculations have seen significant increase in popularity in both the academic and commercial domains of pharmaceutical development. These types of calculations leverage an alchemical description of a molecular perturbation for the purpose of estimating free energies of binding of ligands to a drug target.[151,152,217,218] Absolute Binding Free Energy (ABFE) calculations are not yet routinely used for protein-ligand systems owing to challenges in converging accurate free energy estimates.[219–222] As a result relative binding free energy (RBFE) calculations remain one of the most popular types of AFE techniques, and have become pivotal in modern computational chemistry approaches that support medicinal chemistry campaigns. Its success is largely owed to recent improvements in processing hardware coupled with advances in empirical force fields which has pushed the technique's potential to predict ligand binding affinities with a mean unsigned error below 1 kcal·mol^{-1}, at acceptable computational costs.[208,223–225] The field of RBFE calculations has seen considerable progress over the last several years with both academic and commercial developers pushing its boundaries even further using a variety of community-curated benchmarking series and guidelines.[95,208,226–228]

The community's performance across the available RBFE benchmarking sets is variable due to the heterogeneity of RBFE implementations. This variability is primarily explained by limitations in RBFE software. This results in bottlenecks that can be shared across RBFE software, such as inaccuracies when performing scaffold hopping, net charge adjustments or changes in ligand binding modes,[95,229,230] as well as bottlenecks that are unique to certain implementations due to for instance shortcomings in supported empirical force fields.[85,175,224]

In RBFE the free energy of binding for a series of compounds is estimated from a set of pairwise binding free energy differences (ΔΔG), which are transformed into binding free energies relative to a common reference value (ΔG) via for instance a regression scheme. This requires the planning of a perturbation network (or *graph*) that connects all N compounds in a congeneric series using n edges. To connect all ligands to the network, at least $n = N - 1$ edges is required (a minimally connected network), and up to $n = \frac{N^2-N}{2}$ edges may be used (a fully connected network). Previous work has shown that accuracy of binding free energy estimation generally increases when the number of edges increases, but the computing expense of a fully connected network becomes rapidly impractical as the size of the congeneric series increases.[231]

If no error was made in the prediction of pairwise binding free energy differences (ΔΔG), each possible network for a congeneric series would yield the same binding free energy estimates (ΔG). In practice the choice of a network has a significant influence on predictive power, because a given RBFE protocol makes errors of a different magnitude for each edge. These errors arise from different sources that reflect fundamental limitations in the technology, for instance forcefield inaccuracies leading to systematic errors, and statistical errors that are introduced due to finite sampling of configurational integrals. Additionally, the performance of free energy difference estimation between pairs of compounds is influenced by numerous implementation specific details (*e.g.* softcore parameters, topological coupling methodology, λ schedule). Consequently the choice of a network that maximises accuracy and minimises computing expense for a given RBFE protocol is not trivial (figure 3.1). Such tasks have historically been carried out manually by practitioners relying on expertise in a specific RBFE implementation and intuition to select an efficient network. However, with increased adoption of RBFE and a push for routine applications to large datasets such an approach is increasingly impractical. Currently it is common practice to generate *star-shaped* networks (where all ligands are perturbed to a single reference ligand) for large ligand series ($n > 50$). Although this style of network generation is attractive because of its simplicity, little research has been done to investigate the impact it has on RBFE accuracy.

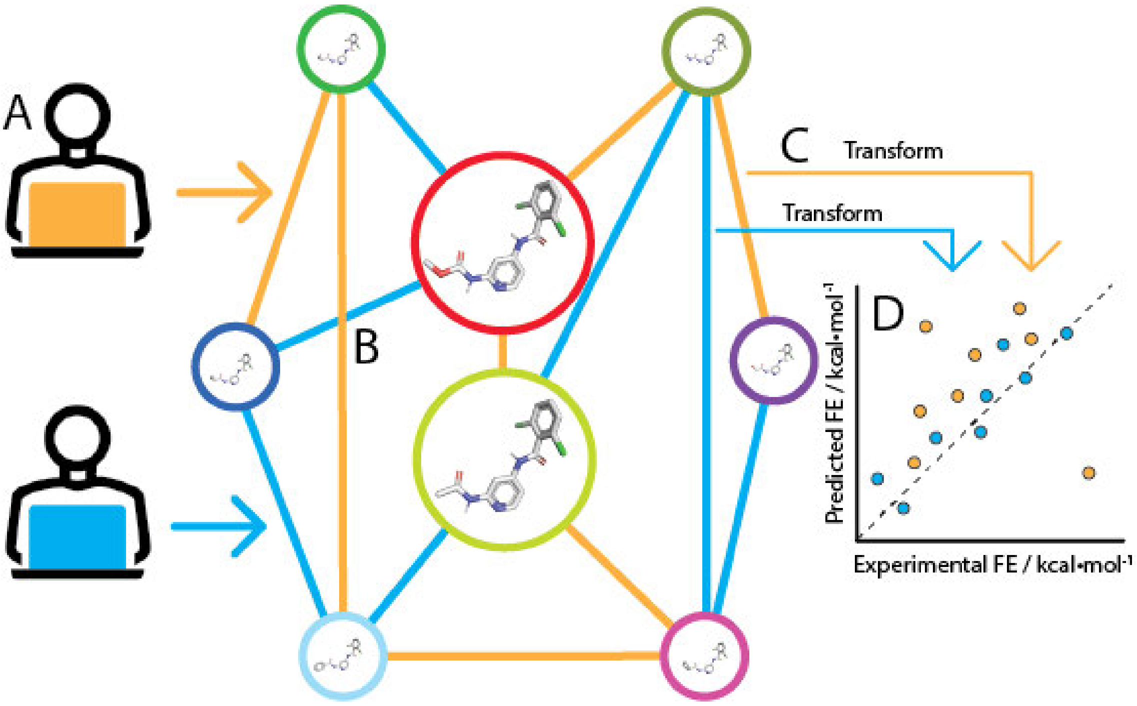


Figure 3.1: The choice of edges for a perturbation network is essential for RBFE prediction accuracy. **A**: given two network generators (orange and blue, these can be humans or machines), **B** a number of perturbations is chosen between the eight ligands in the series such that each ligand is included. For each chosen edge, and RBFE simulation is performed. Some chosen edges will produce errors of higher/lower magnitude which greatly alters the overall predictive power of RBFE. **C**: relative binding free energies are transformed using for example a regression scheme to obtain per-ligand $\Delta\Delta G_{bind}$ estimations in reference to one of the series' ligands. **D**: compared to experimental binding free energies, different perturbation network topologies have different predictive power. In this example, the blue network outperforms the orange because the latter has multiple outliers.

Lead Optimization Mapper (LOMAP[232]) is the primary programmatic approach to RBFE network generation and is used in diverse RBFE software implementations including Flare.[212,233] The LOMAP approach is based on the LOMAP-Score which is a model metric for the reliability or statistical fluctuation (SF) of a given RBFE perturbation. SF is a measure of precision, *i.e.* whether it is possible to get a converged estimate with reasonable computing effort. The LOMAP algorithm in its current form relies on expert knowledge in the form of *rules* that influence the LOMAP-Score. For example, within the LOMAP-Score algorithm a perturbation between a pair of molecules involving removal of a sulfonamide moiety would be penalised heavily as the SF of this perturbation in the context of other molecules has been found to be high during testing of the RBFE software in question. Conversely, a fluorination would result in relatively high LOMAP-Score as SFs for this class of perturbation have been found to be low during testing. Because the collection of perturbations that would ever be performed in RBFE is sufficiently large to prohibit rule generation for all of them, LOMAP-Score models SF imperfectly, resulting in sub-optimal RBFE network design. Additionally, the set of rules in LOMAP-Score has been fine-tuned for years by RBFE experts in order to make it perform acceptably for specific implementations; this has decreased transferability of LOMAP-Score between diverse RBFE implementations.

In practice and in an effort to deal with these shortcomings retrospective RBFE benchmarking studies often feature networks that have been adjusted manually using a LOMAP generated network as a starting point. In almost all cases there is an opacity as to how these networks are augmented, and it is likely that additional edges are frequently added iteratively upon examination of the initial RBFE campaign's accuracy versus experimental measures. Although it can be argued that the augmented RBFE network is a better representation of the specific RBFE implementation's predictiveness, this practice decreases comparability between implementation as augmentation is highly dependent on expertise. Additionally, as not all RBFE practitioners hold expert knowledge for network augmentation, this practice delivers an overstated picture of the true performance of the RBFE implementation in question when applied prospectively. This highlights the need for an objective approach in RBFE network generation that is not based on expert knowledge.

More recently, data-driven approaches based on optimal design that offer a theoretically more objective approach have been proposed .[234,235] Although promising alternatives to LOMAP, these algorithms are still in active development. Notably, NetBFE uses an iterative exploration of congeneric series using knowledge of SF gained incrementally by processing specific edges in the RBFE network;[235] an initial estimation of SF is thus pivotal in this approach. However, a robust SF predictor is currently absent in the field of RBFE, forcing some approaches to revert back to simpler metrics such as molecular similarity.[231] Additionally, novel machine learning (ML) techniques of describing RBFE perturbations have been proposed in the form of siamese neural networks.[190,236]

The current work proposes a data-driven RBFE network generator as an alternative to expert-driven approaches. To accomplish this, a transfer learning ML framework was designed that allows predictions of SF for molecular perturbations typically handled in RBFE. Using all predicted SFs for a given congeneric series, a data-driven RBFE network can be generated. The approach was implemented in LOMAP to generate networks using predicted SFs as input metric instead of the default LOMAP-Score.

This work presents several concepts novel to the field of RBFE network generation. RBFE-Space, a transferable training domain that is composed of a large number of RBFE perturbations (n ∼4000) was created for this work and has been made publicly available to further drive ML research in the field of RBFE. The SF predictor leverages a novel siamese neural network architecture using graph neural network (GNN) legs. The ML predictor is shown to predict SFs more accurately than the expert-driven LOMAP-Score. Finally, a fully-connected network of the TYK2 RBFE benchmarking series was simulated; network analysis on this dataset has revealed several key learning points for RBFE network generation. The prototype data-driven RBFE network generator already performs comparatively to state-of-the-art network generators, is transferable between RBFE implementations and can be objectively improved by training set expansion.

3.2 Methods

We start by defining the error made on predicting a pairwise binding free energy difference between a pair of compounds A-B with a given RBFE protocol as:

$$\Delta\Delta G_{offset,A\rightarrow B} = \Delta\Delta G_{RBFE,A\rightarrow B} - \Delta\Delta G_{EXP,A\rightarrow B}, \tag{3.1}$$

where RBFE and EXP are relative binding free energy prediction and experimental measures, respectively. This heuristic has been previously described by our group and has been used to generate ML models for *post-hoc* correction of free energy predictions.[237]

Optimal design principles suggest that networks containing many edges with low magnitude $|\Delta\Delta\text{G}_{offset}|$ values will yield ΔG estimates more accurate with respect to experimental binding affinities than networks containing many edges with high magnitude $|\Delta\Delta\text{G}_{offset}|$ values.[234,235] However $|\Delta\Delta\text{G}_{offset}|$ is not a practical metric to select an RBFE network *a priori* since it requires knowledge of the experimental measure, and the prediction from the chosen RBFE protocol.

We *hypothesize* that edges in a RBFE network with low statistical fluctuations are associated with low $|\Delta\Delta\text{G}_{offset}|$ values. This hypothesis reflects the empirical observation that, for a given protocol, RBFE predictions with large statistical uncertainties rarely give accurate estimates of experimental measures. Of course a RBFE edge prediction with a low SF could significantly deviate from the experimental measure due to systematic protocol errors (for instance due to a poor description of the energetics by the chosen forcefield), but as long as a reasonable correlation is observed, networks selected according to this metric will approximate the optimal choice. The chief motivation for this assumption is that it only requires estimation of the SF of edges for a given RBFE protocol, which can be done without knowledge of the experimental measure. Later we will show that this hypothesis is supported by data.

However, estimating SF values for every given possible edges in a network via for instance calculation of the standard error of the mean binding free energy change ($\Delta\Delta G_{bind}$ SEM) would be impractically time-consuming. Our task is therefore to find a descriptor that approximates $\Delta\Delta G_{bind}$ SEM and that can be inexpensively computed to plan an RBFE campaign. To do so we turn to machine learning (ML) and subsections 3.2.1-3.2.3 outline the associated methodological steps (training set generation, model training, and model applications).

3.2.1 Generation of a training set that encompasses RBFE-Space.

ML predictors of the SF of an RBFE calculation can in principle be derived using a sufficiently large training set that includes all possible examples of alchemical perturbations between congeneric series. However, computing SFs for a training set representative of drug-like chemical space is computationally intractable owing to the size of the training set required. To address this issue we propose the following *abstractions*: 1) representative RBFE perturbations between compounds in congeneric series reported in the literature are mapped onto a benzene ring (section 3.2.1; figure 3.2); 2) the SF of the perturbation is estimated by computing free energy changes in an aqueous phase environment (section 3.2.1).

Grafting of benchmarking series perturbations onto a common benzene scaffold

To build a collection of representative RBFE perturbations, data was drawn from all publicly available benchmarking series (n=18) as defined in recent work from the Open Force Field Initiative and Merck.[226,227] Within each series, all possible pairs of ligands were picked. Next, perturbations that involved ten or more heavy atoms perturbed or a change in formal charge were discarded (as these were deemed likely to be highly unreliable with the chosen RBFE protocol).

Using primarily the python library RDKit[238](2020.09.5), R-groups were extracted through manipulation of SMARTS-patterns generated from per-pair maximum common substructure (MCS) analyses. The 'anchor' atom for each R-group (*i.e.* the first atom in the MCS that a given R-group is attached to) was stored. Then, for each member ligand of all 3964 perturbations in the dataset, the R-groups were grafted onto benzene molecules while using the anchor atom as a linker, except for cases where the anchor atom was an aromatic carbon atom in which case no anchor atom linker was used. The main ideas for the code of this protocol were inspired by blogposts by Landrum and Schmidtke.[239,240] Whereas grafting a simple (*e.g.* chlorine addition) perturbation is straightforward, more complex perturbations involving for example multiple fused rings or more than six R groups were excluded for simplicity as grafting these becomes exceedingly complex and does not add significant knowledge to the training domain. Additionally, perturbations that involved a benzene ring without other constituents were excluded as these would cause issues when generating an MCS for the RBFE protocol, since this code largely depended on enforcing the benzene scaffold based on its topology. After removing duplicates and the grafting step the complete RBFE-Space dataset consisted of 3964 perturbations saved as dual SMILES entries.

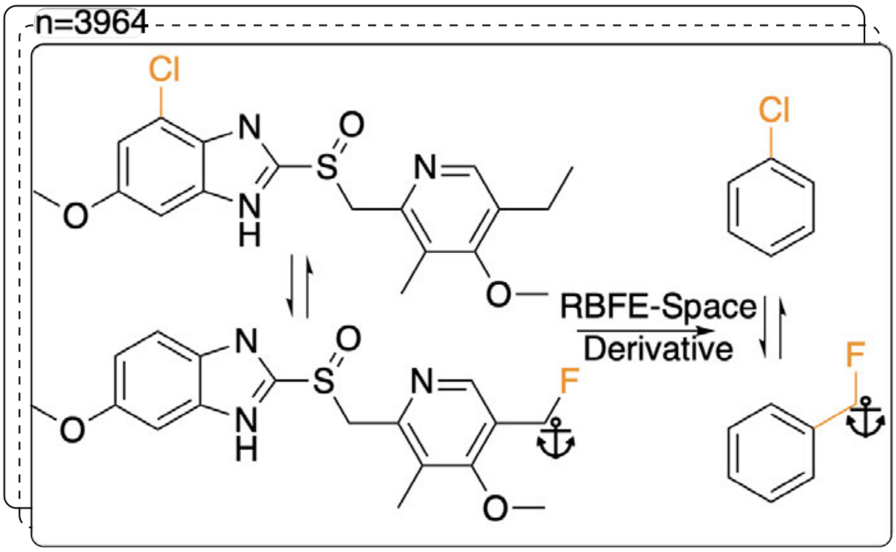


Figure 3.2: *Example grafting of a molecular perturbation onto a benzene scaffold as applied during creation of RBFE-Space in this work.* Shown is an example of a molecular perturbation typical in RBFE between two analogues of omeprazole (left-hand side), where the maximum common substructure (MCS) is shown in black. Grafting R-groups 1 & 2 onto a common benzene scaffold results in a generalised representation of the perturbation (right-hand side). In the RBFE-Space derivative, the chlorine R-group on the first ligand (chlorobenzene) is forced to vanish from the first carbon of the MCS towards the second ligand (benzyl fluoride): in practice this entails changing the chlorine atom into a hydrogen atom. In the same perturbation, the fluoromethyl group is grown on the second carbon atom of the benzene MCS. The anchor symbol denotes the aliphatic carbon atom that is used as a bridge for the methyl/fluorine in R-group 2. See 3.2.1 for a detailed description of the methodology.

Molecular dynamics simulations and free energy calculations

For each pair, a RBFE protocol was set up using BioSimSpace[241] (v2020.1.0 py37h9bf148f_593). For each benzene derivative pair in RBFE-Space, SMILES for ligand 1 and ligand 2 were parsed and an MCS was found while allowing ring breaking and ring size changing. After aligning ligands 1 and 2, a single, perturbable merged ligand was created from the two input molecules that contained the properties of both input ligands; the atom mapping used was stored to describe which R-groups were being perturbed into which ligands. This 'merged ligand' was then solvated in a 3 nm^3 cubic box with TIP3P waters. Simulations were set up with the engine SOMD[79,212,242] using 10000 moves, 50 cycles and a 2 fs timestep, adding up to 1 ns simulation time per λ window. Each perturbation was set to consist in 11 equidistant λ windows (*i.e.* $\lambda \in$ [0.0, 0.1, .. 1.0]). Each perturbation was run in quintuplicate.

Simulations for this work were run using on a variety of computing clusters (Ubuntu 16.01) mostly containing Nvidia GeForce GTX 1080 and 980 GPU cards. The walltime per window for the above described protocol was 8-12 minutes, depending on system size and hardware, totalling to ~24,000 GPUh for the complete series of runs.

For each perturbation the free energy change $\Delta G_{solvated}$ was estimated using pymbar[243] with subsampling enabled, and discarding the first 5% of the trajectories. The statistical fluctuation of a given perturbation was computed as the standard error of the mean across each quintuplicate in RBFE-Space:

$$SEM_{\overline{\Delta G}_{solvated}} = \frac{\sigma}{\sqrt{n}}, \tag{3.2}$$

where n=5 and σ is the standard deviation across the samples of $\Delta G_{solvated}$ in each quintuplicate, calculated as

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(\Delta G_{solvated_i} - \overline{\Delta G}_{solvated})}{n-1}} \tag{3.3}$$

where $\overline{\Delta G}_{solvated}$ is the mean of the five predicted relative free energies of solvation for the given perturbation. For all perturbations in RBFE-Space that were simulated in both directions (*i.e.* both A→B and B→A), SEM values were balanced by reporting the mean SEM value for both perturbations.

The TYK2 and TNKS2 series' RBFE perturbations were run on the same hardware as RBFE-Space simulations. Prior to system setup, proteins were prepared using Flare V4. Ligands (GAFF2) and proteins (FF99SB) were parameterised using BioSimSpace (which uses LEaP, Antechamber and Parmchk) and solvated in TIP3P waterboxes (10Å orthorhombic shell). Note that FF99SB is outdated - repetition of these simulations should be performed with the newer FF14SB to improve predictivity versus experimental measures. Each system (*i.e.* the ligand, protein and waters) was energy minimised (250 steps) and pre-equilibrated at $\lambda = 0.0$ using a sequence of NVT and NPT equilibration with cuda.pmemd using the BioSimSpace API. As with RBFE-Space simulations, 11 λ windows were used for each ligand perturbation, but with 4ns of sampling instead of 1ns (initial tests showed that 1ns of sampling was insufficient for systems of this complexity). For each perturbation, relative free energies of solvation and binding in kcal·mol^{-1} were estimated using pymbar with subsampling enabled, and discarding the first 5% of each trajectory to allow for re-equilibration at each λ value.

3.2.2 Training of machine-learning models that predict RBFE statistical fluctutations.

Given the complete RBFE-Space training domain with calculated $SEM_{\overline{\Delta G}_{solvated}}$ values as per Eq. 3.2, ML models were trained to predict this value for a newly-presented perturbation. From here on, $SEM_{\overline{\Delta G}_{solvated}}$ values predicted by ML models will be referred to as $\widehat{SEM}$.

All ML code was executed using the Keras implementation of TensorFlow 2.6.0. All models (pre-training, transfer-learning and fine-tuning) were run using a LogCosh loss function and Adam optimiser with an initial learning rate of 5e-7. All ML models were run on a system running Ubuntu 18.04.4 LTS with 20 CPU cores (Intel(R) Core(TM) i9-7900X CPU @ 3.30GHz) and four Nvidia GeForce GTX 1080 GPU cards using CUDA 11.2.

Main RBFENN model architecture based on siamese neural networks

To model perturbations between two molecules a novel approach based on siamese neural networks[244,245] was adopted (figure 3.3). This approach has been used in other work for image recognition in low-data regimes where the goal has been to distinguish between images in the testing domain. Typically this approach consists of three concepts: **1**) a two-legged structure, where each 'leg' has one input, **2**) shared weights between legs such that the legs learn the same encoding and **3**) some form of similarity (*e.g.* Euclidian distance) layer that computes the degree to which encodings overlap.

In this work a modified siamese neural network ('RBFENN') was used that adopts concept **1** and **2**, but does not let a similarity layer compute distance. The rationale behind using shared weights is that for a given ligand perturbation, either direction (*e.g.* growing or vanishing an R-group) entails roughly the same statistical fluctuation in RBFE. Because the intended prediction label in this work is $\widehat{SEM}$, not similarity, a concatenation layer was used to join legs of the neural network. After the concatenation layer, several fully-connected layers were used with decreasing numbers of neurons leading to the final single neuron. All fully-connected layers used in the network used ReLu activation function, whereas the final single neuron used a linear activation function. See figure 3.4 for a more low-level overview of the RBFENN architecture.

To encode the chemistry of input structures (ligands A and B) a per-leg message-passing neural network (MPNN) was used. Input graphs were populated with three inputs, namely *atom features* (element, #valence electrons, #hydrogen bonds, orbital hybridisation), *bond features* (bond type, conjugation) and *atom pair indices.* Whereas the MPNN architecture was based on previous work by Gilmer[246] and DeepChem[247], the code implementation of this work was primarily based on examples provided by Kensert[248]. Based on information provided during RBFE setup, the atom-mapping (*i.e.* which R-groups are transformed to which between ligands A and B) is expressed as an array of 50 integers, where each integer index relates to the atom index in ligand A, and the integer value relates to the atom index in ligand B. It is assumed that the model learns atom indexing which is reasonable because the algorithm for graph generation in the MPNN algorithm uses atom indexing to represent bonds in each ligand encoding. Because no training ligands' mappings contained more than 50 atoms, all non-matched values in the mapping array were set to 99 to represent a non-match.

Although the number of allowed epochs was set to 5000, an early-stopping callback was set to quit training when models started overfitting by monitoring mean absolute validation error; the callback was set to restore the model with the lowest validation error.

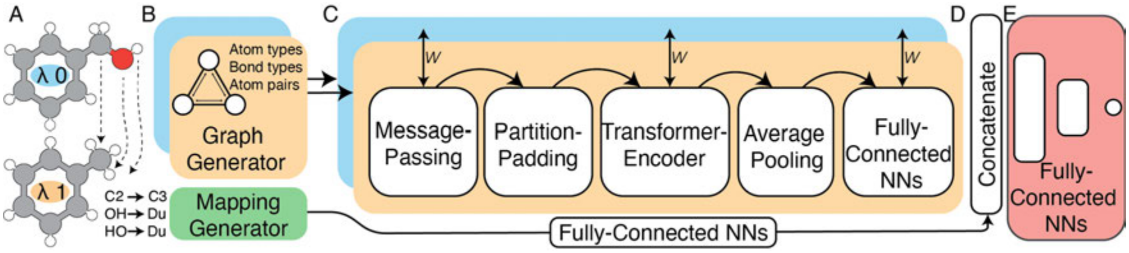


Figure 3.3: *High-level schematic representation of the siamese Relative Binding Free Energy Neural Network (RBFENN) architecture.* **A**: two ligand structures are input as SMILES, where each ligand represents either λ endstate of a given RBFE perturbation. **B**: molecular structures are described as graphs using atom types, bond types and bonds as descriptors. **C**: the bi-legged graph neural network (GNN) component of the architecture that consists of a message-passing neural network sequence ending in several feed forward NN layers. Training weights are shared between the two legs (orange and blue) of this component. **D**: A concatenation layer merges the signal of the two input legs (orange and blue) as well as the atom mapping between λ endstates which has been passed through several feed forward NNS. **E**: multiple feed forward NN layers with linearly decreasing numbers of neurons resulting in a single neuron with a linear activation function. Note that the all layers in section C are frozen during the pre-training stage of the transfer-learning phase described in 3.2.2. See figure 3.4 for lower-level details on the model architecture.

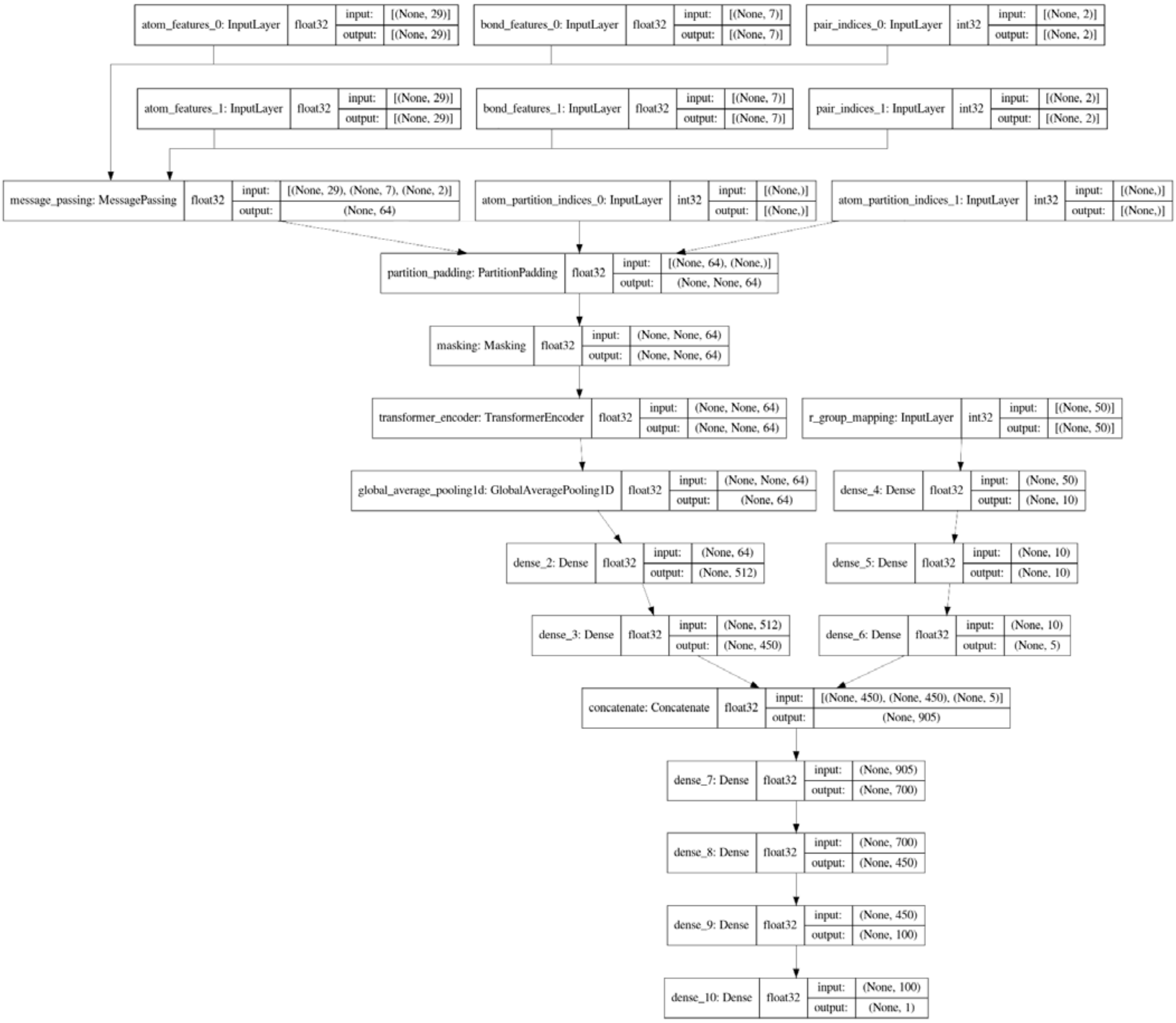

atom_features_0: InputLayer | float32 | input: [(None, 29)] | output: [(None, 29)]
bond_features_0: InputLayer | float32 | input: [(None, 7)] | output: [(None, 7)]
pair_indices_0: InputLayer | int32 | input: [(None, 2)] | output: [(None, 2)]
atom_features_1: InputLayer | float32 | input: [(None, 29)] | output: [(None, 29)]
bond_features_1: InputLayer | float32 | input: [(None, 7)] | output: [(None, 7)]
pair_indices_1: InputLayer | int32 | input: [(None, 2)] | output: [(None, 2)]
message_passing: MessagePassing | float32 | input: [(None, 29), (None, 7), (None, 2)] | output: (None, 64)
atom_partition_indices_0: InputLayer | int32 | input: [(None,)] | output: [(None,)]
atom_partition_indices_1: InputLayer | int32 | input: [(None,)] | output: [(None,)]
partition_padding: PartitionPadding | float32 | input: [(None, 64), (None,)] | output: (None, None, 64)
masking: Masking | float32 | input: (None, None, 64) | output: (None, None, 64)
transformer_encoder: TransformerEncoder | float32 | input: (None, None, 64) | output: (None, None, 64)
r_group_mapping: InputLayer | int32 | input: [(None, 50)] | output: [(None, 50)]
global_average_pooling1d: GlobalAveragePooling1D | float32 | input: (None, None, 64) | output: (None, 64)
dense_4: Dense | float32 | input: (None, 50) | output: (None, 10)
dense_2: Dense | float32 | input: (None, 64) | output: (None, 512)
dense_5: Dense | float32 | input: (None, 10) | output: (None, 10)
dense_3: Dense | float32 | input: (None, 512) | output: (None, 450)
dense_6: Dense | float32 | input: (None, 10) | output: (None, 5)
concatenate: Concatenate | float32 | input: [(None, 450), (None, 450), (None, 5)] | output: (None, 905)
dense_7: Dense | float32 | input: (None, 905) | output: (None, 700)
dense_8: Dense | float32 | input: (None, 700) | output: (None, 450)
dense_9: Dense | float32 | input: (None, 450) | output: (None, 100)
dense_10: Dense | float32 | input: (None, 100) | output: (None, 1)

Figure 3.4: Low-level depiction of the 'RBFENN' siamese neural network architecture. Top to bottom: two (0 and 1) input legs are featurised into atom, bond and pair descriptors. Both legs are passed into a MessagePassing layer, which together with atom partition indices (from both legs 0 and 1) are partitioned and masked before being passed to a TransFormerEncoder layer. After a global average pooling step, two fully-connected feed-forward NN layers join with the encoded atom-mapping into a concatenation layer. Finally, three dense fully-connected feed-forward NN layers with linearly reducing numbers of parameters lead to a single-neuron layer. All dense layers in the network use ReLu activation functions except for the last single linear neuron. Each layer block depicted in this figure shows the indexed layer name (as used within TensorFlow), the class name, the dtype handled as well as the input and output dimensions.

Transfer-learning approach

To deal with the low-data regime (n=3964, see 3.2.1) and the added model complexity of an MPNN (see 3.2.2), a transfer-learning[249] approach was adopted that uses a pre-training regime to learn molecular encodings on a larger dataset with a cheaply computed label. In this way, the RBFENN can be *pre-trained* on a domain of n=(3964/2) $\approx 3.9^6$ points (*i.e.* composed of all possible pairs of molecules in RBFE-Space); as a cheaply computed label the difference in estimated solubility[250,251] (ΔESOL) was used. This property was chosen not because it is necessarily meaningful to this work's purposes, but because it is a complex descriptor that pushes the RBFENN to learn a more complete understanding of chemical structure *a priori*; similar approaches have been reported.[252,253] Early-stopping patience for this phase was set to 5 epochs as early convergence (70-100 epochs) was observed. For the pre-training phase, 800,000 training samples and 200,000 validation samples were used to save memory and because it was observed that larger training/validation sets did not sufficiently improve model training.

Subsequent to pre-training, model weights from the pre-training phase were loaded and the last four fully-connected layers were replaced with re-initialised (*i.e.* weights set to 0) layers. All other pre-trained layers of the RBFENN (MPNN legs and concatenation layer) were 'frozen' by setting `layer.trainable = False` for each layer. In this *transfer-learning phase*, the RBFENN that has learned to encode chemical structure input learns to predict $\widehat{SEM}$ (instead of ΔESOL) by training the newly initialised fully-connected layers on the 2550 $SEM_{\overline{\Delta G}_{solvated}}$ samples in RBFE-Space. For this phase, a k-fold cross-validation approach was used where k=5.

For each k-fold model in the transfer-learning phase, **fine-tuning** was performed by unfreezing all layers (*i.e.* `layer.trainable = True`) and training all layers in an attempt to further minimise validation loss. Both the transfer-learning and fine-tuning phases used a maximum of 5000 epochs with early stopping patience set to 101 epochs. Training was repeated for 9 replicates. Model predictions discussed from here on are thus mean predictions across $5 \cdot 9 = 45$ models.

Baseline shallow machine learning model training

A selection of non-neural-network ML models were used to benchmark the RBFENN model performance against. Similar to previous work,[237] three different descriptors were used:

- *APFP*:Atom pair fingerprints as computed using RDKit with a hash length of 256 bits.
- *ECFP*: Extended connectivity fingerprints as computed using RDKit with a diameter of 6 Å and 1024 bits.
- Molecular properties as computed using Mordred[213] with all 2D descriptors enabled (n=1613) where empty fields were replaced with zeroes.

Because featurisation in this case deals with molecular perturbation and not single molecules, a fingerprint subtraction technique was used where each bit value of ligand B is subtracted from the bit value of ligand A.[254] For each descriptor type, the featurised RBFE-Space was normalised and dimensionalities were reduced using principle component analysis (PCA) using the SKLearn implementation set to keep the 100 most contributing components. Through this, the resulting dimensions were the principal vectors rather than original features exhibiting large enough variance

Two shallow ML algorithms were trained using each of the three descriptor training sets of RBFE-Space:

- *RF*: Random forest regressor using default hyperparameters.
- *SVR*: Support vector machine regressor using default hyperparameters, with the exception of γ which was set to 1e-8.

Normalisation data, fit PCA objects and fit ML models were pickled for testing phases.

3.2.3 Application of RBFE statistical fluctuation predictions to network generation problems.

As outlined in 3.2.2, an ensemble of 10 $\widehat{SEM}$-predicting RBFENNs was generated. From here on, a given $\widehat{SEM}$ prediction for a perturbation between two ligands is computed as the mean of the ensemble's $\widehat{SEM}$ predictions, but is still denoted as $\widehat{SEM}$.

Featurising test sets for network prediction

For a given congeneric series' collection of ligand files, a fully-connected network (*i.e.* all possible pairs of ligands, mono-directional) is generated. For each input perturbation, the RBFE-Space derivatives are created as described in 3.2.1. To ensure that the correct perturbation is represented in the atom-mapping array, all atoms of the input ligands that change AMBER atom-type in the perturbation are stored. Then, by forcing the MCS on the six aromatic carbons of the benzene scaffold of the RBFE-Space derivatives, and generating the AMBER atom-type changes with that mapping, the atom-type change information is compared to the input ligands' atom-type information. By rotating the benzene scaffold forced MCS on one of the ligands (*e.g.* where {0:0, 1:1, 2:2, 3:3, 4:4, 5:5} is the initial forced MCS mapping, a first rotation would be {0:1, 1:2, 2:3, 3:4, 4:5, 5:0}) a second collection of RBFE-Space derivative atom-type changes is created. By repeating this process until all five rotations are completed and picking the mapping that matches the input ligands' mapping atom-type changes, the picked featurised atom-mapping array is ensured to correctly map the per-atom changes between the two ligands.

Processing of predicted statistical fluctuations using LOMAP for network generation

A forked version of LOMAP as developed by Cresset for Flare[212] V4 was cloned and implemented into BioSimSpace. As this version of LOMAP allows the usage of user-input scores per ligand pair, $\widehat{SEM}$ values (or other values such as $SEM_{\Delta\Delta G_{bind}}$ or random values etc...) could be used instead of LOMAP-Score for generating RBFE networks.

Because LOMAP is designed to build networks using the continuous LOMAP-Score that range [0-1] (where 0 is a supposed unreliable edge and 1 is a supposed reliable edge), user-input values needed to be transformed to fit this range. For an example array of SEM values $[SEM]$ that contains all possible combinations of ligands in a congeneric series, the array was scaled to the range [0-1] such that

$$[SEM]^{scaled} = \frac{[SEM]_{inv} - min([SEM]_{inv})}{max([SEM]_{inv}) - min([SEM]_{inv})} \tag{3.4}$$

where $[SEM]_{inv}$ is computed as

$$[SEM]_{inv} = \frac{1}{[SEM]}. \tag{3.5}$$

Equations 3.4 and 3.5 applied to $SEM_{\Delta\Delta G_{bind}}$, $\widehat{SEM}$ and $|\Delta\Delta \text{G}_{offset}|$ result in $SEM^{scaled}_{\Delta\Delta G_{bind}}$, $\widehat{SEM}^{scaled}$ and —$\Delta\Delta\text{G}^{scaled}_{offset}$—, respectively. These arrays offer the ability to be ported into the LOMAP network generating algorithm as they match the range and direction of LOMAP-Score. To avoid cumbersome notation, the *scaled* upperscript symbol is excluded from here on unless otherwise specified.

Network generation and analysis

BioSimSpace[241] (v2020.1.0 py37h9bf148f_593) was used to generate RBFE networks. The main software that handle network generation internally are LOMAP (edge selection; as implemented in Flare V4), RDKit 2020.09.3 (molecular manipulation), networkx 2.6.3 (network manipulation) and matplotlib 3.4.3 with pydot 1.4.2 (network plotting). Similarities between networks (for the same congeneric series) were computed as the percentage of edge overlap between the two networks: given the number of overlapping edges, the percentage relative to both network sizes (n-edges) was computed. The mean percentage was taken as the final network overlap percentage.

Given a set of RBFE predictions, the statistical performance versus experimental ligand binding affinities can be estimated. Whereas a per-edge ('pairwise') statistical analysis is meaningful, in this work a per-ligand free energy estimation is estimated using a weighted least squares regression method implemented in FreeEnergyWorkflows[255] with weights set as the propagated standard error of the mean values across the replicates of each RBFE leg (solvated and bound) in kcal·mol^{-1}. Pearson R, Mean Unsigned Error (MUE) and Kendall τ metrics were estimated using a bootstrapping approach set to 10,000 repeats. Further plotting methodologies adhered to best practices.[95]

3.3 Results and discussion

3.3.1 Early attempts at constructing suitable training domains that encompass RBFE-Space.

For the purposes of this thesis, early attempts at the generation of a suitable training set for the work in this chapter will be highlighted. The final training domain form presented from section 3.3.2 and on was created primarily building on lessons learned from early generations of this dataset. There is an underlying mechanism when generating a novel dataset to describe a complex physical problem: although ideally we would like to describe the physical process as completely as possible, generating data for this complete physical process becomes more expensive the more information is included (as simulated systems become larger). Additionally, the more information is included per data point, the larger the space becomes to describe the information.

To illustrate the latter, if we were to include protein information in simulations of the physical systems, we would be required to run a sufficiently diverse number of proteins to cover the whole of 'protein-space'. This exponential increase of both computational expense and the size of context physical space is shown schematically in figure 3.5. Data is not shown for any early attempts but an explanation as to why they were inadequate will be provided.

Training on as much physical information as possible

The first attempt at generating a training domain aimed to include as much information as possible by simulating drug-like ligands in their corresponding targets in water boxes (figure 3.5H). Because even a few hundred points are expensive to generate, this data was mined retrospectively from earlier RBFE calculations performed over the year, resulting in ∼2000 data points. Although training ML models on this set of SEM values was possible, this dataset suffered from multiple issues. Primarily, as previously mentioned the 'protein-space' was explosively large and the dataset's scope only covered a small fraction of this. This resulted in poor transferability to other congeneric series. Additionally, because RBFE results were mined retrospectively the methodology per data point (or rather, sets of data points) differed significantly. Although all simulations used SOMD, the many different force fields, different versions of SOMD, different numbers of replicates (etc..) introduced a significant amount of noise in the dataset. Although in theory it is possible to correct for this by introducing descriptors for these methodological differences, this was considered extremely challenging to do robustly and thus new approaches were sought.

Training on as little physical information as possible

Building on prior experience[237] with the FreeSolv database,[194] a minimal training set was constructed using these 643 molecules. By generating all possible combinations of molecules in this set, a dataset of $643^2 - 643 = 412,806$ perturbations can be selected for simulations. Because running the complete dataset is prohibitively expensive to run in quintuplicate, only a selection was taken by excluding overly large perturbations from the set. Perturbations were run in both vacuum and solvated phases (figure 3.5A and D, rep.). Although a large number of data points could be generated (∼80,000 for vacuum and ∼13,000 for solvated), results suggested that FreeSolv molecules were not suitable for generating an RBFE-Space domain. The primary reason for this was that FreeSolv molecules are not drug-like (*i.e.* typically weight $<$ 200 Da; often composed of mostly polar groups) even though they are organic molecules. This results in a collection of perturbation datapoints that contains non-drug-like scaffolds (even though the perturbations themselves might be representative of typical RBFE perturbations): ML models trained on this dataset thus learn the non-drug-like scaffold which results in decreased transferability to more drug-like test sets. Removing the FreeSolv scaffolds (by not featurising them) is possible, although in this way a level of intrinsic noise is introduced to the training domain that is not accounted for in the descriptors that are fed to ML models. Additionally, in the vacuum phase it was found that SEMs correlated poorly with drug-like SEMs (in bound phase), suggesting that the solvent context plays an important role in statistical error of RBFE calculations.

3.3.2 Creation of a training domain that encompasses RBFE-Space.

RBFE-Space characteristics

Molecular simulations were performed for perturbations grafted onto a common benzene scaffold (figure 3.2 and 3.2.1) to serve as a training set that captures the space of perturbations that are performed in typical RBFE campaigns. To generate this set, available RBFE benchmarking series were selected and all perturbations within each were extracted and grafted onto benzene (figure 3.6E). Duplicate perturbations and perturbations that involved ten or more perturbed heavy atoms were discarded which resulted in a training set of 3964 points (starting from 16,048). Across this set, the number of perturbed heavy atoms was uniformly distributed (frequency of 400-500 points for n=1-9), except for single-atom substitution perturbations of which only 46 were simulated (figure 3.6B).

$SEM_{\overline{\Delta G}_{solvated}}$ values for all perturbations in RBFE-Space showed a distribution that skewed right; the vast majority of $SEM_{\overline{\Delta G}_{solvated}}$ values were under 1 kcal·mol^{-1}, with a peak frequency of ~0.15 kcal·mol^{-1} (figure 3.6A). Although no relation is observed between the change in molecular weight and the associated $SEM_{\overline{\Delta G}_{solvated}}$ for a given perturbation (figure 3.6C), an increase in median $SEM_{\overline{\Delta G}_{solvated}}$ can be observed by increasing the number of heavy atoms perturbed, although it is clear that there are exceptions to this rule as outliers are present in every scale (figure 3.6D). Only direct single-atom substitutions (*i.e.* n=0) result exclusively in perturbations with $SEM_{\overline{\Delta G}_{solvated}} < 0.5$ kcal·mol^{-1}. Although this relation with the number of perturbed heavy atoms reflects favourably on state-of-the-art MCSS rule-based methods, the noisy nature of this relation suggests that there is scope for more accurate methods to model statistical fluctuations of RBFEs.

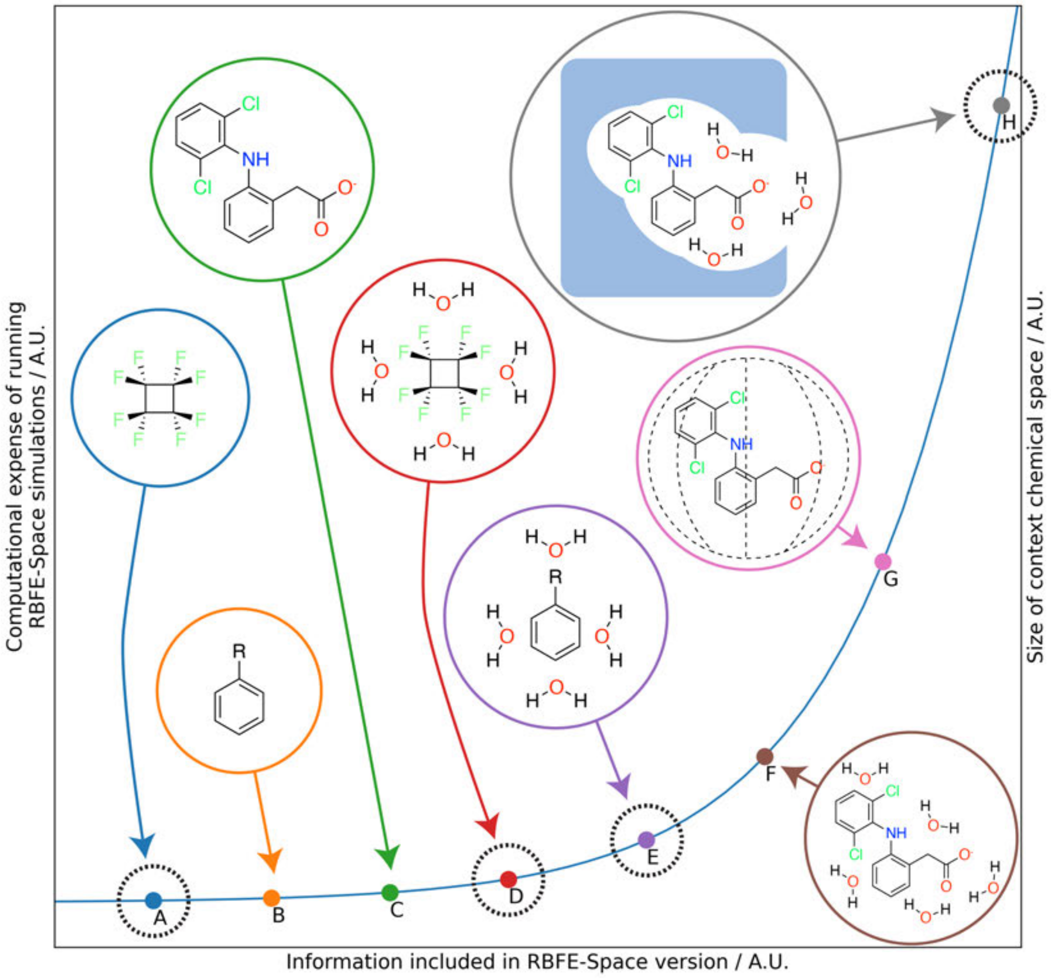

Computational expense of running RBFE-Space simulations / A.U.
Information included in RBFE-Space version / A.U.
Size of context chemical space / A.U.
A
B
C
D
E
F
G
H

Figure 3.5: Concept figure showing different possible versions of RBFE-Space (A-H). For each version, an example molecule/system is shown to represent the nature of the dataset (recall that each data point in each set is in fact a transformation between two molecules). Versions with dashed circles around them are versions that were attempted in experiments. **A**: FreeSolv compounds in vacuum. **B**: benzene derivatives in vacuum. **C**: original ligands in vacuum. **D**: FreeSolv compounds in water phase. **E**: benzene derivatives in water phase. **F** original ligands in water phase. **G**: original ligands in molecular cage or other binding pocket abstraction. **H**: original ligands in original protein systems in water solvent. By increasing the information included per version, both the computational expense for simulating a single point for the version and the size of the context chemical space (*i.e.* the number of simulations required to cover the context chemical space) increases exponentially.

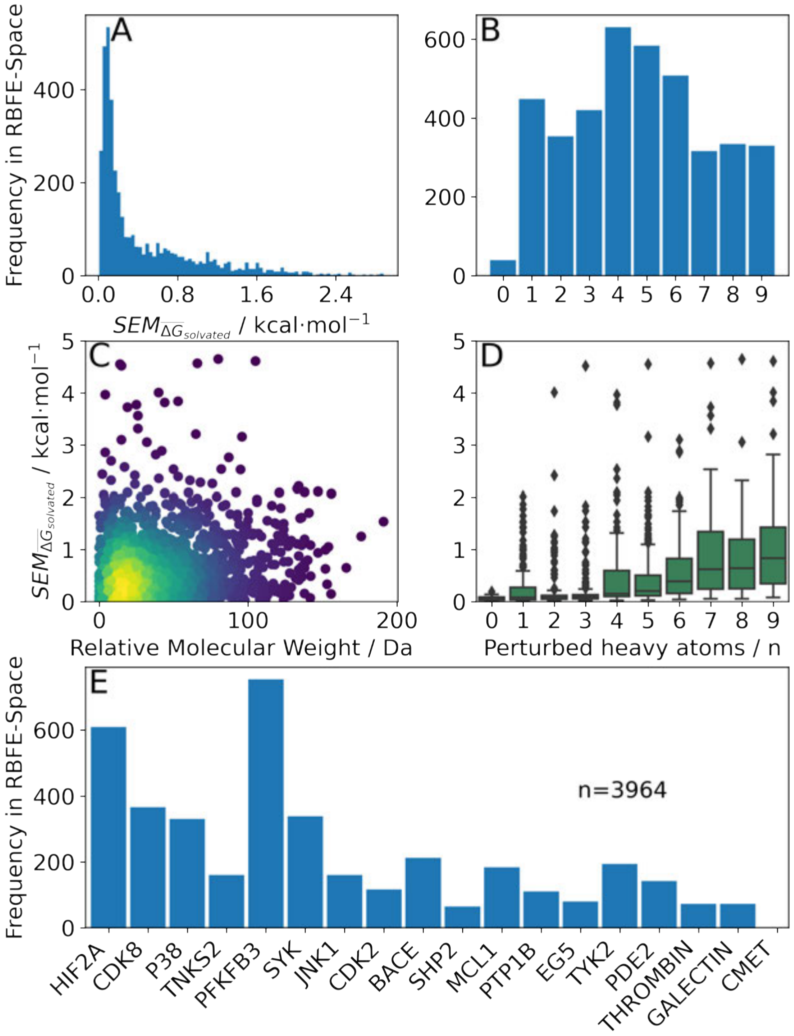


Figure 3.6: Summary of RBFE-Space generated using 3964 molecular perturbations grafted onto a common benzene scaffold (figure 3.2). **A**: histogram of $SEM_{\overline{\Delta G}_{solvated}}$ values (3.2.3). **B**: histogram of the number of perturbed heavy atoms involved in each perturbation. **C**: scatterplot showing the relation between the change in molecular weight per perturbation in Da and the $SEM_{\overline{\Delta G}_{solvated}}$ for each perturbation; colouring shows density (increasing as *blue→green→yellow*). **D**: boxplots of $SEM_{\overline{\Delta G}_{solvated}}$ per perturbation binned by the number of heavy atoms perturbed; horizontal lines in boxes show median values and black diamonds show outliers (95 CI). **E**: histogram that describes how many perturbations of the original congeneric series' were used as templates for grafting onto benzene in RBFE-Space.

RBFE-Space derivatives correlate to their drug-like counterparts

To investigate whether RBFE-Space derivatives are sufficient placeholders to model statistical fluctuations of their drug-like counterpart RBFEs, 214 'original' perturbations (*i.e.* the perturbations from nine publicly available congeneric series) were simulated in quintuplicate for 1ns per λ window (n=11). Subsequently, all combinations of phases were compared (figure 3.7).

For perturbations that give large SEM values quintuplicates runs are insufficient to obtain consistent results for a given edge processed in two directions (A→B or B→A), which introduces noise in correlations of these quantities (figure 3.8, left-hand side). To remedy this, a logarithmic scale was adopted (figure 3.8, right-hand side) when comparing SEM (or any other type of variance) arrays, which squashes larger deviations with respect to smaller deviations. This is justified because our approach does not need to estimate accurately large SEM values since edges with large SEM values will be discarded during network generation.

In the following analysis, benzene-grafted perturbations will be referred to as *RBFE-Space perturbations*, whereas the template perturbation (*i.e.* with the original ligand scaffolds) will be referred to as *drug-like perturbations*.

Drug-like solvated SEM values correlate well (R=0.86) with their bound counterparts, but tend to show lower magnitude (figure 3.7A). This is surprising as a bound system has higher complexity than a solvated box - it is expected however that the short sampling time for this analysis (1-ns/λ) was insufficient to relax the protein topology in the simulation, thus enforcing a relatively rigid environment for the ligand perturbation in which only a narrow range of conformations could be sampled. This was deemed acceptable for the present study as we are mainly interested in correlating SEM values.

RBFE-Space SEM values also correlate well to both drug-like solvated, bound SEM values (R=0.74 and 0.87, resp.) and to $\Delta\Delta$G SEM values (R=0.75); A trend in the number of heavy atoms perturbed increasing with higher SEM values can be observed which reflects the trend seen in figure 3.6D.

It should be noted that in this prototypical version of RBFE-Space there is a possibility that R-groups that are separated from each other in the context of a drug-like scaffold will interact with each other when grafted onto a benzene scaffold. We have observed a trend for greater deviation between RBFE-Space and ligand SEM values for cases where bulky R-groups (each of > 5 heavy atoms; see chapter 4 of this thesis for a more detailed discussion) are being simultaneously grown and vanished in the same perturbation. Although not investigated in depth, this issue is assumed to be present in a small population of RBFE-Space, and will have to be resolved in future versions of the dataset. Early solutions to this problem could for instance be placing the second R-group on the *para* aromatic carbon of the benzene scaffold; however any third (or more) R-groups will reintroduce the issue. Alternatively, larger scaffolds could be explored.

The main objective of this analysis is to assess whether the RBFE-Space placeholders' SEM values sufficiently correlate to $\Delta\Delta G_{bind}$ SEM values of their drug-like counterparts (figure 3.7D). Although only moderate correlation has been reached, we postulate that this is a logical effect of simplifying ligand perturbations by grafting them onto a common benzene scaffold. Such simplification is necessary to obtain a training domain that is transferable to a variety of congeneric series. Through this simplification, several sources of information are discarded: 1) removal of protein topology and ligand-protein interactions 2) removal of ligand scaffold (interacting with protein, itself or solvent) 3) reduced sampling time (1-ns/λ instead of 4-ns/λ). Whereas all of these could be included in RBFE-Space they would require a significant increase in the size of the training domain to enable development of transferable models.

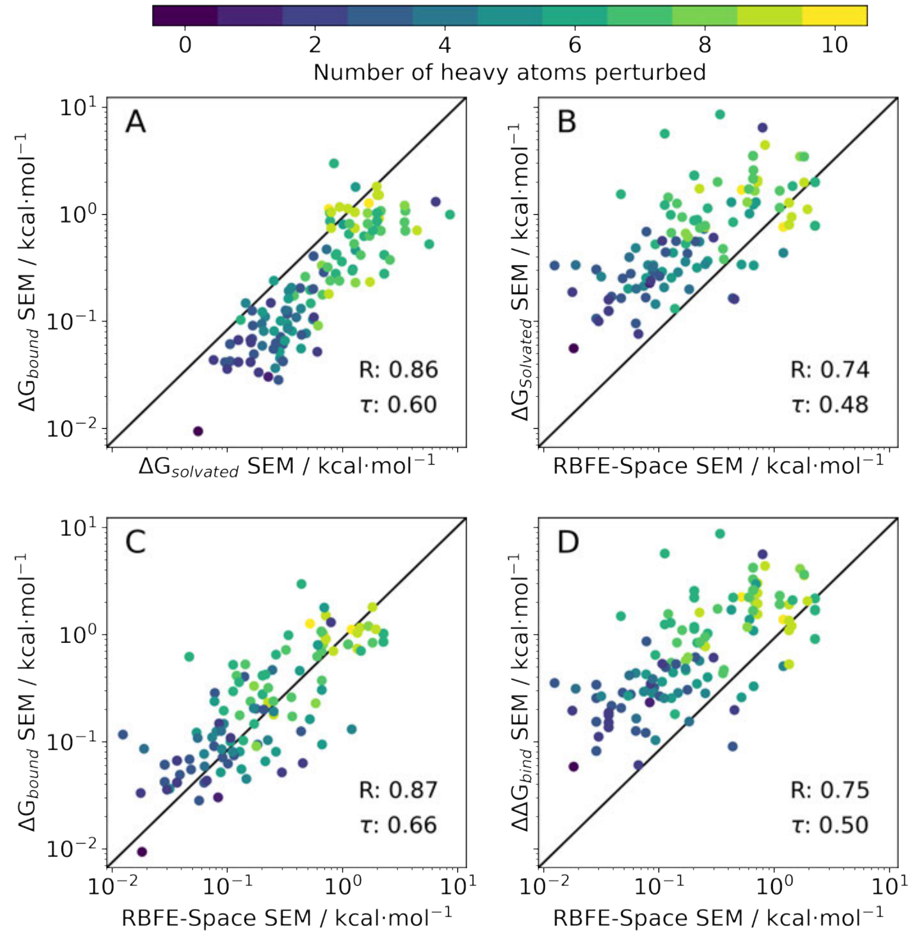


Figure 3.7: Correlation scatter plots of SEM values of 214 quintuplicate perturbations in different phases as extracted from publicly available RBFE benchmarking sets (n=9). The data are shown on a logarithmic scale and points are coloured by the number of heavy atoms that are perturbed in the perturbation (see colour range). Each panel has the data's Pearson R and Kendall τ annotated in its bottom right corner. **A**: Solvated versus bound SEMs of ligands with their original scaffold. **B**: RBFE-Space derivative (solvated) versus the original scaffold's perturbation in solvated phase. **C**: RBFE-Space derivative (solvated) versus the original scaffold's perturbation in bound phase. **D**: RBFE-Space derivative (solvated) versus the original scaffold's perturbation $\Delta\Delta G_{bind}$ value (obtained by $\Delta G_{solvated}$ - ΔG_{bound}).

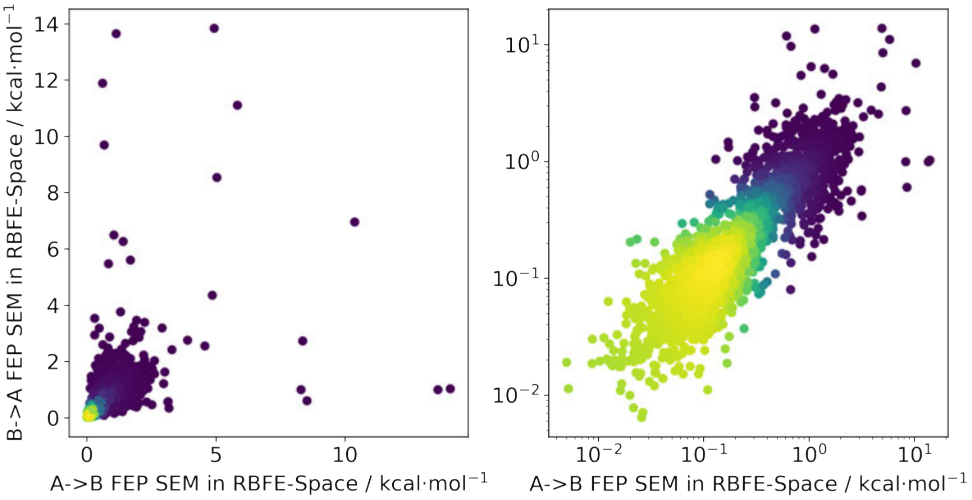


Figure 3.8: Comparisons of standard error of the mean (SEM) of the relative hydration free energy for all ligand pairs in RBFE-Space (n∼4000) between the two directions of a given bidirectional transformation, transforming from A→B (X axes) and back from B→A (Y axes). Shown are data on a linear scale (left-hand side) and on a logarithmic scale (right-hand side). Colour density shows the increase in data density as blue →yellow.

3.3.3 Machine-learning models can train on RBFE-Space to predict statistical fluctuations.

To train a machine-learning model on RBFE-Space, a graph neural network (GNN) approach was taken to describe molecular perturbations. This type of architecture was chosen because of its proven potential to learn molecular structures given enough data.[256–258] One major advantage of learning directly the molecular topology instead of pre-computed molecular descriptors is that no prior knowledge of influential descriptors is required. However, when training complex models with many parameters (such as GNNs) care must be taken to provide a sufficiently large training domain to make sure that weights have been optimised to a point where an understanding of chemical structure (or chemical perturbation, in this work's case) has been reached.[256,259] As RBFE-Space contains only 3864 points, we have opted for a pre-training and transfer-learning approach (figure 3.9) which is a technique that has recently gained popularity in chemistry.[253,260,261] In the *pre-training phase*, a cheaply-computed label, the relative estimated solubility (ΔESOL[251]) was computed for 1M randomly picked combinations of molecules in RBFE-Space and this training domain was used for pre-training the RBFENN model to learn molecular perturbations. Whereas any chemical descriptor could be picked for this application, ΔESOL is a suitable candidate because it is a relatively complex descriptor which prevents the RBFENN from focussing its learning on a specific chemical detail which would likely happen when learning on simpler properties such as molecular weight or lipophilicity. The pre-training protocol in this approach showed sufficient learning convergence after 100 epochs of training, at which point training was interrupted; the runtime for this step was approximately 9h.

After pre-training, the ΔESOL training domain was discarded and the GNN layers' weights of the RBFENN (figure 3.3C) were 'frozen', *i.e.* their weights were not allowed to be adjusted during training. This *transfer-learning phase* thus started with a RBFENN architecture that had already learned molecular perturbations. RBFE-Space was then used as a training domain to train the non-frozen layers in the model to predict $\widehat{SEM}$. Whereas validation MAE varied across replicates, models were observed to converge to 0.4-0.5 kcal·mol^{-1} MAE. In this step, global minimum training MAE values are shown to be higher than global minimum validation MAE values (figure 3.9B). This is likely because of the reduced number of trainable parameters (only weights in 3.3E are trained) in combination with the low-data regime, where the validation set (20% of RBFE-Space) results in occasional small dynamic ranges, skewing statistics on this subset. Because the shown MAE is a *global minimum* the large positive fluctuations due to these effects are not shown - these were especially high in the validation error.

To further maximise the RBFENN $\widehat{SEM}$ predictivity, a *fine-tuning phase* was performed where all weights of the RBFENN were 'un-frozen', *i.e.* all weights were allowed to be adjusted during training. The idea behind this approach is that the GNN component of the RBFENN can further optimise its $\widehat{SEM}$ predictivity in unison with the remaining layers of the model. Learning curves for this phase show further training of the model, lowering the ensemble MAE to 0.1-0.2 kcal·mol^{-1}. Because of the high number of parameters (n=1,827,712) in this phase rapid overfitting of the training set was observed (training MAE rapidly lowering while validation MAE started increasing); see rapidly declining gray lines in figure 3.9C - early stopping was triggered for most replicates between 200-300 epochs because the validation error did not show any further global decrease. For each replicate, the best-performing (*i.e.* lowest validation MAE) model at epoch n was extracted and used as the final model. In cases where fine-tuning showed no decrease in validation MAE over the best model in the transfer-learning phase, the top-performing model of the transfer-learning phase was used.

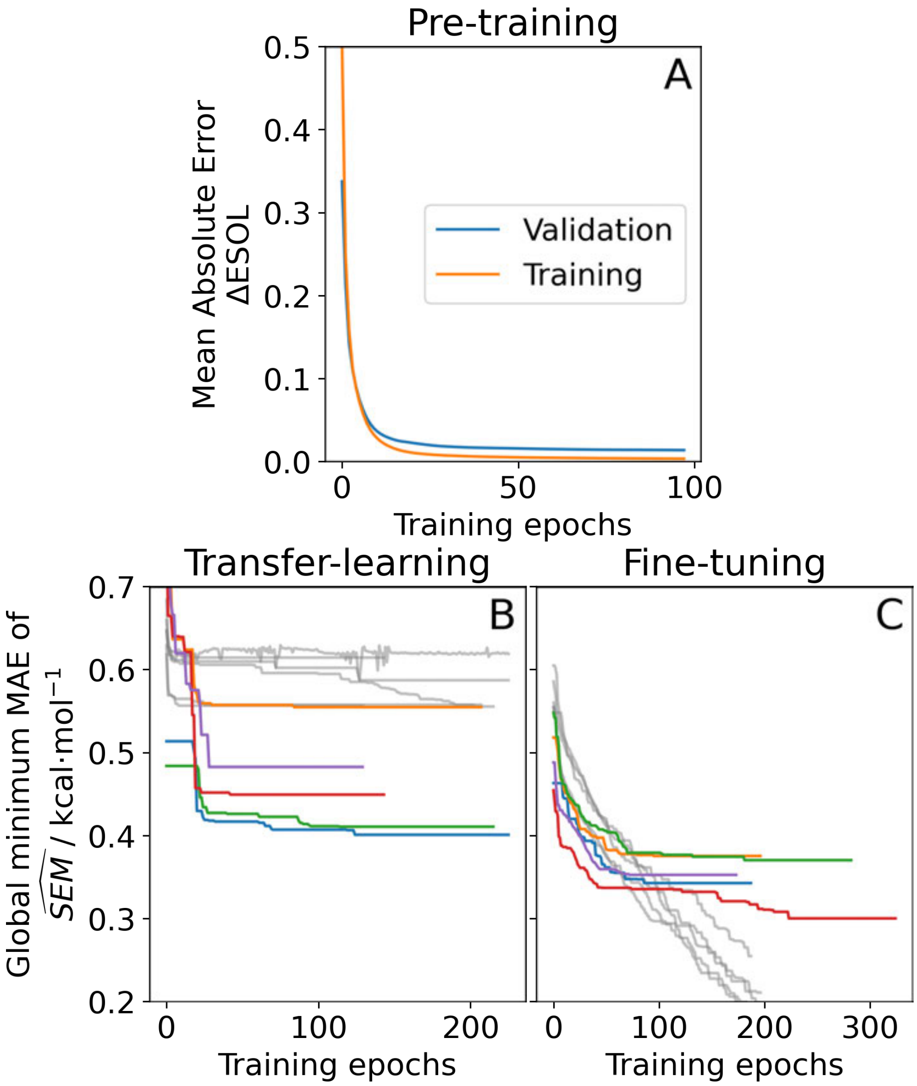


Figure 3.9: Learning curves of the three phases of the RBFENN training protocol for predicting SEM values of RBFE perturbations. **A**: *pre-training phase*, where a cheaply-computed continuous label (the relative estimated solubility, ΔESOL[251]) was used to generate a training set of 1M data points using RBFE-Space ligands. Shown are the validation error and training mean absolute errors (MAE; blue and orange, resp.) per epoch. **B**: *transfer-learning phase*, where the message-passing component (figure 3.3C) weights were forced static ('frozen'), allowing the remaining layers of the RBFENN to learn to predict SEM rather than ΔESOL while the chemistry-processing layers' weights are retained. Shown in colours are validation MAEs of predicted SEM in kcal·mol^{-1} for five replicates. Shown in gray are training MAEs; all error values are reported as their global minimum value. **C**: *fine-tuning phase*, where the message-passing component of the RBFENN architecture is allowed to train (*i.e.* weights are 'un-frozen') in an effort to further increase SEM predictivity (panel formatting same as for panel B).

3.3.4 Applications of the trained RBFENN.

In this section, the RBFENN will be applied to two RBFE benchmarking congeneric series: TYK2 and TNKS2. Although these test sets are present in the training set (figure 3.6), this is assumed acceptable for this work as the majority of perturbations in RBFE-Space have duplicates in other congeneric series. Leaving the TYK2/TNKS2 perturbations out of the training set would thus remove a large number of perturbations that exist in any congeneric series such as (de-)halogenations and other 1-3 heavy atom perturbations. As there is such large overlap in perturbation-space between congeneric series, the majority of TYK2/TNKS2 perturbation would be present in other series as well and would have to be included in RBFE-Space in such a scheme anyway.

Increasing λ windows in RBFE decreases statistical fluctuations

Prior to network generation, λ allocations were benchmarked in the context of RBFE statistical fluctuation (figure 3.10) in the solvated phase for six highly reliable and six highly unreliable perturbations in RBFE-Space. The statistical fluctuation (here expressed as $\mathrm{SEM}_{\overline{\Delta G}_{solvated}}$ of 5 replicates) was recorded at increasing numbers of equidistant λ windows used for MBAR analysis: 3, 5, 9, 17 and 33. For both types of perturbations an exponential decay in $\mathrm{SEM}_{\overline{\Delta G}_{solvated}}$ was observed; typically convergence was reached at 15-20 λ windows, suggesting further sampling is likely not necessary in RBFE calculations with SOMD for the solvated phase, even for highly unreliable perturbations.

The main objective of this analysis was to determine whether the 11 λ windows protocol used in the generation of RBFE-Space was sufficient to describe statistical fluctuations of RBFE perturbations. Although at n=11 convergence does not seem to have been reached in all cases, this number does offer a reasonable approximation of the statistical fluctuation with acceptable sampling cost (note the figure's varying y axis limits). Notably, RBFENN $\widehat{SEM}$ predictions consistently show the correct order of magnitude for all 12 perturbations described in this analysis. This confirms that the $\widehat{SEM}$ estimator can be used to discriminate perturbations with high SF from perturbations with low SF.

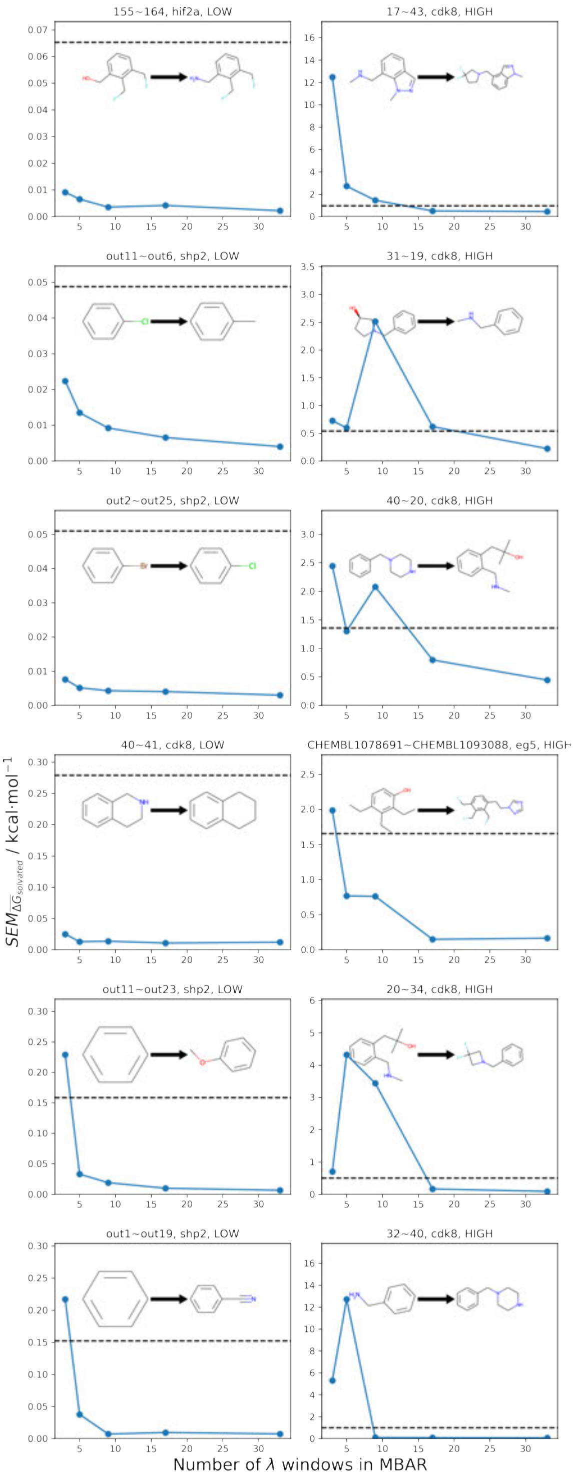


Figure 3.10: Molecular transformations and their statistical fluctuation represented by standard error of the mean (SEM) across five replicates shown at different numbers of λ windows. The title of each plot shows the perturbation name (the tilde signifies a transformation), the protein target and whether the expected statistical fluctuation is LOW or HIGH. The horizontal dashed line in each plot is the $\widehat{SEM}$ value as predicted by the RBFENN described in this work. Reported SEM values are $SEM_{\overline{\Delta G}_{solvated}}$ values in kcal·mol^{-1} extracted from the simulations run for the generation of the RBFE-Space training domain.

RBFENN-based RBFE networks are distinct from state-of-the-art RBFE networks

RBFE networks generated by LOMAP using LOMAP-Score or RBFENN as edge similarity metrics were compared for the entire public RBFE benchmark set (table 3.1). RBFE networks for the TYK2 series show the the highest degree of overlap (55%) between the two methodologies. Across the dataset overlaps range from 11% to 47% with an average value of 32%. Some overlap between the methodologies is expected since both input metrics succeed at modelling the SF to some degree which results in similar assumptions in generating either network. One series of note is Thrombin (n=11) which shows 0% overlap. As the compounds in this series are structurally highly similar it is plausible that a large fraction of possible networks minimise equally well statistical fluctuations. However due to the low number of compounds in the series (n=8) it is difficult to make statistically-sound comparisons of networks performance,[226].

Because of LOMAP's cluster minimisation and connection algorithm there is typically some variance (± 3-4 edges) in the number of edges selected for a congeneric series of $N_{ligands}$. We observe in general a relationship of $n_{edges} \approx 1.4 \cdot N_{ligands}$. Although the number of edges suggested consistently differed between LOMAP-Score and RBFENN networks, no methodology gave a consistently larger network. In general the network overlap % between the two methodologies decreases with congeneric series size (thrombin aside). This likely reflects the combinatorial explosion in the number of distinct networks that can be proposed as $N_{ligands}$ increases.

Table 3.1: Comparison of RBFENN and LOMAP-Score RBFE networks for all publicly available RBFE benchmarking series in terms of network size (n edges) and overlap (%). Rows were sorted by ligand series size (N ligands) in descending order. The network overlap was computed by counting the number of overlapping edges between the two networks and computing the mean percentage with respect to the two networks and rounding to the nearest number. EG5 was excluded from this comparison as benzene grafting failed for the majority of the network due to overly complex perturbations. a-c: these ligand series are further analysed in sections 3.3.4, 3.3.4 and 3.3.4, respectively

Target	Series size (N)	LOMAP-Score network (n)	Network overlap (%)	RBFENN network (n)
SYK	44	63	25	64
MCL1	42	61	11	59
HIF2a	42	59	26	64
PFKFB3	40	57	35	60
BACE	36	52	26	51
P38(MAPK14)	34	45	23	47
CDK8	33	50	44	45
TNKS2[a]	21	27	23	24
SHP2	26	38	42	38
PTP1B	23	32	39	33
PDE2	21	29	35	27
Jnk1	21	27	33	27
CDK2	16	21	47	21
TYK2[b]	16	23	55	27
c-MET	12	15	37	17
Thrombin	11	13	0	14
Galectin[c]	8	10	40	10

A visual example of the networks proposed with RBFENN or LOMAP-Score for Galectin RBFE benchmarking series is presented in figure 3.11. Both methodologies make reasonable suggestions, although there is only 40% network overlap between the two network topologies. As visual comparison is a qualitative measurement of RBFE network generation performance and because one of the main objectives of the data-driven approach is to remove the subjective component in the field, a more quantitative approach is pursued in this work.

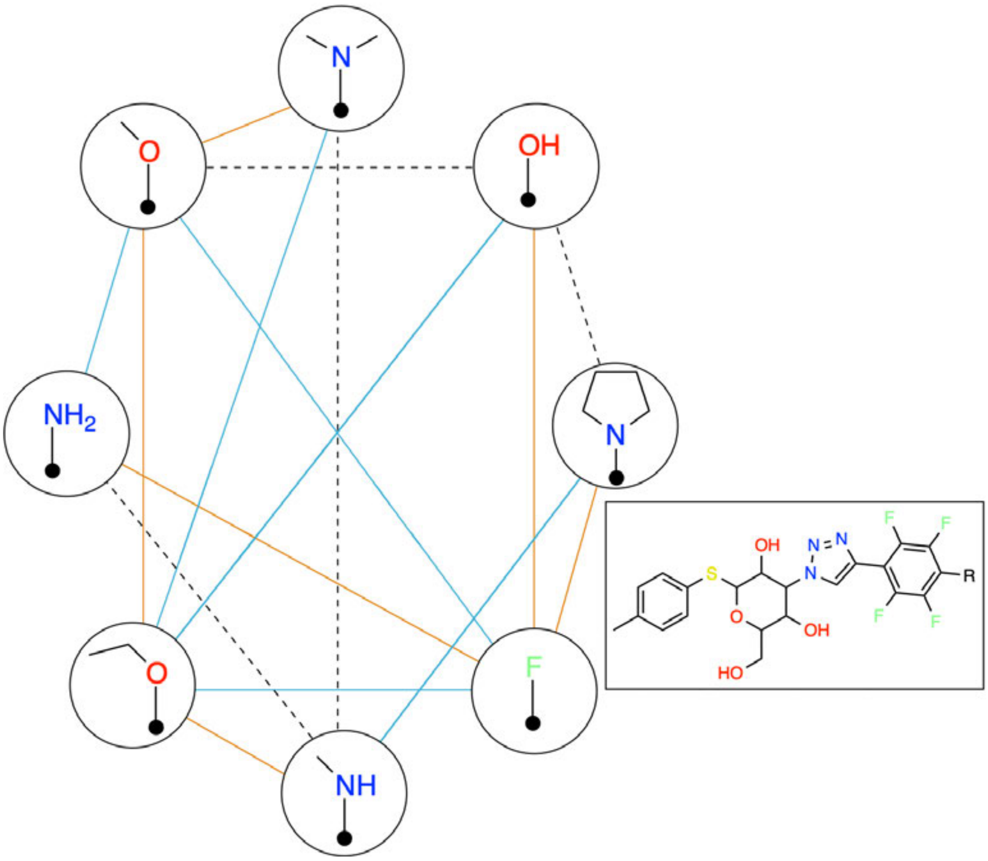


Figure 3.11: Example RBFE networks on the Galectin RBFE benchmarking congeneric series (N=8). Shown are the state-of-the-art LOMAP-Score approach (orange edges; n=10) and novel data-driven approach presented in this work (blue edges; n=10). Edges that are present in both RBFE networks are represented as singular black dashed lines. Ligand scaffolds were replaced with black circles for simplification purposes. The ligand scaffold is shown in the right-hand side box with the R-group location on the right-hand side of the structure.

RBFENN predicts inexpensively the accuracy of RBFE calculations on TYK2

As the performance of RBFE calculations is determined by the errors made along each edge of the chosen network, different network topologies should result in a difference in the estimation of binding free energies (ΔG). A quantitative approach for comparing RBFE networks is thus possible by processing each edge of the networks with the same RBFE protocol, and comparing the estimated binding free energies with experimental data.

To carry out this assessment, the non-receptor tyrosine kinase TYK2 congeneric series[208] was chosen as it is a challenging RBFE benchmarking set of sufficient size to allow reliable statistical analysis[226]. The TYK2 series also involves a mixture of straightforward ligand sub-groups and more challenging perturbations that involve ring-changes.[212] Additionally, the TYK2 series has been used recently in several RBFE works investigating network generation and machine learning potentials.[235,262] For this series (16 ligands), RBFE was run for all possible perturbations in a single direction ($n = \frac{16^2-16}{2} = 120$). Monodirectional edges were chosen with the purpose of halving computational cost. The signs of the relative binding free energy predictions for the 120 edges in this RBFE run were inverted to obtain the remaining 120 RBFE predictions, resulting in a bidirectional fully connected network with 240 edges. The validity of this assumption was supported by data generated during creation of the RBFE-Space training set (figure 3.8).

As stated previously, an ideal RBFE network generator will contain edges with low deviation from experimental measures. Thus edge scoring metrics that correlate more strongly with $|\Delta\Delta \mathrm{G}_{offset}|$ values should select more accurate networks. To verify this, the statistical performances of available heuristics were compared to the $|\Delta\Delta \mathrm{G}_{offset}|$ values (Eq 3.1) gathered from the fully connected TYK2 network. The data in Figure 3.12A shows that $\Delta\Delta G_{bind}SEM$ correlates well with $|\Delta\Delta \mathrm{G}_{offset}|$, therefore supporting the hypothesis that selecting edges with lower SF will lead to RBFE networks with lower errors. Figure 3.12B shows that this correlation is maintained (albeit more weakly) with RBFE-Space $SEM_{\overline{\Delta G}_{solvated}}$ values.

Table 3.2 summarises how different SF predictors correlate with offset values. As expected $\Delta\Delta G_{bind}$ shows the strongest correlation, but this metric is computationally too intensive to be of practical use for network generation. The inexpensive SF estimators $\widehat{SEM}$ and LOMAP-Score show comparable correlation with offset deviations. Surprisingly edge scoring based on ECFP6 shows no relationship with offset deviations. This is likely because the fingerprint is relatively insensitive to the different perturbations, with most edges assigned a similarity score of around 0.7.

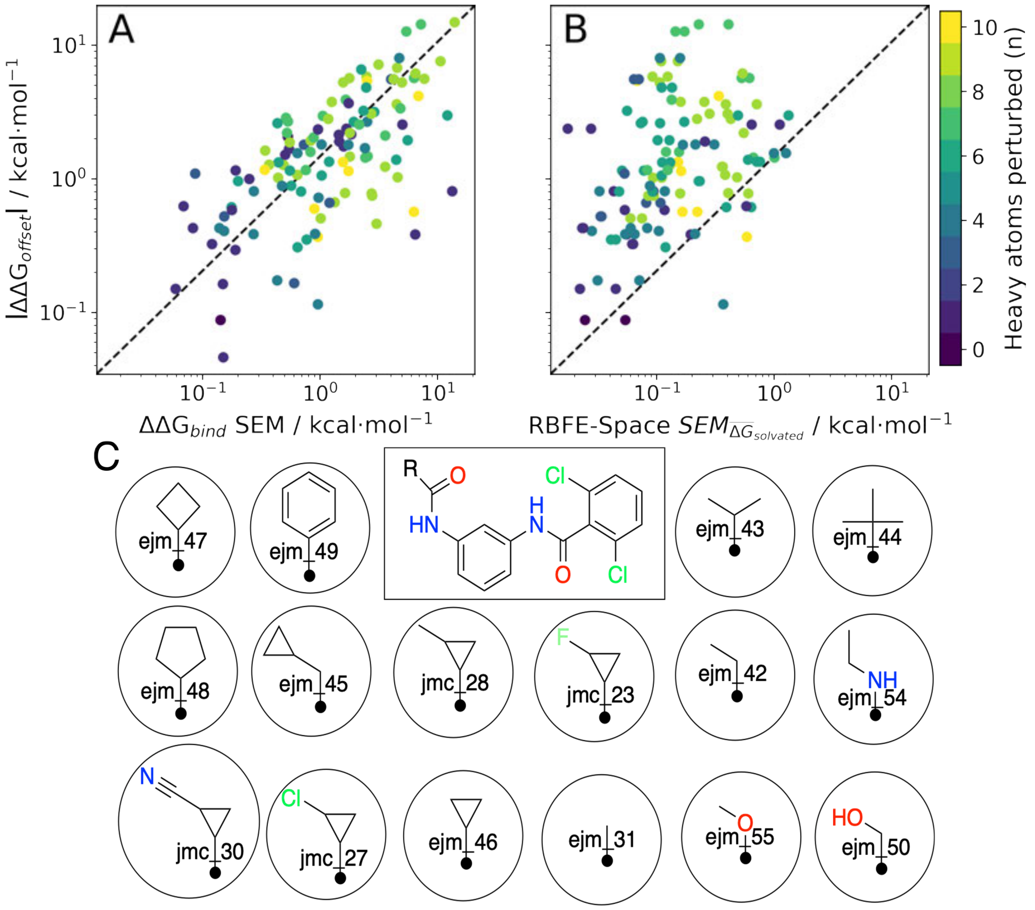


Figure 3.12: Scatter plots of $|\Delta\Delta G_{offset}|$ vs (**A**) $\Delta\Delta G_{bind}$ SEM values for all possible edges in the TYK2 RBFE benchmarking series (n=120), **B** RBFE-Space SEM values for perturbations included in RBFE-Space (n=124). The colourbar shows the increase in the number of heavy atoms perturbed per perturbation in the scatter plots. See table 3.2 for statistical analyses corresponding to these array comparisons and see figure 3.13 for an extended version of this figure. **C**: scaffold (centre) and analogs in the TYK2 RBFE benchmarking series annotated with ligand names used throughout this work.

Table 3.2: Statistical performances of various heuristics versus the $|\Delta\Delta G_{offset}|$ for all possible edges in the TYK2 RBFE benchmarking series (n=120). *: only perturbations included in RBFE-Space were included (n=124; see 3.2.1). See figure 3.13 for scatterplots corresponding to these array comparisons, and figure 3.14 for distributions of these heuristics.

	Pearson r	Kendall τ
$SEM_{\overline{\Delta\Delta G}_{bind}}$	0.63	0.46
RBFE-Space $SEM_{\overline{\Delta G}_{solvated}}{}^{*}$	0.37	0.28
RBFENN $\widehat{SEM}$	0.41	0.25
LOMAP-Score	0.42	0.33
ECFP6 similarity	-0.03	-0.01

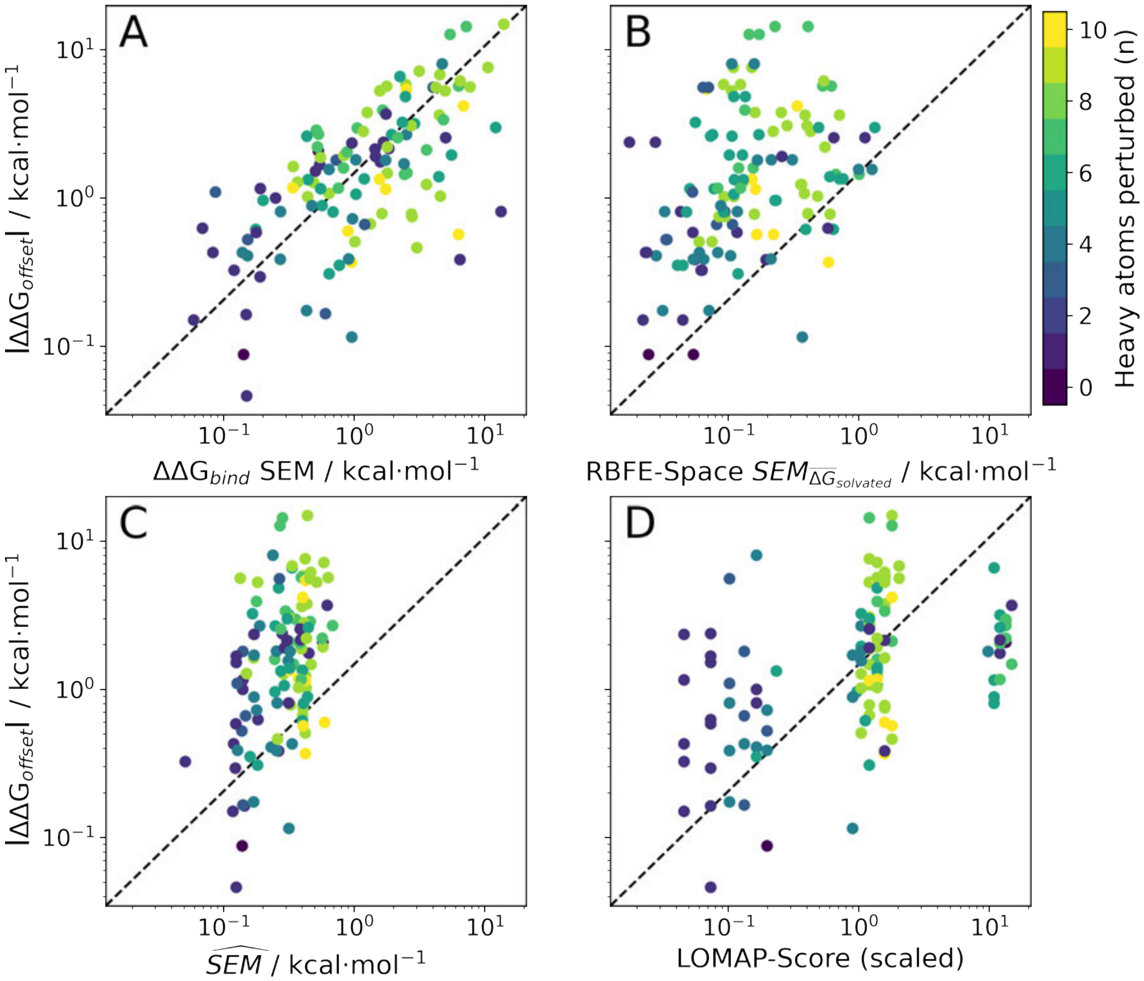


Figure 3.13: Scatter plots of various heuristics versus the $|\Delta\Delta G_{offset}|$ for all possible edges in the TYK2 RBFE benchmarking series (n=240). For RBFE-Space SEM values (**B**) only transformations included in RBFE-Space were included (n=124; see main text body). The colourbar shows the increase in the number of heavy atoms perturbed per perturbation in the scatter plots. See table 2 (main text body) for statistical analyses corresponding to these array comparisons.

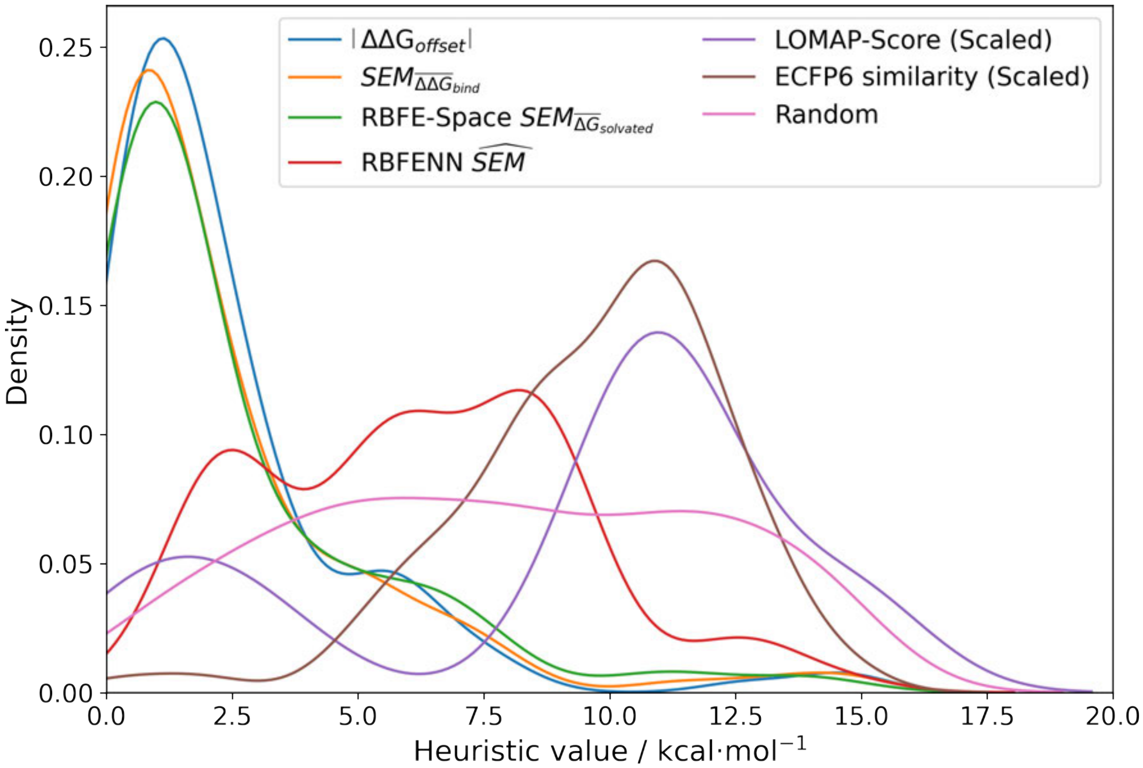


Figure 3.14: Array distributions using a kernel density estimation. Shown are the distributions of a number of heuristics used throughout this work (see legend). These represent statistical fluctuations in RBFE transformations in this case applied to all possible edges of the TYK2 RBFE benchmarking series. The ligand series contained 240 transformations, so $n = 240$ for all shown input heuristics; except $SEM_{\overline{\Delta G}_{solvated}}$ which contains only 124.

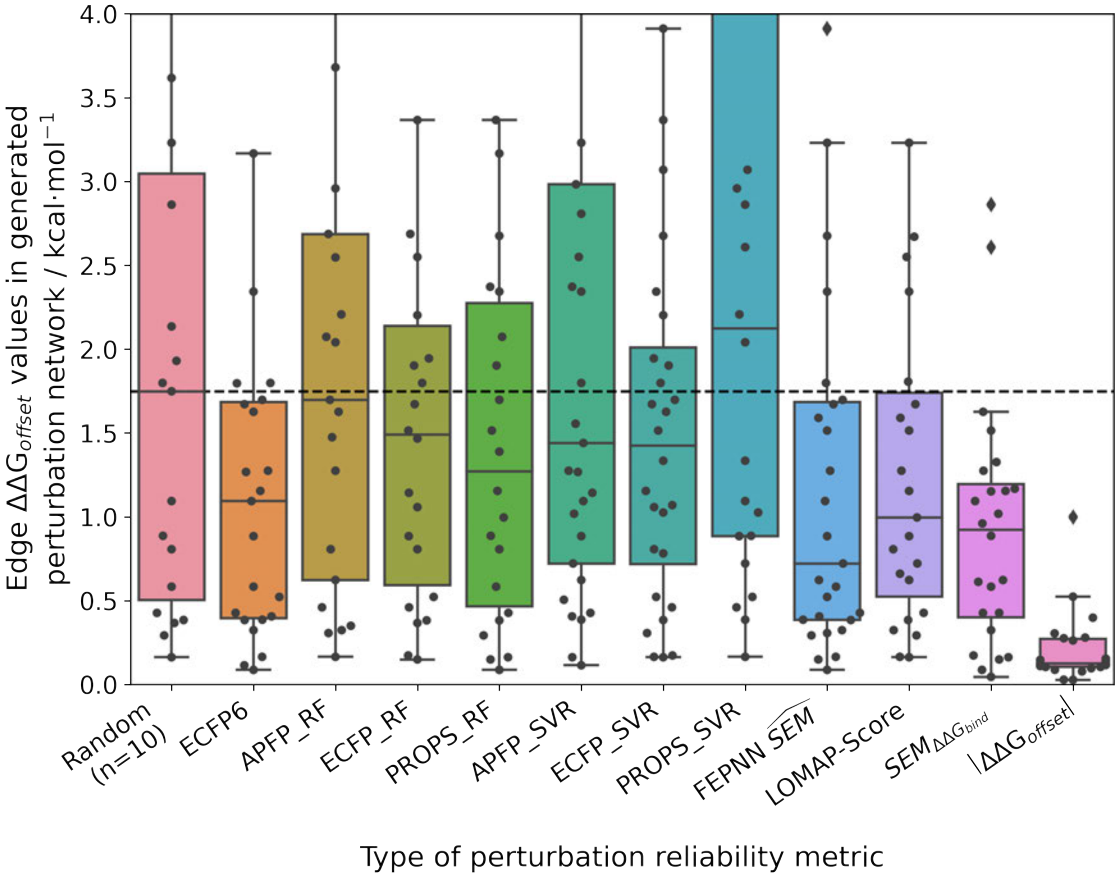


Figure 3.15: Boxplots depicting the distribution of $|\Delta\Delta \mathrm{G}_{offset}|$ of edges that constituted the RBFE networks generated by various input metrics to LOMAP. The Random input metric was repeated ten times to ensure sampling of a diverse set of networks was achieved. ECFP6 is the ECFP6 tanimoto similarity between the original (*i.e.* with original scaffold) ligands. For RBFENN $\widehat{SEM}$, $SEM_{\Delta\Delta G_{bind}}$ and $|\Delta\Delta \mathrm{G}_{offset}|$ the input values were scaled to an inverse 0-1 range to fit the LOMAP algorithm. The horizontal dashed line denotes the median $|\Delta\Delta \mathrm{G}_{offset}|$ value of the Random networks.

RBFENN matches state-of-the-art for TYK2 RBFE network generation

To the best of our knowledge, this work describes the first fully-connected (FC) RBFE network for the TYK2 series. This dataset allows enumeration of all possible RBFE networks: given a network generator and an edge scoring heuristic the network edge accuracy with respect to experimental data can be determined by looking up edge results in the pre-computed FC network. The number of possible networks is vast. For this dataset there are 16^{14}=7.2·10^{17} minimally-connected networks (*i.e.* 15 edges with all nodes included in the network).[263] The actual number of networks theoretically considered by LOMAP is much greater because of additional heuristics to introduce extra cycle closures. In this analysis, six different edge scoring heuristics are used with LOMAP (random, $|\Delta\Delta\mathrm{G}_{offset}|$, RFTOP, ECFP6, RBFENN $\widehat{SEM}$, LOMAP-Score) to generate RBFE networks. The RBFE network topologies per network type can be found in figures 3.18-3.23.

The random protocol that assigns a random score to each edge is a negative control. Figure 3.16G-H shows that repeated applications of this protocol lead to results with significant variability (since the network topology varies between repeats), and on average poor correlation (R = 0.2 ±0.2,τ = 0.15 ±0.15, $MUE = 1.8 \pm 0.2 kcal$·mol^{-1}, n = 20, figure 3.16G/H and 3.17). The $|\Delta\Delta\mathrm{G}_{offset}|$ protocol that assigns a score to each edge by scaling the offset values computed for the fully connected network is a positive control (figure 3.16F). This protocol leads to significantly more accurate results with low uncertainty (R $\sim$ 0.9, $\tau \sim$ 0.72, MUE $\sim$ 0.45 kcal·mol^{-1}, n = 22) and represents near optimal results that may be achieved with the RBFE protocol used here to process each edge of the network (Figure 3.16G-H). This optimal network allocated 22 edges to the TYK2 dataset. Manually augmented RBFE networks used in previous studies for this series contain 30-40 edges.[208,212] While it could be expected that increasing the number of edges present in the network would increases the accuracy of the results we find that this is not the case with the fully connected network (figure 3.16A). The accuracy of the FC (n = 120 edges) network is lower than the network proposed by LOMAP using the —Δ_{offset}— metric (R

$\sim$ 0.67, τ $\sim$ 0.43 , MUE $\sim$ 0.75 kcal·mol^{-1}). The reason this occurs is that the weighted least squares regression algorithm used in this work to convert $\Delta\Delta G$ values into ΔG values penalises insufficiently poorly converged edges, which introduces noise in the final free energy estimates. Example edges in TYK2 that were associated with high noise (standard error across a quintuplicate) were **ejm_49→ejm_54** ($\sim$14 kcal·mol^{-1}), **ejm_44→ejm_49** ($\sim$7 kcal·mol^{-1}) and **ejm_44→ejm_45** ($\sim$7 kcal·mol^{-1}) (figure 3.12C) . This highlights the need to exclude edges with poorly converged $\Delta\Delta G$ values from an RBFE network analysis.

Star-shaped networks (where all ligands are perturbed to a single reference ligand) were also explored in this analysis. Such network topologies offer the lowest processing cost (n=15) but it was found that for all 16 possible networks this choice of design resulted in poor RBFE performance on this ligand series (figure 3.16G-H and 3.17). This poor performance is likely due to the seven ligands in the TYK2 series that require growing or vanishing of cyclic structures which present difficulties for the RBFE protocol used in this study.[212] **ejm_44** and **ejm_48** are the worst reference compounds, resulting in R $\sim$ 0.33&0.20, τ $\sim$ -0.32&-0.12 and MUE $\sim$ 4.35&2.88 kcal·mol^{-1}, respectively. **ejm_31** is the best reference compound to use (R $\sim$ 0.47, τ $\sim$ 0.33 , MUE $\sim$ 0.96 kcal·mol^{-1}) because any R-group can be directly grown onto it rather than having to make direct substitutions.The poor performance of this approach compared to state-of-the-art network generators highlights the need for increased scalability on large-scale RBFE campaigns where star-shaped networks are frequently used.

A comparison to experiment is not shown for all shallow ML models (RF and SVM with varying descriptors), but an analysis on $|\Delta\Delta G_{offset}|$ distribution per suggested network shows that these models generate RBFE networks as poor as random edge scoring (figure 3.15). The top performing shallow ML model (random forest with molecular properties as descriptors, figure 3.16B) RBFE network shows reasonable predictive power (R $\sim$ 0.5, τ $\sim$ 0.25, MUE $\sim$ 1 kcal·mol^{-1} n=22, figure 3.16G-H). Pure molecular similarity of the original ligand scaffolds (ECFP6 tanimoto, figure

3.16C) shows network performance comparable to random edge selection (R $\sim$ 0.25, $\tau \sim 0.17$, MUE $\sim$ 1.2 kcal·mol^{-1} n=23, figure 3.16G-H).

For TYK2, both the RBFENN and the LOMAP-Score RBFE networks show remarkably similar statistical performance (figure 3.16D and E). This is likely because 14 edges are shared between the two networks. The results (figure 3.16G-H) approach the accuracy of the positive control (R $\sim$ 0.75, $\tau \sim 0.55$, MUE $\sim$ 0.55 kcal·mol^{-1}, n=23-27, figure 3.16G-H).

The main topological differences between the RBFENN and the LOMAP-Score networks is related to how each network handles ring changes. Eight ligands feature different cyclical R-group (*i.e.* outside the maximum common substructure, MCS; see figures 3.21 and 3.22). The LOMAP-Score network primarily opts for connecting these to a hub ligand **ejm_31** (preferring perturbations that follow the pattern `MCS-C`→ `MCS-C-Cycle`). The RBFENN network also uses **ejm_31** as a hub for scaffold hopping, but also introduces a second hub (**ejm_42**) as well. The latter perturbations exploit the pattern `MCS-C-C`→ `MCS-C-Cycle`.

Comparison of these networks with the network proposed using the $|\Delta\Delta G_{offset}|$ metric (figure 3.23) suggests that neither of the hub approaches are optimal: instead, this network does not contain any hub ligands in its topology. There appear to be occasional perturbations that are reliable that typically would not be suggested by LOMAP-Score rule-based approaches such as **ejm_50**→ **ejm_45** (`MCS-C-OH`→ `MCS-C-cyclopropyl`), **ejm_44**→ **ejm_47** (`MCS-isopropyl`→ `MCS-cyclobutyl`) and even a direct ring transmutation in **ejm_49**→ **ejm_48** (`MCS-benzene`→ `MCS-cyclopentane`).

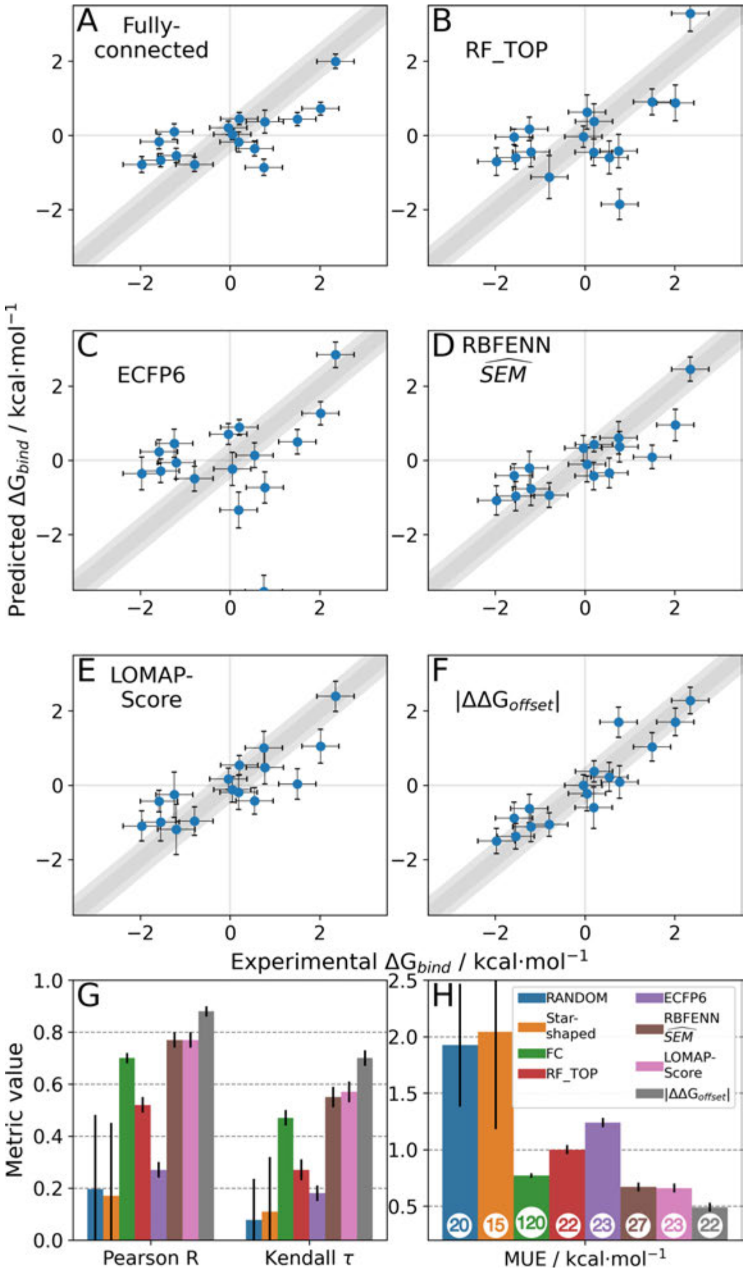


Figure 3.16: RBFE predictions on the TYK2 RBFE benchmarking series versus experimental ligand binding affinities using various RBFE network design methodologies. **A-F**: predicted ΔG_{bind} versus experimental ΔG_{bind} in kcal·mol^{-1} for the fully-connected network, and networks generated using the top-performing shallow ML model (random forest with molecular properties), ECFP6 tanimoto similarity on original ligand scaffolds, RBFENN $\widehat{SEM}$, LOMAP-Score and $|\Delta\Delta G_{offset}|$ values, respectively. Shown data is per-ligand relative binding free energy obtained using a weighted least squares approach. Error bars depict statistical uncertainty of each prediction (SEM) and experimental measure. Each plot is annotated with quadrant lines and a 1/2 kcal·mol^{-1} confidence region (dark gray, gray, resp.). **G/H**: statistical performance calculated using the data shown in A-F as well as star-shaped and random perturbation networks. In *H*, the number of edges per network is annotated on each bar. Depicted error bars show the 95% CI of a bootstrapping approach with 10,000 repeats except for *RANDOM* and *Star-shaped* statistics where an average and standard deviation is shown (10 random repeats or all 16 possible networks).

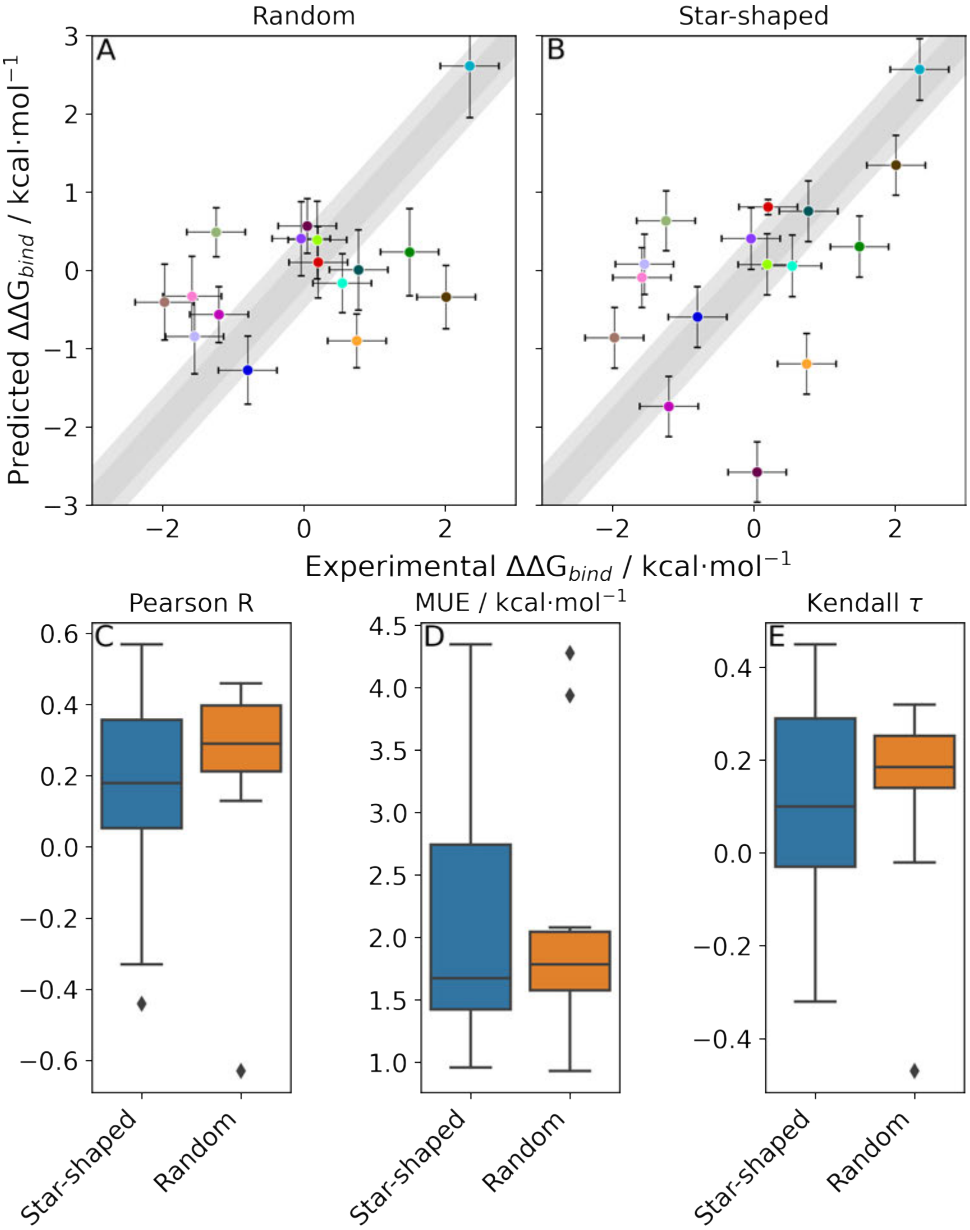


Figure 3.17: Comparison of predictive performances for TYK2 of perturbation networks generated using random selection of edges and the star-shaped approach. **A/B**: scatterplots of representative (*i.e.* $n = 1$) random and star-shaped networks' RBFE predictions compared to experimental measures in kcal·mol^{-1}. Ligands are coloured for direct comparison of positioning between the two plots. **C-E**: boxplots showing distributions of statistical performances for the complete collection of networks for both star-shaped ($n = 16$) and random ($n = 10$) network approaches.

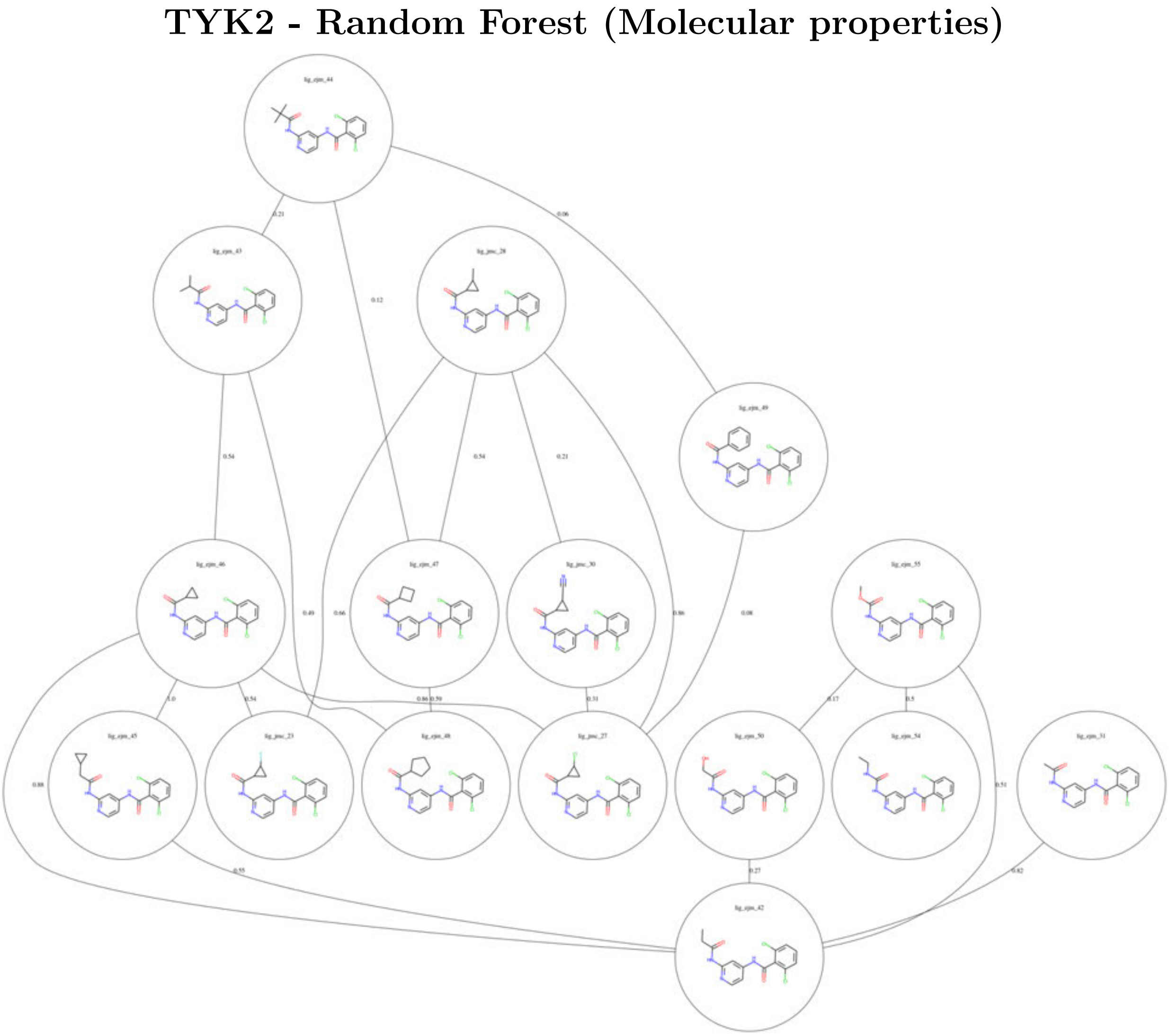

TYK2 - Random Forest (Molecular properties)

Figure 3.18: The TYK2 perturbation network as suggested by LOMAP using $\widehat{SEM}$ predicted by a random forest using molecular descriptors as input. Each node in the network contains the molecular structure and the ligand name; each edge in the network is annotated with the predicted $\widehat{SEM}$ value.

TYK2 - ECFP6

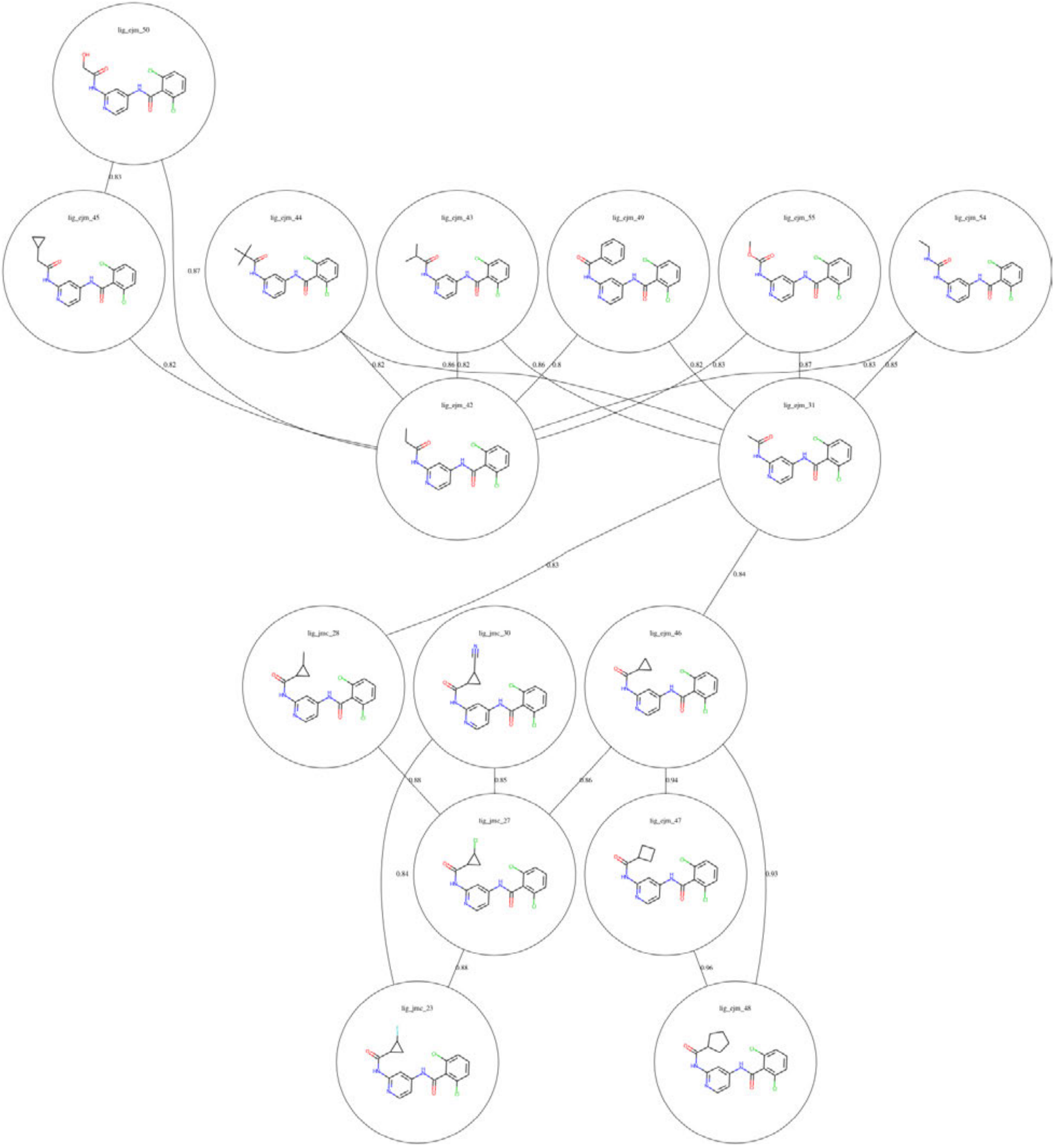


Figure 3.19: The TYK2 perturbation network as suggested by LOMAP using ECFP6 tanimoto similarities (on original ligands) as input. Each node in the network contains the molecular structure and the ligand name; each edge in the network is annotated with the similarity value.

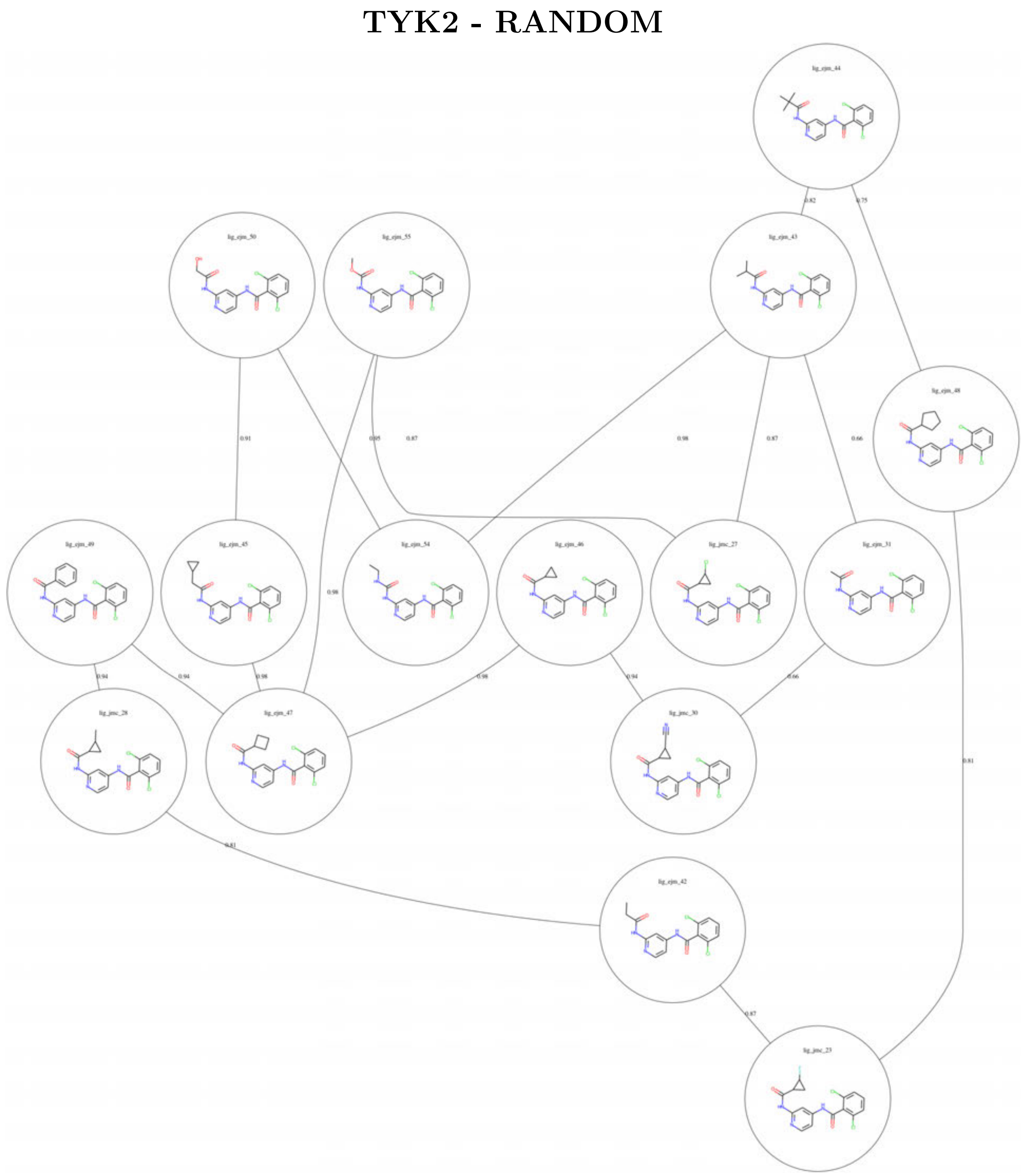


Figure 3.20: The TYK2 perturbation network as suggested by LOMAP using random values as input. Each node in the network contains the molecular structure and the ligand name; each edge in the network is annotated with the random value.

TYK2 - RBFENN

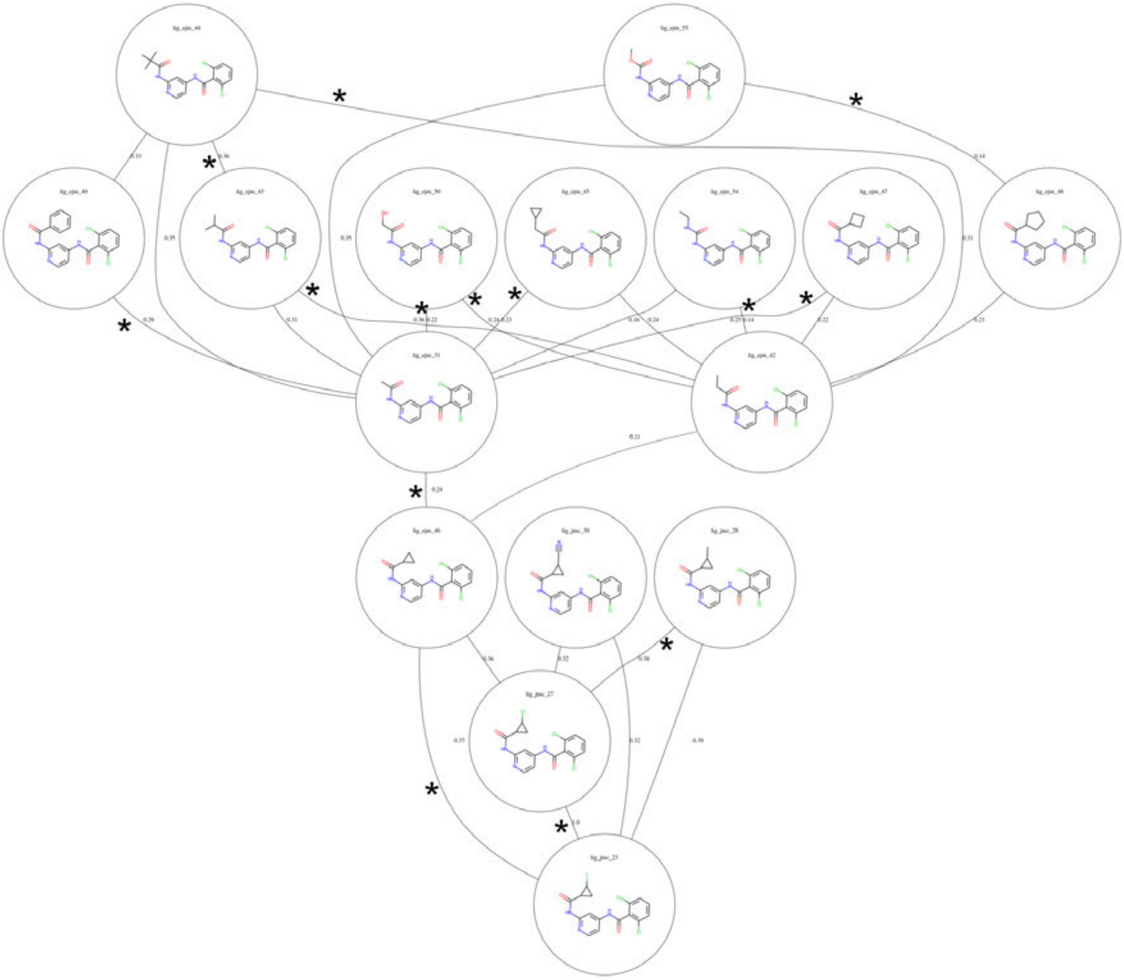
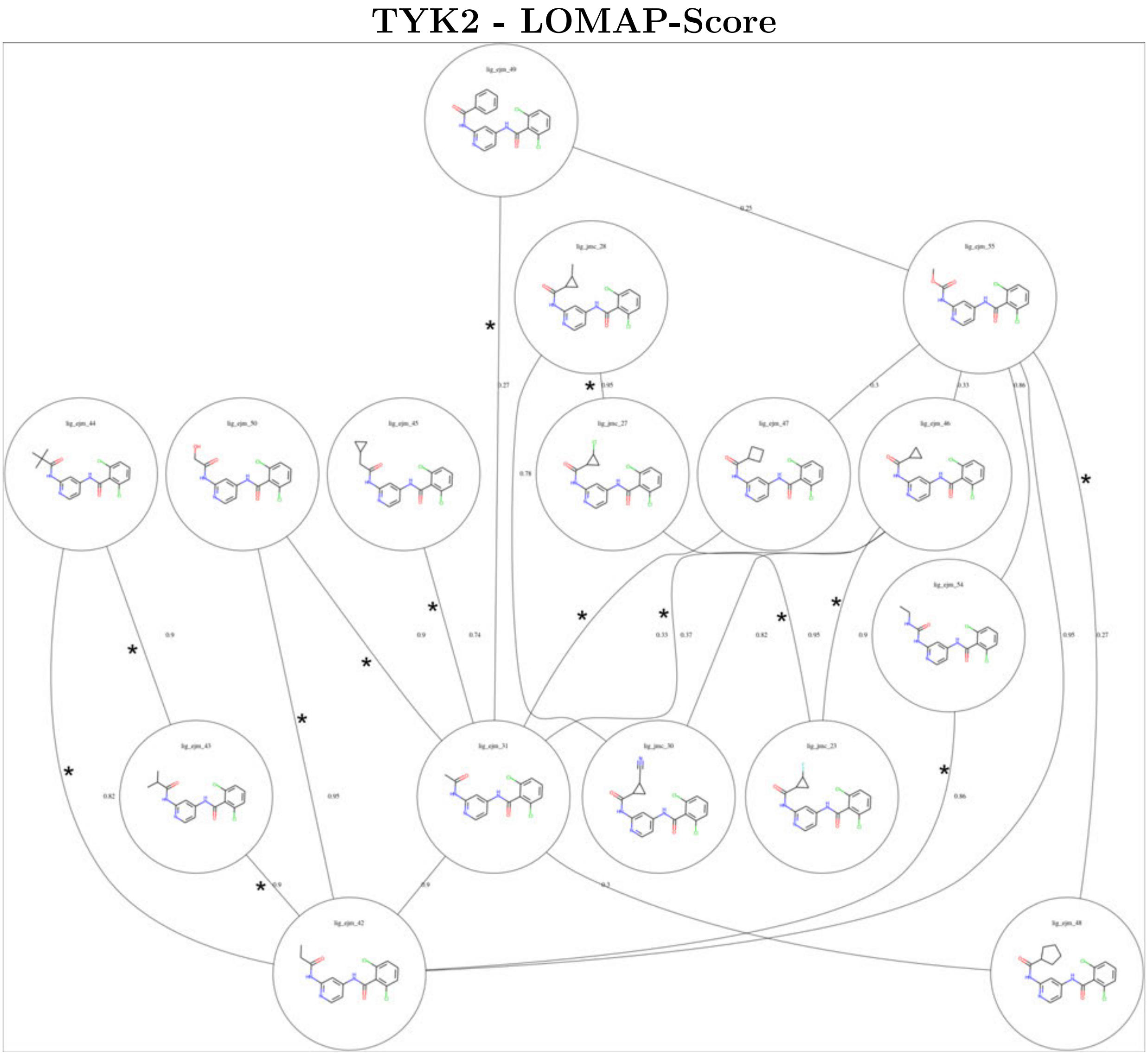


Figure 3.21: The TYK2 perturbation network as suggested by LOMAP using the RBFENN-predicted $\widehat{SEM}$ score as input. Each node in the network contains the molecular structure and the ligand name; each edge in the network is annotated with the RBFENN-predicted $\widehat{SEM}$ value that has been scaled to [0-1] to allow proper handling by the LOMAP algorithm. Asterisks (*) indicate edges that are shared between the RBFENN and LOMAP networks.



TYK2 - LOMAP-Score

Figure 3.22: The TYK2 perturbation network as suggested by LOMAP using the LOMAP-Score as input. Each node in the network contains the molecular structure and the ligand name; each edge in the network is annotated with the assigned LOMAP-Score value. Asterisks (*) indicate edges that are shared between the RBFENN and LOMAP networks.

TYK2 - $|\Delta\Delta \mathrm{G}_{offset}|$

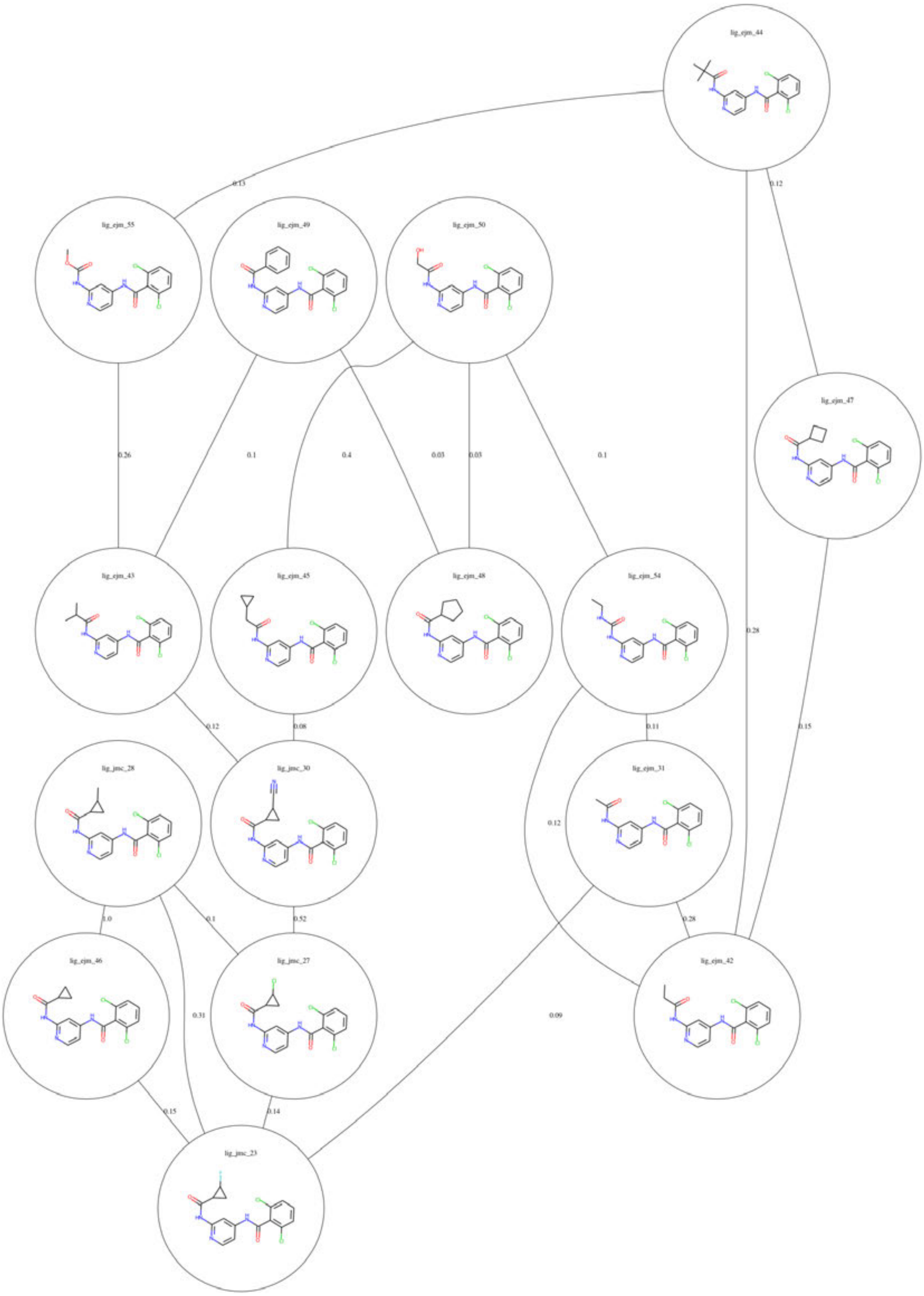


Figure 3.23: The statistically optimal TYK2 perturbation network as suggested by LOMAP using $|\Delta\Delta \mathrm{G}_{offset}|$ values as input. Each node in the network contains the molecular structure and the ligand name; each edge in the network is annotated with the $|\Delta\Delta \mathrm{G}_{offset}|$ value that has been scaled to [0-1] to allow proper handling by the LOMAP algorithm.

RBFENN matches state-of-the-art performance for automated TNKS2 RBFE network generation

The TNKS2 series was selected for additional testing because: it is part of a newer extended benchmark set that has been less studied than the FEP+ set (which includes TYK2); it involves fewer ring changes than TYK2; it contains multiple R-group sites at different sections of the ligand scaffold. Note that six +1 net charge ligands (**8a-f**) were excluded from this series as charge perturbations were considered out of scope for this work, meaning the series included n=21 ligands.

For TNKS2 only the RBFENN and LOMAP-Score network edges were simulated in quintuplicates (see figure 3.25 and 3.26 for RBFE networks). For this series, a 'dynamic' representation is used to investigate statistical performance when adding replicates (figure 3.24). Similar performance was observed between the two networks, with a similar MUE of 0.9 kcal·mol^{-1} when including all replicates (figure 3.24C). This similarity is conserved when including fewer replicates, with little statistical difference between the two approaches.The same holds true for Pearson R and Kendall τ (figures 3.24A and B, resp.).

The dynamic representation of statistical performances across replicates for TNKS2 highlights the importance of assessing protocol repeatability.[264,265]. It appears that none of the statistical metrics have fully reached a plateau after 5 repeats, suggesting that the RBFE protocol could benefit from an even larger number of replicates or other optimisations. Although in this analysis no reference can be made to an optimal network chosen according to $|\Delta\Delta G_{offset}|$ values as in section 3.3.4, the RBFENN (n=28) and LOMAP-Score (n=27) networks can be directly compared. With an overlap of 27%, the overlap is considerably lower than with TYK2. The eight shared edges are predominantly single-atom perturbations. The main observed qualitative difference between the two networks is in how either handles the alkyl-OH motifs and the (de)halogenations: it appears that in general the LOMAP-Score network allocates more edges to (de)halogenations (*e.g.* **5k**→**5m**→**5i** figure 3.26), whereas the RBFENN network focuses more on allocating edges to perturbing the alkyl-OH motifs (*e.g.* **5o**→**5p**→**5i**, figure 3.25). This coincides with a recent observation by Cresset developers that the default simulation protocol for SOMD fared poorly for perturbation involving alkyl-OH motifs. This has been subsequently corrected by tuning softcore parameters. These new parameter settings have not been used for the generation of the current version of RBFE-Space which explains the behaviour of the data-driven approach in this analysis.

Although the main aim of the TNKS2 screen was to compare directly the performances between RBFENN and LOMAP-Score networks, SOMD performance on TNKS2 in this work is poor compared to results published elsewhere. For example, Schindler *et al.*[227] and Gapsys *et al.*[228] report MUE values of 0.62 and 0.73 kcal·mol^{-1}, respectively. Note that these values were computed using edges on neutral ligands only. Both of these examples contained considerably larger RBFE networks (n=45); Schindler *et al.* note that these were obtained by requesting an *optimal topology* from the FEP+ implementation and no manual network augmentation was performed. This suggests that increased performance could have been achieved using networks with a greater number of edges. Indeed, in-house results from Cresset suggest that Flare FEP (which deploys SOMD as its back-end RBFE engine) outperforms per-ligand binding affinity predictions of Schindler *et al.* with a manually adjusted network (n=70 edges), giving a MUE of 0.60 kcal·mol^{-1} and a pearson R value of 0.75 (tables S1 and S2).

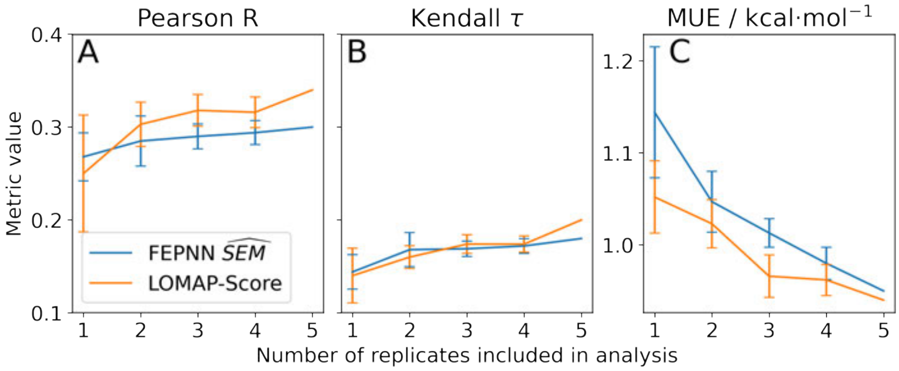


Figure 3.24: Statistical performances of RBFE predictions on the TNKS2 RBFE benchmarking series versus experimental ligand binding affinities using the data-driven approach described in this work (RBFENN; blue) versus the state-of-the-art LOMAP-Score approach (orange) for various statistical metrics. The data is presented as a dynamic representation of replicate inclusion, where for each progression of x all possible combinations of replicates (n=x) are included for the calculation of the mean metric value. Depicted error bars show the standard error of the mean metric across replicates; as for n=5 there is only one combination (all replicates), no confidence has been depicted.

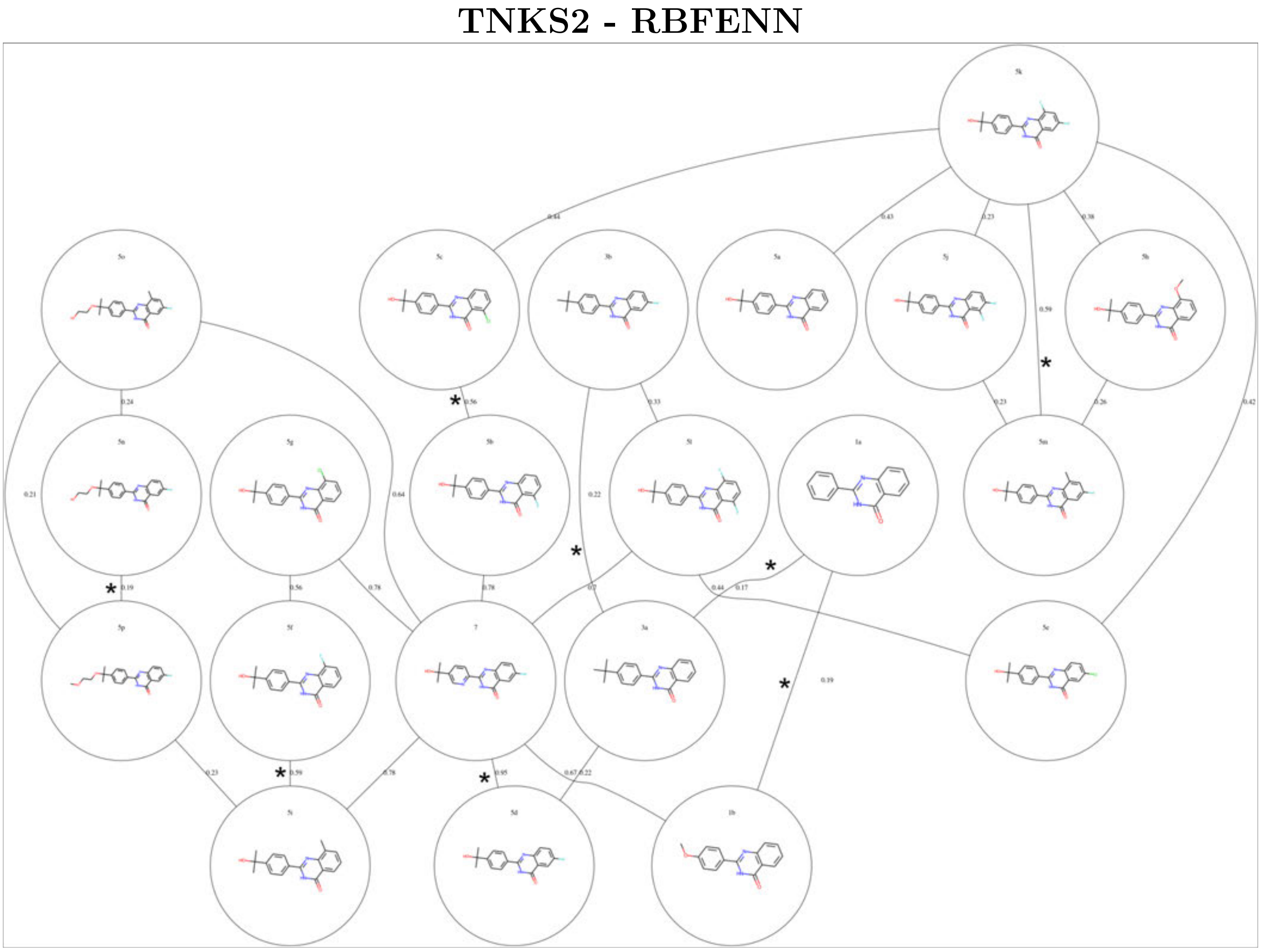


Figure 3.25: The TNKS2 perturbation network as suggested by LOMAP using the RBFENN-predicted $\widehat{SEM}$ score as input. Each node in the network contains the molecular structure and the ligand name; each edge in the network is annotated with the RBFENN-predicted $\widehat{SEM}$ value that has been scaled to [0-1] to allow proper handling by the LOMAP algorithm. Asterisks (*) indicate edges that are shared between the RBFENN and LOMAP networks. For this series, the six ligands with a +1 formal charge have been excluded.

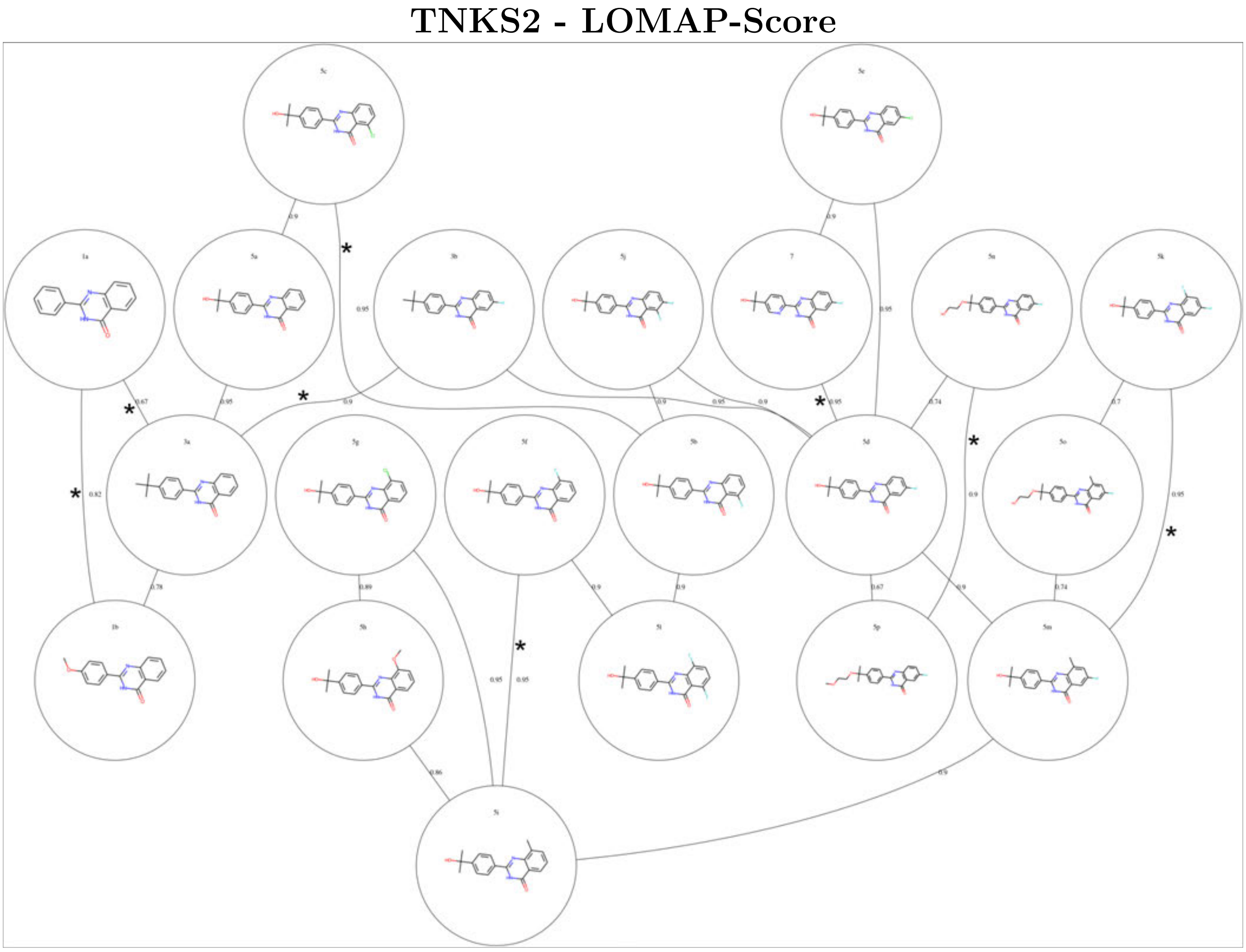

TNKS2 - LOMAP-Score

Figure 3.26: The TNKS2 perturbation network as suggested by LOMAP using the LOMAP-Score as input. Each node in the network contains the molecular structure and the ligand name; each edge in the network is annotated with the assigned LOMAP-Score value. Asterisks (*) indicate edges that are shared between the RBFENN and LOMAP networks. For this series, the six ligands with a +1 formal charge have been excluded.

Table 3.3: In-house results provided by Cresset on the neutral ligands of the TNKS2 RBFE benchmarking series. Shown are results of an RBFE run using a network with 70 edges run using Flare V4. Columns contain data on the experimental binding affinity, the experimental error, the RBFE-predicted binding affinity and the absolute error between experimental and predicted binding affinity for each ligand. Shown below the table are statistics as generated by Flare; Pearson R for this data is 0.75. See table 3.4 for edges and methodology.

Molecule	Experimental Activity	Error	Predicted Activity	abs(err)
1a	-8.55	0.3	-8.07	0.48
1b	-9.93	0.28	-10.04	0.11
3a	-10.99	0.22	-10.99	0
3b	-11.51	0.29	-10.83	0.68
5a	-10.76	0.23	-10.43	0.33
5b	-10.47	0.22	-11.11	0.64
5c	-9.95	0.28	-9.8	0.15
5d	-10.88	0.23	-10.3	0.58
5e	-10.1	0.46	-9.39	0.71
5f	-10.25	0.22	-11	0.75
5g	-10.8	0.3	-11.21	0.41
5h	-10.05	0.28	-9.57	0.48
5i	-12.07	0.31	-10.94	1.13
5j	-11.07	0.27	-11.53	0.46
5k	-10.96	0.28	-11.01	0.05
5l	-10.09	0.25	-11.47	1.38
5m	-12.68	0.33	-11.06	1.62
5n	-10.7	0.45	-10.54	0.16
5o	-12.03	0.69	-13.75	1.72
5p	-10.5	0.29	-11.02	0.52
7	-8.39	0.76	-8.65	0.26

Pearson r^2: 0.56 (95%CI 0.19-0.81)

MUE: 0.60 (95%CI 0.41-0.81) kcal·mol^{-1}

Table 3.4: perturbations run in-house by Cresset on TNKS2 (see table 3.3). Shown are relative binding free energy predictions for each edge in the chosen RBFE network (n = 70) in kcal·mol^{-1} for both the forward (A→B) and reverse (B→A) transformation. This RBFE campaign was run using Flare V4 with a total of 754 λ windows.

Edge	A→B	B→A
1a∼1b	-2.15	2.14
1a∼3a	-3.06	3.39
1b∼3a	-1.1	1.03
1b∼3b	-1.12	0.49
3a∼3b	0.24	-0.37
3a∼5a	0.53	-0.5
3a∼5b	-0.19	0.35
3a∼5f	-0.06	0.25
3b∼5d	0.45	-0.54
5a∼5b	-0.71	0.68
5a∼5d	0.07	-0.21
5a∼5f	-0.81	0.66
5b∼5c	1.3	-1.45
5b∼5j	-0.24	0.42
5b∼5l	-0.33	0.46
5c∼5l	-1.65	1.72
5d∼5e	0.88	-1.21
5d∼5j	-1.27	1.39
5d∼5k	-0.5	0.48
5d∼5m	-0.25	0.5
5d∼5n	0.54	1.95
5d∼5o	-8.06	6.21
5d∼5p	-1.02	1.19
5d∼7	1.26	-2.06
5e∼7	0.87	-0.08
5f∼5g	-0.27	0.34
5f∼5h	1.61	-1.84
5f∼5i	0.24	-0.04
5f∼5l	-0.51	0.82
5g∼5h	1.85	-2.08
5i∼5l	-0.74	0.9
5j∼5k	0.98	-0.58
5k∼5m	0.15	-0.08
5m∼5o	-7.17	4.29
5n∼5p	-0.76	0.96

3.4 Conclusions

In RBFE network generation there exist two main challenges: estimating the reliability of RBFE perturbations that form the edge of a network *a priori*, and optimising resources allocation to process a network that spans all compounds. The current work describes research into the first problem. Investigations into optimal network topology are actively being carried out.[231,234,235] Because the accuracy of an RBFE protocol is sensibly affected by the choice of the perturbation network this has important implications for the field. For instance, forcefield benchmarking studies with a given RBFE implementation should ideally be carried out with the same perturbation network. Benchmarking studies of different RBFE implementations should be made with networks tuned for performance for each implementation. This work introduces several new concepts to the field of RBFE. By grafting a large number of RBFE perturbations onto a common benzene scaffold, a transferable training set was created for RBFE research and development. As this set covers a diverse set of RBFE perturbations it is highly suitable for ML work and is set to drive research in combining RBFE and ML methodologies further. Using a siamese neural network architecture with graph representation of RBFE endpoint ligands, a statistical fluctuation predictor was trained on RBFE-Space. This SF predictor is shown to outperform state-of-the-art heuristics in the context of modelling SF. The prototype SF predictor (RBFENN) was used to generate the first ML-based networks for planning of RBFE calculations.

The prototype data-driven network generators are shown to match performance of state-of-the-art rule-based RBFE network generators that have required extensive calibration over multiple years to perform adequately with specific RBFE implementations. By contrast the data-driven method offers full transferability to other RBFE implementations with the single requirement of running a set of prescribed RBFE-Space simulations to recreate SF values specific to that implementation. Beside network generation, the RBFENN $\widehat{SEM}$ predictor presented in this work could be used to 'boostrap' adaptive sampling schemes for initial resources allocation.[235]

The availability of an inexpensive predictor of SFs could also be exploited by algorithms that sample chemical space to identify molecules whose RBFE reliability to a reference compound can be determined with ease.

As all heuristics depicted in table 3.2 attempt to model $|\Delta\Delta G_{offset}|$ values, this begs the question as to whether a predictor can be trained directly on this quantity instead of statistical fluctuations. For this, instead of grafting perturbations onto benzene (as with RBFE-Space), the original ligands must be featurised as well as the protein system in which the RBFE perturbation takes place. This has been attempted before and offers additional information such as pose differences between input ligands which are highly influential to the RBFE reliability.[190,236] However, the bottleneck in this scenario is that a large number of RBFE simulations must be run. Indeed, during early investigations of this work attempts were made to create a training set that included original ligands, but the chemical space associated with training such a model appeared too large with respect to the data available. For example, the PDBbind v2020 database[266] contains 19,443 protein-ligand complexes (with experimental binding affinities) across 5316 proteins. Assuming equal distribution of ligands per protein in this set brings the average size of congeneric series to $\frac{19,443}{5316} \approx 3.6$. Mapping all edges in each network results in $(3.6^2 - 3.6) \cdot 5316 = 49,758$ RBFE calculations, which is still a (very) conservative estimate as it is likely that some series will be larger than others: the number of possible edges in each series scales $O(n^2)$. Alternatively, a retrospective dataset could be generated gradually using previously completed RBFE calculations. This in turn presents several challenges because each point in the dataset will need be standardised as in general RBFE protocols evolve over time (thus affecting the accuracy of the results for the same perturbation) and even within RBFE campaigns different edges may be allocated different degrees of sampling (*e.g.* different numbers of λ windows).

Alternatively models could be trained on datasets built using $\Delta\Delta G_{bind}$ SEM values (figure 3.7D and table 3.2): this would at least remove the requirement of experimental binding free energies for each data point, opening up the possibility of manually curating the chemical space in order to construct a diverse dataset rather than being restricted to congeneric series that have experimental data. Additionally, an RBFE-Space version with original ligands (*i.e.* not grafted onto benzene) would enable faster SF predictions as this removes the need for additional MCS calculations to map ligands onto RBFE-Space abstractions. However, this method would still require simulations of the bound leg for each data point (to generate the training set) which could still be prohibitively expensive. Training on $\Delta G_{solvated}$ SEM values with original ligands is possible. However this space is still large due to chemical diversity of drug-like molecular scaffolds. We estimate that such dataset would require $\sim$2.5M perturbations.

Another possible future direction is to pursue an active learning approach where the RBFENN is re-trained using newly-obtained $|\Delta\Delta G_{offset}|$ values for edges while a congeneric series is being explored in a live drug discovery project.

Overall this work has demonstrated the importance of perturbation network planning for RBFE calculations, and the potential of machine learning to automate the generation of optimal RBFE networks. Continued efforts in this direction will increase the robustness and effectiveness of RBFE methodologies for drug discovery.

Data availability

All python code and jupyter notebooks used in this work are made publicly available under a GPL-2.0 license at https://github.com/michellab/RBFENN.

Chapter 4

Dissection of Concerted Alchemical Free Energy Calculations Into a Parallel Multi-Step Approach

4.1 Introduction

Relative binding free energy (RBFE) calculations are an invaluable tool in computationally supporting the ligand optimisation problem in early-stage drug discovery, both in hit-to-lead and lead-optimisation phases.[95] Although the technique's robustness has led to a wide variety of academic implementations[216,267–269] and a few commercial packages,[111,160,270] there remain technical limitations that prevent practitioners from freely exploring chemical spaces without restraints.

When transforming one ligand into another in RBFE, λ is defined as a *decoupling parameter* which is used to divide the transformation into a number of bins, where parameters are adjusted in a bin-wise manner, each containing increasingly perturbed parameters. In ideal situations (*i.e.* highly reliable RBFE transformations), the phase space overlap between λ states is high, which allows for effective statistical estimation of the change in free energy change between λ states; ultimately resulting in an effective relative free energy estimation between the two ligands (*i.e.* the λ endstates, $\lambda = 0.0$ and $\lambda = 1.0$). Popular statistical estimators are Thermodynamic Integration (TI[271,272]) and the Multistate Bennett Acceptance Ratio (MBAR).[122,273] Unfortunately there exist situations where it is required to perform perturbations that are unlikely to achieve acceptable phase space overlap such as large transformations (> 5 heavy atoms; *e.g.* when handling a chemically diverse congeneric series). Larger transformations are typically handled by improving sampling through increasing the used number of λ windows. This works because the charge, Van der Waals (VdW) and bond parameter transformations are split into smaller incremental changes between adjacent λ windows - the issue however is that this approach requires considerably more computing resources.

As perturbations in RBFE involve adjustments of multiple parameters (partial charges, VdW, bond parameters) there exist multiple approaches to deal with these during simulations. Arguably the most commonly deployed approach is to adjust all parameters in a single, concerted step where all parameters are adjusted across λ in unison. There exist some examples of implementations that use an alternative approach where not all parameters are adjusted in concert, but rather sequentially, and previous analyses investigating the differences between concerted and multistep approaches have been done.[274–278] Research comparing these approaches in bound-phase RBFE calculations is currently missing from the field. Additionally, work of this type using MBAR as the free energy estimator or with $n > 3$ steps has not been previously published to the best of our knowledge.

This work describes research done to investigate whether running Charge, VdW and bond parameter transformations individually (with variable λ allocations per step) offers an advantage to transforming all parameters in a single step, as is the current standard in the current RBFE implementation by Cresset in their Flare software package. Direct comparison of a concerted (single step) and multistep approach (figure 4.1) offers the opportunity for novel ways of exploring RBFE shortcomings. Although the initial rationale for this work was to investigate the effects of multi-step approaches on the reliability/accuracy of large (*i.e.* $>$ 5 heavy atoms) perturbations (figure 4.2), lessons learned during this project resulted in multiple unexpected victories due to virtues of the ability to compare one-step and multi-step approaches directly.

Initial results in this work qualitatively suggest that the bound leg benefits from a multistep protocol over a concerted protocol, whereas the free leg does not show benefit. Further work was performed by Cresset that showed no observable benefit of the multistep approach over the concerted approach. Several key findings are reported in this work that illustrate the benefits of dissecting an RBFE approach and comparing the two approaches side-by-side. Additional observations made during this research has led to optimised parameters in Flare V6.[214]

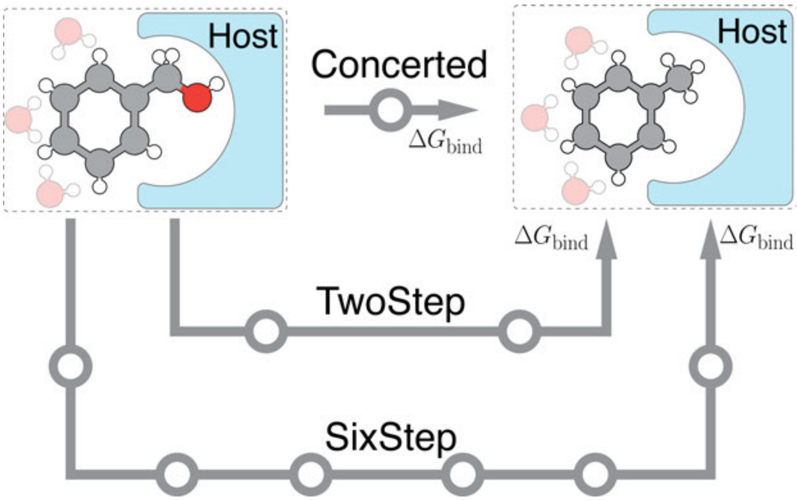
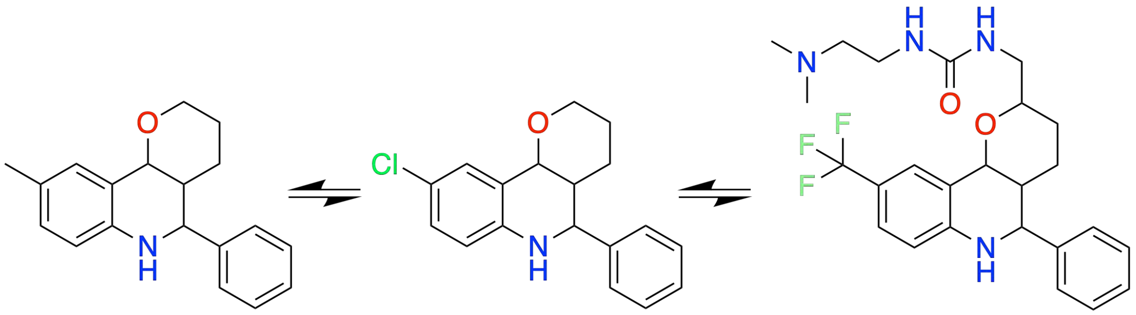


Figure 4.1: Concept schematic of the state-of-the-art concerted ($n_{steps} = 1$) approach and the alternative approaches ($n_{steps} > 1$) presented in this work. The example shows computation of the relative free energy of binding of toluene to benzyl alcohol. Each 'step' is represented as a gray circle.



CHEMBL1088740 CHEMBL1089393 CHEMBL1077204

Figure 4.2: Examples of two transformations to illustrate Flare V5 RBFE predictions that are highly reliable (left-hand side) and highly unreliable (right-hand side). `CHEMBL1088740`→`CHEMBL1089393` only perturbs one heavy atom, whereas `CHEMBL1089393`→`CHEMBL1077204` perturbs 13.

4.2 Theory & methods

All simulations performed in this work were done using SOMD as implemented in BioSimSpace version=2020.1.0=py37h1de35cc_97.[241] Although λ window allocation will vary per analysis, every simulation in any phase can be assumed to have been run for 4 ns unless specified otherwise. Used force fields were GAFF2, ff14SB and TIP3P for waters.

4.2.1 Terminologies used in this work

- *hard vs soft atoms*: a distinction that describes atoms part of the maximum common substructure (MCS; hard atoms) and atoms outside the MCS (soft atoms, *i.e.* atoms that are either being transformed into or from a dummy atom) in a molecular transformation with RBFE. Intuitively, this means that hard atoms are shared between the ligand endpoints whereas soft atoms only exist in one of the ligand endpoints.

- $n_{step} = N$ *approaches*: the approaches proposed in this work, where N denotes the total number of steps in the protocol.

- *MBAR*: Multistate Bennett Acceptance Ratio[122,124] is a modern statistical estimator of free energies that assesses from all states (for a concise description see *e.g.* `alchemistry.org/wiki/Multistate_Bennett_Acceptance_Ratio`)

- *overlap matrix*: a figure derived from MBAR that visually describes the degree of phase space overlap between states (*i.e.* λ windows) used by MBAR.[95]

- *vacuum/free/bound systems*: whether a perturbation was simulated with just the ligand (vacuum), the ligand in a 3 nm^3 cubic water box (free) or the ligand in a protein in a 10 nm^3 cubic water box (bound).

4.2.2 Creation of multistep approach

The multistep approach has been implemented using BioSimSpace on top of the existing Sire/OpenMM-MD (SOMD) code infrastructure.[216,241] The multistep approach was created as an extension to BioSimSpace and not SOMD. The multistep approach is composed of six steps:

1. **discharge_soft**: perturb Coulomb terms to 0.0 for soft atoms transforming to dummy.
2. **vanish_soft**: perturb LJ/Van der Waals terms to 0.0 for soft atoms transforming to dummy.
3. **change_hard**: perturb all Coulomb and LJ/VdW terms from lambda 0 to lambda 1 for hard atoms.
4. **change_bonds**: perturb all bond terms (angles, dihedrals).
5. **grow_soft**: perturb LJ/VdW terms to lambda 1 for soft atoms transforming from dummy.
6. **charge_soft**: perturb Coulomb terms to lambda 1 for soft atoms transforming from dummy.

Each step has starting parameters set to the final parameters of the previous step, *e.g.* in step 2, soft atoms are already discharged at lambda 0. In multistep approaches each step represents adjustment of a certain parameter category: for example in the case of a perturbation that involves transformation of a methyl into a hydrogen, 1) partial charges are adjusted to 0 across $\vec{\lambda}_1$, 2) VdW parameters are set to 0 across $\vec{\lambda}_2$, 3) partial charges are adjusted in the ligand scaffold (to account for the removed methyl group) across $\vec{\lambda}_3$. 4) bonded terms are updated. Steps 5) and 6) are not necessary for this particular case. For each $\vec{\lambda}_n$ the ΔG can be estimated using standard free energy estimation algorithms, and the overall ΔG is obtained through summation by

$$\Delta G_{multistep} = \sum_{n=1}^{n} \Delta G_{\vec{\lambda}_n}. \tag{4.1}$$

Because SOMD uses the Sire molecular framework to handle perturbable molecules, Sire molecular object functionality was directly extended. Sire perturbable molecules contain a method that writes out *pertfiles* that describe the global changes in parameters (*i.e.* charges, VdW, etc.). The Sire molecule object was adjusted such that the method `_toPertFile()` can take additional arguments referring to which step in the multistep approach it should write a *pertfile* as. For example, if `_toPertFile(pert_type="discharge_soft")`, a pseudocode example of how the *pertfile* atom terms will be written is as follows:

```
for atom in molecule do
    # retrieve native atom terms for lambda 0 and 1:
    LJ0 = atom.LJ0;
    LJ1 = atom.LJ1;
    charge0 = atom.charge0;
    charge1 = atom.charge1;
    # change charge1 for soft atoms, freeze all other terms to lambda 0:
    if atom == dummy then
        atom.LJ1 = LJ0;
        atom.charge0 = charge0;
        atom.charge1 = charge1;
    else
        atom.LJ1 = LJ0;
        atom.charge1 = charge0;
    end
    # now write updated atom terms to file:
    atom.LJ.write();
    atom.charges.write();
end
```

Algorithm 1: pseudocode showing the atom terms written to a *pertfile* when pert_type is set to `"discharge_soft"`.

An added benefit of writing adjusted *pertfiles* is that no low-level reprogramming in Sire will have to be done as Sire is able to read the contents of these files natively. The 2-step protocol described in this work consists of **discharge_soft** and the five remaining steps merged into a single step.

4.3 Results & discussion

As this work represents a sequential investigation into multistep approaches, we will highlight several trial protocols. The following subsections will discuss results of an initial simple protocol ($n_{steps} = 2$, 4.3.1) after which the final protocol will be described ($n_{steps} = 5$, 4.3.2). In 4.3.2, instead of the $n_{steps} = 5$ notation, the protocol will be referred to as 'multistep'.

4.3.1 $n_{steps} = 2$ approach

The initial idea of splitting the standard protocol into two steps (first discharge soft atoms, then shrink soft atoms) was tested on a range of eg5 inhibitors as available in the Merck benchmarking set.[125] A selection of large (*i.e.* >10 heavy atoms) perturbations was made and simulations were carried out in both vacuum and solvated systems for 4 ns per λ window. For the majority of RBFEs, the MBAR overlap matrices for the $n_{steps} = 1$ protocol showed poor overlap, whereas the $n_{steps} = 2$ protocol showed improvements in phase space overlap in cases where the transformations consisted of removing (*i.e. shrinking*) of functional groups (see figure 4.3). Of these transformations, the forward (*i.e. growing*) perturbation showed poor overlap. This suggests that when *growing* an atom from dummy, water molecules must first be displaced by increasing VdW terms, and then Coulomb terms should be set (*i.e. charged*). Conversely, when *shrinking* an atom to dummy, Coulomb terms should first be turned off (*i.e. discharged*), after which Van der Waals terms can be turned off - at this point water molecules will be able to take the moiety's place. This aligns with approaches presented in other work.[274,277]

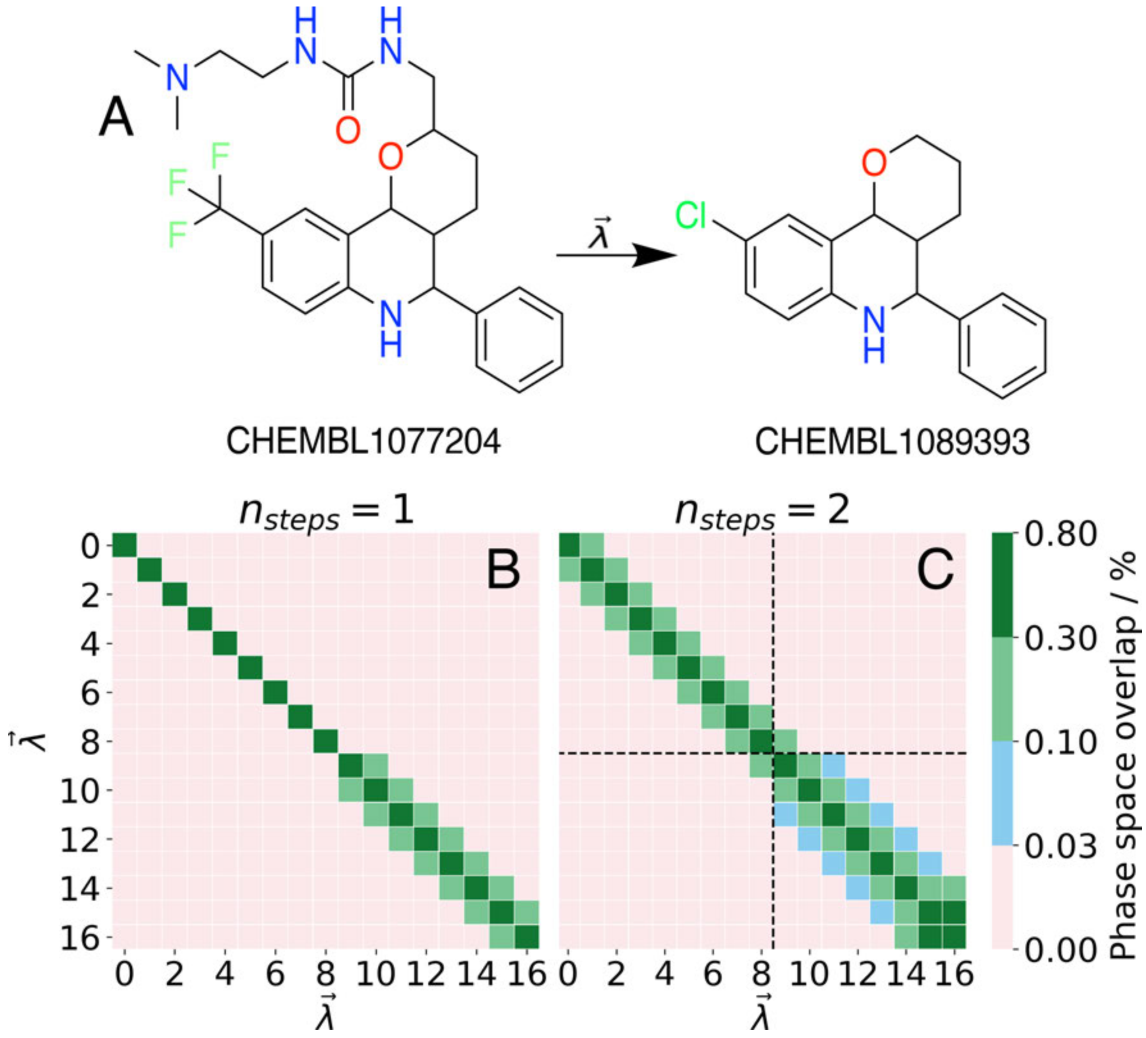


Figure 4.3: Free energy estimations in the free phase suggest increased phase-space overlap of the $n_{steps} = 2$ protocol over the $n_{steps} = 1$ protocol. **A**: a large perturbation (13 perturbed heavy atoms) is simulated in the free phase across λ with $n_\lambda = 17$. **B**: overlap matrix where each block is a discretised colour coding of phase space overlap between the two λ windows in question (in %, see colourbar) for $n_{steps} = 1$. **C**: identical to B, but for the $n_{steps} = 2$ protocol. Dashed black lines indicate the ninth λ window at which λ stops involving partial charge adjustments and starts involving VdW adjustments. A phase space overlap of 0.03 or higher (all colours other than salmon) is generally regarded as the threshold for sufficient overlap.[95]

4.3.2 Multistep approach

The idea of a bidirectional transformation suggests that instead of a $n_{steps} = 2$ approach, a $n_{steps} = 4$ approach should be taken in which a perturbation involves a discharge, shrink, grow and charge. The first two steps of this approach would involve atoms shrinking to dummies, and the final two steps would involve the growing of dummies into atoms. However, even the $n_{steps} = 4$ approach is incomplete, as there is added complexity that originates from partial charges and Van der Waals terms in hard atoms which are influenced by nearby atoms - including soft atoms. Additionally, when switching soft atoms to dummy and vice versa, bond parameters (angles, dihedrals) change for the whole molecule (*i.e.* all soft and hard atoms combined). For a complete approach, these two steps should also be performed separately in between the steps that take care of the soft atoms. The next sections will contain results of our implementation of this *multistep* approach that contains six steps in which we discharge, shrink, change hard atoms, change bond terms, grow and charge (figure 4.4). In this design, for perturbations that involve only shrinking or growing of atoms, steps 1/2 or 5/6 (resp.) can of course be omitted because no parameters should change during these steps.

Although a protocol that involves all possible steps in a perturbation separately would be $n_{steps} = 6$ (figure 4.4), from hereon the used protocol is '$n_{steps} = 5$', which means that steps 3 and 4 were merged into one (the merged step is called '3_flip'). This choice was made based on the observation that during testing these two steps showed far larger degrees of phase-space overlap compared to steps 1/2 and 4/5, combined with the fact that it would save computing time during testing. Additionally, only perturbations which contain uniquely growing or shrinking atoms are used, during which the redundant steps (1/2 when growing, 4/5 when shrinking) are removed before simulation. This results in three steps per perturbation for the multistep protocol, but the protocol will still be denoted as $n_{steps} = 5$.

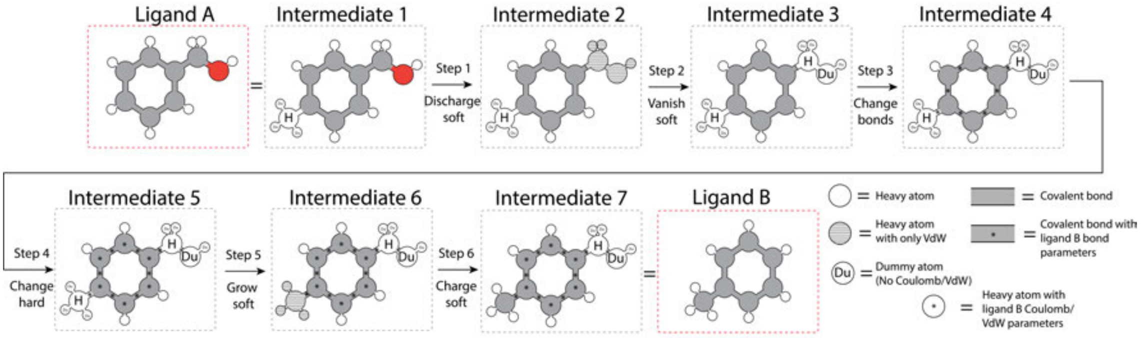


Figure 4.4: Workflow schematic of multistep approach in perturbing from ligand A to ligand B. Shown is a pedagogical transformation where at the top-right position a methanol moiety is removed, and at the bottom-left position a methyl moiety is grown. See section 4.2.2 for a detailed per-step description. Red boxes show ligand endpoints and gray boxes show the seven intermediates (of which intermediate 1 is equal to ligand A and intermediate 7 is equal to ligand B). For simplification, hydrogens are ignored in the depiction but they can be assumed to be perturbed in the same way as the heavy atom they are bound to.

Validation of the multistep approach in ethane↔methanol

A validation analysis was done using an ethane↔methanol perturbation with ample sampling (17 λ windows with 4ns per window) to test whether the multistep protocol was correctly designed. This system was chosen because of its simplicity and previously published benchmarking work in a similar context.

This analysis provides two key observations that validate the multistep approach. Firstly, the multistep approach (as well as the concerted approach) predictions (±6.35 kcal·mol^{-1}) show a high degree of agreement with previously reported relative free energy of hydration predictions (±6.23, ±6.22, ±5.99, ±6.26 kcal·mol^{-1} for AMBER, CHARMM, GROMACS and SOMD, respectively).[224] In the case of SOMD, it is expected that the $\sim$ 0.09 kcal·mol^{-1} is due to incremental updates in the SOMD codebase as the referred study was conducted five years prior to the work performed in the current study.

Secondly, the hydration free energy estimations show no hysteresis when comparing ethane→methanol and methanol→ethane; the multistep and concerted estimations of relative hydration free energies show a high degree of agreement (at an error of ~0.01 kcal·mol^{-1}) which suggests that the steps in the multistep approach completely encompass the parameter adjustments in the concerted approach. Finally, steps where there were no parameter adjustments across λ for perturbations showed a relative hydration free energy of 0.00 kcal·mol^{-1} which confirms that no artefact effects are occurring during these simulations (*e.g.*, ethane→methanol *4_grow_soft*= 0.00 kcal·mol^{-1} in both vacuum and solvated phases).

System	Step	Ethane→Methanol	Methanol→Ethane
ΔG_{vacuum}	1_discharge_soft	-0.61	0.00
	2_vanish_soft	-0.08	0.00
	3_flip	3.21	-3.21
	4_grow_soft	0.00	0.08
	5_charge_soft	0.00	0.61
	multistep_sum	2.51	-2.51
	concerted	2.51	-2.51
$\Delta G_{solvated}$	1_discharge_soft	-1.36	0.00
	2_vanish_soft	-0.08	0.00
	3_flip	-2.39	2.41
	4_grow_soft	0.00	0.08
	5_charge_soft	0.00	1.36
	multistep_sum	-3.83	3.84
	concerted	-3.84	3.86
$\Delta\Delta G_{hydration}$	multistep	**-6.34**	**6.35**
	concerted	**-6.35**	**6.37**

Table 4.1: Validation study on a simple ethane↔methanol perturbation with ample sampling (17 λ windows with 4ns per window) in both vacuum and solvated (solvated in a 3 nm^3 cubic waterbox with TIP3P waters) phases. Shown are free energies in kcal·mol^{-1} estimated by MBAR. Key values of interest are indicated in bold. The reported $\Delta\Delta G_{hydration}$ for this transformation in the validation study is $\sim \pm 6.2$ kcal·mol^{-1}.

4.3.3 Benchmarking the multistep approach in solvated ligand systems

Given that the multistep protocol showed full reproduction of the thermodynamic cycle of the concerted approach in small-system RBFE calculations, larger ($>$ 5 heavy atoms) perturbations were tested to explore whether the two approaches showed differences in the ability to handle these types of perturbations. For this purpose, the Eg5 system from the recent Merck RBFE benchmarking set[279] was chosen because it contained two distinct clusters of ligands that were distinguished by a long, highly rotatable R-group or the absence thereof (see structures in figure 4.3, e.g.), which makes this series contain many large transformations. All ligands were neutralised before simulation.

First, an analysis was done to determine the number of λ windows needed per step in the multistep protocol. To this end, several large perturbations were run with 30 λ windows per step and 90 λ windows for the concerted protocol. Then, equidistant subsets of λ windows were selected to estimate relative free energies using MBAR. This method allows analysing a number of λ arrays of varying size while only doing simulations for the largest λ array. For clarity, an example of this approach for a RBFE transformation run with 9 lambda windows (indices [1, 2..9]) would be obtaining a 5-window ([1, 3 .. 7, 9] and a 3-window ([1, 5, 9]) subselection. Because equidistance is chosen to be retained only a limited number of subselections is possible. Thus, perturbations were run several times at different total numbers of λ windows (*e.g.* 22, 25 and 33) to produce a wide range of λ arrays.

It was observed in an initial run with large transformations in the free phase that there was no clear benefit between using a multistep and a concerted protocol; ΔG convergence was reached at ∼15 λ windows for both the concerted and the multistep protocols (figure 4.5). Although qualitative, these results do suggest that the '3_flip' step is likely to require fewer λ windows than the other two steps as it is observed to be converging rapidly after only ∼ 5 λ windows.

Additionally, a similar transformation involving a phenyl group showed convergence to different values of ΔG between the multistep and concerted protocols, which prompted a run in triplicate of this particular transformation (CHEMBL1089393∼CHEMBL1084431) to check whether anything other than variance caused this discrepancy (see figure 4.6). As can be observed, variance is low across replicates for all protocols - further investigation showed that CHEMBL1084431 was set up improperly (not neutralised), where a Cl^- counterion was present in the simulations (this is a standard feature of BioSimSpace which automatically introduces counterions to systems in cases of non-neutral net charges). When investigating the interaction energy of this anion in these simulations it appeared to interact with the perturbed ligand differently between each protocol (*i.e.*, a ligand with only charges or VdW terms interacts differently with an anion than a ligand with both charges and VdW, non-additively). In further screens no charged perturbations were allowed during setup.

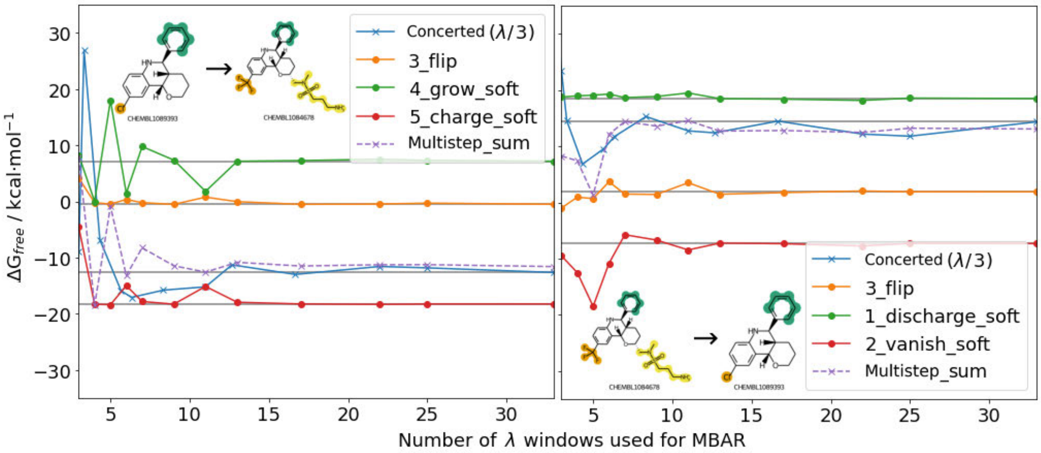


Figure 4.5: Plots of ΔG_{free} across varying sizes of λ for the concerted and multistep protocols in kcal·mol^{-1}. Shown are both directions of a perturbation in the eg5 RBFE benchmarking series: the 'grow' perturbation (CHEMBL1086789→CHEMBL1089393, left-hand side) and the 'shrink' perturbation (CHEMBL1089393→CHEMBL1086789, right-hand side). The total multistep ΔG_{free} values are shown as a purple, dashed, starred line and the concerted approach ΔG_{free} values are shown as blue starred line. Grey horizontal lines indicate the value of ΔG for each step at the largest number of λ windows used. Note that the concerted λ values were divided by three to standardise them with the multistep range.

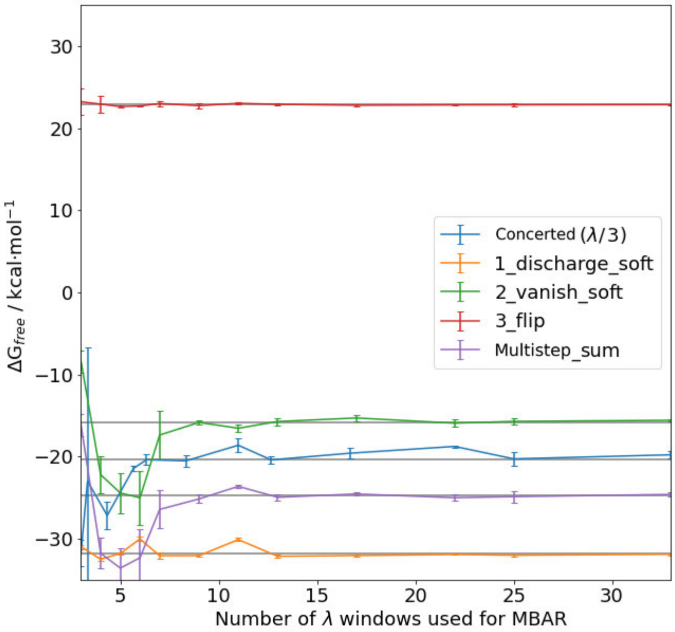


Figure 4.6: Plot of ΔG_{free} across varying sizes of λ for the concerted and multistep protocols in kcal·mol^{-1} for a 'grow' perturbation (CHEMBL1089393→CHEMBL1084431) in the eg5 RBFE benchmarking series. The total multistep ΔG_{free} values are shown as a purple line and the concerted approach ΔG values are shown as blue starred lines. Grey horizontal lines indicate the value of ΔG for each step at the largest number of λ windows used. Note that the concerted λ values were divided by three to standardise them with the multistep range. Shown uncertainties are standard deviations of the mean of three replicates.

4.3.4 Benchmarking the multistep approach in a protein-ligand system

In an attempt to explore whether the multistep approach showed any benefit in more complex systems, the approach presented in 4.3.3 was repeated while including the eg5 protein (figure 4.7). Because of the large increase in system complexity and thereby required computing resources, only a single perturbation has been analysed in this manner (run for both 25 and 33 λ windows).

For this experiment, the multistep protocol sum ΔG_{bound} value converges to ~93 kcal·mol^{-1} between 10 and 15 λ windows. Looking further into the individual steps, it can be observed that steps *1_discharge_soft* and *3_flip* converge early at ~3 λ windows, whereas step 2 converges somewhat later at ~7 λ windows - this observation is consistent with results in figure 4.5. Early convergence of these steps suggests that it could be possible to develop a variable allocation protocol where each step in the multistep protocol is run at different lengths of λ arrays. This could potentially result in decreased computational expense when dealing with perturbations such as the one depicted in figure 4.7; in this example a total of 13 (3+7+3) λ windows for the multistep would have obtained the final ΔG_{bound} prediction for the multistep approach, whereas the concerted approach requires 21 (7·3) λ windows to reach its final ΔG_{bound} prediction.

Additionally, for large λ arrays in the concerted protocol MBAR failed to estimate a free energy value. It appears that for these large λ arrays (lengths=[38, 50, 75, 99], x value in figure 4.7=[12.7, 16.7, 25, 33]), PyMBAR falls back on an alternative solver ('BFGS') because the default - more robust - adaptive solver is unable to handle the data volume. This fallback might explain the upward drift behaviour of the concerted ΔG_{bound} value for these points in figure 4.7 (final ΔG_{bound} values for the sum of the multistep and the concerted protocols are 100 and 93 kcal·mol^{-1}, resp.). Indeed, free energy predictions using thermodynamic integration (TI) for these large λ arrays for the concerted protocol resulted in ΔG_{bound} predictions of 93 kcal·mol^{-1} for these same arrays, which suggests that the TI free energy estimation approach might be more robust to larger (length>38) λ arrays.

Because binding pockets can have charged residues interacting with the ligand during simulations, it is possible that the same discrepant energetics that were observed in 4.3.3 could arise in this context even though the ligand has a net charge of 0. The perturbation discussed in this section (CHEMBL1085859→CHEMBL1089393) does not contain a perturbed phenyl R-group but care should be taken in future work as this effect could be introduced even in the absence of counterions when the ligand is placed in a protein binding pocket during simulations.

A more quantitative analysis of the multistep approach in protein-ligand systems is required. As mentioned, the analysis presented in 4.3.4 is too computationally expensive to repeat sufficiently to arrive at statistically quantifiable data. An alternative analysis is presented in the next section.

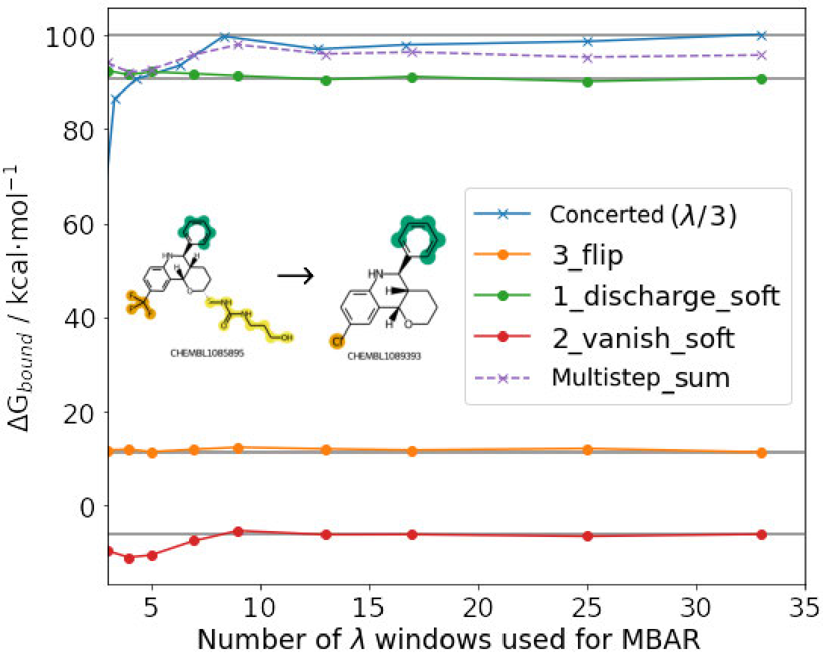


Figure 4.7: Plots of number of lambdas required to reach ΔG (in kcal·mol^{-1}) convergence for a 'shrink' perturbation (CHEMBL1085859→CHEMBL1089393). The total $n_{steps} = 3$ ΔG values are shown as a purple, dashed, starred line and the concerted approach ΔG values are shown as blue starred line. Grey horizontal lines indicate the value of ΔG for each step at the largest number of λ windows used. Note that the concerted lambda values were divided by three to standardise them with the multistep range.

4.3.5 The multistep approach applied to a diverse collection of challenging perturbations

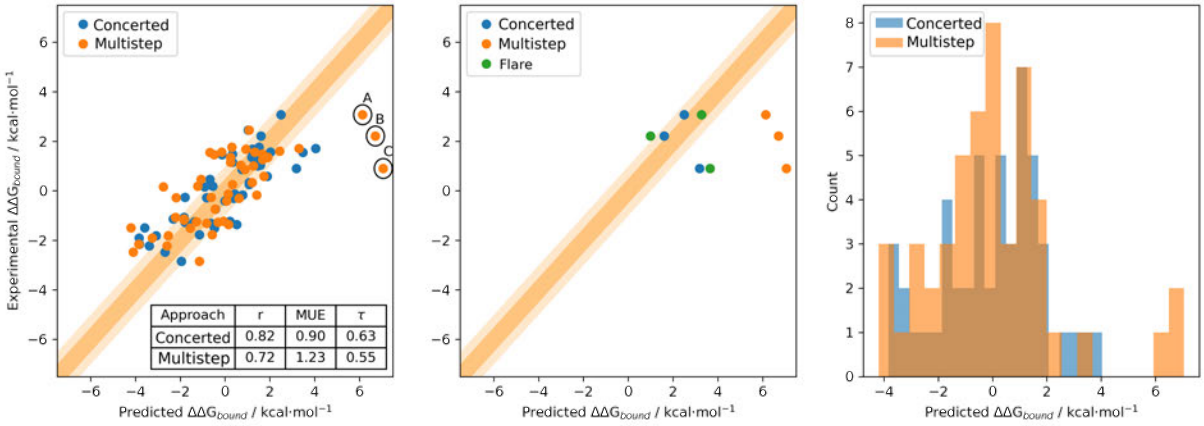
The multistep approach was further investigated by H Loeffler and M Mackey at Cresset. The final section of this chapter consists of analyses on the source data by H/M. The author thanks H/M for providing this source data and for their contributions to discussions on the analyses outlined below. Instead of analysing ΔG convergence as presented in sections 4.3.3 and 4.3.4, this analysis uses predictivity versus experimental measures as well as hysteresis which is defined as the thermodynamic cycle closure error in kcal·mol^{-1} obtained by subtracting the $\Delta\Delta G_{\mathrm{bind}}$ of both directions of a perturbation.

The concerted approach outperforms the multistep approach in ΔΔG$_{bind}$ predictions

A collection of perturbations (n=50) was selected and run in triplicate using both the concerted and multistep approaches. This set was composed of transformations present in publicly-available RBFE benchmarking series[226] that were deemed challenging based on prior experience; targets involved were CDK2 (n=16), CDK8 (n=6), EG5 (n=8), P38 (n=8) and PTP1B (n=12).

The concerted protocol outperformed the multistep protocol in terms of predicting experimental ΔΔG$_{bind}$ values for correlation (Pearson r 0.82 over 0.72), error (MUE 0.9 over 1.23) and ranking (Kendall τ 0.63 over 0.55) metrics (figure 4.8, left). For the multistep approach there exist three outliers which are sulfonamide-growing perturbations (figure 4.9). Whereas in older versions of the concerted approach this particular topology was a known issue, in more recent versions of SOMD this issue has largely been resolved by setting softcore parameters `deltashift=1.0` and `coulombpower=0` ; this is also apparent from the concerted approach behaving normally for these outliers (figure 4.8, center). The fact that these three perturbations were outliers even in terms of ΔΔG$_{bind}$ values with respect to the other 47 perturbations in the set (figure 4.8, right) suggests that the multistep approach may not be fully optimised to deal with this kind of molecular transformation. Sulfonamide parameterisation is a known bottle-neck in GAFF2 which is used in this particular case; trials with alternative force fields such as OFF[83] may negate this issue. Removal of the three outliers from the concerted approach does not dramatically improve statistical performance: the only notable difference is a reduction in MUE from 1.23 to 0.98 kcal·mol^{-1}.

Finally, no notable difference was observed in either approach when investigating *shrink* and *grow* perturbations (figure 4.10). Both approaches correctly capture the effect of *shrink* and *grow* perturbations typically decreasing and increasing (resp.) ligand binding affinity.



Approach	r	MUE	τ
Concerted	0.82	0.90	0.63
Multistep	0.72	1.23	0.55

Figure 4.8: Analysis of multistep versus concerted approach ΔΔG$_{bind}$ predictions versus experimental values in kcal·mol^{-1}. **Left**: RBFE-predicted versus experimental relative binding free energy for the concerted and multistep approach (blue; orange resp.). The 1/2 kcal·mol^{-1} confidence region is shown as a light/dark orange band. The table shows statistical analysis on both approaches in Pearson r, Mean Unsigned Error (MUE / kcal·mol^{-1}) and Kendall τ. The three outliers (A/B/C) discussed in the main text body are outlined with red circles (see figure 4.9 for molecular structures). **Center**: Relative binding free energy predictions for outliers A/B/C (see left-hand side scatterplot) for concerted, multistep and 'Flare' (default Flare settings) approaches in blue, orange and green, resp. **Right**: Histograms of predicted relative binding free energies for both approaches.

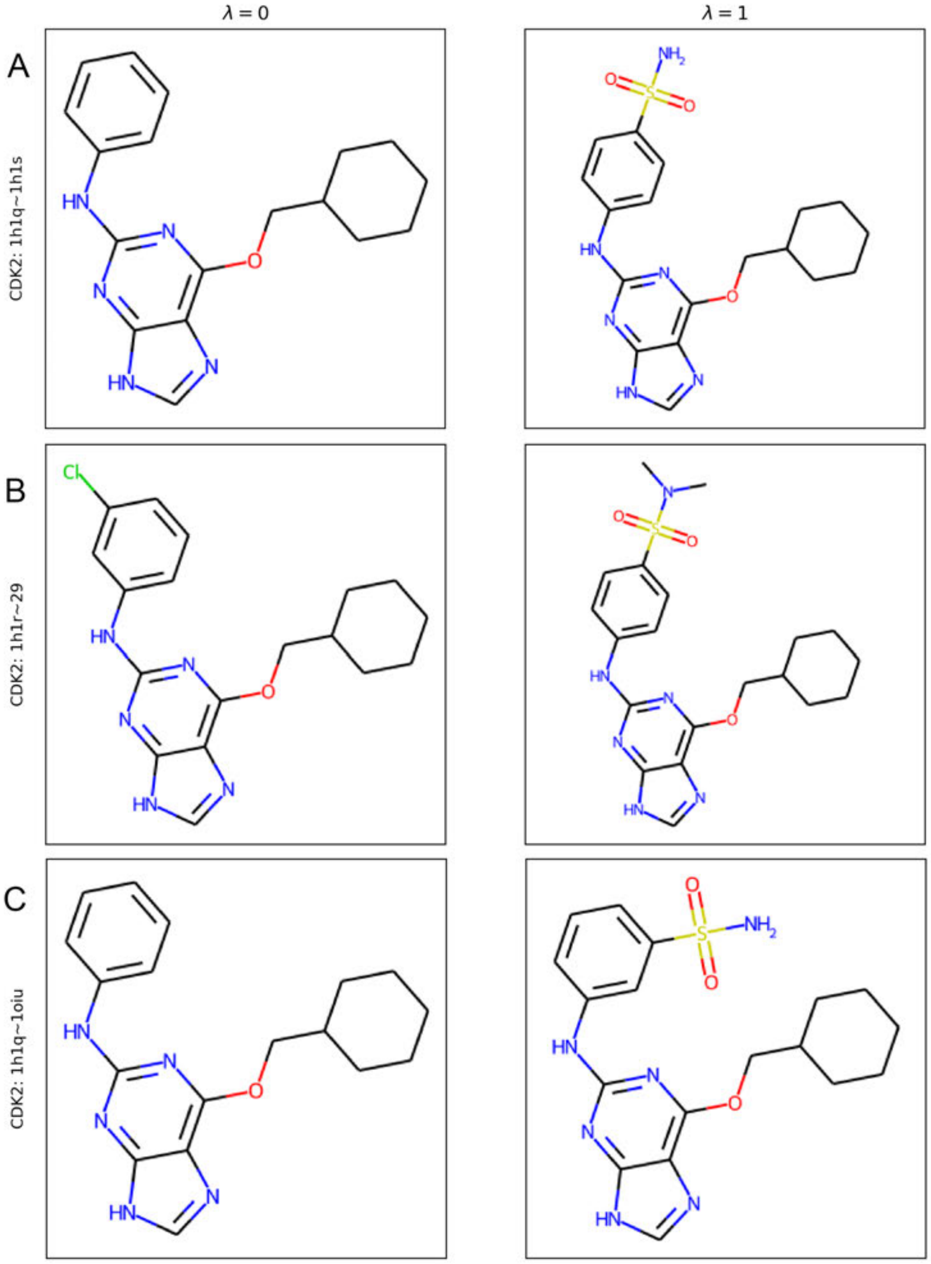


Figure 4.9: Molecular structures of λ enpoints of the three outliers (**A**/**B**/**C**) as reported in 4.8. For each perturbation, the protein target and perturbation name ("ligand A"$\sim$"ligand B") is denoted vertically on the left-hand side.

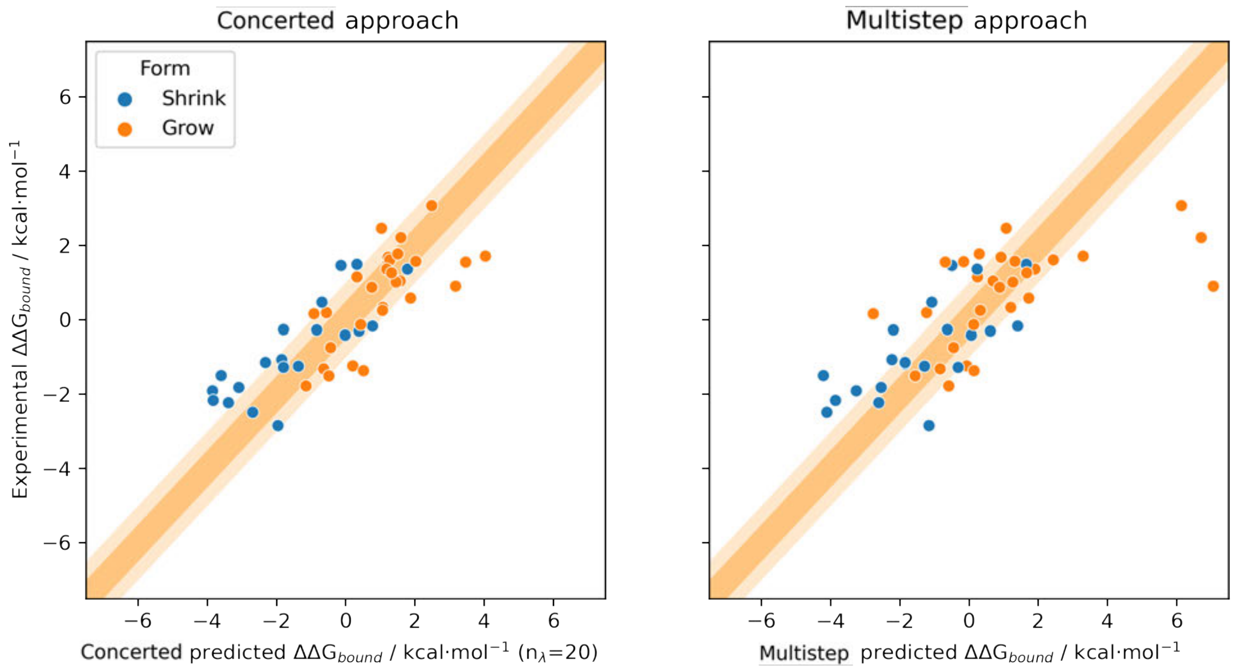


Figure 4.10: Depiction of relative binding free energy predictions in kcal·mol^{-1} for the concerted approach (**left**) and multistep approach (**right**). Shown in blue and orange are shrink (*i.e.* the perturbation involves removal of heavy atoms) and grow (*i.e.* the perturbation involves addition of heavy atoms) transformations, resp.

The concerted and multistep approaches vary in $\Delta\Delta G_{bind}$ prediction hysteresis

As a secondary analysis, the hysteresis for each perturbation per approach was investigated. As the hysteresis is computed as the error in thermodynamic cycle closure it is a measure of RBFE reliability, and a superior RBFE approach would show reduced overall hysteresis (*i.e.* increased reliability). Hysteresis is a commonly used marker for RBFE inaccuracies.[280] Comparing the 50 perturbations present in the set, the distributions of hysteresis for the concerted and multistep were similar (figure 4.11, left), with the concerted approach having a slightly higher incidence of perturbations with very low hysteresis (< 0.25 kcal·mol^{-1}).

It appears that for some perturbations the concerted approach results in higher hysteresis, whereas for the other perturbations the multistep approach results in higher hysteresis. Notably, of all perturbations, the perturbation with the highest hysteresis is a concerted perturbation (CDK2, 32 33, a cyclopropyl addition). In fact, four of the five highest hysteresis perturbations in the concerted approach have some form of cycle growing (see figure 4.12). This particular type of perturbation is a well-known bottleneck in single topology style RBFE software.[95] Each of these four perturbations show a dramatic reduction in hysteresis in the multistep counterpart, suggesting a more gradual (*i.e.* sequential) growing of these ring topologies can be advantageous in a single topology RBFE approach, likely due to a more gentle displacement of solvent molecules. This observation warrants further research to more rigorously investigate cycle RBFEs between the concerted and multistep approaches as the current work does not have enough (n=7) perturbations of this type to allow meaningful statistical analyses.

The top five highest hysteresis perturbations for the multistep approach consisted of large perturbations (figure 4.13). For four of those perturbations, the concerted approach exhibited low hysteresis (¡ 0.65 kcal·mol^{-1}).

Finally, to investigate the RBFE reliability related to the number of heavy atoms perturbed in the transformation (bidirectionally), both approaches were compared (figure 4.14). These results show that even for larger perturbations the multistep approach offers no benefit over the concerted approach, and for some sizes (*e.g.* [1-2] and [5-7]) even increases hysteresis.

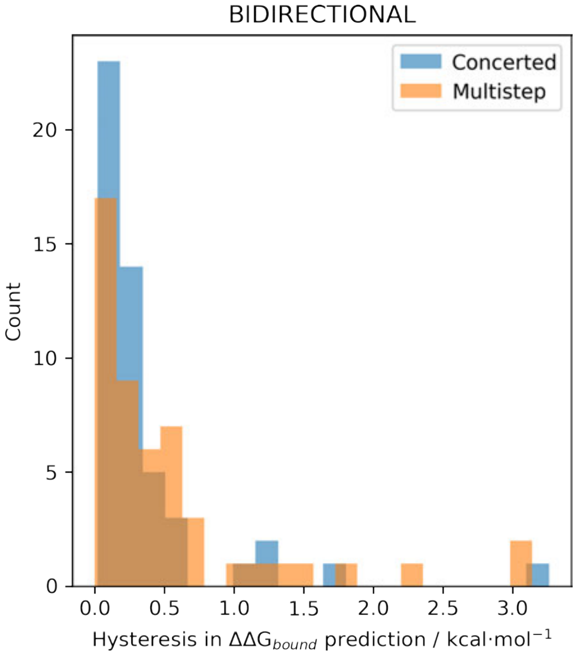


Figure 4.11: Histograms of hysteresis values of $\Delta\Delta G_{bind}$ predictions in kcal·mol^{-1} for both the concerted (blue) and the multistep (orange) approach.

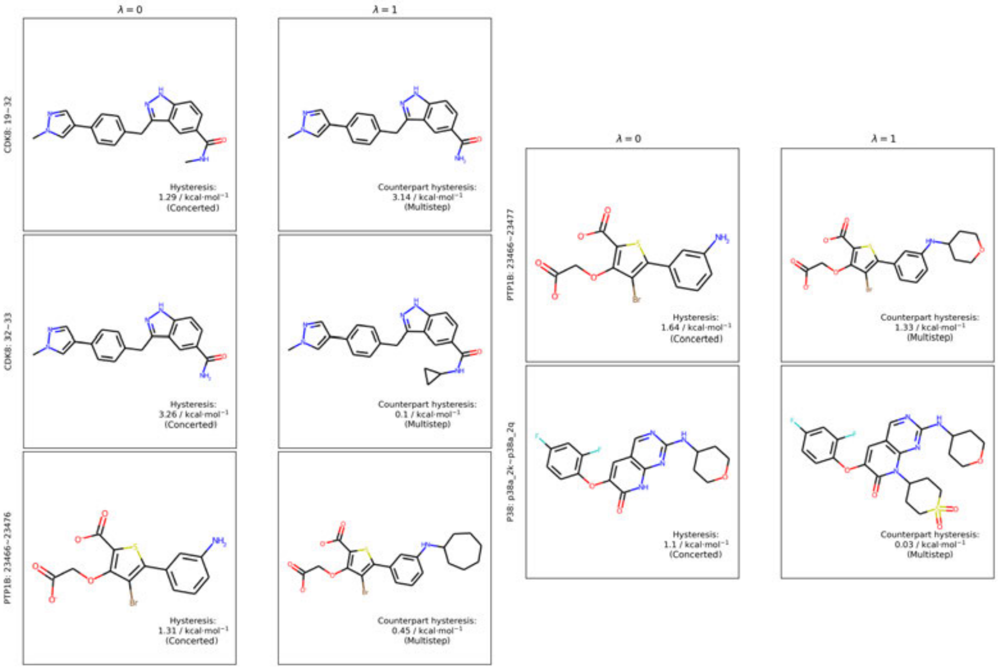

$\lambda = 0$
$\lambda = 1$
CDK8: 19~32
Hysteresis: 1.29 / kcal·mol^{-1} (Concerted)
Counterpart hysteresis: 3.14 / kcal·mol^{-1} (Multistep)
CDK8: 32~33
Hysteresis: 3.26 / kcal·mol^{-1} (Concerted)
Counterpart hysteresis: 0.1 / kcal·mol^{-1} (Multistep)
PTP1B: 23466~23476
Hysteresis: 1.31 / kcal·mol^{-1} (Concerted)
Counterpart hysteresis: 0.45 / kcal·mol^{-1} (Multistep)
$\lambda = 0$
$\lambda = 1$
PTP1B: 23466~23477
Hysteresis: 1.64 / kcal·mol^{-1} (Concerted)
Counterpart hysteresis: 1.33 / kcal·mol^{-1} (Multistep)
P38: p38a_2k~p38a_2q
Hysteresis: 1.1 / kcal·mol^{-1} (Concerted)
Counterpart hysteresis: 0.03 / kcal·mol^{-1} (Multistep)

Figure 4.12: Molecular structures of the five most hysteric perturbations in the **concerted** approach. Shown are the $\lambda = 0$ and 1 endpoints, with the degree of hysteresis (in kcal·mol^{-1}) of the concerted approach shown in the $\lambda = 0$ panel; the corresponding multistep hysteresis is shown in the $\lambda = 1$ panel. Each perturbation's target and perturbation name (”ligand A”$\sim$”ligand B”) is shown beside the $\lambda 0$ panel.

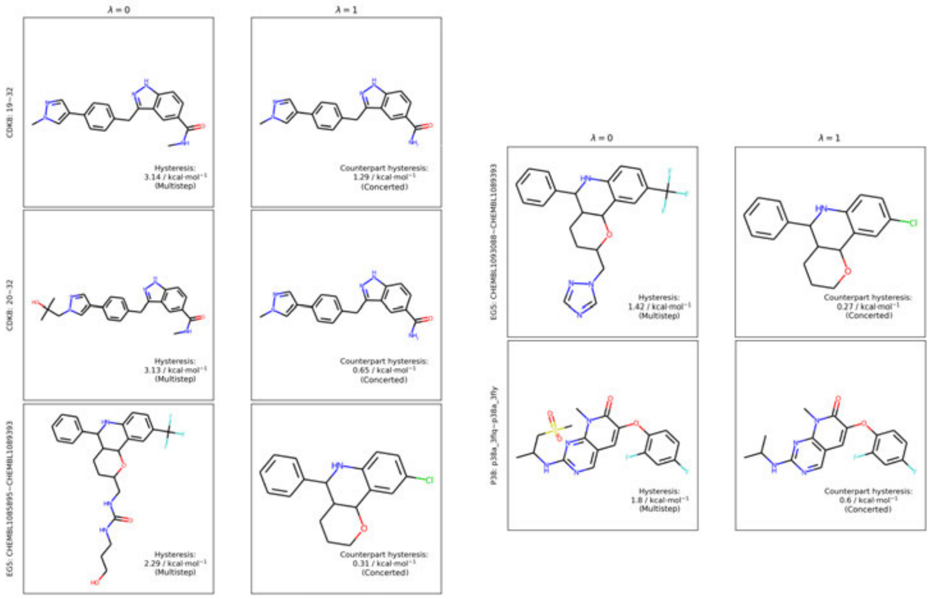

λ = 0
λ = 1
CDK8: 19~32
Hysteresis: 3.14 / kcal·mol⁻¹ (Multistep)
Counterpart hysteresis: 1.29 / kcal·mol⁻¹ (Concerted)
CDK8: 20~32
Hysteresis: 3.13 / kcal·mol⁻¹ (Multistep)
Counterpart hysteresis: 0.65 / kcal·mol⁻¹ (Concerted)
EG5: CHEMBL1085895~CHEMBL1089393
Hysteresis: 2.29 / kcal·mol⁻¹ (Multistep)
Counterpart hysteresis: 0.31 / kcal·mol⁻¹ (Concerted)
λ = 0
λ = 1
EG5: CHEMBL1093088~CHEMBL1089393
Hysteresis: 1.42 / kcal·mol⁻¹ (Multistep)
Counterpart hysteresis: 0.27 / kcal·mol⁻¹ (Concerted)
P38: p38a_3flq~p38a_3fly
Hysteresis: 1.8 / kcal·mol⁻¹ (Multistep)
Counterpart hysteresis: 0.6 / kcal·mol⁻¹ (Concerted)

Figure 4.13: Molecular structures of the five most hysteric perturbations in the **multistep** approach. Shown are the $\lambda = 0$ and 1 endpoints, with the degree of hysteresis (in kcal·mol^{-1}) of the multistep approach shown in the $\lambda = 0$ panel; the corresponding concerted hysteresis is shown in the $\lambda = 1$ panel. Each perturbation's target and perturbation name (”ligand A”∼”ligand B”) is shown beside the λ0 panel.

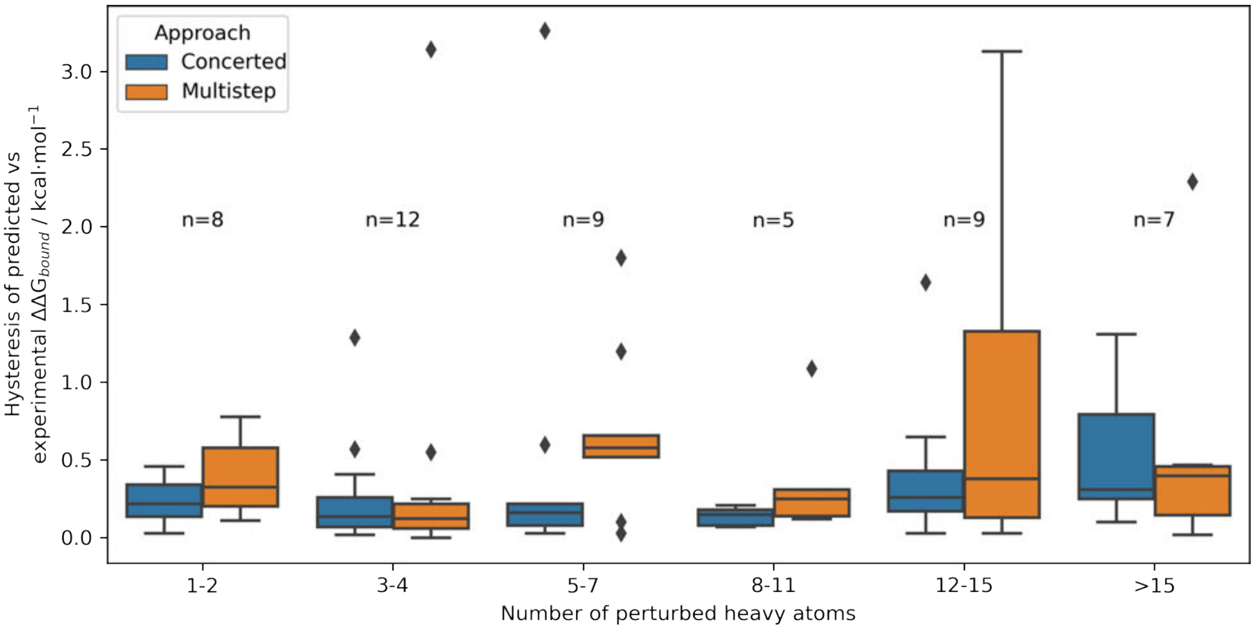


Figure 4.14: Boxplots of hysteresis values of $\Delta\Delta G_{bind}$ predictions versus experimental in kcal·mol^{-1} for both concerted (blue) and multistep (orange) approaches for a range of different numbers of heavy atoms perturbed in each perturbation. For each group of boxes, the population size (*i.e.* number of perturbations) is annotated on the plot. Outliers (as a function of the inter-quartile range) are shown as black diamonds.

4.4 Conclusions & future steps

This work outlines a succesful dissection of a concerted RBFE approach into multiple steps. To the best of our knowledge, this is the first direct comparison between these two approaches within the same RBFE code using a complete dissection of the concerted protocol. Although there is no observable benefit of using the multistep approach over the concerted approach, two key findings are reported in this work. First, it appears that there is a discrepancy in how sulfonamide groups (and likely other similar functional groups) are perturbed in the multistep approach. Whether this discrepancy is caused by an algorithmic error or by systemic issues with growing such functional groups remains to be investigated with newer versions of SOMD and with alternative force fields. Second, the multistep approach outperforms the concerted approach when using hysteresis as the reliability metric for perturbations that involve perturbing a cyclical structure on the ligand scaffold. However, a larger investigation is required to fully determine whether this effect is consistent; such a screening would be worthwhile as cycle-growing perturbations are a common pitfall in RBFE and a novel method of dealing with these could be invaluable to the field.

Chapter 5

Concluding remarks

The work presented in this thesis spans a wide range of applications to AFE methodologies. After an introduction to supporting theory (*chapter 1*), work is presented that shows an example of hybridisation of RBFE and ML predictions (*chapter 2*). In this work it is shown that ML is able to support AFE by correcting for systematic errors resulting mainly from force field shortcomings. Next, in *chapter 3* a novel data-driven method of generating RBFE perturbation networks is presented where ML is used to more efficiently plan AFE calculations. Finally, a pure AFE study is presented in *chapter 4* that outlines investigations into whether deconstructing AFE perturbations into composite steps is beneficial and whether this deconstruction can lead to new discoveries regarding the underlying AFE software shortcomings.

5.1 Underlying themes in the thesis

There are several underlying themes that connect the research chapters presented in this chapter. All chapters touch on AFE errors and how to resolve them. Chapters 2 and 3 primarily focus on $(\Delta)\Delta G_{offset}$ which is a direct metric of systematic errors versus experimental measures, whereas chapter 4 focuses primarily on the statistical error as well as the hysteresis (the error between forward and reverse AFE transformations) as error metrics. The work presented in this thesis makes a strong case for increasing the emphasis on systematic and statistical error metrics in AFE research, and shows that modelling these can effectively benefit the method.

As this thesis is one of the first examples of modelling errors in AFE, there is a significant shortage of datasets to train ML models on. Whereas chapter 2 mostly involves dealing with this data shortage by introducing workarounds such as extensive cross-validation and feature selection protocols, chapter 3 outlines the generation of a novel AFE dataset that is carefully curated to represent a representative space of RBFE perturbations. Using this novel RBFE-Space, it is shown that ML models can be trained to learn statistical errors in AFE.

Although not extensively investigated throughout this thesis, model transferability is a frequently occurring problem in ML fields of research. It is expected that the majority of models presented in this work exhibit poor performance when subjected to different AFE software (with decreasing performance when increase dissimilarity).

5.2 Reflections on the future of AFE and ML methodologies

Hybridisation of AFE and ML methodologies has not been extensively researched prior to the studies performed in this thesis. Fortunately, during the four years leading to the current presentation of this thesis several peers have started doing similar work in this area, albeit in fundamentally different ways. As with seemingly all scientific fields, the future of AFE stands to benefit significantly from ML implementations. Although for the time being it is unlikely that ML methodologies will provide a pure replacement algorithm of AFE, hybridisation algorithms are positioned to drive the field of AFE forward by filling gaps in its methodology. Combined with further advances in computer hardware, CADD will be propelled forward with novel exciting methodologies that will reflect work presented in this thesis.

Bibliography

[1] S. Myers and A. Baker, *Nature Biotechnology*, 2001, **19**, 727–730.

[2] A. L. Nelson, E. Dhimolea and J. M. Reichert, *Nature Reviews Drug Discovery*, 2010, **9**, 767–774.

[3] I. Delany, R. Rappuoli and E. D. Gregorio, *EMBO Molecular Medicine*, 2014, **6**, 708–720.

[4] S. Morgan, P. Grootendorst, J. Lexchin, C. Cunningham and D. Greyson, *Health Policy*, 2011, **100**, 4–17.

[5] J. Hughes, S. Rees, S. Kalindjian and K. Philpott, *British Journal of Pharmacology*, 2011, **162**, 1239–1249.

[6] C. H. Wong, K. W. Siah and A. W. Lo, *Biostatistics*, 2018, **20**, 273–286.

[7] A. Bender and I. Cortés-Ciriano, *Drug Discovery Today*, 2021, **26**, 511–524.

[8] J. Ha, H. Park, J. Park and S. B. Park, *Cell Chemical Biology*, 2021, **28**, 394–423.

[9] S. Fox, S. Farr-Jones, L. Sopchak, A. Boggs, H. W. Nicely, R. Khoury and M. Biros, *SLAS Discovery*, 2006, **11**, 864–869.

[10] A. Bender and I. Cortes-Ciriano, *Drug Discovery Today*, 2021, **26**, 1040–1052.

[11] L. Zhao, H. L. Ciallella, L. M. Aleksunes and H. Zhu, *Drug Discovery Today*, 2020, **25**, 1624–1638.

[12] S. J. Y. Macalino, V. Gosu, S. Hong and S. Choi, *Archives of Pharmacal Research*, 2015, **38**, 1686–1701.

[13] G. Sliwoski, S. Kothiwale, J. Meiler and E. W. Lowe, *Pharmacological Reviews*, 2013, **66**, 334–395.

[14] Y. Hasin, M. Seldin and A. Lusis, *Genome Biology*, 2017, **18**, year.

[15] B. Dafniet, N. Cerisier, B. Boezio, A. Clary, P. Ducrot, T. Dorval, A. Gohier, D. Brown, K. Audouze and O. Taboureau, *Journal of Cheminformatics*, 2021, **13**, year.

[16] T. Rodrigues and G. J. Bernardes, *Current Opinion in Chemical Biology*, 2020, **56**, 16–22.

[17] N. S. Madhukar, P. K. Khade, L. Huang, K. Gayvert, G. Galletti, M. Stogniew, J. E. Allen, P. Giannakakou and O. Elemento, *Nature Communications*, 2019, **10**, year.

[18] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli and D. Hassabis, *Nature*, 2021, **596**, 583–589.

[19] J. García-Nafría and C. G. Tate, *Annual Review of Pharmacology and Toxicology*, 2020, **60**, 51–71.

[20] K. A. Dill and J. L. MacCallum, *Science*, 2012, **338**, 1042–1046.

[21] M. Akdel, D. E. V. Pires, E. P. Pardo, J. Jänes, A. O. Zalevsky, B. Mészáros, P. Bryant, L. L. Good, R. A. Laskowski, G. Pozzati, A. Shenoy, W. Zhu, P. Kundrotas, V. R. Serra, C. H. M. Rodrigues, A. S. Dunham, D. Burke,

N. Borkakoti, S. Velankar, A. Frost, K. Lindorff-Larsen, A. Valencia, S. Ovchinnikov, J. Durairaj, D. B. Ascher, J. M. Thornton, N. E. Davey, A. Stein, A. Elofsson, T. I. Croll and P. Beltrao, 2021.

[22] A. Gaulton, A. Hersey, M. Nowotka, A. P. Bento, J. Chambers, D. Mendez, P. Mutowo, F. Atkinson, L. J. Bellis, E. Cibrián-Uhalte, M. Davies, N. Dedman, A. Karlsson, M. P. Magariños, J. P. Overington, G. Papadatos, I. Smit and A. R. Leach, *Nucleic Acids Research*, 2016, **45**, D945–D954.

[23] Enamine, *REAL compounds*, `https://enamine.net/compound-collections/real-compounds`.

[24] J. J. Irwin, K. G. Tang, J. Young, C. Dandarchuluun, B. R. Wong, M. Khurelbaatar, Y. S. Moroz, J. Mayfield and R. A. Sayle, *Journal of Chemical Information and Modeling*, 2020, **60**, 6065–6073.

[25] P. Ripphausen, B. Nisius and J. Bajorath, *Drug Discovery Today*, 2011, **16**, 372–376.

[26] A.-J. Banegas-Luna, J. P. Cerón-Carrasco and H. Pérez-Sánchez, *Future Medicinal Chemistry*, 2018, **10**, 2641–2658.

[27] P. Willett, *Trends in Biotechnology*, 1995, **13**, 516–521.

[28] V. Venkatasubramanian, K. Chan and J. Caruthers, *Computers &amp Chemical Engineering*, 1994, **18**, 833–844.

[29] Y. Cheng, Y. Gong, Y. Liu, B. Song and Q. Zou, *Briefings in Bioinformatics*, 2021, **22**, year.

[30] M. Olivecrona, T. Blaschke, O. Engkvist and H. Chen, *Journal of Cheminformatics*, 2017, **9**, year.

[31] M. Popova, O. Isayev and A. Tropsha, *Science Advances*, 2018, **4**, year.

[32] R. Kim and J. Skolnick, *Journal of Computational Chemistry*, 2008, **29**, 1316–1331.

[33] T. Pantsar and A. Poso, *Molecules*, 2018, **23**, 1899.

[34] H. M. Berman, *Nucleic Acids Research*, 2000, **28**, 235–242.

[35] Z. Liu, Y. Li, L. Han, J. Li, J. Liu, Z. Zhao, W. Nie, Y. Liu and R. Wang, *Bioinformatics*, 2014, **31**, 405–412.

[36] M. Varadi, S. Anyango, M. Deshpande, S. Nair, C. Natassia, G. Yordanova, D. Yuan, O. Stroe, G. Wood, A. Laydon, A. Žídek, T. Green, K. Tunyasuvunakool, S. Petersen, J. Jumper, E. Clancy, R. Green, A. Vora, M. Lutfi, M. Figurnov, A. Cowie, N. Hobbs, P. Kohli, G. Kleywegt, E. Birney, D. Hassabis and S. Velankar, *Nucleic Acids Research*, 2021, **50**, D439–D444.

[37] M. A. Khamis, W. Gomaa and W. F. Ahmed, *Artificial Intelligence in Medicine*, 2015, **63**, 135–152.

[38] Y. Zhang, Y. Wang, W. Zhou, Y. Fan, J. Zhao, L. Zhu, S. Lu, T. Lu, Y. Chen and H. Liu, *Chemical Biology &amp Drug Design*, 2019, **93**, 685–699.

[39] D. E. Shaw, K. J. Bowers, E. Chow, M. P. Eastwood, D. J. Ierardi, J. L. Klepeis, J. S. Kuskin, R. H. Larson, K. Lindorff-Larsen, P. Maragakis, M. A. Moraes, R. O. Dror, S. Piana, Y. Shan, B. Towles, J. K. Salmon, J. P. Grossman, K. M. Mackenzie, J. A. Bank, C. Young, M. M. Deneroff and B. Batson, Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis - SC '09, 2009.

[40] P. Herrera-Nieto, A. Pérez and G. D. Fabritiis, *Scientific Reports*, 2020, **10**, year.

[41] F. Fogolari, A. Brigo and H. Molinari, *Biophysical Journal*, 2003, **85**, 159–166.

[42] H. Sun, Y. Li, S. Tian, L. Xu and T. Hou, *Phys. Chem. Chem. Phys.*, 2014, **16**, 16719–16729.

[43] T. Hansson, J. Marelius and J. Åqvist, *Journal of Computer-Aided Molecular Design*, 1998, **12**, 27–35.

[44] Z. Cournia, B. K. Allen, T. Beuming, D. A. Pearlman, B. K. Radak and W. Sherman, *Journal of Chemical Information and Modeling*, 2020, **60**, 4153–4169.

[45] V. Limongelli, M. Bonomi and M. Parrinello, *Proceedings of the National Academy of Sciences*, 2013, **110**, 6358–6363.

[46] S. Raniolo and V. Limongelli, *Nature Protocols*, 2020, **15**, 2837–2866.

[47] D. Shukla, C. X. Hernández, J. K. Weber and V. S. Pande, *Accounts of Chemical Research*, 2015, **48**, 414–422.

[48] B. E. Husic and V. S. Pande, *Journal of the American Chemical Society*, 2018, **140**, 2386–2396.

[49] H. P. Rang, M. M. Dale, J. M. Ritter, R. J. Flower and G. Henderson, *Rang &amp Dale's Pharmacology*, Elsevier, 2012, pp. 1–5.

[50] F. L. Lambert, *Journal of Chemical Education*, 2002, **79**, 1241.

[51] I. Y. Ben-Shalom, S. Pfeiffer-Marek, K.-H. Baringhaus and H. Gohlke, *Journal of Chemical Information and Modeling*, 2017, **57**, 170–189.

[52] C. en A. Chang, W. Chen and M. K. Gilson, *Proceedings of the National Academy of Sciences*, 2007, **104**, 1534–1539.

[53] N. Singh and A. Warshel, *Proteins: Structure, Function, and Bioinformatics*, 2010, **78**, 1724–1735.

[54] C.-E. Chang and M. K. Gilson, *Journal of the American Chemical Society*, 2004, **126**, 13156–13164.

[55] J. E. DeLorbe, J. H. Clements, M. G. Teresk, A. P. Benfield, H. R. Plake, L. E. Millspaugh and S. F. Martin, *Journal of the American Chemical Society*, 2009, **131**, 16758–16770.

[56] J. E. DeLorbe, J. H. Clements, B. B. Whiddon and S. F. Martin, *ACS Medicinal Chemistry Letters*, 2010, **1**, 448–452.

[57] X. Du, Y. Li, Y.-L. Xia, S.-M. Ai, J. Liang, P. Sang, X.-L. Ji and S.-Q. Liu, *International Journal of Molecular Sciences*, 2016, **17**, 144.

[58] M. H. Abraham, *Chemical Society Reviews*, 1993, **22**, 73.

[59] R. F. de Freitas and M. Schapira, *MedChemComm*, 2017, **8**, 1970–1981.

[60] E. Barratt, R. J. Bingham, D. J. Warner, C. A. Laughton, S. E. V. Phillips and S. W. Homans, *Journal of the American Chemical Society*, 2005, **127**, 11827–11834.

[61] T. W. Johnson, R. A. Gallego and M. P. Edwards, *Journal of Medicinal Chemistry*, 2018, **61**, 6401–6420.

[62] J. Zhu, H. Wang, S. Yang, L. Guo, Z. Li, W. Wang, S. Wang, W. Huang, L. Wang, T. Yang, Q. Ma and Y. Bi, *PLoS ONE*, 2013, **8**, e71153.

[63] H. J. Dyson, P. E. Wright and H. A. Scheraga, *Proceedings of the National Academy of Sciences*, 2006, **103**, 13057–13061.

[64] P. W. Snyder, M. R. Lockett, D. T. Moustakas and G. M. Whitesides, *The European Physical Journal Special Topics*, 2013, **223**, 853–891.

[65] F. Biedermann, W. M. Nau and H.-J. Schneider, *Angewandte Chemie International Edition*, 2014, **53**, 11158–11171.

[66] J. Schiebel, R. Gaspari, T. Wulsdorf, K. Ngo, C. Sohn, T. E. Schrader, A. Cavalli, A. Ostermann, A. Heine and G. Klebe, *Nature Communications*, 2018, **9**, year.

[67] V. Mikol, C. Papageorgiou and X. Borer, *Journal of Medicinal Chemistry*, 1995, **38**, 3361–3367.

[68] J. Michel, J. Tirado-Rives and W. L. Jorgensen, *Journal of the American Chemical Society*, 2009, **131**, 15403–15411.

[69] E. Braun, J. Gilmer, H. B. Mayes, D. L. Mobley, J. I. Monroe, S. Prasad and D. M. Zuckerman, *Living Journal of Computational Molecular Science*, 2019, **1**, year.

[70] S. Bottaro and K. Lindorff-Larsen, *Science*, 2018, **361**, 355–360.

[71] *Computational Many-Particle Physics*, ed. H. Fehske, R. Schneider and A. Weiße, Springer Berlin Heidelberg, 2008.

[72] D. Fincham, *Computer Physics Communications*, 1986, **40**, 263–269.

[73] C. W. Hopkins, S. L. Grand, R. C. Walker and A. E. Roitberg, *Journal of Chemical Theory and Computation*, 2015, **11**, 1864–1874.

[74] J. Fass, D. Sivak, G. Crooks, K. Beauchamp, B. Leimkuhler and J. Chodera, *Entropy*, 2018, **20**, 318.

[75] T. Schneider and E. Stoll, *Physical Review B*, 1978, **17**, 1302–1322.

[76] H. C. Andersen, *The Journal of Chemical Physics*, 1980, **72**, 2384–2393.

[77] C. Sagui and T. A. Darden, *Annual Review of Biophysics and Biomolecular Structure*, 1999, **28**, 155–179.

[78] G. A. Cisneros, M. Karttunen, P. Ren and C. Sagui, *Chemical Reviews*, 2013, **114**, 779–814.

[79] C. Woods, A. S.J.S. Mey, G. Calabrò and J. Michel, *Sire Molecular Simulation Framework*, 2019.

[80] J. Wang, W. Wang, P. A. Kollman and D. A. Case, *Journal of Molecular Graphics and Modelling*, 2006, **25**, 247–260.

[81] E. Harder, W. Damm, J. Maple, C. Wu, M. Reboul, J. Y. Xiang, L. Wang, D. Lupyan, M. K. Dahlgren, J. L. Knight, J. W. Kaus, D. S. Cerutti, G. Krilov, W. L. Jorgensen, R. Abel and R. A. Friesner, *Journal of Chemical Theory and Computation*, 2015, **12**, 281–296.

[82] W. Yu, X. He, K. Vanommeslaeghe and A. D. MacKerell, *Journal of Computational Chemistry*, 2012, **33**, 2451–2468.

[83] Y. Qiu, D. G. A. Smith, S. Boothroyd, H. Jang, D. F. Hahn, J. Wagner, C. C. Bannan, T. Gokey, V. T. Lim, C. D. Stern, A. Rizzi, B. Tjanaka, G. Tresadern, X. Lucas, M. R. Shirts, M. K. Gilson, J. D. Chodera, C. I. Bayly, D. L. Mobley and L.-P. Wang, *Journal of Chemical Theory and Computation*, 2021, **17**, 6262–6280.

[84] J. T. Horton, A. E. A. Allen, L. S. Dodda and D. J. Cole, *Journal of Chemical Information and Modeling*, 2019, **59**, 1366–1381.

[85] J. A. Maier, C. Martinez, K. Kasavajhala, L. Wickstrom, K. E. Hauser and C. Simmerling, *Journal of Chemical Theory and Computation*, 2015, **11**, 3696–3713.

[86] M. J. Robertson, J. Tirado-Rives and W. L. Jorgensen, *Journal of Chemical Theory and Computation*, 2015, **11**, 3499–3509.

[87] S. P. K. Pathirannahalage, N. Meftahi, A. Elbourne, A. C. G. Weiss, C. F. McConville, A. Padua, D. A. Winkler, M. C. Gomes, T. L. Greaves, T. C. Le, Q. A. Besford and A. J. Christofferson, *Journal of Chemical Information and Modeling*, 2021, **61**, 4521–4536.

[88] W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey and M. L. Klein, *The Journal of Chemical Physics*, 1983, **79**, 926–935.

[89] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren and J. Hermans, *The Jerusalem Symposia on Quantum Chemistry and Biochemistry*, Springer Netherlands, 1981, pp. 331–342.

[90] S. A. Hollingsworth and R. O. Dror, *Neuron*, 2018, **99**, 1129–1143.

[91] J. Gelpi, A. Hospital, R. Goñi and M. Orozco, *Advances and Applications in Bioinformatics and Chemistry*, 2015, 37.

[92] R. B. Shirts, S. R. Burt and A. M. Johnson, *The Journal of Chemical Physics*, 2006, **125**, 164102.

[93] A. R. Leach, *Molecular modelling*, Prentice-Hall, London, England, 2nd edn, 2001.

[94] J. Nickolls and W. J. Dally, *IEEE Micro*, 2010, **30**, 56–69.

[95] A. S. J. S. Mey, B. K. Allen, H. E. Bruce McDonald, J. D. Chodera, D. F. Hahn, M. Kuhn, J. Michel, D. L. Mobley, L. N. Naden, S. Prasad, A. Rizzi, J. Scheen, M. R. Shirts, G. Tresadern and H. Xu, *Living Journal of Computational Molecular Science*, 2020, **2**, 18378.

[96] D. H. D. Jong, L. V. Schäfer, A. H. D. Vries, S. J. Marrink, H. J. C. Berendsen and H. Grubmüller, *J. Comput. Chem.*, 2011, **32**, 1919–1928.

[97] A. C. Pan, H. Xu, T. Palpant and D. E. Shaw, *J. Chem. Theory Comput.*, 2017, **13**, 3372–3377.

[98] I. Teo, C. G. Mayne, K. Schulten and T. Lelièvre, *J. Chem. Theory Comput.*, 2016, **12**, 2983–2989.

[99] L. W. Votapka, B. R. Jagger, A. L. Heyneman and R. E. Amaro, *J. Phys. Chem. B*, 2017, **121**, 3597–3606.

[100] S. Doerr and G. De Fabritiis, *J. Chem. Theory Comput.*, 2014, **10**, 2064–2069.

[101] N. Plattner and F. Noé, *Nat. Commun.*, 2015, **6**, 1–10.

[102] T. Dixon, S. D. Lotz and A. Dickson, *J. Comput. Aided Mol. Des.*, 2018, **32**, 1001–1012.

[103] A. Basavapathruni, L. Jin, S. R. Daigle, C. R. A. Majer, C. A. Therkelsen, T. J. Wigle, K. W. Kuntz, R. Chesworth, R. M. Pollock, M. P. Scott, M. P. Moyer, V. M. Richon, R. A. Copeland and E. J. Olhava, *Chemical Biology & Drug Design*, 2012, **80**, 971–980.

[104] D. E. Hyre, I. L. Trong, E. A. Merritt, J. F. Eccleston, N. M. Green, R. E. Stenkamp and P. S. Stayton, *Protein Sci.*, 2006, **15**, 459–467.

[105] P. Eastman, J. Swails, J. D. Chodera, R. T. McGibbon, Y. Zhao, K. A. Beauchamp, L.-P. Wang, A. C. Simmonett, M. P. Harrigan, C. D. Stern, R. P. Wiewiora, B. R. Brooks and V. S. Pande, *PLoS Comput. Biol.*, 2017, **13**, e1005659.

[106] C. Kutzner, S. Páll, M. Fechner, A. Esztermann, B. L. de Groot and H. Grubmüller, *J. Comput. Chem.*, 2019, **40**, 2418–2431.

[107] H.-J. Woo and B. Roux, *Proc. Natl. Acad. Sci.*, 2005, **102**, 6825–6830.

[108] C. Velez-Vega and M. K. Gilson, *J. Comput. Chem.*, 2013, **34**, 2360–2371.

[109] V. Limongelli, M. Bonomi and M. Parrinello, *Proc. Natl. Acad. Sci.*, 2013, **110**, 6358–6363.

[110] G. Heinzelmann, N. M. Henriksen and M. K. Gilson, *J. Chem. Theory Comput.*, 2017, **13**, 3260–3275.

[111] L. Wang, Y. Wu, Y. Deng, B. Kim, L. Pierce, G. Krilov, D. Lupyan, S. Robinson, M. K. Dahlgren, J. Greenwood, D. L. Romero, C. Masse, J. L. Knight, T. Steinbrecher, T. Beuming, W. Damm, E. Harder, W. Sherman, M. Brewer, R. Wester, M. Murcko, L. Frye, R. Farid, T. Lin, D. L. Mobley, W. L. Jorgensen, B. J. Berne, R. A. Friesner and R. Abel, *J. Am. Chem. Soc.*, 2015, **137**, 2695–2703.

[112] V. Gapsys, D. F. Hahn, G. Tresadern, D. L. Mobley, M. Rampp and B. L.

de Groot, *Journal of Chemical Information and Modeling*, 2022, **62**, 1172–1177.

[113] S. Liu, Y. Wu, T. Lin, R. Abel, J. P. Redmann, C. M. Summa, V. R. Jaber, N. M. Lim and D. L. Mobley, *J. Comput.-Aided Mol. Des.*, 2013, **27**, 755–770.

[114] Q. Yang, W. W. Burchett, G. S. Steeno, D. L. Mobley and X. Hou, 2019.

[115] T. T. Pham and M. R. Shirts, *The Journal of Chemical Physics*, 2011, **135**, 034114.

[116] T. C. Beutler, A. E. Mark, R. C. van Schaik, P. R. Gerber and W. F. van Gunsteren, *Chemical Physics Letters*, 1994, **222**, 529–539.

[117] M. Zacharias, T. P. Straatsma and J. A. McCammon, *The Journal of Chemical Physics*, 1994, **100**, 9025–9031.

[118] A. Blondel, *Journal of Computational Chemistry*, 2004, **25**, 985–993.

[119] V. Gapsys, D. Seeliger and B. L. de Groot, *Journal of Chemical Theory and Computation*, 2012, **8**, 2373–2382.

[120] M. Jorge, N. M. Garrido, A. J. Queimada, I. G. Economou and E. A. Macedo, *Journal of Chemical Theory and Computation*, 2010, **6**, 1018–1027.

[121] C. Shyu and F. M. Ytreberg, *Journal of Computational Chemistry*, 2009, NA–NA.

[122] M. R. Shirts and J. D. Chodera, *The Journal of Chemical Physics*, 2008, **129**, 124105.

[123] P. V. Klimovich, M. R. Shirts and D. L. Mobley, *Journal of Computer-Aided Molecular Design*, 2015, **29**, 397–411.

[124] C. H. Bennett, *Journal of Computational Physics*, 1976, **22**, 245–268.

[125] C. Schindler, H. Baumann, A. Blum, D. Böse, H.-P. Buchstaller, L. Burgdorf, D. Cappel, E. Chekler, P. Czodrowski, D. Dorsch, M. Escguida, B. Follows, T. Fuchß, U. Grädler, J. Gunera, T. Johnson, C. J. Lebrun, S. Karra, M. Klein, L. Kötzner, T. Knehans, M. Krier, M. Leiendecker, B. Leuthner, L. Li, I. Mochalkin, D. Musil, C. Neagu, F. Rippmann, K. Schiemann, R. Schulz, T. Steinbrecher, E.-M. Tanzer, A. U. Lopez, A. V. Follis, A. Wegener and D. Kuhn, *ChemRxiv*, 2020.

[126] G. Scarabelli, E. O. Oloo, J. K. Maier and A. Rodriguez-Granillo, *Journal of Molecular Biology*, 2022, **434**, 167375.

[127] J. L. Knight, K. Leswing, P. H. Bos and L. Wang, *Free Energy Methods in Drug Discovery: Current State and Future Directions*, American Chemical Society, 2021, pp. 205–226.

[128] J. L. Paulsen, H. S. Yu, D. Sindhikara, L. Wang, T. Appleby, A. G. Villaseñor, U. Schmitz and D. Shivakumar, *Journal of Chemical Information and Modeling*, 2020, **60**, 3489–3498.

[129] G. A. Ross, E. Russell, Y. Deng, C. Lu, E. D. Harder, R. Abel and L. Wang, *Journal of Chemical Theory and Computation*, 2020, **16**, 6061–6076.

[130] E. Jacoby, H. V. Vlijmen, O. Querolle, I. Stansfield, L. Meerpoel, M. Versele, G. Hynd and R. Attar, *Future Drug Discovery*, 2020, **2**, year.

[131] F. Deflorian, L. Perez-Benito, E. B. Lenselink, M. Congreve, H. W. T. van Vlijmen, J. S. Mason, C. de Graaf and G. Tresadern, *Journal of Chemical Information and Modeling*, 2020, **60**, 5563–5579.

[132] C.-H. Zhang, K. A. Spasov, R. A. Reilly, K. Hollander, E. A. Stone, J. A. Ippolito, M.-E. Liosi, M. G. Deshmukh, J. Tirado-Rives, S. Zhang, Z. Liang, S. J. Miller, F. Isaacs, B. D. Lindenbach, K. S. Anderson and W. L. Jorgensen, *ACS Medicinal Chemistry Letters*, 2021, **12**, 1325–1332.

[133] F. von Delft, M. Calmiano, J. Chodera, E. Griffen, A. Lee, N. London, T. Matviuk, B. Perry, M. Robinson and A. von Delft, *Nature*, 2021, **594**, 330–332.

[134] H. M. Baumann, V. Gapsys, B. L. de Groot and D. L. Mobley, *The Journal of Physical Chemistry B*, 2021, **125**, 4241–4261.

[135] B. Ramsundar, P. Eastman, P. Walters, V. Pande, K. Leswing and Z. Wu, *Deep Learning for the Life Sciences*, O'Reilly Media, 2019.

[136] M. Ragoza, J. Hochuli, E. Idrobo, J. Sunseri and D. R. Koes, *J. Chem. Inf. Model.*, 2017, **57**, 942–957.

[137] H. Zeng, M. D. Edwards, G. Liu and D. K. Gifford, *Bioinformatics*, 2016, **32**, i121–i127.

[138] H. Chen, O. Engkvist, Y. Wang, M. Olivecrona and T. Blaschke, *Drug Discovery Today*, 2018, **23**, 1241–1250.

[139] H. Chen, T. Kogej and O. Engkvist, *Molecular Informatics*, 2018, **37**, 1800041.

[140] R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams and A. Aspuru-Guzik, *ACS Central Science*, 2018, **4**, 268–276.

[141] P.-S. Huang, S. E. Boyken and D. Baker, *Nature*, 2016, **537**, 320–327.

[142] M. Su, G. Feng, Z. Liu, Y. Li and R. Wang, *Journal of Chemical Information and Modeling*, 2020, **60**, 1122–1136.

[143] D. S. Watson, J. Krutzinna, I. N. Bruce, C. E. Griffiths, I. B. McInnes, M. R. Barnes and L. Floridi, *BMJ*, 2019, l886.

[144] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing and V. Pande, *Chemical Science*, 2018, **9**, 513–530.

[145] G. M. Maggiora, *Journal of Chemical Information and Modeling*, 2006, **46**, 1535–1535.

[146] D. A. Winkler and T. C. Le, *Molecular Informatics*, 2016, **36**, 1600118.

[147] C. W. Coley, W. Jin, L. Rogers, T. F. Jamison, T. S. Jaakkola, W. H. Green, R. Barzilay and K. F. Jensen, *Chemical Science*, 2019, **10**, 370–377.

[148] L. Breiman, *Machine Learning*, 2001, **45**, 5–32.

[149] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer Science & Business Media, 2013.

[150] Z. Cournia, B. Allen and W. Sherman, *J. Chem. Inf. Model.*, 2017, **57**, 2911–2937.

[151] A. D. Simone, C. Georgiou, H. Ioannidis, A. A. Gupta, J. Juárez-Jiménez, D. Doughty-Shenton, E. A. Blackburn, M. A. Wear, J. P. Richards, P. N. Barlow, N. Carragher, M. D. Walkinshaw, A. N. Hulme and J. Michel, *Chem. Sci.*, 2019, **10**, 542–547.

[152] B. Kuhn, M. Tichý, L. Wang, S. Robinson, R. E. Martin, A. Kuglstatter, J. Benz, M. Giroud, T. Schirmeister, R. Abel, F. Diederich and J. Hert, *J. Med. Chem.*, 2017, **60**, 2485–2497.

[153] C. Georgiou, I. McNae, M. Wear, H. Ioannidis, J. Michel and M. Walkinshaw, *J. Mol. Biol.*, 2017, **429**, 2556–2570.

[154] J. Michel, *Phys. Chem. Chem. Phys.*, 2014, **16**, 4465–4477.

[155] S. K. Mishra, G. Calabró, H. H. Loeffler, J. Michel and J. Koča, *J. Chem. Theory Comput.*, 2015, **11**, 3333–3345.

[156] I.-J. Chen and N. Foloppe, *Drug Dev. Res.*, 2011, **72**, 85–94.

[157] G. J. Rocklin, D. L. Mobley and K. A. Dill, *J. Chem. Theory Comput.*, 2013, **9**, 3072–3083.

[158] G. Calabrò, C. J. Woods, F. Powlesland, A. S. J. S. Mey, A. J. Mulholland and J. Michel, *J. Phys. Chem. B*, 2016, **120**, 5340–5350.

[159] H. H. Loeffler, S. Bosisio, G. Duarte Ramos Matos, D. Suh, B. Roux, D. L. Mobley and J. Michel, *J. Chem. Theory Comput.*, 2018, **14**, 5567–5582.

[160] M. Kuhn, S. Firth-Clark, P. Tosco, A. S. J. S. Mey, M. Mackey and J. Michel, *J. Chem. Inf. Model.*, 2020, in press.

[161] L. F. Song, T.-S. Lee, C. Zhu, D. M. York and K. M. Merz, *J. Chem. Inf. Model.*, 2019, **59**, 3128–3135.

[162] V. Gapsys, L. Pérez-Benito, M. Aldeghi, D. Seeliger, H. v. Vlijmen, G. Tresadern and B. L. d. Groot, *Chem. Sci.*, 2020, **11**, 1140–1152.

[163] A. Rizzi, T. Jensen, D. R. Slochower, M. Aldeghi, V. Gapsys, D. Ntekoumes, S. Bosisio, M. Papadourakis, N. M. Henriksen, B. L. de Groot, Z. Cournia, A. Dickson, J. Michel, M. K. Gilson, M. R. Shirts, D. L. Mobley and J. D. Chodera, *J. Comput.-Aided Mol. Des.*, 2020.

[164] J. Yin, N. M. Henriksen, D. R. Slochower, M. R. Shirts, M. W. Chiu, D. L. Mobley and M. K. Gilson, *J. Comput.-Aided Mol. Des.*, 2017, **31**, 1–19.

[165] C. D. Parks, Z. Gaieb, M. Chiu, H. Yang, C. Shao, W. P. Walters, J. M. Jansen, G. McGaughey, R. A. Lewis, S. D. Bembenek, M. K. Ameriks, T. Mirzadegan, S. K. Burley, R. E. Amaro and M. K. Gilson, *J. Comput.-Aided Mol. Des.*, 2020, **34**, 99–119.

[166] A. S. J. S. Mey, J. J. Jiménez and J. Michel, *J. Comput.-Aided Mol. Des.*, 2018, **32**, 199–210.

[167] J. M. Granadino-Roldán, A. S. J. S. Mey, J. J. P. González, S. Bosisio, J. Rubio-Martinez and J. Michel, *PLOS ONE*, 2019, **14**, e0213217.

[168] M. Papadourakis, S. Bosisio and J. Michel, *J. Comput.-Aided Mol. Des.*, 2018, **32**, 1047–1058.

[169] D. Shivakumar, J. Williams, Y. Wu, W. Damm, J. Shelley and W. Sherman, *J. Chem. Theory Comput.*, 2010, **6**, 1509–1519.

[170] S. A. Martins, S. F. Sousa, M. J. Ramos and P. A. Fernandes, *J. Chem. Theory Comput.*, 2014, **10**, 3570–3577.

[171] D. L. Mobley, C. I. Bayly, M. D. Cooper and K. A. Dill, *J. Phys. Chem. B*, 2009, **113**, 4533–4537.

[172] C. C. Bannan, K. H. Burley, M. Chiu, M. R. Shirts, M. K. Gilson and D. L. Mobley, *J. Comput.-Aided Mol. Des.*, 2016, **30**, 927–944.

[173] C. Tian, K. Kasavajhala, K. A. A. Belfon, L. Raguette, H. Huang, A. N. Migues, J. Bickel, Y. Wang, J. Pincay, Q. Wu and C. Simmerling, *J. Chem. Theory Comput.*, 2020, **16**, 528–552.

[174] J. Huang, S. Rauscher, G. Nawrocki, T. Ran, M. Feig, B. L. de Groot, H. Grubmüller and A. D. MacKerell, *Nat. Methods*, 2017, **14**, 71–73.

[175] D. R. Slochower, N. M. Henriksen, L.-P. Wang, J. D. Chodera, D. L. Mobley and M. K. Gilson, *J. Chem. Theory Comput.*, 2019, **15**, 6225–6242.

[176] D. L. Mobley, C. C. Bannan, A. Rizzi, C. I. Bayly, J. D. Chodera, V. T. Lim, N. M. Lim, K. A. Beauchamp, D. R. Slochower, M. R. Shirts, M. K. Gilson and P. K. Eastman, *J. Chem. Theory Comput.*, 2018, **14**, 6076–6092.

[177] Z. Jing, C. Liu, S. Y. Cheng, R. Qi, B. D. Walker, J.-P. Piquemal and P. Ren, *Annu. Rev. Biophys.*, 2019, **48**, 371–394.

[178] F. R. Beierlein, J. Michel and J. W. Essex, *J. Phys. Chem. B*, 2011, **115**, 4911–4926.

[179] G. König, F. C. Pickard, Y. Mei and B. R. Brooks, *J. Comput.-Aided Mol. Des.*, 2014, **28**, 245–257.

[180] J. S. Smith, O. Isayev and A. E. Roitberg, *Chem. Sci.*, 2017, **8**, 3192–3203.

[181] M. Rupp, A. Tkatchenko, K.-R. Müller and O. A. von Lilienfeld, *Phys. Rev. Lett.*, 2012, **108**, 058301.

[182] M. Wójcikowski, P. J. Ballester and P. Siedlecki, *Sci. Rep.*, 2017, **7**, 46710.

[183] Q. U. Ain, A. Aleksandrova, F. D. Roessler and P. J. Ballester, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 2015, **5**, 405–424.

[184] H. Lim and Y. Jung, *Chem. Sci.*, 2019, **10**, 8306–8315.

[185] H. Lim and Y. Jung, *arXiv:2005.06182 [cond-mat, physics:physics, stat]*, 2020.

[186] S. Riniker, *J. Chem. Inf. Model.*, 2017, **57**, 726–741.

[187] S. T. Hutchinson and R. Kobayashi, *J. Chem. Inf. Model.*, 2019, **59**, 1338–1346.

[188] B. Ramsundar, P. Eastman, Patrick Walters, V. Pande, Karl Leswing and Zhenqin Wu, *Deep Learning for the Life Sciences*, OReilly, Sebastopol, CA, US, 2019.

[189] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing and V. Pande, *Chem. Sci.*, 2018, **9**, 513–530.

[190] J. Jiménez-Luna, L. Pérez-Benito, G. Martínez-Rosell, S. Sciabola, R. Torella, G. Tresadern and G. D. Fabritiis, *Chem. Sci.*, 2019, **10**, 10911–10918.

[191] E. N. Feinberg, D. Sur, Z. Wu, B. E. Husic, H. Mai, Y. Li, S. Sun, J. Yang, B. Ramsundar and V. S. Pande, *ACS Cent. Sci.*, 2018, **4**, 1520–1530.

[192] S. Bosisio, A. S. J. S. Mey and J. Michel, *J. Comput.-Aided Mol. Des.*, 2017, **31**, 61–70.

[193] M. J. Abraham, T. Murtola, R. Schulz, S. Páll, J. C. Smith, B. Hess and E. Lindahl, *SoftwareX*, 2015, **1-2**, 19–25.

[194] D. L. Mobley and J. P. Guthrie, *J. Comput.-Aided Mol. Des.*, 2014, **28**, 711–720.

[195] G. Duarte Ramos Matos, D. Y. Kyu, H. H. Loeffler, J. D. Chodera, M. R. Shirts and D. L. Mobley, *J. Chem. Eng. Data*, 2017, **62**, 1559–1569.

[196] J. Wang, R. M. Wolf, J. W. Caldwell, P. A. Kollman and D. A. Case, *J. Comput. Chem.*, 2004, **25**, 1157–1174.

[197] A. Jakalian, B. L. Bush, D. B. Jack and C. I. Bayly, *J. Comput. Chem.*, 2000, **21**, 132–146.

[198] W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey and M. L. Klein, *J. Chem. Phys.*, 1983, **79**, 926–935.

[199] D. L. Mobley, C. I. Bayly, M. D. Cooper, M. R. Shirts and K. A. Dill, *J. Chem. Theory Comput.*, 2009, **5**, 350–358.

[200] D. L. Mobley, K. L. Wymer, N. M. Lim and J. P. Guthrie, *J. Comput.-Aided Mol. Des.*, 2014, **28**, 135–150.

[201] G. Landrum, *RDKit: Open-source cheminformatics*, 2020, `https://github.com/rdkit/rdkit`.

[202] H. Moriwaki, Y.-S. Tian, N. Kawashita and T. Takagi, *J. Cheminf.*, 2018, **10**, 4.

[203] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and Duchesnay, *J. Mach. Learn. Res.*, 2011, **12**, 2825–2830.

[204] T. Head, MechCoder, G. Louppe, I. Shcherbatyi, fcharras, Z. Vinícius, cmmalone, C. Schröder, nel215, N. Campos, T. Young, S. Cereda, T. Fan, rene rex, K. K. Shi, J. Schwabedal, carlosdanielcsantos, Hvass-Labs, M. Pak, SoManyUsernamesTaken, F. Callaway, L. Estève, L. Besson, M. Cherti, K. Pfannschmidt, F. Linzberger, C. Cauet, A. Gut, A. Mueller and A. Fabisch,

scikit-optimize/scikit-optimize: v0.5.2, 2018, https://zenodo.org/record/1207017#.XNWN045KhaQ.

[205] D. Rogers and M. Hahn, *J. Chem. Inf. Model.*, 2010, **50**, 742–754.

[206] L. Sandberg, *J. Comput.-Aided Mol. Des.*, 2014, **28**, 211–219.

[207] J. P. G. S. B. J. R.-M. JM Granadino-Roldán, ASJS Mey and J. Michel, *PLoS ONE*, 2019, **14**, year.

[208] L. Wang, Y. Wu, Y. Deng, B. Kim, L. Pierce, G. Krilov, D. Lupyan, S. Robinson, M. K. Dahlgren, J. Greenwood, D. L. Romero, C. Masse, J. L. Knight, T. Steinbrecher, T. Beuming, W. Damm, E. Harder, W. Sherman, M. Brewer, R. Wester, M. Murcko, L. Frye, R. Farid, T. Lin, D. L. Mobley, W. L. Jorgensen, B. J. Berne, R. A. Friesner and R. Abel, *Journal of the American Chemical Society*, 2015, **137**, 2695–2703.

[209] L. Pérez-Benito, N. Casajuana-Martin, M. Jiménez-Rosés, H. van Vlijmen and G. Tresadern, *J. Chem. Theory Comput.*, 2019, **15**, 1884–1895.

[210] D. Kim, J. E. Kowalchick, L. L. Brockunier, E. R. Parmee, G. J. Eiermann, M. H. Fisher, H. He, B. Leiting, K. Lyons, G. Scapin, S. B. Patel, A. Petrov, K. D. Pryor, R. S. Roy, J. K. Wu, X. Zhang, M. J. Wyvratt, B. B. Zhang, L. Zhu, N. A. Thornberry and A. E. Weber, *J. Med. Chem.*, 2008, **51**, 589–602.

[211] *Forge: Powerful computational tool to understand SAR & design*, 2012, https://www.cresset-group.com/products/forge/.

[212] M. Kuhn, S. Firth-Clark, P. Tosco, A. S. J. S. Mey, M. Mackey and J. Michel, *Journal of Chemical Information and Modeling*, 2020, **60**, 3120–3130.

[213] H. Moriwaki, Y.-S. Tian, N. Kawashita and T. Takagi, *J. Cheminf.*, 2018, **10**, 4.

[214] Cresset, *Flare V6 released: New features, cutting-edge science and improved functionality*, 2022, `https://www.cresset-group.com/about/news/flare-v6-released/`.

[215] M. Wójcikowski, P. Zielenkiewicz and P. Siedlecki, *J. Cheminf.*, 2015, **7**, 26.

[216] S. Bosisio, A. S. J. S. Mey and J. Michel, *J. Comput.-Aided Mol. Des.*, 2016, **30**, 1101–1114.

[217] K. A. Armacost, S. Riniker and Z. Cournia, *Journal of Chemical Information and Modeling*, 2020, **60**, 1–5.

[218] Z. Cournia, B. Allen and W. Sherman, *Journal of Chemical Information and Modeling*, 2017, **57**, 2911–2937.

[219] Y. Khalak, G. Tresadern, M. Aldeghi, H. M. Baumann, D. L. Mobley, B. L. de Groot and V. Gapsys, *Chemical Science*, 2021, **12**, 13958–13971.

[220] C. Mendoza-Martinez, M. Papadourakis, S. Llabrés, A. A. Gupta, P. N. Barlow and J. Michel, *Chemical Science*, 2022, **13**, 5220–5229.

[221] S. Bosisio, A. S. J. S. Mey and J. Michel, *Journal of Computer-Aided Molecular Design*, 2016, **31**, 61–70.

[222] A. Rizzi, T. Jensen, D. R. Slochower, M. Aldeghi, V. Gapsys, D. Ntekoumes, S. Bosisio, M. Papadourakis, N. M. Henriksen, B. L. de Groot, Z. Cournia, A. Dickson, J. Michel, M. K. Gilson, M. R. Shirts, D. L. Mobley and J. D. Chodera, *Journal of Computer-Aided Molecular Design*, 2020, **34**, 601–633.

[223] G. J. Rocklin, S. E. Boyce, M. Fischer, I. Fish, D. L. Mobley, B. K. Shoichet and K. A. Dill, *J. Mol. Biol.*, 2013, **425**, 4569–4583.

[224] H. H. Loeffler, S. Bosisio, G. Duarte Ramos Matos, D. Suh, B. Roux, D. L. Mobley and J. Michel, *Journal of Chemical Theory and Computation*, 2018, **14**, 5567–5582.

[225] Y. Qiu, D. G. A. Smith, S. Boothroyd, H. Jang, D. F. Hahn, J. Wagner, C. C. Bannan, T. Gokey, V. T. Lim, C. D. Stern, A. Rizzi, B. Tjanaka, G. Tresadern, X. Lucas, M. R. Shirts, M. K. Gilson, J. D. Chodera, C. I. Bayly, D. L. Mobley and L.-P. Wang, *Journal of Chemical Theory and Computation*, 2021, **17**, 6262–6280.

[226] D. F. Hahn, C. I. Bayly, H. E. B. Macdonald, J. D. Chodera, V. Gapsys, A. S. J. S. Mey, D. L. Mobley, L. P. Benito, C. E. M. Schindler, G. Tresadern and G. L. Warren, *Best practices for constructing, preparing, and evaluating protein-ligand binding affinity benchmarks*, 2021, `https://arxiv.org/abs/2105.06222`.

[227] C. E. M. Schindler, H. Baumann, A. Blum, D. Böse, H.-P. Buchstaller, L. Burgdorf, D. Cappel, E. Chekler, P. Czodrowski, D. Dorsch, M. K. I. Eguida, B. Follows, T. Fuchß, U. Grädler, J. Gunera, T. Johnson, C. Jorand Lebrun, S. Karra, M. Klein, T. Knehans, L. Koetzner, M. Krier, M. Leiendecker, B. Leuthner, L. Li, I. Mochalkin, D. Musil, C. Neagu, F. Rippmann, K. Schiemann, R. Schulz, T. Steinbrecher, E.-M. Tanzer, A. Unzue Lopez, A. Viacava Follis, A. Wegener and D. Kuhn, *Journal of Chemical Information and Modeling*, 2020, **60**, 5457–5474.

[228] V. Gapsys, D. F. Hahn, G. Tresadern, D. L. Mobley, M. Rampp and B. L. de Groot, *Journal of Chemical Information and Modeling*, 2022, **62**, 1172–1177.

[229] A. S. J. S. Mey, J. J. Jiménez and J. Michel, *Journal of Computer-Aided Molecular Design*, 2017, **32**, 199–210.

[230] M. M. Reif and C. Oostenbrink, *Journal of Computational Chemistry*, 2013, **35**, 227–243.

[231] Q. Yang, W. Burchett, G. S. Steeno, S. Liu, M. Yang, D. L. Mobley and X. Hou, *Journal of Computational Chemistry*, 2020, **41**, 247–257.

[232] S. Liu, Y. Wu, T. Lin, R. Abel, J. P. Redmann, C. M. Summa, V. R. Jaber, N. M. Lim and D. L. Mobley, *Journal of Computer-Aided Molecular Design*, 2013, **27**, 755–770.

[233] *Flare*, `https://www.cresset-group.com/tag/flare/`.

[234] H. Xu, *Journal of Chemical Information and Modeling*, 2019, **59**, 4720–4728.

[235] P. Li, Z. Li, Y. Wang, H. Dou, B. K. Radak, B. K. Allen, W. Sherman and H. Xu, *Journal of Chemical Theory and Computation*, 2022, **18**, 650–663.

[236] A. T. McNutt and D. R. Koes, *Journal of Chemical Information and Modeling*, 2022, **62**, 1819–1829.

[237] J. Scheen, W. Wu, A. S. J. S. Mey, P. Tosco, M. Mackey and J. Michel, *Journal of Chemical Information and Modeling*, 2020, **60**, 5331–5339.

[238] G. Landrum, *Open-Source Cheminformatics Software*, `http://www.rdkit.org/`.

[239] G. Landrum, *Molecule highlighting and R-group decomposition*, 2020, `http://rdkit.blogspot.com/2020/10/molecule-highlighting-and-r-group.html`.

[240] P. Schmidtke, *Grafting fragments onto molecules in rdkit - babysteps*, 2021, `https://pschmidtke.github.io/blog/rdkit/3d-editor/2021/01/23/grafting-fragments.html`.

[241] L. O. Hedges, A. S. Mey, C. A. Laughton, F. L. Gervasio, A. J. Mulholland, C. J. Woods and J. Michel, *Journal of Open Source Software*, 2019, **4**, 1831.

[242] G. Calabrò, C. J. Woods, F. Powlesland, A. S. J. S. Mey, A. J. Mulholland and J. Michel, *The Journal of Physical Chemistry B*, 2016, **120**, 5340–5350.

[243] M. R. Shirts and J. D. Chodera, *The Journal of Chemical Physics*, 2008, **129**, 124105.

[244] D. Chicco, in *Siamese Neural Networks: An Overview*, ed. H. Cartwright, Springer US, New York, NY, 2021, pp. 73–94.

[245] G. Koch, R. Zemel and R. Salakhutdinov, *WCP*, 2015, **37**, year.

[246] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals and G. E. Dahl, Proceedings of the 34th International Conference on Machine Learning - Volume 70, 2017, p. 1263–1272.

[247] B. Ramsundar, P. Eastman, P. Walters, V. Pande, K. Leswing and Z. Wu, *Deep Learning for the Life Sciences*, O'Reilly Media, 2019.

[248] A. Kensert, *Keras documentation: Message-passing neural network for molecular property prediction*, 2021, `https://keras.io/examples/graph/mpnn-molecular-graphs/`.

[249] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong and Q. He, *A Comprehensive Survey on Transfer Learning*, 2019, `https://arxiv.org/abs/1911.02685`.

[250] J. S. Delaney, *Journal of Chemical Information and Computer Sciences*, 2004, **44**, 1000–1005.

[251] P. Walters, 2018, `http://practicalcheminformatics.blogspot.com/2018/09/predicting-aqueous-solubility-its.html`.

[252] G. B. Goh, C. Siegel, A. Vishnu and N. O. Hodas, *Using Rule-Based Labels for Weak Supervised Learning: A ChemNet for Transferable Chemical Property Prediction*, 2018.

[253] X. Li and D. Fourches, *Inductive transfer learning for molecular activity prediction: Next-Gen QSAR Models with MolPMoFiT*, 2020, `https://jcheminf.biomedcentral.com/articles/10.1186/s13321-020-00430-x#citeas`.

[254] N. Schneider, D. M. Lowe, R. A. Sayle and G. A. Landrum, *J. Chem. Inf. Model.*, 2015, **55**, 39–53.

[255] A. Mey, M. Mackey, P. Tosco, J. Scheen and J. Michel, *FreeEnergyNetworkAnalysis*, `https://github.com/michellab/freenrgworkflows/tree/devel`.

[256] V. Fung, J. Zhang, E. Juarez and B. G. Sumpter, *npj Computational Materials*, 2021, **7**, 1.

[257] D. Jiang, Z. Wu, C.-Y. Hsieh, G. Chen, B. Liao, Z. Wang, C. Shen, D. Cao, J. Wu and T. Hou, *Journal of Cheminformatics*, 2021, **13**, year.

[258] Y. Wang, J. Wang, Z. Cao and A. B. Farimani, *Nature Machine Intelligence*, 2022, **4**, 279–287.

[259] C. W. Coley, W. Jin, L. Rogers, T. F. Jamison, T. S. Jaakkola, W. H. Green, R. Barzilay and K. F. Jensen, *Chemical Science*, 2019, **10**, 370–377.

[260] C. Cai, S. Wang, Y. Xu, W. Zhang, K. Tang, Q. Ouyang, L. Lai and J. Pei, *Journal of Medicinal Chemistry*, 2020, **63**, 8683–8694.

[261] G. Pesciullesi, P. Schwaller, T. Laino and J.-L. Reymond, *Nature Communications*, 2020, **11**, 1.

[262] D. A. Rufa, H. E. Bruce Macdonald, J. Fass, M. Wieder, P. B. Grinaway, A. E. Roitberg, O. Isayev and J. D. Chodera, *bioRxiv*, 2020.

[263] L. Takács, *Journal of Combinatorial Theory, Series A*, 1990, **53**, 321–323.

[264] B. Knapp, L. Ospina and C. M. Deane, *Journal of Chemical Theory and Computation*, 2018, **14**, 6127–6138.

[265] A. Grossfield, P. N. Patrone, D. R. Roe, A. J. Schultz, D. Siderius and D. M. Zuckerman, *Living Journal of Computational Molecular Science*, 2019, **1**, 1.

[266] M. Su, Q. Yang, Y. Du, G. Feng, Z. Liu, Y. Li and R. Wang, *Journal of Chemical Information and Modeling*, 2019, **59**, 895–913.

[267] D. A. Rufa, H. E. B. Macdonald, J. Fass, M. Wieder, P. B. Grinaway, A. E. Roitberg, O. Isayev and J. D. Chodera, 2020.

[268] W. Jespers, M. Esguerra, J. Åqvist and H. G. de Terán, *Journal of Cheminformatics*, 2019, **11**, year.

[269] P. Bauer, B. Hess and E. Lindahl, 2022.

[270] J. Zou, C. Tian and C. Simmerling, *Journal of Computer-Aided Molecular Design*, 2019, **33**, 1021–1029.

[271] C. Shyu and F. M. Ytreberg, *Journal of Computational Chemistry*, 2009, **30**, 2297–2304.

[272] M. Jorge, N. M. Garrido, A. J. Queimada, I. G. Economou and E. A. Macedo, *Journal of Chemical Theory and Computation*, 2010, **6**, 1018–1027.

[273] C. H. Bennett, *Journal of Computational Physics*, 1976, **22**, 245–268.

[274] P.-C. Su and M. E. Johnson, *Journal of Computational Chemistry*, 2015, **37**, 836–847.

[275] W. L. Jorgensen and L. L. Thomas, *Journal of Chemical Theory and Computation*, 2008, **4**, 869–876.

[276] S. Bruckner and S. Boresch, *Journal of Computational Chemistry*, 2010, **32**, 1303–1319.

[277] T. Steinbrecher, I. Joung and D. A. Case, *Journal of Computational Chemistry*, 2011, **32**, 3253–3263.

[278] J. W. Pitera and W. F. van Gunsteren, *The Journal of Physical Chemistry B*, 2001, **105**, 11264–11274.

[279] C. E. M. Schindler, H. Baumann, A. Blum, D. Böse, H.-P. Buchstaller, L. Burgdorf, D. Cappel, E. Chekler, P. Czodrowski, D. Dorsch, M. K. I.

Eguida, B. Follows, T. Fuchß, U. Grädler, J. Gunera, T. Johnson, C. Jorand Lebrun, S. Karra, M. Klein, T. Knehans, L. Koetzner, M. Krier, M. Leiendecker, B. Leuthner, L. Li, I. Mochalkin, D. Musil, C. Neagu, F. Rippmann, K. Schiemann, R. Schulz, T. Steinbrecher, E.-M. Tanzer, A. Unzue Lopez, A. Viacava Follis, A. Wegener and D. Kuhn, *Journal of Chemical Information and Modeling*, 2020, **60**, 5457–5474.

[280] D. Cui, B. W. Zhang, Z. Tan and R. M. Levy, *Journal of Chemical Theory and Computation*, 2020, **16**, 67–79.