# Protein-ligand binding affinity prediction using combined molecular dynamics simulations and deep learning algorithms

Pierre-Yves Libouban

# UNIVERSITÉ D'ORLÉANS

***ÉCOLE DOCTORALE***
***SANTE, SCIENCES BIOLOGIQUES ET CHIMIE DU VIVANT***
**Institut de Chimie Organique et Analytique**

# THÈSE

présentée par :

**Pierre-Yves Libouban**

soutenue le : **11 décembre 2023**

pour obtenir le grade de : **Docteur de l'Université d'Orléans**

Discipline / Spécialité : Chimie / Chémoinformatique

# Protein-Ligand binding affinity prediction using combined molecular dynamics simulations and deep learning algorithms

**THÈSE dirigée par :**
**M. BONNET Pascal** — Professeur, Université d'Orléans

**RAPPORTEURS :**
**M. SPERANDIO Olivier** — Chargé de recherche, Institut Pasteur
**M. MONTES Matthieu** — Professeur, CNAM – Paris

---

**AUTRES MEMBRES DU JURY :**
**Mme SOPKOVA Jana** — Professeure, Université de Caen Normandie, **Présidente du jury**
**M. BARRIL Xavier** — Professeur, Université de Barcelone
**M. TRESADERN Gary** — Directeur scientifique, Janssen
**Mme ACI-SECHE Samia** — Chargée de recherche, CNRS Orléans

# UNIVERSITÉ D'ORLÉANS

***ÉCOLE DOCTORALE***
***SANTE, SCIENCES BIOLOGIQUES ET CHIMIE DU VIVANT***
**Institut de Chimie Organique et Analytique**

# THÈSE

présentée par :

**Pierre-Yves Libouban**

soutenue le : **11 décembre 2023**

pour obtenir le grade de : **Docteur de l'Université d'Orléans**

Discipline / Spécialité : Chimie / Chémoinformatique

# Protein-Ligand binding affinity prediction using combined molecular dynamics simulations and deep learning algorithms

**THÈSE dirigée par :**

| | |
|---|---|
| **M. BONNET Pascal** | Professeur, Université d'Orléans |

**RAPPORTEURS :**

| | |
|---|---|
| **M. SPERANDIO Olivier** | Chargé de recherche, Institut Pasteur |
| **M. MONTES Matthieu** | Professeur, CNAM – Paris |

---

**AUTRES MEMBRES DU JURY :**

| | |
|---|---|
| **Mme SOPKOVA Jana** | Professeure, Université de Caen Normandie, **Présidente du jury** |
| **M. BARRIL Xavier** | Professeur, Université de Barcelone |
| **M. TRESADERN Gary** | Directeur scientifique, Janssen |
| **Mme ACI-SECHE Samia** | Chargée de recherche, CNRS Orléans |

# Table of contents

# Acknowledgements

Over the course of four years, my understanding of the field has significantly deepened through interactions with colleagues and mentors. This challenging project has been a tremendous learning experience, and I believe it has equipped me to tackle future research challenges with confidence.

First of all, I would like to express my gratitude to the members of the Thesis Jury for agreeing to evaluate my PhD work. I hope you find my thesis insightful and engaging. Special thanks to Dr Olivier Sperandio and Pr Mathieu Montes for serving as the primary reviewers, as well as Pr Jana Sopkova, Pr Xavier Barril, Dr Gary Tresadern, Dr Samia Aci-Sèche and Pr Pascal Bonnet. Thank you for making yourself available for my PhD defence.

I am immensely thankful to Janssen for funding my PhD. This support allowed me the freedom to explore my research topic and ensured that I had access to ample resources throughout my doctoral journey. Additionally, it provided opportunities for me to attend numerous conferences and further enrich my academic experience. I would like to express my gratitude for financing my fourth year of PhD, which allowed me to achieve the goals I had set for myself during this PhD. I would like also to thanks the CRIANN and the GENCI for providing me with ample computational resources.

A special mention goes to IDRIS for their invaluable support, including deep learning training and daily assistance. I would like to acknowledge the exceptional contributions of the IDRIS advanced support team, particularly Camille Parisel and Maxime Song, who implemented the neural networks crucial to my research. Your support was indispensable in advancing my research to new heights, and I would not have been able to achieve this level of progress without your assistance. I would like to thank you for assisting me even after the end of the support, especially in the implementation of the Gaussian distribution. It has been a pleasure collaborating with you.

I am grateful to the ICOA directors, Pr Pascal Bonnet and Pr Sylvain Routier, for allowing me to work in their laboratory. A special thanks to Pr Pascal Bonnet for accepting me into ICOA and his research team in 2019.

Pascal, your guidance throughout my PhD was invaluable. I appreciated your corrections of my thesis, publications, and various presentations. Despite your busy schedule as the university's research vice-president, you were always available, especially during critical times of my thesis. Your encouragement and flexibility in allowing me to manage my PhD, such as attending numerous conferences, were greatly appreciated. Thank you for organizing team events and sharing those enjoyable moments together.

Samia, I want to express my gratitude for your daily management and your patience in addressing my numerous questions. You were consistently responsive to urgent matters and provided me with excellent guidance, particularly in the molecular dynamics simulation aspect. I greatly appreciated your thorough corrections of my work. It was a pleasure working with you.

Gary, thank you for your guidance. I highly value your diligence and timely corrections of my papers. Our discussions and your insights, particularly on the issue of bias in datasets, have been extremely helpful. I am delighted to continue as a postdoc with Janssen and be a part of your team.

Christophe, I want to express my gratitude for your guidance during my first year of PhD. Although our collaboration could not continue, your advice laid the foundation for this project and has carried me to this point.

Jose Carlos, thank you for your valuable feedback on the deep learning aspect of my research. I greatly appreciated your guidance on how to assess my neural networks' performance and your assistance when needed. I look forward to working with you in Beerse.

Stéphane, thank you for always being available to answer my questions. Your keen eye for catching errors in my presentations has been incredibly helpful.

Gautier, your presence in the lab during the first 2 years of my PhD made it so much more enjoyable. I fondly remember those late working days with music in the lab and our laid-back board game nights together. As the song goes "It's Friday then"… I hope we can share more of those Tabasco shots in the future!

XiaoJun, thank you for the nice moments we shared. I am grateful for our discussions on research-oriented topics and for our mutual language exchange. Our GeoGuessr, dart, and chess sessions were a lot of fun. I trust that you will continue watering the aquarium, and I also hope that "King Croc" (the chark) will continue to watch over the lab for the years to come. Good luck on your last year of PhD!

Jérémy, thank you for bringing energy and liveliness to the lab. Your transition from analytical chemistry to chemoinformatics was impressive, and you did a fantastic job. I hope that you will rehabilitate from your cake betting habits. Laurent, thank you for your presence and your timely solutions to our informatics issues. I thoroughly enjoyed listening to your stories and hope that you will soon reconcile with Brittany and its people. Pascal K., thanks for enlivening our lunches with your engaging debates.

I extend my thoughts to all the postdocs who have been part of our team, including Nicolas, Lydia, Mahesh, and Samdani. A special thank you goes to Adnane for the enjoyable moments we shared together. I would also like to express my gratitude to Lucas for his dedicated work on the project, particularly his contributions to loop modelling.

In addition, I want to acknowledge and thank all the other members of ICOA for the wonderful times we had together, with a special mention of Angelo. I also wish to express my thanks to all the ADSO members with whom we organized numerous memorable parties, and to the Orléans team for those unforgettable Colombian nights.

I also express my gratitude to all my friends from the PACES team, whom I have the pleasure of meeting every so often. I would like to extend my thanks to my former flatmates, Lucile, Morgane, and Eloïse, as well as to Thibaud and Aline, who have visited me in Orléans. Thanks to all of you, for your help and support throughout the years.

Nastassja, thank you for brightening my life over the past two years. Your unwavering support, especially during this final thesis-writing period, has meant a lot to me. I will miss our "lait-caramel" breaks and eagerly anticipate expanding our list of the best restaurants we went to. Saetta McQueen Kachow!

I would like to thank my brothers, David and Julien, as well as their wives, Elodie and Hortense. You have consistently stood by me and offered assistance whenever I needed it. I am looking forward to visit you more frequently in the future, and spend more time with my nephews and nieces.

Finally, I would like to express my deepest gratitude to my parents. Mom and Dad, I am truly thankful for all the support you have provided me and for allowing me to pursue the studies I wished for. I cherished the time spent during these two lockdowns at your place in Brittany, where I was able to write a part of this thesis. From the bottom of my heart, thank you.

# Abbreviations

| | |
|---|---|
| 0D, 1D | No or one dimension |
| 2D, 3D, 4D | Two, three, four dimensions |
| AAE | Adversarial autoencoder |
| ABFE | Absolute binding free energy |
| ADP | Adenosine diphosphate |
| AI | Artificial intelligence |
| AM1-BCC | Austin model 1 – bond charge correction |
| APIF | Atom-pairs-based interaction fingerprints |
| ATP | Adenosine triphosphate |
| AUC | Area under the curve |
| ANN | Artificial neural network |
| BERT | Bidirectional encoder representations from transformers |
| BLAST | Basic local alignment search tool |
| CADD | Computer-aided drug design |
| CART | Classification and regression trees |
| CASF | Comparative assessment of scoring function |
| CASP | Critical assessment of structure prediction |
| CCHTS | Combinatorial chemistry and high-throughput screening |
| CoG | Center of geometry |
| ConvLSTM | Convolutional long short-term memory |
| CSFP | Connected subgraph fingerprints |
| CNN | Convolutional neural network |
| CPU | Central processing unit |
| Cryo-EM | Cryogenic electron microscopy |
| CSAR | Community structure-activity resource |
| CUDA | Compute unified device architecture |
| DA | Data augmentation |
| DFT | Density functional theory |
| DL | Deep learning |
| DNA | Deoxyribonucleic acid |
| DNN | Deep neural network |
| DTA | Drug-target |
| DUD | Directory of useful decoys |
| $EC_{50}$ | Half maximal effective concentration |
| ECFP | Extended-connectivity fingerprints |
| ECIF | Extended connectivity interaction features |
| EF | Enrichment factor |
| FC | Fully connected |
| FCFP | Functional-connectivity fingerprints |
| FCN | Fully connected network |
| FCSP3 | Fraction of SP3 hybridized Carbon |
| FEP | Free energy perturbation |
| FPR | False positive rate |
| GAFF | Generalized amber force field |

| | |
|---|---|
| GAMD | Gaussian accelerated molecular dynamics |
| GAN | Generative adversarial network |
| GAT | Graph attention network |
| GCN | Graph convolutional network |
| GD | Gaussian distribution |
| GDP | Guanosine diphosphate |
| GeLU | Gaussian error linear unit |
| GNB | Gaussian naive bayes |
| GNN | Graph neural network |
| GPU | Graphics processing unit |
| GRU | Gated recurrent unit |
| GT | Graph transformer |
| GTM | Generative topographic mapping |
| GTP | Guanosine triphosphate |
| HBA | Hydrogen bond acceptor |
| HBD | Hydrogen bond donor |
| HDF5 | Hierarchical data format version 5 |
| HPC | High performance computing |
| HPLC | High-performance liquid chromatography |
| HTVS | High throughput virtual screening |
| $IC_{50}$ | Half maximal inhibitory concentration |
| ICOA | Institute of organic and analytical chemistry |
| IHC | Immunohistochemistry |
| KIBA | Kinase inhibitor bioactivity |
| LADS | Latent actives in the decoy set |
| LBDD | Ligand-based drug design |
| $LD_{50}$ | Median lethal dose |
| LIE | Linear interaction energy |
| LLM | Large language model |
| LogP | Lipophilicity |
| LR | Learning rate |
| LRCN | Long-term recurrent convolutional network |
| LRP | Layer-wise relevance propagation |
| LSTM | Long short-term memory |
| MACCS | Molecular access system |
| MAE | Mean absolute error |
| MD | Molecular dynamics |
| MDFP | Molecular dynamics fingerprints |
| MET | Methionine |
| ML | Machine learning |
| MLP | Multilayer perceptron |
| ML-SF | Machine learning scoring function |
| MM-TCNN | Multi-task multichannel topological convolutional neural network |
| MMPB(GB)SA | Molecular mechanics Poisson-Boltzmann (generalized born) surface area |
| MOAD | Mother of all database |
| MPNN | Message passing neural network |
| MRI | Magnetic resonance imaging |
| MSE | Seleno-methionine |
| MUV | Maximum unbiased validation |
| MW | Molecular weight |
| NAR | Number of aromatic rings |

| | |
|---|---|
| NLP | Natural language processing |
| NME | New molecular entity |
| NMR | Nuclear magnetic resonance |
| NN | Neural network |
| NPT | Constant number, pressure and temperature |
| NTB | Number of true binders |
| NVT | Constant number, volume and temperature |
| OOM | Out of memory |
| OS | Operating system |
| ρ | Spearman's rank correlation coefficient |
| PCA | Principal component analysis |
| PCM | Proteochemometry |
| PD | Pharmacodynamics |
| PDB | Protein data bank |
| PK | Pharmacokinetics |
| PLEC | Protein-ligand extended connectivity fingerprints |
| PMD | Property-matched decoys |
| PMEMD | Particle mesh Ewald molecular dynamics |
| PMI | Principal moments of inertia |
| PN | Protein-nonpeptide |
| PP | Protein-peptide |
| pyPLIF | Python-based protein-ligand interaction fingerprints |
| QM | Quantum mechanics |
| QM/MM | Quantum mechanics / molecular mechanics |
| QSAR | Quantitative structure activity relationship |
| QSPR | Quantitative structure property relationship |
| RAM | Random-access memory |
| RBFE | Relative binding free energy |
| RELU | Rectified linear unit |
| RF | Random forests |
| rgyr | Radius of gyration |
| RMSD | Root mean square deviation |
| RMSE | Root mean square error |
| RNA | Ribonucleic acid |
| RNN | Recurrent neural network |
| ROC | Receiver operating characteristic |
| $R_P$ | Pearson's correlation coefficient |
| SASA | Solvent accessible surface area |
| SAscore | Synthetic accessibility score |
| SBDD | Structure-based drug design |
| SCP | Secure copy protocol |
| SD | Standard deviation |
| SF | Scoring function |
| SIFt | Structural interaction fingerprints |
| SiLU | Sigmoid linear unit |
| SMD | Steered molecular dynamics |
| SOM | Self-organizing map |
| SPR | Surface plasmon resonance |
| SSH | Secure shell |
| SVM | Support vector machines |
| SPLIF | Structural protein–ligand interaction fingerprints |
| TI | Thermodynamic Integration |

| | | | |
|---|---|---|---|
| TIP3P | Transferable intermolecular potential with 3 points | UMAP | uniform manifold approximation and projection |
| TMD | Targeted molecular dynamics | VAE | Variational autoencoder |
| TOP | Total operating characteristic | ViT | Vision transformer |
| TPR | True positive rate | VMD | Visual molecular dynamics |
| TPSA | Topological polar surface area | VS | Virtual screening |
| t-SNE | t-distributed stochastic neighbour embedding | XGB | Extreme gradient boosting |

# List of figures

# List of tables



# List of equations

# 1 Introduction

At the beginning of each chapter, a summary has been written in French. Hence, the following part is a summary in French of both the introduction and the computer aided drug design chapter.

Cette thèse s'inscrit dans le cadre du développement de médicaments, un processus long et coûteux qui peut s'étendre sur 15 à 20 ans et nécessiter en moyenne entre 1 et 2 milliards d'euros. Ce processus est constitué de multiples étapes, depuis la découverte de candidats médicamenteux jusqu'à leur évaluation *in vitro* puis *in vivo*. Les coûts augmentent progressivement au fil des étapes, atteignant leur apogée lors des essais *in vivo* sur l'homme, qui sont particulièrement onéreux. Il est donc essentiel de rationnaliser très en amont la sélection des molécules qui passeront aux étapes suivantes.

L'informatique joue un rôle de plus en plus crucial dans la réduction de ces coûts, devenant ainsi un outil indispensable à de multiples étapes du processus de découverte d'un médicament. Par exemple, l'informatique est utilisée pour identifier de nouvelles cibles thérapeutiques, sélectionner les molécules d'intérêt et évaluer les caractéristiques pharmacocinétiques des composés (c'est-à-dire, le devenir d'une molécule dans l'organisme) avant d'entreprendre les tests *in vivo* sur les animaux et les humains.

Mes recherches font partie intégrante de la thématique « découverte de médicaments assistée par ordinateur » (CADD), visant à identifier des molécules d'intérêt à l'aide d'outils informatiques. L'objectif est de guider la synthèse de nouvelles molécules en sélectionnant préalablement les composés les plus prometteurs, ceux ayant le plus de potentiel pour devenir des médicaments. En règle générale, un médicament est une petite molécule, également appelé « ligand », qui interagit avec une cible thérapeutique, souvent une protéine. Cette interaction conduit à la formation d'un complexe protéine-ligand, entraînant soit l'activation, soit l'inhibition de la protéine, ce qui permet de traiter la pathologie sous-jacente.

La sélection des molécules d'intérêt implique plusieurs phases. Initialement, un processus d'identification de touches (hits) est entrepris, souvent en utilisant une méthode de criblage à haut débit (HTS). Ensuite, vient l'étape d'optimisation des touches en têtes de série, qui sont des molécules présentant une forte affinité pour la cible thérapeutique. Enfin, la dernière phase consiste à optimiser les têtes de série pour diverses propriétés pharmacologiques telles que leur solubilité, leur capacité à franchir les barrières biologiques ou leur toxicité.

Tout au long de cette phase de sélection, il est essentiel de choisir les molécules à synthétiser en évaluant leurs propriétés moléculaires. L'une des propriétés les plus critiques est l'affinité du complexe protéine-ligand, qui représente la capacité des molécules à se lier à leur cible thérapeutique. Par principe, nous postulons que les molécules présentant une affinité élevée pour leur cible, auront une forte activité sur celle-ci, et donc l'effet thérapeutique de ces molécules sera important.

De nombreuses méthodologies, y compris des modèles d'intelligence artificielle (IA), ont été développées pour évaluer l'affinité des complexes moléculaires. L'intégration croissante de l'IA dans le domaine du CADD est motivée par plusieurs avancées majeures, telles que celle d'AlphaFold (1), qui ont replacé l'IA au premier plan. Ces algorithmes d'IA reposent sur le principe de l'apprentissage automatique, permettant d'entrainer des modèles à partir de données préalablement collectées.

Ainsi, il est possible d'entrainer des modèles statistiques à partir des résultats expérimentaux d'activité obtenus en testant *in vitro* l'affinité de molécules déjà synthétisées, par exemple à partir des données d'un criblage à haut débit (HTS). Ces modèles, appelés relation quantitative structure activité (QSAR), permettent de prédire l'affinité des molécules pour une protéine cible spécifique en fonction de leurs structures moléculaires. Ils peuvent ainsi établir des corrélations entre certaines caractéristiques moléculaires et l'affinité des complexes, de manière semblable à la détection de pharmacophores au sein de molécules. D'autres modèles peuvent également être développés pour prédire l'affinité des molécules pour plusieurs cibles, tels que la protéochémométrie ou les outils « interaction médicament-cible » (DTI). Dans ce cas, les modèles sont entraînés sur la structure des molécules ainsi que sur les séquences d'acides aminés des protéines.

Alternativement les méthodes dites « basées sur la structure » sont de plus en plus utilisées pour prédire l'affinité. Cette tendance est favorisée par la disponibilité croissante des structures protéiques, en partie grâce à AlphaFold (1), ainsi que par l'augmentation du nombre de structures résolues de haute qualité, facilitée par les progrès des techniques expérimentales telles que la cristallographie et la cryomicroscopie électronique. Ces outils de prédiction de l'affinité analysent la conformation 3D des complexes protéine-ligand ainsi que les interactions entre les deux partenaires. Par exemple, les poses des complexes obtenues par amarrage moléculaire (également appelé *docking*) peuvent être évaluées à l'aide de fonctions de score. Il existe différents types de fonctions de score ; certaines sont basées sur des champs de force, tandis que d'autres sont des modèles statistiques obtenus par apprentissage automatique.

Ces fonctions de score peuvent également être appliquées directement aux structures de complexes protéine-ligand obtenues expérimentalement par cristallographie, résonance magnétique nucléaire (RMN) ou cryomicroscopie électronique. Ainsi, deux types de modèles statistiques peuvent être développés : les modèles locaux et les modèles globaux. Les modèles locaux sont entraînés à partir de complexes provenant d'une même protéine, ce qui limite la quantité de données disponibles. Bien que ces modèles puissent obtenir de bonnes performances, ils ne sont pas applicables pour prédire l'affinité des complexes impliquant d'autres protéines. En revanche, les modèles globaux sont entraînés sur divers complexes protéine-ligand, ce qui leur permet de généraliser plus facilement. On attend de ces modèles qu'ils apprennent à prédire l'affinité en analysant des informations constantes entre tous ces complexes, telles que les interactions protéine-ligand.

La PDBbind (2) recueille les structures de complexes protéine-ligand déterminées expérimentalement, accompagnées des données d'affinité obtenues grâce à des tests in vitro. Dans sa dernière version (v.2020), ce jeu de données comprend 19 000 structures. Il existe plusieurs représentations informatiques de la structure des complexes protéine-ligand. La représentation la plus brute est celle des nuages de points, où chaque atome est représenté par un point dans l'espace, sans liaison entre eux. Une autre représentation consiste en des graphes, où les interactions (covalentes ou non) sont représentées par des arêtes et les atomes par des nœuds. De plus, la surface des complexes peut être représentée à l'aide de mailles. Enfin, une grille peut être utilisée comme représentation, où l'espace est discrétisé en voxels, généralement de dimensions 1×1×1 Å. Chaque voxel correspond à une unité élémentaire de l'espace, ce qui signifie qu'un seul atome peut être assigné par voxel.

Il est possible d'entraîner des modèles statistiques globaux à partir des structures 3D disponibles dans la PDBbind, afin de prédire l'affinité des complexes protéine-ligand. De nombreuses méthodes d'apprentissage automatique ont ainsi été développés, mais nous nous concentrerons sur les réseaux de neurones, qui représentent l'une des méthodes de pointe en intelligence artificielle. Ces réseaux sont composés de couches de neurones artificiels interconnectés, reproduisant ainsi le fonctionnement du cerveau humain, notamment sa plasticité. Grâce à l'apprentissage profond, ces

réseaux, constitués de plusieurs couches de neurones, peuvent atteindre des performances optimales. Cependant, leur principal inconvénient réside dans le manque d'explication des prédictions effectuées, phénomène connu sous le nom d'effet boîte noire.

Principalement deux types de réseaux de neurones ont été utilisés pour analyser les structures des complexes et entraîner les modèles statistiques : les réseaux de neurones à convolution (CNN) et les réseaux de neurones basés sur les graphes (GNN). Les 3D CNN, tels que Pafnucy (3) ou $K_{DEEP}$ (4), traitent les complexes de manière brute, sans prétraitement de l'information, en les considérant comme des images 3D. À l'aide de noyaux de convolution parcourant les voxels de l'image 3D, ces réseaux extraient localement l'information spatiale pour prédire l'affinité des complexes. Le fait que ces réseaux soient appliqués à l'information brute, sans recourir à des descripteurs élaborés par des experts, permet d'éviter d'introduire des biais humains dans le processus de prédiction. Il existe également des 2D CNN, tels que OnionNet (5), qui ont été utilisés pour prédire l'affinité à partir des structures. Dans ce cas, les informations ont été prétraitées en extrayant les contacts protéine-ligand jusqu'à une distance de 30 Å, de manière concentrique à partir de chaque atome du ligand.

Quant aux GNN, ils sont idéalement adaptés pour être appliqués aux molécules, car les molécules et leurs interactions peuvent être facilement représentées sous forme de graphes. Cela permet de réduire la charge de calcul nécessaire pour entraîner les modèles statistiques par rapport aux CNN. Récemment, un grand nombre de GNN ont été publiés, tels que GraphBAR (6) ou PIGNet (7). Cependant, les interactions entre le ligand et la protéine doivent être définies explicitement, contrairement aux CNN, ce qui peut potentiellement introduire des biais humains. Dans PIGNet, les calculs réalisés sur les arêtes des graphes prennent en compte les équations des interactions non-covalentes. Ainsi, l'apprentissage du modèle est guidé par sa connaissance des interactions de van der Waals, des interactions hydrophobes, des liaisons hydrogènes ainsi que des interactions métal-ligand. De plus, ce réseau de neurones tient également compte des effets entropiques.

Pour évaluer les performances des fonctions de score, un jeu de données d'évaluation appelé *Comparative Assessment of Scoring Function* (CASF) (8) a été élaboré. La version de 2016 comprend 285 complexes, provenant de 57 groupes de 5 complexes chacun. Ces groupes ont été constitués sur la base d'une homologie de séquence de 90 %, chaque groupe correspondant à une protéine. Les structures de ces complexes ainsi que leur affinité ont été déterminées expérimentalement. Ainsi, il est possible de comparer les performances des différentes fonctions de score sur ce jeu de test. Les métriques utilisées pour quantifier ces performances sont le coefficient de corrélation et l'erreur quadratique moyenne.

Malgré les performances prometteuses des fonctions de score basées sur l'apprentissage automatique (ML-SF), leurs évaluations sur des jeux de données externes ont révélé leur faible capacité à généraliser. Il semble en effet que le CASF soit biaisé par rapport au jeu d'apprentissage de la PDBbind (9). Ainsi, les modèles statistiques entraînés sur la PDBbind peuvent obtenir des performances élevées sur le CASF en exploitant ces biais. Par exemple, au lieu d'apprendre sur les interactions spécifiques entre la protéine et le ligand, les ML-SF peuvent simplement apprendre que certaines molécules ont tendance à présenter une forte ou une faible affinité, sans tenir compte de la cible spécifique avec laquelle elles interagissent.

En parallèle au développement des fonctions de score, des méthodes basées sur les simulations de dynamique moléculaire ont été élaborées pour évaluer l'affinité des complexes protéine-ligand. La dynamique moléculaire constitue un outil permettant de simuler les molécules et leurs interactions au fil du temps. Ces simulations sont obtenues à l'aide de champs de force, qui permettent de calculer les

forces appliquées à chaque atome d'un système au cours du temps. Elles offrent ainsi des informations précieuses sur la thermodynamique de l'interaction et *in fine* sur l'affinité du complexe.

Ainsi, des outils basés sur les simulations de dynamique moléculaire, tels que la « *free energy perturbation* » (FEP) ou « l'intégration thermodynamique », peuvent être utilisés pour calculer l'énergie libre de liaison relative entre deux molécules similaires. Ces méthodes fournissent des résultats très précis de l'ordre de 1 kcal/mol, cependant elles demandent beaucoup de puissance de calcul et ne peuvent être appliquées qu'à certaines étapes du développement d'un médicament, comme la phase d'optimisation des têtes de série. En revanche, les méthodes « MM-GB/PBSA » et « énergie d'interaction linéaire » (LIE) sont employées pour évaluer l'énergie libre de liaison absolue, ce qui les rend applicables également lors des phases de découverte de touches et d'optimisation des composés en tête de série. Cependant, ces méthodes peinent à évaluer avec précision l'affinité de liaison des complexes protéine-ligand. Dans ce domaine, « *l'absolute binding* FEP » (AB-FEP) semble être une méthode prometteuse, bien qu'elle implique un coût computationnel élevé et nécessite davantage de développement pour être pleinement applicable dans des projets de découverte de nouvelles molécules thérapeutiques.

Face aux limitations de ces différentes méthodes, il est impératif de développer une méthode de prédiction de l'affinité de liaison des complexes protéine-ligand qui soit rapide, peu gourmande en calcul et efficace. Les modèles statistiques semblent être une solution appropriée pour atteindre ces objectifs, mais il est crucial de résoudre le problème du biais dans leurs prédictions. L'une des principales raisons pour lesquelles les modèles statistiques obtiennent des résultats mitigés est principalement due à la faible quantité de données disponibles dans notre domaine. En effet, la PDBbind ne compte que 19 000 complexes, tandis que les algorithmes d'apprentissage profond sont généralement appliqués sur des jeux de données beaucoup plus vastes, de l'ordre du million de données.

Au cours de ce doctorat, notre objectif initial était d'établir des directives méthodologiques pour utiliser les données disponibles de manière optimale, afin d'améliorer la capacité des modèles statistiques à prédire l'affinité de liaison des complexes protéine-ligand. Par la suite, nous nous sommes concentrés sur l'implémentation d'une méthode d'augmentation de données visant à enrichir les jeux de données utilisés pour entrainer les modèles statistiques et ainsi améliorer les performances de ces modèles. Pour ce faire, nous avons développé une nouvelle approche en combinant les simulations de dynamique moléculaire avec des algorithmes d'apprentissage profond. Un jeu de données comprenant 63 000 simulations de dynamique moléculaire a ainsi été constitué, à partir duquel plus de 3 000 000 d'images ont été extraites, ce qui représente une multiplication des données par 150 par rapport à la PDBbind.

Densenucy, un nouveau réseau de neurones, a été développé pour traiter efficacement les images issues des simulations de dynamique moléculaire. Les modèles statistiques entraînés avec Densenucy présentent des performances de pointe grâce à l'augmentation de données provenant des simulations. Il semble donc pertinent de mettre en place des méthodologies d'augmentation de données basées sur les simulations de dynamique moléculaire, ce qui permet d'améliorer la capacité des modèles à prédire l'affinité des complexes protéine-ligand.

Par ailleurs, nous avons développé deux réseaux de neurones novateurs, Timenucy et Videonucy, capables de prendre en charge une simulation de dynamique moléculaire dans son intégralité pour prédire l'affinité. Ces réseaux prennent en compte le lien temporel entre chaque image de la simulation, permettant ainsi une analyse spatiotemporelle inédite en prédiction de l'affinité. À notre connaissance, cette approche n'avait jamais été explorée auparavant dans ce domaine. Nous avançons

que ces réseaux de neurones sont capables d'analyser les mouvements des ligands au cours de la simulation pour mieux prédire l'affinité des complexes protéine-ligand. Au cours de ce travail, nous avons démontré la faisabilité de cette approche, mais il reste à améliorer ces réseaux pour exploiter de manière optimale les données disponibles. En parallèle, un effort supplémentaire de génération de simulations de dynamique moléculaire sera nécessaire pour exploiter au mieux le potentiel de ces réseaux de neurones. Ainsi, nous pouvons nous attendre à obtenir des performances qui repoussent les limites de ce qui est actuellement possible en termes de prédiction de l'affinité.

Ces résultats mettent en évidence le potentiel des approches combinant les simulations de dynamique moléculaire avec les algorithmes d'apprentissage profond. Nous pensons que ces méthodes bénéficieront pleinement de l'accroissement des capacités de calcul. À terme, leur application devrait permettre d'accélérer et d'optimiser le processus de découverte de médicaments, en particulier aux étapes de détection des touches, d'amélioration des touches en têtes de série et d'optimisation des têtes de série.

Il est envisageable d'évaluer les poses obtenues lors du criblage virtuel à partir de modèles statistiques entraînés grâce à l'augmentation de données issues des dynamiques moléculaires. Cette première analyse permettra de repérer des molécules d'intérêt au sein de vastes jeux de données contenant des millions de molécules. Ensuite, les modèles statistiques permettant d'analyser les simulations de manière spatiotemporelle pourront être utilisés pour sélectionner les molécules les plus prometteuses au sein d'un groupe restreint de molécules d'intérêt.

La combinaison de ces deux méthodologies offre une approche complémentaire pour progresser rapidement dans un projet de découverte de médicaments.

## 1.1 The role of informatics in healthcare

The influence of informatics, particularly in the context of a digitalized society, has become increasingly pronounced. One of the sectors significantly transformed by these advancements is healthcare. Notably, telemedicine has experienced a substantial rise and has played a vital role during the COVID-19 pandemic. The ability to predict the dynamics of the pandemic has also aided policymakers in making decisions to reduce the number of infected individuals, such as implementing lockdown measures. Additionally, the integration of AI into healthcare has gained prominence, with applications including the analysis of X-ray scans to detect bone fractures.

Expanding on the topic of diagnosis, bioinformatics thrives on the wealth of recent genomic data, thanks to advancements in genome sequencing. It all began with the sequencing of the first virus genome in 1977, followed by the sequencing of the human genome in 2003. Since then, the cost of genome sequencing has decreased by a factor of 100,000. One of the most anticipated applications of bioinformatics in healthcare is the swift diagnosis of rare genetic diseases, potentially reducing the diagnostic wandering duration, which traditionally lasts 2 years in France (10). Furthermore, genomics analysis has could potentially revolutionize cancer detection, such as with lung cancer, which could shift from microscopic cell shape-based classification (small cell or non-small cell) to the presence or absence of nearly 30 genetic mutations.

Additionally, HER2-positive breast cancer, currently identified through immunohistochemistry (IHC) testing, can also be detected via mutated HER2 oncogenes. This characterization of cancers can

also guide the selection of the most effective treatments, especially for targeted therapy. These therapies are designed for specific genetic profiles, such as HER2-positive cancer, and often require companion diagnostic tests to ensure their suitability for each patient (11). These new genomic methods of cancer identification have transformed the cancer diagnosis paradigm, treating every cancer as a rare cancer (12). Each cancer presents a distinct combination of mutations, resulting in varying behaviours, from aggressive cell reproduction and spread to evading the immune system. Tailoring treatment based on mutation identification can significantly improve the survival rates of patients.

These developments suggest a future where personalized medicine arises from genomics, also enabling customized drug dosages tailored to individual metabolic profiles. Accurately predicting one's ability to metabolize a drug is a key element to evaluate the lifetime of a drug in an organism. This is crucial when dealing with drugs with a narrow therapeutic range (toxic dosage very close to the therapeutic one), as is the case with theophylline (used for asthma treatment) or warfarin (an anti-coagulant).

Studying drug metabolism is a key aspect of pharmacokinetics (PK), which examines how drugs evolve within the body. It encompasses drug absorption, metabolism, distribution, and elimination. Conversely, pharmacodynamics (PD) focuses on how drugs affect the body, often through interactions with specific proteins. To establish a posology for a treatment, both PK and PD must be considered. This approach offers the potential for more effective treatments and improved patient outcomes.

PK/PD modelling, which are part of pharmacometrics, aim to understand the time course of the intensity of drug effects following dosage administration. These studies play a crucial role in predicting the necessary dosages for *in vivo* tests, thereby reducing the reliance on animal testing during the pre-clinical phase of drug development. Furthermore, PK/PD modelling is widely employed in predicting the bioavailability of generic drugs, eliminating the need for *in vivo* testing. Notably, pharmacokinetics is significantly affected by a drug's composition, which includes the active compound and various excipients. While generic drugs contain the same amount of the active compound as their branded counterparts, there may be differences in the type and quantity of excipients. To gain authorization for a generic drug, it is essential to compare the drug's presence in the body after administering both the branded and generic versions and ensure equivalence.

Understanding the functioning of organisms and their pathologies at the macromolecular level is essential for drug development. Structural Bioinformatics plays a crucial role in this field by analysing the 3D structures of biomolecules like proteins and nucleic acids. For instance, it has been instrumental in comprehending the mechanism of the spike protein of coronaviruses, responsible for their entry into host cells (13). This knowledge is crucial in developing drugs to treat COVID-19.

In the field of Cheminformatics, the focus is on applying informatics tools to small molecules, particularly for drug design. The tools developed in this field can predict various crucial factors, such as a molecule's toxicity, including its propensity to be a cytotoxic or mutagenic compound. Additionally, they can assess the molecule's activity and selectivity, aiding in the identification of promising molecules with minimal side effects. Furthermore, they can evaluate variables like solubility and hydrophobicity, critical for a molecule's ability to cross barriers and ensure effective distribution within the body.

Another valuable aspect of cheminformatics is the development of tools able to virtually create new molecules. This can be accomplished through diverse methods, such as database enumeration using reaction databases like Reaxys (14). Other approaches might entail fragment-based drug design, wherein high-potency fragments, typically with a molecular weight below 300 DA, are assembled to

create novel molecules. Subsequent filtering can retain synthetically accessible molecules, and in some cases, retrosynthesis strategies can be proposed for these compounds.

## 1.2 *In Silico* Drug Design

Discovering, developing, and ultimately obtaining marketing authorization for a new drug candidate is a long and challenging process that can span over a decade (15). Depending of the therapeutic area and of the type of treatment, development costs go from several hundreds of millions to few billions of dollars (16, 17). This is especially true for new molecular entities (NME) that are first-in-class drugs requiring further testing. The development cost has steadily risen over the past decades, due to increasingly stringent regulations (18). It is also attributed to the high attrition rate during phase II of the clinical stage (Figure 1).

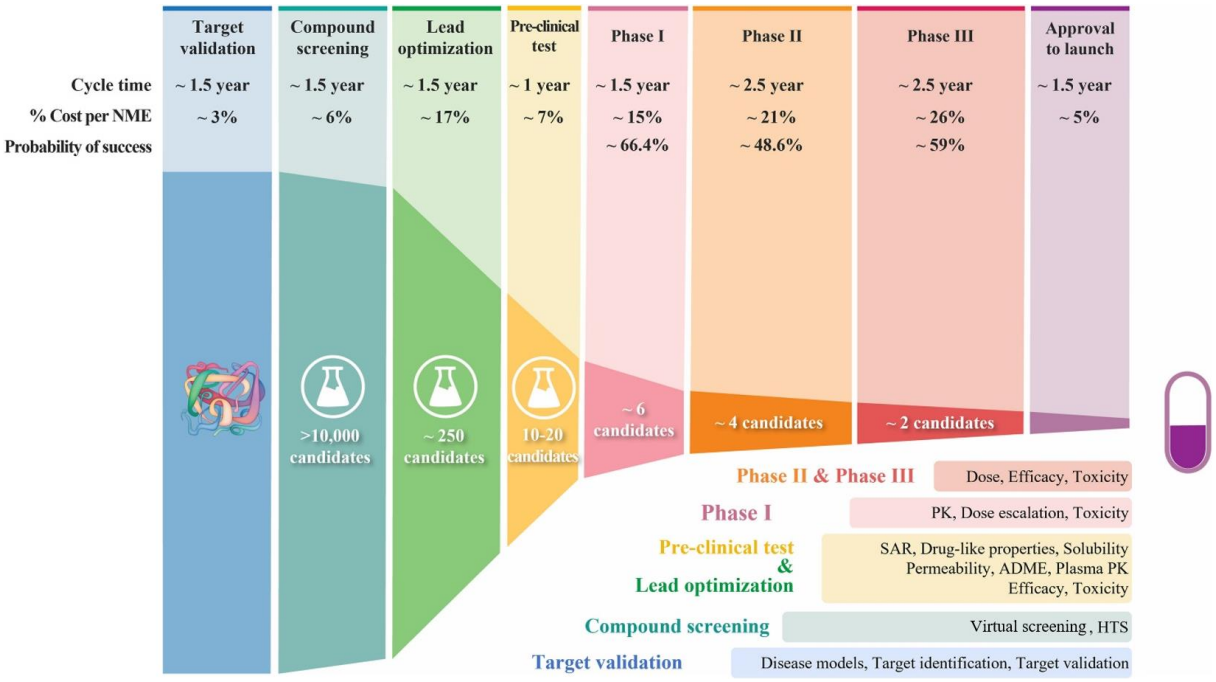



Figure 1: The process of drug discovery and development, and the failure rate at each step. Figure taken from Sun *et al*. (15)

Therefore, it is important to reduce the cost and time of the drug development process. One of the solutions is to use informatics to optimize and automate molecule selection processes. For example, with the use of artificial intelligence (AI) it is possible to predict the outcome of the tasks performed during research. The computer-aided drug design (CADD) has been focusing on guiding the selection of molecules that will be synthetized by medicinal chemists. A variety of *in silico* drug design methods were developed with this intent. The prioritization of the most promising molecules results in a reduced number of compounds synthesized and tested. This leads to several improvements for the early stages of the drug design process, such as:

- Decreased research and development time.
- Reduced global costs.
- Green chemistry through the 3R waste hierarchy (Reduce, Reuse, Recycle), especially due to the reduced consumption of solvent and chemical products (19). However, computational methods also have an energetic cost that needs to be taken into account, especially when using computationally intensive methods such as molecular dynamics simulations.

From this point onward, we will delve into the tools developed for CADD, with a particular focus on the one utilized in this research. This will lead us toward the goal of the PhD.

# 2 Computer-aided drug design

The aforementioned promises in combination with recent developments in the informatics field have sparked a surge in attention to the *in silico* tools. In the last few decades, both computational power and the amount of available data have exponentially increased. These improvements were coupled with more accessible programming languages, such as Python and R. By empowering non-informaticians, especially scientists, these languages enable them to apply informatics tools in their respective fields of study. Among other *in silico* tools, this laid the basis for the artificial intelligence (AI) environment to flourish in the realm of drug design.

The increasing appeal of AI methods, driven by their enhanced predictive capabilities, has resulted in their routine application for drug discovery purposes. In the last decade, this led to the creation of AI companies specialized in the field. To accelerate the pace of discovering new drugs, pharmaceutical companies have started to collaborate with these AI companies (Figure 2). The worldwide AI market for the pharmaceutical industry is projected to attain $4.61 billion by 2027, with a compound annual growth rate of 29.4% (20).

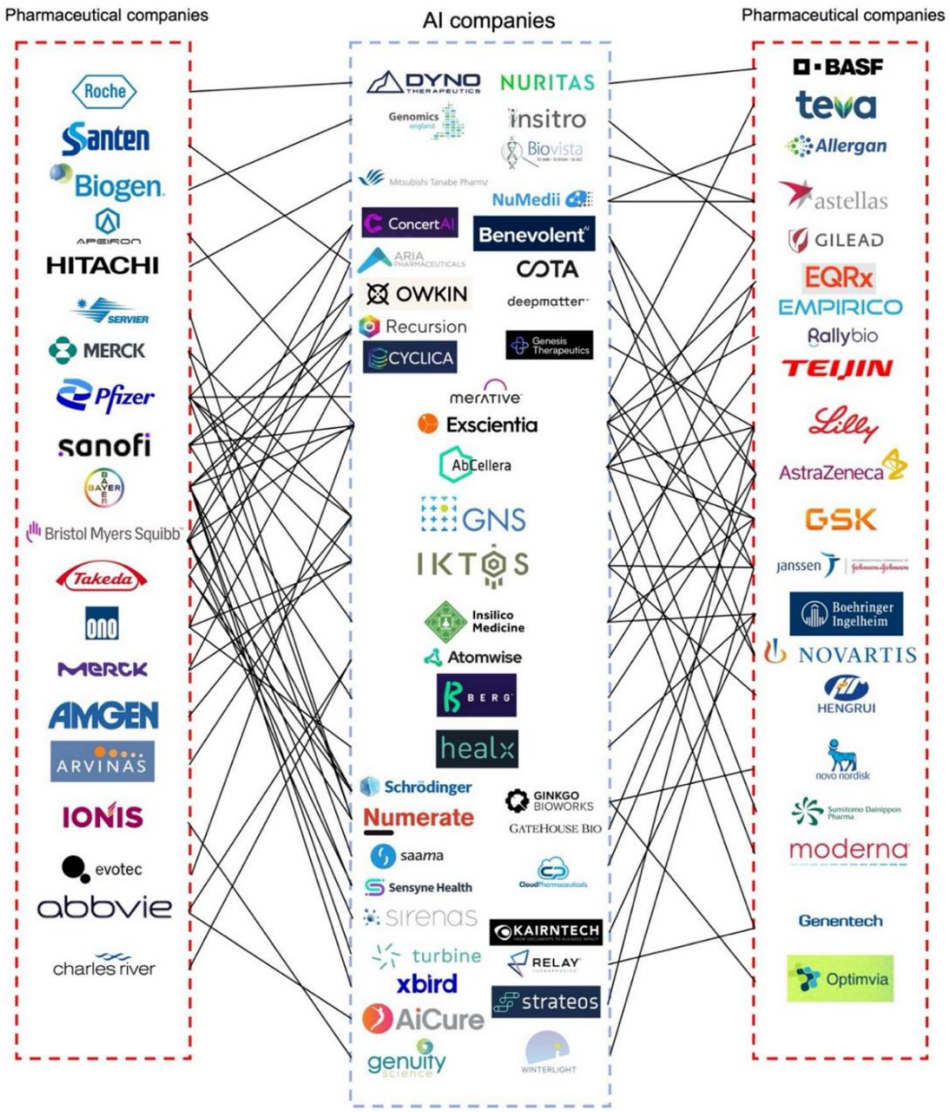



Figure 2: Partnerships between AI and pharmaceutical companies formed for drug product development. Figure taken from Jiang *et al.* (21)

Aside from AI, both the Cheminformatics and structural Bioinformatics fields have greatly benefited from these advancements. These improvements were spearheaded by the development of libraries that enable the manipulation of small molecules and macromolecules, such as RDKit (22), Open Babel (23), ChemPy (24), ChemPython (25), Scikit-chem (26) or BioPython (27). As a result, these tools facilitated the creation of in-house software, like VSPrep for the preparation of chemical databases (28, 29), and Frags2Drugs (30) that was previously developed in our team.

## 2.1 Molecular representation of protein, ligand and their interactions

Molecules need specific encoding to be easily manipulated by software. The amount of accessible information varies depending on the chosen molecular representation method. For instance, in certain

scenarios, the 3D conformations of molecules might remain undetermined or it is unnecessary, while in other cases this information is crucial. Therefore, different types of molecular representations have been developed (31, 32).

### 2.1.1 Molecular representations without 3D

The following examples are molecular representations commonly used in the field that does not provide 3D information about molecules.

In cheminformatics, the SMILES is one of the most famous molecular representation (33). The information is encoded in a linear notation made of ASCII characters. It has the advantage to be human readable. Derived from them, the SMARTS are used to encode functional groups, as well as molecules with unknown substituents (34). The SMARTS and especially the SMIRKS, which is a hybrid between SMILES and SMARTS, are among the preferred encoding to describe chemical reactions.

There has been attempts to use SMILES with generative models for *de novo* drug design purpose (35). The aim of *de novo* drug design is to computationally create new molecules. Unfortunately, these methods often lead to the creation of invalid SMILES. To attend to this problem, another similar representation of molecules, called SELFIES, was recently released (36). When they are used in *de novo* drug design, it effectively reduces the number of invalid molecules.

Alternatively, 2D graphs can be applied to represent molecules, with the graph nodes corresponding to atoms and the edges to bonds. This representation is quite intuitive for molecules, and it has been extensively applied in cheminformatics, especially with artificial intelligence (37, 38).

As for the protein, they can be represented by their amino acid sequences. This information can be obtained experimentally by performing protein sequencing. This molecular representation is extensively used in genomics to compare proteins. For example, it is possible to determine if two proteins are phylogenetically related by calculating their sequences similarity.

### 2.1.2 3D molecular representations

In some cases, it might be important to have information about the 3D conformations of molecules and proteins, especially when studying protein-ligand interactions. To this end, experimental methods were developed to obtain the structure of the complexes, such as X-ray crystallography, nuclear magnetic resonance (NMR) or cryogenic electron microscopy (cryo-EM). The structures are usually stored into the protein data bank (PDB) (39), accessible on the rcsb website (https://www.rcsb.org/).

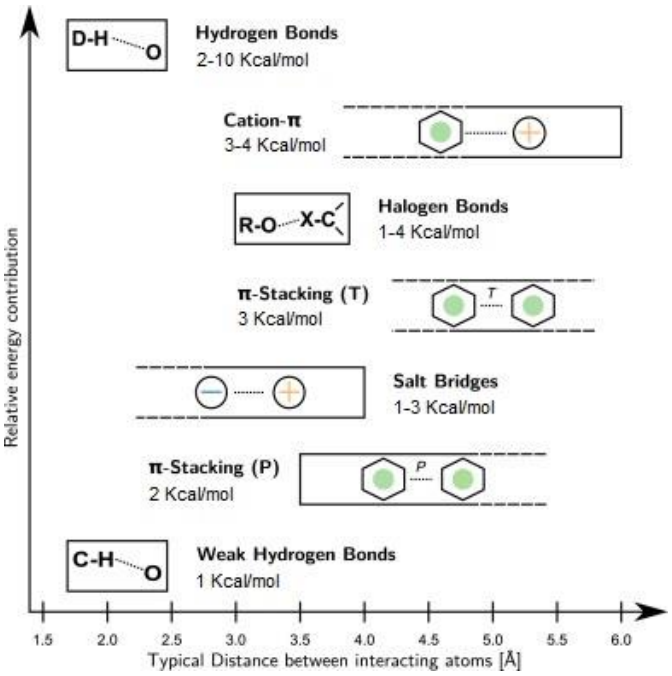



Figure 3: Protein-ligand non-covalent interactions along with their distances and energy contribution. Figure adapted from Salentin *et al.* (40)

The protein-ligand interactions can be assessed by analysing the 3D structure of complexes. Multiple interactions exist between the two partners, each characterized by distinct strengths and ranges of influence (Figure 3). Excluding covalent bonds, hydrogen bonds are the most potent, albeit with a limited range. *π*-stacking display lower bond energy but higher bond distance. Finally, hydrophobic and van der Waals interactions are weak and exert at short distance. Nonetheless, as shown in Figure 4, they are also the most common, and their cumulative influence significantly contributes to the overall affinity of the ligand for the protein. Less common interactions, like electrostatic forces, can also play a key role in increasing the ligand binding affinity.

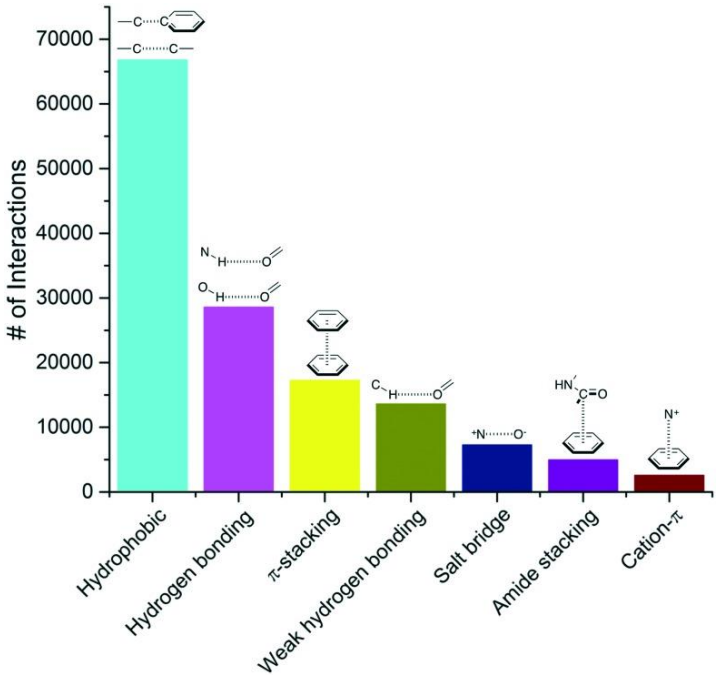



Figure 4: Frequency distribution of the most common non-covalent interactions observed in protein–ligands from the PDB (2017). Figure taken from Ferreira de Freitas *et al.* (41)

Additionally, when the structure of a complex has not been determined experimentally, but the 3D structure of the protein is available, molecular docking tools like AutoDock Vina (42) or DOCK (43) can be employed to obtain models of the complex.

It is also possible to create 3D models of proteins through homology modeling with software like MODELLER (44). Recently, Alphafold2 (1) achieved ground breaking results by producing high quality models. Other tools, such as RosettaFold (45) and Alphafill (46), are complementing the *de novo* modelling field by providing additional models of proteins and complexes.

The structures of proteins and ligands can be saved in several files, including PDB, mmCIF, mol2 and sdf. In some of these files, bonds are explicitly identified, while other files only contain the 3D atomic coordinates. It is possible to visualize such structures with software, such as PyMol (47), Chimera (48) and VMD (49).

There are several 3D representations of these macromolecular structures, either 3D structures or 3D surfaces. These representations can be implemented in 4 different manners (Figure 5):

- Point clouds: It is a basic depiction of macromolecular structures. They are composed solely of nodes without edges. The most common implementation assigns a node to each atom. Each point is allocated 3D coordinates and can be further characterized with atomic features. They are well-suited for tools that are applied directly on structure coordinates (50).
- 3D Graph: In comparison to point clouds, they are also based on nodes but additionally incorporate edges. However, contrary to 2D graphs, the edges are usually characterized by the distance between nodes. Interactions between both partners are explicitly specified, by creating specific edges between the interacting nodes. These edges can be further characterized by the type of interactions, for example hydrogen bonds, π-stacking… This representation has been extensively used with graph neural networks (GNN) (7, 9, 51).
- Mesh: In other cases, tools use the surface of the proteins (52). Meshes are well-suited to render 3D surfaces. Most of the time, the molecular representation is composed of polygons that describe the 3D arrangement of the mesh coordinates.
- 3D grid: In this case, the 3D structures are discretized with voxels over a grid. Usually, the voxels measure around 1x1x1 Å to closely map atomic resolution. They can also be applied to depict 3D surfaces. This representation is typically used with convolutional neural networks (CNN) (3, 4).

Sometimes, the lines can be blurred between these representations. For example in OctSurf (53), the complexes are represented as a surface with point clouds, before being voxelized in a grid representation.

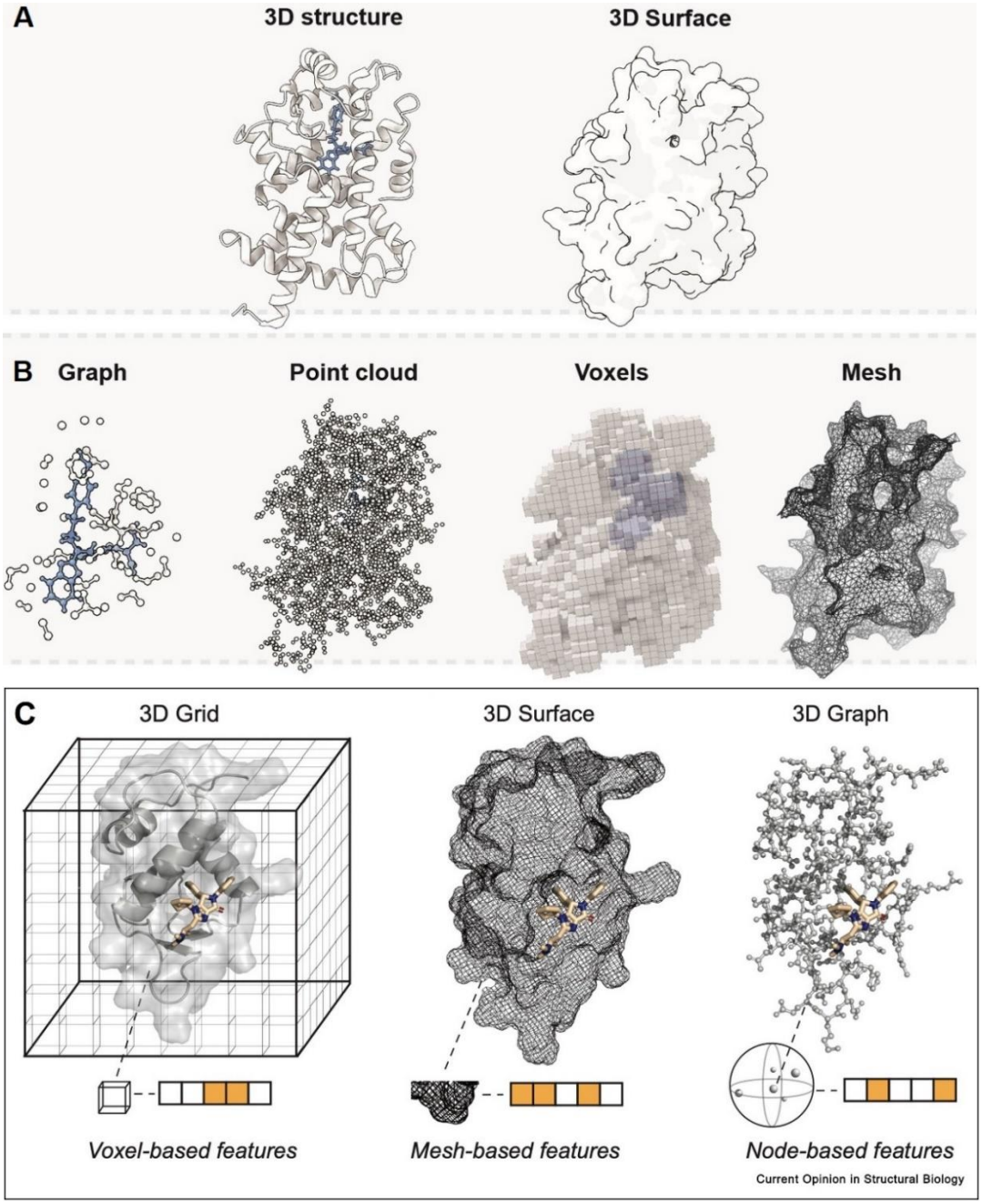



Figure 5: Structural representation of proteins. A – 3D representations of macromolecular structures. B – The four representation methods. C – Detailed encoding of the representations. Figure adapted from Özçelik *et al*. (54) and Isert *et al*. (55)

### 2.1.3 Descriptors

Since it is possible to computationally manipulate molecules, one of the primary applications would be to compare them. For example, molecules can be clustered, or their characteristics could be statistically analyzed. Therefore, it is necessary to describe the molecules with numerical values, also called molecular descriptors.

### *2.1.3.1 Ligands*

Various types of descriptors can be used to characterize ligands (Figure 6). Basic descriptors can be obtained solely from the ligands' atomic composition. They are called 0D descriptors, and here are some examples:

- The number the atoms
- The molecular weight

To calculate 1D descriptors, it is required to know the connectivity between atoms. They can be calculated from linear notations, such as the SMILES. These descriptors offer additional details regarding the molecules, such as the functional groups. For example, we can mention a couple of descriptors:

- The number of aromatic rings (NAR)
- The number of hydrogen bond donors or acceptors (HBD / HBA)

2D descriptors are obtained from molecular graphs, providing topological and graph invariant information, such as:

- The topological (or total) polar surface area (TPSA)
- The Wiener index

As for the 3D descriptors, they provide further information about ligands' shape and volume, including:

- The radius of gyration (rgyr)
- The principal moments of inertia (PMI)
- The solvent accessible surface area (SASA)

Lastly the 4D descriptors are obtained through molecular ensemble, which are composed of different states of the molecules. We can extract several descriptors from the conformers, including:

- The interaction energies (kcal/mol)

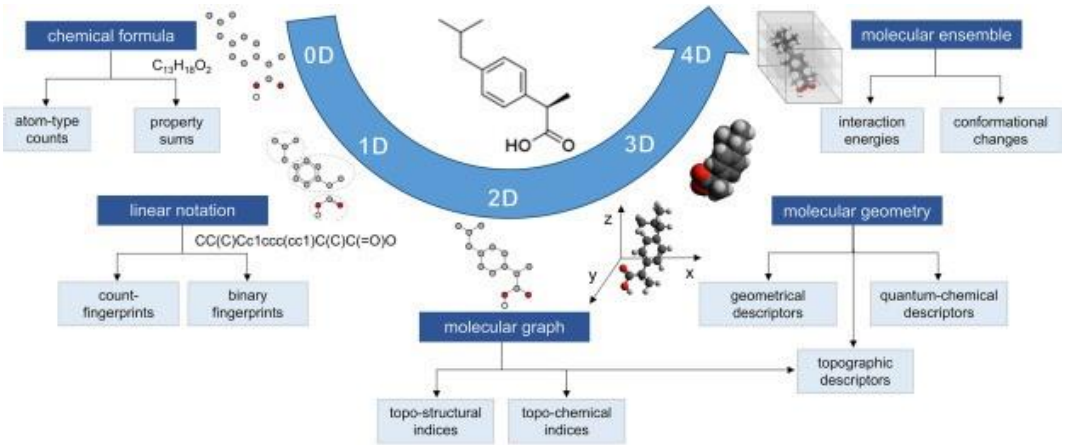



Figure 6: Various types of ligand molecular descriptors. Figure taken from Consonni *et al.* (56)

Fingerprints are alternative methods used to describe molecules. Most of the time, they are binary vectors, with each index referring to a functional group. A positive bit indicates the presence of a specific functional group, while a negative bit indicates its absence. The main difference between all these fingerprints is their calculation methods (Figure 7). For instance:

- MACCS (Molecular ACCess System) keys are built with a predefined set of queries. Each index queries the presence or absence of a specific substructure.
- Morgan, functional (FCFP) or extended-connectivity fingerprints (ECFP) are circular fingerprints, meaning that each bit corresponds to the circular environments of each atom in a molecule (57).
- Connected Subgraph Fingerprints (CSFP) thoroughly enumerate the different substructures by using molecular subgraphs (58).
- Neural graph fingerprints are obtained through convolution operating directly on molecular graphs (59).

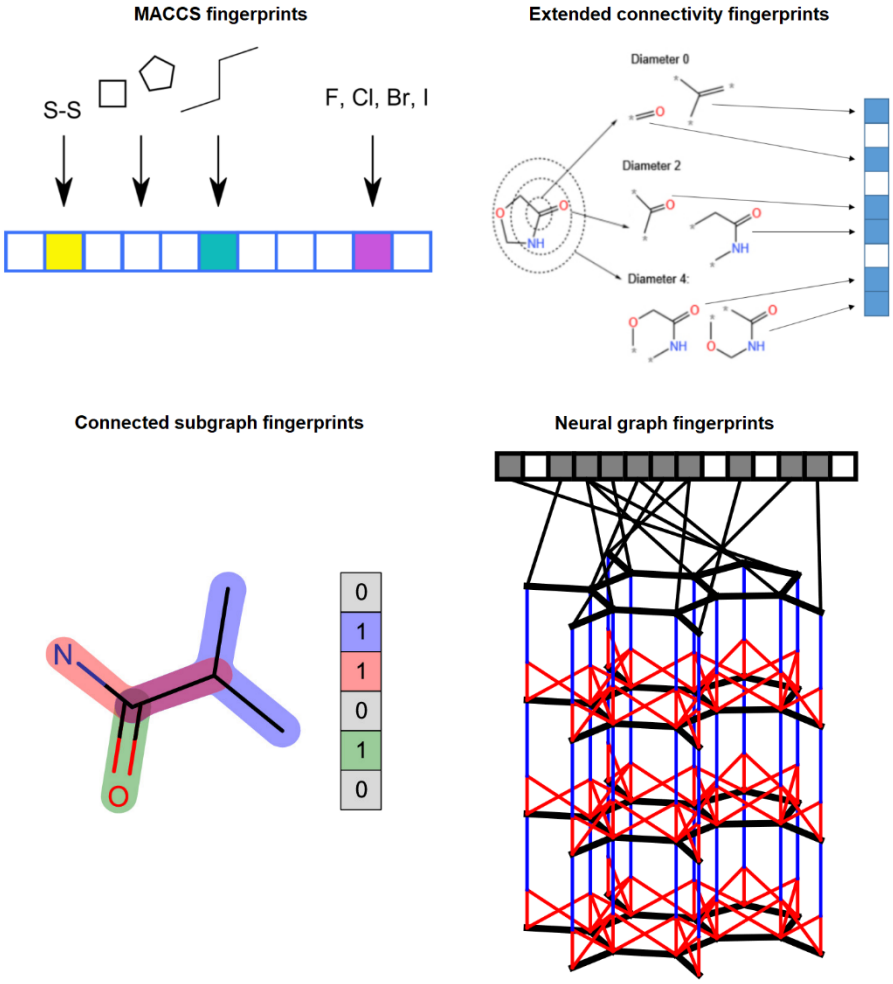



Figure 7: Examples of fingerprint generation. Figures are from Andrea Morger and the following publications (58-60)

The majority of these molecular descriptors and fingerprints have been integrated into RDKit (22). Additionally, other tools have been developed for descriptors computation, including DRAGON (61) and ISIDA (62).

### 2.1.3.2 Proteins

Descriptors have also been designed to characterize proteins. The following study compared 13 of them (63). It encompasses physicochemical property descriptors, topological descriptors and molecular electrostatic potential based descriptors, such as Z-scales (64) or MS-WHIM (65). Recently, MuLiMs-MCoMPAs (66) were released, they are tensor-based 3D protein structural descriptors based on contact matrices.

Local protein information can hold significant value, particularly in the assessment of interactions between a protein and a ligand. Given that proteins are comprised of amino acids, it becomes feasible to deconstruct the protein's description. For instance, relevant information often resides within the amino acids near the ligands. These residues can be effectively characterized using a variety of descriptors, including chemical, geometrical and graph-based descriptors (Figure 8).

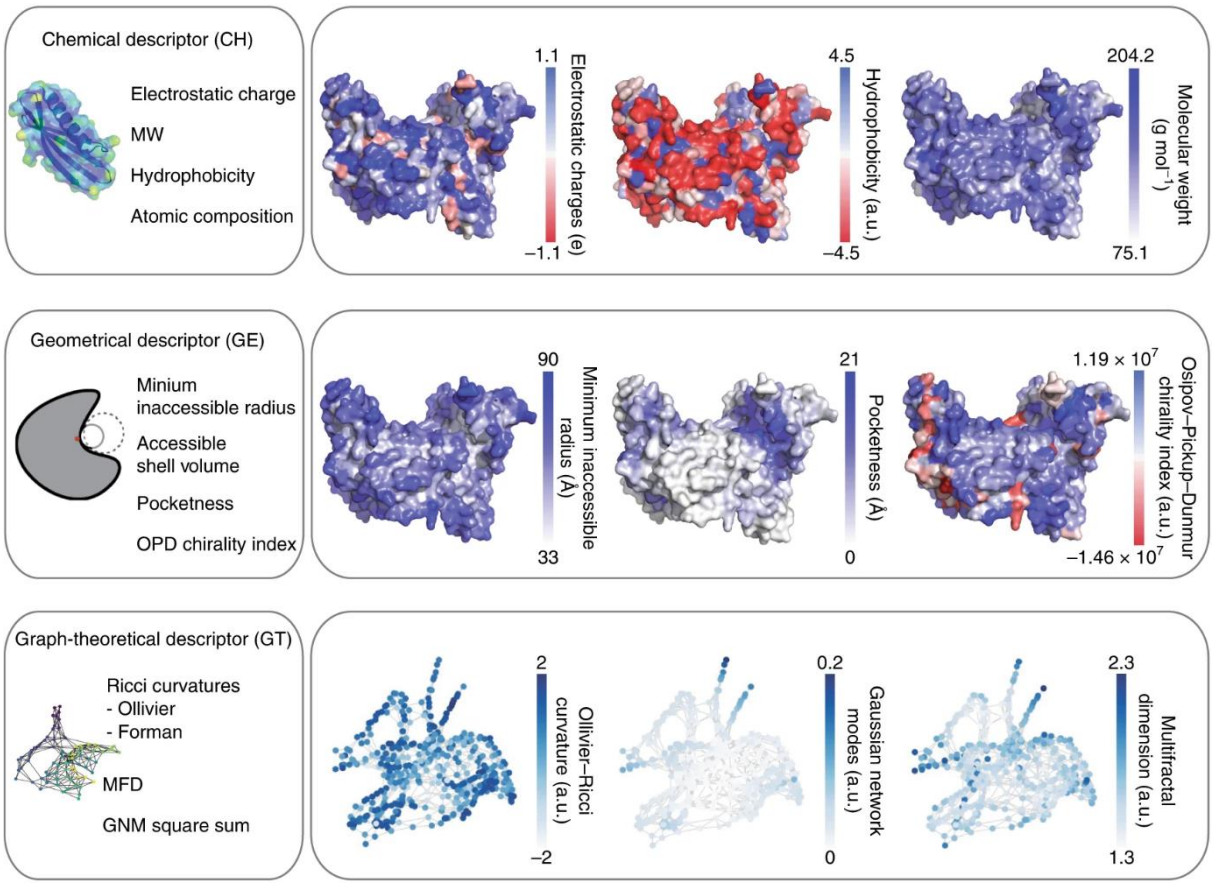



Figure 8: Visualisation of amino acids descriptors projected on the entire protein. Figure taken from Cha *et al.* (67)

### 2.1.3.3 Molecular interactions

On top of being able to characterize molecules and proteins, it is important to accurately describe the interactions between them. Some descriptors like the extended connectivity interaction features (ECIF) (68) have been designed for this purpose. There are also many more fingerprints, including:

- structural interaction fingerprints (SIFt) (69)
- atom-pairs-based interaction fingerprints (APIF) (70)

- structural protein–ligand interaction fingerprints (SPLIF) (71)
- python-based protein-ligand interaction fingerprints (pyPLIF) (72)
- protein-ligand interaction fingerprints (73)
- protein-ligand extended connectivity fingerprints (PLEC) (74)

The types of interactions encoded and their detection methods differ between the different methods. For example, the PLEC are the application of ECFP for protein-ligand interactions (Figure 9). In the same fashion as ECFP, they employ circles of increasing sizes to characterize the interaction over the bonding regions of both partners.

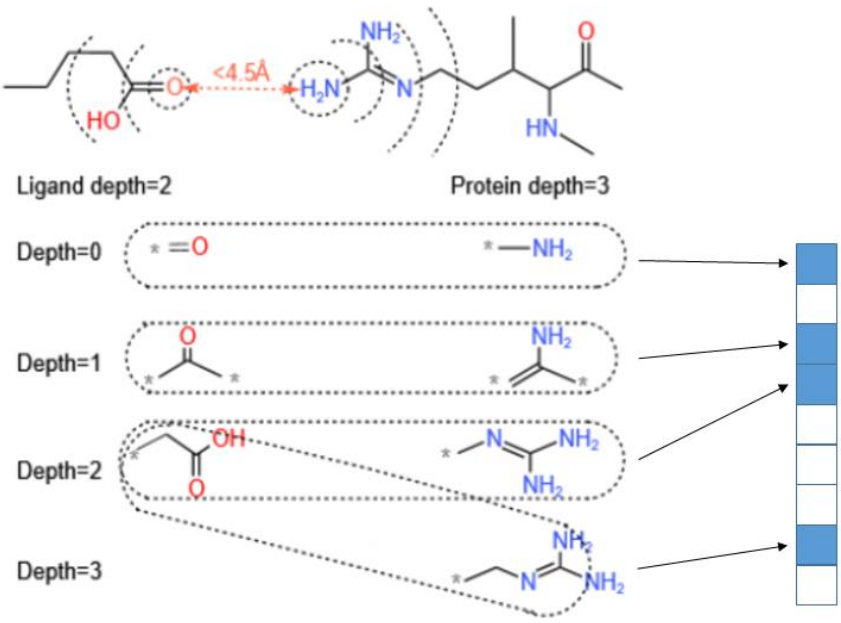



*Figure 9:* PLEC fingerprint creation process. Figure adapted from Yin *et al.* (60)

Additionally, we can mention the molecular dynamics fingerprints (MDFP) that provides information from simulations (75). They are composed of 2D-counts, both total energy and rgyr in vacuum/water, as well as both intramolecular potential energy and SASA in water (Figure 10). This type of fingerprint is well suited for binding free-energies predictions with machine learning.

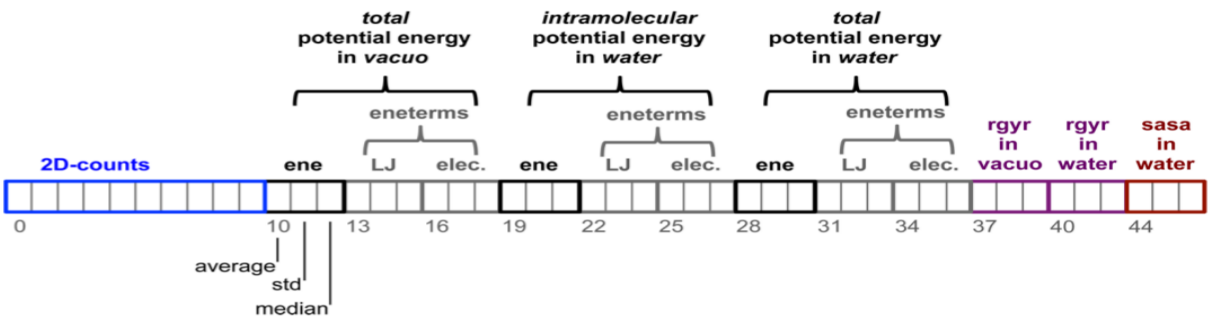



Figure 10: Composition of the MDFP+. Figure taken from Riniker *et al.* (75)

## 2.2 Overview of *in silico* methods

Since the 1960s, tremendous progress has been achieved in the field of computer-aided drug design. Numerous tools have been implemented to accelerate research and provide new insight to scientists (Figure 11). Their applications span from generating computationally molecules, known as *de novo* drug design, to identify the most active and least toxic compounds. The developed tools are used depending on the available data, *e.g.* if there is no available structure of the target protein, then ligand-based drug design (LBDD) methods like quantitative structure-activity relationship (QSAR) can still be applied. However, with the protein structure in hand, structure-based drug design (SBDD) becomes possible, enabling various applications like molecular docking.

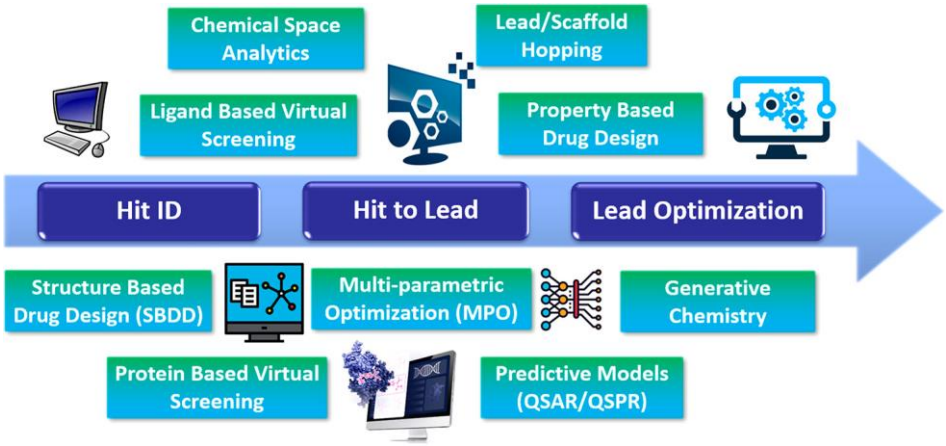



Figure 11: Various *in silico* tools implementation areas for the early stage of the drug design process. Figure taken from Cox *et al.* (76)

In the 1980s, software were developed to virtually dock a ligand or many ligands into the binding site of a protein (77). This task is performed with search algorithms such as genetic algorithms and Monte Carlo simulations (78). The docked poses are then assessed by scoring functions (SF) to select the most energetically favourable ones. Plenty of scoring functions have been developed to accurately evaluate the binding free energy of the docked poses (Figure 12). These scoring functions are classified in 4 groups: Physics-based, Empirical, knowledge-based and Machine learning-based (79, 80).

- Physics-based methods rely on using force fields to compute the binding energies. Solvent models have been added to take into account the solvation/desolvation effects and the torsion entropy. Quantum mechanics (QM) is also used to improve the predicted energies. For instance, the DOCK (43) docking software uses this type of SF.
- Empirical SF decompose protein–ligand binding affinities into distinct energy terms, encompassing factors like hydrogen bonds, hydrophobic effects, and steric clashes. The weights of each term are fine-tuned through training on a dataset of complexes with known affinity. Hence, SF like X-score (81) are computationally less expensive than physics-based SF.
- Knowledge-based methods are obtained through statistical analysis of 3D structural database. They are based on the idea that similar interactions seen before will likely have similar effects on binding affinity in new cases. They compute the probabilities of intermolecular interaction occurrences to assign a score to a given complex.
- Machine learning-based scoring function (ML-SF), such as RFscore (82) or NNscore (83), are obtained by training statistical models on large datasets of protein-ligand complexes. By

leveraging substantial data, they can learn complex interaction patterns, enabling their application to a wider range of complexes.

Crampon *et* al. provide a detailed list of docking software and the types of SFs (84).

Owing to the algorithm's speed, it is possible to apply docking software on large libraries of molecules. This approach, called high throughput virtual screening (HTVS), can identify the molecules that are most likely potent for a protein. Nonetheless, the docking software are using static conformations of the protein, and thus disregards the dynamic nature of protein-ligand interactions. Therefore, such tools face limitations when dealing with complexes in which the consideration of dynamic information is crucial. However, few docking tools allow some limited flexibility on selected side chains (85) and other *in silico* tools have been developed to provide such dynamic information, like molecular dynamics simulations.

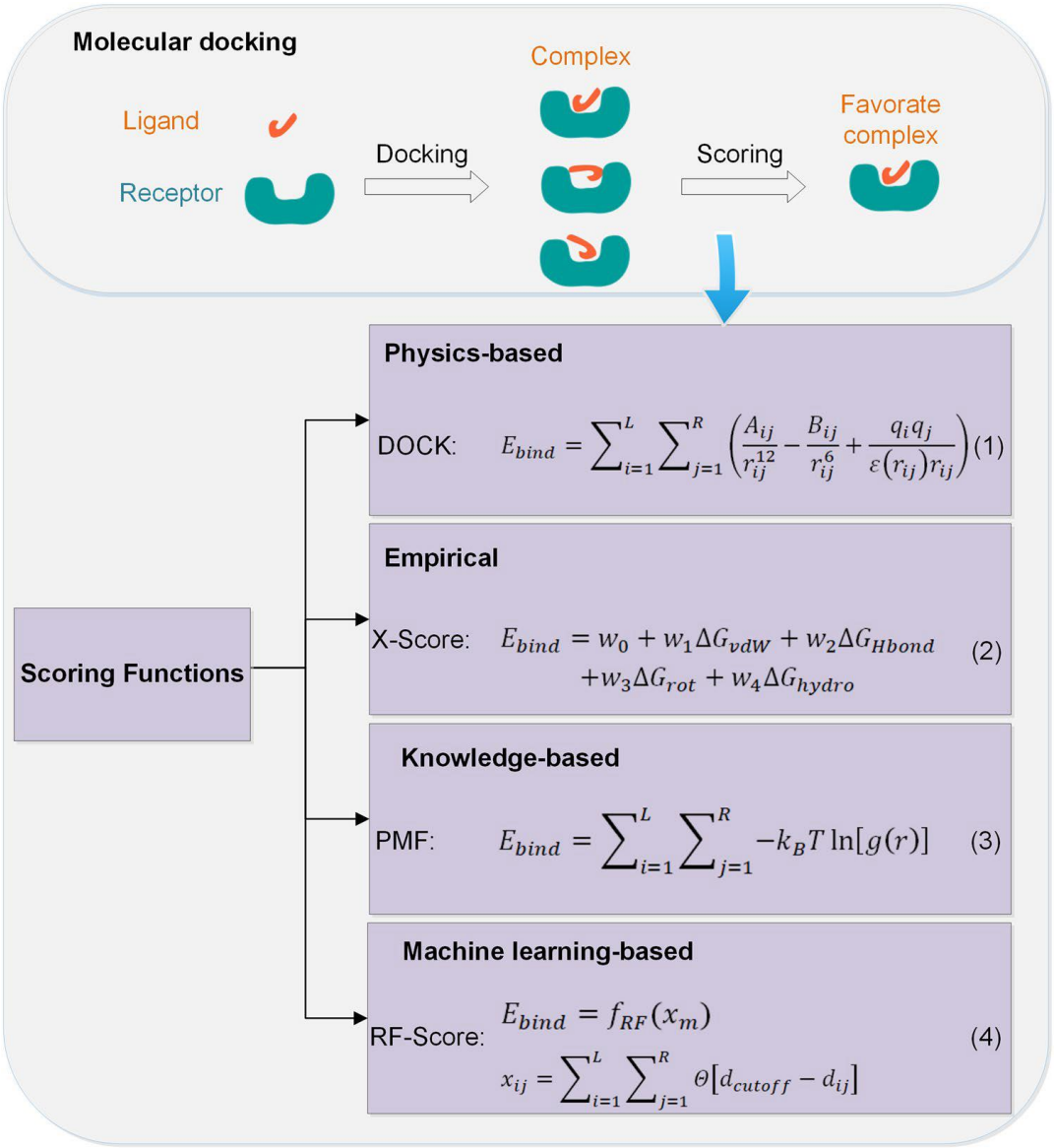



Figure 12: Molecular docking principles and the four categories of scoring functions. Figure taken from Li *et al.* (79)

## 2.3 Molecular dynamics simulations

### 2.3.1 Principles and applications of molecular dynamics

The development of molecular dynamics (MD) simulations goes back to the 1950s with their first applications on a protein in late 1970s (86). Over the years, these tools have attracted more attention, fuelled by the increase in computational power.

The principle is to simulate the movement of the ligands and proteins over a period of time. Following Newton's law of motion, forces are applied on each atom of the system. These forces have to be calculated for each time step of the simulation. Extensive work has been undertaken to parametrize the force fields. That lead to the appearance of few MD simulation software, such as Amber (87), GROMACS (88) or CHARMM (89).

Usually, the system is composed of a protein, a ligand, some ions and cofactors as well as water molecules. When using explicit solvent, the size of the protein directly influences the number of water molecules needed for solvation. Consequently, simulating the entire system becomes computationally demanding, particularly for large proteins. On average, a simulation of 10 ns can take up to a day of computation on a graphics processing unit (GPU). In comparison, when performing molecular docking, it is possible to assess several million ligands within few hours (90). The amount of computation time required to perform MD simulations also varies depending on the simulation time.

Various types of biochemical processes can be assessed depending on the simulation duration (Figure 13). The review of Hollingsworth and Dror gives a good overview of the potential applications of MD simulations (91). Here is summary:

- Explore protein conformations, since experimental studies (crystallography and cryo-EM) only provide average structural conformations. For instance, MD simulations can be used to assess the opened and closed configuration of enzymes, such as protein kinase. Additionally, they provide information on the opening/closing mechanisms by providing transition states.
- Investigate the dynamics of pockets over time. For example, undetected transient pockets can be unveiled with simulations. Furthermore, we can measure the size of pockets over time and evaluate the impact of ligand-induced fit.
- Study protein-ligand complexes:
    - By examining ligand binding and unbinding mechanisms to the pocket. This is achieved by evaluating the association/dissociation kinetic constants, $k_{on}$ and $k_{off}$.
    - By assessing the behaviour of the ligands inside the pockets. It allows measuring the binding free energies, which can help select the ligands with highest affinity for a protein.

These studies bring new insights to scientists, especially for target validation and ligand optimisation phases. As displayed in Figure 13, numerous MD simulations methods have been developed to address the aforementioned applications.

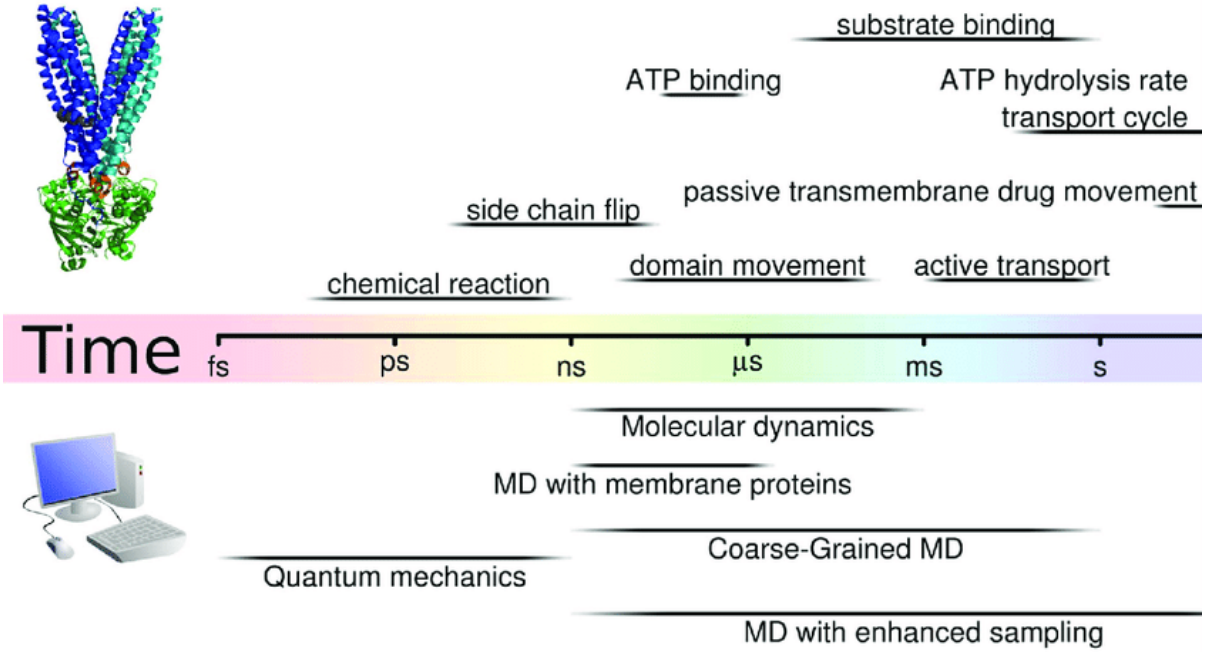



Figure 13: Time scale of the biochemical processes and the simulation methods used assess them. Figure taken from Szöllősi *et al.* (92)

A wide range of biochemical processes are studied, each with varying time scale relevance. Therefore, various methods were developed to address the different modelling challenges that arose. Some of these approaches are very precise but computationally intensive, while other techniques are less accurate and can be applied to larger systems.

### 2.3.2 Quantum mechanics methods

Quantum mechanics methods, like density functional theory (DFT) (93), apply quantum chemistry theory to model chemical reactions. Since even the electrons are simulated, the computational workload is considerable. Therefore, the simulations are applied on a limited number of atoms, and they last for few picoseconds.

When investigating enzymes and their catalytic properties, it is not possible to use QM on the entire system as it would require tremendous amount of computation time. For such cases, hybrid approach called quantum mechanics / molecular mechanics (QM/MM) was developed. The QM calculations are applied and limited to the catalytic site of the enzyme (Figure 14), where the chemical reactions are occurring, while the rest of the protein is simulated through classical MD (94). The development of this method led to the 2013 Nobel Prize in chemistry (95), awarded to Martin Karplus, Michael Levitt and Arieh Warshel.

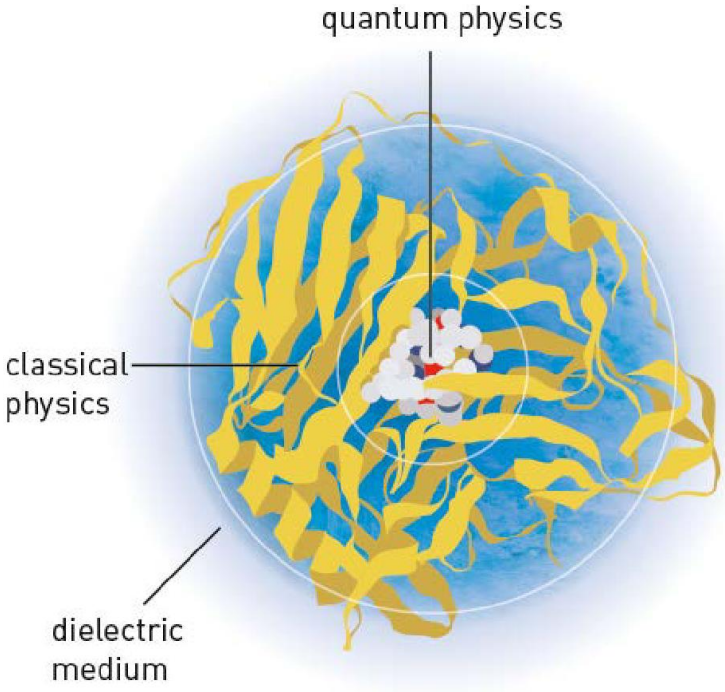



Figure 14: Schematic representation of QM/MM. Figure taken from the royal Swedish academy of sciences (95).

### 2.3.3 Biased MD Methods applied to long duration simulations and/or large systems

As shown in Figure 13, to study some specific biochemical processes, it is required to simulate the system for an extensive amount of time. The computational demand for performing all-atom simulations of several milliseconds is substantial. Therefore, enhance sampling methods are applied to circumvent this challenge. Here are presented some examples of methods implemented (Figure 15):

- Accelerated MD: minimize the energetic barriers between different states of the system. For instance, we can mention the Gaussian accelerated molecular dynamics (GAMD) (96).
- Replica-Exchange MD: exchange conformations between several simultaneous simulations. Usually, the simulations are run with different temperature values. Simulations conducted at higher temperature can more easily overcome energetic barriers, whereas those at lower temperature provide better sampling of the lowest energy minima.
- Metadynamics: accelerate conformational transitions between metastable states. It can be described as overcoming energetic barriers by filling the free energy wells (97).
- Umbrella sampling: favour the occurrence of rare events by performing multiple simulations biased with external potentials.

This topic is discussed in-depth in the review of Hénin *et al.* (98).

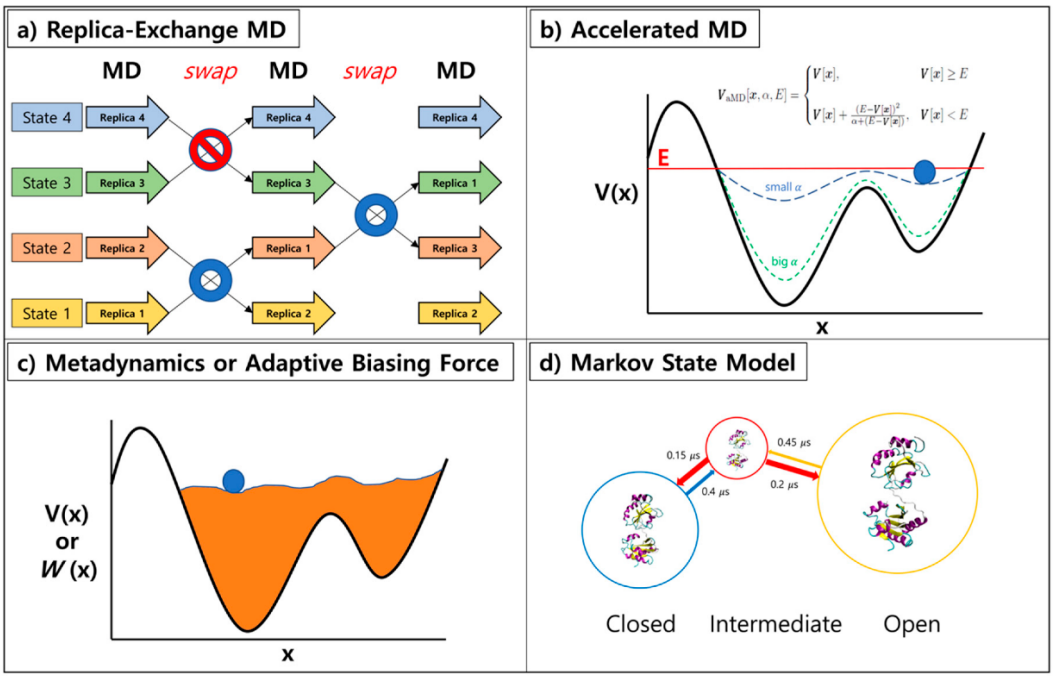



Figure 15: Overview of the enhanced sampling methods. Figure taken from Lazim *et al.* (99)

In the end, all-atoms simulations are not well-suited for exploring the behaviour of large protein complexes over extended periods. Hence, coarse-grained approaches have been implemented to reduce the computational workload. They use a simplified representation of biomolecules; for instance, the Martini coarse-grained method represents amino acids using 1 to 5 pseudo atoms (100). By employing this approach, the number of particles simulated is drastically decreased, resulting in a substantial reduction in computation.

Other biased methods, such as steered MD (101) (SMD) and targeted MD (102) (TMD) have been developed for studying protein folding/unfolding and more general conformational changes in proteins. These methods were also applied to assess the kinetic constants of protein-ligand complexes. Such studies are particularly challenging because they may require long simulations to observe the binding and unbinding events of a ligand and a protein. Consequently, the SMD and TMD were applied to accelerate the occurrence of these rare events within reasonable timeframes. The concept involves either applying forces to guide the ligand in a specific direction or defining a target end state and applying a force to minimize the RMSD to this end state during the simulation (103). Besides, these methods allow to study the path taken by a ligand to reach its target (104, 105).

### 2.3.4 Biased MD Methods for free energy (ΔG) evaluation

Various methods were developed to evaluate the binding free energy of protein-ligand complexes with MD simulations (106). For example, the molecular mechanics Poisson-Boltzmann (generalized born) surface area (MMGB(PB)SA) (107) and the linear interaction energy (LIE) (108) are used to calculate the absolute binding free energy (ABFE) in kcal/mol. The principles of the MMPBSA are to estimate the binding free energies by evaluating the energies of the complex and subtracting from it the energies of each partner when they are not in interaction. Due to the complexity of including solvent energies in the calculations, we perform the energy calculations in vacuum and subsequently

incorporate the solvation term (Figure 16). These methods are able to discriminate between binders and non-binders, but they cannot accurately evaluate binding affinities.

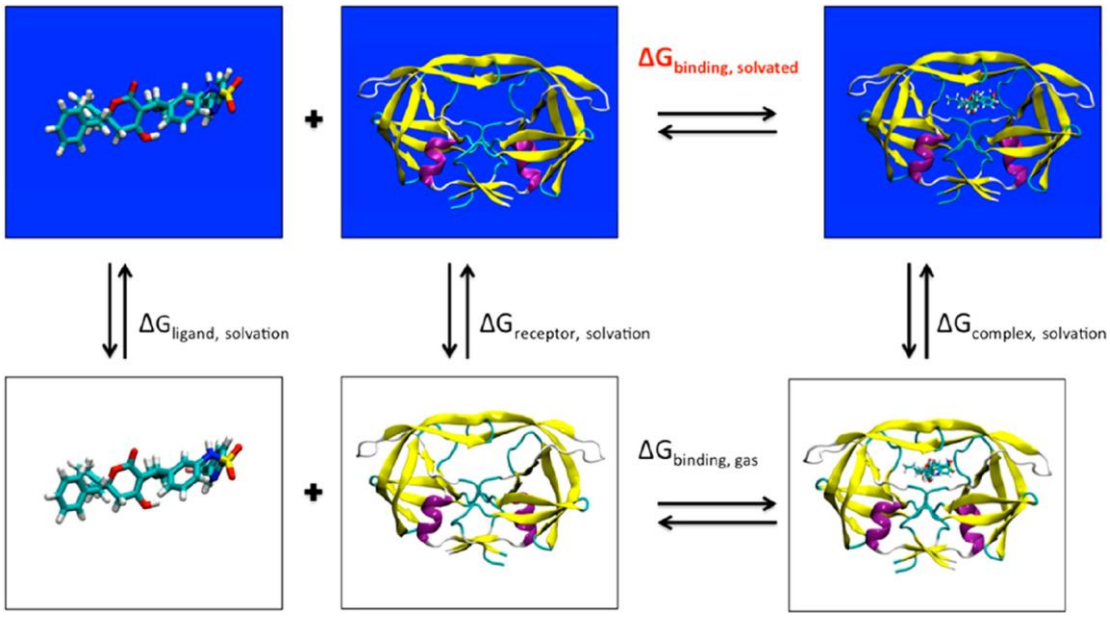



Figure 16: MM-PBSA binding free energy calculation with a thermodynamic cycle. Figure taken from Decherchi *et al.* (109)

In contrast, the relative binding free energy (RBFE) calculation methods, like the free energy perturbation (FEP) (110) or the thermodynamic Integration (TI) (111), can effectively distinguish between ligands that differ by less than one order of magnitude in affinity. However, these methods are limited to assessing the free energy differences between highly similar ligands. For this reason, and considering their significant computational cost, RBFE methods are typically employed in hit-to-lead scenarios rather than being used for virtual screening.

Overall, MD simulations are very powerful tools that provide new insights to scientists. They are well suited to be applied in later stages of the drug discovery process. However, their relatively high computational cost prevents them from being routinely used for virtual screening. Especially, when it comes to determining the binding affinity of protein-ligand complexes, other tools are preferred such as AI based predictions.

## 2.4 Artificial intelligence

AI has become an increasingly indispensable part of our society, however its development traces back to the 1950s. Over the years, it has gone through several cycles of highs and lows, with the latter often referred to as the "AI winters". These "winters" occurred during the 1970s and 1990s, brought about by disillusionment following overly high expectations (112). Recently, the surge in data and computational power allows to fully exploit of AI tools, resulting in an AI boom.

AI is an umbrella term, which can be applied to more or less advanced algorithms. AI and its subfields are displayed in Figure 17, with an increased complexity levels from AI to deep learning (DL).

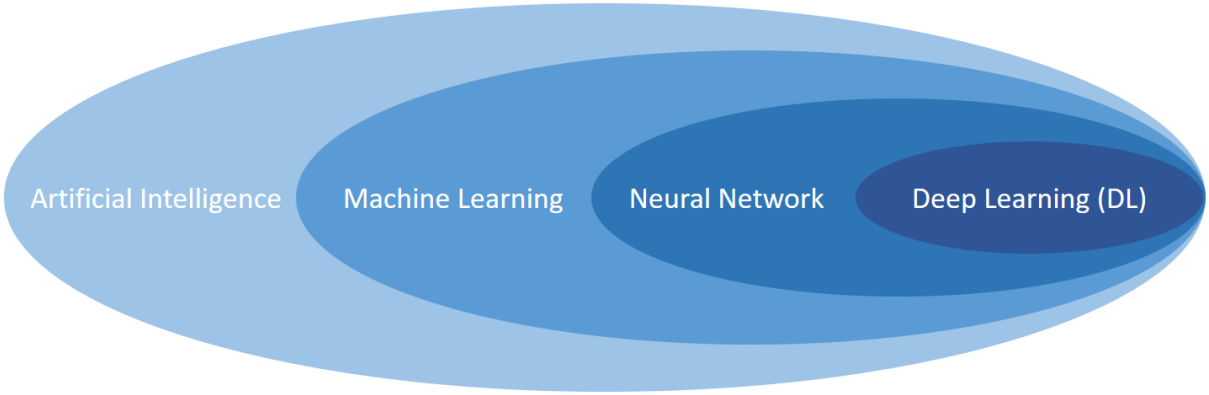



Figure 17: AI and its subfields. Figure taken from the thesis of Fabrice Carles (113).

Simply speaking, AI algorithms can be programmed to react in certain way to inputs. For example, a basic AI can be easily set up to play tic-tac-toe with a series of if/else statements, here is a pseudocode:

```
when it is the AI turn
   if two consecutive boxes are crossed and the 3rd one is empty
      then put a cross in the last one
   else if the opponent has two consecutive box crossed
      then put a cross in the 3rd one
   …
```

When moving to the realm of machine learning (ML), algorithms become more autonomous, as developers no longer need to write pre-programmed sets of actions. The ML algorithms are designed to autonomously seek solutions for a given problem. For instance, certain methods establish statistical models to accurately predict specific values. These models set up correlations between input values and their corresponding expected values, often represented through simple linear correlations like $Y = aX + b$. To do so, they train the underlying models by optimizing their performance on a dataset.

The rich landscape of machine learning features a wide array of tools (Figure 18), such as random forests (RF) (114), support vector machines (SVM) (115) and neural networks (NNs). Hence, in most scenarios, accurate predictions can be accomplished by selecting the appropriate machine learning methods. As for deep learning, it can be viewed as the implementation of advanced neural networks, able to establish complex models that are frequently achieving cutting-edge performance.

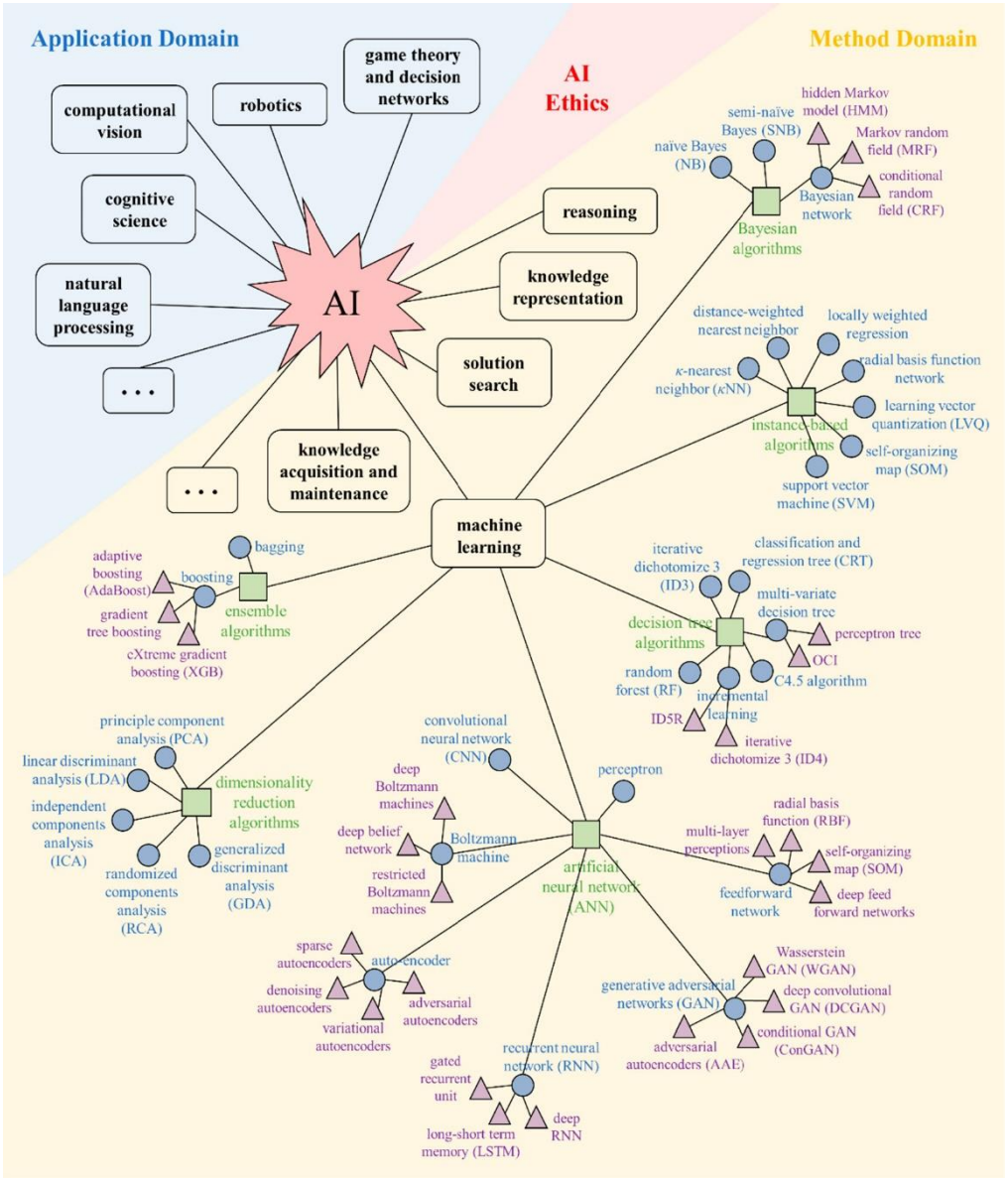



Figure 18: Machine learning landscape and application domains. Figure taken from Yang *et al.* (112)

ML methods have been applied with success in various fields including the drug design. We will review several of these methods and detail some of their applications in CADD.

Machine learning methods can be categorized in 2 groups, the supervised and unsupervised methods. The distinction lies in the fact that supervised methods are trained using data labelled with the expected outcomes, while the others are applied to unlabelled data. Thus, supervised methods aim to predict a specific value, while unsupervised ones are used to cluster and visualize data.

### 2.4.1 Unsupervised ML methods

Numerous unsupervised methods were applied on molecules, such as the one provided with ChemPlot (https://www.amdlab.nl/chemplot/) (116). The main two tasks performed by these tools are visualization and clustering:

- **Visualization**: Various dimension reduction methods have been developed to provide powerful visualisation of the data. Molecules are a prime example, as they can be characterized by a large quantity of descriptors. Comparing the molecules through univariate analysis can offer valuable insights, but it is usually insufficient to grasp a global understanding of the distribution of the molecules inside a dataset. For that reason multivariate tools have been implemented, such as the principal component analysis (PCA) (117), t-distributed stochastic neighbour embedding (t-SNE) (118), uniform manifold approximation and projection (UMAP) (119), generative topographic mapping (GTM) (120) or self-organizing map (SOM) (121).
- **Clustering**: The aim is to create groups of similar molecules based on various criteria. For instance, clustering can be applied to subsample a dataset by selecting only representative molecules from each group. Although some of the previously mentioned visualisation tools can perform data clustering, specific algorithms have been developed explicitly for this purpose. For example, we can mention the hierarchical clustering (122), partitioning methods / medoids (kmeans) (123), density based clustering (124, 125) or affinity propagation (126). Each of these methods possesses both advantages and limitations, making them suitable choices depending on the situation. For instance, some of them require prior knowledge of the number of clusters to create, whereas others can autonomously determine cluster counts based on knowledge of cluster distance for example; however, they might still necessitate user intervention to fine-tune other variables.

### 2.4.2 Supervised ML methods

The supervised methods are utilized for both classification and regression tasks, and a range of diverse techniques have been implemented for this purpose, including classification and regression trees (CART) (127), RF, SVM, Gaussian naive Bayes (GNB) (128) or extreme gradient boosting (XGB) (129). These methods have variable performance, and the interpretability of their results also varies. For instance, basic ML algorithms like CART enable the creation of easily understandable models. In contrast, RF and SVM deliver superior performance but exhibit lower interpretability. Such methods have been applied in the field of drug design to address various objectives:

**Activity predictions**: The binding affinity is experimentally measured in $K_d$/$K_i$/$K_a$ with *in vitro* assays, such as surface plasmon resonance (SPR) (130) or isothermal titration calorimetry (ITC) (131). Other activity values can be experimentally assessed such as the half maximal inhibitory concentration ($IC_{50}$) and the half maximal effective concentration ($EC_{50}$). Although, the $IC_{50}$ and $EC_{50}$ are usually less reliable than $K_d$/$K_i$/$K_a$ as they depend a lot on the experimental conditions, such as the concentration in ligands and proteins.

ML methods have been applied to predict the activity of a molecule for a given target. They can establish a relationship between molecular structures and binding affinities, employing a methodology known as QSAR. This approach, dating back to 1962, relies on training models on a dataset of molecules with known activity against a target (132). As a result, these models identify pharmacophores crucial for the molecules' activity to deliver a prediction. Although several warnings have been raised regarding their limited predictive power, especially in cases involving activity cliffs (133). In 2012, deep neural networks were able to outperform other QSAR models by winning the Merck kaggle challenge (134). Recent deep learning architectures (135) were release to further improve our predictive ability in that field. The following review details the methods recently implemented in the field of QSAR (136).

Likewise, proteochemometry (PCM) models have been devised to forecast the activity of molecules against a protein. By conducting experimental assays, the activity of several ligands against multiple proteins has been measured. Consequently, experimental activity data is available for pairs of protein-ligand, enabling the construction of an activity matrix. The objective of these models is to fill the missing elements in this matrix. For instance, this methodology has been employed to determine the affinity of molecules for protein kinases (113).

An alternative approach is to predict binding affinity using the 3D data from protein-ligand complexes, when such data are available. As mentioned previously, several types of SFs have been developed for this task, including ML-SF. SFs can be used to evaluate the binding affinity of complexes by using 3D structures obtained experimentally, but the ultimate goal is to predict the binding affinity of docking poses in a virtual screening scenario. Classical SFs typically generate docking scores that are usually negative like binding free energies, while ML-SF predict binding affinity values like $pK_i$.

To assess the performance of scoring functions, they are evaluated across various use-cases using four metrics: scoring power, ranking power, docking power, and screening power (Figure 19).

- **Scoring power** (several ligands, several proteins): It is a regression task designed to assess the ability of a SF to predict accurately the binding affinity of a complex. To perform this evaluation, SFs are applied on benchmark datasets composed of crystal poses with experimentally determined affinities.
  The performance is displayed with the correlation coefficient of Pearson ($R_P$). This statistical measure represents the linear correlation between the predicted values and experimental values.

$$R_P = \frac{\sum_{i=1}^{N}(x_i - \bar{x})\,(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2 \sum_{i=1}^{N}(y_i - \bar{y})^2}} \qquad (1)$$

  Where "$x_i$" represents the computed binding score for the ith complex, "$y_i$" stands for the corresponding experimental binding constant of that complex, and "$N$" is the total number of samples.

  Alternatively, the root mean square error (RMSE) can be computed to know the average difference between predicted values and the experimental values.

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}{N}} \qquad (2)$$

  Where "$x_i$" represents the real affinity value for the ith complex, "$\hat{x}_i$" is the predicted affinity value for that complex and "$N$" is the total sample size.

- **Ranking power** (several known ligands, 1 protein): The models are evaluated for their ability to accurately rank known ligands according to their activity for a specific target protein. The ranking of complexes is determined by their predicted binding affinity, spanning from the least active to the most active. Subsequently, the model's ranking is compared the ranking based on experimental activity values.

It differs from scoring power, as ranking power does not necessitate a linear correlation between computed binding scores and experimental binding data. Thus, this metric is better suited for selecting the SFs that will deliver superior results in virtual screening.
Spearman's rank correlation coefficient (ρ) is used as the quantitative indicators of ranking power. It is a nonparametric measure of the rank correlation. It can be calculated for several target proteins.

$$\rho = \frac{\frac{1}{N}\sum_{i=1}^{N}(rx_i - \overline{rx})\,(ry_i - \overline{ry})}{\sqrt{(\frac{1}{N}\sum_{i=1}^{N}(rx_i - \overline{rx})^2)\,(\frac{1}{N}\sum_{i=1}^{N}(ry_i - \overline{ry})^2)}} \qquad (3)$$

Where "$rx_i$" stand for the rank of the binding score of the ith complex, and "$ry_i$" is the rank of the experimental binding constant of that complex. "$\overline{rx}$" and "$\overline{ry}$" are the mean ranks and "$N$" is the total sample size.

- **Docking power** (1 ligand, 1 protein, several poses): It is the ability to distinguish the native ligand binding pose from computer-generated decoy poses. Initially, a ligand is redocked in its binding site. Subsequently, poses exhibiting a ligand root mean square deviation (RMSD) of less than 2 Å from the initial position are classified as native ligand binding poses, while those with an RMSD exceeding 2 Å are designated as decoy poses. The models' performance is assessed for their ability to rank the native ligand binding poses over the decoy poses.
  It is quantified by measuring the success rate of native ligand binding poses among the top-ranked poses.

- **Screening power** (decoys, true actives, 1 protein): Models are employed within virtual screening scenarios, where both known active and decoy molecules are available for a specific target. To obtain structural data, the decoys are docked into the binding site. It is also possible to crossdock the protein-ligand complexes from a dataset like the PDBbind (2), in such case we assume that the ligands docked in other proteins are decoys. The poses of the decoys are analysed by SFs alongside with the crystal poses of the true binders. The aim is to measure models' capabilities to rank actives over decoys.
  To measure the performance, one might draw either a receiver operating characteristic (ROC) curve or a total operating characteristic (TOP) curve. Subsequently, it is possible to compute the area under the curve (AUC). The higher is the AUC, the better is the SF performance.
  Given the substantial number of ligands assessed during virtual screening, the focus often rests on the highest-ranked ligands. Therefore, we use an enrichment factor (EF) to evaluate models' ability to rank active molecules within the top 1-5% ranking.

$$EF_\alpha = \frac{\mathrm{NTB}_\alpha}{NTB_{total} \times \alpha} \qquad (4)$$

Where, $NTB_\alpha$ represents the number of true binders identified among the highest-ranked candidates (*e.g.*, α = 1%, 5%, or 10%) chosen by the SF. $NTB_{total}$ is the overall count of true binders for the particular target protein.

The comparative assessment of scoring function (CASF) benchmark dataset has been used to evaluate SFs for all four power metrics (8). SFs, such as X-Score (137), Autodock Vina default SF (42) or GlideScore-SP (138), tend to perform well for one or two metrics but not for all of them. However, ML-SF consistently outperforms the other SFs, such as $\Delta_{Vina}RF_{20}$ (139) and Graph Transformer (GT) / Gated graph convolutional network (gated GCN) (140). This is a good indicator of ML potential in carrying accurate predictions on specific test sets. The development of ML-SF is a prominent topic in the field of SBDD, as there are ongoing efforts to improve statistical models' accuracy.

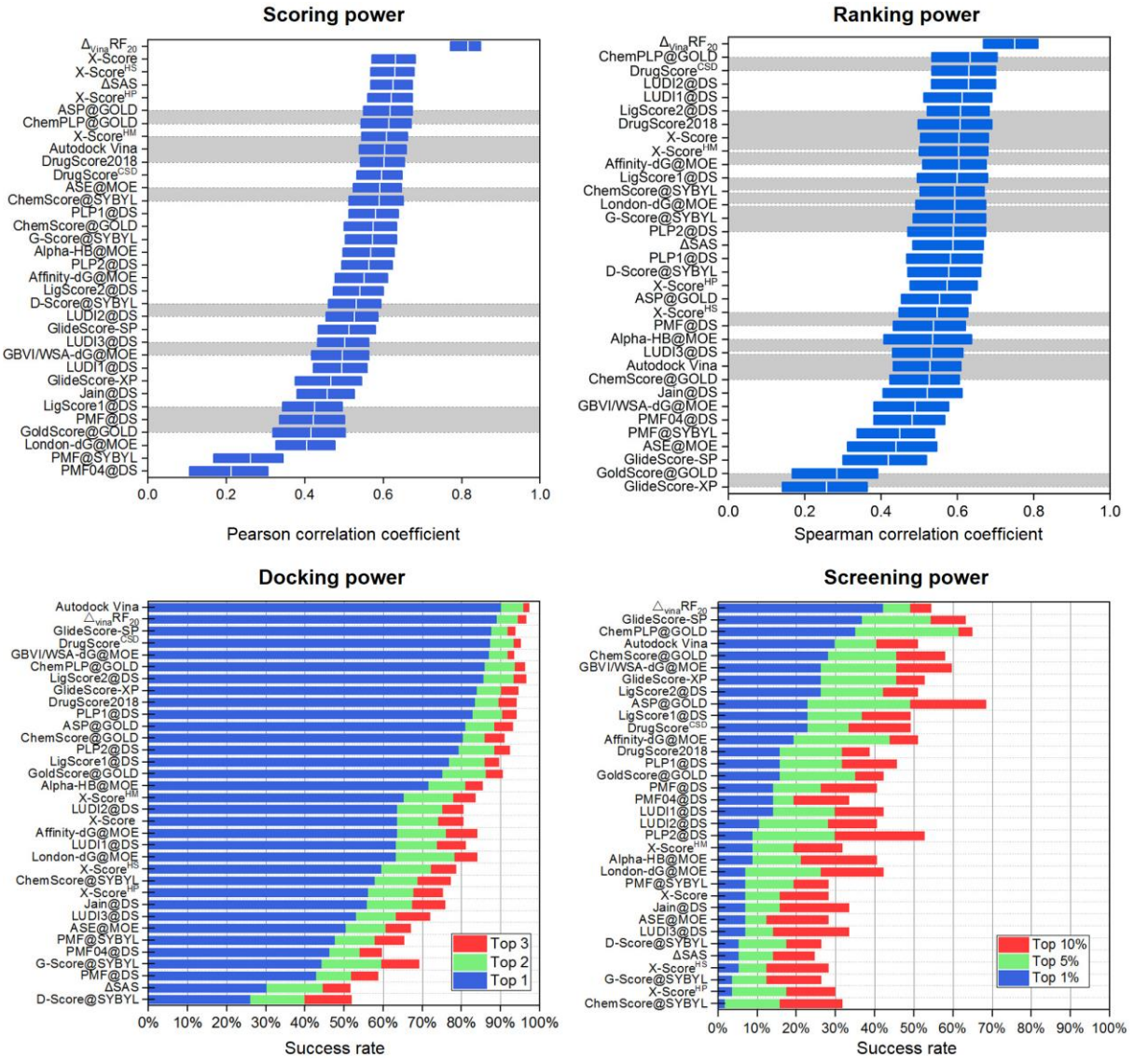



Figure 19: Comparison of scoring functions for the scoring, ranking, docking and screening powers. Methods were evaluated on the CASF-2016 (8). Figures taken from Su *et al.* (8).

**Property predictions**: Beyond the activity of complexes, other molecular properties can also be predicted, such as the lipophilicity (LogP) (141), the solubility (142) or the blood-brain barrier permeability (143). These variables are important in the drug design process, as they influence the molecule viability as a drug. Most of the time, these values are predicted via quantitative structure-property relationship (QSPR) studies, akin to QSAR but extended to other properties (144). Recently,

deep learning tools like DeepGRID (145) were developed for predicting blood-brain barrier permeability.

**Selectivity predictions**: It is also crucial to ascertain whether a molecule exhibits selectivity for a target. Indeed, having a drug that interact exclusively with its intended target reduce the risks of side effects. Several models (146) have been developed to discern the selectivity of molecules between two proteins families. Beyond just identifying promiscuous molecules, these tools can also be employed to search for dual inhibitors, which exhibit synergetic effects. For instance, such treatments can overcome cancer multidrug resistances (147).

**Toxicity predictions**: The toxicity of molecules is another critical factor to take into account. In fact, over 30% of drug candidates are dropped due to toxicity (148). It is especially important to evaluate the toxicity of molecules early on, as the toxicity is only assessed in the later stages of the drug discovery process with *in vitro* and *in vivo* essays. In 2014, the Tox21 Data Challenge was held to benchmark statistical models' predictions for various toxicity endpoints. The victory was achieved by a deep learning model called DeepTox (149), which established a new standard in toxicity prediction at that time. Tran *et al.* (150) give a comprehensive review the recent advances in AI-based drug toxicity prediction across six key toxicity properties: hERG, median lethal dose ($LD_{50}$), DILI, Ames mutagenesis, carcinogenesis, and skin sensitization.

**Synthesisability predictions**: In CADD pipeline *de novo* drug design is becoming increasingly prevalent for generating novel molecules. Nonetheless, barring the fact that the molecules may not be valid at all, there could still be challenges in synthesizing them, or they might even be completely non-synthesizable. Therefore, tools were developed to evaluate their synthetic feasibility like the synthetic accessibility score (SAscore) (151). ML models were also trained to achieve such task including the deep learning model SCScore (152). This topic is discussed in greater detail in Gao *et al.* (153).

**Retrosynthesis predictions**: Instead of merely calculating a synthesisability score, other tools provide retrosynthesis predictions for a molecule. ML, particularly deep learning, is playing a major role in this field with innovations like AiZynthFinder (154). These methods offer synthesis plans for producing molecules, significantly aiding chemists in their work.

As shown in the previous examples, deep learning has been extensively employed in various drug design related topics with notable successes. Indeed, over the past decade, DL has emerged as a leading tool that significantly enhances our predictive capabilities. In the following sections, we will delve into its mechanisms and explore its applications, with a particular focus on its role in drug design.

### 2.4.3 Neural network and deep learning functioning

Artificial neural networks (ANN) have been developed to mimic the functioning of the brain. They are composed of neurons that are interconnected, as exemplified with a simple NN – the perceptron (Figure 20 – A). Each neuron takes several inputs, sum over these variables and output a single value (Figure 20 – B). The information is propagated along the neural network, while doing so each neuron, from the hidden layer, perform a non-linear transformation to it. When reaching the end of the neural network, the processed information is used by the neuron of the output layer to produce a prediction. While training a model, the prediction is compared to the expected value. In function of the results, the weights assigned to each connection between neurons are modified through a mechanism called

back-propagation. This mechanism is performed to reduce the difference between the predictions and the expected values.

An epoch corresponds to one complete training cycle, *i.e.* when the neural network has been trained with all the available training data. Typically, multiple epochs are necessary to obtain an effective model. Consequently, the neural network processes each data point several times before achieving the final model.

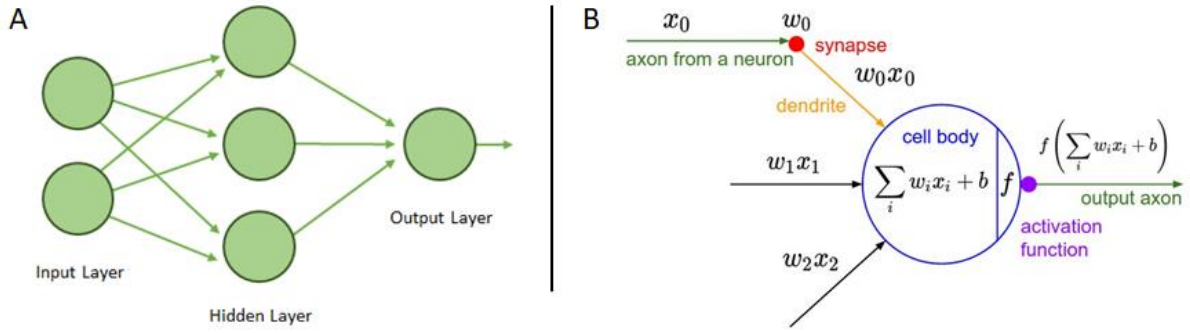



Figure 20: Neuron and neural network. A – Architecture of the perceptron, which is composed of an input layer, a hidden layer and an output layer. B – Composition of a neuron. Where "$x_i$" represent input values, "$w_i$" are the weights, "b" stands for bias, and "f" is the activation function. Figure B is taken from Stanford course (155).

Deep learning is defined as a machine learning method based on the use of neural networks with several hidden layers. The ability of the underlying models to solve complex issues expands as the neural network size grows. With multiple hidden layers, models gain the capability to understand complex patterns.

Activation functions are essential for introducing non-linearity into neural networks. The non-linearity enables the model to capture intricate patterns in the data. These activation functions are applied to each neuron and come in several types (Figure 21):

- **Sigmoid**: The sigmoid function confines the output between 0 and 1, akin to logistic regression in binary classification. This is applied in the output layers of neural networks for binary classification and logistic regression. For example, it might predict an 80% chance that a molecule is toxic.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{5}$$

  Where "$x$" is the input value.

- **Tanh**: The tanh function shares similarities with the sigmoid function but maps input values to an output range of -1 to 1. When employed within hidden layers, it converges more readily and demonstrates enhanced performance compared to the sigmoid function.

$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{6}$$

- **ReLU**: The rectified linear unit (RELU) sets negative input values to zero. Compared to tanh, it achieves similar performance while being six times faster (156). Initially introduced in 1975 (157), it later regained prominence in 2010 (158) and has since become the most widely used activation function.

$$f(x) = \max\{0, x\} \tag{7}$$

- **Leaky ReLU**: It is a modified version of the ReLU, introducing a small positive gradient for negative input values. This helps mitigating the vanishing gradient problem.

$$f(x) = \begin{cases} x & \text{if } x > 0, \\ 0{,}001 & \text{otherwise.} \end{cases} \tag{8}$$

- **GeLU, SiLU (Swish), Mish:** Mish (159), the Gaussian error linear unit (GeLU) and the sigmoid linear unit (SiLU) (160) are smooth approximations of ReLU that have recently been introduced. They have been employed in state-of-the-art neural networks, such as Bidirectional Encoder Representations from Transformers (BERT) (161).

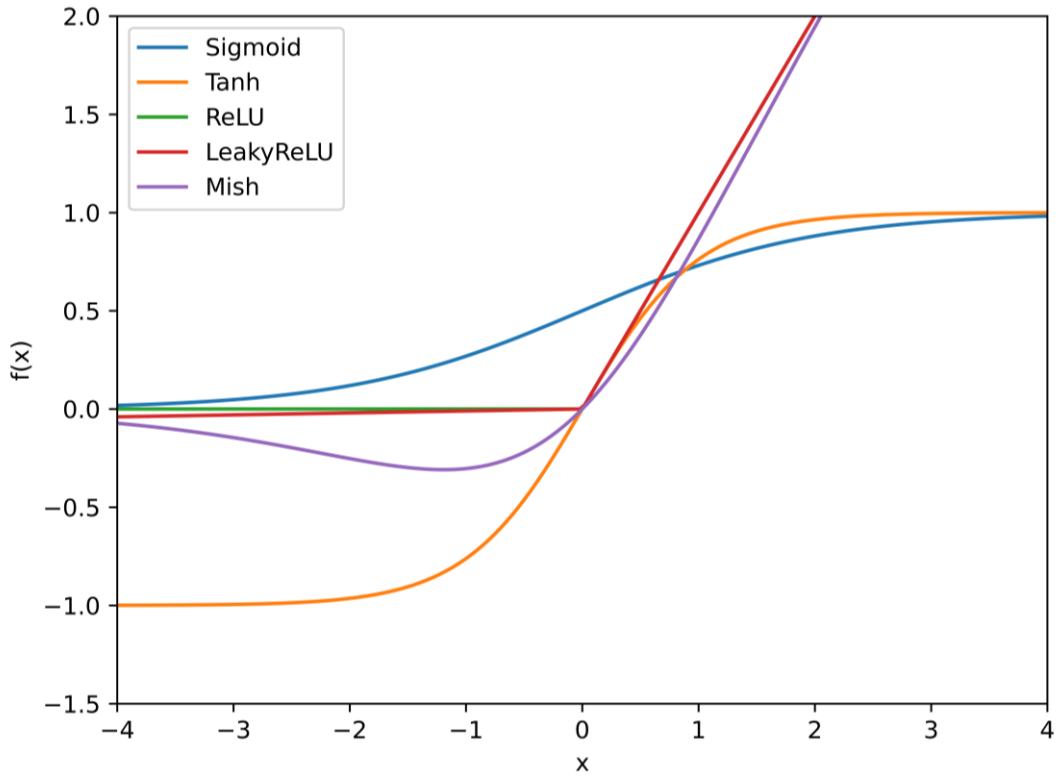



Figure 21: Several types of activation functions.

Deep learning has achieved major milestones across various application domains (1, 162, 163) and is currently spearheading the AI field. Indeed, it leverages the potential of big data to yield outstanding outcomes. However, while DL often surpasses traditional ML algorithms, achieving success with deep learning requires a substantial amount of data and computational resources.

The implementation of deep learning algorithms has been greatly facilitated by the availability of dedicated libraries, with the most prominent ones being TensorFlow (164) and PyTorch (165). These libraries, developed respectively by Google and Meta, have played a pivotal role in advancing the field of deep learning. Various types of neural networks have been developed to perform predictions across a wide range of tasks. Each neural network architecture is optimized to handle specific types of data. In the following sections, we will provide an overview of how some of the most renowned neural network architectures operate.

**Deep neural networks** (DNN):

They are composed solely of interconnected layers of neurons. Their input data are simple variables, such as molecular descriptors that have been computed beforehand. This classic implementation of deep learning has proven to be successful in various drug design tasks, as highlighted earlier, including toxicity prediction (149) or QSAR modelling (134).

Given their fundamental role in deep learning, these neural networks are frequently integrated into other neural network architectures, under the name of “multilayer perceptron” (MLP) or “fully connected” (FC) networks.

**Convolutional neural networks** (CNN):

The concept of CNN was first introduced in 1980 with the neocognitron (166), and the application of CNNs with backpropagation was successfully achieved in 1989 (167). A significant milestone was reached in 2012 with the advent of AlexNet (156), an influential CNN implementation that harnessed the power of GPUs. This development resulted in a notable 10% performance improvement on ImageNet (168), a widely recognized benchmark dataset in the field of computer vision. This achievement was surpassed in 2015 by ResNet (169), a CNN capable of accommodating anywhere from 100 to 1000 convolutional layers. These successes ignited a surge of interest in applying CNNs across a diverse range of domains and applications.

CNN are able to autonomously extract useful features from a grid, typically an image made of pixels. They do not require expert crafting of descriptors which induce bias in the underlying models. This inherent capability makes them well-suited for computer vision tasks, such as image recognition. Comprising multiple layers, including convolutional and pooling layers (Figure 22), CNNs excel at extracting information from images, with convolutional layers capturing image details and pooling layers subsampling this information to enhance overall generalization.

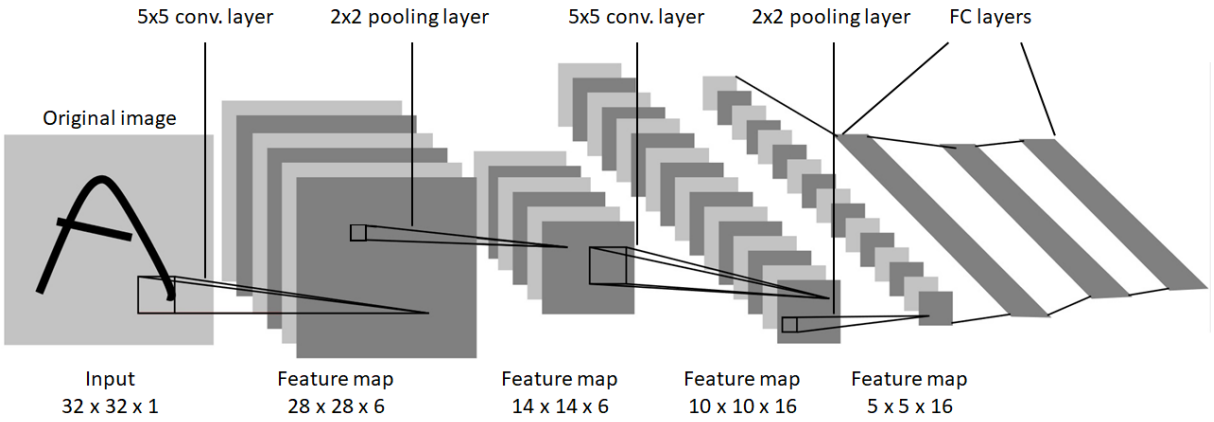



Figure 22: LeNet-5 – a classic implementation of CNN. Figure adapted from LeCun *et al.* (170)

Convolutions are performed by using filters that slide over images, thereby extracting spatial information and generating feature maps. A filter, often referred to as a kernel, constitutes a matrix with dimensions that typically range from 3x3 to 5x5 when applied on 2D images. The values within this matrix represent weights, and as the filter slides over the input data, a point-wise multiplication is carried out with the corresponding receptive field (Figure 23). The results of each element-wise product are summed up onto a feature map. Throughout the training process, the weights of the filters undergo adjustments via backpropagation. Doing so, each filter is modified to capture various details about the input data.

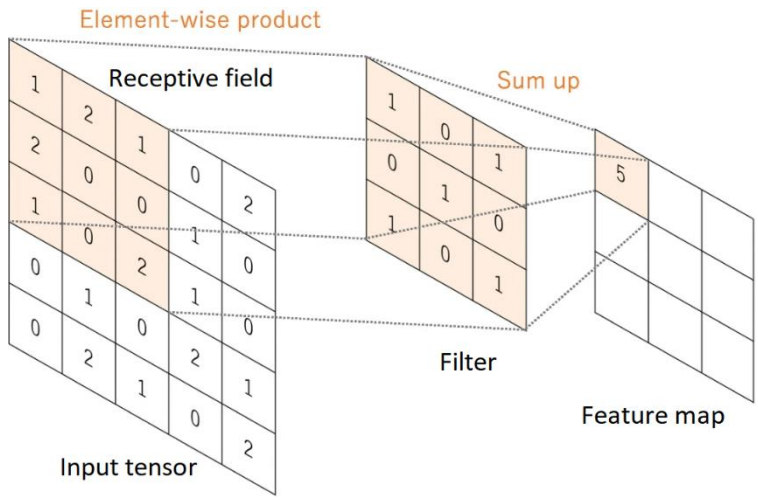



Figure 23: Convolution mechanism. Figure adapted from Yamashita *et al.* (171).

In addition to convolutions, another crucial aspect of CNN is pooling. Pooling facilitates the down-sampling of feature maps by summarizing the presence of features within localized regions (Figure 24). This technique enhances the resulting down-sampled feature maps' resilience to variations in feature position within the image. The main methods employed are max pooling and average pooling. Max pooling involves selecting the value maximum within the localized region, while average pooling computes the average value over the local region.

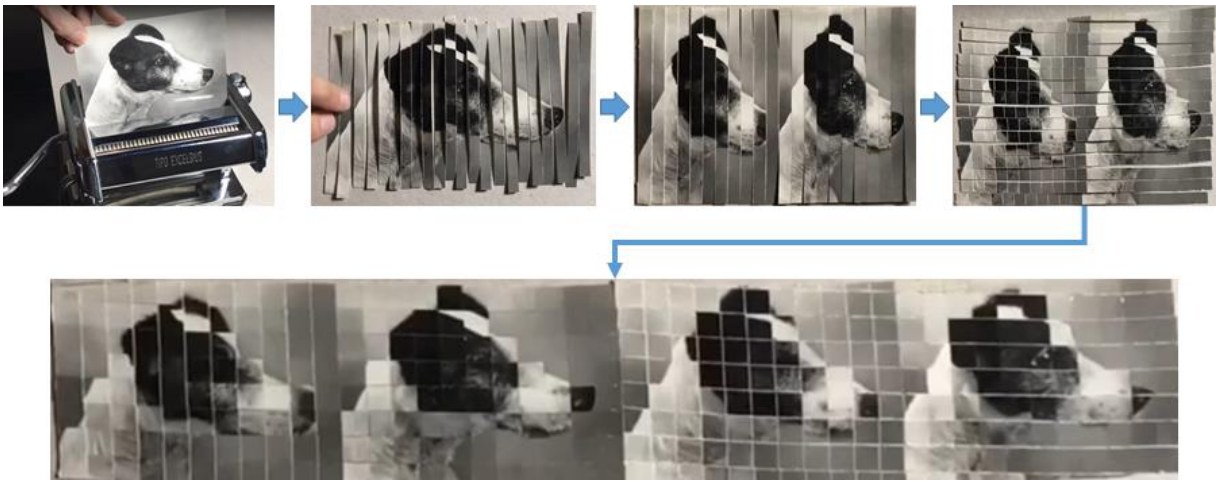


Figure 24: Visual representation of down-sampling. Images are from Kensuke Koike (172).

Performing convolutions and pooling effectively transform the input data. As a result, the feature maps generated throughout this process capture distinct information about the data. The feature maps obtained with the initial layers primarily represent basic shapes, whereas those in subsequent layers are able to capture finer details (Figure 25).

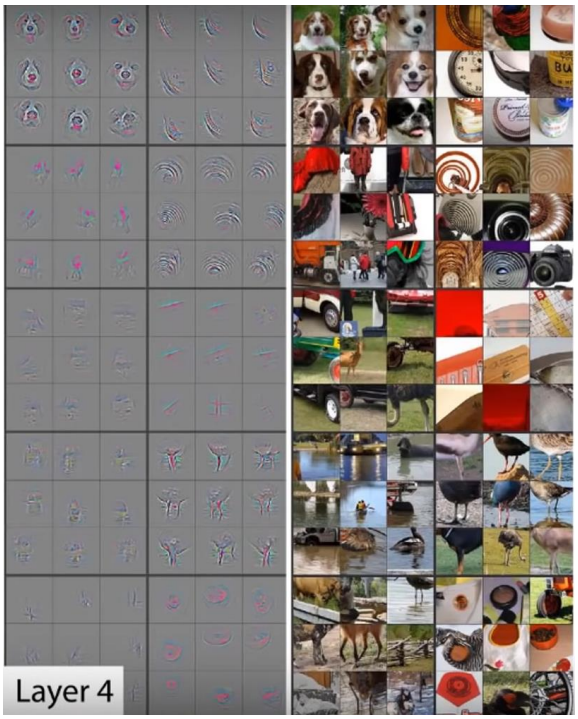



Figure 25: Visualisation of feature maps. Feature maps are on the left and their corresponding input data on the right. Figure taken from Zeiler *et al.* (173).

Following the data extraction phase, which includes the convolutional and pooling layers, the resulting feature maps undergo further processing via an MLP. The FC layers within the MLP identify correlations within the processed data, ultimately leading to a prediction made by the output layer.

CNNs have versatile applications across various data types, including images and 3D structures. Although, when dealing with 3D structures, it is necessary to pre-process the data by discretizing it into a grid to make it compatible with CNNs. It is crucial to approach this discretization with care, particularly when describing 3D molecular structures with voxels. Such voxel-based representations are often sparse matrices, which lead to performing convolutions on uninformative data. This not only slows down the training process but also has the potential to negatively affect performance.

**Graph neural networks** (GNN):

These neural networks are applied on graph data, such as social networks or molecules. A variety of neural networks have been developed including message passing neural networks (MPNN) (174), graph convolutional networks (GCN) (175) or graph attention networks (GAT) (176).

Graphs consist of nodes and edges, with each node possessing an associated embedding, and these nodes are interconnected by edges. The graph's edges can also be oriented, implying a specific direction on them. Typically, during training, the embedding of nodes get updated based on the embedding of their neighbouring nodes.

GNNs can be employed on the same data as CNNs, especially in geometric deep learning scenarios when dealing with 3D molecular data. In such instances, GNNs often prove to be more suitable than CNNs, as they focus on the important information, *i.e.* atoms and their interactions. However, GNNs necessitate the explicit representation of non-covalent interactions. This calls for careful handling in detecting these interactions and accurately assigning the corresponding edges.

**Long short-term memory** (LSTM):

Introduced in 1997, LSTM (177) belongs to the family of recurrent neural networks (RNNs), which are designed for handling time series data, such as text or videos. As a result, they have found extensive application in domains such as speech recognition and natural language processing (NLP). For instance, in a “one to many” configuration (Figure 26), RNN can be employed for text generation based on a single word/sentence. Conversely, in a “many to one”, they can process entire sentences to perform sentiment analysis. Alternatively, they can be applied in a “many to many” setup for translation purposes. They take as input each timeframe sequentially and can also output results in sequential manner.

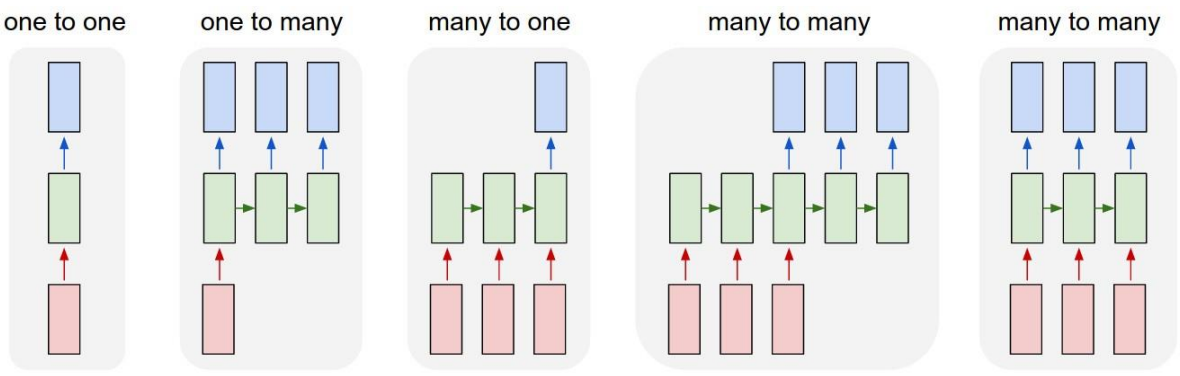



Figure 26: RNN applications. Red, green and blue rectangles represent respectively the inputs, the RNN’s states and the outputs. Figure made by Andrej Karpathy (178).

LSTM are designed to address the vanishing gradient problem, which particularly affects RNNs. This problem arises during backpropagation and prevent weights from being updated. It is especially prevalent when dealing with long input sequences, making it challenging for RNNs to accurately capture long-term dependencies.

The LSTM cell is composed of an input gate, an output gate and a forget gate (Figure 27). These 3 gates regulate the inflow and outflow of information within the cell:

- **Forget gate**: The forget gate determines what previous state information should be discarded. Through a sigmoid function, it assigns a weight between 0 and 1 to the previous state, based on a comparison of the previous state and the current input.
- **Input gate**: The input gate determines which new information should be integrated into the current state. It employs a similar mechanism as the forget gate, to determine which parts of the input to keep.
- **Output Gate**: The output gate oversees the extraction of information from the current state to be passed on as output. In a similar fashion, it considers both the previous and current states to determine what information to transmit to the next step.

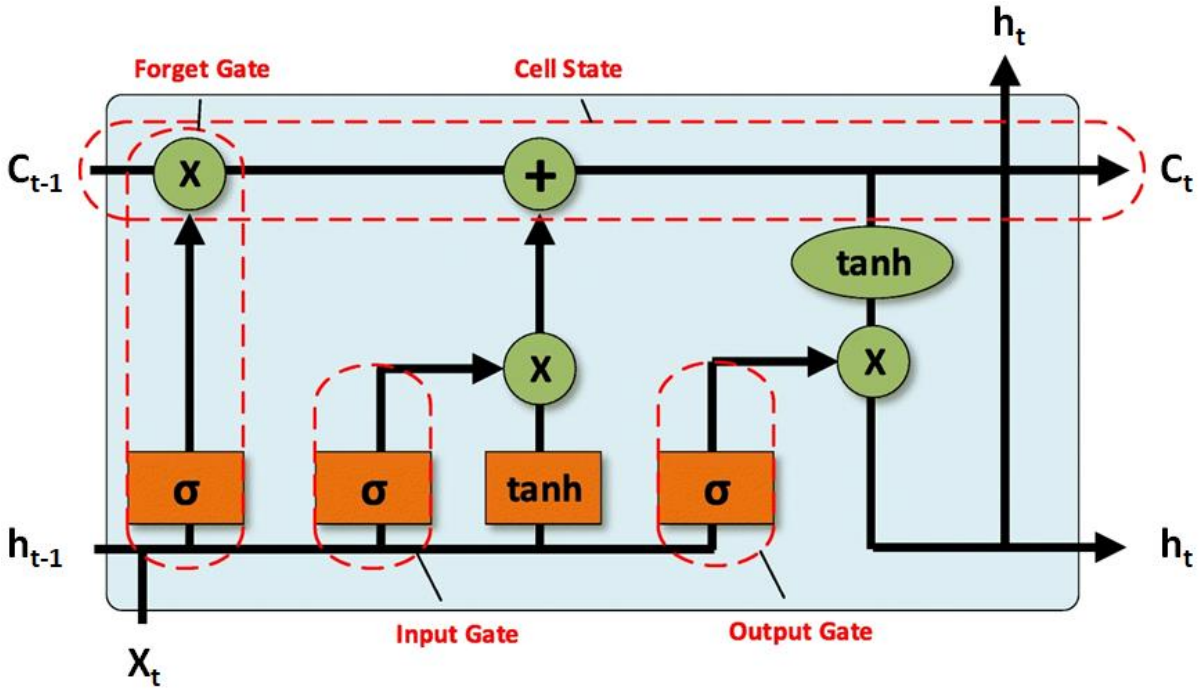



Figure 27: LSTM cell. Where "C" stands for cell state; "h" refers to hidden state; "$X_t$" is the input value. In orange are the sigmoid and tanh activation functions. In green are the multiplication, addition and tanh pointwise operations. Figure adapted from Burgess *et al.* (179).

In 2014, the gated recurrent unit (GRU) (180), a simpler version of the LSTM without output gate, was introduced.

There are several drawbacks associated with using RNNs, primarily stemming from their sequential data processing nature. This characteristic hinders the speed of processing since RNNs are challenging to parallelize effectively. Despite the advancements brought by LSTM cells, the inherent recursive nature of RNNs continues to pose difficulties in effectively capturing and accommodating long-range dependencies in data sequences.

**Transformers**:

Transformers were developed as a solution to address RNNs' limitations. In 2017, the pivotal paper "Attention is all you need" served as the starting point of transformers (181). The fundamental concept behind transformers is to present the entire time series data as a whole to the neural network, with each data point in the time series being tokenized. This novel approach utilizes a multi-head attention mechanism to process these tokens in parallel (Figure 28), effectively capturing both long and short-range dependencies in the data.

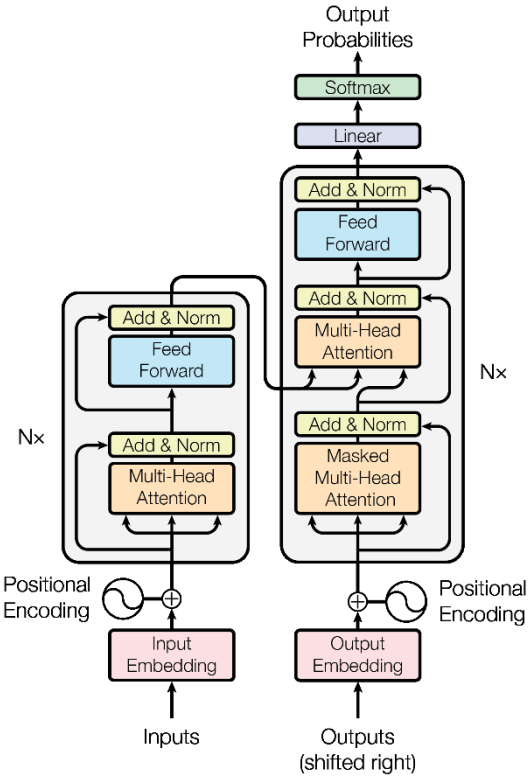



Figure 28: Architecture of a transformer. Figure taken from Vaswani *et al.* (181)

Transformers excel particularly with extensive data sets, prompting the development of large language models (LLMs) like the bidirectional encoder representations from transformers (BERT) (161) which was trained on 3.3 billion of words. Launched in 2018, BERT swiftly established itself as the state-of-the-art solution across a vast variety of NLP tasks, such as classification. Other LLMs emerged as well, including OpenAI's generative pre-trained transformer (GPT) models (182), which specialized in generative tasks and led to the renowned ChatGPT.

These achievements prompted the application of transformers in various fields, including computer vision. In 2020, the successful adaptation of transformers to image analysis was showcased in the paper "An Image is Worth 16x16 Words" (183). Vision transformers (ViT) employ a unique approach to image analysis, breaking down images into patches of 16x16 pixels, which are then tokenized similarly to words in traditional transformers (Figure 29).

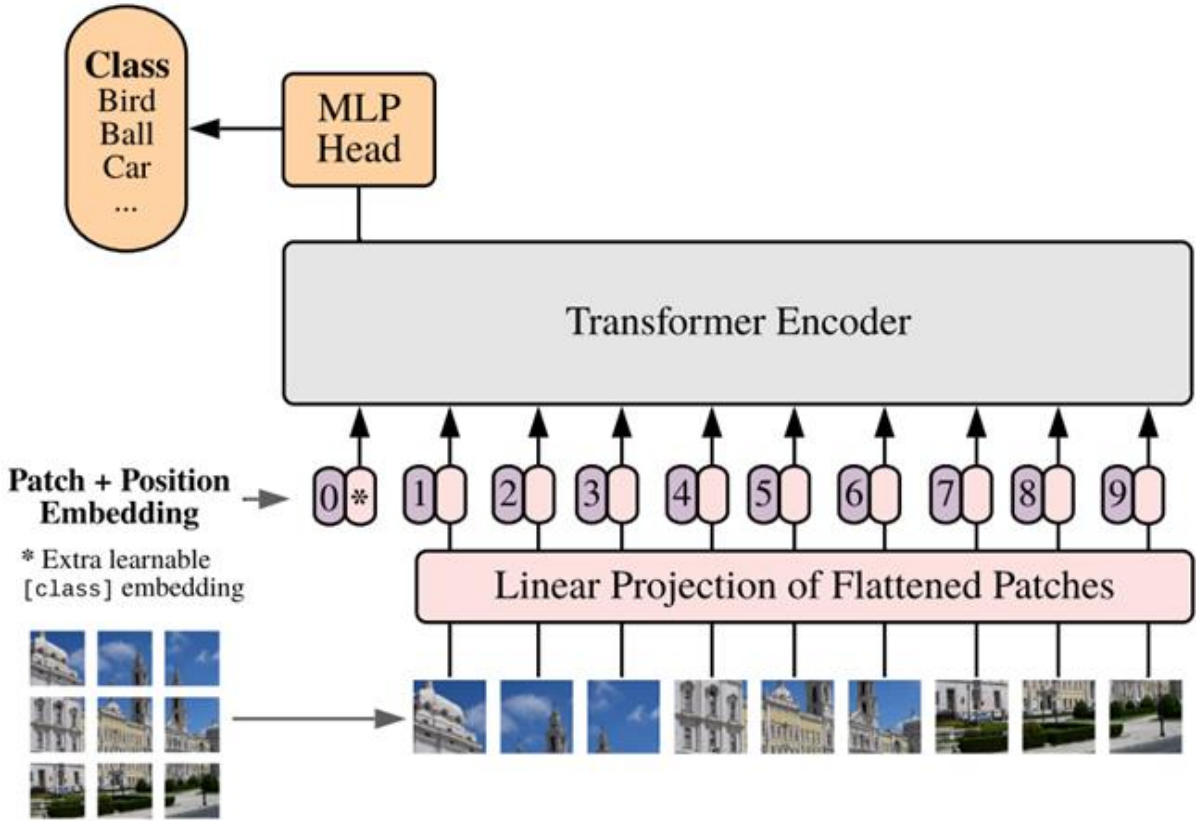



Figure 29: Overview of the vision transformer. Figure taken from Dosovitskiy *et al.* (183)

In contrast to CNNs, ViTs can grasp global and broader-range relationships, albeit demanding more extensive data for training. Depending on the situation, the choice between CNNs and ViTs may vary. Moreover, combining both architectures can also yield very good outcomes (184).

As it is possible to apply transformers on images, their application extended to videos as well (185, 186). Each frame of a video can be treated as a word in a sentence, allowing for their tokenization and their use as input for transformer (Figure 30 – A). Consequently, video transformer models can effectively classify videos for predicting sport activities. These models achieve this by discerning the relevant frames within a video for predicting the activity (Figure 30 – B).

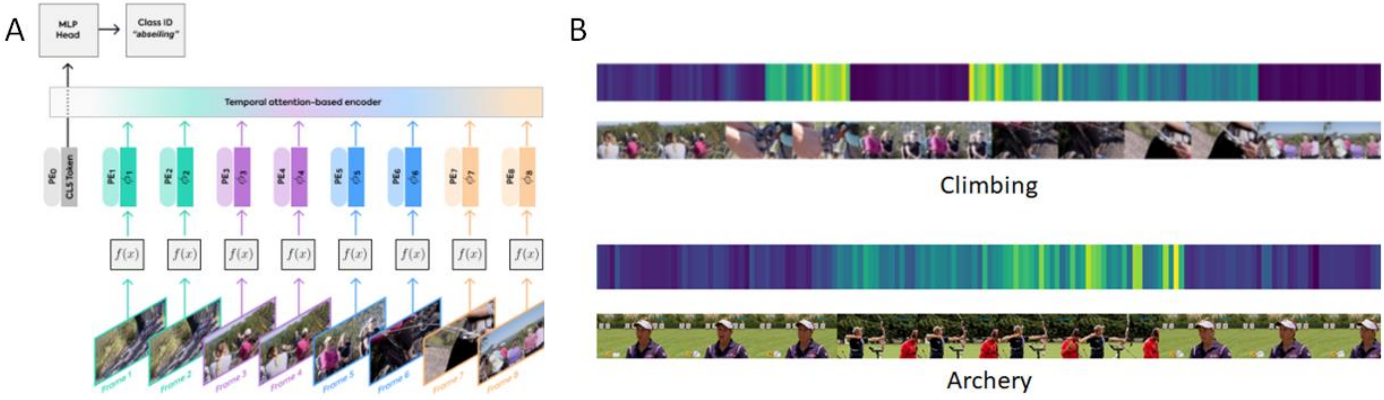



Figure 30: Video transformer. A – Model architecture. B – Activity predictions from videos. Figure taken from Neimark *et al.* (185)

### 2.4.4 Usage of deep learning

Deep learning is applied with success across various domains:

- **Computer vision**: Initially, the focus was ported on image classification, with applications on handwritten datasets like the MNIST dataset (187). Subsequently, more challenging tasks were undertaken, such as image segmentation – the ability to detect objects in an image and pinpoint their location. The YOLO model emerged as a real-time solution for video-based image segmentation (188).
- **Natural language processing**: application emerged in NLP with improvements in web search engines, vocal assistants (Siri, Google assistant, Alexa…), and conversational chat platforms (ChatGPT, Bloom (189)).
- **Image generation**: Generative algorithms have also started to be applied for image generation. Several tools have been released recently, including midjourney, DALE-2 (190) and stable diffusion (191). These tools are based on generative adversarial networks (GAN) and diffusion models (192), harnessing the power of these architectures to create realistic and diverse images.

Deep learning tools were also applied to various drug design tasks, including the generation of new molecules, the prediction of protein-ligand interaction poses, and the selection of molecules of interest.

#### *2.4.4.1 Generative DL*

The Plethora of deep learning tools offers multiple applications in the field of drug design. Here is a list of some DL architectures and their applications in the field:

- **Transformers**: They have been applied to accurately generate 3D models of proteins using only the amino acid sequence. This was exemplified by the remarkable success of AlphaFold2 (1) in the critical assessment of structure prediction (CASP) competition (193). LLMs based on GPT architecture have also been harnessed for protein design. For instance, ZymCTRL (194) generates novel amino acid sequences to create enzymes that are optimized for specific properties.
- **SE(3)-equivariant geometric DL**: In contrast to CNNs, these graph-based neural networks possess the advantage of being independent of rotations when working with 3D structures. This property has been leveraged to create innovative tools like EquiBind (195) and EquiDock (196), designed for conventional docking and protein docking, respectively.
- **Diffusion models**: Another implementation of a docking software, called Diffdock (197), has been developed with diffusion models. Furthermore, these models have been applied in fragment based drug design, with the development of Difflinker (198), a tool used to link fragments that have been docked into pockets.

While these methods showcase innovation, their recent implementation warrants thorough investigations into their limitations and potential enhancements. Notably, PoseBusters (199) sheds light on the limitations of previously mentioned docking software (195, 197, 200).

In the domain of *de novo* drug design, various types of neural networks have been employed:

- **LSTM** models (201, 202) have been used to generate molecules using SMILES (33) by training on extensive datasets of molecules, enabling them to learn general information, such as

SMILES grammar. These models are subsequently fine-tuned for specific targets through retraining on molecules known to interact with those targets, a process known as transfer learning.

- **Variational autoencoders** (VAE) have been employed to generate new molecules with suitable properties (203, 204). These models often work with molecular representations like SMILES or SELFIES (36). They are composed of an encoder and a decoder, both of which are RNNs. The encoder transforms molecules into a latent space, while the decoder converts latent space information back into molecules (Figure 31). Then, it becomes possible to navigate within the latent space, seeking areas with improved characteristics to generate optimized molecules.

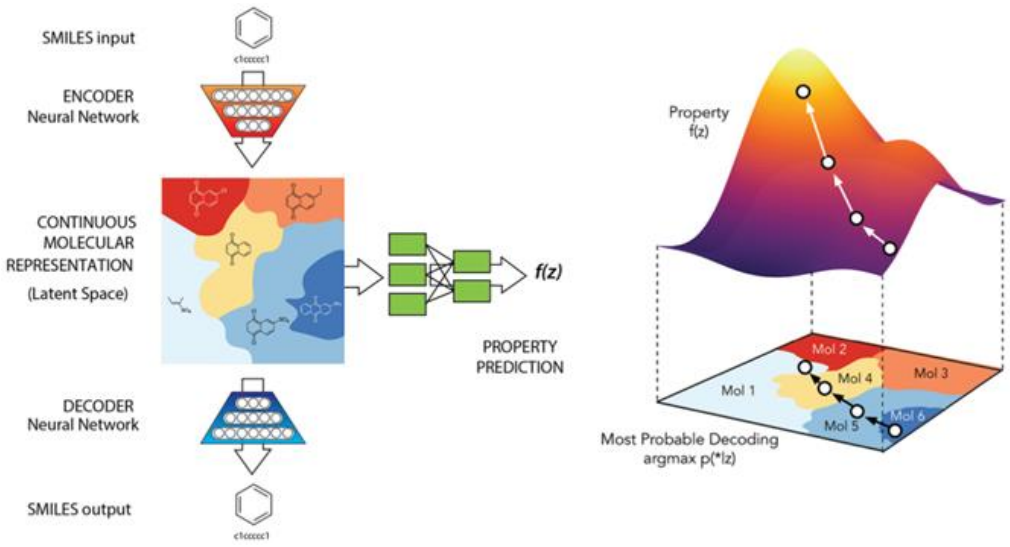



Figure 31: VAE applied to *de novo* drug design. Figure taken from Gómez-Bombarelli *et al.* (203)

- **Adversarial autoencoders** (AAE) have also been utilized for this purpose (205). Comprising autoencoders similarly to VAE, AAEs also include a GAN. The GAN consists of a generator and a discriminator (Figure 32). The generator aims to produce latent vectors that mimic those of real molecules. Both real and fake latent vectors are presented to the discriminator, whose goal to distinguish genuine from fabricated ones. During training, the objective is to enhance the generator until the discriminator can no longer accurately classify the latent vectors. Subsequently, the generated latent vectors are decoded to create new molecules.

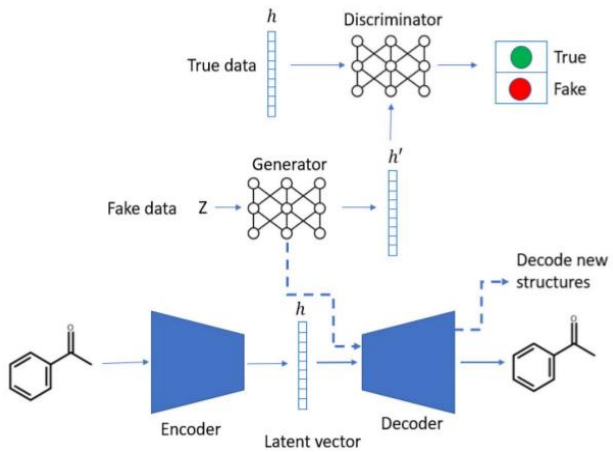



Figure 32: Application of GAN to generate molecules. Figure taken from Prykhodko *et al.* (205)

#### *2.4.4.2 Affinity prediction without structural information*

Similarly to QSAR, that aims to predict biological activities based on 2D chemical structures, drug-target (DTA) interaction prediction models do not use structural data to predict binding affinities. Although, these methods leverage information from both proteins and ligands to carry out predictions. However, they do not have access to the interactions between proteins and ligands. As they do not rely on structural data, these neural networks can potentially be trained on larger datasets compared to structure-based binding affinity models.

##### 2.4.4.2.1 Datasets

Among the available non-structural datasets, we can mention the Davis dataset (206), which is composed of 72 known kinase inhibitors and 442 target kinases. The binding affinities are reported in $K_d$ units, ranging from 5 to 10.8 $pK_d$ (207).

Another dataset is the KIBA dataset (208), which stand for kinase inhibitor bioactivity data. It comprises 52,498 molecules, 467 kinase targets and 246,088 experimental affinity measurements. The affinity data have been originally measured in $K_i$, $K_d$ and $IC_{50}$, and subsequently converted into a KIBA values for homogeneity. The KIBA score spans from 0 to 17.2. Alternatively, a modified version of the dataset retains only pairs with a minimum of 10 measurements (209). This curation results in a dataset comprising 2,116 molecules and 229 targets, with 24.4% coverage of the protein-ligand matrix.

Compared to previous datasets that are specialized for kinases, the BindingDB is a generalist dataset (210, 211). As of now, it contains 2.8 million affinity predictions for 1.2 million compounds and 9,200 targets. The data can be in the form of $IC_{50}$, $K_i$, or $K_d$ values, and can be accessed at the following link: https://www.bindingdb.org/rwd/bind/info.jsp

##### 2.4.4.2.2 DL models

Neural networks have been developed to learn from protein sequences and SMILES notations to predict binding affinities. An early neural network, DeepDTA (212) emerged as one of the pioneering implementations capable of surpassing baseline models like KronLRS (213) and SimBoost (209). DeepDTA employs a 1D-CNN to process the string representations. Subsequently, the processed data is fed to an MLP. It was trained through 5-fold cross-validation and underwent evaluation on both the Davis and KIBA datasets.

Other neural networks, such as AttentionDTA (214) and DgraphDTA (215), were built upon the DeepDTA framework. AttentionDTA introduced an attention mechanism, while DgraphDTA directly operates on graph representations. These adaptations led to improved performance on the Davis and KIBA datasets.

These models are learning patterns that allow distinguishing between molecules with high or low affinity for specific proteins. Although, they tend to have a limited application domain, as illustrated by their specific focus on predicting kinase inhibitors. This limitation contrasts with structure-based affinity prediction, where models should benefit from learning on interactions between molecules and proteins.

#### *2.4.4.3 Structure-based affinity predictions*

Let us delve into the application of deep learning for structure-based binding affinity predictions. They can be assimilated to the ML-SF as they can analyse the 3D structures of complex to predict binding affinities. These models can be trained and evaluated on several datasets:

##### 2.4.4.3.1 Datasets

**Binding MOAD**:

The Binding MOAD (mother of all database) (216) is a database comprising the 3D structures of protein–ligand complexes curated from the protein data bank (PDB) (39). Having started two decades ago, it has been receiving yearly updates. It has now came to an end with its final update in 2023, encompassing a total of 41,409 structures with affinity coverage for 15,223 (37%) complexes (217). The selection of complexes followed specific criteria, including:

- The atomic resolution of complexes must not exceed 2.5 Å.
- Ligands must be biologically relevant, which includes cofactors and peptides (less than 10 amino acids).
- Covalently attached molecules and crystallization agents are excluded.

**PDBbind:**

The PDBbind (218) is the most commonly used dataset to train DL models. It is composed of experimentally obtained 3D structures of protein-ligand complexes with known binding affinities. This dataset was initially released in 2004 with 2,276 complexes. In its latest version (v.2020) it comprises 19,443 complexes. These 3D structures are retrieved from the PDB and undergo additional processing. For instance, PDBbind provides the biological assemblies, and the complexes are protonated at a neutral pH.

The affinities were retrieved through literature review, prioritizing measurements conducted at room temperature and neutral pH. In cases where multiple measurements exist for a given complex, $K_d$ values were retained over Ki values, which in turn were prioritized over IC50 values. Overall, the binding affinities span over 2 to 12 $pK_i$/$pK_d$/$pIC_{50}$.

The entire dataset is referred to as the "general set". Subsequently a subset of 5,316 (version 2020), called “refined set”, was created based on the following quality criteria (219):

- Crystallographic structures with a resolution of 2.5 Å maximum and R-factor < 0.250.
- Complete ligands and pockets, with no missing atoms and without steric clashes with the protein.
- Noncovalently bound complexes and the absence of nonstandard residues within a distance of 5 Å from the ligand.
- The absence of other ligands within the binding site, such as cofactors or substrates.
- Binding affinity assessed in $K_i$ or $K_d$, falling within the $pK_i$ range of 2 to 12.
- Ligands with a molecular weight bellow 1000, and peptides with less than 10 residues.
- Ligands solely composed of the following atoms: C, N, O, P, S, F, Cl, Br, I, and H.

- The ligand's buried surface area exceeds 15% of the total surface area of the complex.

The "refined set" encompasses a subset known as the "core set", which is also commonly referred to as the “Comparative Assessment of Scoring Function (CASF)” set. Figure 33 provides a visual representation of the three sets within PDBbind.

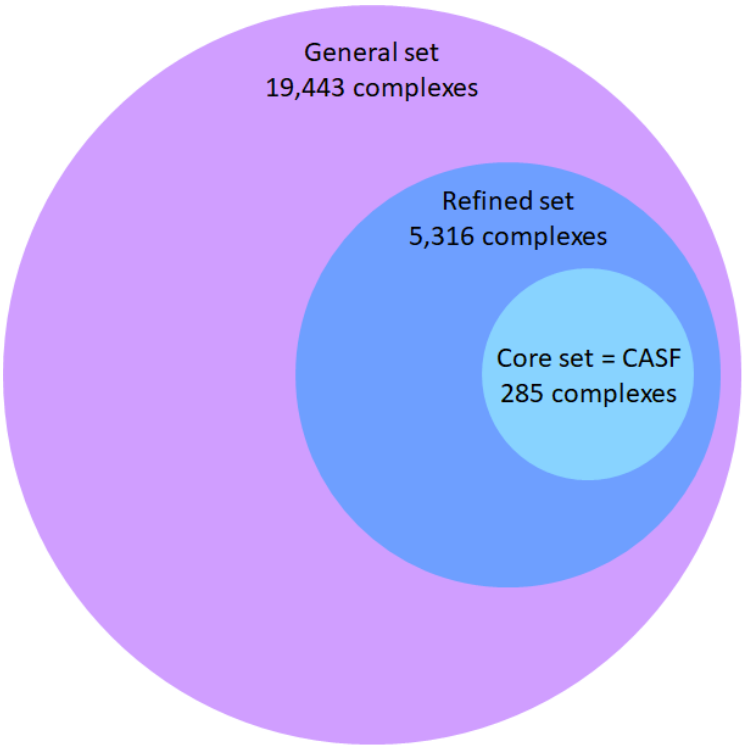



Figure 33: PDBbind v.2020 sets and their corresponding sizes.

**CASF:**

We previously mentioned that dataset for the benchmarking of SFs (Figure 19). In the same fashion, structure-based affinity predictions models are evaluated on this dataset. It was initially introduced in 2007 (220), followed by two subsequent releases: the CASF-2013 and CASF-2016.

In its version 2016, it is composed of 285 protein-ligand complexes with experimentally determined affinities (8). Based on a 90% sequence homology, the complexes are split into 57 clusters, with each cluster containing 5 complexes. Consequently, all the complexes within a cluster have proteins belonging to the same family. Each cluster displays a wide range of binding affinities, spanning differences of up to 8 $pK_i$ (Figure 34).

However, it is important to note that this test set is susceptible to bias, often leading to excessively optimistic performance when models are trained on the rest of the PDBbind (9). For instance, about 30% of CASF’s ligands can be found in the remaining part of the PDBbind v.2019 dataset, along with all its proteins (221). Therefore, it is commonly recommended to evaluate model performance using additional test sets.

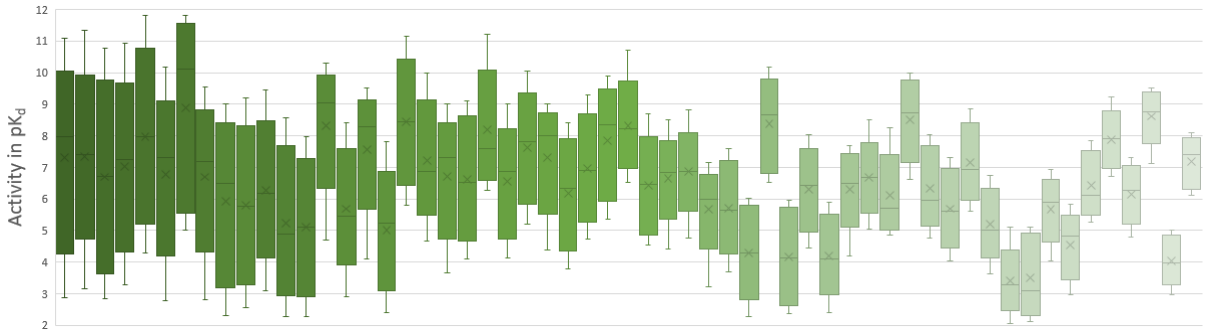



Figure 34: Experimental activity of CASF complexes for the 57 clusters sorted by affinity range

**CSAR**:

Around the period of 2010-2014, the Community Structure-Activity Resource (CSAR) have strived to establish docking and scoring benchmark exercises (222-224). This initiative was further advanced by the Drug Design Data Resource (D3R) from 2015 to 2020 (225-228). CSAR's endeavours resulted in the creation of two benchmark sets, the CSAR-NRC HiQ1 and HiQ2 sets, comprising 176 and 167 complexes, respectively. These 3D structures were retrieved from the PDB, while the affinity data were sourced from the Binding MOAD and the PDBbind. Only $K_d$ and $K_i$ binding affinity measurement were selected, encompassing a range of 14 $pK_d$/$pK_i$. A comprehensive outline of the data curation workflow is accessible in the study of Dunbar *et al.* (222)

**Astex Diverse Set**:

The Astex Diverse Set (229) was conceived for the purpose of validating molecular docking methodologies. It is composed of 85 protein-ligand complexes, 74 of which have known binding affinities. To curate this dataset from the PDB, they followed several rules:

- Atomic resolution must be lower than 2.5 Å.
- No clash between ligands and proteins.
- No interaction with symmetry units.
- Ligands were required to be drug-like molecules with structural diversity.
- The structures had to be relatively recent.

The binding affinities are provided in various forms, including $K_d$, $K_i$, $K_m$ and $IC_{50}$.

**FEP dataset**:

Originally tailored to evaluate FEP methodologies (230), this dataset comprises protein-ligand complexes with known binding affinities. Notably, the ligands within this dataset share structural similarities among themselves, and they interact with the same protein while exhibiting different binding affinities. The dataset encompasses a total of 8 proteins, namely BACE, CDK2, JNK1, MCL1, p38, PTP1B, Thrombin, and Tyk2, along with 200 ligands. Correspondingly, each of these proteins are represented by a 3D structure sourced from the following PDB codes: 4DJW, 1H1Q, 2GMX, 4HW3, 3FLY, 2QBS, 2ZFF and 4GIH.

This dataset has already been utilized to evaluate deep learning models such as $K_{DEEP}$ (4), RosENet (231) and GraphDelta (232). It proves to be a useful resource to ascertain a model's capability to discriminate similar ligands with different binding affinities, thereby evaluating its understanding of activity cliffs.

#### 2.4.4.3.2 DL models

As showcased in Table 1 and Table 2, a variety of deep neural networks have been implemented to enhance our ability to predict the binding affinity of protein-ligand complexes.

While the tables provide a means to compare models' performance, caution should be exercised when interpreting the presented performance values:

- Most of these models underwent training using PDBbind. However, the models were trained on distinct versions of the dataset, with varying amount of data. Furthermore, the models underwent training using different subsets of the PDBbind. For example, certain models were trained on the “refined set” comprising higher-quality complexes, whereas other models were trained on the larger-sized “general set”. In certain instances, neural networks such as GraphBAR (6) underwent training using datasets that were augmented through the incorporation of docking poses.
- While most of the data were evaluated on the same test set, namely the CASF-2016, it is worth noting that they were assessed using two different versions of it. Initially, it contained 290 complexes, but it was later revised to 285 complexes. Furthermore, certain models were evaluated using their own data splits. Additionally, some models were partially trained on the test set complexes; for instance, a part of the CASF-2016 was used to train the “Pair” model (233).
- In alternative cases, as observed with AK-Score (234), the results were presented by employing a consensus methods. In this approach, an ensemble model is established by averaging the predictions of model replicates, which amounted to 30 in the case of AK-score, for each complex. This method mechanically increases the performance.

| Model | Type | Objects | Descriptor | Training set | Test set | Split | R*p* | RMSE | Reference |
|---|---|---|---|---|---|---|---|---|---|
| TNet-BP | CNN | PL | Topological fingerprints | PDBbind 2016 refined (n=3767) | PDBbind 2016 core (n=290) | PDBBind original | 0.810 | 1.34 | Cang and Wei (235) |
| ACNN | CNN | P, L, PL | Atom type-labelled distances (Nat*25 atom types*12 closest neighbors) | PDBbind 2015 refined (n=2965) | PDBbind 2015 refined (n=741) | Temporal | 0.727 | - | Gomes, Ramsundar (236) |
| Brendan | CNN | PL | 3D grid (21*21*21 Å) * 256 bit SPLIF vector | PDBbind 2016 general (10000) | PDBbind 2016 general (1500) | Random | 0.704 | - | Lau and Dror (237) |
| PotentialNet | GNN | P, L, PL | Protein-ligand graph | PDBbind 2007 refined (n=1095) | PDBbind 2007 core (n=195) | PDBBind original | 0.822 | 1.39 | Feinberg, Sur (51) |
| $K_{DEEP}$ | CNN | L, PL | 1-Å 3D grid (25*25*25 Å) * 16 features | PDBbind 2016 refined (n=3767) | PDBbind 2016 core (n=290) | PDBbind original | 0.820 | 1.27 | Jiménez, Škalič (4) |
| Pafnucy | CNN | L, PL | 1-Å 3D grid (21*21*21 Å) * 19 features | PDBbind 2016 general (11906) | PDBbind 2016 core (n=290) | PDBBind original | 0.780 | 1.42 | Stepniewska-Dziubinska, Zielenkiewicz (3) |
| DeepATom | CNN | PL | 1-Å 3D grid (25*25*25 Å) * 24 features | PDBbind 2016 refined (n=3390) | PDBbind 2016 core (n=290) | PDBbind original | 0.807 | 1.32 | Li, Rezaei (238) |
| DeepBindRG | CNN | PL | Ligand (84) * protein (41) atom pair distances < 4 Å | PDBbind 2018 general (n=13500) | PDBbind 2018 general (n=925) | Random | 0.593 | 1.50 | Zhang, Liao (239) |
| OnionNet | CNN | PL | Atom type-labelled distances (Nat*25 atom types*12 closest neighbors) | PDBbind 2016 general (n=11906) | PDBbind 2016 core (n=290) | PDBbind original | 0.816 | 1.28 | Zheng, Fan (5) |
| RosENet | CNN | PL | Voxelized Rosetta interaction energies + pharmacophoric descriptors | PDBbind 2016/2018 refined (n=4463) | PDBbind 2016 core (n=290) | PDBBind original | 0.820 | 1.24 | Hassan-Harrirou, Zhang (231) |
| graphDelta | GNN | L, PL | One-hot encoded ligand atoms + protein environmental descriptors (373) | PDBbind 2018 general (n=8766) | PDBbind 2016 core (n=285) | PDBbind original | 0.870 | 1.05 | Karlov, Sosnin (232) |
| AK-Score | CNN | PL | id $K_{DEEP}$ | PDBbind 2016 refined (n=3772) | PDBbind 2016 core (n=285) | PDBBind original | 0.827 | 1.22 | Kwon, Shin (234) |
| SE-OnionNet | CNN | PL | 1-Å grid (21*21*21)* 64 protein-ligand element distance counts | PDBbind 2018 general (n=11663) | PDBbind 2018 refined (n=463) | Random | 0.853 | 1.59 | Wang, Liu (240) |
| Bypass multitask network | MLP | P, L, PL | Ligand ECFP + Protein ECFP + Protein-Ligand SPLIF | PDBbind 2016 refined (n=3568) | PDBbind 2016 core (n=290) | PDBbind original | 0.735 | 1.09 | Xie, Xu (241) |
| ACNN | CNN | P, L, PL | Atom type-labelled distances (Nat*25 atom types*12 closest neighbors) | PDBbind 2015 refined (n=3706) | PDBbind 2015 core (n=195) | PDBBind original | 0.730 | - | Yang, Shen (242) |
| Pair | MLP | PL | protein-ligand distance pairs | PDBBbind 2018 refined (n=2675) | PDBBbind 2018 refined (n=891) | Random split | 0.660 | 1.61 | Zhu, Zhang (233) |
| DEELIG | CNN | L, PL | Atomic model: 3D grid (10*10*10 Å) * 19 bits (atomic model); Composite model: 3D grid (10*10*10 Å) * 44 bits (pocket) + 14716 bits (ligand) | in-house set (n=4041) | PDBbind 2016 core (n=290) | Random 80/10/10 | 0.889 | - | Ahmed, Mam (243) |
| Interaction GraphNet | GNN | P, L, PL | Independent GNN for intra and inter-molecular interactions | PDBbind 2016 general (n=10366) | PDBbind 2016 core (n=290) | PDBBind original | 0.837 | 1.22 | Jiang, Hsieh (244) |
| midlevel fusion | CNN+GNN | PL | CNN: 1-Å grid (48*48*48)* 19 atomic features; GNN: covalent ( d < 1.5 Å) and | Pdbbind 2016 general + refined (13283) | PDB2016 core set (n=290) | PDBBind original | 0.810 | 1.31 | Jones, Kim (245) |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | non-covalent edges (1.5 < d < 4.5 Å) | | | | | | |
| **Model** | **Type** | **Objects** | **Descriptor** | **Training set** | **Test set** | **Split** | *Rp* | *RMSE* | **Reference** |
| SMPLIP | RF+CNN | L, PL | IFP (140) + interaction distances (140) + SMF descriptors (2282) | Pdbbind 2016 general + refined (13283) | PDB2016 core set (n=290) | PDBBind original | 0.770 | 1.51 | Kumar and Kim (246) |
| OctSurf | CNN | PL | 1-Å 3D grid (64*64*64 Å) * 24 features/octant | PDBbind 2018 general (n=16126) | PDBbind 2016 core (n=285) | PDBBind original | 0.793 | 1.45 | Liu, Wang (53) |
| BAPA | CNN | PL | Protein-ligand interaction descriptors + 6 Vina terms | PDBbind 2016 refined (n=3689) | PDBbind 2016 core (n=285) | PDBBind original | 0.819 | 1.31 | Seo, Choi (247) |
| APMNet | GNN+GNN | P, L | 75 DeepChem atomic features | PDBbind 2016 general (n=11844) | PDBbind 2016 core (n=290) | PDBBind original | 0.815 | 1.27 | Shen, Zhang (248) |
| GraphBAR | GNN | PL | 13 features * 200 protein-ligand atoms | PDBbind 2016 general (n=11146) | PDBbind 2016 core (n=290) | PDBBind original | 0.764 | 1.44 | Son and Kim (6) |

Table 1: Structure-based deep neural networks to predict protein-ligand binding affinities. Table taken from Volkov *et al.* (9)

| Model | References | Architecture | Pearson's *r* | RMSE | *N* |
|---|---|---|---|---|---|
| — | Artemenko (249) | MLP | — | — | — |
| NNScore 2.0 | Durrant and McCammon (83) | MLP | — | — | — |
| BgN- & BsN-Score | Ashtawy and Mahapatra (250) | MLP | — | — | — |
| DLscore | Hassan, Mogollon (251) | MLP | — | — | — |
| PLEC-NN | Wójcikowski, Kukiełka (74) | MLP | 0.82 | — | 290 |
| Pair | Zhu, Zhang (233) | MLP | 0.75 | 1.44 | 285 |
| AEScore | Meli, Anighoro (252) | MLP | 0.83 | 1.22 | 285 |
| 1D2D-CNN | Cang, Mu (253) | CNN | 0.85 | 1.21 | 290 |
| GNINA | Francoeur, Masuda (254) | CNN | 0.80 | 1.37 | 280 |
| AK-Score | Kwon, Shin (234) | CNN | 0.81 | — | 285 |
| LigityScore1D | Azzopardi and Ebejer (255) | CNN | 0.74 | 1.46 | 285 |
| OnionNet-2 | Wang, Zheng (256) | CNN | 0.86 | 1.16 | 285 |
| SE-OnionNet | Wang, Liu (240) | CNN | 0.83 | — | 285 |
| SIGN | Li, Zhou (257) | GNN | 0.80 | 1.32 | 290 |
| GraphBAR | Son and Kim (6) | GNN | 0.78 | 1.41 | 290 |
| PLIG/GATNet | Moesser, Klein (258) | GNN | 0.84 | 1.22 | 272 |
| PIGNet | Moon, Zhung (7) | GNN | 0.76 | — | 283 |
| — | Berishvili, Perkin (259) | CNN / RNN | — | — | — |
| PointTransformer | Wang, Wu (50) | CNN + ATT | 0.85 | 1.19 | 285 |

Table 2: Non-exhaustive list of deep learning architectures for protein-ligand binding affinity prediction and their performance on the CASF-2016 scoring benchmark. Duplicate with Volkov's list were removed, unless different performance were reported. Table taken from Meli *et al.* (80)

The main three neural networks used to carry out binding affinity predictions are the MLP, CNN, and GNN:

#### 2.4.4.3.2.1 Multi-layers perceptron:

The application of NN the predict binding affinities traces back to Artemenko's pioneering work in 2008 (249). In this study, a NN architecture consisting of 9 neurons was employed. The dataset comprising 288 complexes, was divided into training, validation and test sets with an 80/10/10 ratio. It reached an RMSE of 1.77 on the test set. The training process was conducted using pre-calculated molecular descriptors, including:

- The count of close nonbonded contacts
- A score for 'metal-atom' interactions
- The number of flexible bonds
- Van der Waals interaction energy
- Electrostatic interaction energy

In recent years, MLP have been employed alongside diverse approaches to characterize molecules and their interactions. Notably, the use of PLEC fingerprints has been explored both individually (74) and in combination with ECFP fingerprints (60) to predict binding affinity. In the first study, PLEC fingerprints (refer to 2.1.3.3) were applied with a protein depth of 5, a ligand depth of 1, and a size of 65,536. They designed an MLP consisting of 3 layers, each containing 200 neurons. A model was trained using 12,906 complexes sourced from the PDBbind dataset, resulting in a correlation coefficient of 0.817. This level of performance is comparable to that of a linear model employing PLEC fingerprints.

In the second study, a combination of ECFP fingerprints (with a diameter of 6 and a size of 4,096) and PLEC fingerprints (with a protein depth of 10, a ligand depth of 2, and a size of 16,384) was used. The refined set from PDBbind v2020 was divided into sets with a ratio of 3:1:1 for training, validation, and testing. They implemented a 2 hidden layers DNN and achieved a coefficient correlation of 0.74. However, their performance was surpassed by simpler machine learning methods like RF.

In a similar approach, DLscore (251) was developed to predict binding affinities using 348 binding analyser (BINANA) descriptors (260). These descriptors characterize ligand and protein atoms in a range of 2.5 Å – 4 Å, encompassing various interactions, binding pocket flexibility, rotatable bonds, and more. DLscore functions as an ensemble of 10 distinct MLPs, each designed with a varying number of layers and neurons. These NNs were trained using a dataset of 3,191 protein-ligand complexes from the refined PDBbind v2016 dataset.

The "Pair" model (233) was developed using 4,458 complexes from the "refined set" of PDBbind 2018. This dataset was split into training, validation, and test sets in a ratio of 60/20/20. The parameters of protein and ligand atoms were assigned using the Amber and Antechamber programs, with the ff14SB and "generalized amber force field" (GAFF), respectively. Ligand atom charges were added with the AM1-BCC method. The neural network architecture comprises 5 hidden layers of 32, 16, 8, 4, and 2 neurons. A unique approach was taken by training a pairwise function that considers the contribution of atom pairs in predicting binding affinity.

Lastly, multi-task neural network were employed to predict binding affinities (241). Through grid featurization, they calculated intra-ligand and intra-protein ECFP features, alongside protein-ligand SPLIF fingerprints, salt bridge count, and hydrogen bond count for three distance bins: 0–2, 2–3, and 3–4.5 Å. The study featured three types of neural networks: multitask, progressive, and bypass networks (Figure 35). The multitask network concurrently predicts several values, while the progressive network consists of task specific NNs with shared weights. The bypass network serves as a

hybrid combination of the two. The tasks included predicting binding affinities and ranking redocked poses. The multitask and bypass networks are composed of a single hidden layer, while the progressive network comprises three hidden layers. They employed the CASF-2016, the "refined" set and the "general" set, comprising 285, 3,568, and 11,303 complexes, respectively. The sets were further split using an 80/10/10 ratio.

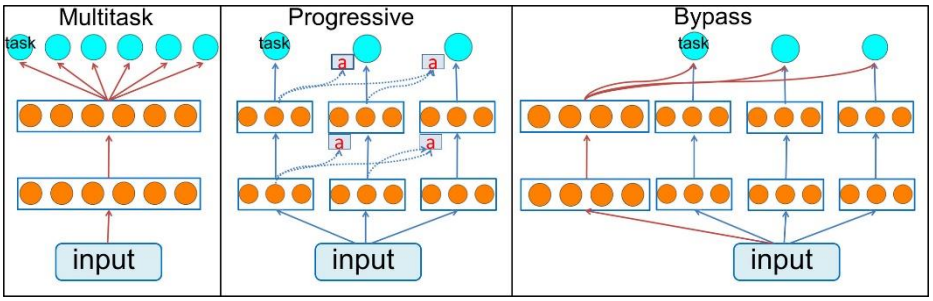



Figure 35: Multitasks used to predict binding affinities. Figure taken from Xie *et al.* (241)

#### 2.4.4.3.2.2 1D and 2D Convolutional neural networks:

To simplify the complexity of geometric data, certain approaches have been developed to analyse 1D or 2D vectors that describe 3D representations. For instance, in the case of TopologyNet (TNet-BP) (235), they employed the element-specific persistent homology method. This method represents the intricate 3D geometric structures using 1D topological invariants, effectively capturing crucial biological information through a multi-channel image-like representation. In their study, the researchers introduced a multi-task multichannel topological convolutional neural network (MM-TCNN). The architecture of this network is characterized by a few 1D convolutional and pooling layers, which are subsequently followed by multiple MLP. Each of these MLPs is designed to predict a specific task among the three following tasks: binding affinities, mutation-induced globular protein folding free energy changes, and mutation-induced membrane protein folding free energy changes. The model was trained using the refined set of the PDBbind 2016, consisting of 3,767 complexes, and its performance was assessed on the CASF-2016 dataset.

The binding affinity prediction with attention (BAPA) (247) employs descriptor embeddings as input, which carry information about the local structures of a protein-ligand complex. It utilizes 1D convolution on these descriptors and integrates an attention layer to identify crucial descriptors for accurate binding affinity prediction. The structure of the neural network includes convolutional layers, followed by attention layers and an MLP. Specifically, there are four convolutional layers with 3, 6, 6, and 9 filters, each of which is followed by a max pooling layer. The information is then flattened before being passed to the attention layers. The prediction task is undertaken by an MLP consisting of three layers, featuring 512, 256, and 128 neurons.

The OnionNet (5) has been designed to harness information from protein-ligand contacts occurring across various distance ranges. The 3D interaction data is transformed into a two-dimensional tensor using 60 concentric shells, initiated at 1 Å and incrementing by 0.5 Å up to 30.5 Å (Figure 36). Each of these shells characterizes protein-ligand atom contacts, resulting in a total of 3,480 features, which are then reshaped into a 2D matrix of dimensions (60x64). The NN architecture consists of three 2D convolutional layers with 32, 64, and 128 filters, sequentially followed by three fully connected (FC) layers containing 400, 200, and 100 neurons.

OnionNet-2 (256) was constructed using a similar methodology, but with a more coarse-grained approach. This time around, the contacts were calculated between ligands' atoms and proteins' amino acids. The NN architecture was also modified by using only two FC layers of 100 and 50 neurons.

In the case of SE-OnionNet (240), the fundamental OnionNet featurization methodology remained unchanged. However, they introduced squeeze-and-excitation (SE) blocks between the convolution layers. This technique, initially introduced in SE-Net (261), enables the model to capture relationships among various feature maps.

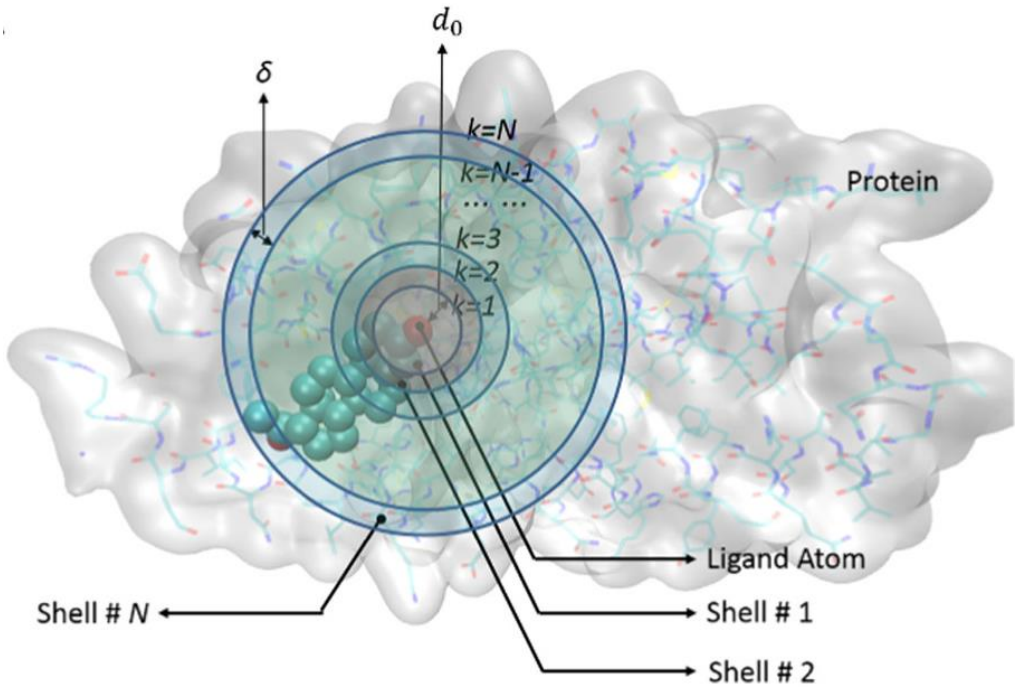



Figure 36: OnionNet featurization based on contact numbers in protein–ligand interaction shells. $d_0$ = 1 Å and δ = 0.5 Å. Figure taken from Zheng *et al.* (5)

#### 2.4.4.3.2.3 3D Convolutional neural networks:

In such case, convolutions are directly applied to the structure, albeit it necessitates voxelizing the space beforehand. This approach enables the extraction of useful information for carrying out predictions. The fundamental idea here is to avoid using expertly tailored descriptors that introduce bias into the model.

An early example of this approach is AtomNet (262), which was introduced in 2015 and aimed to discriminate between active molecules and decoys. AtomNet achieved an AUC greater than 0.9 on 57.8% of the DUD-E benchmark (263), a dataset comprising both active molecules and presumed inactive (decoy) molecules for 102 targets. This dataset is considered partially structural as it contains the 3D structures of proteins but not the complexes themselves. To evaluate AtomNet on DUD-E, they docked 22,886 actives and 50 decoys per active, allowing the utilization of a 3D-CNN for classification. Their approach involved generating a 20 Å cube over the binding pocket, which was subsequently voxelized into a 3D grid. Each voxel's edge measured 1 Å, and it contained vector of values indicating the presence of certain basic structural features in that specific location. The NN consists of four 3D convolutional layers with 128, 256, 256, and 256 kernels, followed by two fully connected layers, each with 1024 neurons.

Building upon this approach, Pafnucy (3) was introduced in 2017 to predict binding affinities. They employed a grid format similar to that of AtomNet, associating voxels adjacent to atoms with their corresponding atomic features. Each voxel was characterized by a 19-feature vector that encoded various information, including:

- Atom types (9 bits: B, C, N, O, P, S, Se, halogen and metal)
- Atom hybridation (1 integer: 1, 2, or 3)
- Number of bonds with heavy atoms and heteroatoms (2 integer)
- SMARTS patterns for hydrophobicity, aromaticity, acceptor, donor, ring (5 bits)
- Partial charges (1 float)
- Whether the atom was part of the ligand or the protein (1 integer: -1 or 1)

It comprises three layers of 3D convolutions with 64, 128, and 256 filters, followed by three FC layers with 1000, 500, and 200 neurons. Pafnucy has served as a reference point in numerous papers due to its simplicity and accessibility on GitHub. Additionally, Pafnucy's voxel-based featurization has been adopted in various research works (6, 53).

Another prime example is $K_{DEEP}$ (4) which was developed concurrently with Pafnucy. In $K_{DEEP}$, a 24x24x24 Å grid is utilized, where its 3D atomic features encompass hydrophobicity, aromaticity, acceptor, donor, positive ionizable, negative ionizable, metallic, and total excluded volume. This feature vector is duplicated to account for both the protein and ligand, resulting in a total of 16 distinct channels. $K_{DEEP}$ is built upon the architecture of SqueezeNet (264), a successful framework in the field of computer vision. Comprising 8 convolutional layers, each layer, starting from the 2nd, is constructed with two parallel convolutional layers whose results are subsequently merged. Except for the last convolutional layer, each convolutional layer is followed by a squeeze layer that reduce the number of channels in the feature maps. Both max pooling and average pooling layers are employed, and the final fully connected (FC) layer comprises 4,096 neurons. $K_{DEEP}$'s performance was assessed on various test sets, including CASF-2016, CSAR NRC-HiQ set 1 and set 2, as well as the FEP dataset.

More recently, new architectures with reduced computational costs have emerged, including OctSurf (53). In contrast to earlier examples that relied on protein-ligand structures, OctSurf employs molecule surfaces to predict binding affinities. The molecule's surface is initially represented as point clouds, which are subsequently voxelized into a 3D grid using the octree data structure (Figure 37 – A). This voxelization process follows a top-down approach, wherein voxels, called "octants" here, are subdivided into smaller units until reaching the desired octant size (Figure 37 – B). Octants that are not in contact with any atoms remain undivided. Convolutions are then exclusively applied to the smallest-sized octants. This strategy prevents convolutions from operating over empty space, leading to efficient utilization of computational power and memory resources. As a result, OctSurf can accommodate larger grids, such as 64x64x64 Å, while utilizing smaller voxel sizes. They have utilized 21 atomic features akin to those in Pafnucy, and for each octant, average values of surface points features have been assigned. These OctSurf representations serve as input with well-known neural network architectures, namely VGG (265) and ResNet (169), to predict binding affinities.

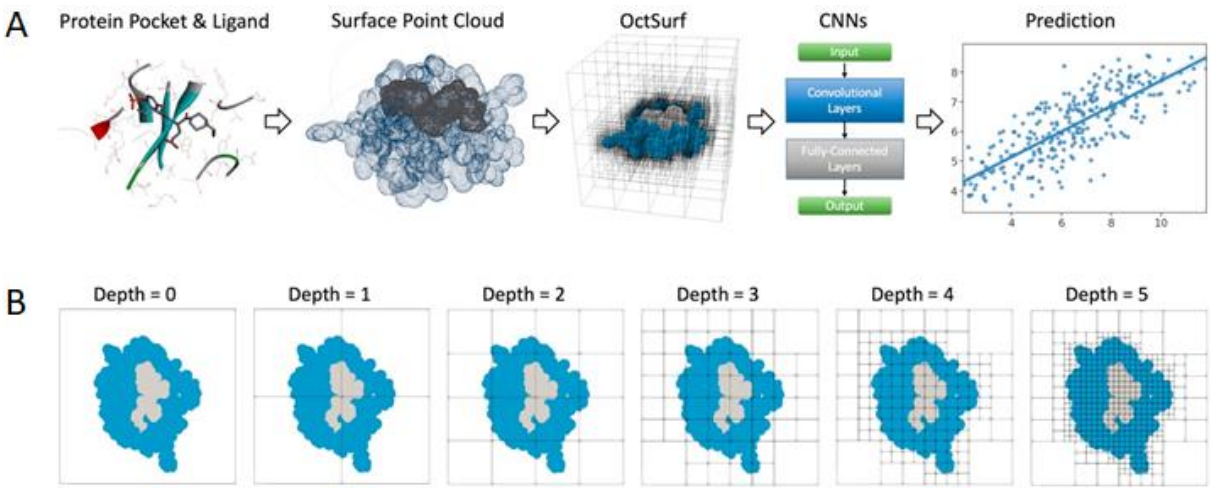



Figure 37: The operation of OctSurf. A – OctSurf workflow. B – Octree division process (2D as example). Figures taken from Liu *et al.* (53)

Until now, we have discussed methods used to predict absolute binding affinity (Table 2 and Table 3). Another application of 3D-CNN is to predict the relative binding affinities between closely related analogues, which is useful in a lead optimization process. A notable example in this context is DeltaDelta (266), a method developed in 2019. This example was not included in the previous tables. It employs a 24 Å grid with voxels of 1 Å per edge. The protein atoms are described by 8 pharmacophoric descriptors (similar to $K_{DEEP}$), while ligands are represented by 10 atom types (C, N, O, F, P, S, Cl, Br, I, H), which are combined into a vector of 18 channels. It is a two-branch convolutional neural network that takes two ligands as input simultaneously. Each branch consists of two convolutional layers with 32 filters, followed by a max pooling layer and another convolutional layer with 32 filters. The resulting feature maps are flattened and combined, before being processed by an FC layer of 192 neurons. DeltaDelta's performance was evaluated on datasets from Janssen, Pfizer, and Biogen, encompassing 3,249 ligands and 13 targets.

While CNNs are well-suited for analysing spatial information, they do face limitations when applied to the structural complexity of protein-ligand complexes. One such drawback is their lack of inherent rotational invariance, which can affect their robustness when predicting affinity across various orientations of complexes. To overcome this challenge, a common strategy is to train these NN on multiple rotations of the input data, often involving all the 24 possible rotations of the cube (90° rotations in all three dimensions).

Additionally, another limitation arises from the significant amount of empty space present in protein-ligand complexes. This emptiness leads to inefficient computational processes when performing convolutions over vacant regions, resulting in reduced computational efficiency. Moreover, this lack of informative data can significantly compromise predictive performance, as convolutions conducted on sparse matrices are prone to discard critical information that plays an important role in accurate predictions.

#### 2.4.4.3.2.4 Graph neural networks:

To circumvent these drawbacks, some research teams have opted to leverage graph data processing for their binding affinity predictions. By working directly with graph structures, neural networks can conduct computations efficiently, mitigating the challenges posed by empty spaces

encountered with 3D-CNN. Most of the time, these approaches are inherently invariant to rotations, preventing the need of using rotations during training.

GraphBAR (6) serves as an example of employing graph convolutional networks (GCN) for binding affinity prediction. The pockets were defined as protein atoms within a range of 4 Å from any ligand atom. Subsequently, a feature matrix of dimensions (200×13) was created. Complexes with a pocket-ligand pair exceeding 200 nodes were excluded, and for the rest, the matrix was padded with zeros up to 200 rows. The feature matrix represents atoms using 13 descriptors: 8 atom types (B, C, N, O, P, S, Se, Halogen, and Metal), atom hybridization, heavy-valence, hetero-valence, and partial charge. Additionally, several adjacency matrices were constructed to represent connectivity within the graph, accounting for intra and intermolecular interactions at different distances. The NN consists of three graph convolutional layers with 128, 128, and 32 filters. Between each pair of consecutive graph convolutional layers, a fully connected (FC) layer consisting of 128 neurons is inserted.

In their work, Volkov *et al.* (9) developed a Message Passing Neural Network (MPNN) that combines graph convolutional layers, Long Short-Term Memory (LSTM) units, and FC layers. The featurization process involved the following steps:

- Nodes were annotated to distinguish between protein and ligand atoms.
- Ligand nodes were labelled with their respective atom elements.
- For protein nodes, six types of pseudoatoms were introduced to represent aliphatic, hydrogen-bond acceptor, aromatic, hydrogen-bond acceptor and donor, hydrogen-bond donor, and metal atoms.
- Edges connecting protein nodes were retained if they are within a distance of less than 4.0 Å.
- Various noncovalent interactions, including hydrophobic, aromatic, hydrogen bonds, ionic bonds, and metal chelation, were calculated.
- Each edge was annotated with information about its corresponding bond length.

This featurization approach allowed the model to partially capture the spatial structure of the protein-ligand complex.

PIGNet (7) architecture employs a gated graph attention network (GAT) in conjunction with physics-informed equations to predict binding affinities. Here are the main steps involved in the PIGNet approach (Figure 38):

- **Node representation**: Atoms within the molecular complex are represented by node vectors with 54 dimensions. These vectors encompass various atomic properties, including atom type, hybridization, formal charge, and aromaticity.
- **Edge definition**: Edges connecting atoms are represented using adjacency matrices that are filled with the distances between pairs of atoms. Notably, two distinct adjacency matrices are established to distinguish intramolecular bonds from intermolecular interactions. When constructing the adjacency matrix for noncovalent interactions, only pairwise distances falling within the range of 0.5 to 5 Å are retained.
- **Node update mechanism**: This process begins with a gated GAT, followed by an interaction network, with each of them consisting of three units. Within the interaction network, there are max pooling and gated recurrent unit (GRU) layers. Both the gated GAT and interaction networks update node features using the covalent bonds adjacency matrix and intermolecular interactions adjacency matrix, respectively.

- **Physics-Informed Equations**: The updated node features are processed by four distinct physics-informed equations, each designed to account for specific types of interactions: van der Waals interactions, hydrophobic interactions, hydrogen bonding, and metal-ligand interactions. Furthermore, a rotor penalty term is introduced to take into account the entropic effects. This whole process enables the calculation of the energy component for each interaction between node pairs.

The total energy of a complex determined by summing up the atom-atom pairwise binding affinities.

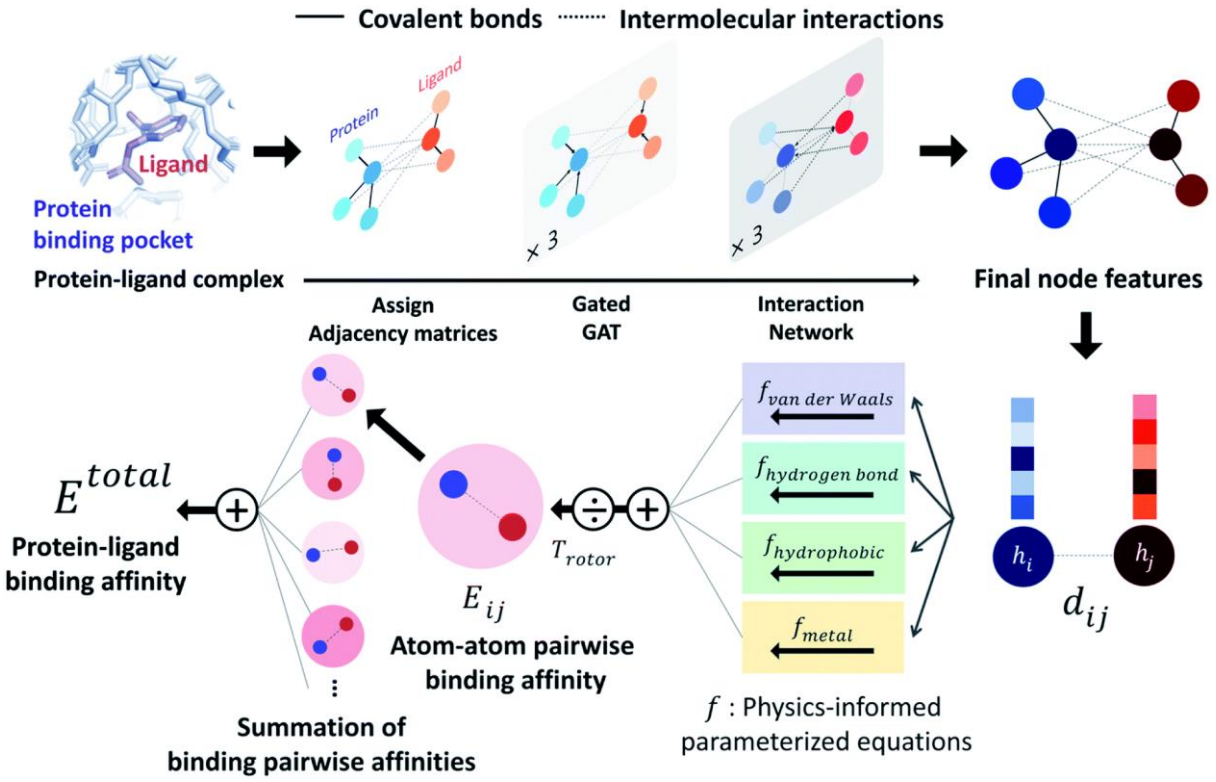



Figure 38: PIGNet workflow. Figure from Moon *et al.* (7)

DL has firmly established itself in the field of binding affinity prediction, owing to its remarkable performance and adaptability to a variety of molecular representations. A key advantage lies in its ability to extract useful information by itself, preventing the need of expert-crafted descriptors.

However, deep neural networks do possess certain limitations. Considered as “black boxes”, it is challenging to discern the rationale behind specific predictions. Nevertheless, the field of explainable AI is rapidly advancing in drug design through the application of various methods, including gradient-based approaches, masking techniques and layer-wise relevance propagation (LRP) (267). For instance, the LRP method has been applied to elucidate predictions made by transformer-CNN (135) for

QSAR/QSPR purposes (Figure 39 – A). Similarly, the masking method was used to explain 3D-CNN binding affinity prediction (267, 268) (Figure 39 – B). Enhancing model explainability is a crucial task, as it is essential to provide insights to chemists when selecting the molecules to synthetize in priority. A comprehensive overview of explainable AI is presented by Jiménez-Luna *et al.* (269).

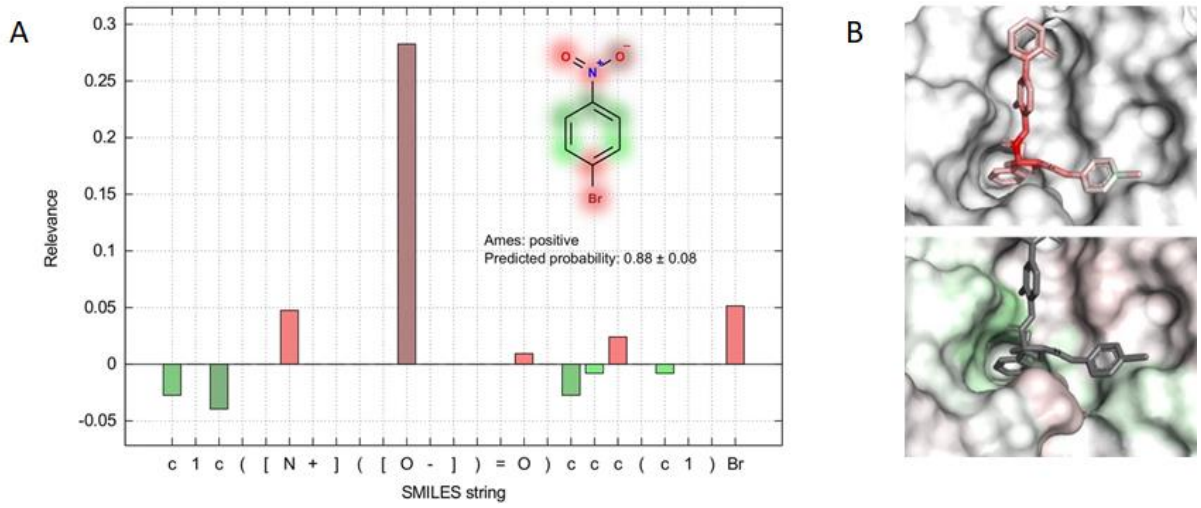



Figure 39: Explainable AI in drug design. A – LRP applied in QSPR. B – Masking method applied in affinity prediction with 3D-CNN. Figures taken from Karpov *et al.* (135) and Hochuli *et al.* (267).

While the high computational cost of DL is becoming more manageable with the availability of GPUs, there are still limitations when it comes to developing DL models. One major constraint is the substantial amount of data required to achieve both strong performance and robust generalizability. This presents a significant challenge in the field of structure-based drug design, primarily because it is not possible to rapidly increasing the volume of data, given that these structures are experimentally obtained.

An alternative to conducting a multitude of experiments is to augment the available data computationally. Several methods have been employed for this purpose, such as docking ligands into protein binding sites of complexes with known binding affinities to generate additional structural data. Another approach involves data imputation, where the binding affinity of experimentally determined structures is predicted and used to train new models. However, it is important to note that both of these methodologies can introduce biases into the data, potentially leading to misleading results.

Similarly to the first method, my PhD research focused on data augmentation by increasing the number of structures. However, we recognized that the behaviour of complexes is a dynamic phenomenon that cannot be simplified to a single static pose. Therefore, instead of solely increasing the quantity of complexes through docking, we explored the sampling of conformational states of complexes. To achieve this, we conducted molecular dynamics simulations on crystal structures with known binding affinities. By extracting frames from these simulations, the aim is to augment the dataset used to train statistical models, while providing the models with the temporal insights of protein-ligand interactions.

# 3 The influence of data on binding affinity predictions with neural network

The following part is a summary in French of the chapter “The influence of data on binding affinity predictions with neural network”.

Nous avons présenté les principaux outils utilisés en CADD, en mettant particulièrement l'accent sur les techniques appliquées au cours de cette thèse ou qui y sont connexes, notamment les méthodes basées sur la dynamique moléculaire et les modèles statistiques développés pour prédire l'affinité de liaison des complexes protéine-ligand.

La première démarche initiée dans cette thèse a consisté à reproduire les résultats des modèles précédemment publiés et à analyser l'impact des données sur leurs performances. Pour cela, nous avons choisi trois réseaux de neurones qui effectuent des prédictions d'affinité en analysant la structure des complexes protéine-ligand. Nous avons sélectionné deux CNN, Pafnucy (3) et OctSurf (53), ainsi qu'un GNN : GraphBAR (6).

Chacun de ces réseaux de neurones exploite différemment l’information contenues dans les structures.

- Pafnucy est appliqué directement sur la structure des complexes en discrétisant l'espace à l'aide d'une grille composée de voxels.
- OctSurf utilise une représentation de la surface des protéines basée sur la surface van der Waals des atomes. Cette surface est initialement décrite à l'aide d'un nuage de points, puis l'espace est voxelisé pour appliquer le processus de convolution 3D.
- GraphBAR traite l'information structurelle sous forme de graphe, où les nœuds correspondent aux atomes et les arêtes représentent les liaisons. Les informations sur les interactions sont contenues dans une matrice de voisins proches.

D'autres considérations ont également influencé la sélection de ces réseaux de neurones. Par exemple, Pafnucy bénéficie d'une reconnaissance étendue et est l'un des premiers 3D CNN développés pour prédire l'affinité. La description atomique utilisée par Pafnucy a été adoptée à maintes reprises dans des réseaux de neurones plus récemment publiés. Ainsi, Pafnucy constitue une référence par rapport à laquelle de nombreux modèles sont évalués.

En ce qui concerne OctSurf, il a été publié plus récemment et intègre une discrétisation de l'espace à l'aide de l'Octree. Cette méthode permet la création de voxels de tailles différentes, par exemple en regroupant les voxels présents dans des régions non pertinentes telles que le solvant, afin de réduire le nombre de convolutions nécessaires. Cela permet de diminuer le coût de calcul de l'entraînement des modèles à partir de CNN, ce qui constitue l'un des principaux points faibles de ces réseaux de neurones.

Dans cette étude, nous avons entrepris d'évaluer l'impact de divers facteurs sur les performances de ces modèles, en mettant particulièrement l'accent sur le traitement des données initiales. Nous avons ainsi examiné l'influence de la quantité et de la qualité des données utilisées pour entraîner les modèles. De plus, nous avons cherché à déterminer la taille de poche idéale pour obtenir les meilleures performances. Ensuite, nous avons exploré la possibilité d'entraîner des modèles impliquant un seul type de ligand, soit des petites molécules, soit des peptides. Puis, nous avons évalué la capacité des modèles à apprendre à partir des interactions moléculaires pour réaliser leurs prédictions. En

conclusion, nous avons souligné la nécessité d'une meilleure évaluation des modèles obtenus en utilisant des jeux de données externes.

Dans un premier temps, nous avons examiné l'importance de la quantité et de la qualité des données sur les prédictions. Cette question revient fréquemment lorsque des outils d'apprentissage automatique sont utilisés. Dans le domaine de la prédiction de l'affinité de liaison à partir des structures des complexes, il existe un débat pour déterminer la meilleure approche : faut-il entraîner les modèles statistiques sur un grand volume de données ou se limiter à des données de meilleure qualité ? Pour répondre à cette question, nous avons entraîné nos réseaux de neurones sur les jeux de données de la PDBbind (2), à savoir le jeu général et le jeu raffiné, comprenant respectivement 17 000 et 5 000 complexes. Le jeu raffiné est une sélection du jeu général, obtenue en appliquant des filtres de qualité. Par exemple, les complexes présentant des structures mal définies, avec une résolution supérieure à 2,5 Å, sont exclus. De plus, les complexes avec des mesures d'affinité en $IC_{50}$ sont écartés, car ces mesures sont souvent moins fiables que celles réalisées en $K_i$ ou $K_d$. Nos évaluations révèlent que Pafnucy produit de meilleures prédictions lorsqu'il est entraîné sur un jeu de données plus volumineux, même si celui-ci contient des données de qualité inférieure.

Pour réduire les temps de calcul, il est courant de se limiter à la poche où le ligand interagit avec la protéine, au lieu d'utiliser la structure en entier. Bien que la PDBbind fournisse une poche pour chaque complexe, nous avons préféré créer nos propres poches à partir des complexes de la PDBbind afin d'évaluer l'influence de la taille des poches sur les prédictions et de reproduire le processus pour réaliser des prédictions sur des données autres que celles de la PDBbind. Pour ce faire, nous avons créé des poches de cinq tailles différentes, en sélectionnant les acides aminés autour des ligands à une distance de 6 à 14 Å. Deux types de poches ont été utilisés : celles centrées sur le centre géométrique (CoG) du ligand et celles constituées par les acides aminés détectés autour de tous les atomes du ligand. Les poches CoG présentent l'avantage de conserver la même dimension quelle que soit la taille du ligand, mais certains ligands peuvent se retrouver coupés avec ces poches. Nous avons entraîné Pafnucy avec ces poches de différentes tailles et évalué les performances des modèles statistiques obtenus. Pour les 2 types de poches, il s'avère que les performances augmentent avec la taille des poches jusqu'à atteindre un plateau autour de 10 à 12 Å. Ainsi, il semble que des informations importantes soient contenues jusqu'à 10 Å autour des ligands, au-delà desquels l'agrandissement des poches n'apporte pas d'informations utiles. Étant donné que la plupart des interactions moléculaires sont effectives jusqu'à une distance de 6 Å, nous envisageons que le gain de performance observé jusqu'à 10 Å soit principalement dû à l'ajout d'informations biaisant les modèles en facilitant la reconnaissance des protéines impliquées dans le complexe. De plus, il est surprenant de constater que, bien que les poches CoG de 10 Å et les poches non CoG de 6 Å soient constituées du même nombre d'acides aminés, les modèles entraînés sur les poches CoG de 10 Å présentent des performances significativement meilleures.

Jusqu'à présent, nous avons entraîné uniquement des modèles globaux, mais il est également courant de développer des modèles locaux spécialisés sur un type spécifique de complexe impliquant une protéine en particulier. Dans notre cas, nous avons choisi de créer des modèles locaux entraînés sur un type de ligand. Pour ce faire, nous avons séparé les complexes impliquant des peptides et des petites molécules de la PDBbind, totalisant respectivement 2 900 et 14 500 complexes. Ensuite, après avoir entraîné Pafnucy sur ces jeux de données distincts, nous avons observé que les modèles entrainés sur un type de ligands obtiennent généralement de meilleures performances lorsqu'ils sont évalués sur ce même type de ligands, comparativement aux modèles entrainés sur l'autre type de ligands. Additionnellement, nous notons que les prédictions effectuées sur les peptides sont généralement

moins bonnes. Cette difficulté peut être attribuée au haut degré de liberté des liaisons peptidiques, ce qui rend la prédiction de leur affinité plus complexe.

Comme mentionné précédemment, il semble qu'il y ait des biais dans les prédictions d'affinité par les modèles statistiques obtenus par apprentissage automatique. Il semblerait que les modèles soit capable d'obtenir de bonnes performances sans analyser les interactions entre la protéine et le ligand. Pour déterminer cela, il est nécessaire d'entraîner des modèles en retirant soit le ligand, soit la protéine du complexe. Ainsi, il est possible de comparer les performances des modèles entrainés sur l'ensemble des complexes, c'est-à-dire avec les deux partenaires du complexe, ou seulement sur l'un des deux partenaires. Dans le cadre de cette étude, nous avons entrainé des modèles à partir des trois réseaux de neurones, Pafnucy, Octsurf et GraphBAR, ainsi qu'avec les complexes entiers, seulement les ligands et seulement les protéines.

En général, nous avons obtenu des performances significativement meilleures en entrainant les modèles sur l'ensemble des complexes. Néanmoins, les modèles entrainés avec un seul partenaire du complexe ont obtenu des performances très proches de celles obtenues en entrainant les modèles avec les deux partenaires. Cela indique que la majeure partie des performances ne provient pas de l'analyse des interactions protéine-ligand, mais plutôt de biais au sein du jeu de données. Ainsi, les modèles vont principalement apprendre que tel ligand ou telle protéine a tendance à former des complexes avec une faible ou une forte activité.

De tels biais dans l'apprentissage des modèles sont généralement révélés lors des prédictions sur les jeux d'évaluation. Cependant, le CASF 2013 et 2016 (220), qui sont les jeux de validation couramment utilisés en prédiction d'affinité, sont très similaires aux jeux d'apprentissage. Par conséquent, les performances élevées obtenues sur les jeux d'apprentissage restent élevées sur ces jeux d'évaluation. Ce biais dans les jeux de test ne permet donc pas de révéler la faible capacité des modèles statistiques à généraliser.

Nous montrons donc qu'il est nécessaire d'utiliser des jeux de validation externe pour mettre en évidence l'incapacité des modèles à apprendre ce qui est réellement important pour prédire l'affinité, c'est-à-dire principalement les interactions protéine-ligand.

Pour conclure cette étude, nous dressons une liste des méthodes et des jeux de données utiles pour évaluer la capacité à prédire l'affinité de liaison. Il existe quatre principaux types d'évaluations :

- « *Scoring power* » : la capacité à prédire la valeur d'affinité des molécules.
- « *Ranking power* » : la capacité à classer ces molécules par affinité.
- « *Screening power* » : la capacité à distinguer les molécules actives des leurres.
- « *Docking power* » : la capacité à retrouver la pose cristallographique parmi des poses leurres obtenues par amarrage moléculaire du ligand dans son site actif (*redocking*).

Pour les « *scoring* » et « *ranking powers* », certains jeux d'évaluation autres que le CASF peuvent être utilisés, tels que le jeu « Astex diverse » (229), les jeux « CSAR-NRC HiQ » 1 et 2 (222, 223), ou le jeu PDE10A de Roche (270). De plus, le jeu de données FEP (230) a été utilisé pour évaluer la capacité des modèles à prédire l'affinité de complexes, en particulier lorsque les ligands sont similaires et ciblent les mêmes protéines. Plus particulièrement, ce jeu d'évaluation permet de déterminer la capacité des modèles à détecter les "activity cliff" et à prédire les affinités en conséquence.

Malgré l'existence de ces jeux d'évaluation externe, de forts biais demeurent, rendant difficile la comparaison des performances des différents modèles. C'est pourquoi d'autres jeux d'évaluation ont

été établis pour mieux évaluer les modèles. Ces jeux ont été constitués en mettant à part certains complexes des jeux d'apprentissage habituels. Les complexes ont été sélectionnés de différentes manières, par exemple en les séparant selon une valeur de similarité de séquence ou d'homologie. Cela permet d'éviter d'avoir les mêmes protéines dans les jeux d'apprentissage et d'évaluation. D'autres méthodes de séparation sont également utilisées, telles que la séparation basée sur les squelettes de molécules pour éviter d'avoir des molécules similaires en commun, ou l'exclusion du jeu d'apprentissage de tous les complexes obtenus à partir d'une certaine date. Cette séparation temporelle du jeu de données permet une évaluation des performances plus réaliste, plus proche de l'utilisation réelle de ces outils. Néanmoins, chaque modèle est évalué à partir de jeux d'évaluation différents, ce qui complique la comparaison des performances de ces différents modèles.

Pour obtenir une évaluation robuste des performances des modèles, il est recommandé de les évaluer pour les quatre pouvoirs de *scoring*, de *ranking*, de *screening* et de *docking* (7). La CASF (8) et le jeu de données BigBind (271) ont été utilisés pour évaluer la capacité des modèles à distinguer quelles molécules sont actives ou inactives (*screening power*). Pour obtenir les poses des molécules inactives, les ligands ont été cross-dockés. Par exemple, dans le cas de la CASF, les ligands de la CASF ont été amarrés sur les autres protéines de la CASF, et ces poses ont été considérées comme ayant de très faibles affinités (inactives). Il existe également des jeux de données plus adaptés pour mesurer le screening power, tels que la DUD-E (263), DEKOIS 2.0 (272) et MUV (*maximum unbiased validation*) (273). Les deux premiers jeux de données sont composés de molécules supposées inactives, tandis que dans le MUV, l'affinité des molécules inactives a été déterminée expérimentalement.

Dans ces jeux de données, il n'y a pas de données structurales, il est donc nécessaire de réaliser au préalable un amarrage moléculaire de ces molécules actives et inactives avant de pouvoir utiliser des fonctions de score. En raison de cela, chaque modèle est évalué sur des poses d'amarrage moléculaire différentes, ce qui complique la comparaison des performances de ces modèles. En ce qui concerne l'évaluation des modèles pour le *screening power*, l'aire sous la courbe ROC (*receiver operating characteristic*) et le facteur d'enrichissement sont les métriques les plus couramment utilisées. En particulier, le facteur d'enrichissement permet de mesurer le pourcentage d'actifs retrouvés parmi les molécules ayant obtenu de fortes prédictions d'affinité.

Pour évaluer le *docking power*, il est nécessaire de redocker les ligands dans leur site actif. Les poses obtenues par amarrage moléculaire et situées à une distance de la pose cristallographique sont considérées comme mauvaises. Cette méthodologie de redocking a été réalisée sur la CASF 2013 et 2016 pour constituer des jeux d'évaluation du *docking power*. Ainsi, les modèles ont pu être évalués pour leur capacité à détecter les poses cristallographiques parmi les mauvaises poses générées par l'amarrage moléculaire.

Dans notre étude, nous avons établi des lignes directrices pour améliorer les performances des modèles de prédiction de l'affinité basés sur l'analyse des structures de complexes protéine-ligand. Nous avons examiné l'impact des données sur ces performances et avons entraîné des modèles spécifiques à certains types de ligands. Une observation importante est que les modèles utilisent peu l'information des interactions présentes dans les complexes, préférant s'appuyer sur les biais des jeux de données. Cela se traduit par des prédictions peu généralisables. Par ailleurs, les jeux d'évaluation usuels, comme la CASF, souffrent également de biais qui limitent notre capacité à détecter l'incapacité des modèles à généraliser. Ainsi, nous avons proposé une synthèse des différentes méthodes et jeux d'évaluation qui pourraient être utilisés pour évaluer de manière plus robuste les performances de ces modèles statistiques.

## 3.1 Influence of data on DL predictions

Prior to exploring the use of MD simulations for data augmentation, we initially focused on assessing the impact of data on binding affinity predictions. Although considerable research in this domain has concentrated on implementing innovative DL architectures for enhancing binding affinity predictions, there has been relatively little exploration into the effective utilization of the diverse neural networks. Therefore, our research aimed to uncover efficient ways to leverage existing data for improved binding affinity predictions using previously established neural networks.

Our objectives were to establish guidelines for binding affinity prediction, which included recommendations on the types of data to employ, efficient data utilization strategies, methods for benchmarking model performance, and more.

*Preprint of the article published in the International Journal of Molecular Sciences (IJMS) the 9th November 2023 (available: https://www.mdpi.com/1422-0067/24/22/16120)*

# The Impact of Data on Binding Affinity Predictions Using Deep Neural Networks

**Pierre-Yves Libouban [1], Samia Aci-Sèche [1], Jose C. Gómez-Tamayo [2], Gary Tresadern [2], and Pascal Bonnet [1,*]**

[1] Institute of Organic and Analytical Chemistry (ICOA); UMR7311, Université d'Orléans, CNRS; Pôle de chimie rue de Chartres - 45067 Orléans Cedex 2, France; pierre-yves.libouban@univ-orleans.fr

[2] Computational Chemistry, Janssen Research & Development; Janssen Pharmaceutica N. V.; B-2340 Beerse, Belgium; gtresade@its.jnj.com

[*] Correspondence: pascal.bonnet@univ-orleans.fr

**Abstract:** Artificial intelligence (AI) has gained significant traction in the field of drug discovery, with deep learning (DL) algorithms playing a crucial role in predicting protein-ligand binding affinities. Despite advancements in neural network architectures, system representation, and training techniques, the performance of DL affinity prediction has reached a plateau, prompting the question of whether it is truly solved or if the current performance is overly optimistic and reliant on biased, easily predictable data. Like other DL related problems, this issue seems to stem from the training and test sets used when building the models. In this work, we investigate the impact of several parameters related to the input data on the performance of neural network affinity prediction models. Notably, we identify the size of the binding pocket as a critical factor influencing the performance of our statistical models; furthermore, it is more important to train a model with as much data as possible, than to restrict the training on only the high-quality datasets. Finally, we also confirm the bias in the typically used current test sets. Therefore, several types of evaluation and benchmarking are required to understand models decision-making process and accurately compare the performance of models.

**Keywords:** protein-ligand; binding affinities; deep learning

## 1. Introduction

The importance of *in silico* work in the drug discovery pipeline has been growing for several decades. Since the 1980's, numerous drugs have been successfully marketed after being initially designed with the help of computers [1]. Approaches for computer-aided drug design, aiming to identify lead compounds, have steadily improved over time. One of the cornerstones of this process is the ability to accurately evaluate the binding affinity of protein-ligand complexes. To this end, various scoring functions, such as knowledge-based, empirical, and force field-based methods, have been developed [2]. The development of scoring functions has advanced further with the integration of machine learning models for bioactivity assessment. Recently, neural networks have gained attention for predicting binding affinity of protein-ligand complexes. With the advent of big data and the access to increased computing power, DL algorithms have emerged as promising tools for prediction purposes. These algorithms can reach state of the art performance for the prediction of the binding affinity. Despite these improvements and the implementation of new deep neural networks, the performance of the statistical models is stagnating [3].

The performance with DL algorithms relies heavily on the amount of data available to train the statistical models. Unfortunately, the amount of data available for the prediction of binding affinity is

relatively low in comparison to other application domains where DL has been successfully applied, like computer vision [4]. Indeed, for binding affinity predictions, models can be trained with the 3D structure of protein-ligand complexes, which are determined by crystallography, NMR or cryogenic electron microscopy (cryo-EM). On top of this, it is required to perform biophysical experiments, like surface plasmon resonance (SPR) or isothermal titration calorimetry (ITC), or more common biochemical assays, in order to evaluate the binding affinity of the complexes. All these experiments require extensive work therefore complicating the generation of new reliable data in this field.

We decided to evaluate the different variables related to the data to assess their impact on the performance. First of all, a crucial question is to evaluate the minimum amount of data necessary to achieve satisfactory performance. Would 10,000 complexes be enough or at least 100,000 are required etc.? To add to these considerations, it is important to keep in mind that increase in the data complexity, leads to higher data size requirements. This is especially true for 3D structural data, which are of higher complexity in comparison to most usual deep learning applications. The current state of the art models are typically trained on the PDBbind [5] dataset. This dataset comprises 3D structures of protein-ligand complexes with known binding affinity ($K_d$, $K_i$ or $IC_{50}$). In the case that several forms of binding data were available for a complex, $K_d$ was selected over $K_i$, and $K_i$ was selected over $IC_{50}$. This dataset contains 19,443 complexes in its current version (v.2020). Despite the size of the PDBbind increasing every year, having more data is not translated into better performance for the underlying models [3]. One of the main reasons is that the data lacks large series of molecules targeting the same protein, as well as having the same molecule in complex with several proteins. It is proposed that the sparsity of the protein-ligand matrix makes it harder for DL to learn from interactions. On top of this, some teams decided to focus on training on complexes of better quality instead of training on all the data available. To validate this approach, we analysed previously reported models trained on the whole PDBbind, and solely PDBbind's high quality subset known as the refined set. Furthermore, we have trained several models with Pafnucy [6], a well-known CNN for the prediction of binding affinities, on both datasets.

Protein-ligand complexes are dynamic, and the binding free energy as ligand passes from solvent to protein represents the energy difference between the ensemble of bound and solvated states. To accurately predict the binding affinity of a complex, several factors have to be taken into account like the association/dissociation kinetic constants as well as the interactions between the ligands and the proteins. Since we traditionally use static data, it is not possible to assess dynamic information such as the $k_{on}$ or the $k_{off}$. Therefore, the models are only based on partial information, they are single snapshots that although capture some experimentally favourable state, may still be incomplete. Since models use only the interactions between the ligands and the proteins, they are generally trained on proteins' pockets instead of using the whole protein. Pockets have already been calculated for the complexes contained in the PDBbind and are readily available when downloading the database. This removes the need for users to detect new pockets by themselves. Nonetheless, binding affinity will be impacted by conformational information from the ligand and protein local environment [3,7]. Therefore, pockets of different sizes can contain more or less information useful for getting performant models. Here, we investigated the impact of the pocket's size on the binding affinity prediction.

Other considerations related to the data are also investigated in this study. Notably, the difficulty to predict the binding affinity of peptides and the impact on the DL models performance of using a

training dataset including peptides or not. These difficulties stem from the higher degrees of freedom of peptides in comparison to small molecules. This leads to increased complexity of the entropic part when calculating free energies [8]. When training on the PDBbind, it appears that predicting the affinity of peptides is quite a challenging task. Therefore, some models have been developed by training only on nonpeptide ligands [9].

Another aspect pointed out in several recent publications [3,10] is related to DL models memorizing ligand and protein information instead of learning from the interactions. Here we have decomposed this, by training neural networks only on proteins or ligands and carrying out the prediction, to evaluate the bias in their predictions. We compared the performances of 3 well-known DL model predicting binding affinities, GraphBar, Pafnucy and OctSurf.

Overall, we find that it is important to train on as much data as possible, while even using complexes deemed of lower quality. Moreover, the size of the pocket does matter for the ability of the model to predict the binding affinity. The performance improves upon reaching a certain size (12 Å around the ligand); increasing pocket size further will not improve the performance. On top of this it is difficult to predict peptides, even by training only on peptides. Finally, we point out that there is a big discrepancy on the ability of neural networks to learn from the interaction. Some models will heavily drop in performance by removing one of the 2 partners from the complex, while others rely on the memorization of bias in the data to carry out a prediction.

## 2. Materials and Methods

### *2.1. Datasets*

The PDBbind dataset (http://www.pdbbind.org.cn) [5] was used to train the different models. It contains protein-ligand complexes with known binding activity. In its current version (v.2020), 19,443 complexes are available. In this publication, three versions of the PDBbind were used:

- The version 2016 that contains 13,308 protein-ligand complexes
- The version 2018 that contains 16,151 protein-ligand complexes
- The version 2019 that contains 17,679 protein-ligand complexes

The complexes present in the PDBbind are selected from the Protein Data Bank (http://www.rcsb.org/) [11]. Several modifications are added to these complexes, *e.g.*, the biological assembly of complexes are recreated, ligands' atoms and bonds are corrected; for the detail of all modifications, please refer to the "readme" provided with the PDBbind.

The PDBbind encompasses three sets of data: the general set, the refined set and the core set. The general set contains the totality of the dataset. The refined set is a subset made of 4,852 complexes (for the version 2019) selected based on the following quality criteria [12]:

- Crystallographic structures, with a resolution of 2.5 Å maximum
- Complete ligands/pockets (without missing atoms) and without steric clash with the protein

- Noncovalently bound complexes, no nonstandard residues at a distance <5 Å from the ligand
- No other ligands are present in the binding site, *e.g.*, cofactors or substrates
- Binding affinity evaluated in $K_i$ or $K_d$, and with a $pK_i$ between 2 and 12
- Ligands with a molecular weight of less than 1000, less than 10 residues for peptides
- With ligands made only of the following atoms: C, N, O, P, S, F, Cl, Br, I, and H
- The buried surface area of the ligand is higher than 15% of the total surface area of the complex

The core set is broadly used as a test set to compare models' performance. Only two versions are available, the version 2013 which is composed of 195 complexes [13,14] and the version 2016 comprising 285 complexes [15]. Both core set have 107 complexes in common. The core set 2016 is made of 57 clusters of 5 complexes belonging to the same protein family. These groups are obtained by clustering complexes based on sequence similarity of 90% minimum.

In this study, peptides were flagged among the ligands coming from PDBbind's complexes. We detected the peptides by looking for ligands having in their mol2 files at least one atom named "CA", "CB", "CD", "CE", "CG", "CZ", "CA1", "CA2", "CB1", "CB2", "CD1", "CD2", "CE1", "CE2", "CG1", "CG2", "CZ1" or "CZ2". On top of this, we analysed the PDBbind list of ligand names and flagged as peptides all the ligands containing "mer" in their name. Finally, ligands wrongly labelled as peptides were removed, by keeping only ligands matching with the following smart, which represent a peptide bond: [$([NX3H2,NX4H3+]),$([NX3H](C)(C))][CX4H]([*])[CX3](=[OX1])[OX2H,OX1-,N]. By doing so, we were able to detect 2,915 peptides in the PDBbind (v.2019).

We used the pockets provided by the PDBbind to evaluate the impact on the performance of:

- The dataset sizes (general set or refined set)
- The types of ligands (peptide or nonpeptide)
- Using only ligand or only protein

We also created our own pockets using Pymol. Residues around the ligands were selected to create pockets. The pockets were constructed with different sizes: 6 Å, 8 Å, 10 Å, 12 Å and 14 Å. Two types of pockets were created, by selecting residues at a specific distance from:

- All the atoms of the ligands
- The center of geometry (CoG) of the ligands (Figure 1)

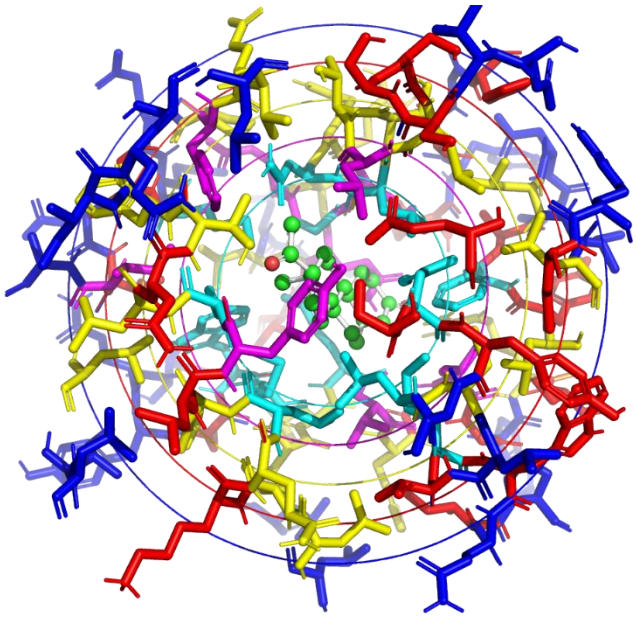


**Figure 1.** Pockets created and visualized with Pymol. The ligand is displayed in green. Residues are coloured in cyan, purple, yellow, red and blue, according to their distance from the CoG of the ligand, respectively at 6 Å, 8 Å, 10 Å, 12 Å and 14 Å.

*2.2. Neural networks*

Only previously published binding affinity neural networks approaches were used in this work. For the purpose of this study, we selected two convolutional neural networks (Pafnucy [6] and OctSurf [16]) and a graph convolutional neural network (GraphBAR [17]). Here, we briefly describe each of them. The full description of the neural networks can be found in the original publications.

Pafnucy is a 3D convolutional neural network published in 2018. It uses the 3D coordinates of atoms and performs convolutions on voxels of 1 $Å^3$. In this paper, we generally used boxes with edges of 21 Å, and modified the size of the box when different pocket sizes were used. 19 features were used to describe an atom:

- 9 bits (one-hot or all null) encoding atom types: B, C, N, O, P, S, Se, halogen and metal
- 1 integer (1, 2, or 3) with atom hybridization: hyb
- 1 integer counting the numbers of bonds with other heavyatoms: heavy_valence
- 1 integer counting the numbers of bonds with other heteroatoms: hetero_valence
- 5 bits (1 if present) encoding properties defined with SMARTS patterns: hydrophobic, aromatic, acceptor, donor and ring
- 1 float with partial charge: partial charge
- 1 integer (1 for ligand, -1 for protein) to distinguish between the two molecules: moltype

This neural network uses data augmentation by learning from systematic rotations of complexes. The systematic rotations are obtained by performing the 24 rotations of the cube on each structure. The data augmentation with systematic rotations allows the models to be more robust since the models are independent of the orientations of the ligands and the proteins.

Here is the reported performance of Pafnucy trained on the pockets provided by the PDBbind 2016:

- Core set 2013: correlation coefficient of 0.70 taken from Stepniewska-Dziubinska *et al.* [6].
- Core set 2016: correlation coefficient of 0.78 take from Stepniewska-Dziubinska *et al.* [6].

We replicated the results of Pafnucy by using the code available here: https://gitlab.com/cheminfIBB/pafnucy.

OctSurf is a 3D convolutional neural network published in 2021. It requires an elaborate data preparation before it can be used as input for the neural network. First, the 3D coordinates of atoms are turned into point clouds [18] representing their van der Waals surfaces. Then the point clouds are rasterized into an octsurf which is a volumetric representation based on octree data structure [19]. An octsurf is composed of octants, on which are performed the convolutions. The octants can have variable sizes. This allows for having octants of different sizes in the same octsurf, describing more or less precisely different parts of the octsurf. Therefore, it is possible to have big octants in the solvent and smaller ones (of 1 Å for example) in contact of the proteins and ligands. This way, we can accelerate the convolution process, while keeping good performance.

The description of octants uses the 19 features described in Pafnucy. On top of that 5 more features were added to reach a total of 24 features:

- The hydrogen atom type
- Van der Waals atomic radius
- A normal vector with three coordinates direction, describing surface curvature and shape complementarity

Data augmentation was performed by randomly rotating and translating the surface points, reaching 40 octsurfs for each complex.

In the publication, OctSurf reached a correlation coefficient of 0.79 [16] on the core set 2016 by training on the pockets provided by the PDBbind 2018.

The code of OctSurf is available here: https://github.uconn.edu/mldrugdiscovery/OctSurf

GraphBAR is a graph convolutional neural network published in 2021. Graphs were created with atoms as nodes, and bonds as edges. Node characterization reuses only 13 features established by Pafnucy, therefore not using the 5 properties encoded by SMARTS patterns (hydrophobic, aromatic, acceptor, donor and ring).

Bonds are summarized in an adjacency matrix having a size of NxN, with N being the number of nodes. In the adjacency matrix, the adjacent atoms are defined by a distance maximum of 4 Å for inter-molecular distances, and 2 Å for intra-molecular distances. It is possible to train the neural network with up to 8 adjacency matrices. If the number of adjacency matrices is increased, the distance range covered by each is reduced. For example, in the case of using only one matrix, this one would cover interactions up to 4 Å. While in the case of using two adjacency matrices, the first one would account for the interactions up to 2 Å, and the other one deals with the interactions from 2 to 4 Å. The model established with two matrices achieved the best performance.

For data-augmentation purpose, docking was performed and best poses with less than 3 Å of RMSD were selected, up to 3 poses.

GraphBAR was trained on the PDBbind 2016, while discarding the complexes (pocket + ligand) containing too many atoms (>200 atoms). The models achieved coefficient correlations of 0.76 on the core set 2016 and 0.70 on the core set 2013. The data-augmentation provided little improvements on the core set 2016 with a coefficient correlation of 0.78, and no improvement were measured for the core set 2013.

Performance was replicated using the code available here: https://github.com/jtson82/graphbar

We carried out each experiment by replicating the training 10 times. All model replicates were performed in the same conditions, *i.e.* with the same neural network, the same hyper-parameters, the same input data, but different weights (randomized seeds) at the initialisation of the neural network. The results were averaged and the standard deviation was calculated, in order to compare the performance of each experiments.

Models were trained with our laboratory cluster, on graphics processing unit (RTX 2080 and RTX 3090).

*2.3. Metrics*

The model performance was evaluated by predicting the binding affinity of each complex of the test sets and comparing the results with real values. Prediction error was measured with the root mean square error (RMSE).

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}{N}}$$

The correlation between predicted binding affinity and the experimentally measured binding affinity were assessed with the Pearson correlation coefficient (R) and its standard deviation (SD).

$$R = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \Sigma(y_i - \bar{y})^2}}$$

Statistical plots were performed with the library statannot (https://pypi.org/project/statannot/). All comparison were performed with independent sample Student t-test with Bonferroni correction. Following p-values correspond to the annotations on the plots:

```
    ns: 5.00e-02 < p <= 1.00e+00
     *: 1.00e-02 < p <= 5.00e-02
    **: 1.00e-03 < p <= 1.00e-02
   ***: 1.00e-04 < p <= 1.00e-03
  ****: p <= 1.00e-04
```

## 3. Results

### *3.1. Impact of the amount of data on the performance*

The 3D structures of protein-ligand complexes can be used to train statistical models in many ways. These data can be considered as 3D images, point clouds or as graphs. Several types of neural networks have been developed to handle these representations of the data, such as the convolutional neural networks (CNN) and the graph neural networks (GNN). The CNN are used on 3D images and point clouds, by discretizing the space in voxels of around 1 $Å^3$. Then they perform convolutions over these voxels to extract the meaningful information for the prediction of binding affinities. The GNN are used with graphs, which are usually constructed of atoms as nodes and bonds as edges. In the case of graph convolutional network, the useful information stored in nodes and bonds is extracted by performing graph convolutions.

To reach good performance with DL algorithms, it is expected that more data is beneficial and that a high amount of data is a requirement to begin. In PDBbind (v.2019), the general set contains 17,679 protein-ligand structures. The refined set is a subset of 4,852 complexes selected from the general set based on quality criteria. A previously published study suggested that training on the general set of the PDBbind does not improve the performance in comparison to training only on the refined set [3]. While other studies [20-22] pointed out that they achieved better performance by training on the general set rather than only on the refined one.

To explore this further, we have trained Pafnucy [6] with the PDBbind general set and with only the refined set. Pafnucy was set up to perform convolutions over voxels of 1 $Å^3$ and on a box with edges of 21 Å, centred on the ligand.

The models were applied to 2 test sets comprised of 285 and 195 complexes and referred to as core set 2016 and core set 2013. The complexes from the test sets were not used in training. Nonetheless as reported in GIGN [23], all the proteins and a third of the ligands from the test set are also used in the training set. Due to this, we can expect biased results when predicting on test sets. Analysing further these sets, we found out that the distribution of the molecular weight of ligands is similar for the test set and the training set (Figure A1). The same can be said about the shape of the ligands, although there is a lack of spherical ligands in the test set (Figure A2). In addition, ligands with extreme affinity are over-represented in the test set in comparison to the training set (Figure A3). This can be a possible explanation for why current networks [3] predict over a small affinity range and therefore tend to fail predicting extreme affinities values of the test set.

When assessing performance, we compare the correlation between predicted and experimental activity using the Pearson correlation coefficient (R).

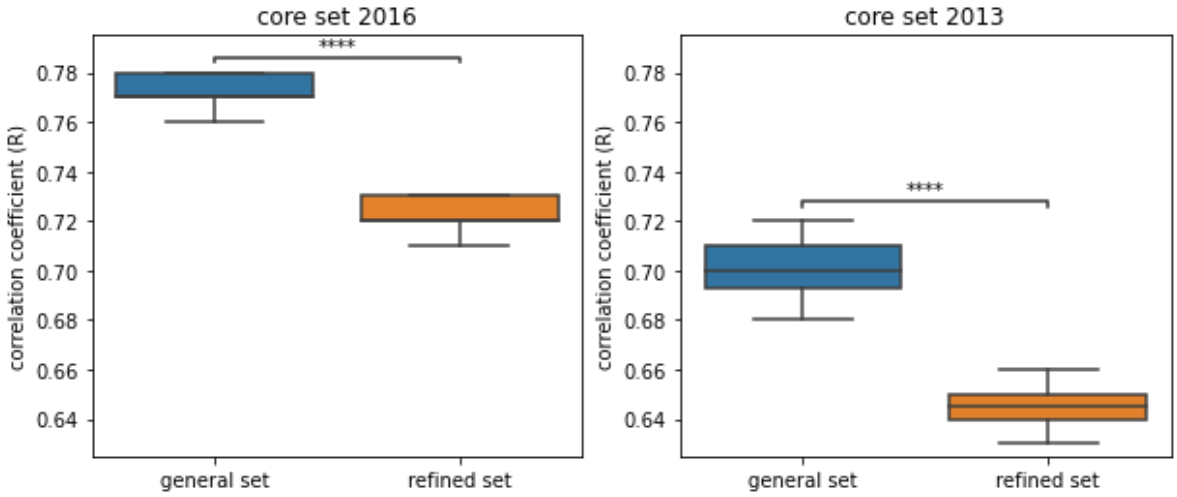



**Figure 2.** Comparison of the performance of Pafnucy [6], after being trained on the general or the refined set of the PDBbind 2019. 10 models were trained on each dataset. (**a**) Performance is evaluated on the core set 2016; (**b**) performance is evaluated on the core set 2013. For both core sets, performance by training on the general set significantly outperform the performance by training on the refined set.

Models trained on the general set perform better than the one trained on the refined set when applied to the frequently used test sets: core set 2016 and core set 2013, Figure 2. These results are in accordance with a previously published comparison of performance of 11 neural networks [20]. For all, these neural networks the RMSE and MAE (Mean Absolute Error) is lower when trained on the general set instead of the refined set. Likewise, the neural networks PointTransformer [21], DeepAtom [22] and the GNINA CNN v2018 [24] perform better by training the general set. These results differ a bit with 3D fusion [7], which is a model composed of a 3D-CNN and of a spatial graph CNN (SG-CNN). In this case, it seems that 3D-CNN perform better by training on the refined set only, unlike the SG-CNN. Overall, this confirms that having more data, albeit of lower quality, gives better performance. One might question whether the observed performance enhancement obtained by training on the general set can be attributed to proper learning. Did the model develop a more profound comprehension of the interactions, or simply enhance its ability to memorize patterns within ligands and proteins?

These results also showcase that there might be a misunderstanding in the field of cheminformatics about the quality of data. Indeed, having data of high quality is very important for carrying out good predictions. Therefore, several teams have decided to train their models only with the refined set, which is considered to be of higher quality. Contrary to that belief, we think that the data that is not in the refined set can be still considered as useful data. Indeed, we can compare this data to fuzzy images in image recognition. These images are essential for the robustness of the models in real-life condition, since in this case not all images presented to the model would be clear. For image recognition, the saying “garbage in, garbage out” indicate that the images have been badly labelled; therefore, impeding the training process and resulting into models with worse performance. In the case of protein-ligand binding affinity predictions, the labelling task of the data has been handled by the team that conceived and update the PDBbind. They have been manually looking into publications

to report the experimentally evaluated binding affinities of complexes [25]. On top of this, the binding affinities obtained were compared to those gathered from MOAD [26], which is another database comprising protein-ligand complexes with binding affinities, in order to reduce the error rate.

### *3.2. Size of pockets*

The GNN are ideally designed to handle the data representing protein-ligand complexes. Indeed, this data is made of nodes (atoms or residues) and bonds (interactions between molecules or intramolecular interactions). Thanks to this design, GNN focus on the important information, being therefore efficient from a computational point of view. This is not the case for CNN that are quite computationally intensive as convolutions are performed on all the voxels of the 3D images. A lot of these voxels do not contain any information about the protein or the ligand, as they are located in the solvent. This increases the calculation time for no performance gain. Although some methodologies have been developed to avoid these hindrances [16], the most common way to reduce the computational requirements while maintaining good performance, is to only train models from the pockets instead of using the whole proteins.

The PDBbind provides pockets to the users for conveniences. They are constituted of all residues within a distance of 10 Å from the ligand. As the amount of data available for the training increases with the size of the ligand and therefore the size of the pockets, we have investigated the influence of the pocket size on the performance of trained models. For this purpose, we have created pockets of different sizes and trained 10 models per size with Pafnucy. We calculated 2 types of pockets by selecting the residues located within a specific distance measured from all the atoms, or from the center of geometry (CoG), of the ligands. The size of pockets was defined by the residue detection distances, ranging from 6 to 14 Å. The size of the box used in Pafnucy is equal to 2*detection distance + 1 Å.

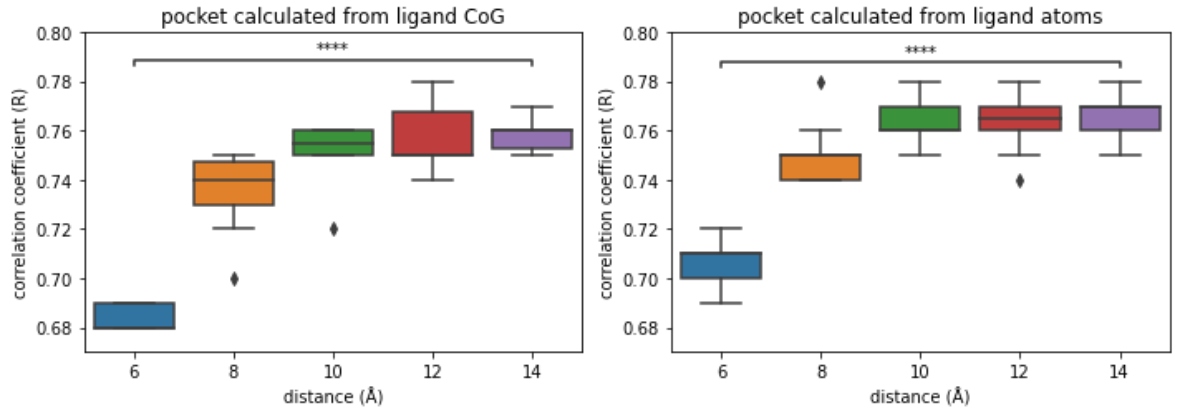



**Figure 3.** Comparison of the performance of models trained with different sizes and types of pockets. For each size, 10 models were trained and tested on the core set 2016. (**a**) Models trained with pockets made of residues located within a specific distance from the center of geometry of the ligands; (**b**) models trained with pockets created with residues located within a specific distance from all atoms of the ligands.

For both type of pockets, there is a significant difference in the performance of models trained on pockets of 6 Å and pockets of 14 Å (Figure 3). This is mostly due to the fact that there is more information available in bigger pockets. Therefore, it is advised to use pockets of 10 Å over pockets of 6 Å for training models, regardless of the type of pockets used.

Nonetheless, there is very little improvement in term of performance between 10 to 14 Å. Thus, there is no interest in using bigger pockets than 10 Å. A compromise is required between using small pockets that do not contain enough information and big pockets that are computationally more expensive, while not adding useful information.

Most of the interaction types fall within a range of 6 Å, thus it is difficult to understand why a pocket size of 6 Å is not sufficient to predict accurately the binding affinity. We think that this can be due to the bias in the data, in which case increasing pocket size, and therefore adding more amino acids would help the model in memorizing and recognizing pattern in the protein. We could be tempted to think that if we keep increasing the size of pockets, the performance would continuously improve. Although this does not appear to be the case. Hence there might be a limit to how much the bias in the data can artificially improve the performance.

Apart from the hypothesis of increased bias in the input data, another possible explanation would be a hidden influence of the amino acids that are not in direct contact with the ligands. Therefore, the model would be able to interpret some long-distance indirect interactions that are not easy for humans to decipher. In this case, the limit in performance reached by using pockets of 10 Å would mean that the amino acids added with bigger pockets are too distant from the ligand to influence it in an indirect fashion. Further investigations are required to confirm or refute these hypotheses.

Our limitation in interpreting such results is mostly due to the black box nature of DL algorithm. We do not know the underlying reasons for a given prediction. Although some methods were developed to alleviate the black box issue, like the layer-wise relevance propagation [27,28], gradient based methods [29], or by masking atoms [30]. Such methods would be useful to better understand the decisions taken by the model which leads to the prediction.

### *3.3. Peptide vs nonpeptide*

Some neural networks have been applied on complexes in interaction with a specific type of ligand. PointTransformer [21] was trained on the PDBbind 2016, after having removed 590 complexes, labelled as involving peptides.

Ahmed *et al.* developed a model by training only on proteins in complex with nonpeptides [9]. They created their own dataset by looking into the PDB for protein-ligand complexes with:

- Crystallographic complexes with a resolution lower than 2.5 Å
- Known binding affinity ($K_d$/$K_i$)
- Ligand that does not have protein chain, and are not DNA/RNA

This selection resulted in a dataset of 4,041 complexes. By using their neural network called DEELIG, they obtained a model that achieved a correlation coefficient of 0.889 on the PDBbind 2016 core set. These results are encouraging, and it seems worth looking into training models with only peptides and without them.

To evaluate the impact of training only with or without peptides, we flagged the complexes with peptide from the PDBbind. Indeed, among the numerous rules that the PDBbind established to select protein-ligand complexes, it has been decided that peptides having 20 residues or less would be considered as ligands [31]. Therefore, we have detected 2,915 complexes interacting with peptides among the 17,679 complexes of the PDBbind (v.2019).

By using Pafnucy, models were trained with complexes interacting with peptides or with complexes interacting with nonpeptides. As the dataset of protein-nonpeptide (PN) complexes is larger than the dataset of protein-peptide (PP) complexes, we randomly subsampled the dataset of PN complexes in order to have datasets of same size. We trained models by training on each of the even size datasets. We obtained a model trained on the PN dataset, and a model trained on the PP dataset. The performance of models was evaluated on the core set 2013 and 2016 (Figure A4). Performance was significantly better by training on PN complexes. Subsequently, we compared the performance of models by evaluating them on each type of molecules from the core set 2016. Therefore, we tested them only on the PN complexes, and only on the PP complexes (Figure 4).

Unsurprisingly, in comparison to the prediction on the whole core set 2016, we see that the prediction gap increases a bit when predicting only on PN complexes. This can be also explained by the fact that all proteins from the PN test set are present in the PN training set, while 40% of them are not in the PP complexes training set. On top of it, 30% of ligands from the PN test set are in the PN training set, and there are none in the PP training set.

As for the prediction carried only on the PP complexes, although the performance of models trained with PP complexes lowers a bit, the drop in performance is more drastic for the model trained on PN complexes. Therefore, it seems that there is information contained in the dataset of PP complexes useful to predict the PP complexes from the core set 2016, albeit the predictions were carried out only on 19 complexes. We can point out that 50% of the ligands are in the PP training set, while none are in the PN training set.

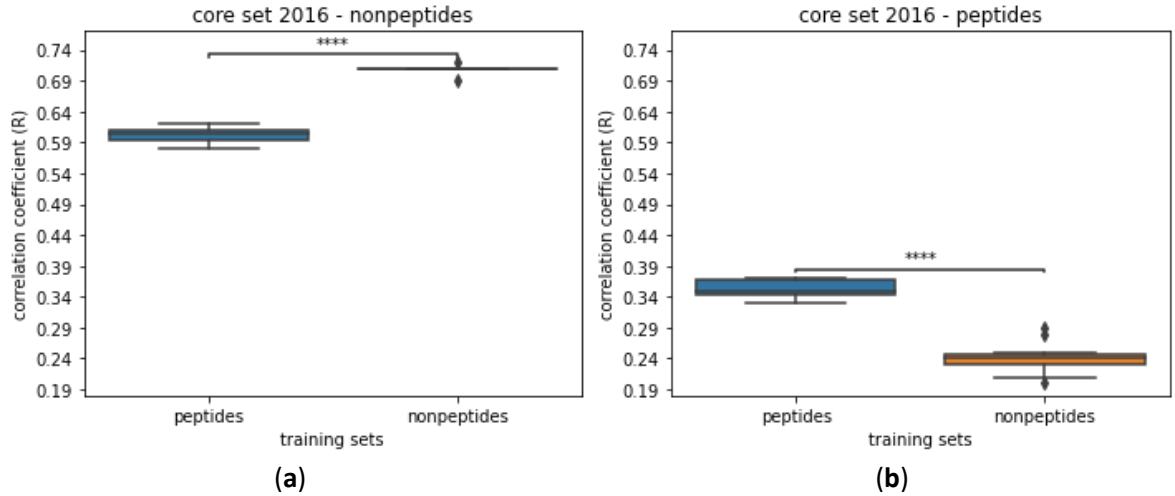
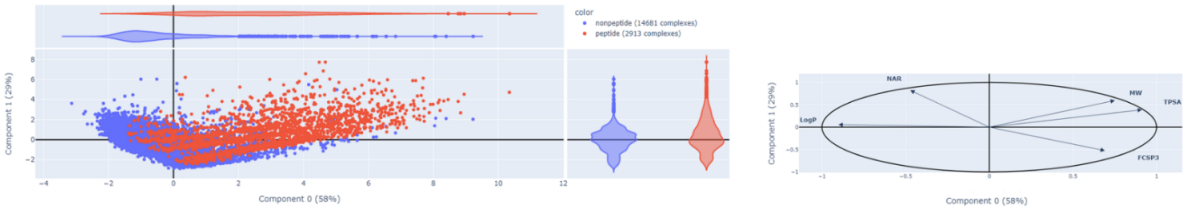



**Figure 4.** Comparison of the performance of models trained with peptide-protein complexes and with nonpeptide-protein complexes. Models were trained with Pafnucy on 2,383 complexes and validated on 492 complexes. (**a**) Performance evaluated on 266 complexes with nonpeptides from the core set 2016; (**b**) performance evaluated on 19 complexes with peptides from the core set 2016.

To better understand the difference in performance between models trained on PN and PP complexes, we performed a principal component analysis (PCA) on the ligand of the complexes from the PDBbind dataset. This allows us to compare the distribution of peptides and nonpeptide ligands (Figure 5). The descriptors used to characterize the ligands were selected based on the literature [32], then the correlated descriptors were removed. The following 5 descriptors were used to carry out the PCA: Lipophilicity (LogP), Topological Polar Surface Area (TPSA), Fraction of SP3 hybridized Carbon (FCSP3), Number of Aromatic Rings (NAR), and Molecular Weight (MW).




(**a**) (**b**)

**Figure 5.** Principal component analysis applied on the ligands of the PDBbind dataset. Peptides were coloured in red, while the rest of the ligands are displayed in blue. (**a**) Plot of the individuals; (**b**) correlation circle.

The PCA displays 87% of the variance of the data. It appears that the 2 populations of ligands are well separated. These results showcase the difference between peptides and small molecules, which help explain the lower performance from training with complexes involving only one type of ligand and predicting on the other type. Furthermore, the peptides are known to have high degrees of freedom especially due to the peptide bonds [8]. This increased flexibility results into high level of entropic energy, which needs to be taken into account when carrying out free energy prediction. Consequently, the evaluation of such values is very challenging. This can be an explanation for the poor performance of models in predicting the binding affinity for PP complexes.

We also evaluated the performance of models trained only on PN in comparison to training with both ligands mixed. Contrary to what we expected, it seems that training only on PN complexes does not improve the performance of the models (Figure A5). This comes as a surprise as we anticipated to obtain better performance in a similar fashion to DEELIG [9]. An explanation for the very high performance (R=0.889) obtained by DEELIG is that 68% of the test set complexes were used for the training, therefore skewing the evaluation of performance.

Nonetheless, even if it is better to train on the maximum amount of data as possible, there are promises to develop some local models focused on specific type of ligands. This is a practice less common than creating local models based on the type of proteins involved, but that can lead to interesting results. Moreover, it would be worth investigating transfer learning on such cases. For example, general models would be developed by learning general rules on the maximum amount of data, and then be specialized on predicting the binding affinities of peptides for example.

Once again, these results should be interpreted with caution, as there are strong indications of bias in the test set. For example, as we pointed out previously, all the proteins from the test are also present in the train set. The same issue applies to the ligands from the test set, with at least 30% of them being also in the train sets.

*3.4. Replication of results*

Most neural networks are non-deterministic. This behaviour leads to variation in the performance of models trained with the same neural network and the same data. Indeed, several factors influence the variability, one of them being that initial weights are assigned randomly across the neural network at the beginning of the training. Due to the randomized assignment of weights, the model is more likely to fall into certain local minima, creating uncertainty for the estimation. One way to overcome this issue is to modify the learning rate during the training, by using learning rate scheduler, and therefore getting out of local minima. The other solution is to train several model replicates, to increase the chances of having a model that did not fall into a local minimum. In any case, it is still necessary to carry out ensemble approaches [33] in order to accurately evaluate model performance and replicability. This implies training several models, averaging their performance and evaluating the standard deviation. This was done in the publication of OctSurf, where each value was averaged from 5 models. For this study, we replicated the results of 3 neural networks (Pafnucy [6], GraphBAR [17] and OctSurf [16]) and evaluated their averaged performance by training 10 models each time (Table 1).

**Table 1.** Replication of results from 3 neural networks (Pafnucy, GraphBAR and OctSurf) compared to results presented in their respective publications. Models are evaluated based on their correlation coefficients on the core set 2016.

| Neural networks | Results from publication | Results from replication |
|---|---|---|
| Pafnucy | R = 0.78 [1] | R = 0.77 ; SD = 0.01 [1] |
| GraphBAR | R = 0.76 [1] | R = 0.76 ; SD = 0.02 [3] |
| OctSurf | R = 0.79 ± 0.01 [2] | R = 0.79 ; SD = 0.01 [2] |

[1] PDBbind v2016; [2] PDBbind v2018; [3] PDBbind v2019

We were able to reproduce the performance displayed in the publication of each neural networks. All the standard deviations (SD) have low values like 0.01 or 0.02. Nonetheless, a SD of 0.02 means that, with GraphBAR, it is as likely to get models with a correlation coefficient of 0.74 as of 0.78 on a similar test set. As this is relatively a big difference in term of performance, we think that deep ensemble averaging [34] should always be applied when publishing the results of training models with a neural network. Although this is computationally intensive, it gives more reliable expectations for people re-using the same neural network, as well as preventing bias like selecting the best model and publishing its results as representative of the neural network performance.

Another use of model replicates is to build ensemble models. Instead of measuring the coefficient correlation for each model and calculating the mean and the standard deviation, it is possible to calculate the mean prediction for each sample and then to calculate the correlation coefficient. This methodology has already been applied for several deep learning models like PIGNet [35] and in Francoeur *et al.* [24]. It leads to some small gain of performance, for example by using this methodology, Pafnucy and GraphBAR get an R = 0.79. As for their RMSE, Pafnucy improve from 1.41 to 1.38 and GraphBAR from 1.43 to 1.37. Such consensus methods are therefore a good way of improving performance while being less subject to variations.

*3.5. Learning from ligand only, protein only or interactions*

Achieving good performance on a test set is the primary goal in model development, but it is also necessary to verify if such high performance is not due to learnt biases from the data. As mentioned previously, the PDBbind core set is heavily biased, with both proteins (all) and ligands (~30%) represented in the training set. Therefore, models will tend to shortcut learning by using easily learnable biases which might be not present in other datasets. This is what is called a noncausal bias, where there is correlation but no causation. As mentioned in Sieg *et al.* [36] models can artificially achieve good predictions by learning patterns that are not related to meaningful physico-chemical mechanisms for binding. For example, it appears that most of the reported binding affinity prediction models only memorize ligand and protein information instead of learning from their interactions [3]. This appears to be a major issue in the field, as it leads to poor generalization power.

Several strategies have been suggested to compel neural networks to learn from interactions for virtual screening purpose [37,38]. For instance, decoy poses have been generated by modifying the position of ligands. These decoys poses were obtained by redocking active compounds and selecting a low energy poses with a high RMSD from the initial position. Even simpler methods like rotating and translating the ligands have been applied. In a similar way, we propose that this could be applied on the PDBbind dataset, by either redocking, rotating or translating high affinity ligands. The resulting decoy poses would be labelled with low affinity. Consequently, when trained on such datasets, models will encounter several occurrences of the same complexes, with different ligand positions and different binding affinities. Therefore, we expect such models to adapt from doing mostly QSAR to actually understanding protein-ligand interactions. Previous works were published on the topic of data augmentation with docking for scoring functions [17,24,39,40]. To the best of our knowledge, all of them focused on selecting poses similar to the crystallographic one, and assigning similar binding affinities. Another idea would be to dock ligands with low affinity from the CHEMBL, especially the ones that are structurally similar to high affinity ligands from the PDBbind. In the case that these ligands interact with the same proteins, we would add the notion of activity cliff to the models. These data augmentation methods would help the models generalize by making it focus on the interactions rather than memorizing the bias inside the dataset. We have not used the aforementioned methods in this study, and we will discuss this in more detail in future work.

As mentioned previously, there are several visualization tools that reveal which parts of a structure are important when carrying out a prediction. In Hochuli *et al.* [30] those methods were applied on GNINA CNN v2017 [38] to understand its underlying reasoning for the classification of active and inactive molecules. Another way to uncover if a model truly learnt from the protein-ligand interactions, is to train other models by masking either the protein or the ligand. Then, it is possible to evaluate the gap of prediction between learning on the full complexes and learning on one of the 2 partners. The bigger the gap in prediction is, the better the model's understanding of the interactions. However, these considerations are relatively recent. Only a few neural networks have been evaluated for their ability to learn from interactions, and not only memorize structural patterns in proteins or ligands. For this purpose, at the Figure 6, we have evaluated the ability of learning on interactions for two already published neural networks: a convolutional neural network (Pafnucy) and graph convolutional neural network (GraphBAR).

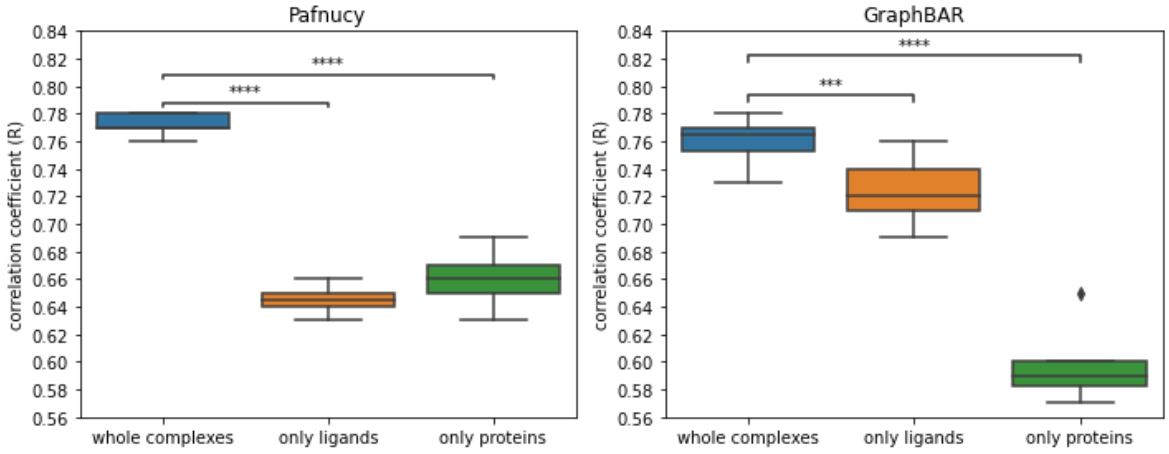



**Figure 6.** Comparison of the performance of Pafnucy and GraphBAR without either the protein or the ligand. The performance of models was evaluated on the core set 2016. Learning on the whole complexes lead to significantly better performance. (**a**) The mean prediction gap between training on whole complexes and training on ligands alone is at 0.13 of coefficient correlation for Pafnucy; (**b**) while it is only at 0.04 for GraphBAR by training only on the ligands.

With both neural networks, training on the whole complexes give significantly better performance than training on the ligand or protein structures alone. Nonetheless, we can see disparities between the two neural networks as the difference in correlation coefficient by training only on the ligands compared to the whole complexes is 0.13 for Pafnucy, while it is only at 0.04 for GraphBAR. This means that Pafnucy does a better job at analysing the interactions made between the proteins and the ligands, while GraphBAR seems to more heavily rely on learning patterns from ligands and then correlate them to binding affinities.

In the publication of OctSurf the performance was also evaluated by training only on ligands and only on proteins. A correlation coefficient of 0.79 was reported for the full complex, while reaching 0.73 with ligands, and 0.65 with proteins. Thus, the prediction gap is at 0.06, which is between Pafnucy and GraphBAR.

Other binding affinity models have been tested for their ability to learn from the interactions, by training only on proteins or only on ligands. All these results have been summarized in Table 2. The results of the Modular MPNN are in accordance with previously evaluated neural networks. Nonetheless Deep Fusion and PointTransformer achieve a way bigger prediction gap by removing either the ligand or the protein. It goes up to 0.41 for PointTransformer when learning only on ligands.

**Table 2.** Comparison of performance of several neural networks, on PDBbind core set 2016 with/without protein/ligand.

| Neural networks | Whole complex (R) | Only ligand (R) | Only protein (R) |
|---|---|---|---|
| Pafnucy | 0.77 | 0.65 | 0.66 |
| GraphBAR | 0.76 | 0.73 | 0.60 |
| OctSurf | 0.79 | 0.73 | 0.64 |
| Modular MPNN [3] | 0.81 | 0.75 | 0.73 |
| Deep Fusion [7] | 0.81 | 0.49 | 0.5 |
| PointTransformer [21] | 0.86 | 0.45 | 0.2 |

From these results, it seems that the ability of neural networks to learn from the interactions can vary importantly. The PDBbind 2019 was used as training data for both Pafnucy and GraphBAR, and both used similar descriptions of atoms. Therefore, the main factor differencing the two is the underlying structure of the networks, and the ensuing way of handling the data and carrying out prediction.

Accordingly, Deep Fusion reuse the same preparation protocol as Pafnucy, in terms of atomic description for example. Furthermore, it combines a 3D-CNN and a spatial graph CNN; this unique approach might be the reason for the model ability to better understand the protein-ligand interactions.

PointTransformer is a point cloud based neural network, like OctSurf. Therefore, we expected this tool to have similar prediction gap to OctSurf. On the contrary, the prediction gap was much more important with PointTransformer.

### *3.6. Other test sets*

As shown throughout this paper, there are numerous biases contained in the core sets from the PDBbind. Due to this, we think it is important to use other types of benchmark datasets to accurately validate the new models developed. Indeed, the evaluation of models across several test sets grants a higher confidence when comparing performance. Across time, several other tests set have been developed to evaluate the scoring and ranking power of models. The scoring and ranking power are, respectively, the model ability to predict accurately the binding affinity and its ability to correctly rank ligands by using the predicted binding affinity.

There are test datasets that have already been used in numerous publications [2]. For example, the Astex diverse set [41] has been used to validate Pafnucy [6], DeepAtom [22] and RosENet [42]. It includes 85 protein-ligand complexes, 74 of which have known binding affinity. There are, as well, other test sets called the CSAR-NRC HiQ set 1 and set 2 [43,44] which are composed of 176 and 167 complexes from the Binding MOAD [26] and the PDBbind. After removing the complexes overlapping with usual training set, around 50 and 40 complexes remain for both test sets (Table A6). They have been used to evaluate $K_{DEEP}$ [45], RosENet [42], OnionNet-2 [46], graphDelta [47], GraphBAR [17], PIGNet [35], BAPA [48], CAPLA [49] and GIGN [23].

The FEP dataset [50] originally used in free energy perturbation studies has also been applied to evaluate the binding affinity predictions of several models [42,45,47]. It is used to test the ability of a model to discriminate between several similar ligands with different binding affinities for the same protein. It is composed of 8 proteins: BACE, CDK2, JNK1, MCL1, p38, PTP1B, Thrombin and Tyk2. Each protein family is represented by one structure. There are 200 ligands obtained from a small number of scaffolds. Their 3D positions in the binding site are provided. Their affinities have been obtained experimentally. This information is summarized in the Table A7.

Hold-out test sets have also been developed to evaluate performance of models on recent data. These test sets are obtained by performing a temporal split over a dataset, *i.e.* training models on complexes released before a specific date, and testing them on complexes released afterward. The hold-out test sets are generally big sized, with complexes that were not cherry-picked and thus are less likely to be biased.

- An example of such dataset can be found in Volkov *et al.* [3], where a modular MPNN and Pafnucy were trained on PDBbind 2016 and were evaluated by predicting on a 2019 hold-out

set. To create this test set, they selected 3,386 complexes from the PDBbind 2019 that are not in the PDBbind 2016. Instead of using the files provided by the PDBbind, they downloaded the structures from the Protein Data Bank [11]. The complexes were curated and processed with Protoss v.4.0 [51] and IChem [52], *e.g.* protonation was optimized. Subsequently Isert *et al.* [53] reused these data to train models with electron density-based geometric neural networks, and they validated their binding affinity predictions on the same 2019 hold-out set.

- Another 2019 hold-out set of 4,366 complexes was used to evaluate GIGN [23]. They compared their results against a dozen of neural networks, including OnionNet [54], Pafnucy and GNN-DTI [55]. It is worth mentioning that the protein overlap rate between test and training sets is of 69% instead of 100% for the core set 2016. As for the ligand overlap rate, it goes down to 25% when it was at 38% for the core set 2016.
- Due to similar consideration, Deep Fusion [7] was evaluated on a test set of 222 complexes that was developed from the 2019 hold-out set, by removing complexes with ligand or protein already present in the PDBbind 2016. Deep Fusion, $K_{DEEP}$ and Pafnucy were trained on the PDBbind 2016, and evaluated on this test set.
- AK-score [56] was trained on the refined set of the PDBbind 2016 and it was evaluated by predicting the binding affinity of 534 complexes newly released in the refined set of the PDBbind 2018. For comparison purposes, they also evaluated the performance of other scoring functions, namely X-score [57] and ChemPLP [58].
- The atomic convolutional neural network (ACNN) [59] was trained and tested on several different splits of the PDBbind dataset. On top of a temporal split, they used a stratified split based on the $pK_i$ value of complexes and a ligand scaffold split. The stratified split allowed to select complexes covering all binding affinities in the train and test sets. In the case of the scaffold split, ligands with unusual scaffold were placed in the test set, therefore preventing the effect of QSAR in the prediction.
- In a similar way, MoleculeNet [60] has been trained and tested on PDBbind dataset with a temporal split. As for PotentialNet [61], they performed cross-validation by performing a pairwise structural homology split and a sequence similarity split. Both splits are explained in detail in Li & Yang [62]. They were carried out via an agglomerative hierarchical clustering, on the PDBbind 2007 refined set, resulting in a test set of 118 and 101 samples, respectively.

The PDE10A dataset [63] have been recently released, with 1,162 docked or co-crystalized PDE10A inhibitors. These data are sourced from a former project of Roche; thus the binding affinity (IC50) were obtained in a consistent way. There are 77 PDE10A complex structures obtained by crystallography, and the rest of the complexes were generated through multi-template docking. The test sets were obtained by using temporal and binding mode splits. There are three temporal split test sets, the 2011, 2012 and 2013 test sets, with 250, 141 and 73 complexes respectively. Similarly, there are three binding mode split test sets, the aminohetaryl_c1_amide, c1_hetaryl_alkyl_c2_hetaryl and the aryl_c1_amide_c2_hetaryl test sets, composed of 452, 291 and 419 complexes respectively. They compared their 2D3D ML methods against PotentialNet [61] and ACNN [59]. Isert *et al.* [53] also benchmarked their neural networks on these test sets.

Apart from the scoring and the ranking power, there are other criteria that can be used to evaluate drug-target interactions models, like the virtual screening (VS) power. This criterion defines the ability of a model to discriminate between decoys and active molecules. As brought up in PIGNet [35], to accurately access the performance of a model, it is advised to evaluate not only its scoring power but also its virtual screening power. For evaluating such ability, datasets incorporating decoys have also

been used as test set. Nonetheless, warnings must be raised about using these datasets. Indeed, most of them are also biased [36], especially when splitting one of them in training and test sets, which usually leads the underlying models achieving artificially high performance. On the contrary when training a scoring function on the PDBbind and predicting on VS datasets the results are usually lower. The performance of models evaluated on VS datasets are measured by calculating the area under the ROC curve (AUC), which increases when active molecules are predicted with higher binding affinities than decoys. Furthermore, it is possible to evaluate scoring functions by calculating the enrichment factor (EF) from the ROC curve. The EF is obtained by measuring the true positive rate (TPR) for a given false positive rate (FPR). Therefore, it is possible to evaluate the model ability to find active molecules over decoys for its best scored docking poses. Hence, The EF is more representative of the use of VS tools in real condition, as users are mostly interested by the ligands with the highest score.

Examples of such datasets are the DUD [64] (directory of useful decoys) and DUD-E (enhanced DUD) [65]. They are used for benchmarking molecular docking by providing active molecules and decoys (assumed inactive) for given targets. They have been developed to deal with usual dataset problems, like "artificial enrichment" which correspond to having decoys that are very different from active molecules, and "false negative bias" referring to decoy turning out to be active after being tested experimentally. The DUD-E is an enhanced version of the DUD with increased amount of data. It is designed to address the "analogue bias" of having highly similar active molecules. The DUD and DUD-E are composed respectively of 2,950 and 22,886 active molecules, as well as 95,326 and 1,411,214 decoys (up to 50 decoys per active molecule charge states), for 40 and 102 targets. Unfortunately, there are still biases present in the DUD-E [66]. Especially, an analogue bias intra and inter target was detected. These biases add up with the decoy bias, which is the similarity of decoy from the same target. When trained on a part of the DUD-E and evaluated on the other part, models obtain the same high performance (AUC > 0.9) if we keep the whole complexes or only use the structure of the ligand. Therefore, it leads to similar issues as the ones related to the PDBbind core set.

- The DUD-E was used to train AtomNet [67] and to evaluate its virtual screening power. AtomNet is the first CNN applied on 3D grids to predict protein-ligand binding affinities. 30 targets from DUD-E were used as test set, while the remaining 72 targets were used as the training set. On top of using DUD-E dataset, a derived dataset, called "ChEMBL-20 PMD", has been compiled to further benchmark AtomNet. It was created based on several quality criteria and it is composed of 78,904 actives, 2,367,120 property-matched decoys (PMD), and 290 targets. That dataset is composed of decoy structurally different from the active molecules to prevent the false negative bias issue which on the other hand results in an artificial enrichment issue. Therefore, another dataset, called "ChEMBL-20 inactives", was developed to evaluate AtomNet's ability to classify experimentally verified active and inactive molecules. ChEMBL-20 inactives was obtained by replacing the PMD by 363,187 molecules known to be inactive.
- In Lim *et al.* [55], they used the DUD-E and the PDBbind to constitute a training set and a test set. Molecules were docked with Smina [68], resulting in a dataset of docked poses for DUD-E's 21,705 active molecules and 1,337,409 decoys. As for PDBbind, the molecules were re-docked with Smina. If the pose had a RMSD < 2 Å from the crystallographic pose, then it was classified it as a positive sample and if the pose was at > 4 Å from crystallographic pose, then it was classified it as a negative sample. Therefore, 2,094 positives and 12,246 negatives samples were obtained. The training set was subsequently created with the docked poses of 72 proteins from the DUDE and 70% of PDBbind redocked dataset. The test set consisted of the docked poses from the remaining 25 proteins from the DUDE and 30% of PDBbind redocked dataset. PDBbind split of data was based on a split of the targets, so no proteins would be in the training and test sets. Thereafter another test set was developed by selecting,

from the CHEMBL, molecules with known binding affinity for the 25 proteins from the DUDE test set. The affinity threshold was put to an IC50 of 1.0 μM, splitting the test set in 27,389 active and 26,939 inactive molecules.

Similarly to the DUD/DUD-E, the DEKOIS 2.0 [69] dataset was developed to evaluate scoring functions for their virtual screening power. It is composed of 81 benchmark sets for 80 protein targets (one target having 2 different binding sites and benchmark sets). There are 40 active molecules per benchmark set. For each active molecule, 30 structurally diverse decoys were selected, resulting into 1,200 decoys per benchmark set. The DEKOIS dataset is constituted of decoys that have not been tested experimentally, therefore decoys were selected by matching the properties of the active molecules to avoid artificial enrichment. Furthermore, the selection of the decoy has been tailored to prevent the occurrence of latent actives in the decoy set (LADS). LADS are molecules supposed to be decoy, which actually have an activity for the target. This issue was previously referred to in the study as false negative bias. Only 4 targets out of the 81 of the DEKOIS dataset are in common with the DUD-E [70], but 26 targets have at least 95% of sequence identity with DUD-E targets [71]. As pointed out in Ballester's paper [72], several machine learning scoring functions [70,71,73] were trained on DUD-E and evaluated on DEKOIS.

The Maximum unbiased validation (MUV) is another dataset developed to benchmark virtual screening tools. It is composed of active and inactive molecules experimentally tested for 17 target proteins. For each target protein, there are 30 actives and 15,000 decoys with known binding affinities. In a similar fashion, Riniker and Landrum [74] created a dataset from CHEMBL comprising 50 targets, with 100 diverse active molecules per target and 2 decoys per active molecule leading to 10,000 decoys. The GNN-DTI from Lim *et al.* [55] was evaluated on the MUV dataset. GNINA CNN v2017 [38] and the DenseNet CNN from Imrie *et al.* [75] were evaluated on a part of both the MUV dataset and the ChEMBL dataset from Riniker and Landrum. The active molecules and decoys were docked with smina [68] or AutoDock [76]. For the MUV dataset, out of the 17 target proteins, 9 were used in the test set. Therefore, leading to 1,913 poses associated with the 270 actives molecules and 1,177,989 poses associated with the 135,000 decoys. As for the CHEMBL dataset, 13 targets among the 50 targets were used, leading to 11,406 poses associated with 1,300 active compounds and 663,671 poses associated with 10,000 decoys.

In the CASF update [15], the scoring power, the ranking power, the docking power and the screening power of several scoring functions were evaluated on the core set 2016. The docking power correspond to the ability of a scoring functions, to identify the native ligand binding pose among several decoy poses of the same ligand. More than 30 scoring functions were evaluated for these criteria.

- To assess the docking power, decoy poses were generated by redocking PDBbind's ligands in their binding site. For each complex, up to 100 decoy poses were selected, by setting up 10 bins of 1 Å based on their RMSD values (0-10 Å) to the initial pose. For each bin, ligand poses were clustered based on their conformation, and up to 10 poses were selected. This leads to a dataset composed of 22,492 decoy poses.
- In order to evaluate virtual screening power, the ligands were cross-docked. The dataset is composed of 16,245 protein–ligand interaction pairs, by docking 285 ligands into 57 proteins. The docking was performed on the protein structure with highest affinity for each cluster. 100 poses were selected for each protein–ligand interaction pair. Overall, 1,624,500 decoy poses make up this dataset.

In Francoeur *et al.* [24] several docking datasets have been compiled to train and test their neural networks. They obtained a test set of 4,618 poses by redocking 280 complexes from the PDBbind core set 2016 and selecting up to 20 poses per complex. In a similar fashion they redocked 3,805 complexes from the refined set and 11,324 from the general set, leading to respectively 66,953 and 201,839 poses. Thereafter, they created the CrossDocked2020 dataset, by crossdocking complexes from the Protein Data Bank [11] that were selected based on the similarity of the binding pockets. They trained their neural networks on a first version of this dataset, then selected wrongly predicted poses as data augmentation for retraining the model. This iterative reinforcement learning method led to a dataset of 22,584,102 poses (786,960 redocked poses and 21,797,142 cross-docked poses) from 18,450 complexes. 42% of these complexes have known binding affinities from the PDBbind. From there, the BigBind dataset [77] was created, by mapping CHEMBL activities to the 3D structures of protein pockets in CrossDocked. By doing so, the number of pockets was reduced from 2,922 (in CrossDocked2020) to 1,067. The resulting dataset contains 11,430 3D structures, with 851,359 activities spanning 531,560 unique compounds.

In the GNINA CNN v2017 publication [38], the docking power was evaluated by redocking the 2013 PDBbind core (195 complexes). They obtained 98 low RMSD poses (<2 Å from the crystallographic pose) among a total of 897 poses. The training was carried on redocked complexes from the CSAR-NRC HiQ data set [43] and the CSAR HiQ Update. From the initial 466 complexes, they redocked 377 complexes having a binding affinity > 5 pK. Poses at less than 2 Å from crystallographic poses, were labelled as positive, while the one at more than 4 Å were labelled as negative. The one between 2 and 4 Å were discarded. This leads to a dataset composed of 745 positive poses (from 327 complexes) and 3,251 negative poses (from 300 complexes).

Famous datasets like the PDBbind/CASF, the DUD-E or the MUV have been applied to train and evaluate many models. Unfortunately, it appears that most of the famous datasets are biased. Although they may still be relevant to some extend for comparison purpose, we have seen the development of a myriad of new benchmark datasets. Many papers presenting new neural networks demonstrated their performance on custom test sets. For example, six papers developed their own training and test sets by performing a temporal split. For a better comparison of models, it would be preferable to evaluate their performance on a common benchmark dataset obtained through temporal split.

Overall, we think that it is important to evaluate the scoring power of models on several benchmark datasets, to get an accurate evaluation of their performance. On top of that, we advise for the evaluation of their ranking, docking and screening powers. By doing so, we can get a better idea of their usefulness in real case scenario.

## 4. Conclusion

For some years now, deep learning models have been developed to predict protein-ligand binding affinity. The scientific community has been trying to establish guidance on how to use these tools. Data plays a central role in training DL models. Therefore, we have been investigating how the data can impact the performance of models, as well as the intrinsic biases from the PDBbind. Among all the problems related to the data, the question of quality and the quantity of the data used to train DL algorithms seems crucial. A lot of neural networks have been trained only on the PDBbind's refined set, instead of the totality of the data available. The refined set is made of complexes selected based on quality criteria. The reasoning for training on only the refined set is to avoid the "garbage in, garbage out" issue. We have evaluated this factor by training Pafnucy, a well-known CNN for the prediction of

protein-ligand binding affinity, on the refined set only and on the entire dataset. We found out that the performance was lower by training on the refined set. Therefore, we think that it is important to train on most of the data available, as long as the data has been accurately labelled.

The PDBbind database groups several types of ligands together, with peptides and small molecules being the main populations involved in protein-ligand complexes. As only a few neural networks [9] have focused on training on complexes involving a specific type of ligand, we trained Pafnucy on the protein-peptide and protein-nonpeptide complexes of the PDBbind. We compared the performance by training on similar sized datasets and found out that models trained with peptides were able to better predict the binding affinity of protein-peptide complexes. Therefore, it would be interesting to investigate transfer learning on such type of data, to reach good performance for the prediction of binding affinity of protein-peptide complexes.

Due to the computational expensive nature of CNN and their high requirement on RAM, it is not possible to train models on the whole protein structure. Indeed, beforehand it is required to create pockets around the ligands. We have evaluated performance of models trained on pockets made of the amino acids detected at 6, 8, 10, 12 and 14 Å from the ligand. By increasing the size of pockets, we see performance increase until 10 Å, thereafter performance stagnate. As most protein-ligand interactions should be already considered at a distance of 6 Å from the ligand, we propose that the increase in performance is due to the bias in the data. In other word, adding more information about the proteins, would not add any useful physical information but just help the models to overfit. Another possible explanation would be related to the existence of some long-distance influences of these amino acids on the ligand, which would impact the affinity of the complexes. Therefore, the AI would detect these indirect interactions that would be hard to notice for a human.

Following on the topic of biases in the PDBbind core set, we evaluated different types of neural networks for their ability to learn from the interactions instead of memorizing the biases in the data. From these results, it seems that GraphBar does mostly QSAR since it has nearly the same performance with and without the proteins, or in other words Pafnucy seems to better understand the interaction between the protein and the ligand. On that topic, published work [7,21] reported even bigger performance gaps.

Finally, we pointed out some flaws inside PDBbind 2016 core set. For example, 30% of the ligands from the test set are also in PDBbind general set. As for the proteins, this value goes up to 100%. In the GNINA CNN v2017 publication [38], this was mitigated by removing test targets with more than 80% sequence similarity with a target from the training set. In a similar fashion, PIGNet [35] exclude, from the CSAR NRC-HiQ, the complexes that have at least 60% of sequence similarity with a target from the training set. Following these examples, Yang *et al.* [10] advocate for the removal, from test sets, of complexes with structurally similar proteins and ligands in comparison to training sets. Although doing as such prevents the evaluation of models in the situation of drug repurposing and hit to lead optimization [3]. Therefore, we recommend evaluating models on several test sets to better assess their ability to generalize and to accurately predict the binding affinity. On top of the CASF and the CSAR NRC-HIQ, we can list the Astex diverse set, the FEP dataset and the holdout test sets. Several neural networks have already evaluated their performance on such datasets, allowing for easier comparison with the newly developed methodologies.

For a thorough evaluation of the models, we also advise evaluating their screening power. To measure that criterion, it is required to dock active molecules and decoys, before evaluating their binding affinities and ranking the molecules. Some datasets propose list of decoys and active molecules, like the DUD-E [65], DEKOIS [69], MUV [78] or the “Riniker and Landrum CHEMBL” [74]. The

difference between these datasets depends mostly on the way they define decoys, and how they tried to prevent the appearance of biases. Unfortunately, biases can still be found in these datasets. In the end, models trained on the PDBbind did not outperform docking software in term of VS power when applied on the DUD-E [66]. Nonetheless, if it is possible to obtain better VS power, even at the expense of lowering scoring power performance on PDBbind core set, this would mean we are likely going in the right direction. This should be achievable by training models on a decoy poses augmented PDBbind dataset, which should force models to learn from the interactions instead of memorizing ligand and protein structures. However, by using decoy poses, we might not represent accurately the physico-chemical reality of the interactions of a protein and a ligand. Indeed, the interactions between them are dynamic, thus the ligand might take several positions inside the binding site across time. As mentioned previously in the literature [79], it would be more suitable to perform data augmentation with molecular dynamics simulations. For example, snapshots could be extracted from the simulations and fed to neural networks. This way, we can expect to improve models understanding of protein-ligand interactions.

**Supplementary Materials:** Not applicable.

**Author Contributions:** Investigation and writing—original draft preparation, PY.L.; supervision, S.AS., P.B., and G.T.; writing—review and editing, S.AS., P.B., JC.GT. and G.T.; All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by JANSSEN.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable

**Conflicts of Interest:** The authors declare no conflict of interest.

**Sample Availability:** Not applicable.

**Appendix A**

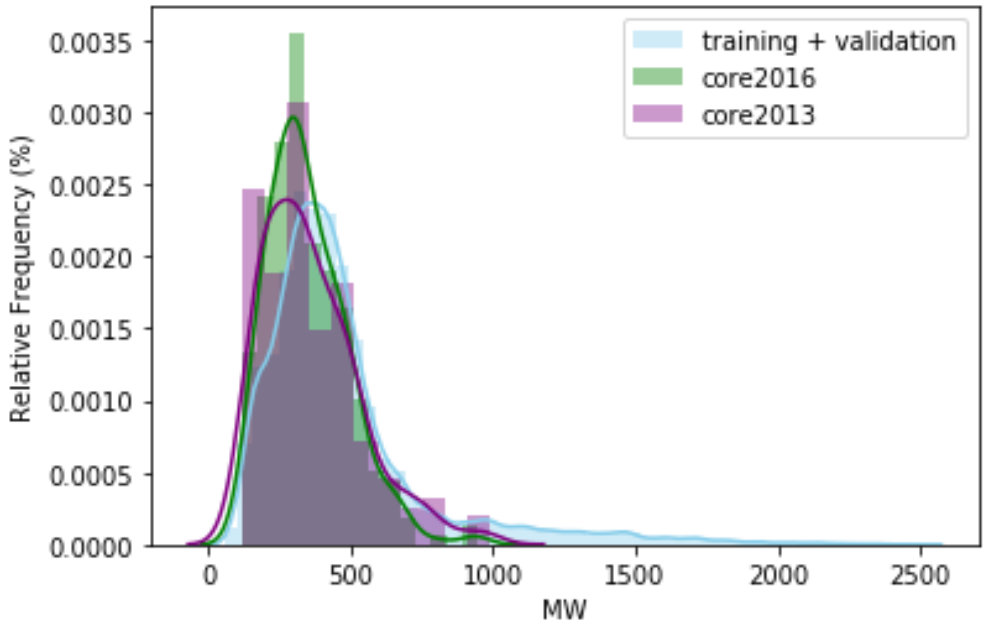



**Figure A1.** Distribution of PDBbind's ligands in function of their molecular weight. The training and validation set are plotted together in blue, the test sets are colored in pink and green, corresponding to the core set 2013 and core set 2016 respectively.

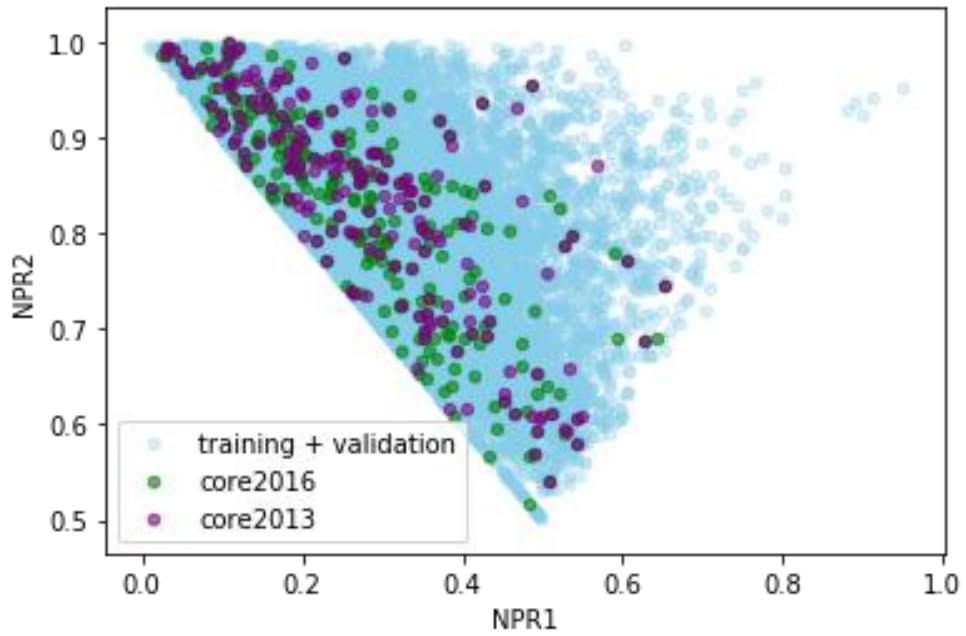



**Figure A2.** Distribution of PDBbind's ligands in function of their shape. The normalized PMI ratio (NPR) is calculated from the principal moment of inertia (PMI) of the ligands. The ligands located at the top right of the plot are spherical, while the one at the top left are rod-like. Lastly, the ligands in the bottom of the plot are with the shape of a disc. The training and validation set are plotted together in blue. While the test sets are in pink and green, corresponding to the core set 2013 and core set 2016 respectively.

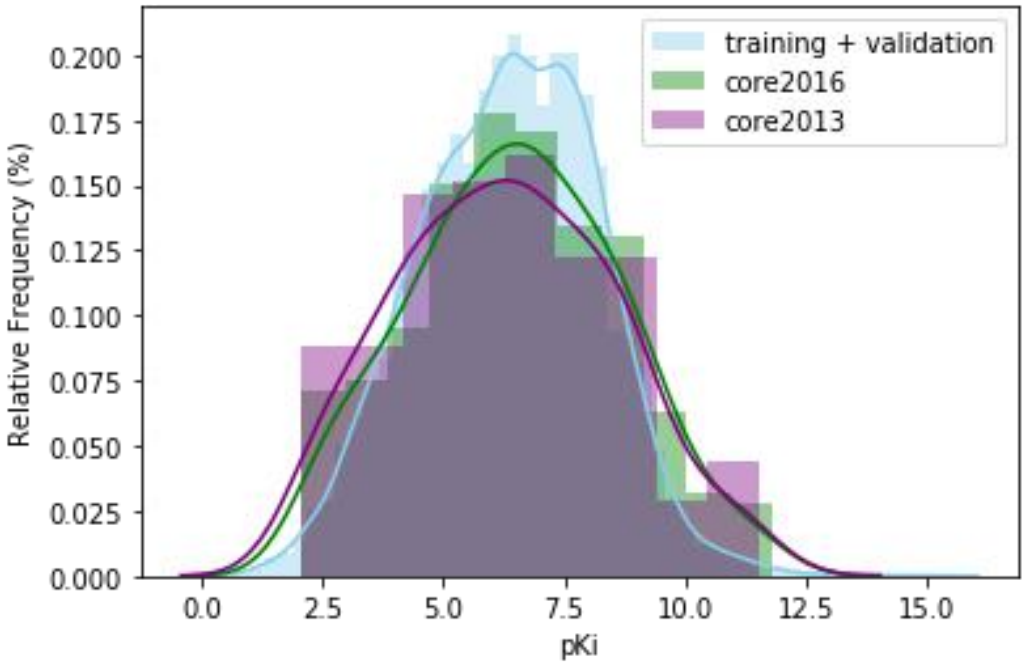
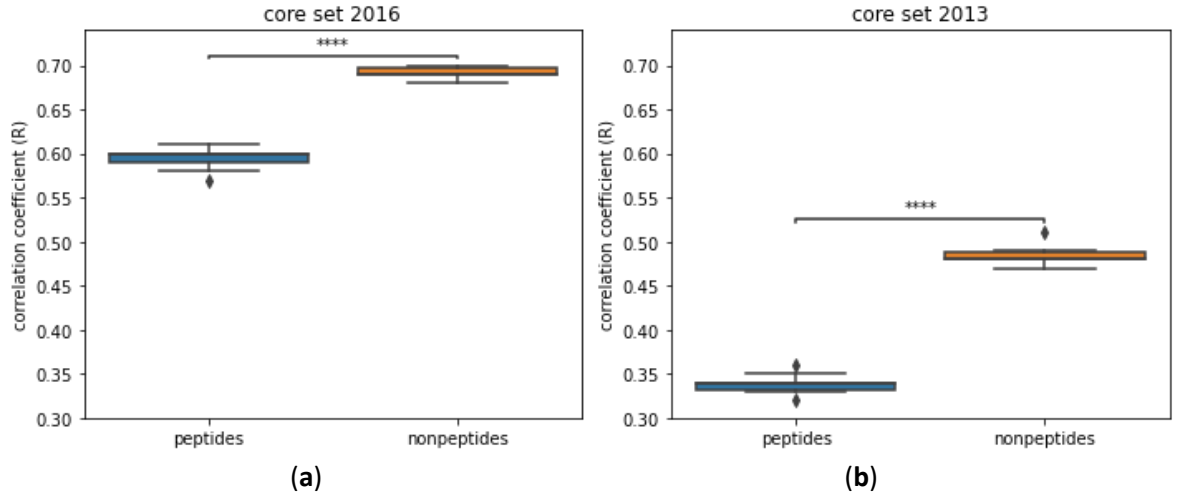



**Figure A3.** Distribution of PDBbind's complexes in function of their affinity. The training and validation set are plotted together in blue. While the test sets are in pink and green, corresponding to the core set 2013 and core set 2016 respectively.




(**a**) (**b**)

**Figure A4.** Comparison of the performance of models trained with peptide-protein complexes and with nonpeptide-protein complexes. Models were trained with Pafnucy on 2,383 complexes and validated on 492 complexes. (**a**) Performance is evaluated on the core set 2016; (**b**) performance are evaluated on the core set 2013.

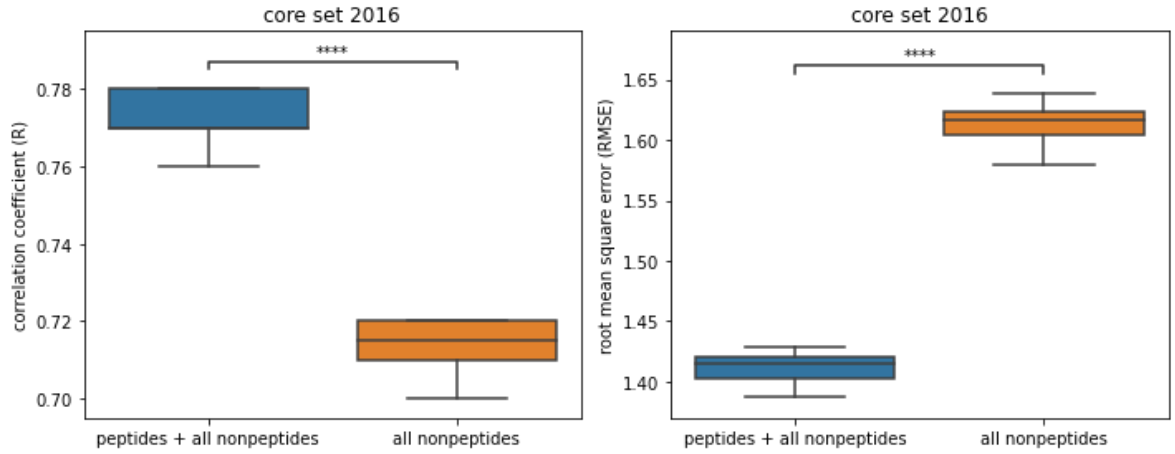



**Figure A5.** Comparison of the performance of models trained on the whole PDBbind and with only nonpeptide-protein complexes (trained on 13,403 complexes and validated on 1000 complexes). Models were trained with Pafnucy. Performance is evaluated on the core set 2016 (285 complexes). (a) The performance is evaluated with the coefficient correlation; (b) The performance is evaluated with the root mean square error (RMSE).

**Table A6.** Number of complexes of the CSAR NRC-HiQ set 1 & 2, used in each publication. In GIGN, the sets were merged together.

| Neural networks | CSAR NRC-HiQ set1 | CSAR NRC-HiQ set2 |
|---|---|---|
| $K_{DEEP}$ [45] | 55 | 49 |
| RosENet [42] | 33 | 10 |
| OnionNet-2 [46] | 55 | 49 |
| graphDelta [47] | 53 | 49 |
| GraphBAR [17] | 51 | 36 |
| PIGNet [35] | 48 & 37 | 37 & 22 |
| BAPA [48] | 50 | 44 |
| CAPLA [49] | 51 | 36 |
| GIGN [23] | 47 | |

**Table A7.** Summary of the FEP dataset from $K_{DEEP}$ [45] and Wang *et al.* [50]. This table displays the target (protein family), the reference PDB id used, the number of ligands positioned in 3D in each structure and the experimental affinity range of complexes belonging to the same protein family.

| Target | PDB ID | Number of ligands | Affinity range (kcal/mol) |
|---|---|---|---|
| MCL1 | 4HW3 | 42 | 4.2 |
| BACE | 4DJW | 36 | 3.5 |
| p38 | 3FLY | 34 | 3.8 |
| PTP1B | 2QBS | 23 | 5.1 |
| JNK1 | 2GMX | 21 | 3.4 |
| CDK2 | 1H1Q | 16 | 4.2 |
| Tyk2 | 4GIH | 16 | 4.3 |
| Thrombin | 2ZFF | 11 | 1.7 |

**References**


1. Baig, M.H.; Ahmad, K.; Roy, S.; Ashraf, J.M.; Adil, M.; Siddiqui, M.H.; Khan, S.; Kamal, M.A.; Provazník, I.; Choi, I. Computer Aided Drug Design: Success and Limitations. Current pharmaceutical design 2016, 22, 572-581, doi:10.2174/1381612822666151125000550.

2. Meli, R.; Morris, G.; Biggin, P. Scoring functions for protein-ligand binding affinity prediction using structure-based deep learning: a review. Frontiers in Bioinformatics 2022, 2, doi:10.3389/fbinf.2022.885983.

3. Volkov, M.; Turk, J.-A.; Drizard, N.; Martin, N.; Hoffmann, B.; Gaston-Mathé, Y.; Rognan, D. On the Frustration to Predict Binding Affinities from Protein–Ligand Structures with Deep Neural Networks. Journal of medicinal chemistry 2022, doi:10.1021/acs.jmedchem.2c00487.

4. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Kai, L.; Li, F.-F. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, 20-25 June, 2009; pp. 248-255.

5. Wang, R.; Fang, X.; Lu, Y.; Wang, S. The PDBbind database: collection of binding affinities for protein-ligand complexes with known three-dimensional structures. Journal of medicinal chemistry 2004, 47, 2977-2980, doi:10.1021/jm030580l.

6. Stepniewska-Dziubinska, M.M.; Zielenkiewicz, P.; Siedlecki, P. Development and evaluation of a deep learning model for protein-ligand binding affinity prediction. Bioinformatics (Oxford, England) 2018, 34, 3666-3674, doi:10.1093/bioinformatics/bty374.

7. Jones, D.; Kim, H.; Zhang, X.; Zemla, A.; Stevenson, G.; Bennett, W.F.D.; Kirshner, D.; Wong, S.E.; Lightstone, F.C.; Allen, J.E. Improved Protein–Ligand Binding Affinity Prediction with Structure-Based Deep Fusion Inference. Journal of chemical information and modeling 2021, 61, 1583-1592, doi:10.1021/acs.jcim.0c01306.

8. Unarta, I.C.; Xu, J.; Shang, Y.; Cheung, C.H.P.; Zhu, R.; Chen, X.; Cao, S.; Cheung, P.P.; Bierer, D.; Zhang, M.; et al. Entropy of stapled peptide inhibitors in free state is the major contributor to the improvement of binding affinity with the GK domain. RSC chemical biology 2021, 2, 1274-1284, doi:10.1039/d1cb00087j.

9. Ahmed, A.; Mam, B.; Sowdhamini, R. DEELIG: A Deep Learning Approach to Predict Protein-Ligand Binding Affinity. Bioinformatics and biology insights 2021, 15, doi:10.1177/11779322211030364.

10. Yang, J.; Shen, C.; Huang, N. Predicting or Pretending: Artificial Intelligence for Protein-Ligand Interactions Lack of Sufficiently Large and Unbiased Datasets. Frontiers in Pharmacology 2020, 11, doi:10.3389/fphar.2020.00069.

11. Berman, H.M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T.N.; Weissig, H.; Shindyalov, I.N.; Bourne, P.E. The Protein Data Bank. Nucleic acids research 2000, 28, 235-242, doi:10.1093/nar/28.1.235.

12. Li, Y.; Liu, Z.; Li, J.; Han, L.; Liu, J.; Zhao, Z.; Wang, R. Comparative Assessment of Scoring Functions on an Updated Benchmark: 1. Compilation of the Test Set. Journal of chemical information and modeling 2014, 54, 1700-1716, doi:10.1021/ci500080q.

13. Li, Y.; Han, L.; Liu, Z.; Wang, R. Comparative Assessment of Scoring Functions on an Updated Benchmark: 2. Evaluation Methods and General Results. Journal of chemical information and modeling 2014, 54, 1717-1736, doi:10.1021/ci500081m.

14. Li, Y.; Su, M.; Liu, Z.; Li, J.; Liu, J.; Han, L.; Wang, R. Assessing protein–ligand interaction scoring functions with the CASF-2013 benchmark. Nature protocols 2018, 13, 666-680, doi:10.1038/nprot.2017.114.

15. Su, M.; Yang, Q.; Du, Y.; Feng, G.; Liu, Z.; Li, Y.; Wang, R. Comparative Assessment of Scoring Functions: The CASF-2016 Update. Journal of chemical information and modeling 2019, 59, 895-913, doi:10.1021/acs.jcim.8b00545.

16. Liu, Q.; Wang, P.-S.; Zhu, C.; Gaines, B.B.; Zhu, T.; Bi, J.; Song, M. OctSurf: Efficient hierarchical voxel-based molecular surface representation for protein-ligand affinity prediction. Journal of Molecular Graphics and Modelling 2021, 105, 107865, doi:10.1016/j.jmgm.2021.107865.

17. Son, J.; Kim, D. Development of a graph convolutional neural network model for efficient prediction of protein-ligand binding affinities. PLoS One 2021, 16, e0249404, doi:10.1371/journal.pone.0249404.

18. Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; Bennamoun, M. Deep learning for 3d point clouds: A survey. IEEE transactions on pattern analysis and machine intelligence 2020, 43, 4338-4364.

19. Meagher, D. Octree Encoding: A New Technique for the Representation, Manipulation and Display of Arbitrary 3-D Objects by Computer; 1980.

20. Li, S.; Zhou, J.; Xu, T.; Huang, L.; Wang, F.; Xiong, H.; Huang, W.; Dou, D.; Xiong, H. Structure-Aware Interactive Graph Neural Networks for the Prediction of Protein-Ligand Binding Affinity. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, August 14 - 18, 2021; pp. 975–985.

21. Wang, Y.; Wu, S.; Duan, Y.; Huang, Y. A point cloud-based deep learning strategy for protein-ligand binding affinity prediction. Briefings in bioinformatics 2022, 23, doi:10.1093/bib/bbab474.

22. Li, Y.; Rezaei, M.A.; Li, C.; Li, X. DeepAtom: A Framework for Protein-Ligand Binding Affinity Prediction. IEEE International Conference on Bioinformatics and Biomedicine (BIBM) 2019, 303-310, doi:10.1109/BIBM47256.2019.8982964.

23. Yang, Z.; Zhong, W.; Lv, Q.; Dong, T.; Yu-Chian Chen, C. Geometric Interaction Graph Neural Network for Predicting Protein–Ligand Binding Affinities from 3D Structures (GIGN). The Journal of Physical Chemistry Letters 2023, 14, 2020-2033, doi:10.1021/acs.jpclett.2c03906.

24. Francoeur, P.G.; Masuda, T.; Sunseri, J.; Jia, A.; Iovanisci, R.B.; Snyder, I.; Koes, D.R. Three-Dimensional Convolutional Neural Networks and a Cross-Docked Data Set for Structure-Based Drug Design. Journal of chemical information and modeling 2020, 60, 4200-4215, doi:10.1021/acs.jcim.0c00411.

25. Wang, R.; Fang, X.; Lu, Y.; Yang, C.-Y.; Wang, S. The PDBbind Database: Methodologies and Updates. Journal of medicinal chemistry 2005, 48, 4111-4119, doi:10.1021/jm048957q.

26. Hu, L.; Benson, M.L.; Smith, R.D.; Lerner, M.G.; Carlson, H.A. Binding MOAD (Mother Of All Databases). Proteins: Structure, Function, and Bioinformatics 2005, 60, 333-340, doi:10.1002/prot.20512.

27. Montavon, G.; Binder, A.; Lapuschkin, S.; Samek, W.; Müller, K.-R. Layer-Wise Relevance Propagation: An Overview. In Proceedings of the Explainable AI, 2019.

28. Karpov, P.; Godin, G.; Tetko, I.V. Transformer-CNN: Swiss knife for QSAR modeling and interpretation. Journal of Cheminformatics 2020, 12, 17, doi:10.1186/s13321-020-00423-w.

29. Nielsen, I.E.; Dera, D.; Rasool, G.; Ramachandran, R.P.; Bouaynaya, N.C. Robust Explainability: A tutorial on gradient-based attribution methods for deep neural networks. IEEE Signal Processing Magazine 2022, 39, 73-84, doi:10.1109/MSP.2022.3142719.

30. Hochuli, J.; Helbling, A.; Skaist, T.; Ragoza, M.; Koes, D.R. Visualizing convolutional neural network protein-ligand scoring. Journal of Molecular Graphics and Modelling 2018, 84, 96-108, doi:10.1016/j.jmgm.2018.06.005.

31. Liu, Z.; Li, Y.; Han, L.; Li, J.; Liu, J.; Zhao, Z.; Nie, W.; Liu, Y.; Wang, R. PDB-wide collection of binding data: current status of the PDBbind database. Bioinformatics (Oxford, England) 2014, 31, 405-412, doi:10.1093/bioinformatics/btu626.

32. Bournez, C.; Carles, F.; Peyrat, G.; Aci-Sèche, S.; Bourg, S.; Meyer, C.; Bonnet, P. Comparative Assessment of Protein Kinase Inhibitors in Public Databases and in PKIDB. Molecules 2020, 25, 3226, doi:10.3390/molecules25143226.

33. Jiménez-Luna, J.; Grisoni, F.; Schneider, G. Drug discovery with explainable artificial intelligence. Nature Machine Intelligence 2020, 2, 573-584, doi:10.1038/s42256-020-00236-4.

34. Lakshminarayanan, B.; Pritzel, A.; Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, California, USA, 2017; pp. 6405–6416.

35. Moon, S.; Zhung, W.; Yang, S.; Lim, J.; Kim, W.Y. PIGNet: a physics-informed deep learning model toward generalized drug–target interaction predictions. Chemical Science 2022, 13, 3661-3673, doi:10.1039/D1SC06946B.

36. Sieg, J.; Flachsenberg, F.; Rarey, M. In Need of Bias Control: Evaluating Chemical Data for Machine Learning in Structure-Based Virtual Screening. Journal of chemical information and modeling 2019, 59, 947-961, doi:10.1021/acs.jcim.8b00712.

37. Scantlebury, J.; Brown, N.; Von Delft, F.; Deane, C.M. Data Set Augmentation Allows Deep Learning-Based Virtual Screening to Better Generalize to Unseen Target Classes and Highlight Important Binding Interactions. Journal of chemical information and modeling 2020, 60, 3722-3730, doi:10.1021/acs.jcim.0c00263.

38. Ragoza, M.; Hochuli, J.; Idrobo, E.; Sunseri, J.; Koes, D.R. Protein-Ligand Scoring with Convolutional Neural Networks. Journal of chemical information and modeling 2017, 57, 942-957, doi:10.1021/acs.jcim.6b00740.

39. Li, H.; Leung, K.-S.; Wong, M.-H.; Ballester, P.J. Correcting the impact of docking pose generation error on binding affinity prediction. BMC Bioinformatics 2016, 17, 308, doi:10.1186/s12859-016-1169-4.

40. Boyles, F.; Deane, C.M.; Morris, G.M. Learning from Docked Ligands: Ligand-Based Features Rescue Structure-Based Scoring Functions When Trained on Docked Poses. Journal of chemical information and modeling 2022, 62, 5329-5341, doi:10.1021/acs.jcim.1c00096.

41. Hartshorn, M.J.; Verdonk, M.L.; Chessari, G.; Brewerton, S.C.; Mooij, W.T.M.; Mortenson, P.N.; Murray, C.W. Diverse, High-Quality Test Set for the Validation of Protein−Ligand Docking Performance. Journal of medicinal chemistry 2007, 50, 726-741, doi:10.1021/jm061277y.

42. Hassan-Harrirou, H.; Zhang, C.; Lemmin, T. RosENet: Improving Binding Affinity Prediction by Leveraging Molecular Mechanics Energies with an Ensemble of 3D Convolutional Neural Networks. Journal of chemical information and modeling 2020, 60, 2791-2802, doi:10.1021/acs.jcim.0c00075.

43. Dunbar, J.B., Jr.; Smith, R.D.; Damm-Ganamet, K.L.; Ahmed, A.; Esposito, E.X.; Delproposto, J.; Chinnaswamy, K.; Kang, Y.-N.; Kubish, G.; Gestwicki, J.E.; et al. CSAR Data Set Release 2012: Ligands, Affinities, Complexes, and Docking Decoys. Journal of chemical information and modeling 2013, 53, 1842-1852, doi:10.1021/ci4000486.

44. Dunbar, J.B., Jr.; Smith, R.D.; Yang, C.-Y.; Ung, P.M.-U.; Lexa, K.W.; Khazanov, N.A.; Stuckey, J.A.; Wang, S.; Carlson, H.A. CSAR Benchmark Exercise of 2010: Selection of the Protein–Ligand Complexes. Journal of chemical information and modeling 2011, 51, 2036-2046, doi:10.1021/ci200082t.

45. Jiménez, J.; Škalič, M.; Martínez-Rosell, G.; De Fabritiis, G. KDEEP: Protein–Ligand Absolute Binding Affinity Prediction via 3D-Convolutional Neural Networks. Journal of chemical information and modeling 2018, 58, 287-296, doi:10.1021/acs.jcim.7b00650.

46. Wang, Z.; Zheng, L.; Liu, Y.; Qu, Y.; Li, Y.-Q.; Zhao, M.; Mu, Y.; Li, W. OnionNet-2: a convolutional neural network model for predicting protein-ligand binding affinity based on residue-atom contacting shells. Frontiers in chemistry 2021, 913, doi:10.3389/fchem.2021.753002.

47. Karlov, D.S.; Sosnin, S.; Fedorov, M.V.; Popov, P. graphDelta: MPNN Scoring Function for the Affinity Prediction of Protein–Ligand Complexes. ACS Omega 2020, 5, 5150-5159, doi:10.1021/acsomega.9b04162.

48. Seo, S.; Choi, J.; Park, S.; Ahn, J. Binding affinity prediction for protein–ligand complex using deep attention mechanism based on intermolecular interactions. BMC Bioinformatics 2021, 22, 542, doi:10.1186/s12859-021-04466-0.

49. Jin, Z.; Wu, T.; Chen, T.; Pan, D.; Wang, X.; Xie, J.; Quan, L.; Lyu, Q. CAPLA: improved prediction of protein–ligand binding affinity by a deep learning approach based on a cross-attention mechanism. Bioinformatics (Oxford, England) 2023, 39, doi:10.1093/bioinformatics/btad049.

50. Wang, L.; Wu, Y.; Deng, Y.; Kim, B.; Pierce, L.; Krilov, G.; Lupyan, D.; Robinson, S.; Dahlgren, M.K.; Greenwood, J.; et al. Accurate and Reliable Prediction of Relative Ligand Binding Potency in Prospective Drug Discovery by Way of a Modern Free-Energy Calculation Protocol and Force Field. Journal of the American Chemical Society 2015, 137, 2695-2703, doi:10.1021/ja512751q.

51. Bietz, S.; Urbaczek, S.; Schulz, B.; Rarey, M. Protoss: a holistic approach to predict tautomers and protonation states in protein-ligand complexes. Journal of Cheminformatics 2014, 6, 12, doi:10.1186/1758-2946-6-12.

52. Da Silva, F.; Desaphy, J.; Rognan, D. IChem: A Versatile Toolkit for Detecting, Comparing, and Predicting Protein–Ligand Interactions. ChemMedChem 2018, 13, 507-510, doi:10.1002/cmdc.201700505.

53. Isert, C.; Atz, K.; Riniker, S.; Schneider, G. Exploring protein-ligand binding affinity prediction with electron density-based geometric deep learning. ChemRxiv preprint 2023, doi:10.26434/chemrxiv-2023-585vf.

54. Zheng, L.; Fan, J.; Mu, Y. OnionNet: a Multiple-Layer Intermolecular-Contact-Based Convolutional Neural Network for Protein–Ligand Binding Affinity Prediction. ACS Omega 2019, 4, 15956-15965, doi:10.1021/acsomega.9b01997.

55. Lim, J.; Ryu, S.; Park, K.; Choe, Y.J.; Ham, J.; Kim, W.Y. Predicting Drug–Target Interaction Using a Novel Graph Neural Network with 3D Structure-Embedded Graph Representation. Journal of chemical information and modeling 2019, 59, 3981-3988, doi:10.1021/acs.jcim.9b00387.

56. Kwon, Y.; Shin, W.-H.; Ko, J.; Lee, J. AK-Score: Accurate Protein-Ligand Binding Affinity Prediction Using an Ensemble of 3D-Convolutional Neural Networks. International Journal of Molecular Sciences 2020, 21, 8424, doi:10.3390/ijms21228424.

57. Wang, R.; Lai, L.; Wang, S. Further development and validation of empirical scoring functions for structure-based binding affinity prediction. J Comput Aided Mol Des 2002, 16, 11-26, doi:10.1023/a:1016357811882.

58. Korb, O.; Stützle, T.; Exner, T.E. Empirical scoring functions for advanced protein-ligand docking with PLANTS. Journal of chemical information and modeling 2009, 49, 84-96, doi:10.1021/ci800298z.

59. Gomes, J.; Ramsundar, B.; Feinberg, E.N.; Pande, V.S. Atomic convolutional networks for predicting protein-ligand binding affinity. arXiv preprint 2017, doi:arXiv:1703.10603.

60. Wu, Z.; Ramsundar, B.; Feinberg, Evan N.; Gomes, J.; Geniesse, C.; Pappu, A.S.; Leswing, K.; Pande, V. MoleculeNet: a benchmark for molecular machine learning. Chemical Science 2018, 9, 513-530, doi:10.1039/C7SC02664A.

61. Feinberg, E.N.; Sur, D.; Wu, Z.; Husic, B.E.; Mai, H.; Li, Y.; Sun, S.; Yang, J.; Ramsundar, B.; Pande, V.S. PotentialNet for Molecular Property Prediction. ACS Central Science 2018, 4, 1520-1530, doi:10.1021/acscentsci.8b00507.

62. Li, Y.; Yang, J. Structural and Sequence Similarity Makes a Significant Impact on Machine-Learning-Based Scoring Functions for Protein–Ligand Interactions. Journal of chemical information and modeling 2017, 57, 1007-1012, doi:10.1021/acs.jcim.7b00049.

63. Tosstorff, A.; Rudolph, M.G.; Cole, J.C.; Reutlinger, M.; Kramer, C.; Schaffhauser, H.; Nilly, A.; Flohr, A.; Kuhn, B. A high quality, industrial data set for binding affinity prediction: performance comparison in different early drug discovery scenarios. Journal of Computer-Aided Molecular Design 2022, 36, 753-765, doi:10.1007/s10822-022-00478-x.

64. Huang, N.; Shoichet, B.K.; Irwin, J.J. Benchmarking Sets for Molecular Docking. Journal of medicinal chemistry 2006, 49, 6789-6801, doi:10.1021/jm0608356.

65. Mysinger, M.M.; Carchia, M.; Irwin, J.J.; Shoichet, B.K. Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. Journal of medicinal chemistry 2012, 55, 6582-6594, doi:10.1021/jm300687e.

66. Chen, L.; Cruz, A.; Ramsey, S.; Dickson, C.J.; Duca, J.S.; Hornak, V.; Koes, D.R.; Kurtzman, T. Hidden bias in the DUD-E dataset leads to misleading performance of deep learning in structure-based virtual screening. PLOS ONE 2019, 14, e0220113, doi:10.1371/journal.pone.0220113.

67. Wallach, I.; Dzamba, M.; Heifets, A. AtomNet: A Deep Convolutional Neural Network for Bioactivity Prediction in Structure-based Drug Discovery. arXiv e-prints 2015, arXiv:1510.02855.

68. Koes, D.R.; Baumgartner, M.P.; Camacho, C.J. Lessons learned in empirical scoring with smina from the CSAR 2011 benchmarking exercise. Journal of chemical information and modeling 2013, 53, 1893-1904, doi:10.1021/ci300604z.

69. Bauer, M.R.; Ibrahim, T.M.; Vogel, S.M.; Boeckler, F.M. Evaluation and Optimization of Virtual Screening Workflows with DEKOIS 2.0 – A Public Library of Challenging Docking Benchmark Sets. Journal of chemical information and modeling 2013, 53, 1447-1462, doi:10.1021/ci400115b.

70. Wójcikowski, M.; Ballester, P.J.; Siedlecki, P. Performance of machine-learning scoring functions in structure-based virtual screening. Sci Rep 2017, 7, 46710, doi:10.1038/srep46710.

71. Chen, P.; Ke, Y.; Lu, Y.; Du, Y.; Li, J.; Yan, H.; Zhao, H.; Zhou, Y.; Yang, Y. DLIGAND2: an improved knowledge-based energy function for protein–ligand interactions using the distance-scaled, finite, ideal-gas reference state. Journal of Cheminformatics 2019, 11, 52, doi:10.1186/s13321-019-0373-4.

72. Ballester, P.J. Selecting machine-learning scoring functions for structure-based virtual screening. Drug Discovery Today: Technologies 2019, 32-33, 81-87, doi:10.1016/j.ddtec.2020.09.001.

73. Yasuo, N.; Sekijima, M. Improved Method of Structure-Based Virtual Screening via Interaction-Energy-Based Learning. Journal of chemical information and modeling 2019, 59, 1050-1061, doi:10.1021/acs.jcim.8b00673.

74. Riniker, S.; Landrum, G.A. Open-source platform to benchmark fingerprints for ligand-based virtual screening. Journal of Cheminformatics 2013, 5, 26, doi:10.1186/1758-2946-5-26.

75. Imrie, F.; Bradley, A.R.; van der Schaar, M.; Deane, C.M. Protein Family-Specific Models Using Deep Neural Networks and Transfer Learning Improve Virtual Screening and Highlight the Need for More Data. Journal of chemical information and modeling 2018, 58, 2319-2330, doi:10.1021/acs.jcim.8b00350.

76. Trott, O.; Olson, A.J. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. J Comput Chem 2010, 31, 455-461, doi:10.1002/jcc.21334.

77. Brocidiacono, M.; Francoeur, P.; Aggarwal, R.; Popov, K.; Koes, D.; Tropsha, A. BigBind: Learning from Nonstructural Data for Structure-Based Virtual Screening. ChemRxiv preprint 2022, doi:10.26434/chemrxiv-2022-2t0dq-v3.

78. Rohrer, S.G.; Baumann, K. Maximum Unbiased Validation (MUV) Data Sets for Virtual Screening Based on PubChem Bioactivity Data. Journal of chemical information and modeling 2009, 49, 169-184, doi:10.1021/ci8002649.

79. Pérez, A.; Martínez-Rosell, G.; De Fabritiis, G. Simulations meet machine learning in structural biology. Current opinion in structural biology 2018, 49, 139-144, doi:10.1016/j.sbi.2018.02.004.

## 3.2 Analysis of the PDBbind dataset

We conducted an investigation of the PDBbind dataset to understand its data distribution and identify potential biases within the dataset.

The PDBbind dataset includes proteins with varying sizes, ranging from 24 amino acids to 4,720 amino acids (*e.g.*, 1rbo). The distribution of protein sizes exhibits a peak around 250 amino acids (Figure 40 – A). Notably, there are approximately 200 proteins in the dataset that have more than 2,000 amino acids. Considering the use of MD simulations, these large protein structures are likely to demand significant computational resources.

Regarding the ligands in the dataset, there is a peak in the distribution of their molecular weights around 400 (Figure 40 – B). It is worth noting that the heavier molecular weights are often associated with peptides, but the dataset also contains macrolides and large sugars.

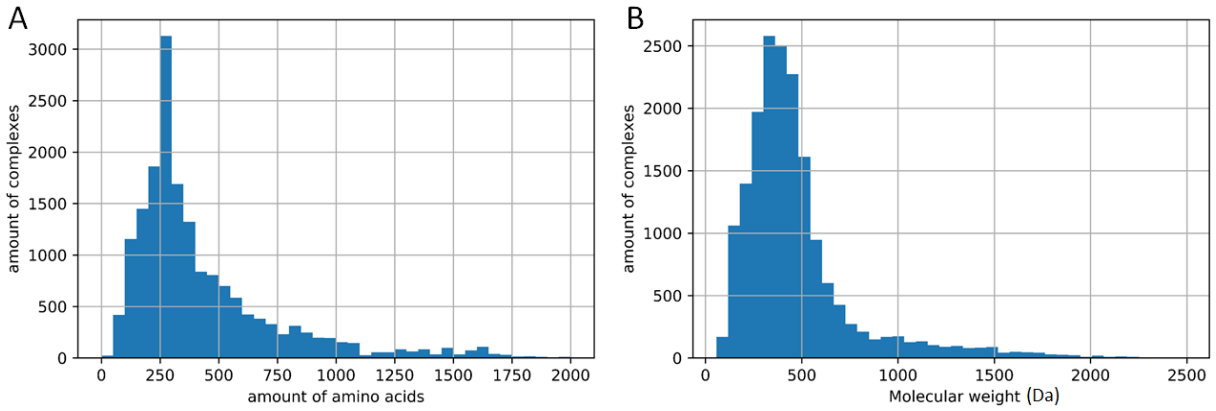



Figure 40: PDBbind complexes distribution. A – Distribution in function of the size of proteins. Proteins over 2,000 amino acids are not shown. B – Distribution in function of the molecular weight of molecules.

In Figure 41, we examined the distribution of protein families within the PDBbind dataset using the CATH classification (274). The CATH classification provides information about chains, in the case of heterodimer, information is provided for each domain. However, due to the lack of information about which chains interact with the ligand, we randomly selected the information about one chain to gain insight into the dataset's distribution. Our analysis revealed that certain protein families, such as Phosphorylase kinase and Phosphotransferase, were disproportionately represented. This overrepresentation could introduce bias into models trained on the PDBbind dataset.

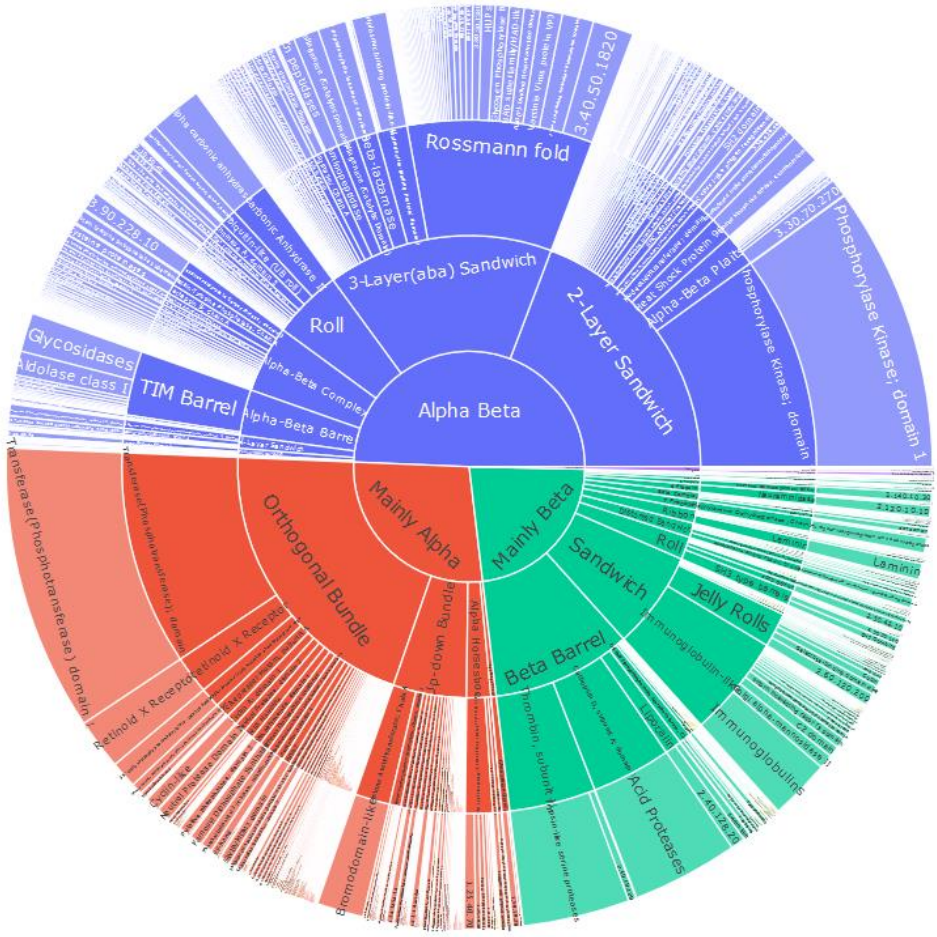



Figure 41: Sunburst representation of the distribution protein family inside the PDBbind using the CATH classification (274). The size of each section depends on the number of domains within each category.

Subsequently, we used the target information provided by the PDBbind, and discovered that approximately 3,000 protein families are represented in it. Interestingly, the first 30 protein families account for a quarter of the entire dataset. This observation suggests that models trained on the PDBbind dataset may be prone to overfitting on these specific protein families.

Additionally, we noticed a substantial presence of peptides within the dataset, totalling 2,915 ligands. Hence, we proceeded to examine the data distribution between peptides and small ligands. Both types of ligands exhibited a similar distribution across protein families, with the first 30 protein families collectively representing 25% of the data (Figure 42). However, it is important to note that the most represented protein families differ between small molecules and peptides. For small molecules, the top 5 protein families include carbonic anhydrase 2, HIV-1 protease, beta-secretase 1, bromodomain-containing protein 4, and heat shock protein HSP90-alpha. Conversely, for peptides, the top 5 protein families consist of HIV-1 protease, WD repeat-containing protein 5, Cetuximab FAB light chain, oligo-peptide binding protein, and glutamate carboxypeptidase 2.

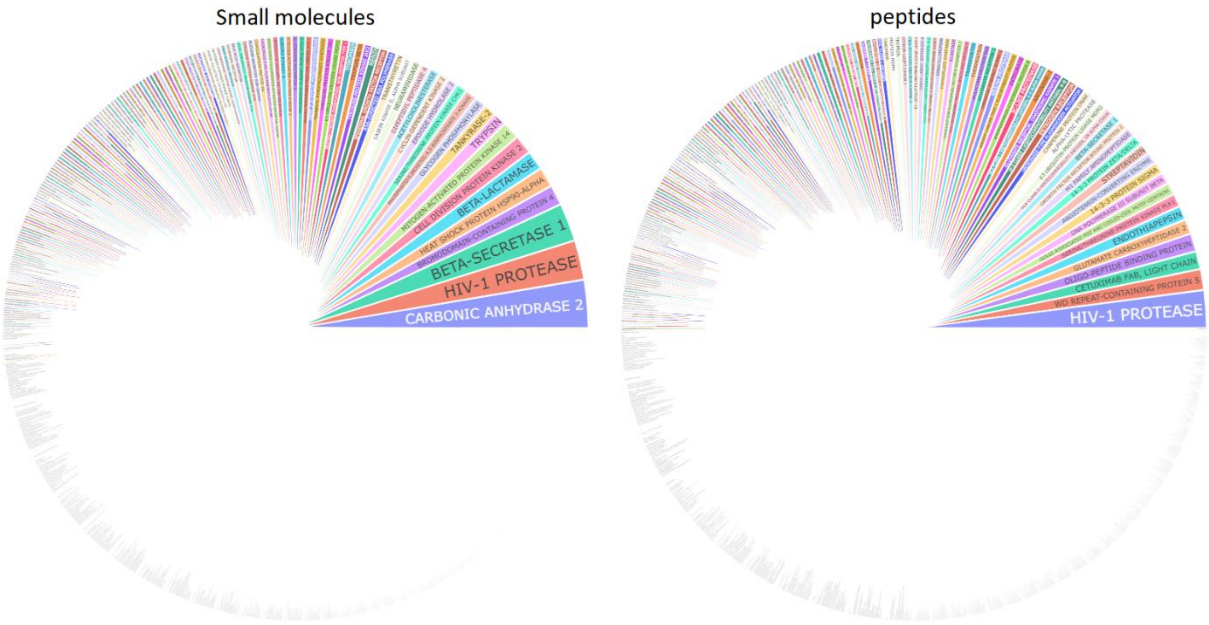



Figure 42: Sunbursts representing the protein families inside the PDBbind for complexes with small molecules and peptides. The bigger is a label, the more represented is the protein family.

As mentioned earlier, it is important to note that the molecular weights of peptides are generally much higher than those of small molecules (Figure 43 – A). Additionally, while the distributions in $pK_i$ values for small molecules and peptides are similar, there is a tendency for small molecules to have higher affinity, with a peak around 7-8 $pK_i$, whereas peptides tend to have a peak around 5-6 $pK_i$ (Figure 43 – B).

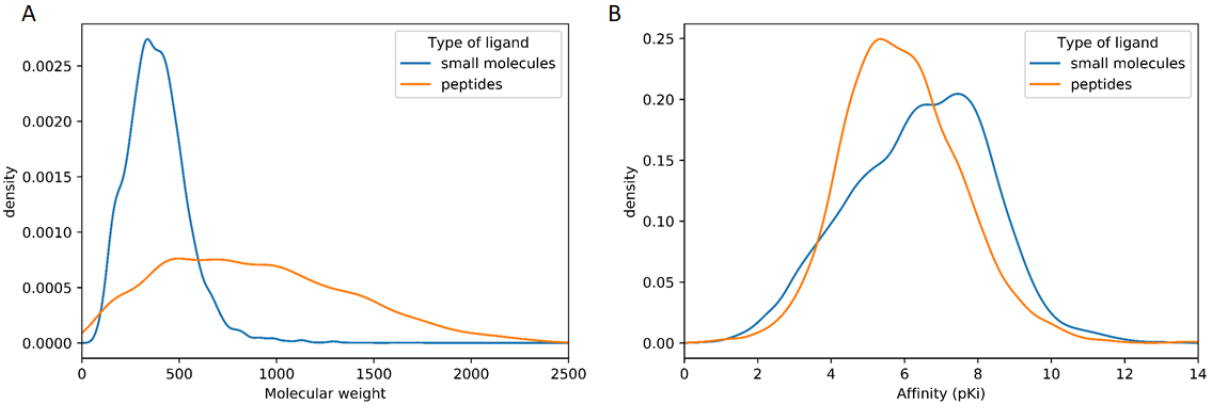



Figure 43: Comparing the distribution of ligand in the PDBbind. A – Distribution in function of the molecular weight. B – Distribution in function of the affinity ($pK_i$).

The observed differences in molecular weights, binding affinities, and distribution patterns between peptides and small molecules highlight the challenges in predicting one type of ligand when training on the other type. As mentioned in the publication, these findings suggest the importance of exploring the development of specialized models tailored to learning and predicting the affinity of specific types of ligands.

## 3.3 Analysis of the models performance

We conducted an analysis to evaluate performance differences between size 6 and size 12 pockets, aiming to identify trends in the prediction distribution and specific scenarios where better binding affinity predictions are achieved with size 12 pockets (Figure 44). The results suggest that when using size 12 pockets, we tend to overpredict the binding affinities of low-affinity complexes to a slightly lesser extent.

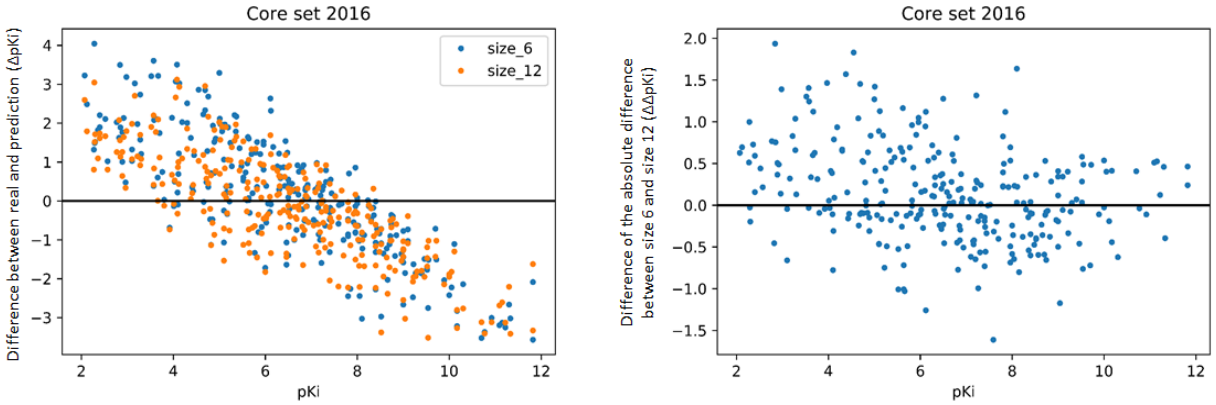



Figure 44: Comparison of the performance of Pafnucy trained on size 6 and size 12 pockets. A – Difference between real and prediction ($\Delta pK_i$). B – Difference of the absolute $\Delta pK_i$ between size 6 and size 12 pockets ($\Delta\Delta pK_i$). In the $\Delta\Delta pK_i$ plot, positive values mean that size 12 predictions are better, while size 6 predictions are better when values are negative. "Size" pockets are calculated by selecting all the amino acids at a specific distance from all ligand's atoms.

We attempted to discern the reasons behind the overall improved binding affinity predictions with size 12 pockets and visualized the complex that exhibited the most significant improvement (a two-log increase) when using size 12 pockets instead of size 6 pockets (Figure 45). As anticipated, there is an increase in the number of amino acids, although it is worth noting that most of the amino acids interacting with the ligand are already present in the size 6 pocket.

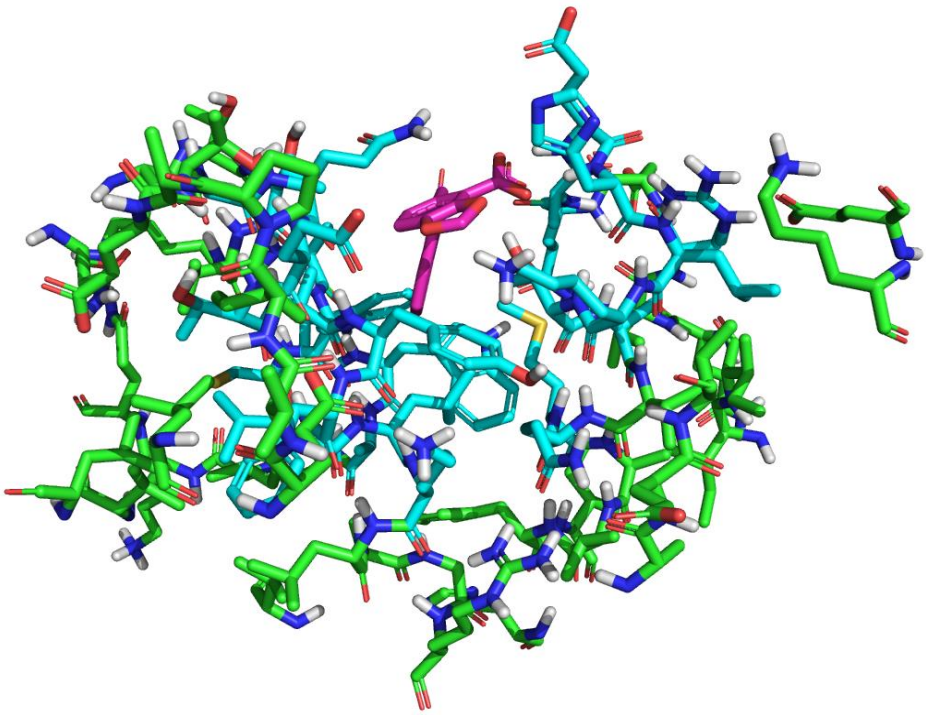


Figure 45: Size 6 and size 12 pockets from 3zt2. The size 6 pocket is displayed in cyan, while the rest of the size 12 pocket is in green. The ligand is shown in pink.

Therefore, we speculate that using size 12 pockets may primarily introduce additional information about the protein, potentially increasing the model's bias towards learning affinity patterns from the protein rather than from the interactions.

Thereafter, we compared the number of amino acids per pocket between different types and sizes of pockets (Figure 46). We observed that certain pockets exhibited similar quantities of amino acids, for instance, size 6 / CoG 10 pockets had an average of 25 amino acids, while size 8 / CoG 12 pockets had an average of 50 amino acids.

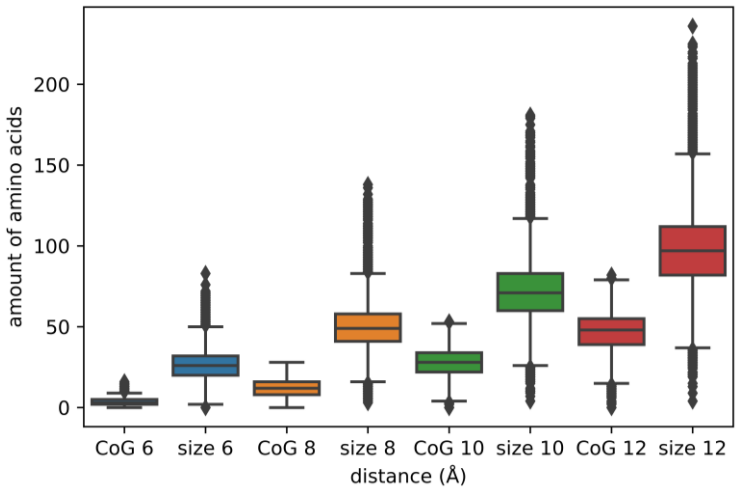



Figure 46: Boxplot of the number of amino acids per pocket size and type. “Size” pockets are calculated by selecting all the amino acids at a specific distance from all ligand’s atoms, while “CoG” pockets are obtained by selecting all the amino acids at a specific distance from the center of geometry of the ligand.

Surprisingly, we observed that Pafnucy achieved significantly worse predictions with size 6 pockets compared to CoG 10, while size 8 pockets achieved the same performance as CoG 12 (Figure 47). It is unexpected that size 6 and CoG 10 do not yield the same performance despite having the same number of amino acids.

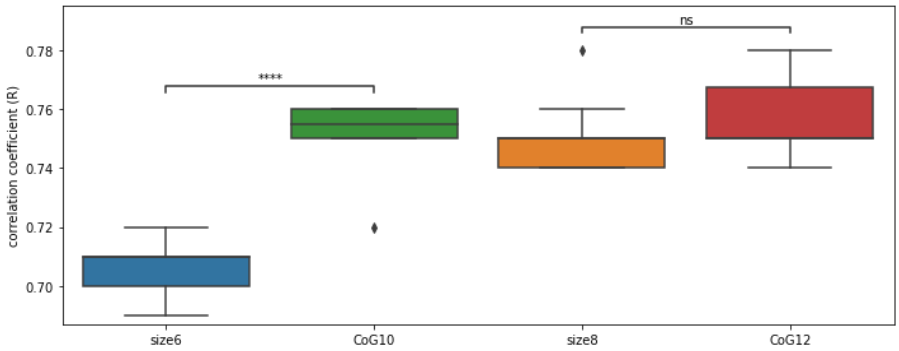



Figure 47: Boxplot of Pafnucy performance on pockets with similar number of amino acids.

We visually inspected the pockets created with varying ligand sizes. When comparing size 6 pockets to CoG 10 pockets, we noticed that CoG 10 pockets offer minimal additional information about the protein for small ligands (Figure 48 – A). As for long ligands, size 6 pockets provide crucial information about the protein that is absent in CoG 10 pockets (Figure 48 – B).

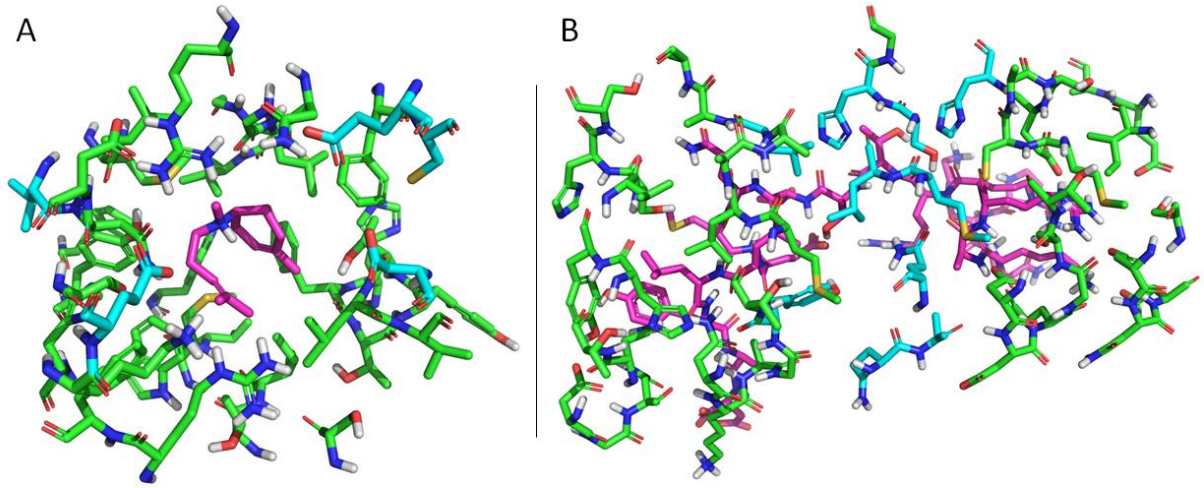



Figure 48: Comparison of size 6 and CoG 10 pockets for different ligand sizes. The size 6 pocket is displayed in green, while the CoG 10 pocket is shown in cyan. The ligands are in purple. A – Pockets obtained from a small ligand. The CoG 10 pocket is slightly bigger than the size 6 pocket. B – Pockets obtained from a big ligand. The size 6 pocket is much bigger than the CoG 10 pocket. In both cases (A and B), the larger pocket contains the smaller pocket and only the additional residues for the larger pocket are displayed.

Based on the visual inspection, it becomes evident that size 6 pockets are better suited than CoG 10 pockets for analysing protein-ligand interactions. However, it is important to note that pockets cannot be directly provided to the neural network, as they must fit into a grid of uniform size. Consequently, cubic boxes are created over the pockets, effectively truncating the sides of size 6

pockets when dealing with long ligands. This explains why the performance of size 6 pockets does not match or outperform CoG 10 pockets. It also accounts for the decreased performance on peptides, which are typically larger molecules. Attempting to train with boxes large enough to fully encompass size 6 pockets results in "out of memory" (OOM) errors. We also experimented with larger boxes, with sides measuring 40 Å instead of 25 Å, which led to very poor performance. Consequently, for the remainder of this work, we relied on calculating CoG 12 pockets, which can easily fit into 25 Å boxes.

In this first part of the PhD, we delved into utilizing the available data to its fullest extent. However, we contemplated whether the optimizations we implemented, especially concerning pocket selection, might have introduced bias into the model. In addition, we provided a list of datasets at the end of the publication that can be employed to assess model performance while mitigating the risk of biased results.

Nonetheless, the fundamental challenge in the field persists: data scarcity. Hence, our next undertaking was to augment the volume of data available for training our models.

# 4 Combining molecular dynamics and deep learning

The following part is a summary in French of the chapter “Combining molecular dynamics and deep learning”.

Comme précédemment exposé, nous avons investigué les meilleures façons d'utiliser les modèles de prédiction d’affinité basés sur la structure des complexes. Cette analyse nous a permis de mettre en lumière les limites des algorithmes actuels et de définir une voie d’amélioration pour ces outils. Ainsi, nous avons constaté que la qualité des données utilisées pour entraîner les modèles est le principal facteur limitant pour obtenir des modèles plus performants. En effet, tous les réseaux de neurones, quelle que soit leur architecture, présentent des lacunes similaires, notamment leur incapacité à prédire l’affinité en fonction des interactions présentes dans les structures 3D des complexes protéine-ligand. Lorsque les modèles sont entraînés en utilisant uniquement l'un des partenaires du complexe, ils obtiennent des performances comparables à ceux entraînés sur l'ensemble des complexes. Ces résultats soulignent le manque de pertinence des modèles existants, qui s'appuient principalement sur les biais présents dans la PDBbind et la CASF pour effectuer leurs prédictions.

Face à ce constat, nous avons décidé d’augmenter la quantité de données utilisées pour entrainer les modèles statistiques. L'objectif de cette démarche est d'incorporer de nouvelles données pour pallier aux limitations des modèles actuels. Cependant, il est difficile, voire impossible, d'obtenir expérimentalement et de manière rapide une grande quantité de structures de complexes protéine-ligand, accompagnées de leurs affinités respectives. Une approche computationnelle pour générer de telles données consiste à sélectionner des complexes dont l'affinité a déjà été déterminée *in vitro*, puis à effectuer un amarrage moléculaire lorsque la structure de la protéine est connue. Cependant, il est important de noter que les structures obtenues par amarrage moléculaire ne sont que des modèles (à ne pas confondre avec les modèles statistiques) et l'utilisation de telles informations peut introduire du bruit lors de l'entraînement des modèles statistiques.

Il convient de souligner que l'interaction protéine-ligand est un processus dynamique, et non statique. En prenant compte de cela, une approche alternative à l’amarrage moléculaire est de réaliser des simulations de dynamique moléculaire à partir des structures disponibles dans la PDBbind. En effet, ces simulations permettent de capturer la dynamique inhérente à l'interaction protéine-ligand. Ces informations dynamiques sont déjà exploitées par des méthodes telles que la « *Free Energy Perturbation »* (FEP) pour prédire l'affinité avec une grande précision. Par conséquent, il est envisageable de développer des réseaux de neurones capables d'exploiter ces simulations afin de créer des modèles statistiques moins biaisés et offrant des performances supérieures. Certains réseaux ont déjà été développés pour prédire l'affinité de complexes en analysant des descripteurs extraits de simulations de dynamique moléculaire (259). D'autres réseaux sont entraînés directement sur les images extraites des simulations pour effectuer des prédictions d'affinité sur des poses cristallographiques (275). Dans la continuité de ces efforts, nous avons mis au point un protocole combinant les simulations de dynamique moléculaire avec les algorithmes d'apprentissage profond pour prédire l'affinité des complexes protéine-ligand.

Tout d'abord, nous avons mis au point un jeu de données constitué de simulations de dynamique moléculaire. Ce jeu de données, baptisé MDbind, a été généré en réalisant 10 simulations de 10 ns sur 6 300 complexes provenant de la PDBbind. Cela a abouti à un total de 63 000 simulations, dont nous avons extrait 50 images pour chacune, représentant ainsi 3 000 000 d'images au total. Ces simulations

ont été réalisées à l'aide d'Amber20 (276) ainsi que des champs de force GAFF pour le ligand et FF14SB pour la protéine. Le solvant a été modélisé de manière explicite, en utilisant le modèle TIP3P pour les molécules d'eau.

Au total, plus de 200 000 heures de calcul ont été nécessaires pour réaliser les simulations et obtenir la MDbind, principalement en utilisant des cartes graphiques (GPU). Pour gérer cette charge de calcul, nous avons exploité les GPU disponibles dans notre laboratoire ainsi que les ressources de calcul régionales, telles que le centre « Myria » du CRIANN, et nationales, notamment le centre « Jean Zay » du CINES/GENCI/IDRIS. L'entraînement des modèles statistiques avec les réseaux de neurones, présentés plus tard dans ce résumé, a aussi nécessité plusieurs dizaines de milliers d'heures de calcul à partir de GPU. Pour réduire les temps de calcul, nous avons dû optimiser l'implémentation et l'utilisation des réseaux de neurones sur les serveurs, par exemple en utilisant les nœuds de calcul adéquats et en parallélisant les apprentissages.

La préparation des complexes, l'exécution des simulations et l'échange des données entre les serveurs ont été automatisés à l'aide de scripts développés en interne. Pour la préparation des complexes, nous avons utilisé la suite logicielle AmberTools20 (277), comprenant PDB4Amber, Antechamber, parmchk2 ainsi que Leap. Les charges partielles des ligands ont été ajoutées en utilisant la méthode AM1-BCC. Cependant, nous avons rencontré plusieurs problèmes au sein de la PDBbind, tels que la présence de mauvais ligands et de mauvais assemblages biologiques. De plus, de nombreux complexes n'étaient pas compatibles avec la réalisation de simulations de dynamique moléculaire, car certains atomes inhabituels des ligands ne sont pas paramétrables ou des boucles ne sont pas résolues au sein des protéines. Nous avons tenté de résoudre certains de ces problèmes, par exemple en automatisant l'ajout de boucles manquantes sur les protéines avec Modeller (44). Cependant, en raison de contraintes de temps, les simulations obtenues à partir de ces 2 200 complexes n'ont pas pu être incluses dans le jeu de données final.

Les systèmes ont été minimisés puis chauffés jusqu'à 300K avant d'être équilibrés en NVT / NPT (1 ns). À l'issue de l'équilibration, 10 réplicas de simulations ont été effectuées avec des graines différentes. Nous avons opté pour la réalisation de 10 réplicas de 10 ns au lieu d'une simulation unique de 100 ns. Cette approche a été préférée car elle offre une meilleure couverture de l'échantillonnage des conformations des complexes. Les données ont été échangées entre les serveurs par scp à l'aide de la bibliothèque python « paramiko » (278). Les images ont été extraites des simulations et les poches calculées à l'aide de Pytraj (279) et MDAnalysis (280). Le jeu de données de simulations représente 5 To de données, et une fois les images extraites et les poches calculées, cela représente 50 To de données.

Avant de passer au développement des réseaux de neurones, nous avons entrepris d'évaluer si certaines informations extraites des simulations étaient liées à l'affinité des complexes. Notre hypothèse était que la stabilité des ligands dans le site actif pouvait être corrélée à l'affinité. Pour tester cette hypothèse, nous avons calculé la racine de l'écart quadratique moyen (RMSD) du ligand par rapport à sa position initiale. Nous avons observé que le RMSD moyen des ligands présentant une faible affinité était plus élevé que celui des ligands ayant une forte affinité. Ces résultats suggèrent donc qu'il existe une corrélation entre la mobilité du ligand dans le site actif et l'affinité. Cette constatation nous conduit à penser que de telles informations pourraient être utilisées par des modèles statistiques pour améliorer la prédiction de l'affinité des complexes protéine-ligand.

Ainsi, nous avons exploré la possibilité d'utiliser les données disponibles dans la MDbind pour entraîner des modèles statistiques de prédiction de l'affinité en utilisant des réseaux de neurones. Deux approches ont été mises en œuvre :

- **Augmentation de données** : Dans cette approche, les images extraites des simulations sont utilisées pour enrichir le jeu de données d'entraînement existant. Cela signifie qu'on peut utiliser les réseaux de neurones actuels avec ces images, en leur assignant l'affinité du complexe expérimentalement obtenue. Ainsi, les modèles statistiques sont entraînés à partir de 3 000 000 de structures au lieu de 17 000 précédemment. Néanmoins, il est important de noter que le nombre de complexes utilisés pour entraîner les modèles reste le même. Cette approche permet de prendre en compte le caractère dynamique de l'interaction en présentant aux modèles de nombreuses poses extraites des simulations. Cependant, chaque pose est traitée indépendamment des autres, par conséquent l'aspect temporel de l'interaction n'est pas complètement restitué.
- **Apprentissage spatio-temporel** : Cette approche utilise les simulations dans leur intégralité en tant que données. Elle tient compte de la donnée temporelle reliant les images des simulations, permettant entre autres de suivre le mouvement des ligands dans leur poche au cours du temps et l'évolution temporelle des interactions. Cependant, cela nécessite le développement de nouveaux réseaux de neurones capables de traiter de telles données. Un inconvénient majeur de cette approche est la nécessité de réaliser des simulations à partir des structures des complexes que l'on souhaite évaluer, contrairement à l'approche d'augmentation de données où les prédictions sont réalisées sur une seule pose.

Pour cette étude, nous avons réimplémenté Pafnucy sous le nom de Proli en utilisant PyTorch, une bibliothèque moderne pour le développement de réseaux de neurones. Par la suite, nous avons développé Densenucy, une version améliorée de Pafnucy/Proli basée sur l'architecture DenseNet. Densenucy est ainsi plus compact et mieux adapté pour traiter l'information parcellaire contenue dans les structures des complexes protéine-ligand.

Tout d'abord, nous avons démontré que contrairement à la pratique courante dans le domaine, il n'est pas nécessaire d'effectuer des rotations systématiques (24 rotations du cube) des complexes pour obtenir les meilleures performances avec des 3D CNN et rendre les modèles statistiques plus robustes. En effet, des rotations aléatoires de ces structures sont suffisantes pour atteindre ce niveau de performance.

Par la suite, nous avons comparé les performances de Proli et Densenucy sur la CASF, entraînés avec et sans l'augmentation de données issues des simulations de dynamique moléculaire. Il semble que Proli ne bénéficie pas de cette approche. En revanche, malgré une forte variation, il a été possible d'obtenir des modèles performants avec Densenucy. La supériorité de Densenucy sur Pafnucy est d'autant plus marquée lorsque l'on utilise une méthode de consensus, moyennant les prédictions de 10 répliques de modèles statistiques. Nous avons observé précédemment que Pafnucy, ainsi que GraphBAR et OctSurf, ont tendance à faire des prédictions d'affinité situées dans la plage de 4 à 8 pKi. En revanche, il semble que Densenucy soit capable de mieux prédire l'affinité des complexes présentant une forte affinité, avec des prédictions pouvant atteindre jusqu'à 10 pKi. Ces résultats sont remarquables tant pour la précision des prédictions d'affinité (*scoring power*) que pour la capacité à classer correctement les complexes selon leur affinité (*ranking power*). Ces conclusions ont également été confirmées sur le jeu de données FEP, pour lequel les meilleures performances ont été obtenues par Proli et Densenucy entraînés avec l'augmentation de données issues des simulations de dynamique moléculaire.

Nous avons également évalué les performances de Proli et Densenucy, entraînés avec l'augmentation de données issues des simulations, sur des images extraites de ces simulations. Nous montrons ainsi qu'en moyennant les prédictions des images extraites d'une même simulation, il est possible d'obtenir des performances significativement meilleures qu'en réalisant des prédictions à

partir des structures cristallographiques sur le jeu d'évaluation de MDbind. Ce jeu d'évaluation est constitué de 41 500 images extraites de 830 simulations issues de 83 complexes de la CASF.

Pour réaliser l'analyse spatio-temporelle des simulations de dynamique moléculaire, nous avons développé deux réseaux de neurones, Timenucy et Videonucy. Timenucy est un « *long recurrent convolutional network* » (LRCN), tandis que Videonucy est un « *convolutional long short-term memory* » (ConvLSTM).

Timenucy se compose de deux réseaux de neurones en série : tout d'abord, un CNN qui analyse l'information spatiale des images extraites des simulations, puis un LSTM qui établit le lien temporel entre toutes ces images. En pratique, la composante LSTM de Timenucy ne reçoit de la part du CNN qu'une information prétraitée, qu'on peut assimiler à des prédictions de l'affinité pour chaque image de la simulation.

En revanche, dans le cas de Videonucy, les convolutions sont intégrées au mécanisme de LSTM, ce qui permet d'exploiter pleinement l'information spatiale et temporelle présente dans les simulations. Par conséquent, on s'attend à ce que ce réseau soit capable de mieux prédire l'affinité des complexes à partir de ces simulations.

Effectivement, nos résultats montrent que les performances de Videonucy sont supérieures à celles de Timenucy sur le jeu d'évaluation de MDbind. Cependant comparé à Pafnucy (qui est évalué sur les structures cristallographiques des 83 complexes présents dans ce jeu), Videonucy n'améliore pas les prédictions. Une explication possible de ces résultats réside dans la quantité limitée de simulations utilisées pour l'apprentissage des modèles. En effet, plus les données sont complexes (les simulations étant des données 4D extrêmement complexes), plus il est nécessaire d'avoir une quantité importante de données pour que les modèles puissent obtenir des performances optimales.

Par la suite, nous avons entraîné des modèles en utilisant uniquement un partenaire des complexes pour évaluer le degré de biais des modèles générés par les nouveaux réseaux de neurones (Proli, Densenucy, Timenucy et Videonucy). Les résultats obtenus sont équivalents à ceux de Pafnucy, suggérant que ces modèles partagent des biais similaires.

Cependant, nous avons aussi investigué l'utilisation de deux types de poches par Timenucy et Videonucy. Jusqu'alors, nous avions utilisé une poche fixe, déterminée par les acides aminés détectés autour du ligand dans la pose cristallographique, et appliqué à toutes les images de la simulation. Par la suite, nous avons calculé des poches qui suivent le mouvement du ligand au cours de la simulation. Dans le premier cas, on peut observer le ligand se déplacer dans la poche, voire même en sortir. Tandis que dans le second cas, le ligand est maintenu au centre de la poche, empêchant ainsi le modèle de percevoir les mouvements du ligand. Il s'est avéré que les performances obtenues avec des poches suivant le ligand étaient significativement inférieures par rapport à celles obtenus avec des poches fixes. Ce résultat suggère que les modèles statistiques développés avec Timenucy/Videonucy utilisent en partie les mouvements du ligand dans la poche pour prédire l'affinité des complexes. Cette observation laisse entrevoir que les méthodes spatio-temporelles pourraient être moins biaisées et tirer parti des informations temporelles pour réaliser leurs prédictions.

Finalement, plusieurs analyses supplémentaires ont été entreprises pour améliorer les performances des modèles. Par exemple, afin de réduire le biais lié à la surreprésentation de certaines familles de protéines au sein de la MDbind, nous avons exploré la possibilité de sous-échantillonner ces familles. De plus, nous avons mis en œuvre une distribution gaussienne des descripteurs atomiques pour contrer la nature clairsemée des structures de complexes. En outre, pour réduire les temps de calcul nécessaires à l'apprentissage des modèles, nous avons commencé à sélectionner les images

extraites des simulations en fonction de leur pertinence pour la prédiction de l'affinité (par exemple en se basant sur des empreintes d'interactions), au lieu d'utiliser l'ensemble des images disponibles. Toutefois, davantage d'investigations sont nécessaires pour évaluer pleinement ces initiatives.

En conclusion, cette étude a abouti à la création d'un jeu de données de simulations de dynamique moléculaire nommé MDbind, comprenant 63 000 simulations dont 3 000 000 d'images ont pu être extraites. Deux protocoles ont été développés pour exploiter ces données : l'augmentation de données et l'apprentissage spatio-temporel. En plus de réimplémenter Pafnucy avec PyTorch, nous avons également développé de nouveaux réseaux de neurones : Densenucy, Timenucy et Videonucy. Densenucy, entraîné avec l'augmentation de données issues des simulations de dynamique moléculaire, a démontré des performances similaires à celles de l'état de l'art. Celle-ci ont été encore améliorées en moyennant les prédictions obtenues pour des images extraites d'une même simulation. Bien que Timenucy et Videonucy aient affiché des performances moins optimales, nous suggérons que ces modèles sont capables d'analyser les mouvements du ligand pour effectuer leurs prédictions, ce qui les rend potentiellement moins biaisés que d'autres modèles. Ainsi, les deux protocoles développés dans cette étude offrent des résultats prometteurs. Avec un élargissement de la MDbind et le développement de réseaux de neurones plus modernes, tels que des vidéo-transformeurs, il est envisageable d'atteindre des performances proches de la FEP, tout en réduisant les coûts de calcul et en utilisant des modèles statistiques moins biaisés.

## 4.1 Background

To address the challenge of data scarcity in the field, various data augmentation strategies have been employed, including the incorporation of docking poses that closely resemble crystallographic poses into the training dataset (6, 254, 281, 282). Alternatively, Pérez *et al.* proposed to perform data augmentation using MD simulations (283), as illustrated in Figure 49. They introduced two data augmentation methods:

- **Obtain binding affinities for already determined 3D structure**: MD simulations can be applied to PDB structures lacking binding affinity data to assess affinity values (*e.g.* with FEP). As of 2018, this approach had the potential to expand the existing dataset by approximately 40,000 complexes. This estimation was calculated by excluding complexes already included in the PDBbind and Binding MOAD datasets from the pool of valid molecular complexes in the PDB.
- **Generate 3D structure for complexes with known binding affinities**: In cases where the 3D binding poses are not known, they proposed carrying out MD simulations for an extensive duration of 10 µs using adaptive sampling methods. This method may appear computationally intensive when compared to less costly molecular docking techniques. However, the latter cannot be readily employed as the binding sites have yet to be identified. By employing MD simulations, it becomes possible to expand the dataset by potentially more than a million structures. This number corresponds to the complexes with known affinities in the BindingDB, excluding those with 3D structures that are already determined and are in the PDBbind/binding MOAD datasets.

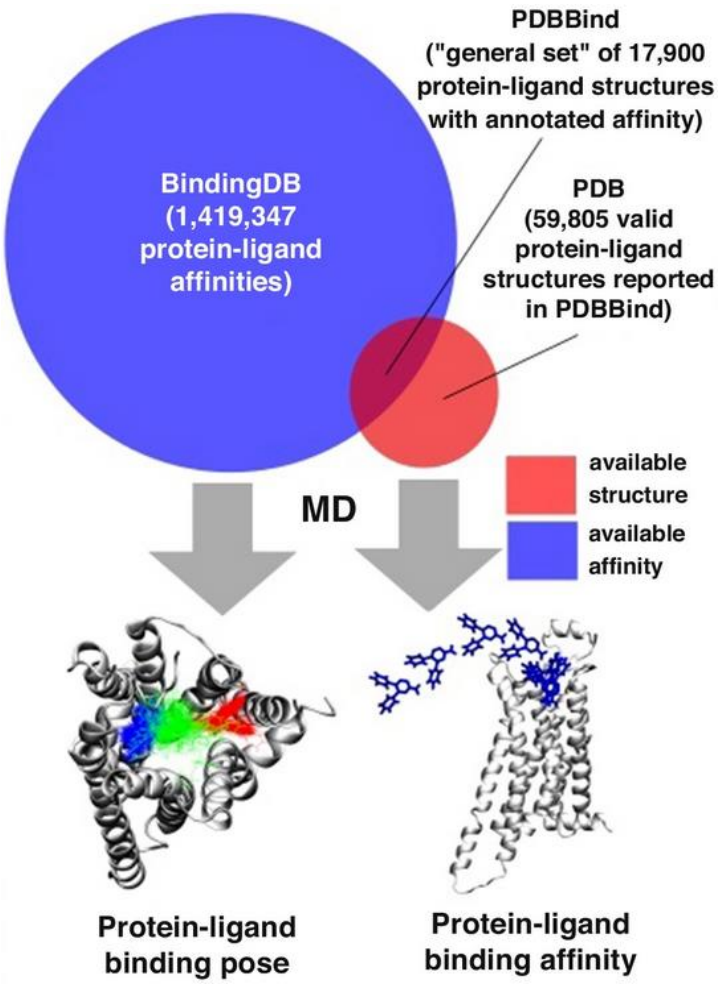



Figure 49: MD simulations used as a data augmentation tool for structural-based binding affinity predictions. Values from 2018. Taken from Pérez *et al.* (283).

However, it is important to note that all these data augmentation methods primarily used to provide static data. Given that protein-ligand interactions evolve over time, it becomes essential to incorporate dynamic information into binding affinity prediction models. The frames extracted from MD simulations can serve as a means for data augmentation, providing several examples of binding poses for the same complex. They can also provide the crucial temporal context that can significantly enhance model performance.

Therefore, we consider the construction of large MD datasets particularly beneficial for DL applications. The rise of DL/ML derived scoring functions and their reliance on static single structures to capture the dynamic process of protein ligand binding seemed an area for likely improvement. Recently two datasets were released, they are composed of MD simulations applied on complexes with known binding affinities:

- Plas-5k (284) comprises MD simulations carried out on 5,000 complexes, 2,000 of which have known binding affinities. They performed 5 simulation replicates of 4 ns per complex and extracted 40 snapshots per simulation. Complexes in interaction with either ligand or peptide were selected from the Protein Data Bank (39). Those with a resolution greater than 2.5 Å were discarded. MODELLER (285) and the H++ server (286) were used respectively for loop modelling and residue protonation. The entire database creation is displayed Figure 50. Using

OnionNet (5), models were trained to predict binding affinities in kcal/mol. Through a 10-fold validation, the models achieved a very high correlation coefficient of 0.947 but a quite poor RMSE of 5.7 kcal/mol (approx. 4 log units of error if converted to pK affinities). No external test set was used to examine the model performance in more realistic scenarios.

- MISATO (287) comprises MD simulations performed on 16,972 complexes from the PDBbind (v.2019). The simulations were run for 8 ns and 100 snapshots were taken from each trajectory. The molecules were optimized with semi-empirical quantum mechanical methods. For example, QM was used to refine the protonation states of proteins and ligands. As a mean to simplify the use of DL on MISATO, they provided PyTorch data loaders that are adapted for their data. To date, no binding affinity models were trained on this dataset.

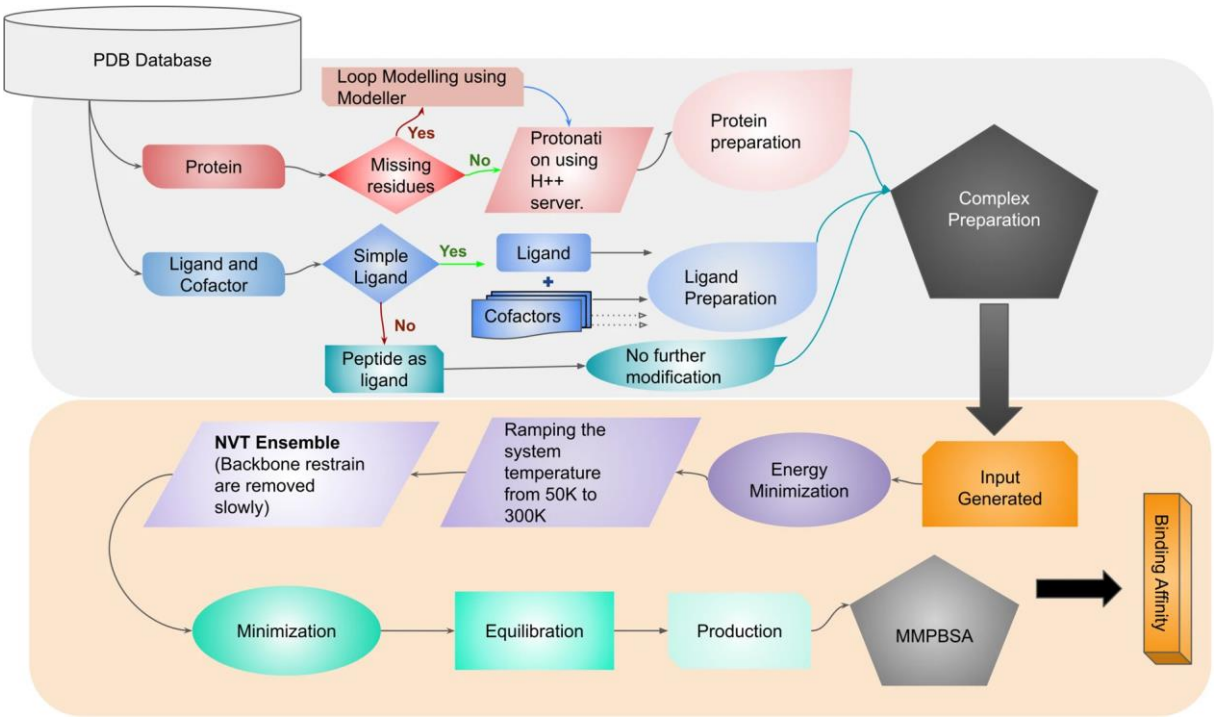



Figure 50: PLAS-5k creation workflow. Figure taken from Korlepara *et al*. (284)

Machine learning models have already been trained on 4D descriptors extracted from MD simulations (288, 289). There are a few recent examples of binding affinity prediction models trained on data obtained from MD simulations of protein-ligand complexes:

- Berishvili *et al*. developed a MLP, a CNN and a LSTM leveraging interaction descriptors computed from MD simulation frames (259). Their approach involved a single 2 ns simulation per complex, yielding 151 frames. Each frame assessed with 37 interaction descriptors, which included 14 descriptors calculated via GROMACS and 23 descriptors based on Smina scoring functions. The training dataset used simulations conducted on 356 complexes from the PDBbind v2017 refined set. They established two distinct test sets: one encompassing simulations performed on 57 complexes from a tankyrase inhibitors set, and another involving 31 complexes from the CSAR benchmark dataset (223). The CNN reached a correlation coefficient of 0.70 across both test sets, while the LSTM achieved a correlation coefficient of 0.58 on the tankyrase test set and a coefficient of 0.68 on the CSAR test set.

- ProtMD (290) is an E(3)-equivariant graph matching network pre-trained on 62,800 frames obtained from MD simulations (Figure 51). The simulations were performed on 64 protein-ligand complexes and lasted for 100 ns. The pre-training goals were to learn the flexibility information and temporal dependencies. To achieve this, they pre-trained the model for two specific tasks: firstly, the ability to order MD snapshots, and secondly, the capacity to generate novel frames. The pre-trained model was then fine-tuned to predict the binding affinities. To achieve this, they reused the dataset curated in the publication of ATOM3D (291). In that dataset, PDBbind complexes were split between a training, validation and test set based on a 30% sequence identity, leading to 3507, 466, and 490 complexes respectively. They predicted the binding affinity of complexes using their crystallographic poses, as well as docking poses obtained by re-docking ligands with EquiBind (195). They obtained a correlation coefficient of 0.6 and a RMSE of 1.367 $pK_i$ on crystallographic poses, as well as a correlation coefficient of 0.52 and a RMSE of 1.474 $pK_i$ on docking poses.
- Dynaformer (275) is a graph transformer, composed of a multi-head self-attention module and a feed-forward network. They created a dataset of MD simulations, which were performed on 3,218 protein-ligand complexes from the PDBbind. They did not carry out simulations on complexes with covalent/multiple/complicated ligands, with proteins having too many missing residues or membrane proteins. They performed a single MD simulation of 10 ns per complex; 100 frames were extracted from each simulation. Dynaformer is pre-trained on the extracted frames; it is subsequently fine-tuned on PDBbind complexes and used to carry out prediction on single structure. Nonetheless, they also proposed to predict binding affinity from MD simulations by averaging the predictions obtained for all the frames. By doing so, they obtain a more accurate binding affinity prediction of the complexes. They compared their results on the CASF 2016 and obtained a correlation coefficient of 0.858 and a RMSE of 1.114 $pK_i$.

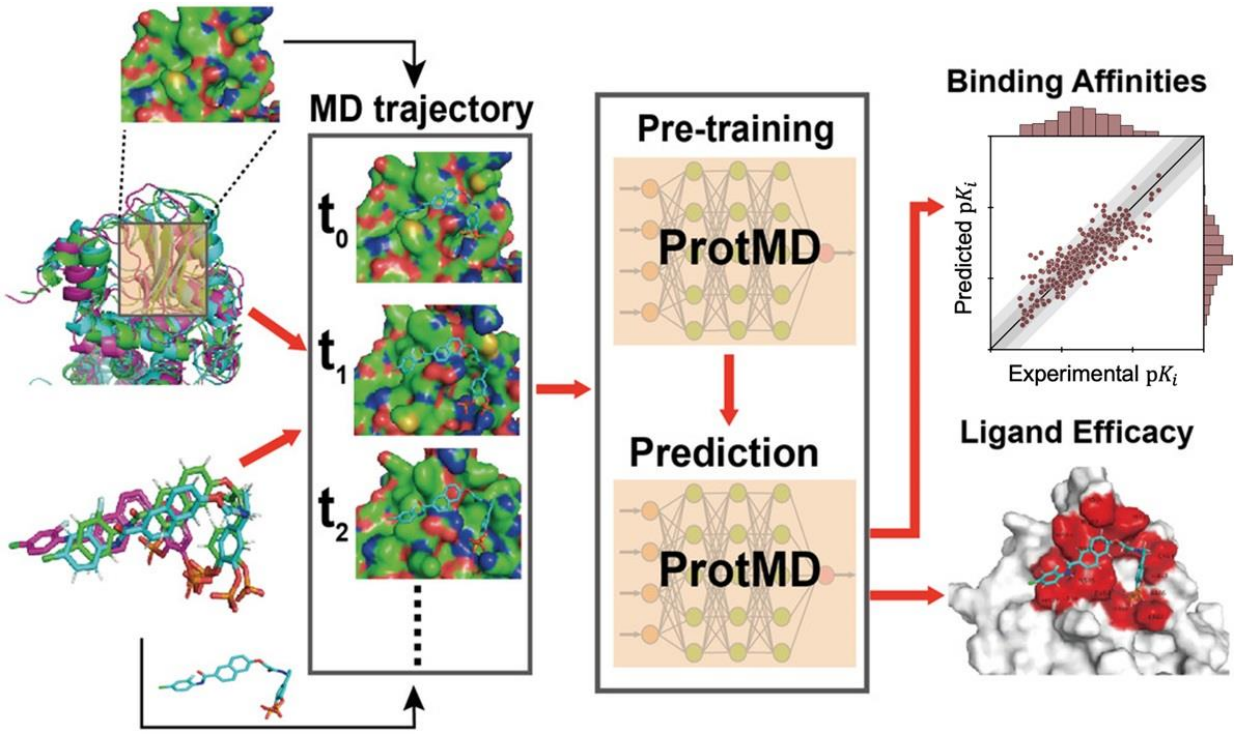



Figure 51: ProtMD pipeline. Models are trained on MD trajectories and used to predict binding affinities and ligand efficacy. Modified figure adapted from Wu *et al*. (290) and Min *et al.* (275)

## 4.2 Methodology development

In the aforementioned works, custom MD simulation datasets were developed to train the models with neural networks. In a similar way, we created our own MD simulation dataset from complexes with known binding affinities, dubbed MDbind, and developed neural networks able to train from this data. We introduce two innovative methodologies, the MD data augmentation and the spatio-temporal learning, to leverage this data for binding affinities predictions. The spatio-temporal learning method shares similarities with the Berishvili approach in that both involve making predictions based on simulations. However, in our case, neural networks directly analyse the 3D structures without the need for pre-calculated descriptors. The implementation of the spatio-temporal learning method led to the creation of two novel neural networks, Timenucy and Videonucy, able to train from entire simulations.

*Article*

# Spatio-temporal learning from MD simulations for protein-ligand binding affinity prediction

**Pierre-Yves Libouban [1], Camille Parisel [2], Maxime Song [2], Samia Aci-Sèche [1], Jose C. Gómez-Tamayo [3], Gary Tresadern [3,*], and Pascal Bonnet [1,*]**

[1] Institute of Organic and Analytical Chemistry (ICOA); UMR7311, Université d'Orléans, CNRS; Pôle de chimie rue de Chartres – 45067 Orléans Cedex 2, France; pascal.bonnet@univ-orleans.fr

[2] Institute for Development and Resources in Intensive Scientific Computing (IDRIS), CNRS; Rue John Von Neumann – 91403 Orsay Cedex, France

[3] Computational Chemistry, Janssen Research & Development; Janssen Pharmaceutica N. V.; B-2340 Beerse, Belgium; gtresade@its.jnj.com

**ABSTRACT**: In the field of computer-aided drug design, one of the principal objectives is to evaluate accurately protein-ligand binding affinities. While deep learning (DL) models have reached state-of-the-art performance in this task, the prediction of bioactivity using 3D structural information of protein-ligand complexes remains a major challenge, despite continuous evolution of DL methods. This challenge can be attributed to the lack of data to train the models and their inability to learn from protein-ligand interactions. To alleviate both problems, we hereby propose a protocol which combines molecular dynamics (MD) simulations with DL for binding affinity predictions. On one side, MD simulations were conducted to increase the existing amount of data used to train models. On the other hand, these MD simulations provide additional insights into the intermolecular interactions between proteins and ligands. For this purpose, we created MDbind, a dataset of 63,000 MD simulations and developed several DL architectures able to train from this dataset. Models trained with MD data augmentation demonstrated superior performance compared to current state-of-the-art methods, while spatio-temporal learning approaches achieved performance levels similar to those of state-of-the-art methods. These results are encouraging, and we foresee the work here to be of future value to the field of structure-activity modelling.

## INTRODUCTION

The accurate prediction of the binding affinity of protein-ligand complexes is a major goal in drug design. Such predictions serve as key criteria used to guide the selection of molecules for bioactivity measurement or to recommend new molecules to be synthetized by medicinal chemists. To prioritize molecules by their potential activity, a usual approach in the molecular modelling field is to dock them in the binding site of a protein; the resulting docking poses are then evaluated by their fit and the associated scoring function. Various scoring functions have been implemented for this purpose [1]. In the last decade, machine learning scoring functions were able to outperform other methods when benchmarked on datasets composed of experimentally determined protein-ligand complexes with known binding affinities [2]. In particular, deep learning (DL) models drastically improved the predictions on these test sets [3]. These models were mainly developed by using convolutional neural network (CNN) [4-11] or graph neural network (GNN) [12-16]; both designed to learn from the 3D structures of protein-ligand complexes.

Unfortunately, published performances are generally overly optimistic as models were evaluated on data for which biases have been identified [13]. Additionally, when applied on new or more challenging test sets the performance drops, hence limiting the usefulness of the developed models.

This poor generalizability is due to several factors, the first being the lack of substantial available data in the field. Most binding affinity prediction models are trained on experimental data that are compiled in the PDBbind dataset [17]. In its last version (v.2020), it is composed of 19,443 complexes with known binding affinities an increase of 6,135 complexes compared to v.2016. Unfortunately, this quantity of data is considerably lower than in other fields where DL is applied successfully, such as image classification, where typical datasets comprise on the order of millions of instances, such as ImageNet [18], currently composed of 14,000,000 images. The limitation on data availability, especially for challenging problems like binding affinity prediction, leads to low model performance and overfitting issues. Furthermore, the PDBbind dataset is sparse compared to drug discovery lead optimization use cases, lacking series of proteins interacting with the same ligand and series of ligands targeting the same protein. Thus, the protein-ligand affinity training is oblivious to the details that drive structure-activity relationships and existence of activity cliffs [13]. These limitations result in statistical models that typically do not learn from the actual protein-ligand interactions but instead memorize biases in the data, including patterns of affinity that correlate solely with the target proteins or ligands [13, 19]. Some models have been shown to perform similarly to conventional ligand-based QSAR [20,21]; therefore, alternative methods need to be investigated to obtain models able to learn on the protein-ligand interactions.

On the other hand, classical physics-based methods calculate the absolute binding free energies (ABFE) for the association of protein and ligand (ΔG), like LIE [22], MMPB(GB)SA [23] and variants of free energy perturbation (FEP) [24-26]. These methods calculate the binding free energies via molecular dynamics (MD) simulations that assess different conformational states of the protein and the ligand. Although ABFE FEP is approaching the accuracy of relative binding free energy (RBFE) calculations (ΔΔG around 1 kcal/mol root mean square error (RMSE)), it does have more inherent difficulties arising from the sampling required to assess the different energies of the relevant protein conformations, whether from *apo* to *holo*, or from different *holo* states. As such, the evaluation of the free energies requires heavy computation, especially with the FEP variants. Therefore, it is more suitable to apply these tools for a lead optimization process rather than for virtual screening.

Data augmentation has proven to be an effective strategy to mitigate some of the limitations of DL models in the field [16,27-29]. Since it is not possible to rapidly increase the amount of experimental data due to time and cost, many researchers have turned to MD simulations as a viable alternative. Consequently, two MD simulation datasets, called Plas-5k [30] and MISATO [31], were introduced to address this need. PLAS-5k [30] is composed of 5 simulation replicates lasting 4 ns each for a total of 5,000 complexes, with 2,000 of them having known binding affinities. OnionNet [32] was trained on this dataset and achieved a high correlation coefficient $R^2CV$ of 0.947 with a 10-fold cross-validation, although with a poor RMSE of 5.7 kcal/mol (approx. 4 log units of error if converted to pK affinities). On the other hand, MISATO [31] encompasses MD simulations of 8 ns performed on 16,972 complexes from the PDBbind (v.2019).

Multiple machine learning models were implemented to train on 4D descriptors extracted from MD simulations [33,34]. These 4D descriptors, such as MD fingerprints [35], can be generated by calculating 3D molecular descriptors (e.g., solvent accessible surface area (SASA), radius of gyration, or potential energies) for each frame of the simulation and subsequently calculating their average or concatenating them into a vector. In addition, a few DL models were also developed, including ProtMD [36], an E(3)-equivariant graph matching network that underwent pre-training on 62,800 frames obtained from MD simulations lasting 100 ns on 64 protein-ligand complexes. After fine-tuning using 90% of the ATOM3D [37] binding affinity dataset, this model achieved a correlation coefficient of 0.6 and a RMSE of 1.4 pKi on the 10% remaining dataset. In another study from Berishvili *et al.* [38], a MLP

(Multi-Layer Perceptron), a CNN and a LSTM (Long Short Term Memory) models were trained on MD simulations of 2 ns performed on 356 complexes from the PDBbind dataset. Using molecular descriptors extracted from the MD simulations, they achieved correlation coefficients of 0.70 and 0.68 with the CNN and the LSTM, respectively, on the CSAR (Community Structure-Activity Resource) test set. The Dynaformer [39] method is a graph transformer model that was pre-trained on frames extracted from MD simulations, which lasted 10 ns and were performed on 3,218 complexes from the PDBbind. It was subsequently fine-tuned on PDBbind complexes and reached a correlation coefficient of 0.86 and a RMSE of 1.1 pKi on the PDBbind v.2016 core set.

In a similar way, we have established a protocol combining MD simulations with DL algorithms. We created a dataset of MD simulations, called MDbind, which encompasses 63,000 simulations. This was achieved by conducting 10 replicate simulations of 10 ns per complex, for 6,300 complexes. Beyond simply expanding the dataset, the MD simulations provide physical insights of protein-ligand interactions. They are based on an all-atom parameterized system and capture the dynamics of the protein ligand complex. One of the objectives of this protocol was to enable DL models to discern differences between high and low affinity ligands and to capture the variations of their interactions, where most models trained on static single 3D protein-ligand complexes often fail to capture. Subsequently, we developed a series of neural networks to harness this data for the prediction of binding affinities. These neural networks use the MD simulations in two distinct ways. First, frames can be extracted from simulations and considered independently from each other as new structures while being labelled with the binding affinity of the initial complex. Hence, MD simulations act as a data augmentation method compatible with the current state-of-the-art neural networks. Second, it is possible to train models from the whole simulations. In this case, the simulations are labelled with the binding affinity of the initial complex. We coined the term 'spatio-temporal learning' for this training methodology. This method exploits fully the temporal information contained sequentially in the frames of the simulations. For that purpose, we have developed two neural networks, a long-term recurrent convolutional network (LRCN) and a convolutional long short-term memory (ConvLSTM), both able to carry out binding affinity predictions by analysing simulations. Models trained with MD data augmentation outperformed current state-of-the-art methods, while spatio-temporal learning approaches are reaching performance similar to state-of-the-art methods.

## MATERIALS AND METHODS

### DATASETS

The PDBbind dataset (http://www.pdbbind.org.cn) [17] comprises 3D structures of protein-ligand complexes with known binding affinities. These complexes were used in this study for two purposes; first, to carry out MD simulations and, second to train and evaluate DL models. We used the PDBbind (v.2019) general set (17,679 complexes) which includes the refined set (4,852 complexes). This latter set includes the core set (v.2016) which consists of 285 complexes. The core set, also referred as CASF [2], is a commonly used "scoring benchmark" for statistical modelling. It is composed of 57 protein clusters based on a 90% sequence homology. Each cluster contains 5 complexes with a wide range of binding affinities. All the PDBbind complexes were made available with several modifications with regards to the original structures from the Protein Data Bank (http://www.rcsb.org/) [40]. For example, they provide the biological assemblies. Further information is available in the PDBbind "readme" file.

In this work, we created the MDbind dataset, which is composed of 63,000 MD simulations. It was obtained by carrying out 10 replicates of MD simulations of 10 ns each on 6,300 complexes from the PDBbind dataset. We carried out several short simulations per complexes to allow a better sampling of the conformational space and to decrease the uncertainty for binding affinity predictions [41-44]. Also, it has been shown that repeated short MD simulations were able to differentiate true from false binding poses [45-48]. Running such a number of MD simulations requires the development of an adapted workflow. However, despite the complex preparation protocol, various factors hindered the successful execution of the simulations on the entire PDBbind. For example, we carried out simulations only on fully determined protein structures and complexes where proteins have missing residues were discarded. The MD simulation protocol can be divided in three steps: the preparation of the protein-ligand complexes, the pre-production and the production.

- Preparation of the protein-ligand complexes: We used AmberTools20 [49] to prepare both proteins and ligands. As mentioned earlier, some preparation steps have already been performed by PDBbind. For example, they provided the biological assemblies, and both the proteins and ligands were protonated at neutral pH. However, we created a generalized protocol suitable for all complexes of different datasets. Therefore, protein hydrogens were reintroduced, and disulfide bonds were established using PDB4amber. Some ligand atoms were renamed to be readable by Antechamber [50] and some ligand valence issues were corrected. Thereafter, we used Antechamber to add partial charges to the ligands with AM1-BCC method [51]. We used frcmod files from various sources [52-64] and also employed parmchk2 to generate force field parameter files for ligands. Finally, leap was used to create the topological files of the complexes. The ff14SB force field [65] was applied for proteins, and the general amber force field (gaff) [66] was used for ligands. Simulations were carried out with explicit solvent; thus, systems were solvated in a TIP3P [67] solvent box of 10 Å radii from any atoms of the system. Finally, the systems were neutralized by adding Na+ and Cl- ions.
- Pre-production: The simulations were performed with the Amber20 MD package [49]. Systems were minimized in three steps. First, with restraint on the complexes, then with restraint on the water, and finally without restraint. Subsequently, systems were heated to 300K in three steps. Each heating step increased the temperature by 100K, with restraint on the complexes. Lastly, the equilibration of the systems was performed in five steps. For the three first steps, constraints were applied on the complexes. Equilibrations were performed first in NVT and then in NPT, and lasted 2 ns in total. We used the Berendsen thermostat and barostat, and the pressure was set to 1 atm.
- Production: We conducted 10 replicate simulations of 10 ns for each complex. They were performed in NPT, with the SHAKE algorithm [68] and using the particle mesh Ewald MD (PMEMD) engine [69]. The PME distance cutoff was set to 10 Å. A new seed was generated for each replicate by performing a short equilibration.

MD simulations were processed to be compatible as input for the neural networks. We extracted 50 frames from each simulation, resulting in 3,000,000 frames from the 63,000 simulations. Models trained on protein-ligand 3D data typically focus on the binding pockets, the regions where ligand interact with the protein. By adopting this approach, we effectively reduced RAM (random-accessed memory) and computational power consumption, concentrating solely on the part of the protein involved in binding. In previous work [19], we identified the optimal pocket definition as the one obtained by the selection of residues located at 12 Å from the geometric center of the ligand in the

crystallographic poses. After defining the pocket, we extracted the selected residues from all frames of the simulation. Consequently, the pockets remain consistent throughout the simulation. We used a Pymol [70] script to extract pockets from crystallographic poses, while both Pytraj [71,72] and MDAnalysis [73] were employed to process the frames. Thereafter, we followed the data preparation workflow of Pafnucy [4], a CNN used for binding affinity prediction. Pocket pdb files were converted to mol2 files with Chimera (https://www.rbvi.ucsf.edu/chimera) [74]. The information containing the 3D coordinates of atoms from both pockets and ligands were stored in two types of datasets depending on the type of neural network used:

- For molecular dynamics data augmentation, we created HDF5 (hierarchical data format) datasets following the Pafnucy workflow [4]. They are composed of pockets and ligands from both crystallographic poses and frames. The downside of such datasets is that all the data is loaded into RAM at the same time during training leading potentially to memory failures depending on hardware architecture.
- For spatio-temporal learning we need to train on the entire simulations. In that case, a numpy file is created per simulation. This leads to data being loaded sequentially. While this approach may be slower in processing the data, it is also less demanding in terms of RAM.

The datasets were split with a ratio of 80/20 between training and validation set. Within the molecular dynamics data augmentation framework, the complexes in the validation set, and their subsequent MD simulation frames, were randomly selected from the PDBbind refined set composed of a total of 4,852 complexes. In a similar fashion with spatio-temporal learning, the validation set comprises simulations of complexes randomly picked from the refined set. All extracted frames or simulations of a complex are exclusively allocated to either the training or the validation set. This approach ensures that no information about a complex is present in both sets. In some cases, extracting the frames and calculating the pockets proved to be difficult, so we discarded the frames in MD data augmentation or the entire simulations in spatio-temporal learning. Thus, in the subsequent figures, a 50-fold increase between the number of simulations and frames, as well as a 10-fold increase between the number of complexes and simulations, is not consistently observed.

For the MD data augmentation, the sets used in this work are composed of:

- validation set: the crystallographic poses of 1,198 complexes randomly selected from the refined set (4,852 complexes) and 585,372 frames extracted from 11,940 simulations (1,194 complexes)
- training set: the crystallographic poses of 16,076 complexes selected from the general set (17,679 complexes) without the validation set (1,198 complexes) and 2,340,237 frames extracted from 47,501 simulations (4,751 complexes)
- test set: the crystallographic poses of 285 complexes of the PDBbind v.2016 core set

For the spatio-temporal learning, the sets are composed of:

- training set: 46,632 simulations performed on 4,753 complexes
- validation set: 11,668 simulations performed on 1,179 complexes
- test set: 830 simulations performed on 83 complexes. This test set is referred to as MDbind test set. The 41,500 frames extracted from this test set are also used to evaluate MD data augmentation methods.

The PDBbind v.2016 core set, the FEP dataset [75] and the MDbind test set were used as test sets to evaluate the DL models' performance. The FEP dataset comprises 200 protein ligand complexes from 8 different proteins (BACE, CDK2, JNK1, MCL1, PTP1B, Thrombin, TYK2 and P38) and a limited amount of ligand scaffolds. Each series of molecules targets the same protein, while exhibiting a wide range of binding affinities. This allows for the evaluation of models in a lead optimization scenario.

To obtain the MDbind test set, simulations had to be carried out on the complexes of the PDBbind v.2016 core set. Since we removed complexes where proteins have missing residues, the simulations were conducted on 83 complexes of the PDBbind v.2016 core set. Hence, spatio-temporal learning model performance was evaluated on 30% of the core set complexes.

**NEURAL NETWORKS:**

Pafnucy [4], a well-known 3D CNN developed for binding affinity predictions, was used to train models with MD data augmentation. A box is created around the ligand accounting for the pocket information. By default, each side of the cubic box measures 25 Å. The box space is discretized in voxels of 1 $Å^3$. Voxels in contact with atoms were assigned atomic features. The following 19 features were used to describe atoms:

- 9 bits (one-hot or all null) encoding atom types: B, C, N, O, P, S, Se, halogen and metal
- 1 integer (1, 2, or 3) with atom hybridization: hyb
- 1 integer counting the numbers of bonds with other heavy atoms: heavy_valence
- 1 integer counting the numbers of bonds with other heteroatoms: hetero_valence
- 5 bits (1 if present) encoding properties defined with SMARTS patterns: hydrophobic, aromatic, acceptor, donor and ring
- 1 float with partial charge: partial charge
- 1 integer (1 for ligand, -1 for protein) to distinguish between the two molecules: moltype
- Each atomic feature corresponds to a channel in the CNN, and the convolutions are performed on the voxels.

SE(3)-equivariant neural networks which are independent of rotational and translational motion in space, are able to output the same results on 3D structures that have been translated or rotated. Since Pafnucy is not SE(3)-equivariant, it is required to train models with rotations of the complexes to optimize their performance. Pafnucy was trained on all the 90° rotations of the cube. This rotational data augmentation is obtained by performing 24 rotations. Unfortunately, employing both the systematic rotations and the MD data augmentation to train models, results in an excessively high computational cost. Therefore, we used only one random rotation of each complex when training the models.

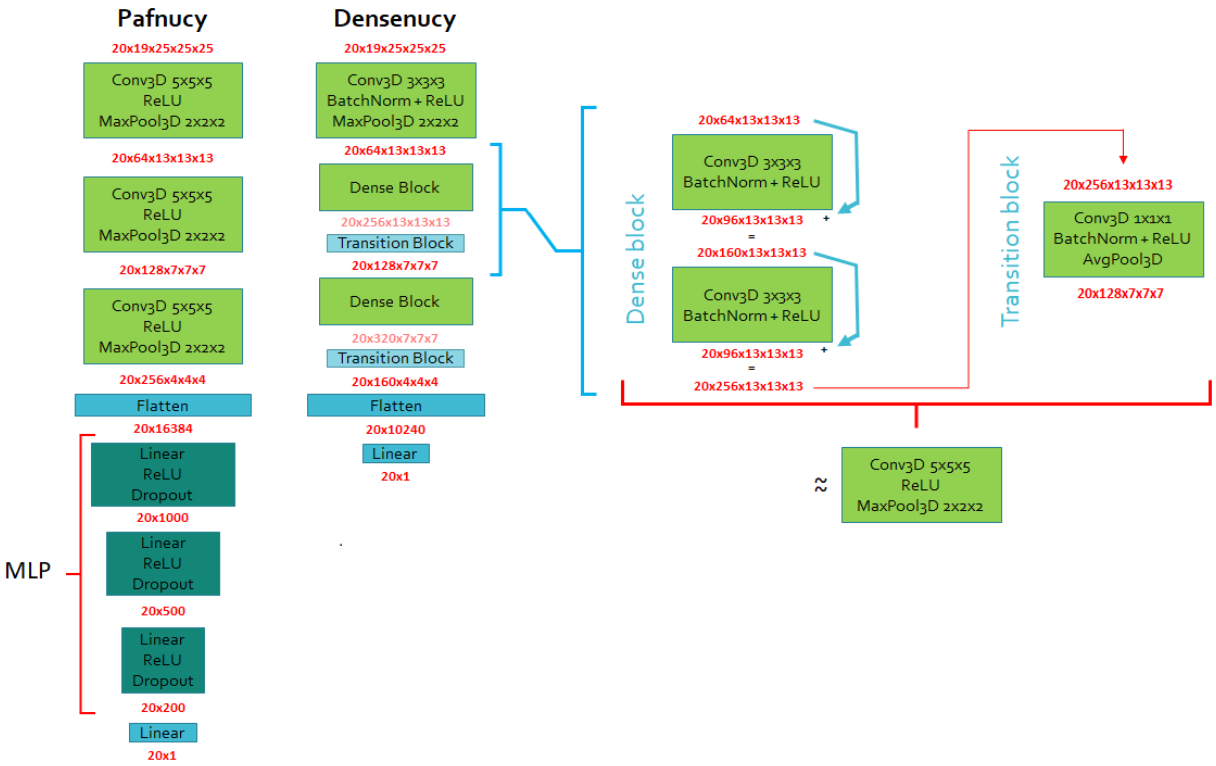



**Figure 1.** Comparison of Pafnucy and Densenucy. A detailed breakdown of the dense and transition blocks is displayed. The dimensions of layers' input and output are displayed in red. The first number represents the batch size, followed by the number of channels and the dimension of the box. The flatten layer compresses the spatial dimensions into the channel dimension. The convolutional layers are shown in light green, while the multi-layer perceptron (MLP) is in dark green. The flatten and output layers are in cyan. Conv3D is an abbreviation for 3D convolution. MaxPool3D and AvgPool3D refers to 3D max pooling and 3D average pooling, respectively. BatchNorm stands for batch normalization, and ReLU is short for rectified linear unit.

CNNs were used to extract information directly from the raw 3D structure data and learn from them, specifically from the frames generated during simulations, in order to avoid introducing bias through the use of expertly crafted descriptors. Pafnucy [4] was the starting point for the development of the other neural networks presented here. Since Pafnucy was originally developed in Tensorflow 1.2, we re-implemented Pafnucy using a more modern deep learning framework. Thus, we migrated the code in PyTorch on the HPC center (http://www.idris.fr/jean-zay/ HPE SGI 8600) and named the resulting neural network Proli. Following the example set by Imrie *et al*. with dense networks [76], we have developed an updated version of Pafnucy called Densenucy (**Figure 1**). All the neural networks presented here were developed using PyTorch version 1.11.0. 5×5×5 convolutional and 2×2×2 max pooling layers in Pafnucy were replaced in Densenucy by a 3×3×3 convolutional input layer, two dense blocks and two transition blocks. The size of filters was reduced to lower memory load and computational cost. The first convolutional layer is composed of 64 filters with sides of 3 Å. It is followed by dense blocks, that were introduced with DenseNet [77]. The dense block is a modern framework that allowed reaching better performance in computer vision. It facilitates scaling up the neural networks size by adding convolutional layers. Furthermore, it mitigates the loss of information during the convolution. This is achieved by adding the input of layers to the output, therefore preserving the initial information. All convolutional layers are the same in the dense block, their

number of neurons is defined by a growth rate. In the end, Densenucy is more compact than Pafnucy with a lower number of parameters.

Dense blocks are composed of two 3×3×3 convolutional layers, with a growth rate of 96. The input of each convolutional layer was appended to the output. The transition blocks were used to halve the number of channels and dimensions. It is composed of a 1×1×1 convolutional layer with a 2×2×2 average-pooling layer. A 3D batch normalization was added after each 3D convolution layer. Lastly, the multi-layer perceptron (MLP) was removed, as the performance was better without it.

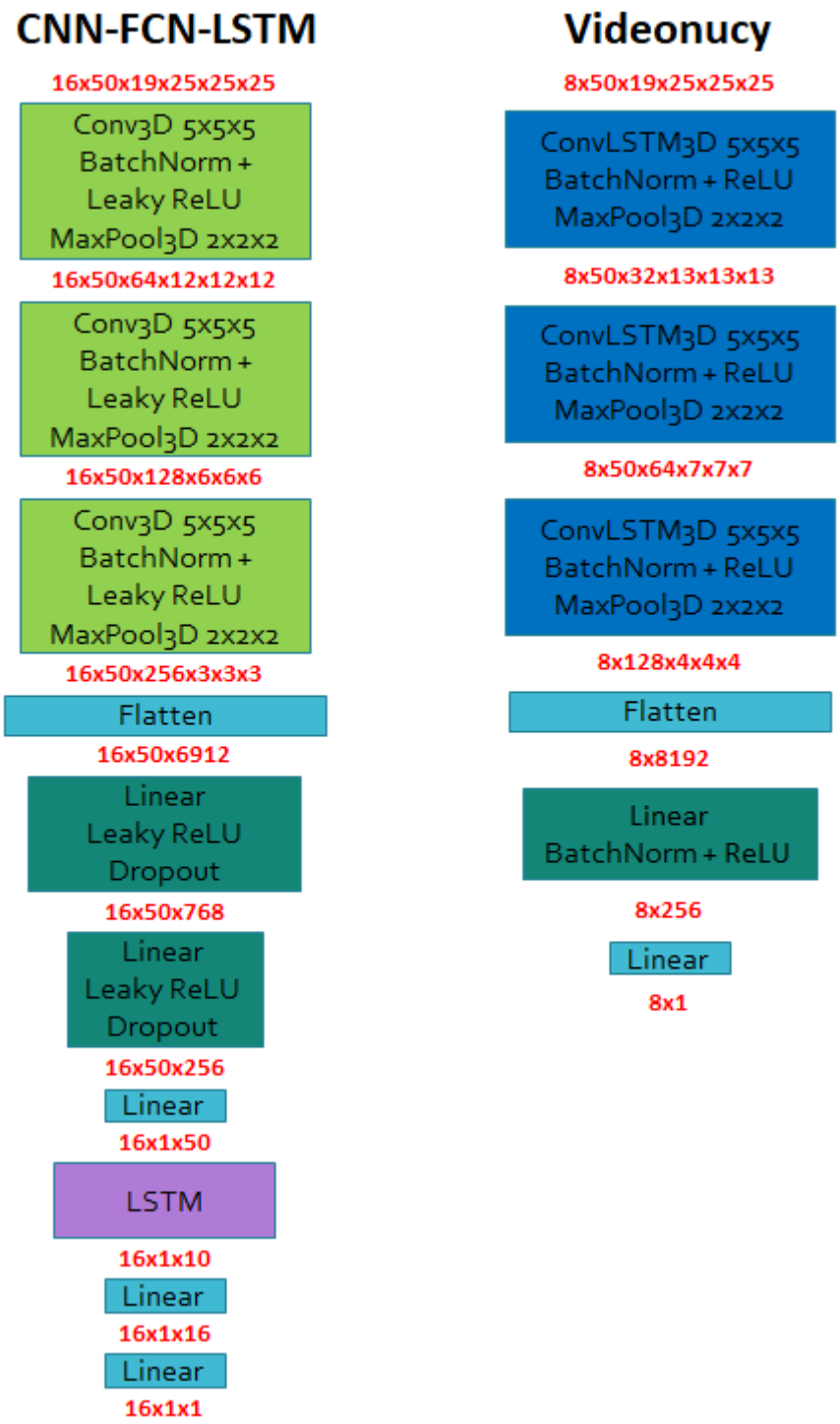



**Figure 2**. Detailed architecture of Timenucy (LRCN) and Videonucy (ConvLSTM). The convolutional layers are shown in light green, while the ConvLSTM layers are in dark blue. The LSTM is in purple, and the multi-layer perceptron (MLP) is in dark green. The flatten and output layers (also output of the CNN and LSTM) are in cyan.

Thereafter, a LRCN and a ConvLSTM were developed to train from the whole simulations (**Figure 2**), transitioning from using 3D to 4D input data. They use numpy files as inputs, which contain the information of each atom, including their features and positions, for each frame of the simulation. The atomic description and the voxelisation are identical to the ones implemented in Pafnucy.

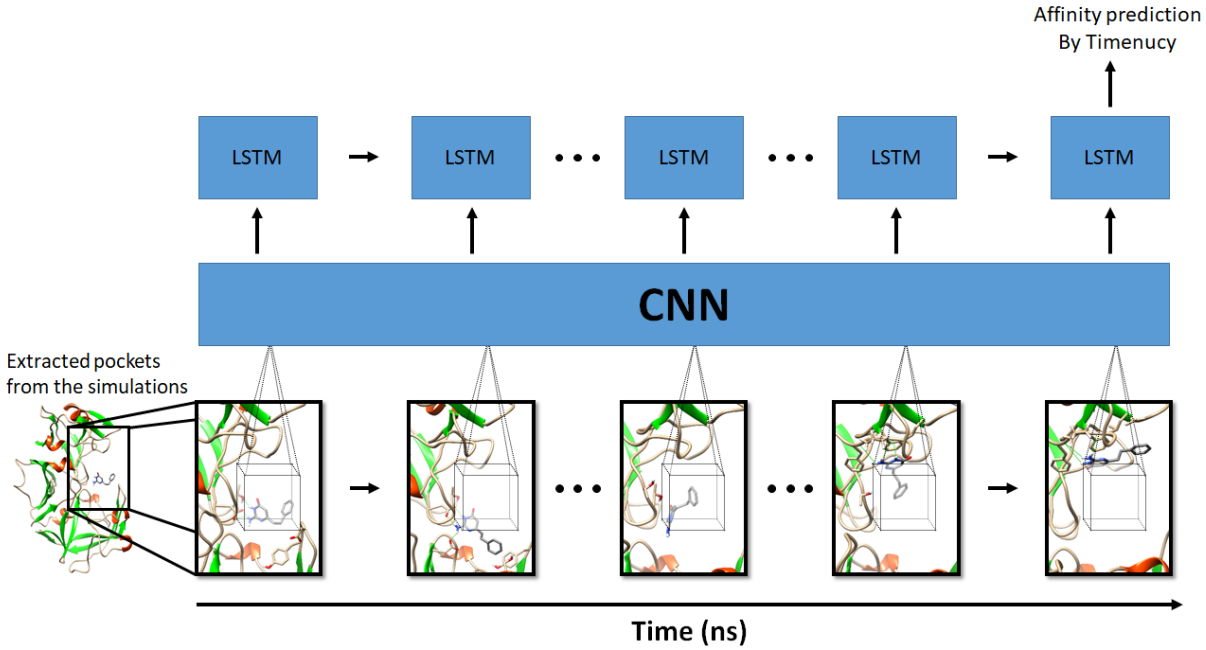



**Figure 3**. Workflow of the Timenucy (LRCN, long-term recurrent convolutional networks). "CNN" stands for convolutional neural network, while "LSTM" means long short-term memory.

Long-term recurrent convolutional networks (LRCN) were introduced for video activity recognition [78]. It is composed of a CNN followed by a LSTM. LRCN analyses videos frame by frame, examining each image individually, before carrying a final prediction. In a similar fashion, we created Timenucy, a LRCN capable of processing 4D data by using MD simulations as input (**Figure 3**). It works in two steps: first, the CNN performs convolutions on all the frames, and then the concatenated output is sent to the LSTM for the final analysis. Similarly to Pafnucy, the CNN component is composed of three 5×5×5 convolutional and 2×2×2 max pooling layers. The 3D convolutional layers consist of 64, 128 and 256 filters, and 3D batch normalizations are applied after each layer. These layers are followed by a fully connected network (FCN), also called an MLP. The FCN consists of two fully connected (FC) layers with 768 and 256 neurons respectively. The LSTM is composed of a single layer, succeeded by an FC layer of 16 neurons before the output layer.

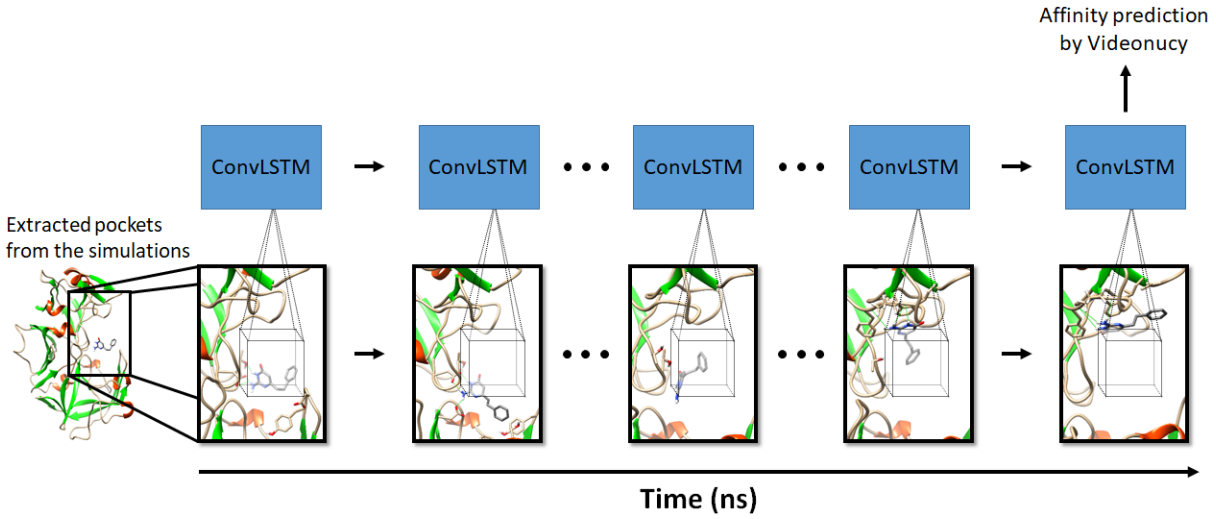



**Figure 4**. Workflow of Videonucy (ConvLSTM, convolutional long short-term memory).

The convolutional LSTM, called Videonucy (**Figure 4**), combines the convolution process with the LSTM mechanism (**Figure 5**) [79]. The original implementation of the convolutional LSTM comes from Shi *et al*. [80], and we adapted the code from https://github.com/ndrplz/ConvLSTM_pytorch. It is composed of three 5×5×5 ConvLSTM and 2×2×2 max pooling layers, followed by an FC layer. There are 32, 64 and 128 filters for the ConvLSTM layers, and the FC layer is composed of 256 neurons. A batch normalization was used after the FC layer.

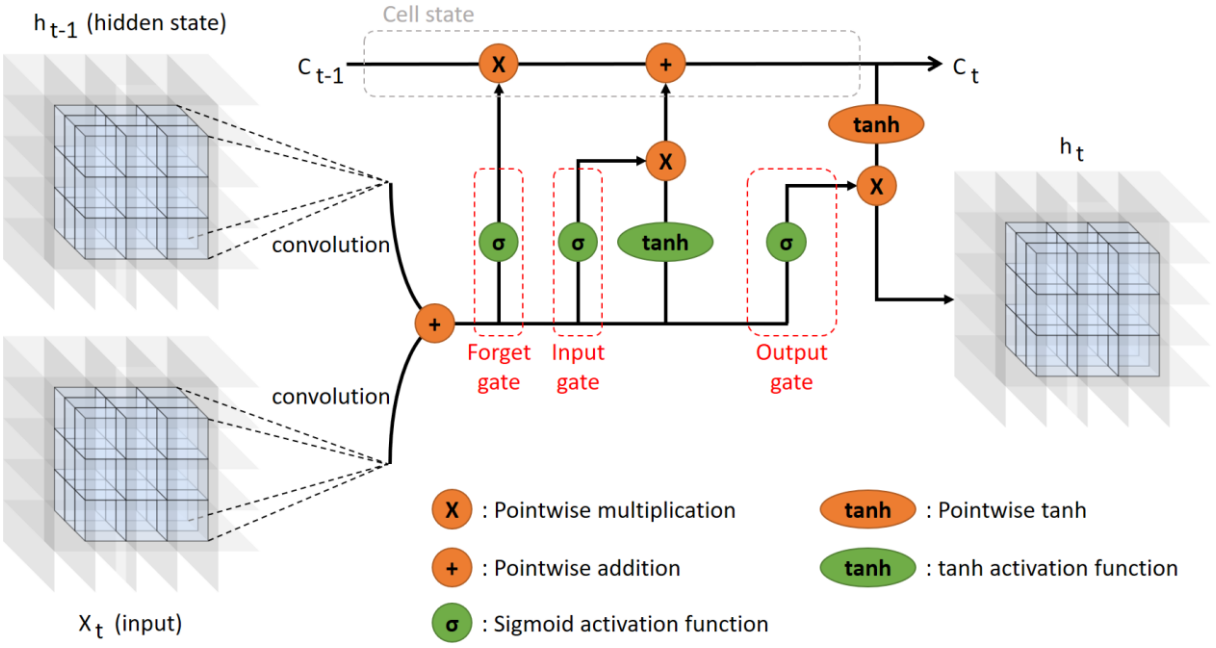



**Figure 5**. The intrinsic mechanism of the ConvLSTM applied on 4D data.

The dropout was implemented for regulation purposes, allowing models to generalize better. It was set to 0.5 for Proli's MLP (three FC layers). For Timenucy, a dropout of 0.5 was applied on the layers of the FCN, while it was set to 0.2 for the LSTM layer.

The Adam optimizer was used on all neural networks with a weight decay of $10^{-4}$. The optimizer was applied with a learning rate of $10^{-4}$ for Timenucy, while the others used a learning rate of $10^{-5}$. A learning rate scheduler was also added to Timenucy with 10 epochs of warmup. It was implemented to help models escape from local minima during training.

Aside from Timenucy, the rectified linear unit (ReLU) activation function was used everywhere. In the case of Timenucy, leaky ReLU were used with the CNN and the FCN. The leaky ReLU is a modified version of the ReLU, which helps mitigate the vanishing gradient problem. A batch size of 20 was used with Pafnucy and Densenucy, while it was set to 16 and 8 for Timenucy and Videonucy respectively. The codes of Densenucy, Timenucy and Videonucy were adapted to run with data parallelization. Therefore, several GPUs can be used to process different batches in parallel to speed up the training.

**METRICS**

The performance of models was evaluated on test sets. We compared the agreement between predicted and experimentally measured binding affinity values. Both the Pearson's correlation coefficient (R) and root mean square error (RMSE) were used to assess the scoring power of the model.

$$R = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2 \sum_{i=1}^{N}(y_i - \bar{y})^2}}$$

Where "$x_i$" represents the computed binding score for the ith complex, "$y_i$" stands for the corresponding experimental binding constant of that complex, and "*N*" is the total number of samples.

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathrm{x_i} - \hat{\mathrm{x}}_\mathrm{i})^2}{\mathrm{N}}}$$

Where "$x_i$" represents the real affinity value for the ith complex, "$\hat{x}_i$" is the predicted affinity value for that complex and "*N*" is the total sample size.

The ranking power was assessed with Spearman's correlation coefficient (ρ).

$$\rho = \frac{\frac{1}{N}\sum_{\mathrm{i}=1}^{\mathrm{N}}(rx_i - \overline{rx})\,(ry_i - \overline{ry})}{\sqrt{(\frac{1}{N}\sum_{\mathrm{i}=1}^{\mathrm{N}}(rx_i - \overline{rx})^2)\,(\frac{1}{N}\sum_{\mathrm{i}=1}^{\mathrm{N}}(ry_i - \overline{ry})^2)}}$$

Where "$rx_i$" stand for the rank of the binding score of the ith complex, and "$ry_i$" is the rank of the experimental binding constant of that complex. "$\overline{rx}$" and "$\overline{ry}$" are the mean ranks and "*N*" is the total sample size.

**RESULTS & DISCUSSION**

Firstly, Pafnucy [4], a well-known CNN for binding affinity prediction, was re-implemented in PyTorch. This version of the neural network is hereafter referred to as Proli. We evaluated its performance to ensure it matches the performance achieved in the original release. Then, we implemented Densenucy, an improved version of Pafnucy and Proli obtained by replacing the convolutional layers with dense blocks. We compared their performance by training with random and systematic rotations of PDBbind complexes (**Figure 6**). Models were trained and applied on the pockets that were calculated beforehand. These pockets were obtained using the same methodology employed for calculating the pockets from simulation frames. Consequently, the pockets provided by PDBbind were not utilized. The three neural networks, Pafnucy, Proli, and Densenucy, achieved comparable performance. Moreover, we observe no gain in performance by training with rotational data augmentation. In further training, we selected a random rotation for each complex, thus reducing the amount of computational load required to train models.

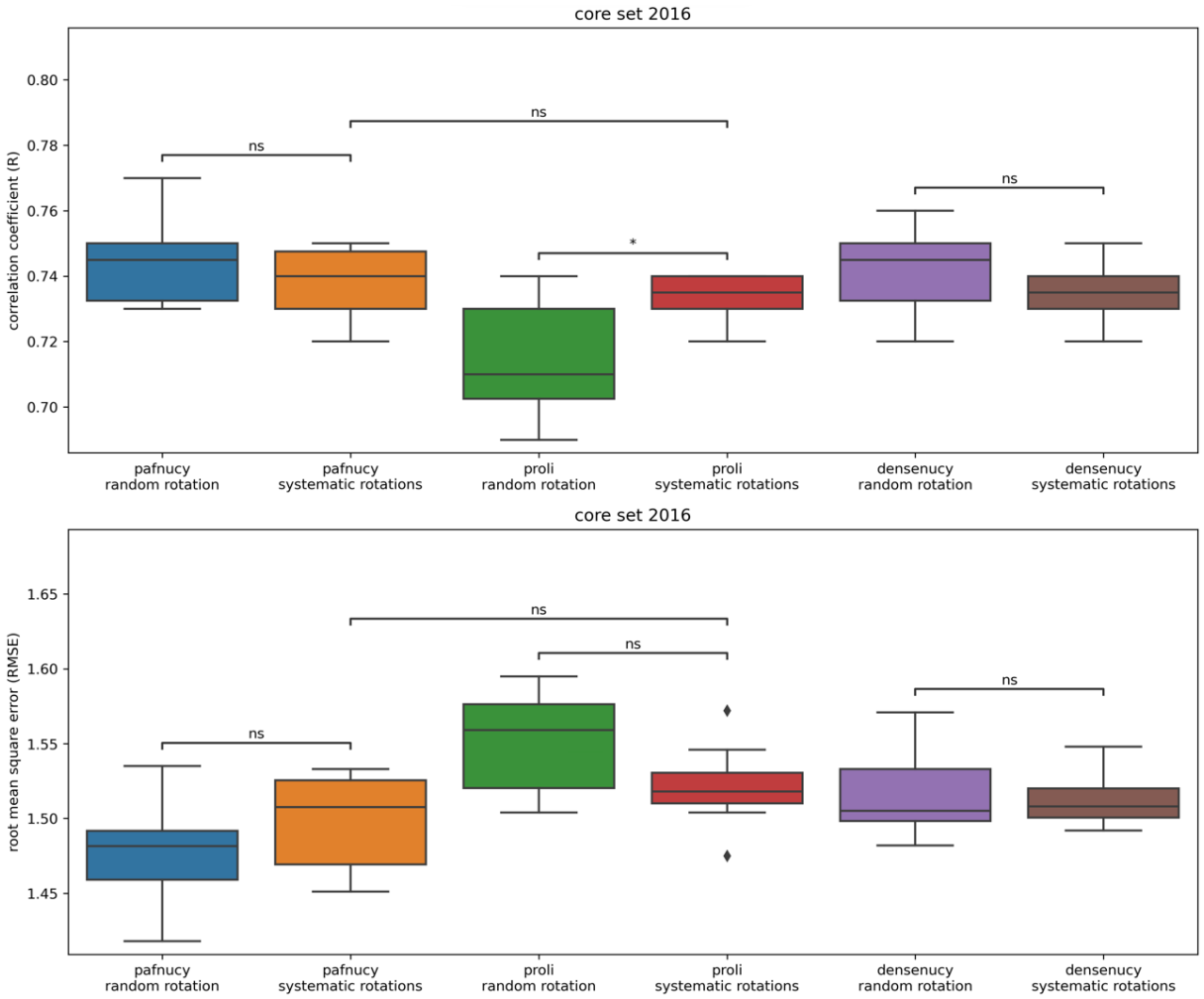



**Figure 6**. The performance of the models trained with Pafnucy, Proli (Pafnucy re-implementation in PyTorch) and Densenucy, using random or systematic rotations. Results are evaluated on the PDBbind v.2016 core set. For each training setting, the boxplot displays the results over 10 replicate models. The following p-values correspond to the annotations on the plots: ns: $5.00 \times 10^{-2} < p \leq 1.00 \times 10^{0}$, *: $1.00 \times 10^{-2} < p \leq 5.00 \times 10^{-2}$, and ♦ are possible outliers.

To improve further the performance, we investigated the use of other data augmentation methods. We focused on using additional information from the MD simulations such as the flexibility of the proteins, the stability of the ligands and the protein-ligand interactions across time. Indeed, it is possible to increase the current dataset size by extracting information of frames from MD simulations. In this case, the frames are considered as new structures, and each labelled with the binding affinity of the original complex. In this context, we trained Proli and Densenucy on a dataset of 3,000,000 structures instead of 18,000. We evaluated the gain in performance obtained by training the models with such data augmentation (**Figure 7**). We were unable to use rotational data augmentation in combination with the MD data augmentation due to the huge computational cost. When using MD data augmentation, there were no significant improvements in the case of the Proli models. In fact, MD data augmentation led to a significant drop in performance for Pafnucy (**Figure S1** in the supporting information). Although, this result might be due to technical issues like not training Pafnucy for long enough. Nonetheless, by using MD data augmentation during the training of Densenucy, we achieved

highest performance: R = 0.83 and RMSE = 1.28. This suggests an incentive for implementing MD data augmentation, based on its potential to enhance the performance of certain models.

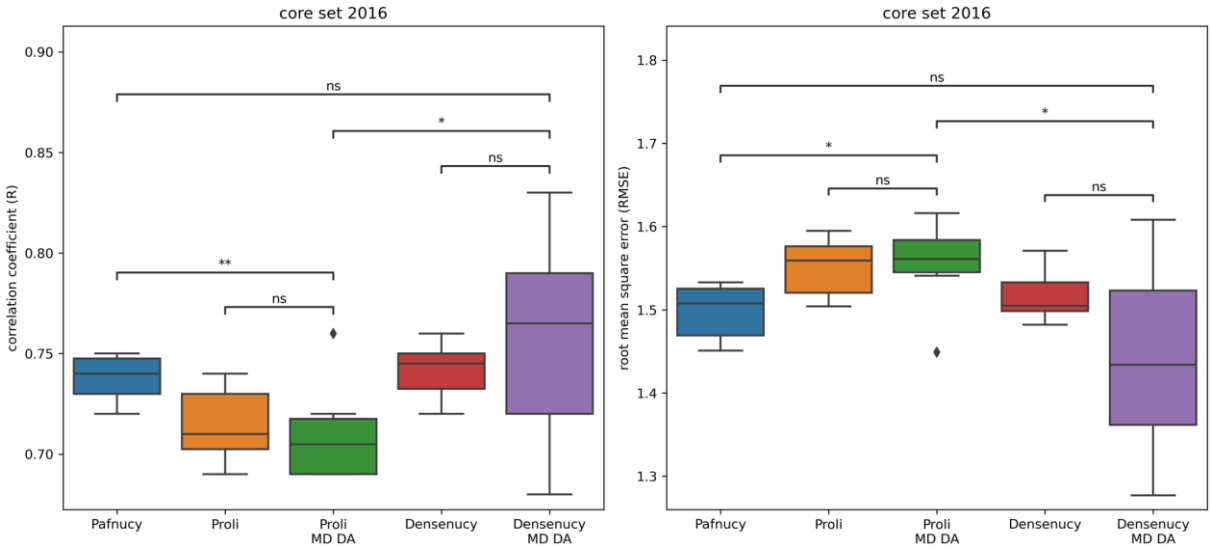



**Figure 7**. Comparison of the performance of Proli and Densenucy with or without data augmentation. Pafnucy performance without data augmentation on the same pockets was added for benchmark purpose. Results are evaluated on the PDBbind v.2016 core set. For each training setting, the boxplot displays the results over 10 replicate models. "MD DA" refers to molecular dynamics data augmentation. The following p-values correspond to the annotations on the plots: ns: $5.00 \times 10^{-2} < p \leq 1.00 \times 10^{0}$, *: $1.00 \times 10^{-2} < p \leq 5.00 \times 10^{-2}$, **: $1.00 \times 10^{-3} < p \leq 1.00 \times 10^{-2}$, and ♦ are possible outliers.

We also conducted a detailed analysis of the impact of MD data augmentation by examining the performance within each cluster of the PDBbind v.2016 core set. To achieve this, we computed the difference of absolute difference (ΔΔ) predictions of 2 methods such as Proli and Densenucy, with and without MD data augmentation (see the **Figure S2** in supporting information). To calculate ΔΔ predictions, we initially employed a consensus method to average predictions from 10 replicate models. Then, we determined the absolute difference between these averaged predictions and the experimental pKi. Subsequently, we calculated the difference in absolute Δ affinity values between predictions obtained for models trained with and without MD data augmentation.

Our findings suggest that MD data augmentation does not yield significant improvement with Proli model. However, Densenucy presents an improved performance, since it enhanced the predictions in 30% of the protein clusters for at least 4 out of 5 complexes. In 58% of the protein clusters, MD data augmentation had no discernible impact, while in 12% of the cases, it led to worsened predictions for at least 4 out of 5 complexes.

The improved predictions for Densenucy in comparison to Proli are most likely due to the dense architecture. It helps preserve the input data, therefore lowering the loss of information, which is very detrimental in case of sparse data. Additionally, the dense blocks in Densenucy are performing convolutions with 3×3×3 filters, which limit the number of parameters in comparison to 5×5×5 filters [81]. Moreover, to ensure that a 5×5×5 filter captures an equivalent amount of information as a dense block, it is necessary to have the same number of channels as the dense block, in addition to the input

channels. Hence, with an even number of channels, it is possible to incorporate additional layers of convolution within dense blocks, enabling the extraction of more information. Subsequently, the information is filtered through transition layers to retain only the most pertinent information. As a result, Densenucy is more adapted to MD data augmentation than Proli and provide better performance.

Not only can predictions be carried out on crystal structures, but also on the frames extracted from simulations. Therefore, we have decided to perform simulations on complexes of the PDBbind v.2016 core set, leading to the MDbind test set. This dataset is composed of 41,500 frames extracted from 830 simulations. Each frame was assigned the affinity of the initial complex. To evaluate the models, it is possible to predict the affinity for all the frames and compute the metrics (coefficient correlation and RMSE). Another approach is to average the prediction for all the frames of the same simulation, doing so we see improved performance using Pafnucy, Proli and Densenucy (see the **Figure S3** in supporting information). We can propose that there is meaningful information obtained by averaging the predicted values for each frame of a simulation, allowing to better predict the binding affinities of protein-ligand complexes.

Then, we compared the performance of MD data augmentation models evaluated on crystal structures and frames extracted from simulations (**Figure 8**). The models were benchmarked on 83 complexes from the PDBbind v.2016 core set, and the MDbind test set (41,500 frames). Furthermore, the predicted values obtained from frames were average for each simulation. We can see significant improvements by predicting on frames and averaging the predictions in comparison to predicting on the crystal structures. As reported previously [82], it seems that models are performing better when applied on molecular objects identical to the one used during training. Nonetheless, there seem to be an interest in averaging the predictions for each simulation. It might be worth investigating such methodology in the future.

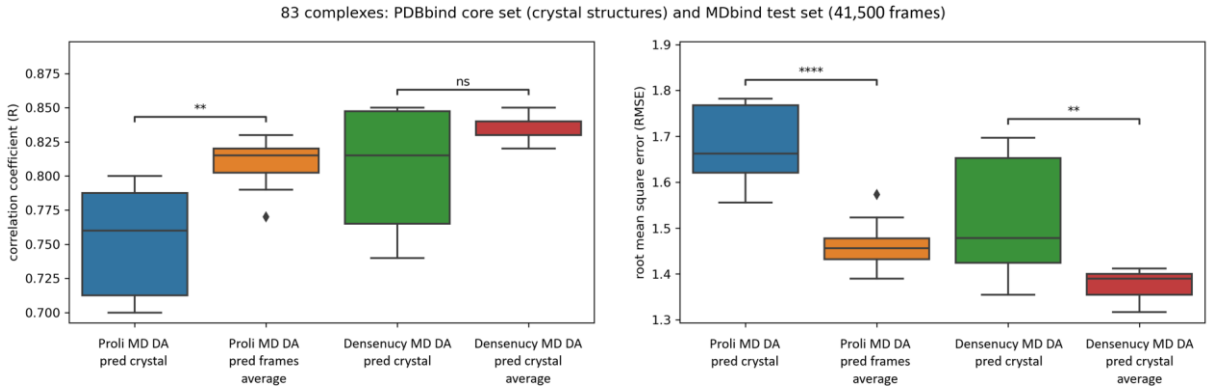



**Figure 8**. Comparison of the performance of predicting on crystal structures and frames extracted from simulations, for Proli and Densenucy. Predictions on frames were average for each simulation. Predictions were carried on 83 complexes from the PDBbind v.2016 core set. When evaluated on the MDbind test set, models were applied on 41,500 frames. The following p-values correspond to the annotations on the plots: ns: $5.00 \times 10^{-2} < p \leq 1.00 \times 10^{0}$, **: $1.00 \times 10^{-3} < p \leq 1.00 \times 10^{-2}$, ****: $p \leq 1.00 \times 10^{-4}$, and ♦ are possible outliers.

As shown previously, it is possible to carry out data augmentation by using the frames extracted from MD simulations. The MD data augmentation increases the amount of training data to reach quantities that are more in line with other fields where DL is applied successfully. Moreover, such data augmentation allows the neural networks to learn on more protein and ligand conformations. This strategy has some similarities to other data augmentation methods based on docking. For instance, PDBbind ligands have been redocked and several highly scored poses were selected for data augmentation purpose [7,12,83,84]. To have a better confidence in the selected docking poses, further criteria were implemented to increase the similarity to the experimental pose. For example, the poses were selected based on a maximal distance threshold, usually an RMSD of 2.0 Å, from the experimental pose. In other cases, docking poses exhibiting a similar interaction fingerprint to the experimental pose were selected [82]. Nonetheless, MD data augmentation also adds information about the flexibility of the proteins, the different binding modes of the ligands, and captures physically relevant differences between true and false poses and is the basis for accurate binding free energy calculations.

However, to analyse such data accurately, it is crucial to take into account its dynamic characteristics. Indeed, there are limitations in using the extracted frames independently. When analysing a specific frame, the models do not have access anymore to the temporal information, which is provided by the successive frames along the simulation. The removal of the temporal link between the frames results in the loss of information regarding the dynamic behaviour of the protein-ligand complexes. For example, information like the creation and removal of protein-ligand interaction across time are not taken into account anymore.

We developed two neural networks, a LRCN and a ConvLSTM, both able to analyse the MD simulation data in a time dependent manner. They take as input a series of frames extracted from a simulation, which are then sequentially analysed. The prediction is made once the last frame has been analysed. Therefore, it is required to perform an MD simulation before predicting the binding affinity of a complex. As mentioned in the Methods section, the MD simulations were performed on 83 complexes of the PDBbind v.2016 core set. We compared the results of Pafnucy and spatio-temporal models (**Figure 9**). Pafnucy and spatio-temporal models were trained, validated and tested on the same complexes. This resulted in a training set comprising 4,753 complexes and a validation set encompassing 1,179 complexes. To obtain a single prediction for each complex when employing spatio-temporal learning methods, we averaged predictions across all simulations of that complex. Likewise, when assessing Pafnucy's performance on frames, predictions for frames associated with the same complex were averaged. While Timenucy and Videonucy exhibit significantly lower performance compared to Pafnucy, the RMSE of Pafnucy drops drastically when it is evaluated on the frames instead of the crystallographic poses. In such cases, Videonucy models achieve a better RMSE compared to Pafnucy.

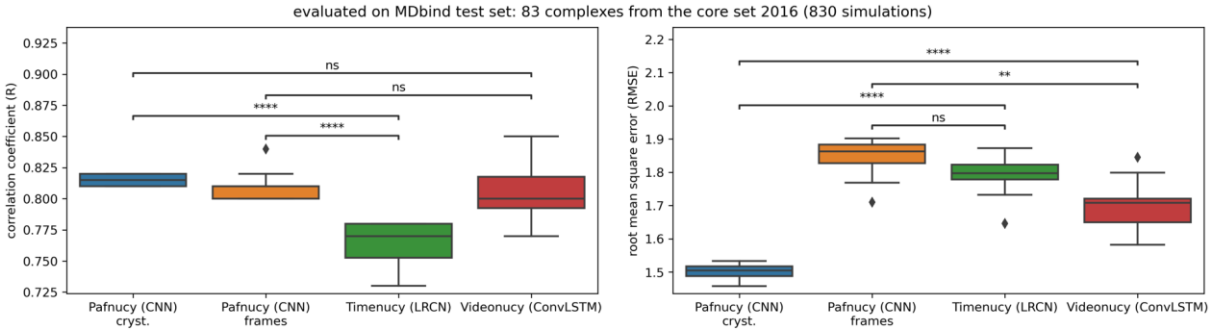



**Figure 9.** Comparison of the performance of CNN (Pafnucy) and the spatio-temporal models LRCN (Timenucy) and ConvLSTM (Videonucy). The models were trained on 4,753 complexes, validated on 1,179 complexes and evaluated on the MDbind test set composed of 83 complexes from the PDBbind v.2016 core set. Pafnucy was applied on two test sets, the crystallographic poses (Pafnucy (CNN) Cryst.) or the frames (Pafnucy (CNN) frames) extracted from the simulations. Timenucy and Videonucy were trained and evaluated on the simulations. Given that each complex was simulated using 10 replicates, both the LRCN and ConvLSTM were evaluated on a total of 830 simulations. For each training setting, the boxplot displays the results over 10 replicate models. The following p-values correspond to the annotations on the plots: ns: $5.00 \times 10^{-2} < p \leq 1.00 \times 10^{0}$, **: $1.00 \times 10^{-3} < p \leq 1.00 \times 10^{-2}$, ****: $p \leq 1.00 \times 10^{-4}$, and ♦ are possible outliers.

Several studies have brought up the issue that most DL models trained on the PDBbind are biased [13,16]. Allegedly, models trained using PDBbind dataset are learning biases in the data instead of using the protein-ligand interactions to predict the binding affinity of a complex. In light of these concerns, we implemented a method to evaluate the ability of the models to learn on the interactions. Model performance was evaluated by training the models by removing information of the protein or the ligand. We calculate the gap in prediction between learning on the whole complex or only on one of the two molecular partners. Small gap in predictions would indicate that the model barely uses the information contained in the whole complex. In this case, the model would be most likely biased and a quantitative structure-activity relationships (QSAR) model on one of the components alone is likely sufficient.

We investigated whether our models could learn from the interactions of the complex or from the protein or ligand only. Interestingly, Proli and Densenucy have better performance by learning on both the pockets and the ligands together than by training on only one partner (**Figure 10**). The difference in performance between training on the complex or solely on proteins or ligands is evaluated by comparing the mean performance across 10 model replicates for each training condition. This led to a gap in performance for "only protein" of $\Delta R_{prot} = 0.11$ and $\Delta R_{prot} = 0.27$ for Proli and Densenucy, respectively. The gaps in performance in using the whole complex and the ligand only are $\Delta R_{lig} = 0.12$ and $\Delta R_{lig} = 0.08$ for Proli and Densenucy, respectively. For reference, the performance gaps of Pafnucy without MD data augmentation are $\Delta R_{prot} = 0.11$ and $\Delta R_{lig} = 0.13$ [19].

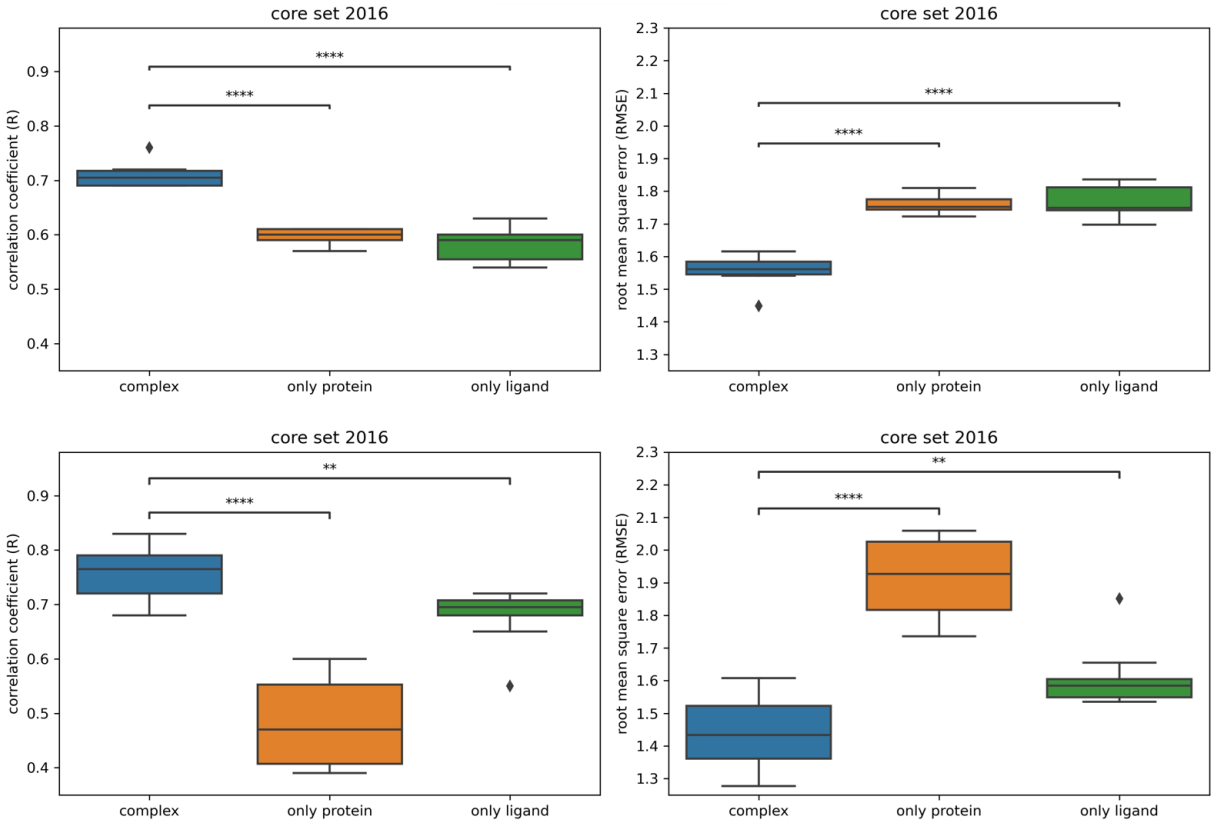



**Figure 10**. Comparison of the performance of Proli (top) and Densenucy (bottom) using MD data augmentation (MD DA) when trained without proteins or ligands. Performance was assessed on the PDBbind v.2016 core set, while removing the protein ("only ligand") or the ligand ("only protein") information during the training of the models. The following p-values correspond to the annotations on the plots: **: $1.00 \times 10^{-3} < p \leq 1.00 \times 10^{-2}$, ****: $p \leq 1.00 \times 10^{-4}$, and ♦ are possible outliers.

Timenucy and Videonucy exhibit significantly better performance when trained on complexes compared to training on ligands or proteins alone (**Figure 11**). The gaps in performance in using the whole complex and the protein only are $\Delta R_{prot} = 0.13$ and $\Delta R_{prot} = 0.19$ for Timenucy and Videonucy, respectively.

When evaluating the performance gap between the complex and the ligand only using spatio-temporal methods, we also conducted this assessment using two different settings, each utilizing a different center for the input box. Initially, the pockets extracted from the simulation frames are composed of predefined amino acids. The neural networks take as input boxes centred on these pockets. Therefore, during some simulations, ligands are moving inside the boxes and might even exit in some cases. When training only on the ligands, we kept the same approach with the boxes centred on the pockets. We also evaluated the performance of the models trained with only the ligands ("only ligand") and the boxes were centred on the ligand, named "only ligand (tracking)". The performance gap between models trained on the complex and only the ligand with boxes centred on the pockets is denoted as $\Delta R_{lig}$, while the one with boxes centred on the ligands is referred to as $\Delta R_{lig\_tracking}$.

The performance gaps for "only ligand" are $\Delta R_{lig} = 0.06$ and $\Delta R_{lig} = 0.07$ for Timenucy and Videonucy respectively. Surprisingly, the spatio-temporal models trained solely on ligands show performance levels remarkably close to those trained on the complexes. Nonetheless, we observe that the performance of spatio-temporal models trained without proteins and with boxes centred on the

ligands (tracking mode), are significantly lower than with boxes centred on the pockets ($\Delta R_{lig\_tracking}$ = 0.13 and $\Delta R_{lig\_tracking}$ = 0.11 for Timenucy and Videonucy respectively). An interpretation could be that valuable information is derived from the ligand stability within the binding site, which can improve binding affinity predictions. This suggests that the spatio-temporal models gain insights from the movements of the ligands throughout the boxes, and they acquire an understanding of ligand dynamic behaviour. This applies whether or not there is direct observation of the protein-ligand interactions, and it could be an explanation for the small gap in performance with "only ligand" for spatio-temporal models.

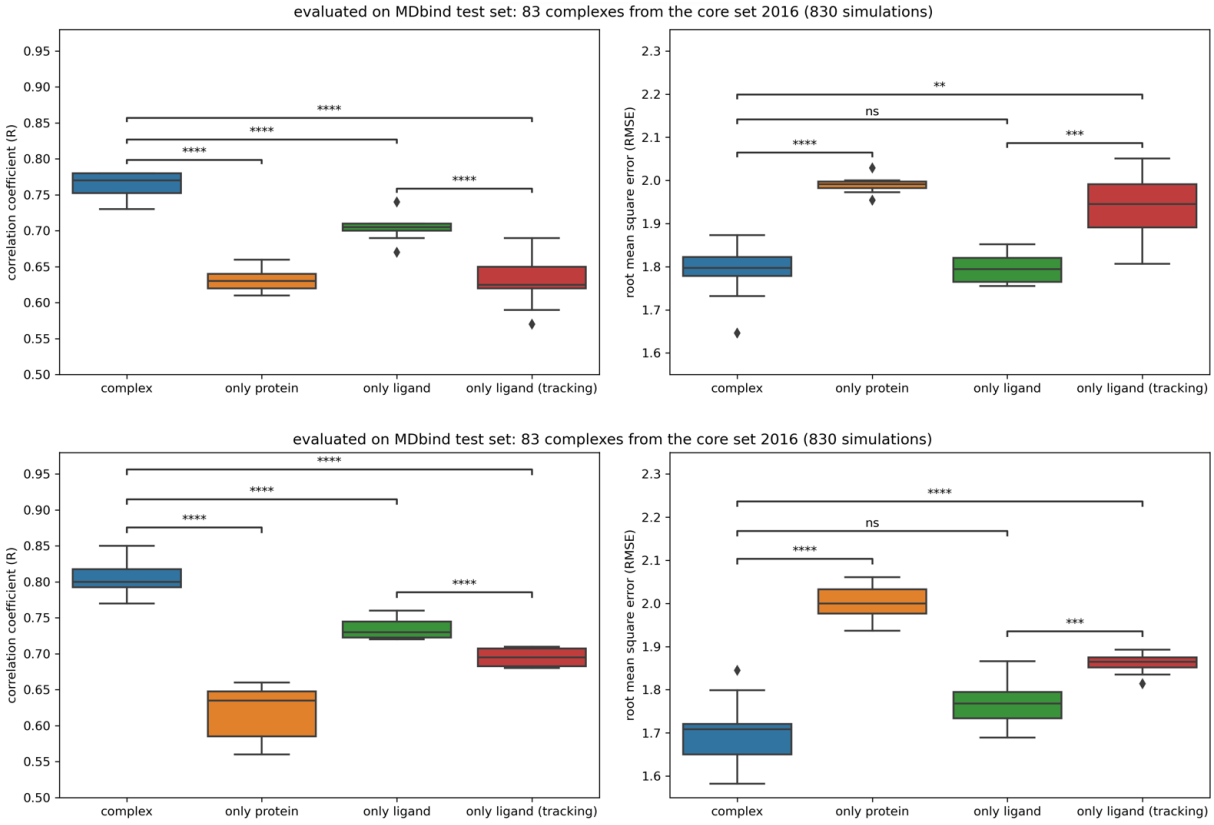



**Figure 11**. Comparison of the performance of spatio-temporal learning methods, Timenucy (top) and Videonucy (bottom) when trained without protein or ligand information. Performance was assessed on the MDbind test set, composed of 830 simulations from 83 complexes from the PDBbind v.2016 core set. The protein or the ligand were removed from the frames according to the training method. Models with "complex", "only protein" and "only ligand" are trained with boxes centered on pockets. The "only ligand (tracking)" models use boxes centered on the ligands. The following p-values correspond to the annotations on the plots: ns: $5.00 \times 10^{-2} < p \leq 1.00 \times 10^{0}$, **: $1.00 \times 10^{-3} < p \leq 1.00 \times 10^{-2}$, ***: $1.00 \times 10^{-4} < p \leq 1.00 \times 10^{-3}$, ****: $p \leq 1.00 \times 10^{-4}$, and ♦ are possible outliers.

Overall, the $\Delta R_{prot}$ and $\Delta R_{lig\_tracking}$ of the spatio-temporal learning models reach similar values in comparison to Pafnucy prediction gaps ($\Delta R_{prot}$ = 0.11 and $\Delta R_{lig}$ = 0.13). Interestingly, we obtain significantly lower ΔRlig in comparison to $\Delta R_{lig\_tracking}$ for Timenucy and Videonucy. These results might indicate that our models learned from the movement of the ligand in the binding site during the MD simulations.

The performance (R, RMSE and ρ) of each model is summarized in **Table 1**. A consensus method over 10 model replicates was used to display the best performances for all models.

**Table 1**. Performance of Pafnucy, Proli and Densenucy models is assessed on the whole CASF-2016 dataset (285 complexes) with or without MD data augmentation (MD DA) and performance of Pafnucy reduced, Timenucy and Videonucy on the MDbind test set, composed of 83 complexes from CASF-2016. Performance metrics are presented using a consensus method applied on the 10 replicate models for each CASF cluster. The Spearman's correlation coefficients were computed on all complexes and on each protein cluster and then averaged, resulting in the "ρ all" and "ρ cluster" metrics, respectively. 57 protein clusters were used for Pafnucy, Proli and Densenucy, and 5 for Pafnucy reduced, Timenucy and Videonucy. The best results are displayed in bold.

| Models | R | RMSE | ρ all | ρ cluster |
|---|---|---|---|---|
| Pafnucy | 0.76 | 1.45 | 0.75 | 0.59 |
| Pafnucy MD DA | 0.70 | 1.62 | 0.68 | 0.56 |
| Proli | 0.74 | 1.52 | 0.74 | 0.62 |
| Proli MD DA | 0.74 | 1.51 | 0.74 | 0.61 |
| Densenucy | 0.77 | 1.48 | 0.76 | 0.62 |
| Densenucy MD DA | **0.81** | **1.36** | **0.81** | **0.67** |
| Pafnucy reduced | 0.83 | **1.47** | 0.82 | 0.74 |
| Timenucy | 0.78 | 1.78 | **0.83** | 0.66 |
| Videonucy | **0.84** | 1.66 | **0.83** | **0.76** |

We used another external dataset of protein-ligand complexes previously used to demonstrate FEP performance [75] to further benchmark our models. The FEP dataset comprises similar ligands with different binding affinities for the same proteins. It can be used to determine the ability of the models to predict the binding affinities of ligands having small structural modifications and therefore if the models can predict activity cliffs. Therefore, this dataset is useful for evaluating potential applications of such models in lead optimization projects. 10 ligands from the FEP dataset (5%) were found in PDBbind. All proteins from the FEP dataset were found in PDBbind. For most of them there are only a few examples in the PDBbind, although we found 80 occurrences for CDK2, 182 for thrombin and 330 for BACE. Proli and Densenucy were evaluated with MD data augmentation on the FEP dataset (**Table 2**), and they were compared to RF-score [85], $K_{DEEP}$ [5] and Pafnucy.

**Table 2**. Comparison of the performance of RF-score, $K_{DEEP}$, Pafnucy, Proli MD DA and Densenucy MD DA for each protein of the FEP dataset. The results of RF-score and $K_{DEEP}$ were published by Jiménez *et al* [5]. "MD DA" stands for MD data augmentation. The best results are displayed in bold.

| | RF-score | | $K_{DEEP}$ | | Pafnucy | | Pafnucy MD DA | | Proli MD DA | | Densenucy MD DA | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | R | RMSE | R | RMSE | R | RMSE | R | RMSE | R | RMSE | R | RMSE |
| Bace | -0.12 | **0.65** | -0.06 | 0.84 | **0.33** | 0.75 | -0.38 | 0.74 | -0.22 | 1.02 | -0.05 | 0.72 |
| CDK2 | -0.24 | 1.05 | 0.69 | 1.26 | 0.21 | 0.85 | 0.70 | 0.78 | **0.71** | 0.90 | 0.62 | **0.69** |
| Jnk1 | 0.61 | **0.50** | **0.69** | 1.18 | 0.34 | 0.66 | 0.68 | 0.55 | 0.65 | 0.62 | -0.66 | 0.88 |
| MCL1 | 0.51 | 0.99 | 0.34 | 1.04 | 0.46 | 1.00 | 0.29 | 0.89 | 0.47 | 0.98 | **0.59** | **0.71** |
| PTP1B | 0.26 | 0.90 | 0.58 | 0.93 | **0.81** | 0.97 | 0.66 | 0.84 | 0.68 | **0.74** | 0.76 | 0.85 |
| Thrombin | 0.08 | 0.71 | 0.58 | 0.44 | 0.67 | **0.39** | 0.67 | 0.75 | **0.64** | 1.06 | 0.50 | 0.95 |
| Tyk2 | 0.41 | 0.94 | -0.22 | 1.13 | -0.55 | 1.24 | 0.22 | 0.91 | 0.31 | **0.92** | **0.69** | 1.18 |
| p38 | 0.48 | 0.90 | 0.36 | 1.57 | 0.63 | 0.95 | 0.47 | 0.87 | 0.58 | 0.78 | **0.66** | **0.62** |
| average | 0.25 | 0.83 | 0.37 | 1.05 | 0.36 | 0.85 | 0.41 | **0.79** | **0.48** | 0.88 | 0.39 | 0.83 |
| Weighted average | 0.28 | 0.84 | 0.33 | 1.08 | 0.40 | 0.88 | 0.33 | 0.80 | **0.42** | 0.88 | 0.38 | **0.78** |
| Median | 0.34 | 0.9 | 0.47 | 1.09 | 0.40 | 0.90 | 0.56 | 0.81 | **0.61** | 0.91 | **0.61** | **0.79** |
| Whole dataset | NA | NA | NA | NA | 0.45 | 0.90 | 0.57 | 0.81 | 0.41 | 0.89 | **0.59** | **0.79** |

In general, there is no improvement on the performance by training Pafnucy with MD data augmentation (MD DA), while Densenucy MD DA tends to have better performance than Proli MD DA (**Figure S4** in the supporting information). However, when we delve into the details for each protein, we observe some noteworthy variations (**Figure S5** in the supporting information). For instance, Proli and Densenucy with MD data augmentation exhibit improved RMSE for p38, while they obtain a worse RMSE for Thrombin. They also demonstrate improved RMSE on CDK2 compared to $K_{DEEP}$. In comparison to the other methods, Densenucy achieves a favourable correlation coefficient on Tyk2 and a commendable RMSE on MCL1. On the contrary, for Jnk1, Densenucy exhibits a negative correlation, while Pafnucy correlation coefficient improves with MD data augmentation. Both Proli and $K_{DEEP}$ achieve strong correlation coefficients on Jnk1, but Proli outperforms $K_{DEEP}$ in terms of RMSE. Pafnucy is the only model to have a positive correlation coefficient on Bace.

These results suggest that the RMSE of most models except for $K_{DEEP}$, consistently remains below 1 for each protein family. This trend could be attributed to the fact that the pKi values of the FEP dataset predominantly fall between 5 and 8, which corresponds to common prediction range of neural networks. Consequently, these results that appear very good at first, might be only due to the FEP dataset not having enough extreme affinity values, which artificially improves the RMSE. Moreover, as

the affinity values are in a short range for each protein family, small variation in the prediction might importantly impact the correlation coefficient. Therefore, these results must be evaluated with care.

We note that Densenucy was able to achieve enhanced predictions for specific protein clusters, such as p38, where the performance improved from R=0.36 and RMSE=1.57 with $K_{DEEP}$ to R=0.66 and RMSE=0.62 with Densenucy MD data augmentation. Given that we achieved robust performance on a dataset comprising similar ligands targeting the same protein with varying affinities, it opens up the possibility of using such tools in lead optimization.

We envision that new kinds of neural networks will be developed in the future to properly exploit MD datasets. Within the context of using MD simulations as a means of data augmentation, performance improvement should be achievable by using multi-instance learning [86]. Multi-instance learning could be used to select the relevant frames from the simulation, since not all of them may be carrying useful information for the prediction of the binding affinities. Multi-instance learning can be trivially explained by grouping all the frames of a simulation in a bag, which is labelled with the binding affinity of the complex. Unlike with spatio-temporal learning, the frames are not connected to each other temporally. Then, the attention mechanism implemented in multi-instance learning selects the relevant frames that are subsequently used to predict the binding affinity. This method has recently been applied in the CADD field to predict conformers [87].

In the case of spatio-temporal learning, other neural networks could be implemented to improve further the binding affinity predictions. The CNN component of Timenucy could be replaced by a GNN that is better suited to analyse molecules, thus benefiting from its reduced computational cost. Further improvement may be achieved by using physics-informed neural networks such as PIGNet [16]. Another promising direction to explore would be the implementation of video transformers [88-90] applied to 4D data. Currently video transformers are able to perform activity detection by focusing on the relevant images of a video via their attention mechanism. Therefore, it would be worthwhile to develop a graph transformer, like Dynaformer [39], in a modified version to analyse series of frames like a video transformer. The trained models would process the data in a time-dependent manner while focusing on the relevant frames of a simulation to predict the binding affinity.

**CONCLUSION**

Despite the implementation of new neural networks, the models trained on the commonly used training and test sets are reaching the limits in their ability to predict the binding affinity of protein-ligand complexes. Indeed, several challenges arose in the field, including dataset biases and difficulties to train models able to predict binding affinity based on protein-ligand interactions. A major focus has been to augment current training datasets to improve the performance of DL models. Aside from the increased amount of data, the aim of the data augmentation is also to provide new insights to models to alleviate aforementioned challenges.

In that sense, several data augmentation methods were developed; they were based mainly on molecular docking protocols [7,12,82-84]. To provide dynamic information on protein-ligand complexes, we set up a new MD data augmentation protocol. First, we created MDbind, a dataset comprising 63,000 simulations obtained by performing 10 simulations of 10 ns on 6,300 PDBbind complexes. Then, we evaluated the performance of two CNN, Proli and Densenucy, on the PDBbind core set [2] and the FEP dataset [75]. MD data augmentation enables the development of high-

performance models with Densenucy by achieving a coefficient correlation of 0.83 and an RMSE of 1.28 on PDBbind v.2016 core set. Furthermore, by using a prediction consensus approach on 10 replicate models, it appears that Densenucy with data augmentation is outperforming Pafnucy in both regression and ranking tasks, not only on the PDBbind v.2016 core set, but also on the FEP data set. We note that carrying prediction on frames (MDbind test set) lead to higher performances for models trained with MD data augmentation, especially when averaging the performance over all the frames of each simulation. Additionally, we explored the advantages of implementing rotational data augmentation and demonstrated that employing random rotations delivers comparable performance with a lower computational cost.

Furthermore, we developed two neural networks that perform spatio-temporal learning on the whole MD simulations; a LRCN called Timenucy and a ConvLSTM referred to as Videonucy. Models were trained on MDbind and they achieved modest performance. Due to computation time constraints, the training is performed on a randomly selected simulation of each complex per epoch. In addition, we did not implement systematic rotations, as training with all the rotations did not seem to improve Proli or Densenucy performance while it would increase the computation time by 24-fold.

Overall, we developed two methodologies to train from MD simulations, namely the MD data augmentation and spatio-temporal learning, which can be applied in different stages of the drug design process. On the one hand, the MD data augmentation approach aims at improving the performance of neural networks initially developed to analyse static data. Models trained with this method can be easily applied in a virtual screening process to score docking poses. On the other hand, the spatio-temporal learning method can only be applied on complexes that have undergone MD simulations beforehand. Despite this drawback, predictions would be more accurate due to the temporal analysis of the MD simulations. Therefore, such methods would be preferably applied in a hit-to-lead or lead optimization phase, to select promising molecules among reduced number of ligands.

We foresee improvements in the field by combining MD simulations with DL. Both tools are complementary, as MD simulations provide additional dynamic information about protein-ligand interactions to the neural networks. We have noted some beneficial effects of the use of the MD simulation data for the DL affinity prediction models. The best model is Densenucy using MD data augmentation with improvements on the PDBbind v.2016 core set from R = 0.77 and RMSE = 1.48 to R = 0.81 and RMSE = 1.36. Similar improvements were observed for Densenucy when applied to entirely novel external test sets taken from the FEP dataset, with an increase from ρ all = 0.76 and ρ cluster = 0.62 to ρ all = 0.81 and ρ cluster = 0.67. Despite this, the results do not show major improvement compared to previous studies, despite a substantial computational cost. However, as mentioned above, we are still limited for the most part to learning from biases inside the PDBbind dataset. In addition, the use of the inappropriate PDBbind core set for external testing does not provide a difficult enough challenge for affinity prediction models. In brief, overtrained models return high performance statistics without knowing if there is really any underlying improvement due to their methodology. Our study is partly affected by the same concern but in our case the good performance of Densenucy on the FEP data is a good indication in this regard. We also note that Timenucy and Videonucy seem to learn from ligand behaviour in the pocket, as they get better performance by using boxes centred on the pockets rather than on the ligands when solely using the ligand. We will continue to investigate the impact of this large dataset for DL models in the future and strive to improve the ability of the model to better learn from protein-ligand interactions while taking into account the dynamic information available from MD simulations.

## ASSOCIATED CONTENT

### Supporting Information.

The following docx file is available free of charge.

Figure S1: Comparison of the performance of Pafnucy trained with or without MD data augmentation (MD DA).

Figure S2: ΔΔ Predictions with and without MD data augmentation for Proli and Densenucy on the PDBbind v.2016 core set.

Figure S3: Comparison of the performance of MD data augmentation models (Pafnucy, Proli and Densenucy) evaluated on MDbind test set (41,500 frames) with or without averaging the predicted values per simulation/complex.

Figure S4: Comparison of the performance of Pafnucy, Pafnucy MD DA, Proli MD DA and Densenucy MD DA on the FEP dataset.

Figure S5: Correlation plots of Pafnucy, Pafnucy MD DA, Proli MD DA and Densenucy MD DA for each protein of the FEP dataset.

## AUTHOR INFORMATION


### Corresponding Author

Pascal Bonnet - Institute of Organic and Analytical Chemistry (ICOA); UMR7311, Université d'Orléans, CNRS; Pôle de chimie rue de Chartres - 45067 Orléans Cedex 2, France; https://orcid.org/0000-0001-6485-138X; Email: pascal.bonnet@univ-orleans.fr

Gary Tresadern - Computational Chemistry, Janssen Research & Development; Janssen Pharmaceutica N. V.; B-2340 Beerse, Belgium; https://orcid.org/0000-0002-4801-1644; Email: gtresade@its.jnj.com


### Author Contributions

The manuscript was written through contributions of all authors. All authors have given approval to the final version of the manuscript.


### Funding Sources

This research was funded by JANSSEN. Grant number 262402.


### Notes

The authors declare no competing financial interest.


## ACKNOWLEDGMENT

This work was granted access to the HPC resources of CRIANN (grant 2021002, 10,000 hours.gpu allocated) and IDRIS under the allocation *2021-A0100712496* (200,000 hours.gpu allocated) & 2022-*AD011013521* (50,000 hours.gpu allocated) made by GENCI. Authors gratefully acknowledge major

financial support from Janssen which made this study possible. P.Y.L, S.A.S and P.B. are thankful to the projects CHemBio (FEDER-FSE 2014-2020-EX003677), Valbiocosm (FEDER-FSE 2014-2020-EX003202), Techsab (FEDER-FSE 2014-2020-EX011313), QUALICHIM (APR-IA-PF 2021-00149467), the RTR Motivhealth (2019-00131403) and the Labex programs SYNORG (ANR-11-LABX-0029) and IRON (ANR-11-LABX-0018-01) for their financial support of ICOA, UMR 7311, University of Orléans, CNRS.

**ABBREVIATIONS**

ABFE, absolute binding free energy; AM1-BCC, Austin model 1 – bond charge correction; CASF, comparative assessment of scoring functions; CNN, convolutional neural network; DA, data augmentation; DL, deep learning; FC, fully connected; FEP, free energy perturbation; gaff, general amber force field; GNN, graph neural network; HDF5, hierarchical data format version 5; LIE, linear interaction energy; LSTM; long short-term memory; QSAR, quantitative structure-activity relationship; MD, molecular dynamics; MMPB(GB)SA, molecular mechanics Poisson-Boltzmann (generalized born) surface area; PDB, protein data bank; PMEMD, particle mesh Ewald MD; RAM, random-access memory; RBFE, relative binding free energy; t-MD, targeted MD; tip3p, transferable intermolecular potential with 3 points

**REFERENCES**

1. Li, J.; Fu, A.; Zhang, L., An Overview of Scoring Functions Used for Protein–Ligand Interactions in Molecular Docking. Interdisciplinary Sciences: Computational Life Sciences 2019, 11, 320-328.

2. Su, M.; Yang, Q.; Du, Y.; Feng, G.; Liu, Z.; Li, Y.; Wang, R., Comparative Assessment of Scoring Functions: The CASF-2016 Update. Journal of chemical information and modeling 2019, 59, 895-913.

3. Meli, R.; Morris, G.; Biggin, P., Scoring functions for protein-ligand binding affinity prediction using structure-based deep learning: a review. Frontiers in Bioinformatics 2022, 2.

4. Stepniewska-Dziubinska, M. M.; Zielenkiewicz, P.; Siedlecki, P., Development and evaluation of a deep learning model for protein-ligand binding affinity prediction. Bioinformatics (Oxford, England) 2018, 34, 3666-3674.

5. Jiménez, J.; Škalič, M.; Martínez-Rosell, G.; De Fabritiis, G., KDEEP: Protein–Ligand Absolute Binding Affinity Prediction via 3D-Convolutional Neural Networks. Journal of chemical information and modeling 2018, 58, 287-296.

6. Liu, Q.; Wang, P.-S.; Zhu, C.; Gaines, B. B.; Zhu, T.; Bi, J.; Song, M., OctSurf: Efficient hierarchical voxel-based molecular surface representation for protein-ligand affinity prediction. Journal of Molecular Graphics and Modelling 2021, 105, 107865.

7. Francoeur, P. G.; Masuda, T.; Sunseri, J.; Jia, A.; Iovanisci, R. B.; Snyder, I.; Koes, D. R., Three-Dimensional Convolutional Neural Networks and a Cross-Docked Data Set for Structure-Based Drug Design. Journal of chemical information and modeling 2020, 60, 4200-4215.

8. Wu, Z.; Ramsundar, B.; Feinberg, Evan N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V., MoleculeNet: a benchmark for molecular machine learning. Chemical Science 2018, 9, 513-530.

9. Li, Y.; Rezaei, M. A.; Li, C.; Li, X., DeepAtom: A Framework for Protein-Ligand Binding Affinity Prediction. IEEE International Conference on Bioinformatics and Biomedicine (BIBM) 2019, 303-310.

10. Hassan-Harrirou, H.; Zhang, C.; Lemmin, T., RosENet: Improving Binding Affinity Prediction by Leveraging Molecular Mechanics Energies with an Ensemble of 3D Convolutional Neural Networks. Journal of chemical information and modeling 2020, 60, 2791-2802.

11. Kwon, Y.; Shin, W.-H.; Ko, J.; Lee, J., AK-Score: Accurate Protein-Ligand Binding Affinity Prediction Using an Ensemble of 3D-Convolutional Neural Networks. International Journal of Molecular Sciences 2020, 21, 8424.

12. Son, J.; Kim, D., Development of a graph convolutional neural network model for efficient prediction of protein-ligand binding affinities. PLoS One 2021, 16, e0249404.

13. Volkov, M.; Turk, J.-A.; Drizard, N.; Martin, N.; Hoffmann, B.; Gaston-Mathé, Y.; Rognan, D., On the Frustration to Predict Binding Affinities from Protein–Ligand Structures with Deep Neural Networks. Journal of medicinal chemistry 2022.

14. Lim, J.; Ryu, S.; Park, K.; Choe, Y. J.; Ham, J.; Kim, W. Y., Predicting Drug–Target Interaction Using a Novel Graph Neural Network with 3D Structure-Embedded Graph Representation. Journal of chemical information and modeling 2019, 59, 3981-3988.

15. Feinberg, E. N.; Sur, D.; Wu, Z.; Husic, B. E.; Mai, H.; Li, Y.; Sun, S.; Yang, J.; Ramsundar, B.; Pande, V. S., PotentialNet for Molecular Property Prediction. ACS Central Science 2018, 4, 1520-1530.

16. Moon, S.; Zhung, W.; Yang, S.; Lim, J.; Kim, W. Y., PIGNet: a physics-informed deep learning model toward generalized drug–target interaction predictions. Chemical Science 2022, 13, 3661-3673.

17. Wang, R.; Fang, X.; Lu, Y.; Wang, S., The PDBbind database: collection of binding affinities for protein-ligand complexes with known three-dimensional structures. Journal of medicinal chemistry 2004, 47, 2977-80.

18. Deng, J.; Dong, W.; Socher, R.; Li, L. J.; Kai, L.; Li, F.-F. ImageNet: A large-scale hierarchical image database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, 20-25 June, 2009; 2009; pp 248-255.

19. Libouban, P.-Y.; Aci-Sèche, S.; Gómez-Tamayo, J. C.; Tresadern, G.; Bonnet, P., The Impact of Data on Structure-Based Binding Affinity Predictions Using Deep Neural Networks. International Journal of Molecular Sciences 2023, 24, 16120.

20. Sieg, J.; Flachsenberg, F.; Rarey, M., In Need of Bias Control: Evaluating Chemical Data for Machine Learning in Structure-Based Virtual Screening. Journal of chemical information and modeling 2019, 59, 947-961.

21. Scantlebury, J.; Brown, N.; Von Delft, F.; Deane, C. M., Data Set Augmentation Allows Deep Learning-Based Virtual Screening to Better Generalize to Unseen Target Classes and Highlight Important Binding Interactions. Journal of chemical information and modeling 2020, 60, 3722-3730.

22. Hansson, T.; Marelius, J.; Åqvist, J., Ligand binding affinity prediction by linear interaction energy methods. Journal of Computer-Aided Molecular Design 1998, 12, 27-35.

23. Hou, T.; Wang, J.; Li, Y.; Wang, W., Assessing the performance of the MM/PBSA and MM/GBSA methods. 1. The accuracy of binding free energy calculations based on molecular dynamics simulations. Journal of chemical information and modeling 2011, 51, 69-82.

24. Chen, W.; Cui, D.; Jerome, S. V.; Michino, M.; Lenselink, E. B.; Huggins, D. J.; Beautrait, A.; Vendome, J.; Abel, R.; Friesner, R. A.; Wang, L., Enhancing Hit Discovery in Virtual Screening through Absolute Protein–Ligand Binding Free-Energy Calculations. Journal of chemical information and modeling 2023, 63, 3171-3185.

25. Gapsys, V.; Hahn, D. F.; Tresadern, G.; Mobley, D. L.; Rampp, M.; de Groot, B. L., Pre-Exascale Computing of Protein–Ligand Binding Free Energies with Open Source Software for Drug Design. Journal of chemical information and modeling 2022, 62, 1172-1177.

26. Khalak, Y.; Tresadern, G.; Aldeghi, M.; Baumann, H. M.; Mobley, D. L.; de Groot, B. L.; Gapsys, V., Alchemical absolute protein–ligand binding free energies for drug design. Chemical Science 2021, 12, 13958-13971.

27. Ragoza, M.; Hochuli, J.; Idrobo, E.; Sunseri, J.; Koes, D. R., Protein-Ligand Scoring with Convolutional Neural Networks. Journal of chemical information and modeling 2017, 57, 942-957.

28. Xie, L.; Xu, L.; Chang, S.; Xu, X.; Meng, L., Multitask deep networks with grid featurization achieve improved scoring performance for protein–ligand binding. Chemical Biology & Drug Design 2020, 96, 973-983.

29. Pereira, J. C.; Caffarena, E. R.; dos Santos, C. N., Boosting Docking-Based Virtual Screening with Deep Learning. Journal of chemical information and modeling 2016, 56, 2495-2506.

30. Korlepara, D. B.; Vasavi, C. S.; Jeurkar, S.; Pal, P. K.; Roy, S.; Mehta, S.; Sharma, S.; Kumar, V.; Muvva, C.; Sridharan, B.; Garg, A.; Modee, R.; Bhati, A. P.; Nayar, D.; Priyakumar, U. D., PLAS-5k: Dataset of Protein-Ligand Affinities from Molecular Dynamics for Machine Learning Applications. Scientific Data 2022, 9, 548.

31. Siebenmorgen, T.; Menezes, F.; Benassou, S.; Merdivan, E.; Kesselheim, S.; Piraud, M.; Theis, F. J.; Sattler, M.; Popowicz, G. M., MISATO - Machine learning dataset for structure-based drug discovery. bioRxiv 2023, 2023.05.24.542082.

32. Zheng, L.; Fan, J.; Mu, Y., OnionNet: a Multiple-Layer Intermolecular-Contact-Based Convolutional Neural Network for Protein–Ligand Binding Affinity Prediction. ACS Omega 2019, 4, 15956-15965.

33. Gu, S.; Shen, C.; Yu, J.; Zhao, H.; Liu, H.; Liu, L.; Sheng, R.; Xu, L.; Wang, Z.; Hou, T.; Kang, Y., Can molecular dynamics simulations improve predictions of protein-ligand binding affinity with machine learning? Briefings in bioinformatics 2023, 24.

34. Jamal, S.; Grover, A.; Grover, S., Machine Learning From Molecular Dynamics Trajectories to Predict Caspase-8 Inhibitors Against Alzheimer's Disease. Frontiers in Pharmacology 2019, 10.

35. Riniker, S., Molecular Dynamics Fingerprints (MDFP): Machine Learning from MD Data To Predict Free-Energy Differences. Journal of chemical information and modeling 2017, 57, 726-741.

36. Wu, F.; Jin, S.; Jiang, Y.; Jin, X.; Tang, B.; Niu, Z.; Liu, X.; Zhang, Q.; Zeng, X.; Li, S. Z., Pre-Training of Equivariant Graph Matching Networks with Conformation Flexibility for Drug Binding. Advanced Science 2022, 9, 2203796.

37. Townshend, R. J.; Vögele, M.; Suriana, P.; Derry, A.; Powers, A.; Laloudakis, Y.; Balachandar, S.; Jing, B.; Anderson, B.; Eismann, S., Atom3d: Tasks on molecules in three dimensions. arXiv preprint arXiv:2012.04035 2020.

38. Berishvili, V. P.; Perkin, V. O.; Voronkov, A. E.; Radchenko, E. V.; Syed, R.; Venkata Ramana Reddy, C.; Pillay, V.; Kumar, P.; Choonara, Y. E.; Kamal, A.; Palyulin, V. A., Time-Domain Analysis of Molecular Dynamics Trajectories Using Deep Neural Networks: Application to Activity Ranking of Tankyrase Inhibitors. Journal of chemical information and modeling 2019, 59, 3519-3532.

39. Min, Y.; Wei, Y.; Wang, P.; Wu, N.; Bauer, S.; Zheng, S.; Shi, Y.; Wang, Y.; Wang, X.; Zhao, D., Predicting the protein-ligand affinity from molecular dynamics trajectories. arXiv preprint arXiv:2208.10230 2022.

40. Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E., The Protein Data Bank. Nucleic acids research 2000, 28, 235-242.

41. Adler, M.; Beroza, P., Improved Ligand Binding Energies Derived from Molecular Dynamics: Replicate Sampling Enhances the Search of Conformational Space. Journal of chemical information and modeling 2013, 53, 2065-2072.

42. Wright, D. W.; Hall, B. A.; Kenway, O. A.; Jha, S.; Coveney, P. V., Computing Clinically Relevant Binding Free Energies of HIV-1 Protease Inhibitors. Journal of Chemical Theory and Computation 2014, 10, 1228-1241.

43. Wright, D. W.; Wan, S.; Meyer, C.; van Vlijmen, H.; Tresadern, G.; Coveney, P. V., Application of ESMACS binding free energy protocols to diverse datasets: Bromodomain-containing protein 4. Scientific Reports 2019, 9, 6017.

44. Wan, S.; Tresadern, G.; Pérez-Benito, L.; van Vlijmen, H.; Coveney, P. V., Accuracy and Precision of Alchemical Relative Free Energy Predictions with and without Replica-Exchange. Advanced Theory and Simulations 2020, 3, 1900195.

45. Liu, K.; Watanabe, E.; Kokubo, H., Exploring the stability of ligand binding modes to proteins by molecular dynamics simulations. Journal of Computer-Aided Molecular Design 2017, 31, 201-211.

46. Liu, K.; Kokubo, H., Exploring the Stability of Ligand Binding Modes to Proteins by Molecular Dynamics Simulations: A Cross-docking Study. Journal of chemical information and modeling 2017, 57, 2514-2522.

47. Liu, K.; Kokubo, H., Prediction of ligand binding mode among multiple cross-docking poses by molecular dynamics simulations. Journal of Computer-Aided Molecular Design 2020, 34, 1195-1205.

48. Guterres, H.; Im, W., Improving Protein-Ligand Docking Results with High-Throughput Molecular Dynamics Simulations. Journal of chemical information and modeling 2020, 60, 2189-2198.

49. Pearlman, D. A.; Case, D. A.; Caldwell, J. W.; Ross, W. S.; Cheatham, T. E.; DeBolt, S.; Ferguson, D.; Seibel, G.; Kollman, P., AMBER, a package of computer programs for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to simulate the structural and energetic properties of molecules. Computer Physics Communications 1995, 91, 1-41.

50. Wang, J.; Wang, W.; Kollman, P. A.; Case, D. A., Automatic atom type and bond type perception in molecular mechanical calculations. Journal of Molecular Graphics and Modelling 2006, 25, 247-260.

51. Jakalian, A.; Bush, B. L.; Jack, D. B.; Bayly, C. I., Fast, efficient generation of high-quality atomic charges. AM1-BCC model: I. Method. Journal of Computational Chemistry 2000, 21, 132-146.

52. Banáš, P.; Hollas, D.; Zgarbová, M.; Jurečka, P.; Orozco, M.; Cheatham, T. E., 3rd; Šponer, J.; Otyepka, M., Performance of Molecular Mechanics Force Fields for RNA Simulations: Stability of UUCG and GNRA Hairpins. J Chem Theory Comput 2010, 6, 3836-3849.

53. Giammona, D. University of California, Davis, 1984.

54. Holmberg, N.; Ryde, U.; Bülow, L., Redesign of the coenzyme specificity in L-lactate dehydrogenase from bacillus stearothermophilus using site-directed mutagenesis and media engineering. Protein engineering 1999, 12, 851-6.

55. Khoury, G. A.; Thompson, J. P.; Smadbeck, J.; Kieslich, C. A.; Floudas, C. A., Forcefield_PTM: Ab Initio Charge and AMBER Forcefield Parameters for Frequently Occurring Post-Translational Modifications. Journal of Chemical Theory and Computation 2013, 9, 5653-5674.

56. Khoury, G. A.; Smadbeck, J.; Tamamis, P.; Vandris, A. C.; Kieslich, C. A.; Floudas, C. A., Forcefield_NCAA: Ab Initio Charge Parameters to Aid in the Discovery and Design of Therapeutic Proteins and Peptides with Unnatural Amino Acids and Their Application to Complement Inhibitors of the Compstatin Family. ACS Synthetic Biology 2014, 3, 855-869.

57. Krepl, M.; Zgarbová, M.; Stadlbauer, P.; Otyepka, M.; Banáš, P.; Koča, J.; Cheatham, T. E., 3rd; Jurečka, P.; Sponer, J., Reference simulations of noncanonical nucleic acids with different χ variants of the AMBER force field: quadruplex DNA, quadruplex RNA and Z-DNA. J Chem Theory Comput 2012, 8, 2506-2520.

58. Meagher, K. L.; Redman, L. T.; Carlson, H. A., Development of polyphosphate parameters for use with the AMBER force field. J Comput Chem 2003, 24, 1016-25.

59. Ryde, U., On the role of Glu-68 in alcohol dehydrogenase. Protein Science 1995, 4, 1124-1132.

60. Ryde, U., Molecular dynamics simulations of alcohol dehydrogenase with a four- or five-coordinate catalytic zinc ion. Proteins 1995, 21, 40-56.

61. Schneider, C.; Sühnel, J., A molecular dynamics simulation of the flavin mononucleotide–RNA aptamer complex. Biopolymers 1999, 50, 287-302.

62. Zgarbová, M.; Otyepka, M.; Šponer, J.; Mládek, A.; Banáš, P.; Cheatham, T. E., III; Jurečka, P., Refinement of the Cornell et al. Nucleic Acids Force Field Based on Reference Quantum Chemical Calculations of Glycosidic Torsion Profiles. Journal of Chemical Theory and Computation 2011, 7, 2886-2902.

63. Zgarbová, M.; Luque, F. J.; Sponer, J.; Cheatham, T. E., 3rd; Otyepka, M.; Jurečka, P., Toward Improved Description of DNA Backbone: Revisiting Epsilon and Zeta Torsion Force Field Parameters. J Chem Theory Comput 2013, 9, 2339-2354.

64. Zgarbová, M.; Šponer, J.; Otyepka, M.; Cheatham, T. E., III; Galindo-Murillo, R.; Jurečka, P., Refinement of the Sugar–Phosphate Backbone Torsion Beta for AMBER Force Fields Improves the Description of Z- and B-DNA. Journal of Chemical Theory and Computation 2015, 11, 5723-5736.

65. Maier, J. A.; Martinez, C.; Kasavajhala, K.; Wickstrom, L.; Hauser, K. E.; Simmerling, C., ff14SB: Improving the Accuracy of Protein Side Chain and Backbone Parameters from ff99SB. Journal of Chemical Theory and Computation 2015, 11, 3696-3713.

66. Wang, J.; Wolf, R. M.; Caldwell, J. W.; Kollman, P. A.; Case, D. A., Development and testing of a general amber force field. J Comput Chem 2004, 25, 1157-74.

67. Mark, P.; Nilsson, L., Structure and Dynamics of the TIP3P, SPC, and SPC/E Water Models at 298 K. The Journal of Physical Chemistry A 2001, 105, 9954-9960.

68. Ryckaert, J.-P.; Ciccotti, G.; Berendsen, H. J. C., Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes. Journal of Computational Physics 1977, 23, 327-341.

69. Essmann, U.; Perera, L.; Berkowitz, M. L.; Darden, T.; Lee, H.; Pedersen, L. G., A smooth particle mesh Ewald method. The Journal of Chemical Physics 1995, 103, 8577-8593.

70. DeLano, W. L., Pymol: An open-source molecular graphics tool. CCP4 Newsl. Protein Crystallogr 2002, 40, 82-92.

71. Nguyen, H.; Roe, D. R. Pytraj. https://github.com/Amber-MD/pytraj

72. Roe, D. R.; Cheatham, T. E., III, PTRAJ and CPPTRAJ: Software for Processing and Analysis of Molecular Dynamics Trajectory Data. Journal of Chemical Theory and Computation 2013, 9, 3084-3095.

73. Gowers, R. J.; Linke, M.; Barnoud, J.; Reddy, T. J.; Melo, M. N.; Seyler, S. L.; Domanski, J.; Dotson, D. L.; Buchoux, S.; Kenney, I. M. MDAnalysis: a Python package for the rapid analysis of molecular dynamics simulations. In Proceedings of the 15th python in science conference, 2016; SciPy Austin, TX: 2016; Vol. 98; p 105.

74. Pettersen, E. F.; Goddard, T. D.; Huang, C. C.; Couch, G. S.; Greenblatt, D. M.; Meng, E. C.; Ferrin, T. E., UCSF Chimera—A visualization system for exploratory research and analysis. Journal of Computational Chemistry 2004, 25, 1605-1612.

75. Wang, L.; Wu, Y.; Deng, Y.; Kim, B.; Pierce, L.; Krilov, G.; Lupyan, D.; Robinson, S.; Dahlgren, M. K.; Greenwood, J.; Romero, D. L.; Masse, C.; Knight, J. L.; Steinbrecher, T.; Beuming, T.; Damm, W.; Harder, E.; Sherman, W.; Brewer, M.; Wester, R.; Murcko, M.; Frye, L.; Farid, R.; Lin, T.; Mobley, D. L.; Jorgensen, W. L.; Berne, B. J.; Friesner, R. A.; Abel, R., Accurate and Reliable Prediction of Relative Ligand Binding Potency in Prospective Drug Discovery by Way of a Modern Free-Energy Calculation Protocol and Force Field. Journal of the American Chemical Society 2015, 137, 2695-2703.

76. Imrie, F.; Bradley, A. R.; van der Schaar, M.; Deane, C. M., Protein Family-Specific Models Using Deep Neural Networks and Transfer Learning Improve Virtual Screening and Highlight the Need for More Data. Journal of chemical information and modeling 2018, 58, 2319-2330.

77. Huang, G.; Liu, Z.; Maaten, L. V. D.; Weinberger, K. Q., In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); IEEE Computer Society: 2017, pp 2261-2269.

78. Donahue, J.; Hendricks, L. A.; Rohrbach, M.; Venugopalan, S.; Guadarrama, S.; Saenko, K.; Darrell, T., Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. IEEE Transactions on Pattern Analysis and Machine Intelligence 2017, 39, 677-691.

79. Yuan, Z.; Zhou, X.; Yang, T., In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; Association for Computing Machinery: London, United Kingdom, 2018, pp 984–992.

80. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-k.; Woo, W.-c. Convolutional LSTM Network: a machine learning approach for precipitation nowcasting. In Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, Canada, 2015; MIT Press: Montreal, Canada, 2015; Vol. 1; pp 802–810.

81. Simonyan, K.; Zisserman, A., Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 2014.

82. Volkov, M. Design and application of deep learning methods to structure-based drug design. 2023.

83. Li, H.; Leung, K.-S.; Wong, M.-H.; Ballester, P. J., Correcting the impact of docking pose generation error on binding affinity prediction. BMC Bioinformatics 2016, 17, 308.

84. Boyles, F.; Deane, C. M.; Morris, G. M., Learning from Docked Ligands: Ligand-Based Features Rescue Structure-Based Scoring Functions When Trained on Docked Poses. Journal of chemical information and modeling 2022, 62, 5329-5341.

85. Ballester, P. J.; Mitchell, J. B. O., A machine learning approach to predicting protein-ligand binding affinity with applications to molecular docking. Bioinformatics (Oxford, England) 2010, 26, 1169-1175.

86. Ilse, M.; Tomczak, J.; Welling, M. Attention-based deep multiple instance learning. In International conference on machine learning, 2018; PMLR: 2018; pp 2127-2136.

87. Zankov, D. V.; Matveieva, M.; Nikonenko, A. V.; Nugmanov, R. I.; Baskin, I. I.; Varnek, A.; Polishchuk, P.; Madzhidov, T. I., QSAR Modeling Based on Conformation Ensembles Using a Multi-Instance Learning Approach. Journal of chemical information and modeling 2021, 61, 4913-4923.

88. Du, Z.; Zhang, G.; Lu, W.; Zhao, T.; Wu, P., Spatio-Temporal Transformer for Online Video Understanding. Journal of Physics: Conference Series 2022, 2171, 012020.

89. Neimark, D.; Bar, O.; Zohar, M.; Asselmann, D. Video transformer network. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021; 2021; pp 3163-3172.

90. Selva, J.; Johansen, A. S.; Escalera, S.; Nasrollahi, K.; Moeslund, T. B.; Clapés, A., Video transformers: A survey. arXiv preprint arXiv:2201.05991 2022.

# Support information

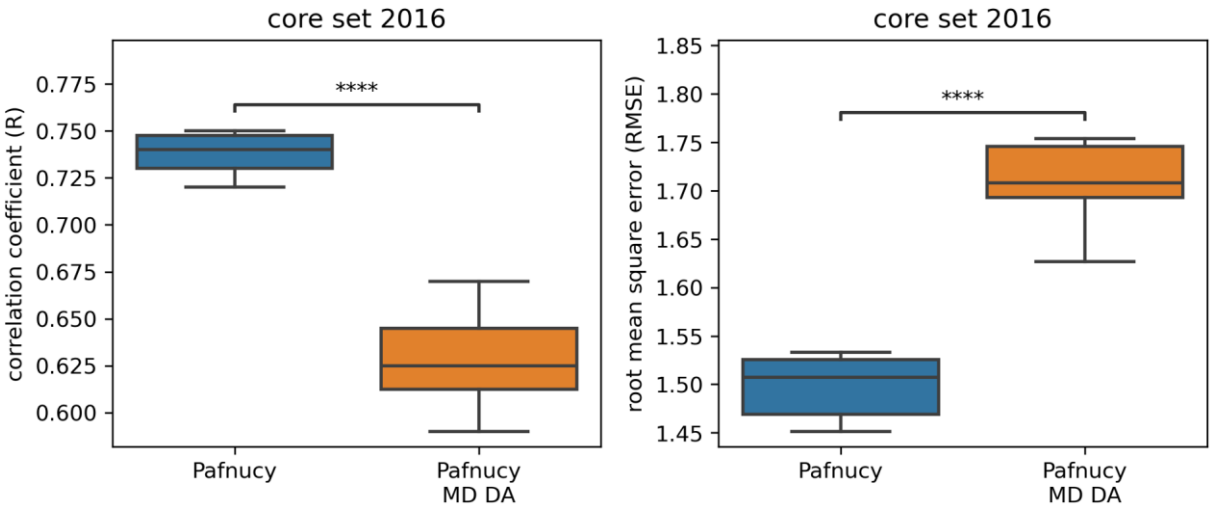



Figure S1: Comparison of the performance of Pafnucy trained with or without MD data augmentation (MD DA). In both case, models were evaluated against the PDBbind v.2016 core set. The following p-values correspond to the annotations on the plots: ****: $p \leq 1.00 \times 10^{-4}$.

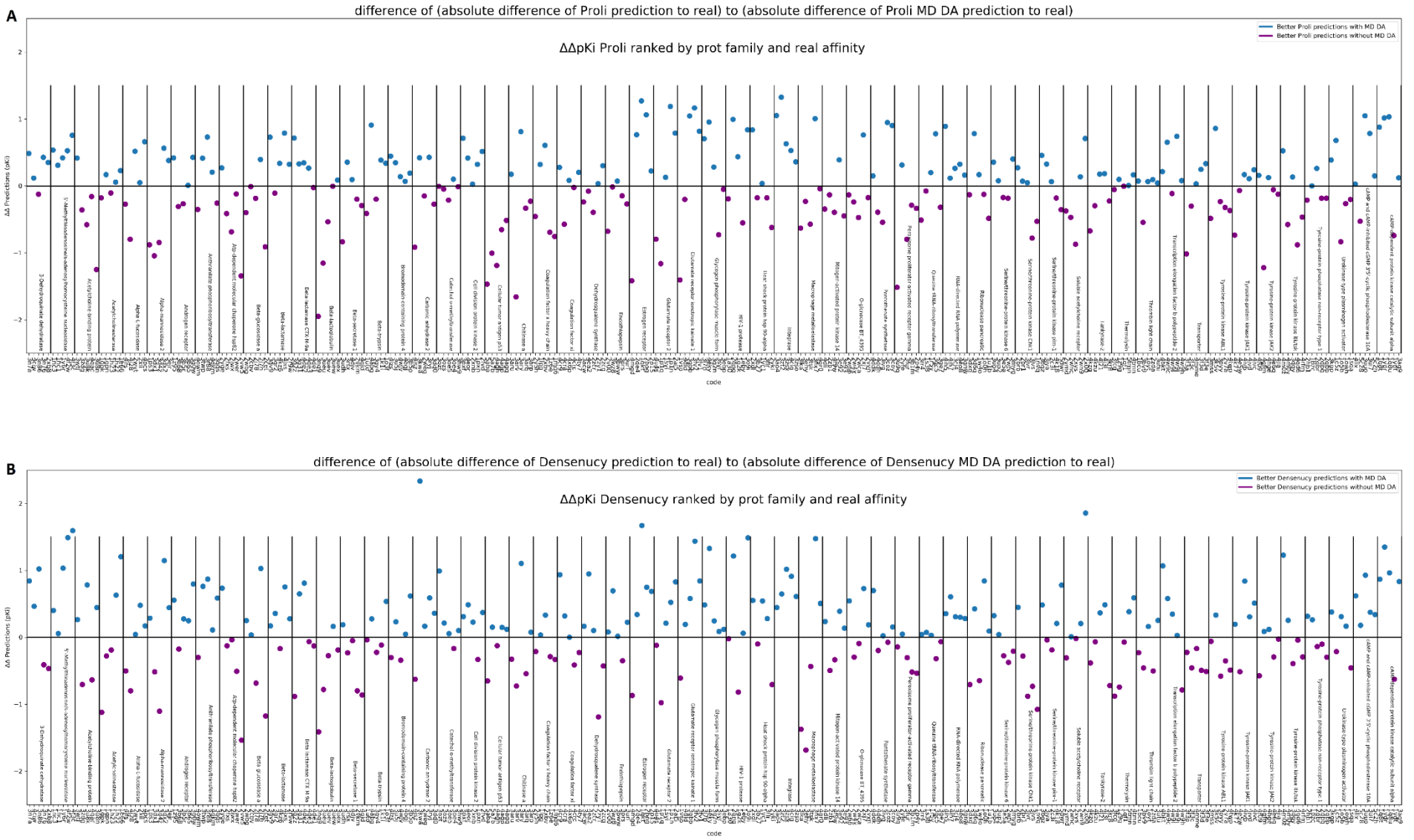

A
difference of (absolute difference of Proli prediction to real) to (absolute difference of Proli MD DA prediction to real)
ΔΔpKi Proli ranked by prot family and real affinity
Better Proli predictions with MD DA
Better Proli predictions without MD DA
ΔΔ Predictions (pKi)
code
B
difference of (absolute difference of Densenucy prediction to real) to (absolute difference of Densenucy MD DA prediction to real)
ΔΔpKi Densenucy ranked by prot family and real affinity
Better Densenucy predictions with MD DA
Better Densenucy predictions without MD DA
ΔΔ Predictions (pKi)
code

Figure S2: ΔΔ Predictions with and without MD data augmentation for Proli and Densenucy on the PDBbind core set 2016. The differences are presented individually for each complex, organized by protein and binding affinity ranking. Vertical lines separate each cluster of five complexes. Blue dots indicate an enhancement in performance with MD data augmentation, while purple denotes lower performance with MD data augmentation.

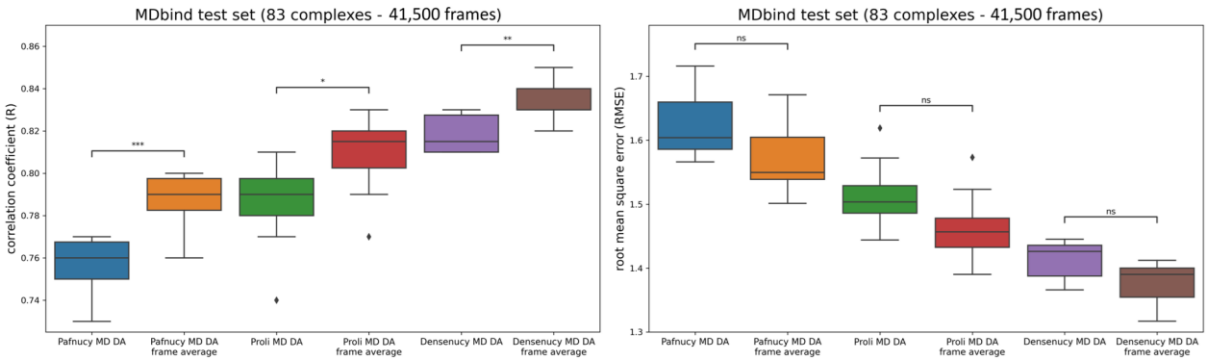



Figure S3: Comparison of the performance of MD data augmentation models (Pafnucy, Proli and Densenucy) evaluated on MDbind test set (41,500 frames) with or without averaging the predicted values per simulation/complex. “MD DA” stands for MD data augmentation. The following p-values correspond to the annotations on the plots: ns: $5.00 \times 10^{-2} < p \leq 1.00 \times 10^{0}$, *: $1.00 \times 10^{-2} < p \leq 5.00 \times 10^{-2}$, **: $1.00 \times 10^{-3} < p \leq 1.00 \times 10^{-2}$, ***: $1.00 \times 10^{-4} < p \leq 1.00 \times 10^{-3}$, and ♦ are possible outliers.

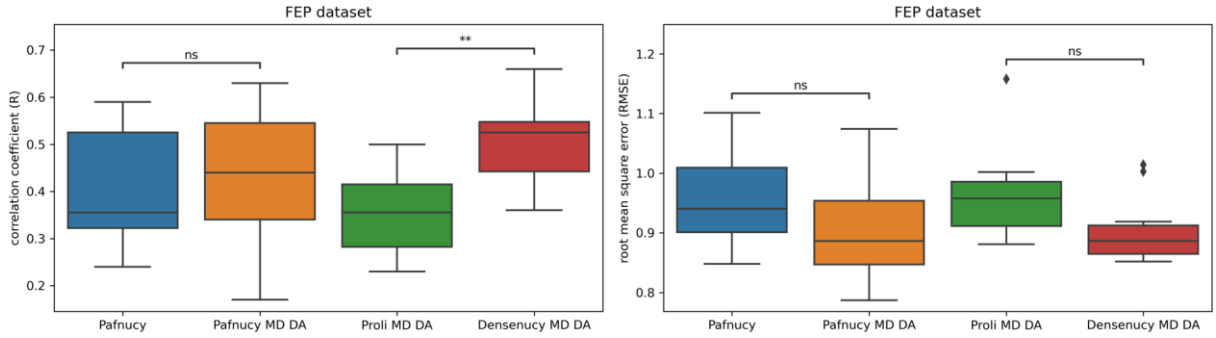



Figure S4: Comparison of the performance of Pafnucy, Pafnucy MD DA, Proli MD DA and Densenucy MD DA on the FEP dataset. “MD DA” stands for MD data augmentation. The following p-values correspond to the annotations on the plots: ns: $5.00 \times 10^{-2} < p \leq 1.00 \times 10^{0}$, **: $1.00 \times 10^{-3} < p \leq 1.00 \times 10^{-2}$, and ♦ are possible outliers.

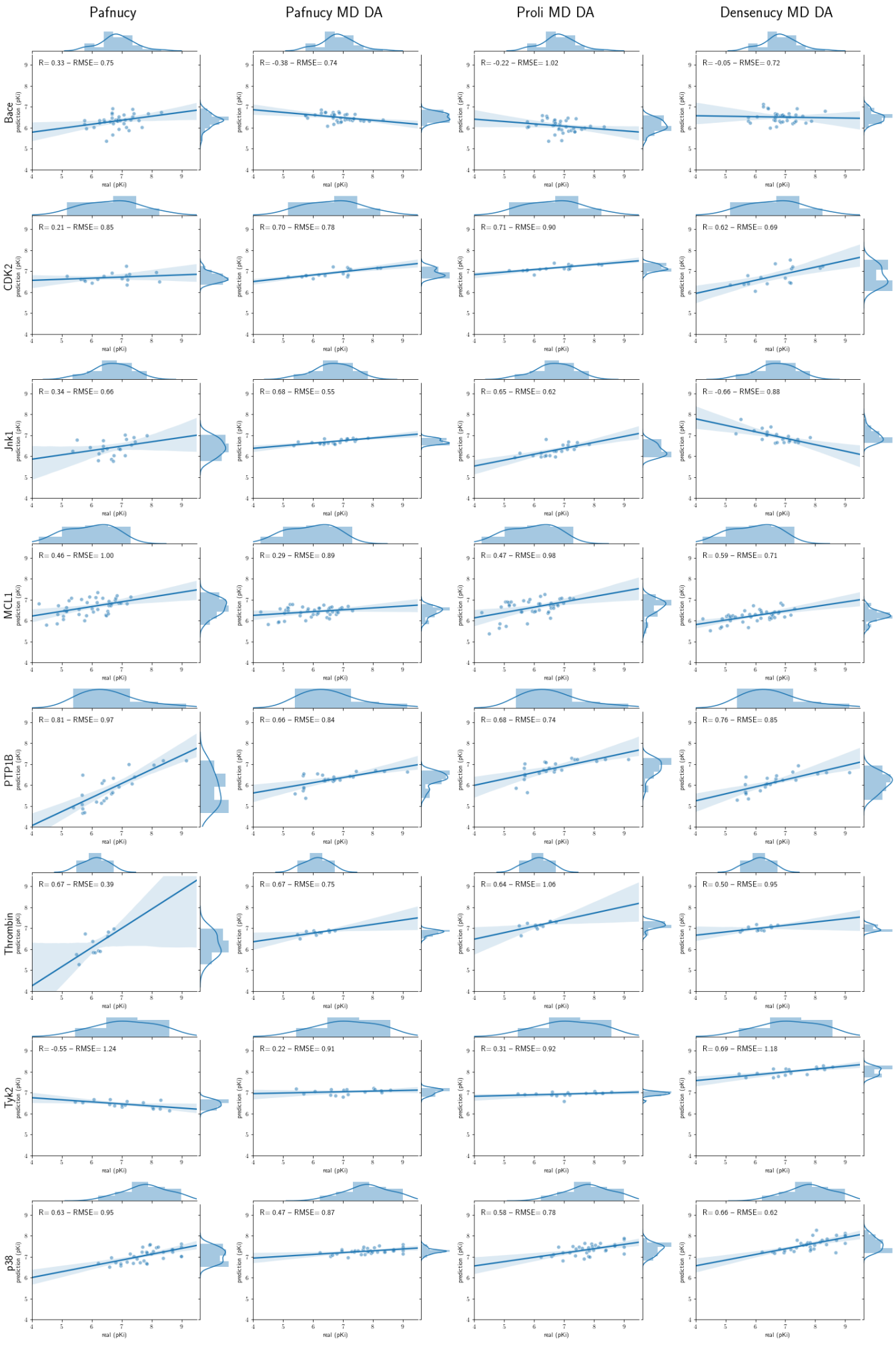

Pafnucy
Pafnucy MD DA
Proli MD DA
Densenucy MD DA
Bace
CDK2
Jnk1
MCL1
PTP1B
Thrombin
Tyk2
p38
R= 0.33 – RMSE= 0.75
R= -0.38 – RMSE= 0.74
R= -0.22 – RMSE= 1.02
R= -0.05 – RMSE= 0.72
R= 0.21 – RMSE= 0.85
R= 0.70 – RMSE= 0.78
R= 0.71 – RMSE= 0.90
R= 0.62 – RMSE= 0.69
R= 0.34 – RMSE= 0.66
R= 0.68 – RMSE= 0.55
R= 0.65 – RMSE= 0.62
R= -0.66 – RMSE= 0.88
R= 0.46 – RMSE= 1.00
R= 0.29 – RMSE= 0.89
R= 0.47 – RMSE= 0.98
R= 0.59 – RMSE= 0.71
R= 0.81 – RMSE= 0.97
R= 0.66 – RMSE= 0.84
R= 0.68 – RMSE= 0.74
R= 0.76 – RMSE= 0.85
R= 0.67 – RMSE= 0.39
R= 0.67 – RMSE= 0.75
R= 0.64 – RMSE= 1.06
R= 0.50 – RMSE= 0.95
R= -0.55 – RMSE= 1.24
R= 0.22 – RMSE= 0.91
R= 0.31 – RMSE= 0.92
R= 0.69 – RMSE= 1.18
R= 0.63 – RMSE= 0.95
R= 0.47 – RMSE= 0.87
R= 0.58 – RMSE= 0.78
R= 0.66 – RMSE= 0.62
prediction (pKi)
real (pKi)

Figure S5: Correlation plots of Pafnucy, Pafnucy MD DA, Proli MD DA and Densenucy MD DA for each protein of the FEP dataset. “MD DA” stands for MD data augmentation.

## 4.3 MD dataset preparation

Given the goal of performing MD simulations on a large scale, the entire process needed to be automated. As a result, MD simulations were not conducted on all the PDBbind's complexes, primarily because we could not meticulously prepare complexes as it is typically required. Nonetheless, we tried our best to fix the issues that arose during this stage. The rationale behind addressing issues for individual complexes was that the automated process could potentially address similar issues in several tens or hundreds of other complexes.

Since the PDBbind dataset was not originally designed for carrying MD simulations, we had to carefully choose the complexes compatible with MD simulations and address any issues that would prevent MD simulation from running smoothly. Here, we provide examples of the challenges and problems we encountered within the PDBbind dataset.

### 4.3.1 PDBbind limitations

Here is a list of specific ligands we encountered within the PDBbind dataset. In some cases, the complexes containing these ligands were identified and subsequently excluded to avoid conducting MD simulations on them:

- **Unusual ligand atoms**: Ligands composed of atoms that have no parameters in the GAFF force-field, such as Arsenic (As), Boron (B), Beryllium (Be), Copper (Cu), Iron (Fe), Mercury (Hg), Iridium (Ir), Magnesium (Mg), Osmium (Os), Rhenium (Re), Rhodium (Rh), Ruthenium (Ru), Antimony (Sb), Selenium (Se), Silicon (Si), Vanadium (V), and Zinc (Zn). There are approximately 260 complexes with ligands containing these unusual atoms, including examples like 3qlb or 4rlp that contain Fe and Ru atoms, respectively. Some ligands even had up to 10 Boron atoms, such as 3vjt and 3vjs.
- **Sugar as ligand**: Many ligands were sugars, including both mono and polysaccharides. An example is 5kvm, which contains sialic acid as a ligand.
- **Natural ligands**: Some complexes contained natural ligands like GTP, ATP, GDP, and ADP. An example is 1e8h.
- **ARN as ligands**: There were complexes with ARN as ligands, such as 4g0a, 1b2m, 4i67, and 3bbb.
- **Peptide as ligands and targets**: Within the PDBbind dataset, we identified a total of 2,915 peptides serving as ligands, including numerous instances of peptides containing more than 10 amino acids. Additionally, there was a case of peptide-peptide interaction in complex 2lyw, where a peptide comprising 24 amino acids interacted with another peptide composed of 13 amino acids.

Here is a list of issues attributed to PDBbind modifications to files:

- **Wrong ligands**: In the case of 2r1w, the ligand provided is actually the ligand of 2r1y. These ligands are not the same, and their 3D positions do not match; the ligand is entirely outside of the binding pocket (Figure 52).

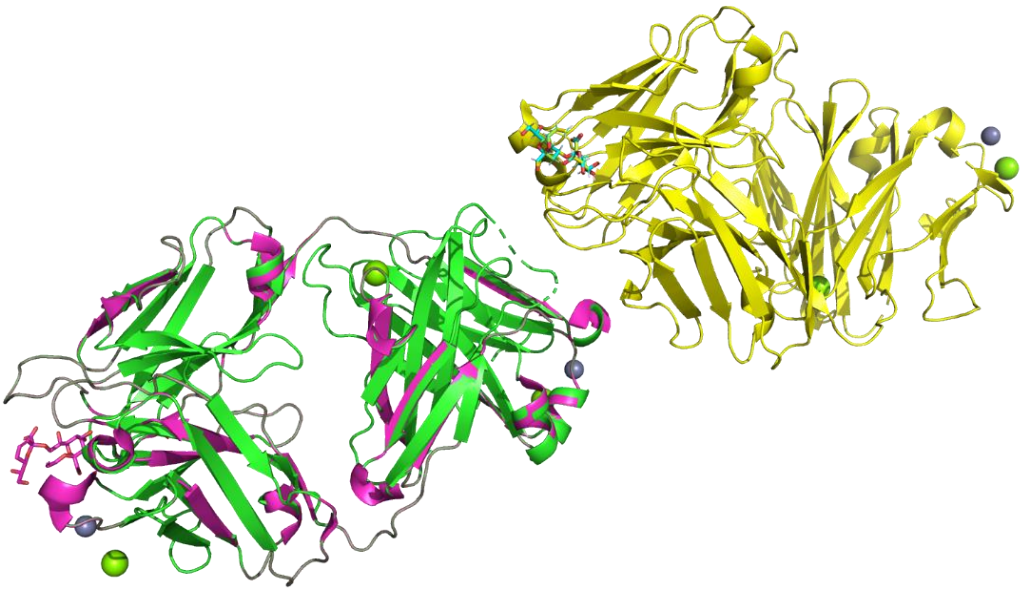


Figure 52: PDBbind provides an incorrect ligand for 2r1w. In the image, the green structure represents the 2r1w protein provided by PDBbind, while the blue structure represents the ligand provided by PDBbind. The 2r1w complex obtained via the PDB is depicted in purple, while the 2r1y complex, also obtained from the PDB, is represented in yellow. This illustrates the mismatch between the provided ligand and the actual complex structure.

- **Wrong ligand names**: Approximately 175 ligands were named “NON”, as observed in the ligand file provided for 1ux7. To rectify this, we referred to the information available in PDBbind's index folder, which includes ligand names. We replaced the "NON" labels in the ligand file with the names specified in the INDEX_general_PL_data.2019 file. Nonetheless, there are also errors in this list of names; for example, in the case of 3fqa, the name “GAB&PMP” is used when the ligand provided is only “GAB”. In the case of 3fuc, the name “9D9&9DG” is used, while only “9D9” is provided. Additionally, “9DG” does not even exist in the original PDB.
- **Incorrect ligand atom valence**: In certain structures like 3bho, some ligand atoms had valences that did not match their expected values. Additionally, abnormal bonds were present in this structure. To rectify these valence issues, manual corrections were made for several structures (1qpf, 1a7x, 1h07, and 4rlp) by changing double or triple bonds to single bonds. Following this, an automated procedure was established to address valence problems in specific cases.
- **Wrong ligand hydrogens**: Hydrogens were erroneously added, leading to valence issues in some cases, such as 4kbi. In other instances, when a ligand consisted of two sub-units, *e.g.*, "GAL-SIA" in 5vkm, the hydrogens of the "SIA" sub-unit were incorrectly labelled as part of the "GAL" sub-unit.
- **Renamed/renumbered ligand atoms**: The nomenclature used in PDBbind was found to be incompatible with Antechamber. For instance, we encountered atom names like “Furan”, “Ouran”, “Nuran” and “Suran” in certain cases. To resolve these issues, we implemented an automated process to replace “Furan” with “C”, “Ouran” with “O”, “Nuran” with “N”, and “Suran” with “S”.

  They also removed the « ‘ » from the atom notation. Regarding nucleotides, this led to the creation of duplicated atom names, as « ‘ » are typically used to differentiate between sugar atoms and base atoms (*e.g.* O1’, O3’, C1’). An example of this can be observed with 1ag9 (Figure 53). Both renamed and renumbered atoms prevent the use of the “reduce” tool (292) to readd hydrogens.

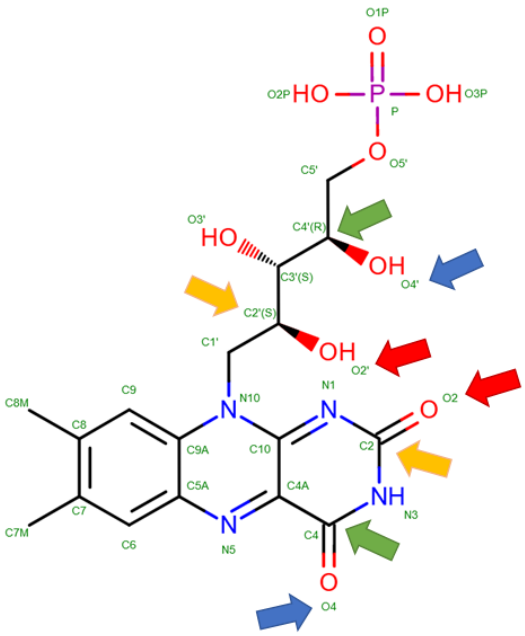



Figure 53: PDBbind duplicated atom name by removing « ' » in ligand. Example with FMN ligand from 1ag9.

- **Inconsistent biological assemblies**: There is inconsistency in biological assemblies within the same protein family. For example, in the acetylcholine binding protein family, 2ymd and 2c9t are presented as decamers (dimers of pentamers), while 2x00, 2yme, 4xk9, 5lxb, and 4xhe are represented as pentamers (Figure 54). To ensure consistency, we manually removed one pentamer from 2ymd and 2c9t.

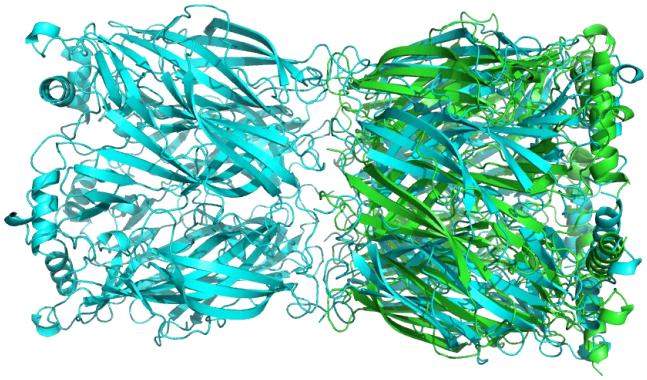


Figure 54: Several biological assemblies for the same protein. Visualisation of two complexes of the acetylcholine binding protein family provided by the PDBbind. 2ymd (2,111 amino acids) is displayed in cyan, while 2x00 (1,068 amino acids) is shown in green.

- **Incorrect biological assemblies**: PDBbind sometimes provides incomplete or inaccurate biological assemblies that are not suitable for binding affinity analysis and particularly for conducting MD simulations. For instance, in the case of the IP3R1 ion channel (6mu1), PDBbind provides only a partial structure of it (Figure 55 – A). Additionally, for 1vj5 and 1mqg, the structures provided consist of chains that are separated from each other, as demonstrated in Figure 55 – B. In the previous examples it was relatively straightforward to determine that PDBbind did not provide the biological assembly. However, in the case of 3ith, we were unable

to retrieve the biological assembly provided by PDBbind among the ones proposed in the PDB. Additionally, our efforts to recreate it using Chimera proved unsuccessful. This situation has raised substantial doubts about the accuracy of the biological assemblies provided by the PDBbind.

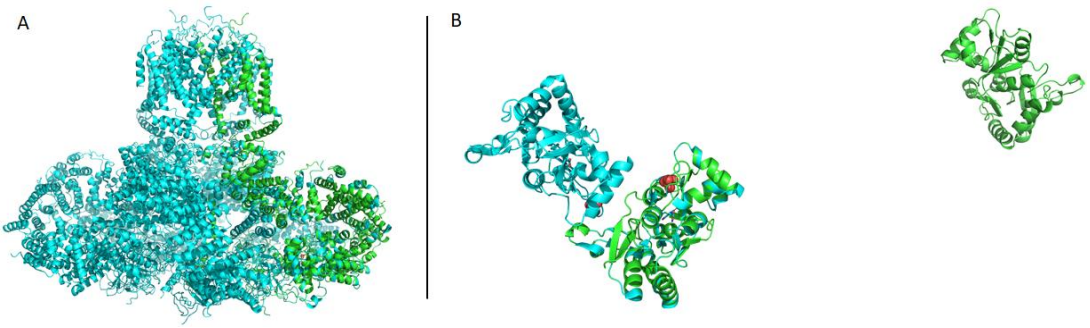



Figure 55: PDBbind provides incorrect biological assemblies. The blue structures represent the biological assemblies obtained via the PDB, while the green structures represent the biological assemblies provided by PDBbind. A – A part of the quaternary structure of an ions channel is missing (6mu1). B – Chains spread apart (1mqg).

Several preparation steps are necessary before conducting MD simulations. For example, it is essential to ensure that the complexes are fully resolved, which involves filling in any missing loops. Additional preparations tasks are required, such as ensuring correct protonation of the complexes, and adding partial charges, among other tasks.

### 4.3.2 Data preparation and fixation efforts

Custom codes were developed to address several key tasks, including the reassignment of appropriate ligand names, the renaming of ligand atoms unrecognized by Antechamber, and the correction of valence issues in ligands. Additionally, various filters were implemented to categorize peptides, exclude ligands containing unusual atoms, and flag complexes with protein missing residues.

The header of PDB files typically contains information about missing residues in proteins. Unfortunately, the header was not retained in PDBbind files. Consequently, we established a protocol to identify proteins with missing loops in the PDBbind. Initially, we looked for gaps in the amino acid numbering within the PDB files. For various reasons, this method yielded some false positives and false negatives. To enhance our detection method, we used the module pdb4amber from the AmberTools package (277). pdb4amber detects amino acid gaps by examining the distances between consecutive C-N atoms (amide bond); if the distance exceeds 2 Å, a gap is identified. However, pdb4amber ignores modified amino acids, resulting in numerous false positives. By combining both gap detection methods, most errors were mitigated.

Ultimately, we identified 7,642 complexes as having proteins with missing loops. Two distinct protocols were employed for loop reconstruction based on the size of the gaps. For gaps smaller than 10 consecutive amino acids, a "*de novo*" reconstruction method was employed. For larger gaps, we relied on modelling with structural templates. Around 58.5% of the complexes with missing loops (4,472 in total) had gaps of less than 10 consecutive amino acids. Modeller (44) was employed to reconstruct the missing loops (Figure 56).

- **"*De novo*" reconstruction method**: The amino acid sequences of the PDBbind proteins were compared to their reference sequences, which were contained within FASTA files downloaded

from the PDB. Various alignment methods, such as salign (Modeller integrated alignment tool) and emboss needle (293), were employed to identify the missing amino acids to be added to the structure. The PIR file generated from the alignment was used as input for Modeller, along with the protein structure. Reconstruction was carried out chain by chain, resulting in a more resilient but slower iterative process. Consequently, when a chain was reconstructed, the resulting structure was used as input of the reconstruction of the next chain. As PDBbind duplicated chains to provide biological assemblies, we replicated the corresponding segments within the reference sequence. This duplication was performed for homodimers, but it was not implemented for heterodimers. To prevent steric clashes, we added the ligand during the reconstruction process. As a result, we successfully reconstructed 3,871 proteins among the 4,472 with gaps of fewer than 10 amino acids.

- **Template-based reconstruction method**: When addressing gaps exceeding 10 amino acids, it becomes necessary to find a suitable structural template to accurately model the loops. To identify these templates, we compared the sequences of PDBbind proteins against the PDB_95 database, which contains representative sequences after clustering the PDB data at a 95% identity threshold. Our analysis encompassed a range of alignment methods, including basic local alignment search tool (BLAST) (294). In cases where no suitable structure was found, our plan was to use AlphaFold-generated models (1) as templates for reconstructing the missing loops. Unfortunately, due to time constraints, we did not undertake the reconstruction of proteins with missing loops exceeding 10 amino acids.

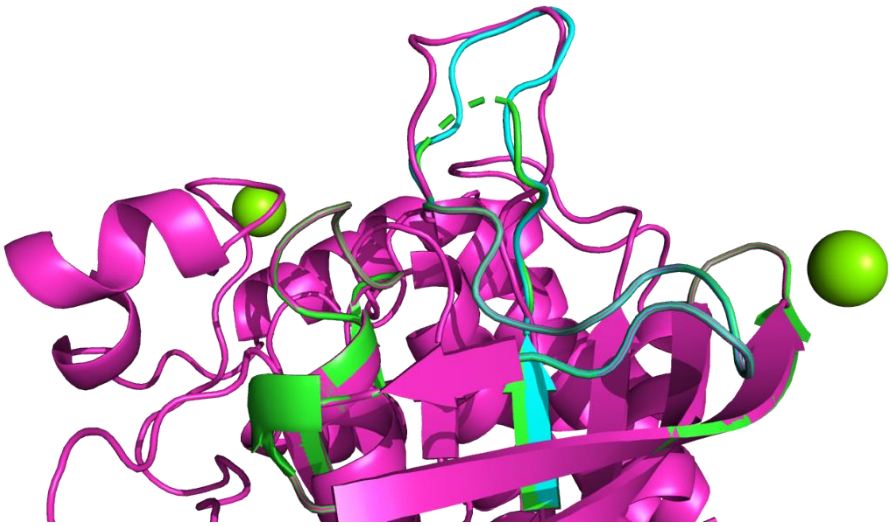


Figure 56: Examples of missing loop reconstructions using various modelling methods. In these visualizations, the structure provided by PDBbind is displayed in green, the reconstruction achieved with the *de novo* method is shown in cyan, and the structure reconstructed using a structural template is depicted in purple.

Further adjustments were made to the proteins, including the renaming of “MSE” modified residues (seleno-methionine) to “MET” residues (methionine). Selenium atoms were also turned into sulfur atoms (by renaming “SE” to “SD”), and the element designation was changed from “SE” to “S”. These conversions were necessary because selenium atoms are not parameterized in the force fields.

In our endeavour to address ligand-related issues, we established a workflow employing multiple software tools, including Open Babel (23), RDKit (22) and “reduce” (292). Our primary objectives were to sanitize, standardize, and re-add hydrogen atoms to the molecules. However, when it came to the hydrogen reassignment process, we encountered challenges when using the "reduce" tool. This was due to discrepancies in atom and ligand naming between the PDB data and the files provided by PDBbind, which had undergone renaming.

In an attempt to reconcile the atom name disparities, we explored options such as using the atom names provided in the reduce_wwPDB_het_dict.txt. Unfortunately, this approach did not work out as the PDBbind also modified the atom order. We also experimented with reassigning atom names based on the 3D positions from the original PDB files. However, this method proved problematic as PDBbind had either duplicated or omitted certain chains, rendering it impractical for our purposes. Consequently, we turned to Open Babel as an alternative for adding hydrogen atoms. In the molecular dynamics workflow, we used ligands with newly added hydrogen atoms as a fall back solution to address issues that might arise when using the initial ligands.

The calculation of partial charges using the AM1-BCC method (295) with Antechamber (296) necessitated the determination of the molecule's formal charge. We explored various methods to ascertain this formal charge, including summing the partial charges provided by the "reduce_wwPDB_het_dict.txt". However, these attempts did not yield satisfactory results. Ultimately, we resorted to calculate the formal charges with Chimera, and used RDKit as a fallback solution. It is worth mentioning that the utilization of RDKit was limited by being compatible only with mol2 files produced by Corina (297), which is not the case of the mol2 ligand files in the PDBbind dataset. For instance, this compatibility issue leads to challenges in handling PDBbind's phosphorus atoms, which are not supported by RDKit.

To reduce the computational workload associated with deep learning, we focused on training statistical models solely on the pockets, which are the key areas where protein and ligand interactions occur. Because we used frames extracted from molecular dynamics simulations as inputs for our models, we had to calculate the pockets for each frame. Subsequently, we developed two workflows for pocket calculation:

- We initiated our pocket calculation process by first evaluating the optimal pocket size that would yield the best performance without significantly increasing the computational cost of model training. As a result, we opted to disregard the pockets provided by PDBbind and, in turn, established our own pocket calculation protocol based on the crystallographic poses provided by PDBbind. We employed a Pymol (47) script capable of creating pockets by selecting the amino acids surrounding the ligands or within a specific range from the center of geometry of the ligand. However, limitations emerged when applying this method to frames. For instance, the inclusion of water molecules in the simulations occasionally caused the PDB file to exceed 9999 atoms, leading to a reset in atom numbering. As Pymol selections are based on atom numbers, selecting a specific atom number could inadvertently lead to selecting several atoms.
- Therefore, we investigated alternative methods for calculating pockets from frames and established a protocol comprising two steps. Initially, we employed pytraj (298) to autoimage the trajectory, a process that automatically centres and adjusts molecules within periodic boundaries, and to extract the frames. Subsequently, we applied MDtraj (299) to these extracted frames to compute the pockets. In this approach, we used the pockets calculated from the crystallographic poses as a reference for the determination of pockets within the frames. Consequently, we selected the residues from the crystal pose's pocket in the frames.

### 4.3.3 Large scale MD simulations

Due to the extensive number of MD simulations to perform, we established an automatized workflow capable of running on multiple computing centres.

Within our laboratory, we maintain two computing clusters:

- A central processing unit (CPU) cluster featuring 156 Intel Xeon processors.
- A GPU cluster comprising 2 RTX 3090 GPUs and 32 RTX 2080 GPUs.

Additionally, we were granted access to several external computing centres:

- CaSciModOT: A CPU/GPU computing cluster located in Orléans, equipped with 7 V100 GPUs.
- Myria (CRIANN): A regional computing center situated in Normandy, boasting 20 GPU V100s, 16 GPU P100s, and 413 Intel Xeon CPUs (766 TFlop/s) (300).
- Jean Zay (IDRIS): A national computing center composed of 456 A100 GPUs, 2,696 V100 GPUs, and 3,056 Intel Cascade Lake 6248 CPUs (37 Pflop/s) (Figure 57) (301).

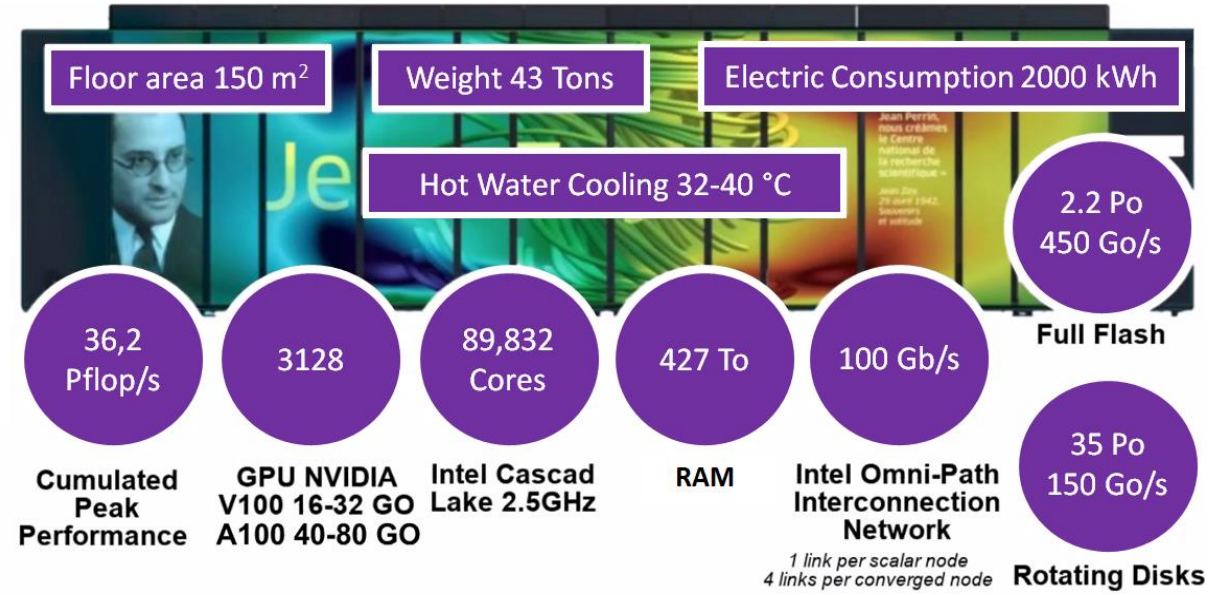



Figure 57: Jean Zay – First French converged supercomputer for Artificial Intelligence (AI) and High-Performance Computing (HPC). Updated figure sourced from https://www.irit.fr/wp-content/uploads/2022/01/20211215_Pres_Jean_Zay_IRIT.pdf

The preparation of complexes was conducted on our CPU cluster. Subsequently, the prepared complexes were sent to various GPU clusters to undergo preproduction tasks, including minimization, heating, and equilibration, followed by the production phase.

To ensure efficient data transfer, we utilized the "paramiko" Python library (278), which allowed us to set up a single SSH connection and transfer all files via scp, reducing the number of requests on different servers and streamlining the process.

Although Amber has been parallelized to run on multiple GPUs, we opted to utilize one GPU at a time to optimize the utilization of the allocated computational time, as running on several GPUs tends to be less efficient. A multitude of jobs were launched simultaneously, with as many as 500 jobs running concurrently on the Jean Zay computing cluster.

On the Jean Zay cluster, we utilized 150,000 hours of computation with V100 GPUs to conduct the MD simulations. On the Myria cluster, we employed 25,000 hours of computation with P100 GPUs. Additionally, our GPU cluster was heavily utilized for MD simulations.

The computational workload for MD simulations varied significantly depending on the protein's size within the complex. Here is a comparison of the time required for simulations on both small and large proteins using an RTX 2080 GPU:

- Small protein: For example, 6fap (50 amino acids), only necessitated one hour of computation.
- Large protein: On the other hand, for 1rbo (4,700 amino acids), the simulations required a more substantial 12 hours of computation.

We conducted MD simulations on 8,500 complexes, including 2,200 complexes with reconstructed loops. The simulations performed on these additional complexes were performed later in the PhD project, and thus, they were not integrated into the dataset utilized to train neural networks. Consequently, we extracted frames from a total of 63,000 simulations and computed the pockets, before creating a dataset tailored for use with DL models. Due to time constraints, we were unable to assess our models on the MD-augmented FEP dataset, even though we conducted 115 simulations out of the 200 complexes in the dataset. For similar reasons, while we expanded the number of complexes with simulations in the PDBbind core set 2016 from 83 to 160 complexes, we could not evaluate the performance of our models on this updated version.

## 4.4 DL implementation:

### 4.4.1 Reproducibility

Similarly to the MD simulations, we employed DL algorithms extensively across various environments, including other laboratory computers, our GPU cluster, and the Jean Zay cluster. To ensure reproducibility across various environments, multiple tools were employed:

- Anaconda (302) is a package management software that allows the creation of Python environments containing specific Python packages required to run specific software. To facilitate the replication and execution of an environment on another computer, it is possible to list the packages and their respective versions within a specific environment. However, this may not always guarantee that the code will run smoothly on different environments.
- Docker (303) is a tool for creating containers that encapsulate an entire environment. It installs a Linux system within the container, making it independent of the host operating system (OS). While this independence allows code to run on most computers, it has some limitations, such as reduced communication with GPU. Nvidia Docker can help overcome these limitations by enabling communication between the software, the CUDA toolkit (Nvidia's proprietary GPU communication toolkit), and the GPU/driver (Figure 58). However, running Docker containers on computing clusters with GPUs can still be challenging, as the GPUs are not located on the master node where the code and environment are situated.
- Singularity (304) is a container solution designed for scientists to run applications in high-performance computing (HPC) systems, including computing clusters with workload managers like SLURM (305). Docker containers can be easily transformed into Singularity containers, making them suitable for deployment in HPC environments.

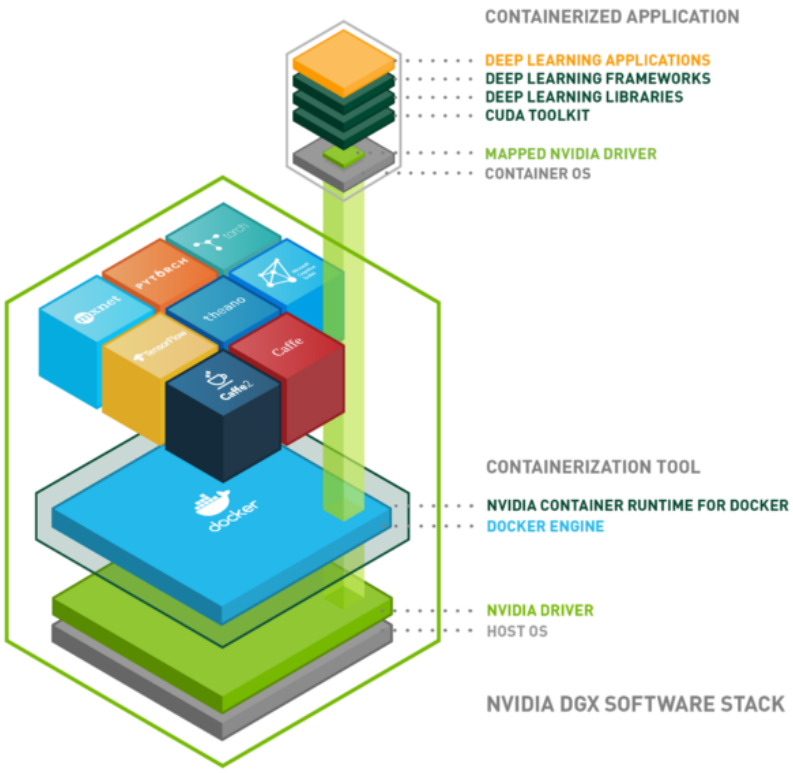



Figure 58: Nvidia Docker to allow software inside containers to communicate with GPU. Figure taken from Nvidia website (306).

We created a Docker container to execute Pafnucy on multiple computers within our laboratory and at the Janssen laboratory. Given that our laboratory's GPU cluster is not connected to internet for security purposes, traditional environment installation via Anaconda was infeasible. Thus, singularity containers were developed to run Pafnucy on our GPU cluster.

### 4.4.2 Tools to facilitate development

The development of the neural networks presented in this research was made possible through collaboration with the advanced support team at Jean Zay. Two engineers, Camille Parisel and Maxime Song, played a crucial role in implementing Proli, Densenucy, Timenucy, and Videonucy. To facilitate collaborative work, the code was shared on Github (307).

In addition, we resorted to using several tools to simplify the developments of DL algorithms. One such tool is Hydra (308), a python framework that enables the creation of preset configuration files tailored for specific DL experiments. These configuration files, in the form of .yaml files, contain a list of settings that are likely to be modified when using DL, including hyperparameters (Figure 59). This design allows for easy customization and the execution of DL experiments under different conditions.

MD_ConvLSTM > proli > configs > ! default.yaml

```yaml
# hydra: #prevent hydra log from filing home
#   run:
#     dir: ${env:luh_CCFRSCRATCH}/outputs/${now:%Y-%m-%d}/${now:%H-%M-%S}

mlflow:
  # path to store mlflow experience
  path: /run/media/libouban/data2/code/logs
  # name of the mlflow experience
  run_name: replicate_1

io:
  # path to train, val and test dataset as h5py files
  input_dir: /mnt/data2/dataset/dynamic/single_frame2/datasets/CoG_12
  # path where are saved the trained model
  model_path: /run/media/libouban/data2/code/models
  specific_test_dir: /mnt/data2/dataset/dynamic/single_frame2/datasets/CoG_12

data:
  # distance between grid points in angstrom
  grid_spacing: 1.0
  # max distance from complex center
  max_dist: 12
  # transform data into voxel
  voxel: false

training:
  learning_rate: 1e-5 # learning rate
  weight_decay: 0.0001 # lambda for weight decay
  batch_size: 20 # batch size
  num_epochs: 200 # max number of epochs 200
  patience: 20 # early stopping
  only_test:

network:
  conv_channels: [ 19, 64, 128, 256 ] # number of filters in conv layers
  conv_kernel_size: 5 # patch size for convolutional layers
  pool_kernel_size: 2 # patch size for pooling layers
  dense_sizes: [ 16384, 1000, 500, 200 ] # number of neurons in dense layers
  drop_p: 0.2 # originally 0.5 # probability for dropout
  pretrained_path:

experiment_name: proli
```

Figure 59: Hydra configuration file used as preset for DL experiment.

We employed an additional tool, MLflow (309), to oversee the performance of the statistical models trained using deep learning. This tool enables straightforward comparisons between multiple runs of the same experiment (Figure 60 – A). Additionally, it offers interactive visualizations to monitor performance changes during the training process (Figure 60 – B).

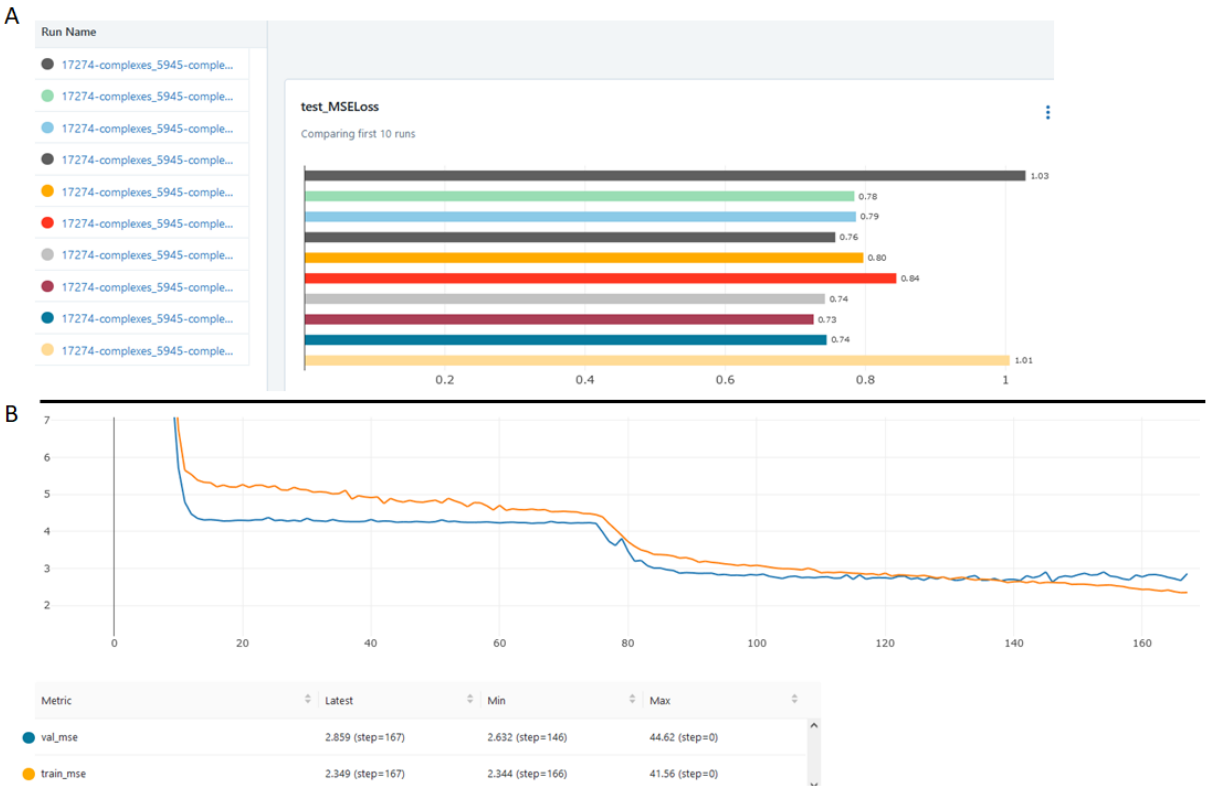



Figure 60: MLflow performance monitoring software. A – Comparison of different runs for various metrics. B – Evaluation of metrics throughout the training process.

### 4.4.3 Optimisation of DL

At the national computational center, we utilized approximately 35,000 hours of computation on V100 and 10,000 hours of computation on A100 for training deep learning models.

To reduce the computational cost of model training, we optimized their implementation on the national cluster, by considering various factors:

- **Batch Size Optimization:** One straightforward method is to increase the batch size, which speeds up training to a certain extent. However, this approach has some drawbacks, including increased RAM usage, which can lead to "out of memory" (OOM) issues as more data are loaded simultaneously. It may also impact model performance, so it is crucial to monitor metrics when adjusting the batch size.
- **Precision Considerations:** Another strategy to decrease computational time and reduce memory usage is to implement half-precision for model calculations. This entails utilizing 16-bit floating-point formats rather than the standard 32-bit format. However, this can sometimes result in poorer performance. As a compromise, employing mixed precision techniques can be an effective option, balancing computational efficiency and model accuracy. Nonetheless, we did not implement this approach in our work.
- **Data Parallelization:** Data parallelization involves training models across multiple GPUs simultaneously. However, this approach requires careful consideration, as communication between GPUs can sometimes take longer than the actual GPU computation, especially when using GPUs from multiple nodes. Furthermore, data parallelization increases batch sizes, which can affect model performance, necessitating performance monitoring. In our work, we parallelized Densenucy, Timenucy, and Videonucy.

- **Data Loading:** Before GPU computation, data is loaded into memory through the CPU. During this step, on-the-fly data augmentation can be performed, such as image rotation. Profiling tools should be used to ensure that the data loader does not become a bottleneck in comparison to GPU computations, especially when employing data parallelization. Various data loader settings can be adjusted to optimize computational performance, including settings like `persistent_worker`, `prefetch_factor`, `pin_memory`, and `non_blocking`. However, it is important to note that each setting comes with potential drawbacks, such as increased RAM usage.
- **CPU Allocation:** When launching jobs with SLURM, it is possible to allocate more or fewer GPUs and CPUs for a job. Increasing the number of CPUs can boost data loading speed and increase available RAM. The ideal number of CPUs per GPU may vary depending on the cluster's structure. For example, on Jean Zay, different GPU partitions have varying CPU-to-GPU ratios:
    - V100-16g / V100-32g: 612 nodes with 192GB RAM and 4 V100 GPUs per node (16 GB / 32 GB memory per GPU) with 10 CPU cores per GPU.
    - gpu_p2 (AI dedicated partition): 31 nodes with 384/768 GB RAM and 8 V100 GPUs per node (32 GB memory per GPU) with 3 CPU cores per GPU.
    - gpu_p4: 3 nodes with 768 GB RAM and 8 A100 GPUs per node (40 GB memory per GPU) with 6 CPU cores per GPU.
    - gpu_p5: 52 nodes with 512 GB RAM and 8 A100 GPUs per node (80 GB memory per GPU) with 8 CPU cores per GPU.

  Understanding the cluster's structure and optimizing CPU-GPU allocation can help make the most of its computational power. In specific situations, it might be necessary to utilize a higher ratio of CPUs per GPU than the one recommended in order to address and prevent RAM out-of-memory (OOM) errors effectively. However, this adjustment will lead to increased consumption of the allocated GPU computational time, even if the number of GPU used is unchanged. We also encountered OOM errors in GPU memory when using spatio-temporal learning methods. To address this issue and prevent OOM errors, we had to utilize GPUs with larger memory capacity, specifically GPUs with at least 32 GB of memory.
- **Data Storage Considerations:** Storing data in locations easily accessible by the GPU can help reduce training time. On Jean Zay, it is recommended to store data in the SCRATCH disk space due to its better bandwidth. However, it is important to be aware that data stored in the SCRATCH is automatically removed after one month.

When it comes to enhancing the performance of deep learning models, several variables can be fine-tuned. Firstly, selecting the appropriate optimizer for backpropagation is crucial. Various optimization methods have been proposed, including stochastic gradient descent, Adam (310), AdamW (311), and AdaGrad (312). Additionally, regularization techniques, such as weight decay, dropout, and L1/L2 regularization, help for mitigating overfitting.

Furthermore, optimizing hyperparameters can significantly enhance the performance of DL algorithms. These hyperparameters encompass factors like the number of layers, the number of nodes per layer, and the learning rate (LR). In particular, the LR warrants careful tuning as it controls the magnitude of weight updates during backpropagation, effectively determining the size of steps taken in stochastic gradient descent. To illustrate, you can think of it as akin to skiing down a mountain (Figure 61): A higher LR equates to faster skiing, which means reaching the bottom of the mountain quickly (*i.e.*, faster model training). However, there is a risk of overshooting and heading up the

opposite side, resulting in lower performance. Conversely, a smaller LR (skiing more slowly) prolongs training time and may lead to getting stuck in a local minimum.

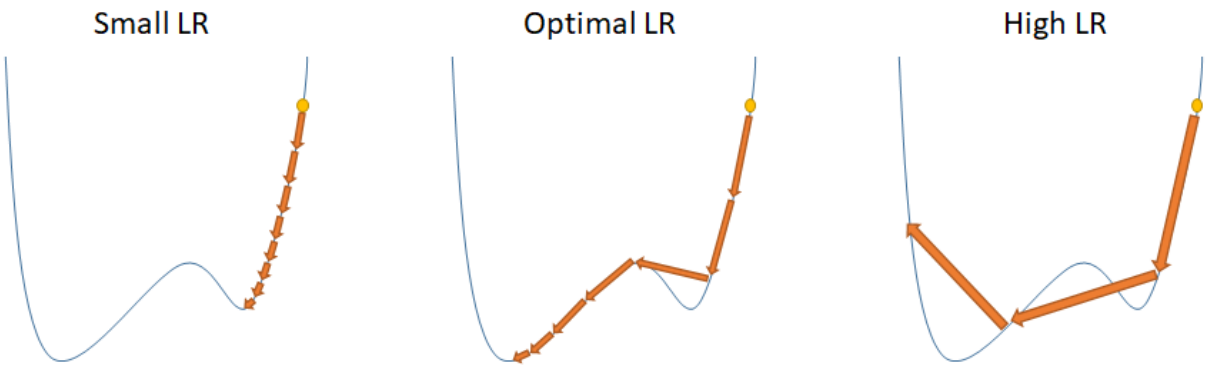



Figure 61: Stochastic gradient descent in function of learning rate (LR).

To address these issues, one can employ a LR scheduler. LR schedulers adapt the learning rate based on the training progress, starting with a high LR to accelerate learning and then gradually decreasing it to achieve better performance. However, using a high LR from the beginning can lead to instability in the early stages of training. To mitigate this, it is recommended to use a warm-up LR, which gradually increases the LR to the desired value.

Another strategy to enhance performance is the use of cyclic LR, particularly the one-cycle LR policy. This approach helps models escape local minima during training (Figure 62). It works by having spike of high LR, followed by epochs with smaller LR. An example of such a LR scheduler was implemented in Timenucy to improve its performance.

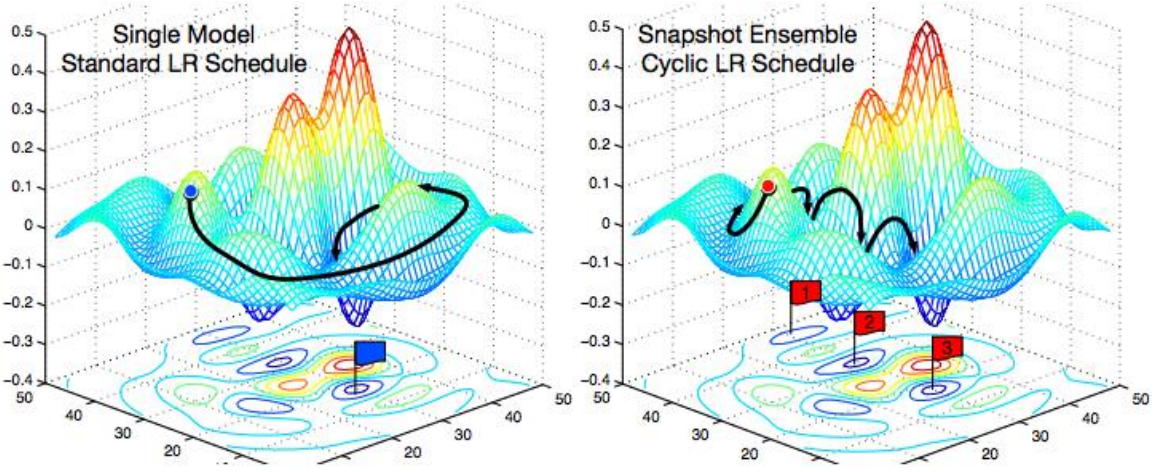



Figure 62: Application of cyclic LR schedule in comparison to standard LR schedule. Figure taken from Huang *et al.* (313).

Thereafter, we conducted additional investigations that were not included in the publication on the combination of MD and DL. These include a preliminary study of complexes stability in function of their affinity, additional results from deep neural networks, and ongoing work related to data selection and representation to enhance model performance.

## 4.5 Additional investigations:

### 4.5.1 Stability of ligands across simulations

Early in the project, we expected that the neural networks would benefit from the temporal information available in MD simulations. We anticipated several potential improvements associated with MD data, including a better understanding of molecular interactions and the ability to gain insights from ligand behaviour within binding pockets. Therefore, we evaluated the ligand RMSD during simulations to assess whether the ligand behavior within the pocket would be a relevant information to take into account by neural networks. Our inspiration for this approach stemmed from the work of Guterres *et al*. (314), who effectively employed ligand RMSD during simulations to differentiate active molecules from decoys. This methodology was successfully applied in a virtual screening context to rank molecules, as depicted in Figure 63.

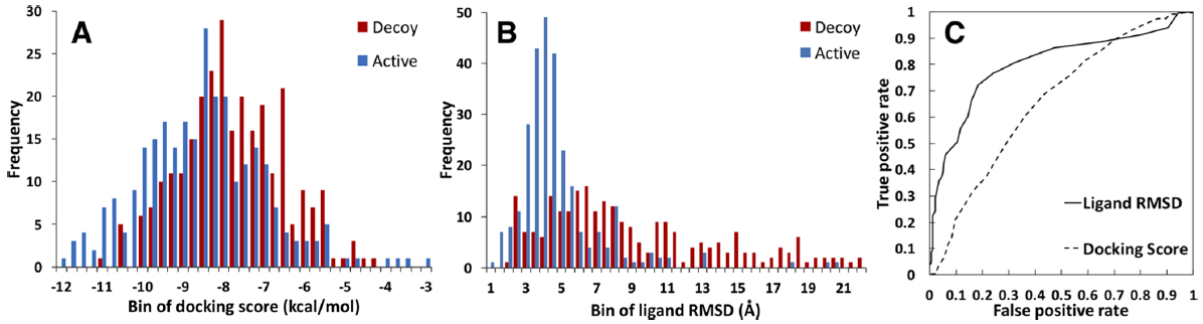



Figure 63: Histogram distributions of (A) docking scores and (B) ligand RMSD, with decoy compounds displayed in red and active compounds in blue. (C) ROC plot comparing docking scores with ligand RMSDs obtained from MD simulations. The AUC values are 0.683 for the docking score and 0.832 for the ligand RMSD. Figure taken from Guterres *et al.* (314).

We hypothesized that low-affinity complexes would tend to exhibit instability during simulations. Such information could be used by the neural networks to carry out predictions. In Figure 64, we display the mean ligand RMSD across 10 simulation replicates in function of the $pK_i$ for all complexes. In line with the approach by Liu *et al*. (315), we applied a threshold of 2.0 Å mean ligand RMSD to distinguish stable complexes from unstable ones. To compute this value, we superposed the protein across all simulation frames. As a result, it appears that there are as much stable complexes than unstable. Although only 36% of low-affinity complexes (< 4 $pK_i$) were stable, while this percentage increases to 58% for high-affinity complexes (> 8 $pK_i$). There appears to be a trend wherein complexes with higher affinity tend to display greater stability during simulations.

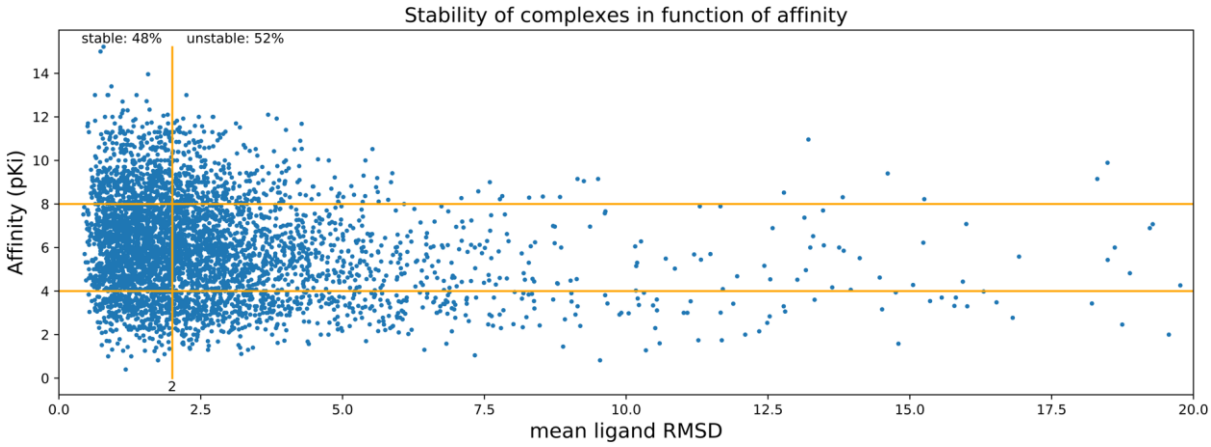



Figure 64: Stability of complexes during the simulations in function of their affinity. The stability of complexes is assessed by measuring the mean ligand RMSD across the 10 simulation replicates. A stability threshold of 2.0 Å for the maximum ligand RMSD is applied to distinguish between stable and non-stable complexes (315). The plot provides the percentage of complexes that fall within or outside this stability limit. Additional thresholds were introduced at 4 and 8 $pK_i$ to distinguish between low and high-affinity complexes. Please note that the plot does not display complexes with a mean ligand RMSD exceeding 20 Å.

Upon closer examination of the simulations, we observed that low-affinity ligands infrequently exited the binding pockets across the 10 simulation replicates, especially in the PDBbind core set 2016. Consequently, we conducted an analysis of the maximum ligand RMSD across these replicates. We focused our analysis to the PDBbind core set 2016 and investigated the stability of complexes within each cluster in function of their affinity (Figure 65). We observed a trend where low-affinity complexes tend to have more unstable ligands. While stability levels may vary among clusters, in most cases, the complexes with the lowest affinity within a cluster also exhibited the greatest instability.

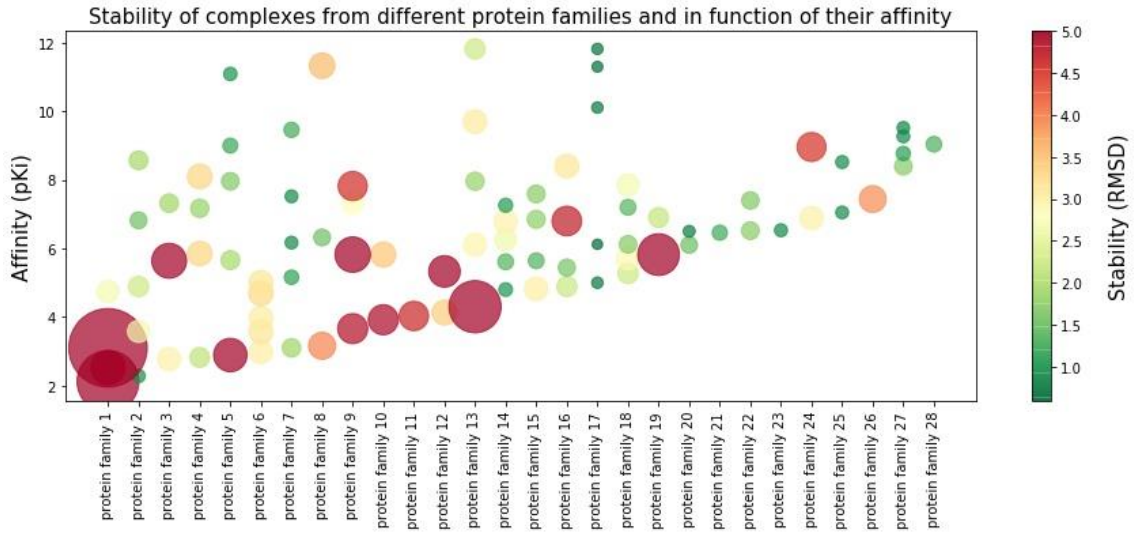



Figure 65: Distribution of the ligand stability (max ligand RMSD) in function of the affinity for each cluster. Both the size and colour of dots are influenced by the ligand stability. For the colour we set a limit at 5.0 Å, everything over that number have the same color.

Figure 66 provides a visual representation of the maximum ligand RMSD poses obtained during the simulations. Specifically, it showcases the lowest and highest affinity complexes from the CGMP 3',5'-CYCLIC PHOSPHODIESTERASE cluster within the PDBbind core set 2016. We observe that the ligand has exited the binding site for the low-affinity complex (Figure 66 – A), while there have been minimal changes in ligand positions for the high-affinity complex (Figure 66 – B).

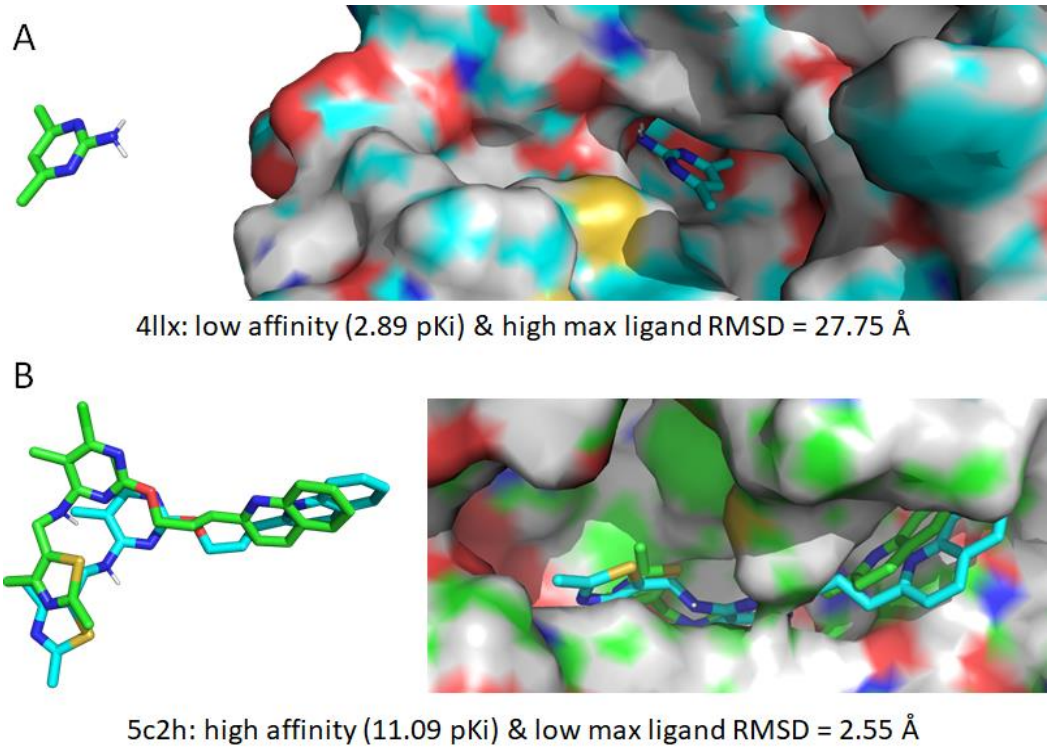



Figure 66: Visualisation of the max ligand RMSD pose of (A) low and (B) high affinity complexes. In these visualizations, the crystallographic poses are displayed in cyan, while the max ligand RMSD poses are shown in green. Both complexes are from the same PDBbind core set 2016 cluster: CGMP 3',5'-CYCLIC PHOSPHODIESTERASE.

As described in the publication, Timenucy and Videonucy demonstrated improved performance when trained exclusively on the ligand, using a box centred on the binding pocket rather than on the ligand itself. This outcome suggests that these models have the capability to extract valuable information from the ligand's movement. This is an encouraging finding that warrants further in-depth investigation and exploration.

### 4.5.2 Evaluation of models' performance

We quickly raised the question of identifying the complexes for which the current models struggled to accurately predict binding affinities. Ultimately the goal was to determine whether we could improve the prediction of binding affinities for such complexes by developing our own models. We hypothesized that since other models struggle with assessing molecular interactions, they would face challenges in predicting affinities in the case of activity cliffs. To assess if our models have a better understanding of molecular interactions, we aimed to determine if they performed better in predicting such cases.

Our initial investigation focused on identifying activity cliffs within the clusters of the PDBbind core set 2016. To assess the similarity of molecules within each cluster, we utilized ECFP (57) and MACCS fingerprints on the molecules and subsequently calculated the Tanimoto coefficient between the ligands with the lowest and highest affinity within the same cluster. Then, we selected the complexes that exhibited the highest degree of similarity among them (Figure 67) and compared Pafnucy and Densenucy's performance on these complexes. Pafnucy's predictions were obtained by using the model published in reference (3), whereas for Densenucy predictions, we employed a model trained with MD data augmentation. Results for selected complexes:

- MTA/SAH NUCLEOSIDASE: Real $\Delta pK_i$ = 6.82, Pafnucy $\Delta pK_i$ = 0.76, Densenucy $\Delta pK_i$ = 3.3 (Figure 67 – A).
- HEAT SHOCK PROTEIN HSP82: Real $\Delta pK_i$ = 3.46, Pafnucy $\Delta pK_i$ = 0.53, Densenucy $\Delta pK_i$ = -0.44 (the negative value is due to 2vw5 predicted with higher affinity than 2yge) (Figure 67 – B).
- CATECHOL O-METHYLTRANSFERASE: Real $\Delta pK_i$ = 4.87, Pafnucy $\Delta pK_i$ = 1.12, Densenucy $\Delta pK_i$ = 1.36 (Figure 67 – C).

While Densenucy significantly improved the prediction for MTA/SAH NUCLEOSIDASE, its overall impact on activity cliffs remains uncertain. Therefore, as outlined in the publication, we evaluated models on the FEP dataset to gain a more comprehensive understanding of how different neural networks perform in scenarios involving activity cliffs. Notably, Densenucy with MD data augmentation displayed promising performance. However, it is important to approach the results on the FEP dataset with caution, as there is a lack of complexes with extreme affinities in this dataset.

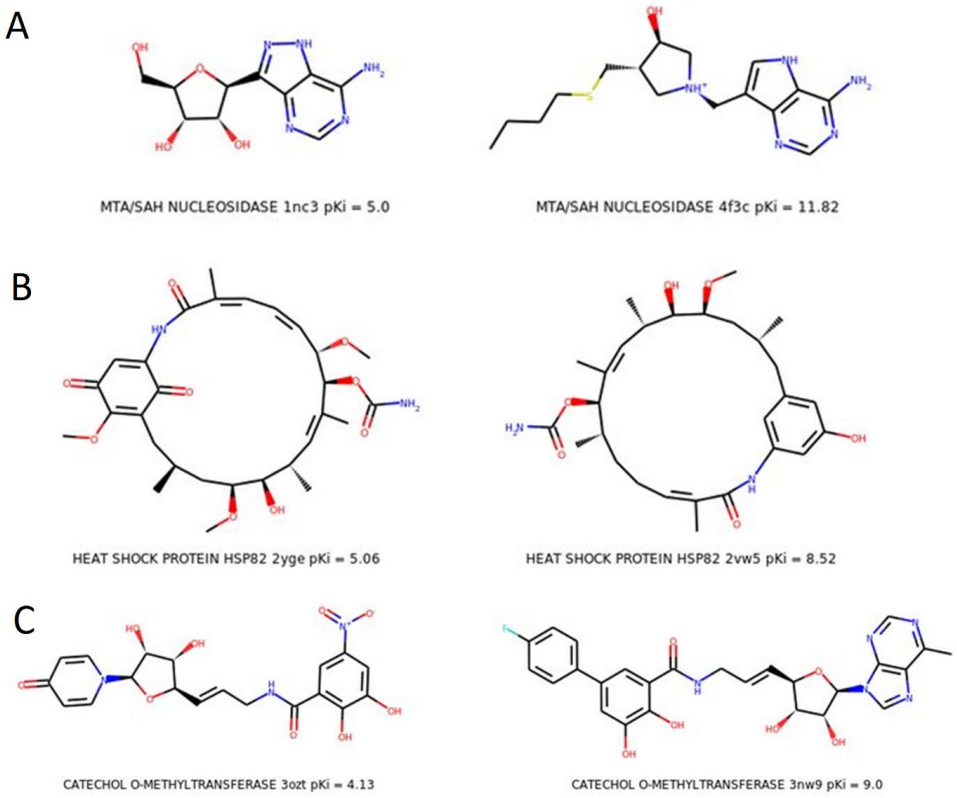



Figure 67: Activity cliff inside the PDBbind core set 2016. A – MTA/SAH NUCLEOSIDASE: $\Delta pK_i$ = 6.82, Tanimoto coefficients - ECFP: 0.26, MACCS: 0.59. B – HEAT SHOCK PROTEIN HSP82: $\Delta pK_i$ = 3.46, Tanimoto coefficients - ECFP: 0.46, MACCS: 0.82. C – CATECHOL O-METHYLTRANSFERASE: $\Delta pK_i$ = 4.87, Tanimoto coefficients - ECFP: 0.45, MACCS: 0.68.

Thereafter, we compared the distributions of the prediction error of Pafnucy, Octsurf and Graphbar (Figure 68). A consistent trend emerged across these models: they tended to overpredict affinities for complexes with low binding affinity and underpredict affinities for complexes with high binding affinity. This trend was especially evident when examining the difference between the real and predicted affinities. This observation aligns with findings by Volkov *et al*. (9), which suggested that models often predict values centred around the mean affinity value of the training dataset.

While complexes with high and low affinity values currently pose the most significant prediction challenges, as previously demonstrated. These complexes tend to exhibit significantly higher or lower stability, respectively. Moreover, our results suggest that spatio-temporal learning models incorporate information from ligand movement during simulations to make predictions. Anticipating further research and the development of enhanced models capable of leveraging complex stability for improved predictions, we can reasonably foresee substantial performance improvements, particularly for complexes with extreme affinities.

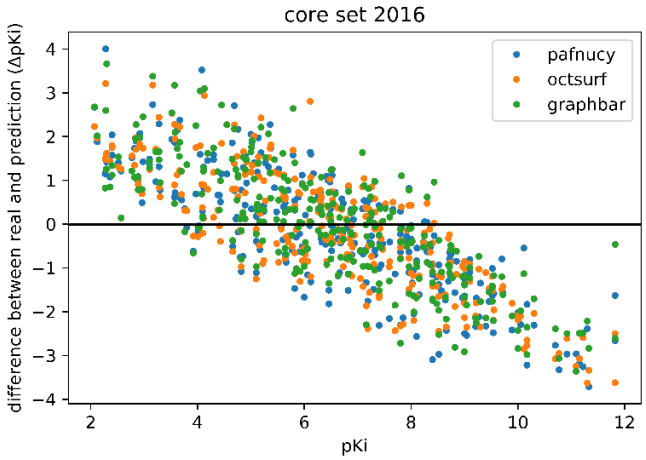



Figure 68: The distribution of the error in prediction ($\Delta pK_i$) for Pafnucy, Octsurf and GraphBAR on PDBbind core set 2016.

In the dataset consisting of 285 complexes from the PDBbind core set 2016, there are several complexes that consistently receive poor predictions from Pafnucy, Octsurf, and GraphBAR. Specifically, when we examine the 15 worst-predicted complexes, we find that the following 6 complexes are common to all three models: 2x00, 4tmn, 3ozt, 4gid, 5c2h, and 2ymd. Additionally, Pafnucy and OctSurf share 4 other complexes within their top 15 worst predictions: 3myg, 1mq6, 3utu, and 1ps3. This could be attributed to the fact that both Pafnucy and Octsurf are CNN, while GraphBAR is a GNN.

When we evaluate the performance of Densenucy with MD data augmentation on these 6 complexes, we observe that the prediction for 3ozt significantly improved by one log compared to Pafnucy, and the prediction for 2x00 improved by two logs.

We applied the same evaluation protocol to the 83 complexes from the PDBbind core set 2016 used to assess spatio-temporal models. Out of the 10 complexes with the worst predictions from Pafnucy, Octsurf, and Graphbar, Videonucy demonstrated an enhancement in performance for the complex 2ymd, surpassing Pafnucy's prediction by one log.

Subsequently, we computed the average $\Delta pK_i$ by protein, focusing on the 57 clusters within the PDBbind core set, each comprising 5 complexes. Our aim was to identify the complexes with the poorest performance. We specifically looked at the top 10 protein families with the worst predictions for Pafnucy, Octsurf, and GraphBAR, and identified the 5 clusters that were common among them: MTA/SAH NUCLEOSIDASE, ACETYLCHOLINE-BINDING PROTEIN, ESTROGEN RECEPTOR, ACETYLCHOLINE RECEPTOR, and HIV-1 INTEGRASE.

Densenucy exhibited performance improvements for 2 of these protein families: ESTROGEN RECEPTOR and ACETYLCHOLINE RECEPTOR. In both cases, Densenucy achieved an improvement of more than half a logarithm compared to Pafnucy.

In Figure 69, we have depicted the distribution of predictions for MD data augmentation models and compared them to Pafnucy. The experimental binding affinity values are shown in red. Both Proli (Figure 69 – A) and Densenucy (Figure 69 – B) exhibit similar trends to Pafnucy. However, Densenucy predictions span from 3 to 10 $pK_i$, with notable improvements for extremely high-affinity complexes in comparison to Proli/Pafnucy.

A similar comparison was carried out for spatio-temporal models in Figure 70. We note that Timenucy's predictions cover a range of 2-3 logarithms of $pK_i$ (Figure 70 – A). Timenucy seems to struggle even more than Pafnucy at predicting extreme binding affinities. On the other hand, while Videonucy performs a bit better than Timenucy on the extremes, it does not quite match the performance of Pafnucy (Figure 70 – B).

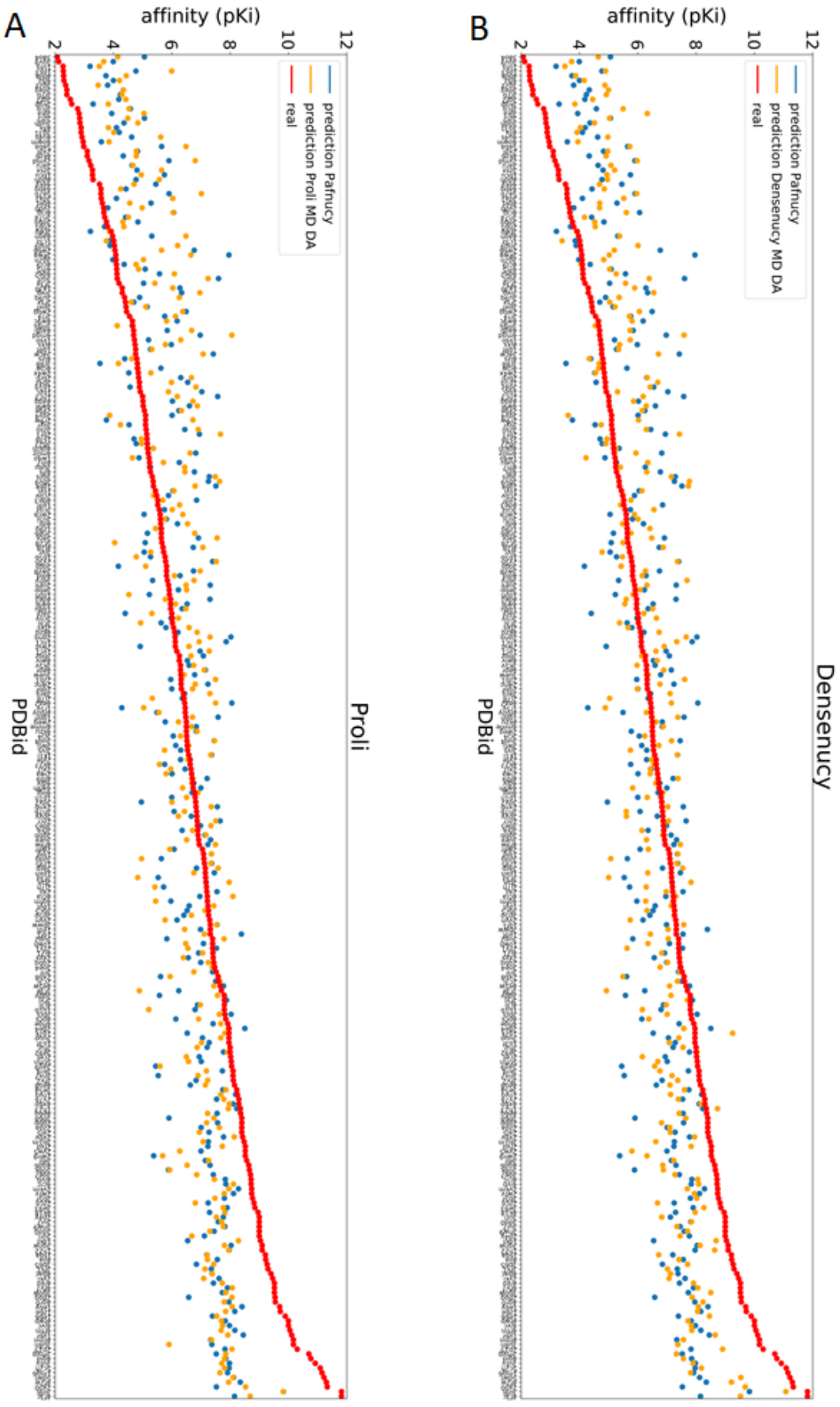



Figure 69: Distribution of the prediction of MD data augmentation models compared to Pafnucy's prediction. A – Proli MD data augmentation. B – Densenucy MD data augmentation.

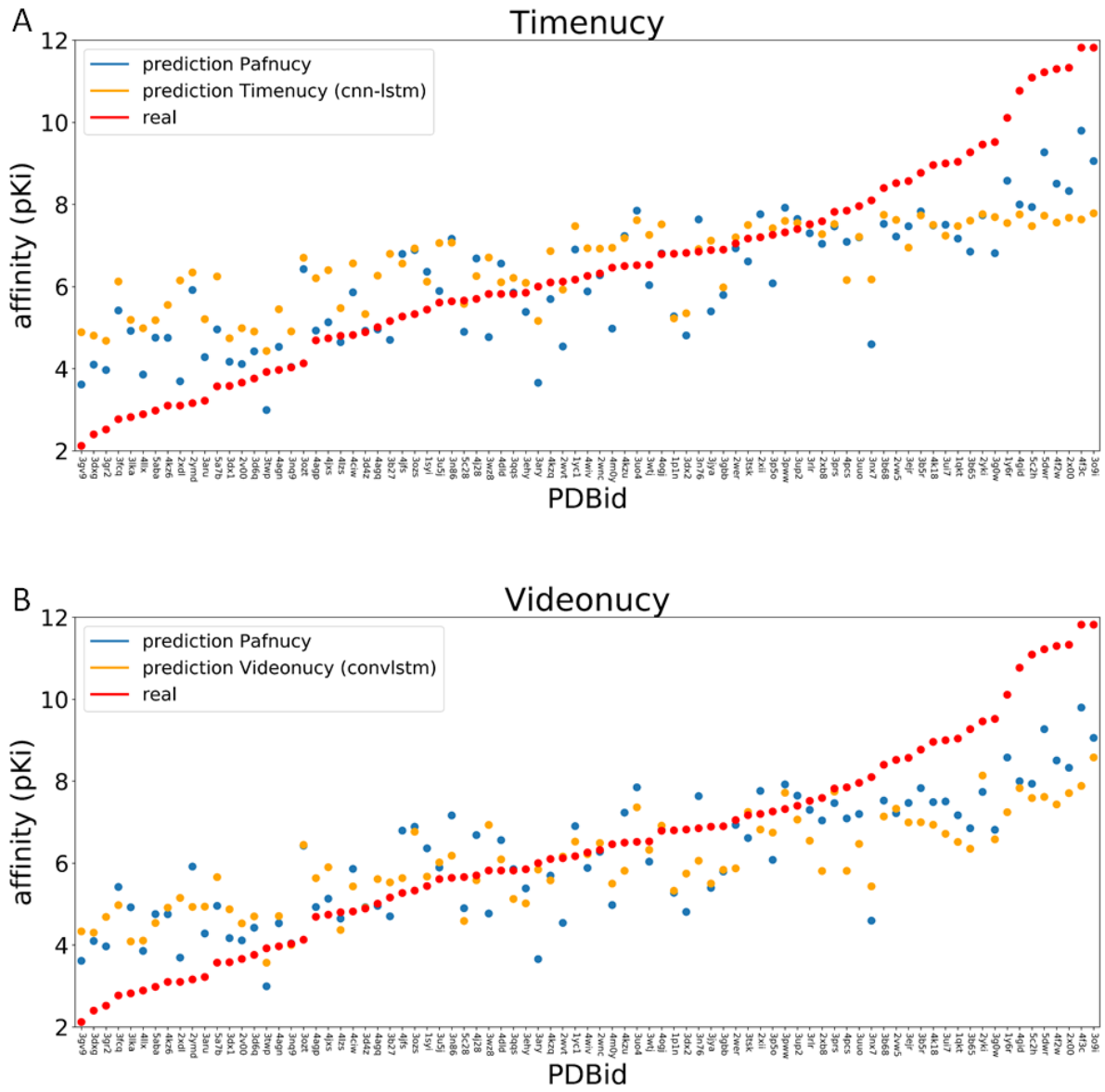



Figure 70: Distribution of the prediction of spatio-temporal learning models compared to Pafnucy's prediction. A – Timenucy. B – Videonucy.

As previously mentioned, it is also possible to carry out predictions on frames with Pafnucy, Proli and Densenucy. Further assessments were carried out to gauge the performance of neural networks when applied to crystallographic poses or frames extracted from simulations. Densenucy's results are presented in Figure 71. We employed consensus methods over 10 model replicates. In the last column, we averaged the predictions for all frames within each complex. Notably, Densenucy demonstrated improved RMSE on the reduced test set of 83 complexes when predicting on frames, whether the predictions were averaged or not for each complex.

Figure 71: Performance of Densenucy on the PDBbind core set (285 complexes) and reduced version of 83 complexes used to evaluate spatio-temporal learning methods. On the first column, predictions are carried on crystallographic poses, while the predictions are carried on the crystallographic + frames (41,100 structures in total when applied on 285 complexes) on the second column. Finally, predictions were averaged for each complex (average over frames) in the last column. In all cases, the results displayed are consensus methods over 10 models.

Following this, we designated the test set composed of only of frames as the “MDbind test set”. It comprises frames extracted from simulations involving 83 complexes from the PDBbind core set 2016, totalling 41,500 frames for assessing models trained using MD data augmentation, as well as 830 simulations for evaluating spatio-temporal models.

As previously discussed in the publication, using MD data augmentation models without consensus methods across replicates, it is possible to achieve better performance on the MDbind test set by averaging the predicted values across simulations and complexes, rather than computing the metrics based on the 41,500 individual frames (Figure S3 in supporting information). Therefore, there might be potential benefits in conducting a simulation on a complex, followed by predicting binding affinity for all the frames of the simulation and averaging their predictions to achieve more accurate predictions.

Moreover, it seems that employing either a consensus method across models or averaging predictions over the frames results in a comparable performance enhancement. The combination of both approaches yields only a marginal performance gain compared to using either method individually.

The impact of conducting multiple simulations, as opposed to a single one, yielded varying results depending on the neural network employed. In the case of Densenucy, the performance remained consistent when assessed on each simulation individually. However, for Pafnucy and Proli, we observed a correlation coefficient discrepancy of up to 0.02 and a RMSE gap of 0.05 between two simulations.

Subsequently, we compared the performance of MD data augmentation (MD DA) and spatio-temporal models on the reduced PDBbind core set 2016 and MDbind test set (Table 3). The performance on the MDbind test set is obtained by averaging predictions from each frame of a complex for MD augmentation models and from each simulation of a complex for spatio-temporal models. We note that MD DA models clearly outperformed spatio-temporal models. Additionally, we observed significant improvements when predicting on the MDbind test set for MD DA models. For instance, Pafnucy MD DA achieved a 0.16 RMSE reduction when applied to that test set. This suggests that models tend to perform better when making predictions on the same type of data they were trained on. For instance, training on frames results in improved predictions on frames. This observation aligns with other research (316) that found enhanced performance in ranking docking poses when models were trained on docking poses rather than crystallographic poses.

| | Pafnucy MD DA | | Proli MD DA | | Densenucy MD DA | | Timenucy | | Videonucy | |
|---|---|---|---|---|---|---|---|---|---|---|
| | R | RMSE | R | RMSE | R | RMSE | R | RMSE | R | RMSE |
| Reduced PDBbind core set 2016 (83 complexes) | 0.78 | 1.68 | 0.79 | 1.64 | **0.86** | **1.43** | NA | NA | NA | NA |
| MDbind test set + averaged over frames/simulations | 0.81 | 1.52 | 0.81 | 1.45 | **0.85** | **1.34** | 0.78 | 1.78 | 0.84 | 1.66 |

Table 3: Comparison of models trained with MD data augmentation (MD DA) and spatio-temporal models. Results are evaluated on the reduced PDBbind core set 2016 (83 complexes crystal structure) and on the MDbind test set (41,500 frames or 830 simulations). The displayed results employ a consensus method across 10 model replicates. Additionally, predictions on the MDbind test set are averaged over frames for MD DA models and simulations for spatio-temporal models.

In the case of spatio-temporal learning, the models were assessed on simulations from the MDbind test set, which comprises 83 complexes. Given that there are 10 simulations for each complex, we have presented the results for each simulation individually (Figure 72) or by averaging the predictions for each complex (Figure 73). We have depicted the results using a consensus method across 10 model replicates.

When compared to predictions carried on the frames (*e.g.* Figure 71 for Densenucy), we can observe that the predictions on the simulations are more consistent, with smaller standard deviations (Figure 72). This is particularly noticeable with Timenucy, but it also holds true for MD data augmentation models when averaging predictions over the frames of each simulation (not shown here). This suggests that the information obtained from simulation replicates tends to be consistent and evaluating the entire simulation as a whole results in steady predictions compared to evaluating individual frames separately.

Videonucy demonstrates superior performance to Timenucy, particularly in terms of coefficient correlation and RMSE. While both models still exhibit limitations in terms of RMSE compared to other models, it is important to highlight that they were trained and validated on a dataset comprising 60,000 simulations derived from 6,000 complexes. However, this dataset, although substantial, is relatively small considering the inherent complexity of the data. It is worth noting that despite the significant number of available complexes in the latest release of the PDBbind (v.2020), which stands at approximately 19,400, there are still inherent limitations when it comes to effectively learning from 3D structures. As data complexity increases, more sophisticated neural networks are typically required for analysis, and this, in turn, necessitates even larger datasets for effective model training. Hence, it is anticipated that a larger volume of data would be required to effectively train spatio-temporal learning models and achieve improved performance.

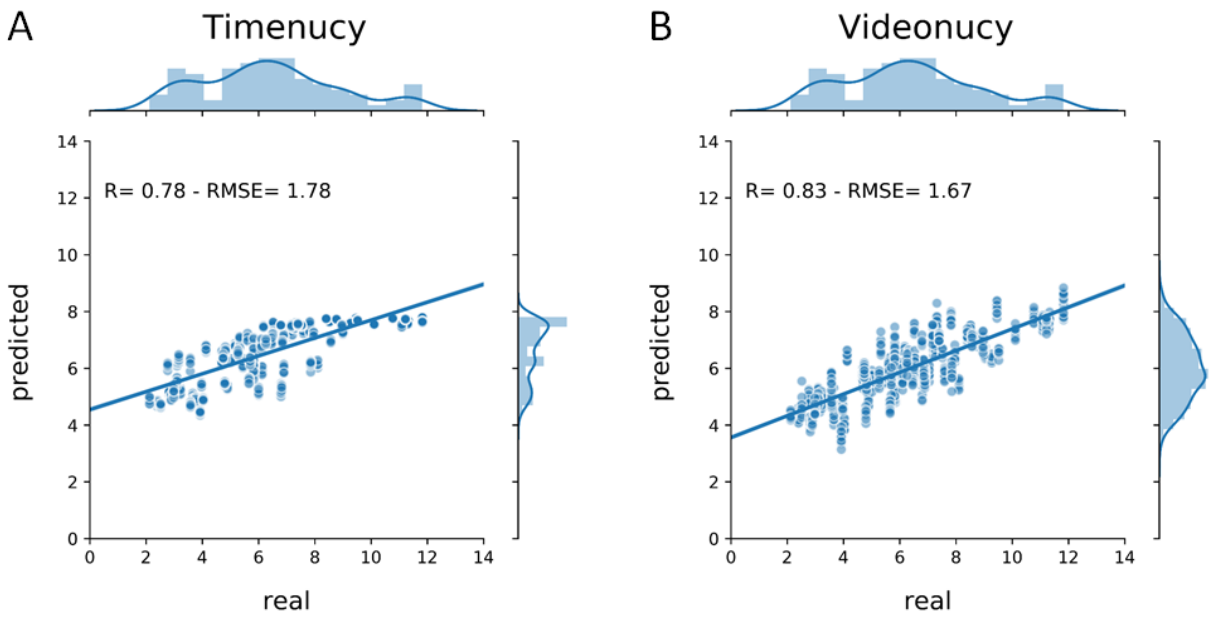



Figure 72: Performance of Spatio-temporal learning on the MDbind test set (830 simulations from 83 complexes). A – Timenucy. B – Videonucy.

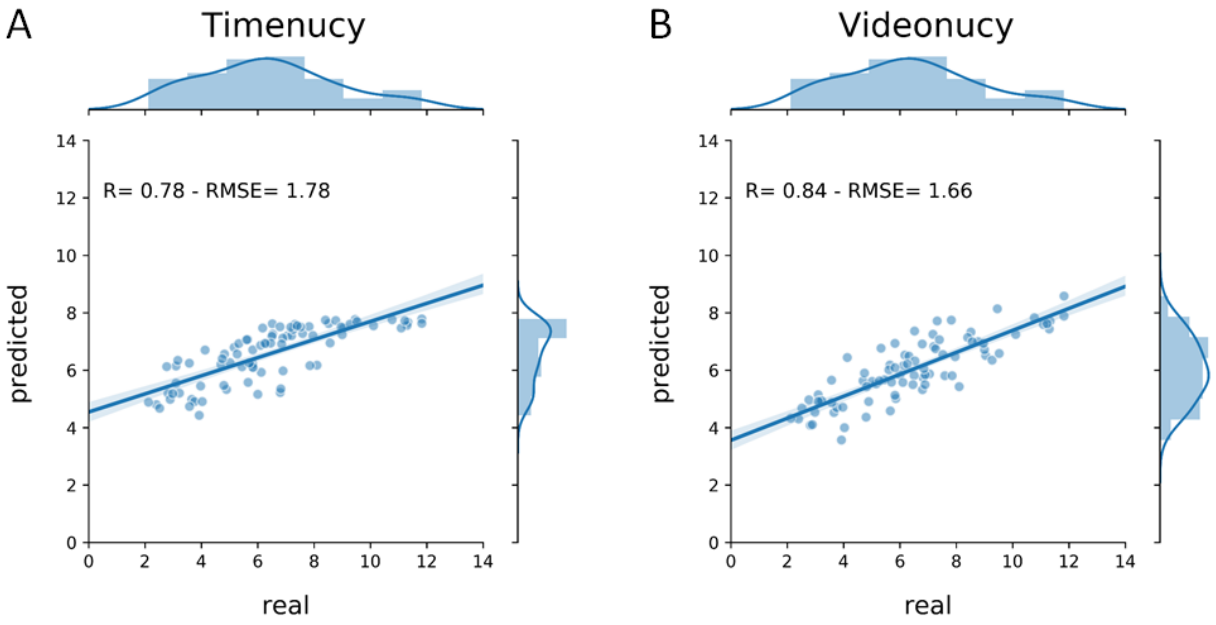



Figure 73: Performance of Spatio-temporal learning on averaged predictions for the MDbind test set (830 simulations from 83 complexes). Predictions were averaged for each complex (average over simulations). A – Timenucy. B – Videonucy.

Considering that spatio-temporal models constitute a pioneering neural network paradigm, there is substantial room for enhancement to achieve their optimal performance in predicting binding affinities. These models possess significant potential due to their ability to analyse ligand movements within binding sites, a feature that can greatly enhance their predictive capabilities, especially for complexes with high or low affinity.

### 4.5.3 Data curation and DL input representation

Several methods were investigated to try to improve models' performances by modifying the input data.

#### *4.5.3.1 Subsampled dataset*

As mentioned earlier, the PDBbind dataset exhibits significant redundancy especially in terms of protein families, which can introduce bias in training models. We also had similar concerns regarding the characteristics of the ligands in the dataset. To address these issues, we decided to create a reduced training set by implementing a clustering approach on the available data, selecting representative members to ensure a more balanced and less biased training dataset.

To perform this subsampling, we initially divided the dataset based on protein families. Subsequently, we applied clustering techniques on the following ligand descriptors:

- The affinity of complexes
- The molecular weight of ligands
- The principal moment of inertia (PMI) of ligands to account for their shape

Various clustering algorithms were applied, including K-means, K-medoids, and DBSCAN (124). Using the K-medoids method, we successfully downsized the MDbind dataset while preserving a similar distribution, as demonstrated in Figure 74. We retained clusters with more than 10 members and selected a representative member from each of these clusters, resulting in a subsampled dataset comprising 360 complexes.

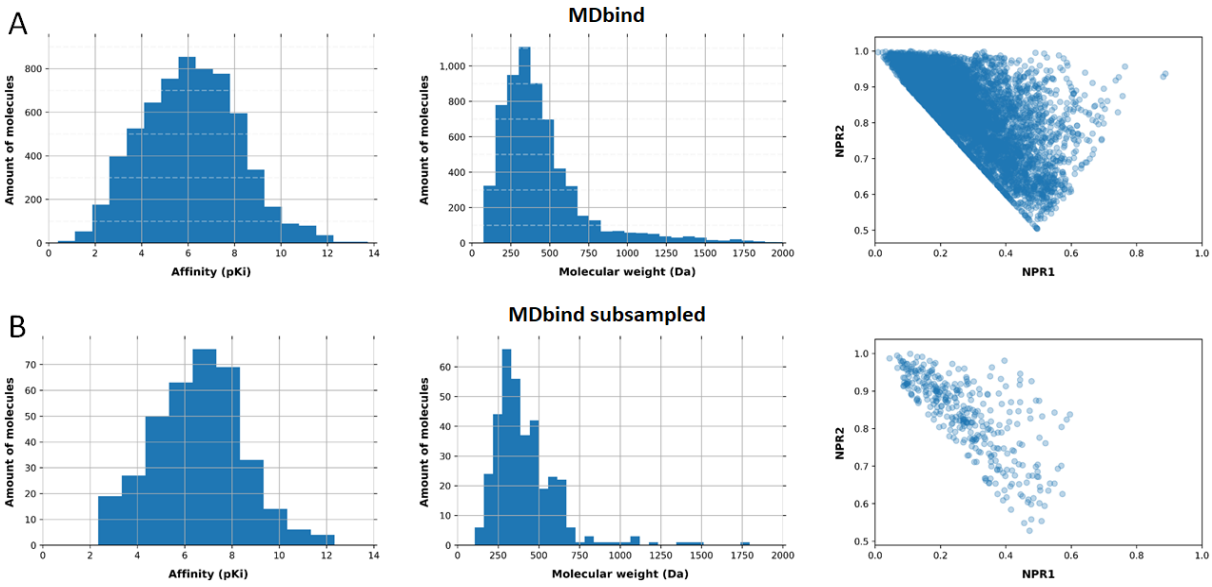



Figure 74: Distribution of the complexes in MDbind and the subsampled dataset from MDbind. A – MDbind is composed of 6,300 complexes. B – MDbind subsampled is composed of 360 complexes.

Within the subsampled dataset, the number of distinct protein families was reduced to 135. This clustering approach has led to a more balanced distribution of data among the different protein family clusters, as illustrated in Figure 75. However, it is worth noting that the HIV-1 PROTEASE protein family

still appears to be overrepresented. Therefore, some adjustments to the clustering methodology may be necessary to address this imbalance.

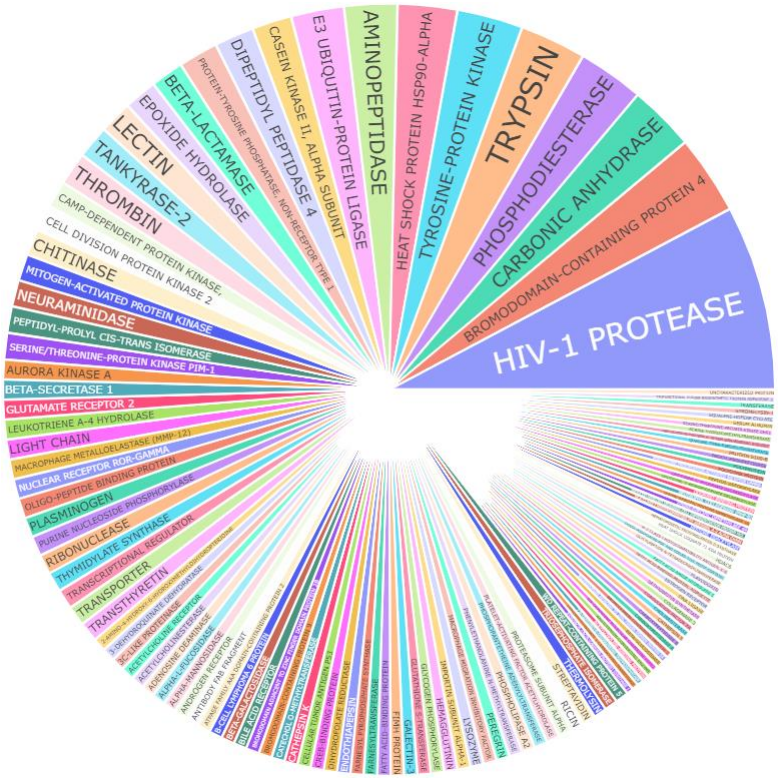



Figure 75: Sunburst displaying the protein family diversity inside the subsampled dataset.

Creating these subsets serves multiple objectives:

First, we aim to train models on the subsampled dataset to leverage its reduced redundancy, with the hope of obtaining models that are less biased and capable of better generalization. Notably, the DBscan clustering approach has yielded a subsampled dataset with a more evenly distributed range of affinity values. It will be interesting to determine whether models trained on this dataset still exhibit a bias toward predicting affinities mainly between 4-8 pKi.

Secondly, we want to assess whether our newly developed neural networks can achieve good performance when trained on a smaller dataset. We plan to compare the performance of these models to other neural networks trained on the same dataset, such as Pafnucy, Octsurf, or GraphBAR. While Densenucy may potentially achieve relatively good performance in comparison to the other neural networks, it seems unlikely that spatio-temporal models will perform well with such limited data, given the complexity of the data they are being trained on. To fully leverage the MD data, spatio-temporal models are expected to benefit significantly from accessing larger datasets than what is currently available in MDbind.

Finally, reducing the amount of data in the training set allows us to explore the use of systematic rotations for training neural networks. This approach will enable us to evaluate the impact of systematic rotations in combination with MD data augmentation on model performance.

Due to time constraints, we were unable to train models on this subsampled dataset, prompting the need for further investigations. Future research endeavours could explore developments similar to this study, focusing on evaluating the performance of MD data augmentation when training specialized models designed to predict affinity for specific protein families. This approach could prove beneficial in scenarios where the availability of data is limited, enabling the development of models that achieve improved performance.

#### *4.5.3.2 Selecting frame from simulations*

We also have initiated an investigation into frame selection in MD simulations. Up to this point, all the results presented were obtained by training neural networks on all the frames from the simulations. However, we are uncertain whether all these frames provide equally valuable information to the models. Indeed, it seems that many frames may be redundant and provide similar information. Hence, training models on a selection of frames to focus on the most relevant parts of the simulations could be beneficial for both models' performance and training speed.

In a preliminary study, we explored the training of models using various frame selection methods, such as choosing the initial and final frames of MD simulation or selecting frames at regular intervals (*e.g.*, every 10 frames). However, it is essential to employ a method to identify the most relevant frames, and one such approach is to cluster the frames and select a subset of representative ones. Initially, our approach has been to cluster the frames based on the ligand RMSD. We applied different clustering techniques to the 3D positions of ligands using Pytraj (298) and MDAnalysis (280), including KMeans, DBscan (124), and affinity propagation (126).

We achieved promising results when clustering the frames of the simulation from a low affinity (1.3 $pK_i$) stable complex (1a0t) using DBscan (Figure 76). DBscan effectively grouped the most similar ligand poses together, as indicated by the orange frames from frame 6 to 38 in Figure 76 – A, and represented as crosses in Figure 76 – B. The remaining frames, located at the beginning and at the end of the simulation, were grouped together in a "garbage" cluster.

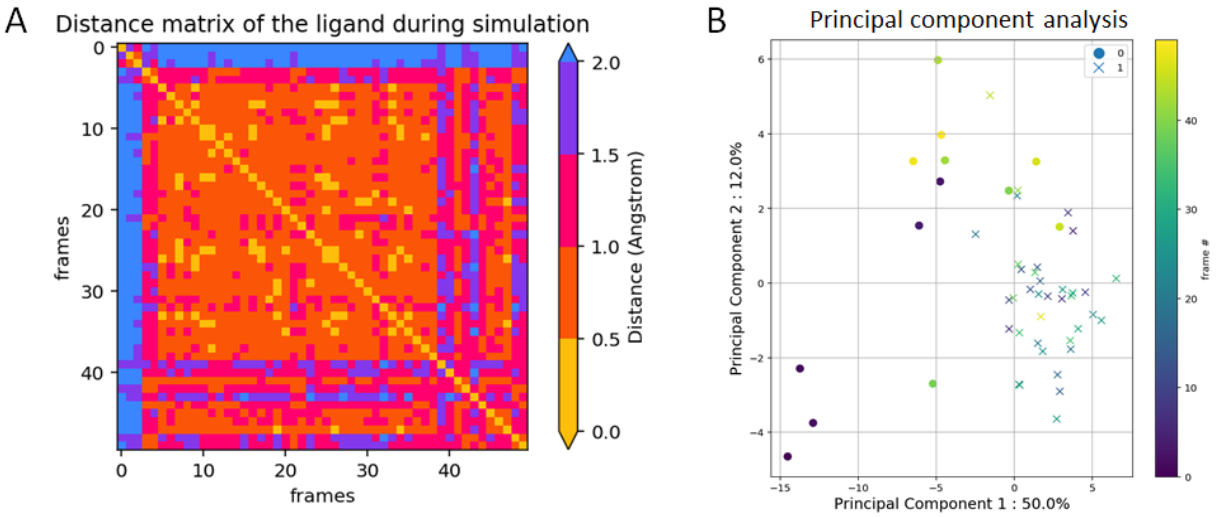



Figure 76: Selection of frames from a MD simulation of a stable complex (1a0t). A – Visualisation of the distance matrix based on ligand RMSD. The colour is used to display the ligand RMSD across all the frames of the simulation. B – Principal component analysis of the cluster obtained with DBscan on that simulation. The colour gradient applied indicate the frame number.

In our exploratory efforts, we also attempted to cluster frames based on their protein-ligand interaction fingerprints, including the structural protein–ligand interaction fingerprints (SPLIF) (71) and the protein-ligand extended connectivity (PLEC) fingerprint (74).

Further work is necessary to apply these methods on a larger scale. A comprehensive evaluation of clustering tools designed for grouping conformations within MD simulations is warranted. Notable approaches, such as those based on self-organized maps, hold promise in this regard (317). Additionally, it would be valuable to find or develop a tool capable of creating clusters in a time-dependent manner, as most existing tools do not consider temporal information. If a cluster comprises frames from different time intervals of the simulation, selecting representative members at each time interval could help preserve a pseudo-temporal information.

#### *4.5.3.3 Gaussian distribution on DL dataset*

One of the main issues of using CNN is that they are badly adapted to sparse matrices. In such scenarios, convolutions are performed over vacant space, resulting in a significant waste of computational power. To address partially this issue, tools like OctSurf (53) have been developed. Nonetheless, sparse matrices make it also harder for CNNs to effectively extract meaningful patterns, as important information can be lost during convolution, potentially resulting in poor performance. This situation is particularly relevant to protein-ligand complexes, where atoms are sparsely distributed throughout the structure. When the space of the binding pockets is discretized by assigning values to voxels in contact with atoms, only about 20% of the voxels contain atom-related information. This information is then distributed across 19 channels, further diluting the signal. Consequently, convolutions are performed on channels with minimal information content.

An effective method to increase the number of informative voxels and thereby prevent the loss of crucial information involves propagating atomic features to neighbouring voxels using a Gaussian distribution (Figure 77). This approach, inspired by previous implementations (262, 318), has been integrated into our neural networks as an option. By applying Gaussian distribution to adjacent voxels, the coverage of informative voxels increases to around 70%. Consequently, the use of Gaussian distribution proves to be an efficient technique for mitigating the sparsity of these matrices.

On a smaller dataset, we successfully trained Videonucy using this methodology, achieving a correlation coefficient of 0.80 and a RMSE of 1.55 on the MDbind test set. These results are promising as they bring Videonucy's performance, especially the RMSE, closer to that of Pafnucy. Currently, we are in the process of training models on the complete MDbind dataset using this methodology and anticipate further improvements in performance.

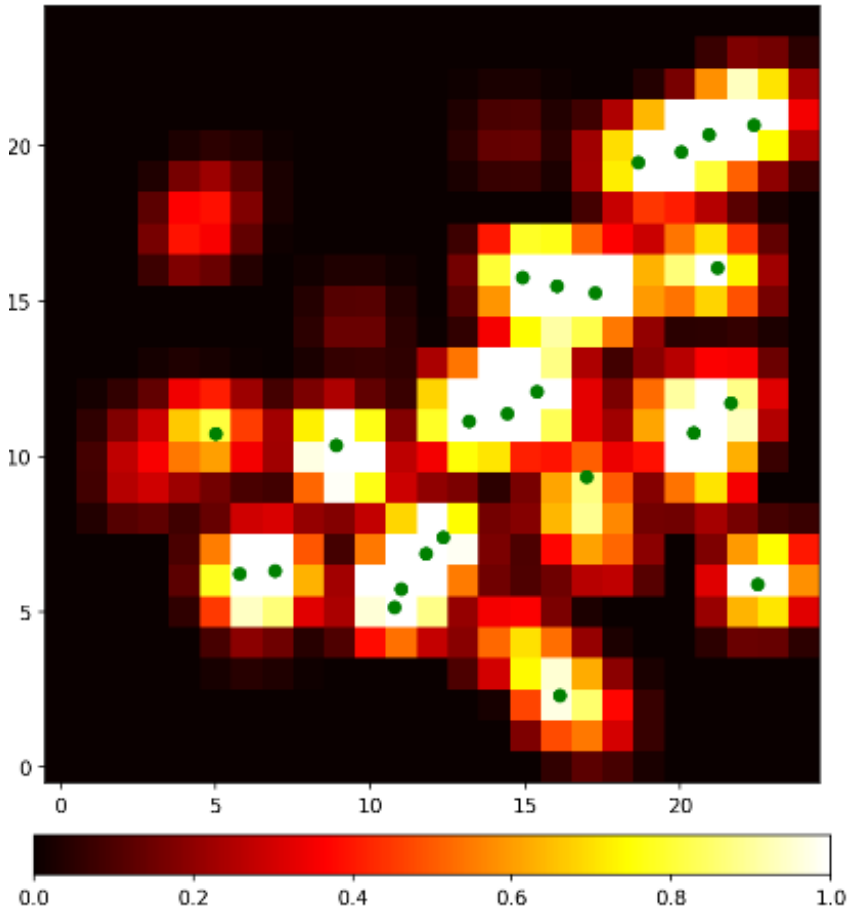



Figure 77: Visual representation of the Gaussian distribution of atomic properties. Visualization of the voxels on the 4th z-plane of the cubic input box with sides of 25 Å. Carbon atoms that are 1 Å away from the z plane, are displayed in green dots.

## 4.6 Conclusion

This work enabled us to compile a substantial dataset of MD simulations and develop neural networks capable of training on this data. We established a streamlined pipeline that takes complexes from the PDBbind, prepares them, and conducts simulations on high-performance computing clusters in an automated fashion. We had to clean and fix PDBbind's complexes as they were not initially prepared for conducting MD simulations. This comprehensive process resulted in the creation of MDbind, a dataset containing 63,000 simulations derived from 6,300 complexes. Following the simulations, the data was post-processed to generate datasets compatible with neural networks.

Before proceeding to the second part of this PhD project, which involves developing neural networks capable of training from the simulations, we assessed whether any valuable information could be gleaned from the MD simulations that might correlate with binding affinity. We observed that complex stability throughout the simulation tends to be lower for low-affinity complexes, whereas high-affinity complexes exhibit higher stability. Consequently, this information could potentially be leveraged by neural networks to enhance their ability to predict binding affinities.

Initially, we re-implemented Pafnucy, a widely recognized CNN in the field, in PyTorch, and we have named this implementation Proli. The objective of employing CNNs was to extract valuable information from the raw 3D structures, all while avoiding any bias introduced by expertly crafted descriptors. To enhance CNN performance, we replaced the convolutional layers with dense blocks,

leading to the creation of a neural network dubbed Densenucy. These neural networks are primarily trained on 3D structures and can be further enhanced through MD data augmentation, treating each frame extracted from MD simulations as a distinct structure. The newly added structures would offer novel binding poses to the neural networks, assisting them in learning about protein-ligand interactions. The MD augmented dataset used for training and validation consists of an extensive 3,000,000 frames. This dataset includes the original complexes from the PDBbind, amounting to approximately 17,000 structures.

Following this, we developed Timenucy and Videonucy, two neural networks capable of training models using entire MD simulations as input. This methodology, which we refer to as “spatio-temporal learning”, involves neural networks consisting of both CNN and LSTM layers. In the case of Timenucy, these layers are used sequentially, while Videonucy combines them into a convolutional LSTM. The primary objective behind these neural networks' development was to enable comprehensive analysis of entire simulations, with a particular emphasis on monitoring ligand movement within the binding site and tracking the evolution of interactions over time. Our models were trained and validated using a dataset consisting of 58,000 simulations obtained from 6,000 complexes. These figures represent a smaller subset of the MDbind dataset due to challenges encountered while handling certain frames, resulting in the exclusion of entire simulations from the training dataset.

We proceeded to train multiple models using the computational resources provided by JeanZay, a French national computational center. These models were trained on various datasets, including the PDBbind and MDbind, utilizing the full dataset as well as reduced datasets. Additionally, we initiated an exploration into training models on specific frames selected from MD simulations. The performance of these models was evaluated on three different datasets: the PDBbind core set 2016, the MDbind test set, and the FEP dataset.

For the MDbind test set, we implemented a method to average predictions across frames within the same simulations (this applies only to models providing predictions on frames). We further averaged predictions across all simulations for a given complex. As we trained 10 replicates of each model, this allowed us to employ a consensus approach to improve prediction accuracy. Additionally, we conducted a more detailed analysis of the performance of Densenucy MD data augmentation and Pafnucy, specifically focusing on activity cliffs in the PDBbind core set 2016. We also conducted a comprehensive investigation involving all models by assessing their performance on the FEP dataset.

In summary, our findings indicate that models trained with MD data augmentation methods outperformed the spatio-temporal learning models. One possible explanation for this performance gap is that the spatio-temporal learning models were trained on a smaller dataset, given the limited number of simulations compared to frames and a lower total number of complexes. Notably, Densenucy, in contrast to Pafnucy and Proli, exhibited improved performance from MD data augmentation. Densenucy's ability to leverage MD data augmentation more effectively may be attributed to its more compact architecture, which accommodates additional convolutional layers. Specifically, training Densenucy with MD data augmentation led to enhanced performance on the PDBbind core set 2016, MDbind test set and the FEP dataset. While evaluating models on these test sets, it is important to exercise caution, as biases in the data can artificially inflate the performance of models in comparison to real-life situations.

Furthermore, we conducted additional simulations after reconstructing the missing loops in the initial structures. This expansion of our dataset will increase our training data by 20-30%. It will also enhance the MDbind test set, enabling more thorough assessments of our models, especially the spatio-temporal learning models. Additionally, simulations were carried out on the FEP dataset,

facilitating the evaluation of the performance of spatio-temporal learning models on an external test set.

Overall, we are still struggling at creating models that truly understand protein-ligand interactions. Although CNNs help mitigate bias by not hand-picking specific descriptors to represent the data, our models still suffer inherent biases in the training data. As a result, models tend to predict binding affinity based on patterns within the proteins and molecules themselves rather than focusing on their interactions. Nonetheless, it is worth noting that Timenucy, and Videonucy appear capable of utilizing ligand positions and complex stability to predict binding affinity.

While we are hopeful that MD data can enhance the training of robust models, the current approach explored in this work requires further refinement to make the most of MD data.

## 4.7 Perspectives

Given the scale of our work, we were unable to comprehensively examine each and every simulation in detail. Therefore, a curation effort is advisable to enhance both the quality and quantity of the dataset by relaunching failed MD simulations for example. Retrospectively, there are areas where our approach could have been enhanced. For instance, prior to conducting MD simulations, protonation of proteins could have been performed, taking into account the local pH environment using tools like Propka, (319), the H++ server (286), or pKa-ANI (320). This adjustment would likely improve the behaviour of ligands in binding sites involving histidine residues, for instance.

For the DL part of the project, many improvements could be implemented to improve performance. First of all, the data could be handled in a different fashion by using GNN instead of CNN. This approach could potentially eliminate the need of a box and might better account for protein-ligand interactions, especially with larger ligands. Additionally, incorporating hydrogen atoms, which were not explicitly considered in our neural networks, could enhance models' understanding of protein-ligand interactions, particularly hydrogen bonds. Moreover, DL algorithms did not account for water molecules, despite their potential importance in binding mechanisms through water-mediated interactions. By considering water molecules, models could potentially achieve enhanced performance. Additionally, it might be necessary to guide the models to learn on interactions and ligand stability. This could involve providing descriptors like ligand RMSD or binding free energy decomposition calculated with methods like MMPBSA. However, this approach could introduce bias into the models and deviate from the philosophy of providing raw data for models to extract useful information by itself.

Further investigations are warranted to fully exploit MDbind. Building upon current research, it would be beneficial to explore MDbind downsampling methods to mitigate redundancy and enhance generalizability. Additionally, frame selection for neural network input requires further investigation. Not all frames may be equally informative for binding affinity prediction, so focusing on the most relevant frames could accelerate training while retaining critical information. Clustering could be used for frame selection, and weights assigned to the representative frames based on cluster size. This reduction in frames supplied to neural networks would facilitate the use of computationally intensive techniques such as systematic rotations. This would also allow spatio-temporal learning models to be trained using all 10 simulations of each complex during each epoch, in contrast to the current method of randomly selecting just one.

On the topic of frame selection, an alternative approach could involve implementing multi-instance learning (321), which serves as an intermediary method between MD data augmentation and spatio-temporal learning techniques. In this scenario, frames from the same simulation would be grouped together without considering their temporal interdependence. Neural networks would then learn to identify the most relevant frames within each simulation for predicting binding affinity.

To prevent the need for performing rotations, SE(3)-equivariant CNN (322) could be implemented, potentially enhancing performance without significantly increasing training time compared to random rotations. Another approach could involve the implementation of a graph transformer similar to Dynaformer (275), with the possibility of developing an advanced version for spatio-temporal learning, drawing inspiration from video transformers (185, 186). Additionally, methods like transfer learning could be employed to create local models that leverage MDbind. In this scenario, models trained on MDbind could be fine-tuned for specific targets, potentially improving performance.

It would be valuable to assess the performance of our models in a virtual screening scenario to gauge their practical utility. This can be accomplished by evaluating the models on datasets containing active molecules and docked decoys, such as DUD-E dataset (263). A compilation of such datasets is available at the end of my first publication.

Another aspect worth investigating is gaining an understanding of the decisions made by models when predicting binding affinities. Several tools in the field of explainable AI (XAI) have been developed to assess this information (267-269), by visually highlighting key part of molecules and pockets that are crucial for making predictions. This approach could provide additional insights into our models' behaviour and guide future development efforts aimed at improving their performance.

As we developed an expertise in kinetics in our team, it would be interesting to combine MD simulations and DL to predict kinetic parameters. Currently, while we consider the dynamic interactions of ligands with proteins, we do not assess their kinetic properties. Binding events were not observable in our simulations since we focused on complexes in their already bound state. Additionally, even if some unbinding events occurred during the 10 ns of simulation, they were rare across the entire MDbind dataset. To access kinetic events within a reduced simulation time, biased MD simulations like targeted MD (TMD) (102) should be utilized. However, the biases introduced in these simulations must be appropriately accounted for by the neural networks, which does not seem to be trivial.

# 5 General conclusion

The following part is a translation in French of the general conclusion.

La récente montée en puissance de l'apprentissage profond a donné lieu au développement de nombreux modèles de prédiction d'affinité basés sur les structures tridimensionnelles des complexes protéine-ligand. Cependant, l'enthousiasme initial pour ces modèles s'est atténué lorsque l'on a réalisé que leurs résultats prometteurs étaient en grande partie attribuables aux biais présents dans les jeux de données d'évaluation utilisés. Ainsi, ces modèles peinent à généraliser leurs prédictions et obtiennent de mauvaises performances lorsqu'ils sont évalués sur des jeux de test externes, principalement en raison de la faible quantité de données utilisées pour les entraîner, ainsi que de la nature sporadique de la matrice d'interaction protéine-ligand. De plus, il est encore nécessaire de développer des méthodes permettant d'orienter les modèles à apprendre à prédire l'affinité à partir des interactions protéine-ligand.

Pour résoudre ces problèmes, nous avons d'abord exploré l'utilisation optimal des données disponibles afin de tirer pleinement parti des informations qu'elles contiennent et ainsi améliorer les performances des modèles statistiques. Entre autres, nous avons constaté que la taille des poches fournies aux modèles influence de manière importante leurs performances. Cela nous a permis de mieux comprendre les limitations des réseaux de neurones utilisés pour prédire l'affinité de liaison. Par la suite, nous avons décidé de combiner les simulations de dynamique moléculaire avec l'apprentissage profond, car nous pensons que les simulations de dynamique moléculaire peuvent fournir un contexte temporel précieux sur les interactions protéine-ligand, ce qui peut s'avérer utile pour améliorer les performances des modèles. Pour ce faire, nous avons développé MDbind, un jeu de données composé de 63 000 simulations de dynamique moléculaire. Ces simulations ont été réalisées à partir de 6 300 complexes provenant de la PDBbind v.2019 (2) et dont les affinités de liaison sont connues. La MDbind a été utilisé pour entraîner une variété de réseaux neuronaux selon différentes méthodologies.

Dans une première approche, nous avons exploité les images extraites des simulations de dynamique moléculaire en tant qu'augmentation de données. À cet effet, nous avons introduit Densenucy, une version améliorée de Pafnucy (3), un 3D CNN reconnu dans le domaine de la prédiction de l'affinité de liaison. Dans une méthodologie alternative appelée « apprentissage spatio-temporel », nous avons utilisé les simulations en entier pour entraîner nos modèles. Pour ce faire, nous avons développé deux réseaux de neurones : Timenucy et Videonucy, composés respectivement d'un « *long recurrent convolutional network* » et d'un « *convolutional LSTM* ».

De manière générale, l'augmentation de données basée sur les simulations de dynamique moléculaire s'est révélée être un outil efficace pour améliorer les prédictions d'affinité de liaison, notamment avec Densenucy, qui a atteint des performances remarquables. En revanche, les méthodes spatio-temporelles ont affiché des performances modestes, mais avec une quantité suffisante de données d'entraînement, il est raisonnable de s'attendre à ce que ces outils puissent exploiter pleinement le potentiel des simulations de dynamique moléculaire et fournir des prédictions très précises. Étant donné que les méthodes spatio-temporelles nécessitent d'être appliquées sur des simulations pour effectuer leurs prédictions, elles conviennent mieux à une utilisation lors de l'étape d'optimisation des candidats médicamenteux plutôt que lors d'une campagne de criblage virtuel.

Nous sommes convaincus que les simulations de dynamique moléculaire ont le potentiel d'améliorer considérablement les performances des modèles entraînés par apprentissage profond. Cette voie d'amélioration des prédictions de l'affinité de liaison mérite d'être explorée davantage et nécessite des recherches supplémentaires.

The rise of deep learning has brought significant attention to affinity prediction models based on the 3D structures of protein-ligand complexes. However, Initial optimism regarding these models has diminished as it became apparent that their promising results were often based on biased test sets. These models struggle to generalize when applied to more challenging test sets, primarily due to the limited amount of available training data in this field and the sparsity of the protein-ligand interaction matrix. Additionally, there is still a need to develop methods that can effectively force models to learn from protein-ligand interactions.

To address these issues, we initially explored the optimal usage of data to fully use the information available and enhance performance. Among our findings, we discovered that the pocket size significantly influenced the model's performance. Subsequently, we opted to combine MD simulations with deep learning, as we believed that MD simulations could provide valuable temporal context about protein-ligand interactions, thereby enhancing models' performance. To achieve this, we curated MDbind, a dataset consisting of 63,000 simulations conducted on 6,300 complexes with known binding affinities drawn from the PDBbind v.2019 dataset (2). MDbind was employed to train a variety of neural networks using different methodologies.

In one approach, we utilized the frames extracted from the simulations as MD data augmentation. To implement this, we introduced Densenucy, an enhanced version of the well-known CNN, Pafnucy (3). In an alternative methodology referred to as "spatio-temporal learning", we employed simulations directly to train our models. Within this framework, we developed two neural networks: Timenucy and Videonucy, which are composed of a long recurrent convolutional network and a convolutional LSTM, respectively.

Overall, MD data augmentation demonstrated its benefits in improving binding affinity predictions, particularly with Densenucy, which achieved state-of-the-art performance. On the other hand, spatio-temporal methods showed modest performance, but it is anticipated that with a sufficient amount of training data, these tools could harness the full potential of MD simulations and deliver highly accurate predictions. Spatio-temporal methods, relying on simulations for predictions, are better suited for lead optimization rather than virtual screening scenarios.

We are confident that MD simulations have the potential to significantly enhance the performance of DL models, and this warrants further exploration and research.

# Scientific communications

## Publications

Libouban, Pierre-Yves, Samia Aci-Sèche, Jose Carlos Gómez-Tamayo, Gary Tresadern, and Pascal Bonnet. "The Impact of Data on Structure-Based Binding Affinity Predictions Using Deep Neural Networks" International Journal of Molecular Sciences (IJMS), **2023**, *24*, 16120. https://doi.org/10.3390/ijms242216120

## Oral communications

### Oral communications in an international congress:

- **Libouban, P.-Y.** ; Aci-Sèche, S. ; Tresadern, G. ; Bonnet, P.
  A novel protocol for protein-ligand binding affinity prediction via spatio-temporal deep learning and molecular dynamics simulations
  XXIII Group of Graphism and Molecular Modeling (GGMM) congress
  **May 2023** – Toulouse.

- **Libouban, P.-Y.** ; Aci-Sèche, S. ; Tresadern, G. ; Bonnet, P.
  The use of deep neural networks on molecular dynamics simulations for the prediction of binding affinities
  **RSC- Analyst prize**
  23rd European Symposium on Quantitative Structure-Activity Relationship (EuroQSAR)
  **September 2022** – Heidelberg (Germany).

### Oral communications in a national congress:

- **Libouban, P.-Y.** ; Aci-Sèche, S. ; Tresadern, G. ; Bonnet, P.
  Predicting protein-ligand binding affinity by applying spatio-temporal deep learning on molecular dynamics simulations
  Thematic days « Deep Learning, theory and application » RTR DIAMS
  **March 2023** – Tours.

- **Libouban, P.-Y.** ; Aci-Sèche, S. ; Tresadern, G. ; Bonnet, P.
  Spatio-temporal deep learning combined with molecular dynamics simulations to predict protein-ligand binding affinity
  Thematic days « Human in the loop for data mining and machine learning » RTR DIAMS
  **January 2023** – Orléans.

- **Libouban, P.-Y.** ; Aci-Sèche, S. ; Tresadern, G. ; Bonnet, P.
  Protein-Ligand binding affinity prediction by combining molecular dynamics simulations with deep learning approaches
  3rd PhD students' days MIPTIS / RTR DIAMS
  **April 2022** – Tours.

- **Libouban, P.-Y.** ; Aci-Sèche, S. ; Tresadern, G. ; Bonnet, P.
  Protein-Ligand binding affinity prediction using combined molecular dynamics and deep learning approaches
  Data and computation days (JCAD)
  **December 2021** – Dijon.

- **Libouban, P.-Y.** ; Aci-Sèche, S. ; Tresadern, G. ; Bonnet, P.
  Prediction of binding affinity of protein-ligand complexes combining molecular dynamics simulations with deep learning algorithms
  3rd Young research fellow days (J2C-2021)
  **March 2021** – Orléans.

### Flash communication in an international congress:

- **Libouban, P.-Y.** ; Aci-Sèche, S. ; Tresadern, G. ; Bonnet, P.
  Learning from molecular dynamics simulations with deep neural networks for the prediction of protein-ligand binding affinity
  29th Young Research Fellow Meeting (YRFM)
  **July 2022** – Nantes.

### Flash communications in a national congress:

- **Libouban, P.-Y.** ; Aci-Sèche, S. ; Tresadern, G. ; Bonnet, P.
  Learning from MD simulations dataset
  1st deep learning for sciences days (JDLS)
  **May 2023** – Orsay.

- **Libouban, P.-Y.** ; Aci-Sèche, S. ; Tresadern, G. ; Bonnet, P.
  Protein-Ligand binding affinity prediction using combined molecular dynamics and deep learning approaches
  XXII Group of Graphism and Molecular Modeling (GGMM) congress & 10th French Society of Chemoinformatics (SFCi) days
  **October 2021** – Lille.

- **Libouban, P.-Y.** ; Aci-Sèche, S. ; Tresadern, G. ; Bonnet, P.
  Prediction of binding affinity of protein-ligand complexes combining molecular dynamics simulations with deep learning algorithms
  33rd Biotechnocentre conference
  **October 2021** – Les Hauts de Bruyères.

## Poster communications

### Poster communication in an international congress:

- **Libouban, P.-Y.** ; Aci-Sèche, S. ; Tresadern, G. ; Bonnet, P.
  Learning from molecular dynamics simulations with deep neural networks for the prediction of protein-ligand binding affinity
  29th Young Research Fellow Meeting (YRFM)
  **July 2022** – Nantes.

- **Libouban, P.-Y.** ; Aci-Sèche, S. ; Tresadern, G. ; Bonnet, P.
  Development of a method combining molecular dynamics simulations and deep learning for the prediction of protein-ligand binding affinity
  8th Strasbourg Summer School in Chemoinformatics (CS3-2022)
  **June 2022** – Strasbourg.

### Poster communications in a national congress:

- **Libouban, P.-Y.** ; Aci-Sèche, S. ; Tresadern, G. ; Bonnet, P.
  Protein-Ligand binding affinity prediction using combined molecular dynamics and deep learning approaches
  XXII Group of Graphism and Molecular Modeling (GGMM) congress & 10th French Society of Chemoinformatics (SFCi) days
  **October 2021** – Lille.

- **Libouban, P.-Y.** ; Aci-Sèche, S. ; Tresadern, G. ; Bonnet, P.
  Prediction of binding affinity of protein-ligand complexes combining molecular dynamics simulations with deep learning algorithms
  33rd Biotechnocentre conference
  **October 2021** – Les Hauts de Bruyères.

# Bibliography


1. Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, et al. Highly accurate protein structure prediction with AlphaFold. Nature. 2021.
2. Wang R, Fang X, Lu Y, Wang S. The PDBbind database: collection of binding affinities for protein-ligand complexes with known three-dimensional structures. Journal of medicinal chemistry. 2004;47(12):2977-80.
3. Stepniewska-Dziubinska MM, Zielenkiewicz P, Siedlecki P. Development and evaluation of a deep learning model for protein-ligand binding affinity prediction. Bioinformatics (Oxford, England). 2018;34(21):3666-74.
4. Jiménez J, Škalič M, Martínez-Rosell G, De Fabritiis G. KDEEP: Protein–Ligand Absolute Binding Affinity Prediction via 3D-Convolutional Neural Networks. Journal of chemical information and modeling. 2018;58(2):287-96.
5. Zheng L, Fan J, Mu Y. OnionNet: a Multiple-Layer Intermolecular-Contact-Based Convolutional Neural Network for Protein–Ligand Binding Affinity Prediction. ACS Omega. 2019;4(14):15956-65.
6. Son J, Kim D. Development of a graph convolutional neural network model for efficient prediction of protein-ligand binding affinities. PLoS One. 2021;16(4):e0249404.
7. Moon S, Zhung W, Yang S, Lim J, Kim WY. PIGNet: a physics-informed deep learning model toward generalized drug–target interaction predictions. Chemical Science. 2022;13(13):3661-73.
8. Su M, Yang Q, Du Y, Feng G, Liu Z, Li Y, et al. Comparative Assessment of Scoring Functions: The CASF-2016 Update. Journal of chemical information and modeling. 2019;59(2):895-913.
9. Volkov M, Turk J-A, Drizard N, Martin N, Hoffmann B, Gaston-Mathé Y, et al. On the Frustration to Predict Binding Affinities from Protein–Ligand Structures with Deep Neural Networks. Journal of medicinal chemistry. 2022.
10. Logé F, Besson R, Allassonnière S. Optimisation des parcours patients pour lutter contre l'errance de diagnostic des patients atteints de maladies rares. arXiv preprint arXiv:201014167. 2020.
11. Myers MB. Targeted therapies with companion diagnostics in the management of breast cancer: current perspectives. Pharmacogenomics and personalized medicine. 2016;9:7-16.
12. Nogrady B. How cancer genomics is transforming diagnosis and treatment. Nature. 2020;579(7800):S10-s1.
13. Huang Y, Yang C, Xu X-f, Xu W, Liu S-w. Structural and functional properties of SARS-CoV-2 spike protein: potential antivirus drug development for COVID-19. Acta Pharmacologica Sinica. 2020;41(9):1141-9.
14. Elsevier. Reaxys. 2009. Available from: https://www.reaxys.com.
15. Sun D, Gao W, Hu H, Zhou S. Why 90% of clinical drug development fails and how to improve it? Acta Pharmaceutica Sinica B. 2022;12(7):3049-62.
16. Schlander M, Hernandez-Villafuerte K, Cheng C-Y, Mestre-Ferrandiz J, Baumann M. How Much Does It Cost to Research and Develop a New Drug? A Systematic Review and Assessment. PharmacoEconomics. 2021;39(11):1243-69.
17. Wouters OJ, McKee M, Luyten J. Estimated Research and Development Investment Needed to Bring a New Medicine to Market, 2009-2018. Jama. 2020;323(9):844-53.
18. Dickson M, Gagnon JP. Key factors in the rising cost of new drug discovery and development. Nature Reviews Drug Discovery. 2004;3(5):417-29.
19. Aliagas I, Berger R, Goldberg K, Nishimura RT, Reilly J, Richardson P, et al. Sustainable Practices in Medicinal Chemistry Part 2: Green by Design. Journal of medicinal chemistry. 2017;60(14):5955-68.
20. AI in Pharma Global Market Report. 2023. [Accessed 10 Aug 2023]. Available from: https://www.researchandmarkets.com/reports/5767266/ai-in-pharma-global-market-report.
21. Jiang J, Ma X, Ouyang D, Williams RO. Emerging Artificial Intelligence (AI) Technologies Used in the Development of Solid Dosage Forms. Pharmaceutics. 2022;14(11):2257.

22. Landrum G. RDKit: Open-source cheminformatics. 2010. Available from: https://www.rdkit.org.
23. O'Boyle NM, Banck M, James CA, Morley C, Vandermeersch T, Hutchison GR. Open Babel: An open chemical toolbox. Journal of Cheminformatics. 2011;3(1):33.
24. Dahlgren B. ChemPy: A package useful for chemistry written in Python. Journal of Open Source Software. 2018;3(24):565.
25. ChemPython. Available from: http://www.chempython.org/.
26. Lewis R. Scikit-chem. 2016. Available from: https://scikit-chem.readthedocs.io/en/latest/.
27. Cock PJ, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, et al. Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics (Oxford, England). 2009;25(11):1422-3.
28. Gally J-M, Bourg S, Do Q-T, Aci-Sèche S, Bonnet P. VSPrep: A General KNIME Workflow for the Preparation of Molecules for Virtual Screening. Molecular informatics. 2017;36(10):1700023.
29. Gally J-M, Bourg S, Fogha J, Do Q-T, Aci-Sèche S, Bonnet P. VSPrep: A KNIME Workflow for the Preparation of Molecular Databases for Virtual Screening. Current Medicinal Chemistry. 2020;27(38):6480-94.
30. Frags2Drugs. [Accessed 15 september 2023]. Available from: http://frags2drugs.icoa.fr/.
31. David L, Thakkar A, Mercado R, Engkvist O. Molecular representations in AI-driven drug discovery: a review and practical guide. Journal of Cheminformatics. 2020;12(1):56.
32. Atz K, Grisoni F, Schneider G. Geometric deep learning on molecular representations. Nature Machine Intelligence. 2021;3(12):1023-32.
33. Weininger D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. Journal of Chemical Information and Computer Sciences. 1988;28(1):31-6.
34. Daylight. Theory: SMARTS - A Language for Describing Molecular Patterns. [Accessed 1 Aug 2023]. Available from: https://www.daylight.com/dayhtml/doc/theory/theory.smarts.html.
35. Mouchlis VD, Afantitis A, Serra A, Fratello M, Papadiamantis AG, Aidinis V, et al. Advances in de Novo Drug Design: From Conventional to Machine Learning Methods. Int J Mol Sci. 2021;22(4).
36. Krenn M, Häse F, Nigam A, Friederich P, Aspuru-Guzik A. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. Machine Learning: Science and Technology. 2020;1(4):045024.
37. Hung C, Gini G. QSAR modeling without descriptors using graph convolutional neural networks: the case of mutagenicity prediction. Molecular Diversity. 2021;25(3):1283-99.
38. Jiang D, Wu Z, Hsieh C-Y, Chen G, Liao B, Wang Z, et al. Could graph neural networks learn better molecular representation for drug discovery? A comparison study of descriptor-based and graph-based models. Journal of Cheminformatics. 2021;13(1):12.
39. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, et al. The Protein Data Bank. Nucleic acids research. 2000;28(1):235-42.
40. Salentin S, Haupt VJ, Daminelli S, Schroeder M. Polypharmacology rescored: Protein–ligand interaction profiles for remote binding site similarity assessment. Progress in biophysics and molecular biology. 2014;116(2):174-86.
41. Ferreira de Freitas R, Schapira M. A systematic analysis of atomic protein-ligand interactions in the PDB. MedChemComm. 2017;8(10):1970-81.
42. Trott O, Olson AJ. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. J Comput Chem. 2010;31(2):455-61.
43. Allen WJ, Balius TE, Mukherjee S, Brozell SR, Moustakas DT, Lang PT, et al. DOCK 6: Impact of new features and current docking performance. Journal of computational chemistry. 2015;36(15):1132-56.
44. Eswar N, Webb B, Marti-Renom MA, Madhusudhan MS, Eramian D, Shen MY, et al. Comparative protein structure modeling using Modeller. Current protocols in bioinformatics. 2006;Chapter 5:Unit-5.6.

45. Baek M, DiMaio F, Anishchenko I, Dauparas J, Ovchinnikov S, Lee GR, et al. Accurate prediction of protein structures and interactions using a three-track neural network. Science (New York, NY). 2021;373(6557):871-6.
46. Hekkelman ML, de Vries I, Joosten RP, Perrakis A. AlphaFill: enriching AlphaFold models with ligands and cofactors. Nat Methods. 2023;20(2):205-13.
47. DeLano WL. Pymol: An open-source molecular graphics tool. CCP4 Newsl Protein Crystallogr. 2002;40(1):82-92.
48. Pettersen EF, Goddard TD, Huang CC, Couch GS, Greenblatt DM, Meng EC, et al. UCSF Chimera—A visualization system for exploratory research and analysis. Journal of Computational Chemistry. 2004;25(13):1605-12.
49. Humphrey W, Dalke A, Schulten K. VMD: visual molecular dynamics. Journal of molecular graphics. 1996;14(1):33-8, 27-8.
50. Wang Y, Wu S, Duan Y, Huang Y. A point cloud-based deep learning strategy for protein-ligand binding affinity prediction. Briefings in bioinformatics. 2022;23(1).
51. Feinberg EN, Sur D, Wu Z, Husic BE, Mai H, Li Y, et al. PotentialNet for Molecular Property Prediction. ACS Central Science. 2018;4(11):1520-30.
52. Dai B, Bailey-Kellogg C. Protein interaction interface region prediction by geometric deep learning. Bioinformatics (Oxford, England). 2021;37(17):2580-8.
53. Liu Q, Wang P-S, Zhu C, Gaines BB, Zhu T, Bi J, et al. OctSurf: Efficient hierarchical voxel-based molecular surface representation for protein-ligand affinity prediction. Journal of Molecular Graphics and Modelling. 2021;105:107865.
54. Özçelik R, van Tilborg D, Jiménez-Luna J, Grisoni F. Structure-Based Drug Discovery with Deep Learning. ChemBioChem. 2023;24(13):e202200776.
55. Isert C, Atz K, Schneider G. Structure-based drug design with geometric deep learning. Proceedings of the 2022.
56. Consonni V, Ballabio D, Todeschini R. Chapter 12 - Chemical space and molecular descriptors for QSAR studies. In: Roy K, editor. Cheminformatics, QSAR and Machine Learning Applications for Novel Drug Development: Academic Press; 2023. p. 303-27.
57. Rogers D, Hahn M. Extended-Connectivity Fingerprints. Journal of chemical information and modeling. 2010;50(5):742-54.
58. Bellmann L, Penner P, Rarey M. Connected Subgraph Fingerprints: Representing Molecules Using Exhaustive Subgraph Enumeration. Journal of chemical information and modeling. 2019;59(11):4625-35.
59. Duvenaud D, Maclaurin D, Aguilera-Iparraguirre J, Gómez-Bombarelli R, Hirzel T, Aspuru-Guzik A, et al. Convolutional networks on graphs for learning molecular fingerprints. Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2. 2015:2224–32.
60. Yin Z, Song W, Li B, Wang F, Xie L, Xu X. Neural networks prediction of the protein-ligand binding affinity with circular fingerprints. Technology and Health Care. 2023;31:487-95.
61. Mauri A, Consonni V, Pavan M, Todeschini R. Dragon software: An easy approach to molecular descriptor calculations. Match: Communications in Mathematical and in Computer Chemistry. 2006;56(2):237-48.
62. Ruggiu F, Marcou G, Varnek A, Horvath D. ISIDA Property-Labelled Fragment Descriptors. Molecular informatics. 2010;29(12):855-68.
63. van Westen GJP, Swier RF, Wegner JK, Ijzerman AP, van Vlijmen HWT, Bender A. Benchmarking of protein descriptor sets in proteochemometric modeling (part 1): comparative study of 13 amino acid descriptor sets. Journal of Cheminformatics. 2013;5(1):41.
64. Sandberg M, Eriksson L, Jonsson J, Sjöström M, Wold S. New Chemical Descriptors Relevant for the Design of Biologically Active Peptides. A Multivariate Characterization of 87 Amino Acids. Journal of medicinal chemistry. 1998;41(14):2481-91.
65. Zaliani A, Gancia E. MS-WHIM Scores for Amino Acids: A New 3D-Description for Peptide QSAR and QSPR Studies. Journal of Chemical Information and Computer Sciences. 1999;39(3):525-33.

66. Contreras-Torres E, Marrero-Ponce Y, Terán JE, García-Jacas CR, Brizuela CA, Sánchez-Rodríguez JC. MuLiMs-MCoMPAs: A Novel Multiplatform Framework to Compute Tensor Algebra-Based Three-Dimensional Protein Descriptors. Journal of chemical information and modeling. 2020;60(2):1042-59.
67. Cha M, Emre EST, Xiao X, Kim J-Y, Bogdan P, VanEpps JS, et al. Unifying structural descriptors for biological and bioinspired nanoscale complexes. Nature Computational Science. 2022;2(4):243-52.
68. Sánchez-Cruz N, Medina-Franco JL, Mestres J, Barril X. Extended connectivity interaction features: improving binding affinity prediction through chemical description. Bioinformatics (Oxford, England). 2020;37(10):1376-82.
69. Deng Z, Chuaqui C, Singh J. Structural Interaction Fingerprint (SIFt): A Novel Method for Analyzing Three-Dimensional Protein−Ligand Binding Interactions. Journal of medicinal chemistry. 2004;47(2):337-44.
70. Pérez-Nueno VI, Rabal O, Borrell JI, Teixidó J. APIF: A New Interaction Fingerprint Based on Atom Pairs and Its Application to Virtual Screening. Journal of chemical information and modeling. 2009;49(5):1245-60.
71. Da C, Kireev D. Structural Protein–Ligand Interaction Fingerprints (SPLIF) for Structure-Based Virtual Screening: Method and Benchmark Study. Journal of chemical information and modeling. 2014;54(9):2555-61.
72. Radifar M, Yuniarti N, Istyastono EP. PyPLIF: Python-based Protein-Ligand Interaction Fingerprinting. Bioinformation. 2013;9(6):325-8.
73. Leidner F, Kurt Yilmaz N, Schiffer CA. Target-Specific Prediction of Ligand Affinity with Structure-Based Interaction Fingerprints. Journal of chemical information and modeling. 2019;59(9):3679-91.
74. Wójcikowski M, Kukiełka M, Stepniewska-Dziubinska MM, Siedlecki P. Development of a protein–ligand extended connectivity (PLEC) fingerprint and its application for binding affinity predictions. Bioinformatics (Oxford, England). 2018;35(8):1334-41.
75. Riniker S. Molecular Dynamics Fingerprints (MDFP): Machine Learning from MD Data To Predict Free-Energy Differences. Journal of chemical information and modeling. 2017;57(4):726-41.
76. Cox PB, Gupta R. Contemporary Computational Applications and Tools in Drug Discovery. ACS Medicinal Chemistry Letters. 2022;13(7):1016-29.
77. Kuntz ID, Blaney JM, Oatley SJ, Langridge R, Ferrin TE. A geometric approach to macromolecule-ligand interactions. Journal of molecular biology. 1982;161(2):269-88.
78. Halperin I, Ma B, Wolfson H, Nussinov R. Principles of docking: An overview of search algorithms and a guide to scoring functions. Proteins: Structure, Function, and Bioinformatics. 2002;47(4):409-43.
79. Li J, Fu A, Zhang L. An Overview of Scoring Functions Used for Protein–Ligand Interactions in Molecular Docking. Interdisciplinary Sciences: Computational Life Sciences. 2019;11(2):320-8.
80. Meli R, Morris G, Biggin P. Scoring functions for protein-ligand binding affinity prediction using structure-based deep learning: a review. Frontiers in Bioinformatics. 2022;2.
81. Murray CW, Auton TR, Eldridge MD. Empirical scoring functions. II. The testing of an empirical scoring function for the prediction of ligand-receptor binding affinities and the use of Bayesian regression to improve the quality of the model. J Comput Aided Mol Des. 1998;12(5):503-19.
82. Ballester PJ, Mitchell JBO. A machine learning approach to predicting protein-ligand binding affinity with applications to molecular docking. Bioinformatics (Oxford, England). 2010;26(9):1169-75.
83. Durrant JD, McCammon JA. NNScore 2.0: A Neural-Network Receptor–Ligand Scoring Function. Journal of chemical information and modeling. 2011;51(11):2897-903.
84. Crampon K, Giorkallos A, Deldossi M, Baud S, Steffenel LA. Machine-learning methods for ligand–protein molecular docking. Drug Discovery Today. 2022;27(1):151-64.
85. Pagadala NS, Syed K, Tuszynski J. Software for molecular docking: a review. Biophysical Reviews. 2017;9(2):91-102.
86. McCammon JA, Gelin BR, Karplus M. Dynamics of folded proteins. Nature. 1977;267(5612):585-90.

87. Pearlman DA, Case DA, Caldwell JW, Ross WS, Cheatham TE, DeBolt S, et al. AMBER, a package of computer programs for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to simulate the structural and energetic properties of molecules. Computer Physics Communications. 1995;91(1):1-41.
88. Berendsen HJC, van der Spoel D, van Drunen R. GROMACS: A message-passing parallel molecular dynamics implementation. Computer Physics Communications. 1995;91(1):43-56.
89. Brooks BR, Brooks CL, 3rd, Mackerell AD, Jr., Nilsson L, Petrella RJ, Roux B, et al. CHARMM: the biomolecular simulation program. J Comput Chem. 2009;30(10):1545-614.
90. Bender BJ, Gahbauer S, Luttens A, Lyu J, Webb CM, Stein RM, et al. A practical guide to large-scale docking. Nature protocols. 2021;16(10):4799-832.
91. Hollingsworth SA, Dror RO. Molecular Dynamics Simulation for All. Neuron. 2018;99(6):1129-43.
92. Szöllősi D, Rose-Sperling D, Hellmich UA, Stockner T. Comparison of mechanistic transport cycle models of ABC exporters. Biochimica et Biophysica Acta (BBA) - Biomembranes. 2018;1860(4):818-32.
93. Kohn W, Becke AD, Parr RG. Density Functional Theory of Electronic Structure. The Journal of Physical Chemistry. 1996;100(31):12974-80.
94. Groenhof G. Introduction to QM/MM Simulations. In: Monticelli L, Salonen E, editors. Biomolecular Simulations: Methods and Protocols. Totowa, NJ: Humana Press; 2013. p. 43-66.
95. The royal Swedish academy of sciences. Development of multiscale models for complex chemical systems. 2013. Available from: https://www.nobelprize.org/uploads/2018/06/advanced-chemistryprize2013.pdf.
96. Miao Y, Feher VA, McCammon JA. Gaussian Accelerated Molecular Dynamics: Unconstrained Enhanced Sampling and Free Energy Calculation. Journal of Chemical Theory and Computation. 2015;11(8):3584-95.
97. Bussi G, Laio A. Using metadynamics to explore complex free-energy landscapes. Nature Reviews Physics. 2020;2(4):200-12.
98. Hénin J, Lelièvre T, Shirts MR, Valsson O, Delemotte L. Enhanced sampling methods for molecular dynamics simulations. arXiv preprint arXiv:220204164. 2022.
99. Lazim R, Suh D, Choi S. Advances in Molecular Dynamics Simulations and Enhanced Sampling Methods for the Study of Protein Systems. International Journal of Molecular Sciences. 2020;21(17):6339.
100. Marrink SJ, Risselada HJ, Yefimov S, Tieleman DP, de Vries AH. The MARTINI Force Field: Coarse Grained Model for Biomolecular Simulations. The Journal of Physical Chemistry B. 2007;111(27):7812-24.
101. Izrailev S, Stepaniants S, Balsera M, Oono Y, Schulten K. Molecular dynamics study of unbinding of the avidin-biotin complex. Biophysical Journal. 1997;72(4):1568-81.
102. Schlitter J, Engels M, Krüger P. Targeted molecular dynamics: a new approach for searching pathways of conformational transitions. Journal of molecular graphics. 1994;12(2):84-9.
103. Huang H, Ozkirimli E, Post CB. A Comparison of Three Perturbation Molecular Dynamics Methods for Modeling Conformational Transitions. J Chem Theory Comput. 2009;5(5):1301-14.
104. Braka A, Garnier N, Bonnet P, Aci-Sèche S. Residence Time Prediction of Type 1 and 2 Kinase Inhibitors from Unbinding Simulations. Journal of chemical information and modeling. 2020;60(1):342-8.
105. Ziada S, Diharce J, Raimbaud E, Aci-Sèche S, Ducrot P, Bonnet P. Estimation of Drug-Target Residence Time by Targeted Molecular Dynamics Simulations. Journal of chemical information and modeling. 2022;62(22):5536-49.
106. King E, Aitchison E, Li H, Luo R. Recent developments in free energy calculations for drug discovery. Frontiers in Molecular Biosciences. 2021;8:712085.
107. Hou T, Wang J, Li Y, Wang W. Assessing the performance of the MM/PBSA and MM/GBSA methods. 1. The accuracy of binding free energy calculations based on molecular dynamics simulations. Journal of chemical information and modeling. 2011;51(1):69-82.

108. Hansson T, Marelius J, Åqvist J. Ligand binding affinity prediction by linear interaction energy methods. Journal of Computer-Aided Molecular Design. 1998;12(1):27-35.
109. Decherchi S, Cavalli A. Thermodynamics and Kinetics of Drug-Target Binding by Molecular Simulation. Chemical Reviews. 2020;120(23):12788-833.
110. Lu N, Kofke DA. Accuracy of free-energy perturbation calculations in molecular simulation. I. Modeling. The Journal of Chemical Physics. 2001;114(17):7303-11.
111. Bhati AP, Wan S, Wright DW, Coveney PV. Rapid, Accurate, Precise, and Reliable Relative Free Energy Prediction Using Ensemble Based Thermodynamic Integration. Journal of Chemical Theory and Computation. 2017;13(1):210-22.
112. Yang X, Wang Y, Byrne R, Schneider G, Yang S. Concepts of Artificial Intelligence for Computer-Assisted Drug Discovery. Chemical Reviews. 2019;119(18):10520-94.
113. Carles F. Développement d'une approche protéo-chimiométrique tridimensionnelle pour l'identification d'inhibiteurs de protéines kinases: Orléans; 2019.
114. Breiman L. Random Forests. Machine Learning. 2001;45(1):5-32.
115. Cortes C, Vapnik V. Support-vector networks. Machine Learning. 1995;20(3):273-97.
116. Cihan Sorkun M, Mullaj D, Koelman JMVA, Er S. ChemPlot, a Python Library for Chemical Space Visualization. Chemistry–Methods. 2022;2(7):e202200005.
117. Svante W, Kim E, Paul G. Principal component analysis. Chemometrics and Intelligent Laboratory Systems. 1987;2(1):37-52.
118. Van der Maaten L, Hinton G. Visualizing data using t-SNE. Journal of machine learning research. 2008;9(11).
119. McInnes L, Healy J, Melville J. Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:180203426. 2018.
120. Kireeva N, Baskin II, Gaspar HA, Horvath D, Marcou G, Varnek A. Generative Topographic Mapping (GTM): Universal Tool for Data Visualization, Structure-Activity Modeling and Dataset Comparison. Molecular informatics. 2012;31(3-4):301-12.
121. Kohonen T. The self-organizing map. Proceeding of the IEEE. 1990;78(9):1464-80.
122. Ward JH. Hierarchical Grouping to Optimize an Objective Function. Journal of the American Statistical Association. 1963;58(301):236-44.
123. MacQueen J. Some methods for classification and analysis of multivariate observations. Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. 1967;1(14):281-97.
124. Ester M, Kriegel H-P, Sander J, Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. 1996:226–31.
125. Rodriguez A, Laio A. Clustering by fast search and find of density peaks. Science (New York, NY). 2014;344(6191):1492-6.
126. Frey BJ, Dueck D. Clustering by Passing Messages Between Data Points. Science (New York, NY). 2007;315(5814):972-6.
127. Breiman L, Friedman J, Stone CJ, Olshen RA. Classification and Regression Trees: Taylor & Francis; 1984.
128. Zhang H. The optimality of naive Bayes. Proceedings of the 17th International Florida Artificial Intelligence Research Society Conference; 2004; Menlo Park.
129. Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016:785–94.
130. Liedberg B, Nylander C, Lunström I. Surface plasmon resonance for gas detection and biosensing. Sensors and Actuators. 1983;4:299-304.
131. Leavitt S, Freire E. Direct measurement of protein binding energetics by isothermal titration calorimetry. Current opinion in structural biology. 2001;11(5):560-6.
132. Hansch C, Maloney PP, Fujita T, Muir RM. Correlation of Biological Activity of Phenoxyacetic Acids with Hammett Substituent Constants and Partition Coefficients. Nature. 1962;194(4824):178-80.

133. Sheridan RP, Karnachi P, Tudor M, Xu Y, Liaw A, Shah F, et al. Experimental Error, Kurtosis, Activity Cliffs, and Methodology: What Limits the Predictivity of Quantitative Structure–Activity Relationship Models? Journal of chemical information and modeling. 2020;60(4):1969-82.
134. Dahl GE, Jaitly N, Salakhutdinov R. Multi-task neural networks for QSAR predictions. arXiv preprint arXiv:14061231. 2014.
135. Karpov P, Godin G, Tetko IV. Transformer-CNN: Swiss knife for QSAR modeling and interpretation. Journal of Cheminformatics. 2020;12(1):17.
136. Muratov EN, Bajorath J, Sheridan RP, Tetko IV, Filimonov D, Poroikov V, et al. QSAR without borders. Chemical Society Reviews. 2020;49(11):3525-64.
137. Wang R, Lai L, Wang S. Further development and validation of empirical scoring functions for structure-based binding affinity prediction. J Comput Aided Mol Des. 2002;16(1):11-26.
138. Friesner RA, Banks JL, Murphy RB, Halgren TA, Klicic JJ, Mainz DT, et al. Glide: a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy. Journal of medicinal chemistry. 2004;47(7):1739-49.
139. Wang C, Zhang Y. Improving scoring-docking-screening powers of protein-ligand scoring functions using random forest. J Comput Chem. 2017;38(3):169-77.
140. Shen C, Zhang X, Hsieh C-Y, Deng Y, Wang D, Xu L, et al. A generalized protein–ligand scoring framework with balanced scoring, docking, ranking and screening powers. Chemical Science. 2023;14(30):8129-46.
141. Muehlbacher M, Kerdawy AE, Kramer C, Hudson B, Clark T. Conformation-Dependent QSPR Models: logPOW. Journal of chemical information and modeling. 2011;51(9):2408-16.
142. Lee S, Lee M, Gyak K-W, Kim SD, Kim M-J, Min K. Novel Solubility Prediction Models: Molecular Fingerprints and Physicochemical Features vs Graph Convolutional Neural Networks. ACS Omega. 2022;7(14):12268-77.
143. Faramarzi S, Kim MT, Volpe DA, Cross KP, Chakravarti S, Stavitskaya L. Development of QSAR models to predict blood-brain barrier permeability. Frontiers in Pharmacology. 2022;13.
144. Grover M, Singh B, Bakshi M, Singh S. Quantitative structure–property relationships in pharmaceutical research – Part 1. Pharmaceutical Science & Technology Today. 2000;3(1):28-35.
145. Storchi L, Cruciani G, Cross S. DeepGRID: Deep Learning Using GRID Descriptors for BBB Prediction. Journal of chemical information and modeling. 2023.
146. Miyazaki Y, Ono N, Huang M, Altaf-Ul-Amin M, Kanaya S. Comprehensive Exploration of Target-specific Ligands Using a Graph Convolution Neural Network. Molecular informatics. 2020;39(1-2):1900095.
147. Stanković T, Dinić J, Podolski-Renić A, Musso L, Burić SS, Dallavalle S, et al. Dual Inhibitors as a New Challenge for Cancer Multidrug Resistance Treatment. Curr Med Chem. 2019;26(33):6074-106.
148. Vo AH, Van Vleet TR, Gupta RR, Liguori MJ, Rao MS. An Overview of Machine Learning and Big Data for Drug Toxicity Evaluation. Chemical Research in Toxicology. 2020;33(1):20-37.
149. Mayr A, Klambauer G, Unterthiner T, Hochreiter S. DeepTox: Toxicity Prediction using Deep Learning. Frontiers in Environmental Science. 2016;3.
150. Tran TTV, Surya Wibowo A, Tayara H, Chong KT. Artificial Intelligence in Drug Toxicity Prediction: Recent Advances, Challenges, and Future Perspectives. Journal of chemical information and modeling. 2023;63(9):2628-43.
151. Ertl P, Schuffenhauer A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. Journal of Cheminformatics. 2009;1(1):8.
152. Coley CW, Rogers L, Green WH, Jensen KF. SCScore: Synthetic Complexity Learned from a Reaction Corpus. Journal of chemical information and modeling. 2018;58(2):252-61.
153. Gao W, Coley CW. The Synthesizability of Molecules Proposed by Generative Models. Journal of chemical information and modeling. 2020;60(12):5714-23.
154. Genheden S, Thakkar A, Chadimová V, Reymond J-L, Engkvist O, Bjerrum E. AiZynthFinder: a fast, robust and flexible open-source software for retrosynthetic planning. Journal of Cheminformatics. 2020;12(1):70.

155. Stanford. CS231n: Convolutional Neural Networks for Visual Recognition. 2018. Available from: https://cs231n.github.io/neural-networks-1/.
156. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Lake Tahoe, Nevada: Curran Associates Inc.; 2012. 1097–105 p.
157. Fukushima K. Cognitron: A self-organizing multilayered neural network. Biological Cybernetics. 1975;20(3):121-36.
158. Nair V, Hinton GE. Rectified linear units improve restricted boltzmann machines. Proceedings of the Proceedings of the 27th international conference on machine learning (ICML-10); 2010.
159. Misra D. Mish: A self regularized non-monotonic activation function. arXiv preprint arXiv:190808681. 2019.
160. Hendrycks D, Gimpel K. Gaussian error linear units (gelus). arXiv preprint arXiv:160608415. 2016.
161. Devlin J, Chang M-W, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:181004805. 2018.
162. Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, et al. Language models are few-shot learners. Advances in neural information processing systems. 2020;33:1877-901.
163. Silver D, Huang A, Maddison CJ, Guez A, Sifre L, van den Driessche G, et al. Mastering the game of Go with deep neural networks and tree search. Nature. 2016;529(7587):484-9.
164. Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, et al. TensorFlow: A system for large-scale machine learning. 2016.
165. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems. 2019;32.
166. Fukushima K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological cybernetics. 1980;36(4):193-202.
167. LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, et al. Backpropagation Applied to Handwritten Zip Code Recognition. Neural Computation. 1989;1(4):541-51.
168. Deng J, Dong W, Socher R, Li LJ, Kai L, Li F-F. ImageNet: A large-scale hierarchical image database. Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition; 2009 20-25 June.
169. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition; 2016.
170. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE. 1998;86(11):2278-324.
171. Yamashita R, Nishio M, Do RKG, Togashi K. Convolutional neural networks: an overview and application in radiology. Insights into Imaging. 2018;9(4):611-29.
172. Koike K. Top breeder. 2018. Available from: https://www.youtube.com/watch?v=f1fXCRtSUWU.
173. Zeiler MD, Fergus R. Visualizing and Understanding Convolutional Networks. Proceedings of the Computer Vision – ECCV 2014; 2014; Cham: Springer International Publishing.
174. Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE. Neural message passing for quantum chemistry. Proceedings of the International conference on machine learning; 2017: PMLR.
175. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:160902907. 2016.
176. Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y. Graph attention networks. arXiv preprint arXiv:171010903. 2017.
177. Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation. 1997;9(8):1735-80.
178. Karpathy A. The Unreasonable Effectiveness of Recurrent Neural Networks. 2015. Available from: http://karpathy.github.io/2015/05/21/rnn-effectiveness/.
179. Burgess J, O'Kane P, Sezer S, Carlin D. LSTM RNN: detecting exploit kits using redirection chain sequences. Cybersecurity. 2021;4(1):25.

180. Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:14061078. 2014.
181. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. Advances in neural information processing systems. 2017;30.
182. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I. Language models are unsupervised multitask learners. OpenAI blog. 2019;1(8):9.
183. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:201011929. 2020.
184. Xiao T, Singh M, Mintun E, Darrell T, Dollár P, Girshick R. Early convolutions help transformers see better. Advances in Neural Information Processing Systems. 2021;34:30392-400.
185. Neimark D, Bar O, Zohar M, Asselmann D. Video transformer network. Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021.
186. Du Z, Zhang G, Lu W, Zhao T, Wu P. Spatio-Temporal Transformer for Online Video Understanding. Journal of Physics: Conference Series. 2022;2171(1):012020.
187. Bottou L, Cortes C, Denker JS, Drucker H, Guyon I, Jackel LD, et al. Comparison of classifier methods: a case study in handwritten digit recognition. Proceedings of the Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol 3 - Conference C: Signal Processing (Cat No94CH3440-5); 1994 9-13 Oct. 1994.
188. Redmon J, Divvala S, Girshick R, Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016 27-30 June 2016.
189. Scao TL, Fan A, Akiki C, Pavlick E, Ilić S, Hesslow D, et al. Bloom: A 176b-parameter open-access multilingual language model. arXiv preprint arXiv:221105100. 2022.
190. Ramesh A, Dhariwal P, Nichol A, Chu C, Chen M. Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:220406125. 2022;1(2):3.
191. Rombach R, Blattmann A, Lorenz D, Esser P, Ommer B. High-resolution image synthesis with latent diffusion models. Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2022.
192. Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models. Advances in neural information processing systems. 2020;33:6840-51.
193. Kryshtafovych A, Schwede T, Topf M, Fidelis K, Moult J. Critical assessment of methods of protein structure prediction (CASP)-Round XIII. Proteins. 2019;87(12):1011-20.
194. Munsamy G, Lindner S, Lorenz P, Ferruz N. ZymCTRL: a conditional language model for the controllable generation of artificial enzymes. Proceedings of the Machine Learning for Structural Biology Workshop NeurIPS 2022; 2022.
195. Stärk H, Ganea O-E, Pattanaik L, Barzilay R, Jaakkola T. EquiBind: Geometric Deep Learning for Drug Binding Structure Prediction. Proceedings of the ICML; 2022.
196. Ganea O-E, Huang X, Bunne C, Bian Y, Barzilay R, Jaakkola T, et al. Independent SE(3)-Equivariant Models for End-to-End Rigid Protein Docking. ArXiv. 2021;abs/2111.07786.
197. Corso G, Stärk H, Jing B, Barzilay R, Jaakkola T. DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking. arXiv preprint arXiv:221001776. 2022.
198. Igashov I, Stärk H, Vignac C, Garcia Satorras V, Frossard P, Welling M, et al. Equivariant 3D-Conditional Diffusion Models for Molecular Linker Design2022 October 01, 2022:[arXiv:2210.05274 p.]. Available from: https://ui.adsabs.harvard.edu/abs/2022arXiv221005274I.
199. Buttenschoen M, Morris GM, Deane CM. PoseBusters: AI-based docking methods fail to generate physically valid poses or generalise to novel sequences. arXiv preprint arXiv:230805777. 2023.
200. Lu W, Wu Q, Zhang J, Rao J, Li C, Zheng S. TANKBind: Trigonometry-Aware Neural NetworKs for Drug-Protein Binding Structure Prediction. bioRxiv. 2022:2022.06.06.495043.

201. Segler MHS, Kogej T, Tyrchan C, Waller MP. Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks. ACS Central Science. 2018;4(1):120-31.
202. Gupta A, Müller AT, Huisman BJH, Fuchs JA, Schneider P, Schneider G. Generative Recurrent Networks for De Novo Drug Design. Molecular informatics. 2018;37(1-2).
203. Gómez-Bombarelli R, Wei JN, Duvenaud D, Hernández-Lobato JM, Sánchez-Lengeling B, Sheberla D, et al. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. ACS Central Science. 2018;4(2):268-76.
204. Zhavoronkov A, Ivanenkov YA, Aliper A, Veselov MS, Aladinskiy VA, Aladinskaya AV, et al. Deep learning enables rapid identification of potent DDR1 kinase inhibitors. Nature Biotechnology. 2019;37(9):1038-40.
205. Prykhodko O, Johansson SV, Kotsias P-C, Arús-Pous J, Bjerrum EJ, Engkvist O, et al. A de novo molecular generation method using latent vector based generative adversarial network. Journal of Cheminformatics. 2019;11(1):74.
206. Davis MI, Hunt JP, Herrgard S, Ciceri P, Wodicka LM, Pallares G, et al. Comprehensive analysis of kinase inhibitor selectivity. Nature Biotechnology. 2011;29(11):1046-51.
207. Xia M, Hu J, Zhang X, Lin X. Drug-Target Binding Affinity Prediction Based on Graph Neural Networks and Word2vec. Proceedings of the Intelligent Computing Theories and Application; 2022; Cham: Springer International Publishing.
208. Tang J, Szwajda A, Shakyawar S, Xu T, Hintsanen P, Wennerberg K, et al. Making Sense of Large-Scale Kinase Inhibitor Bioactivity Data Sets: A Comparative and Integrative Analysis. Journal of chemical information and modeling. 2014;54(3):735-43.
209. He T, Heidemeyer M, Ban F, Cherkasov A, Ester M. SimBoost: a read-across approach for predicting drug–target binding affinities using gradient boosting machines. Journal of Cheminformatics. 2017;9(1):24.
210. Liu T, Lin Y, Wen X, Jorissen RN, Gilson MK. BindingDB: a web-accessible database of experimentally determined protein–ligand binding affinities. Nucleic acids research. 2006;35(suppl_1):D198-D201.
211. Gilson MK, Liu T, Baitaluk M, Nicola G, Hwang L, Chong J. BindingDB in 2015: A public database for medicinal chemistry, computational chemistry and systems pharmacology. Nucleic acids research. 2015;44(D1):D1045-D53.
212. Öztürk H, Özgür A, Ozkirimli E. DeepDTA: deep drug–target binding affinity prediction. Bioinformatics (Oxford, England). 2018;34(17):i821-i9.
213. Pahikkala T, Airola A, Pietilä S, Shakyawar S, Szwajda A, Tang J, et al. Toward more realistic drug–target interaction predictions. Briefings in bioinformatics. 2014;16(2):325-37.
214. Zhao Q, Xiao F, Yang M, Li Y, Wang J. AttentionDTA: prediction of drug–target binding affinity using attention model. Proceedings of the 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM); 2019 18-21 Nov. 2019.
215. Jiang M, Li Z, Zhang S, Wang S, Wang X, Yuan Q, et al. Drug–target affinity prediction using graph neural network and contact maps. RSC Advances. 2020;10(35):20701-12.
216. Hu L, Benson ML, Smith RD, Lerner MG, Carlson HA. Binding MOAD (Mother Of All Databases). Proteins: Structure, Function, and Bioinformatics. 2005;60(3):333-40.
217. Wagle S, Smith RD, Dominic AJ, DasGupta D, Tripathi SK, Carlson HA. Sunsetting Binding MOAD with its last data update and the addition of 3D-ligand polypharmacology tools. Scientific Reports. 2023;13(1):3008.
218. Wang R, Fang X, Lu Y, Yang C-Y, Wang S. The PDBbind Database: Methodologies and Updates. Journal of medicinal chemistry. 2005;48(12):4111-9.
219. Li Y, Liu Z, Li J, Han L, Liu J, Zhao Z, et al. Comparative Assessment of Scoring Functions on an Updated Benchmark: 1. Compilation of the Test Set. Journal of chemical information and modeling. 2014;54(6):1700-16.
220. Cheng T, Li X, Li Y, Liu Z, Wang R. Comparative Assessment of Scoring Functions on a Diverse Test Set. Journal of chemical information and modeling. 2009;49(4):1079-93.

221. Yang Z, Zhong W, Lv Q, Dong T, Yu-Chian Chen C. Geometric Interaction Graph Neural Network for Predicting Protein–Ligand Binding Affinities from 3D Structures (GIGN). The Journal of Physical Chemistry Letters. 2023;14(8):2020-33.
222. Dunbar JB, Jr., Smith RD, Yang C-Y, Ung PM-U, Lexa KW, Khazanov NA, et al. CSAR Benchmark Exercise of 2010: Selection of the Protein–Ligand Complexes. Journal of chemical information and modeling. 2011;51(9):2036-46.
223. Dunbar JB, Jr., Smith RD, Damm-Ganamet KL, Ahmed A, Esposito EX, Delproposto J, et al. CSAR Data Set Release 2012: Ligands, Affinities, Complexes, and Docking Decoys. Journal of chemical information and modeling. 2013;53(8):1842-52.
224. Carlson HA, Smith RD, Damm-Ganamet KL, Stuckey JA, Ahmed A, Convery MA, et al. CSAR 2014: A Benchmark Exercise Using Unpublished Data from Pharma. Journal of chemical information and modeling. 2016;56(6):1063-77.
225. Gathiaka S, Liu S, Chiu M, Yang H, Stuckey JA, Kang YN, et al. D3R grand challenge 2015: Evaluation of protein–ligand pose and affinity predictions. Journal of Computer-Aided Molecular Design. 2016;30(9):651-68.
226. Gaieb Z, Liu S, Gathiaka S, Chiu M, Yang H, Shao C, et al. D3R Grand Challenge 2: blind prediction of protein–ligand poses, affinity rankings, and relative binding free energies. Journal of Computer-Aided Molecular Design. 2018;32(1):1-20.
227. Gaieb Z, Parks CD, Chiu M, Yang H, Shao C, Walters WP, et al. D3R Grand Challenge 3: blind prediction of protein–ligand poses and affinity rankings. Journal of Computer-Aided Molecular Design. 2019;33(1):1-18.
228. Parks CD, Gaieb Z, Chiu M, Yang H, Shao C, Walters WP, et al. D3R grand challenge 4: blind prediction of protein–ligand poses, affinity rankings, and relative binding free energies. Journal of Computer-Aided Molecular Design. 2020;34(2):99-119.
229. Hartshorn MJ, Verdonk ML, Chessari G, Brewerton SC, Mooij WTM, Mortenson PN, et al. Diverse, High-Quality Test Set for the Validation of Protein−Ligand Docking Performance. Journal of medicinal chemistry. 2007;50(4):726-41.
230. Wang L, Wu Y, Deng Y, Kim B, Pierce L, Krilov G, et al. Accurate and Reliable Prediction of Relative Ligand Binding Potency in Prospective Drug Discovery by Way of a Modern Free-Energy Calculation Protocol and Force Field. Journal of the American Chemical Society. 2015;137(7):2695-703.
231. Hassan-Harrirou H, Zhang C, Lemmin T. RosENet: Improving Binding Affinity Prediction by Leveraging Molecular Mechanics Energies with an Ensemble of 3D Convolutional Neural Networks. Journal of chemical information and modeling. 2020;60(6):2791-802.
232. Karlov DS, Sosnin S, Fedorov MV, Popov P. graphDelta: MPNN Scoring Function for the Affinity Prediction of Protein–Ligand Complexes. ACS Omega. 2020;5(10):5150-9.
233. Zhu F, Zhang X, Allen JE, Jones D, Lightstone FC. Binding Affinity Prediction by Pairwise Function Based on Neural Network. Journal of chemical information and modeling. 2020;60(6):2766-72.
234. Kwon Y, Shin W-H, Ko J, Lee J. AK-Score: Accurate Protein-Ligand Binding Affinity Prediction Using an Ensemble of 3D-Convolutional Neural Networks. International Journal of Molecular Sciences. 2020;21(22):8424.
235. Cang Z, Wei G-W. TopologyNet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions. PLOS Computational Biology. 2017;13(7):e1005690.
236. Gomes J, Ramsundar B, Feinberg EN, Pande VS. Atomic convolutional networks for predicting protein-ligand binding affinity. arXiv preprint. 2017.
237. Lau T, Dror RO. Brendan-A Deep Convolutional Network for Representing Latent Features of Protein-Ligand Binding Poses. Proceedings of the 2017.
238. Li Y, Rezaei MA, Li C, Li X. DeepAtom: A Framework for Protein-Ligand Binding Affinity Prediction. IEEE International Conference on Bioinformatics and Biomedicine (BIBM). 2019:303-10.
239. Zhang H, Liao L, Saravanan KM, Yin P, Wei Y. DeepBindRG: a deep learning based method for estimating effective protein–ligand affinity. PeerJ. 2019;7:e7362.

240. Wang S, Liu D, Ding M, Du Z, Zhong Y, Song T, et al. SE-OnionNet: A Convolution Neural Network for Protein–Ligand Binding Affinity Prediction. Frontiers in Genetics. 2021;11.
241. Xie L, Xu L, Chang S, Xu X, Meng L. Multitask deep networks with grid featurization achieve improved scoring performance for protein–ligand binding. Chemical Biology & Drug Design. 2020;96(3):973-83.
242. Yang J, Shen C, Huang N. Predicting or Pretending: Artificial Intelligence for Protein-Ligand Interactions Lack of Sufficiently Large and Unbiased Datasets. Frontiers in Pharmacology. 2020;11(69).
243. Ahmed A, Mam B, Sowdhamini R. DEELIG: A Deep Learning Approach to Predict Protein-Ligand Binding Affinity. Bioinformatics and biology insights. 2021;15.
244. Jiang D, Hsieh C-Y, Wu Z, Kang Y, Wang J, Wang E, et al. InteractionGraphNet: A Novel and Efficient Deep Graph Representation Learning Framework for Accurate Protein–Ligand Interaction Predictions. Journal of medicinal chemistry. 2021;64(24):18209-32.
245. Jones D, Kim H, Zhang X, Zemla A, Stevenson G, Bennett WFD, et al. Improved Protein–Ligand Binding Affinity Prediction with Structure-Based Deep Fusion Inference. Journal of chemical information and modeling. 2021;61(4):1583-92.
246. Kumar S, Kim M-h. SMPLIP-Score: predicting ligand binding affinity from simple and interpretable on-the-fly interaction fingerprint pattern descriptors. Journal of Cheminformatics. 2021;13(1):28.
247. Seo S, Choi J, Park S, Ahn J. Binding affinity prediction for protein–ligand complex using deep attention mechanism based on intermolecular interactions. BMC Bioinformatics. 2021;22(1):542.
248. Shen H, Zhang Y, Zheng C, Wang B, Chen P. A Cascade Graph Convolutional Network for Predicting Protein–Ligand Binding Affinity. International Journal of Molecular Sciences. 2021;22(8):4023.
249. Artemenko N. Distance Dependent Scoring Function for Describing Protein−Ligand Intermolecular Interactions. Journal of chemical information and modeling. 2008;48(3):569-74.
250. Ashtawy HM, Mahapatra NR. BgN-Score and BsN-Score: Bagging and boosting based ensemble neural networks scoring functions for accurate binding affinity prediction of protein-ligand complexes. BMC Bioinformatics. 2015;16(4):S8.
251. Hassan M, Mogollon DC, Fuentes O. DLSCORE: A deep learning model for predicting protein-ligand binding affinities. ChemRxiv preprint. 2018.
252. Meli R, Anighoro A, Bodkin MJ, Morris GM, Biggin PC. Learning protein-ligand binding affinity with atomic environment vectors. Journal of Cheminformatics. 2021;13(1):59.
253. Cang Z, Mu L, Wei G-W. Representability of algebraic topology for biomolecules in machine learning based scoring and virtual screening. PLOS Computational Biology. 2018;14(1):e1005929.
254. Francoeur PG, Masuda T, Sunseri J, Jia A, Iovanisci RB, Snyder I, et al. Three-Dimensional Convolutional Neural Networks and a Cross-Docked Data Set for Structure-Based Drug Design. Journal of chemical information and modeling. 2020;60(9):4200-15.
255. Azzopardi J, Ebejer JP. LigityScore: A CNN-Based Method for Binding Affinity Predictions. Proceedings of the 2022; Cham: Springer International Publishing.
256. Wang Z, Zheng L, Liu Y, Qu Y, Li Y-Q, Zhao M, et al. OnionNet-2: a convolutional neural network model for predicting protein-ligand binding affinity based on residue-atom contacting shells. Frontiers in chemistry. 2021:913.
257. Li S, Zhou J, Xu T, Huang L, Wang F, Xiong H, et al. Structure-Aware Interactive Graph Neural Networks for the Prediction of Protein-Ligand Binding Affinity. Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining; 2021 August 14 - 18.
258. Moesser MA, Klein D, Boyles F, Deane CM, Baxter A, Morris GM. Protein-ligand interaction graphs: Learning from ligand-shaped 3d interaction graphs to improve binding affinity prediction. bioRxiv. 2022:2022.03. 04.483012.
259. Berishvili VP, Perkin VO, Voronkov AE, Radchenko EV, Syed R, Venkata Ramana Reddy C, et al. Time-Domain Analysis of Molecular Dynamics Trajectories Using Deep Neural Networks: Application to Activity Ranking of Tankyrase Inhibitors. Journal of chemical information and modeling. 2019;59(8):3519-32.

260. Durrant JD, McCammon JA. BINANA: a novel algorithm for ligand-binding characterization. Journal of molecular graphics & modelling. 2011;29(6):888-93.
261. Hu J, Shen L, Sun G. Squeeze-and-excitation networks. Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition; 2018.
262. Wallach I, Dzamba M, Heifets A. AtomNet: A Deep Convolutional Neural Network for Bioactivity Prediction in Structure-based Drug Discovery. arXiv e-prints. 2015:arXiv:1510.02855.
263. Mysinger MM, Carchia M, Irwin JJ, Shoichet BK. Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. Journal of medicinal chemistry. 2012;55(14):6582-94.
264. Iandola FN, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. arXiv preprint arXiv:160207360. 2016.
265. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:14091556. 2014.
266. Jiménez-Luna J, Pérez-Benito L, Martínez-Rosell G, Sciabola S, Torella R, Tresadern G, et al. DeltaDelta neural networks for lead optimization of small molecule potency. Chem Sci. 2019;10(47):10911-8.
267. Hochuli J, Helbling A, Skaist T, Ragoza M, Koes DR. Visualizing convolutional neural network protein-ligand scoring. Journal of Molecular Graphics and Modelling. 2018;84:96-108.
268. Ragoza M, Hochuli J, Idrobo E, Sunseri J, Koes DR. Protein-Ligand Scoring with Convolutional Neural Networks. Journal of chemical information and modeling. 2017;57(4):942-57.
269. Jiménez-Luna J, Grisoni F, Schneider G. Drug discovery with explainable artificial intelligence. Nature Machine Intelligence. 2020;2(10):573-84.
270. Tosstorff A, Rudolph MG, Cole JC, Reutlinger M, Kramer C, Schaffhauser H, et al. A high quality, industrial data set for binding affinity prediction: performance comparison in different early drug discovery scenarios. Journal of Computer-Aided Molecular Design. 2022;36(10):753-65.
271. Brocidiacono M, Francoeur P, Aggarwal R, Popov K, Koes D, Tropsha A. BigBind: Learning from Nonstructural Data for Structure-Based Virtual Screening. ChemRxiv preprint. 2022.
272. Vogel SM, Bauer MR, Boeckler FM. DEKOIS: demanding evaluation kits for objective in silico screening--a versatile tool for benchmarking docking programs and scoring functions. Journal of chemical information and modeling. 2011;51(10):2650-65.
273. Rohrer SG, Baumann K. Maximum Unbiased Validation (MUV) Data Sets for Virtual Screening Based on PubChem Bioactivity Data. Journal of chemical information and modeling. 2009;49(2):169-84.
274. Orengo CA, Michie AD, Jones S, Jones DT, Swindells MB, Thornton JM. CATH--a hierarchic classification of protein domain structures. Structure (London, England : 1993). 1997;5(8):1093-108.
275. Min Y, Wei Y, Wang P, Wu N, Bauer S, Zheng S, et al. Predicting the protein-ligand affinity from molecular dynamics trajectories. arXiv preprint arXiv:220810230. 2022.
276. Case DA, Cheatham Iii TE, Darden T, Gohlke H, Luo R, Merz Jr KM, et al. The Amber biomolecular simulation programs. Journal of Computational Chemistry. 2005;26(16):1668-88.
277. Case DA, Aktulga HM, Belfon K, Cerutti DS, Cisneros GA, Cruzeiro VWD, et al. AmberTools. Journal of chemical information and modeling. 2023;63(20):6183-91.
278. Zadka M. Paramiko. In: Zadka M, editor. DevOps in Python: Infrastructure as Python. Berkeley, CA: Apress; 2019. p. 111-9.
279. Nguyen H, Roe DR. Pytraj. 2016. Available from: https://github.com/Amber-MD/pytraj.
280. Gowers RJ, Linke M, Barnoud J, Reddy TJ, Melo MN, Seyler SL, et al. MDAnalysis: a Python package for the rapid analysis of molecular dynamics simulations. Proceedings of the Proceedings of the 15th python in science conference; 2016: SciPy Austin, TX.
281. Li H, Leung K-S, Wong M-H, Ballester PJ. Correcting the impact of docking pose generation error on binding affinity prediction. BMC Bioinformatics. 2016;17(11):308.
282. Boyles F, Deane CM, Morris GM. Learning from Docked Ligands: Ligand-Based Features Rescue Structure-Based Scoring Functions When Trained on Docked Poses. Journal of chemical information and modeling. 2022;62(22):5329-41.

283. Pérez A, Martínez-Rosell G, De Fabritiis G. Simulations meet machine learning in structural biology. Current opinion in structural biology. 2018;49:139-44.
284. Korlepara DB, Vasavi CS, Jeurkar S, Pal PK, Roy S, Mehta S, et al. PLAS-5k: Dataset of Protein-Ligand Affinities from Molecular Dynamics for Machine Learning Applications. Scientific Data. 2022;9(1):548.
285. Sali A, Blundell TL. Comparative protein modelling by satisfaction of spatial restraints. Journal of molecular biology. 1993;234(3):779-815.
286. Gordon JC, Myers JB, Folta T, Shoja V, Heath LS, Onufriev A. H++: a server for estimating pKas and adding missing hydrogens to macromolecules. Nucleic acids research. 2005;33(Web Server issue):W368-71.
287. Siebenmorgen T, Menezes F, Benassou S, Merdivan E, Kesselheim S, Piraud M, et al. MISATO - Machine learning dataset for structure-based drug discovery. bioRxiv. 2023:2023.05.24.542082.
288. Gu S, Shen C, Yu J, Zhao H, Liu H, Liu L, et al. Can molecular dynamics simulations improve predictions of protein-ligand binding affinity with machine learning? Briefings in bioinformatics. 2023;24(2).
289. Jamal S, Grover A, Grover S. Machine Learning From Molecular Dynamics Trajectories to Predict Caspase-8 Inhibitors Against Alzheimer's Disease. Frontiers in Pharmacology. 2019;10.
290. Wu F, Jin S, Jiang Y, Jin X, Tang B, Niu Z, et al. Pre-Training of Equivariant Graph Matching Networks with Conformation Flexibility for Drug Binding. Advanced Science. 2022;9(33):2203796.
291. Townshend RJ, Vögele M, Suriana P, Derry A, Powers A, Laloudakis Y, et al. Atom3d: Tasks on molecules in three dimensions. arXiv preprint arXiv:201204035. 2020.
292. Word JM, Lovell SC, Richardson JS, Richardson DC. Asparagine and glutamine: using hydrogen atom contacts in the choice of side-chain amide orientation. Journal of molecular biology. 1999;285(4):1735-47.
293. Rice P, Longden I, Bleasby A. EMBOSS: The European Molecular Biology Open Software Suite. Trends in Genetics. 2000;16(6):276-7.
294. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. Journal of molecular biology. 1990;215(3):403-10.
295. Jakalian A, Bush BL, Jack DB, Bayly CI. Fast, efficient generation of high-quality atomic charges. AM1-BCC model: I. Method. Journal of Computational Chemistry. 2000;21(2):132-46.
296. Wang J, Wang W, Kollman PA, Case DA. Antechamber: an accessory software package for molecular mechanical calculations. J Am Chem Soc. 2001;222(1).
297. Corina software. Available from: https://mn-am.com/products/corina/.
298. Roe DR, Cheatham TE, III. PTRAJ and CPPTRAJ: Software for Processing and Analysis of Molecular Dynamics Trajectory Data. Journal of Chemical Theory and Computation. 2013;9(7):3084-95.
299. McGibbon RT, Beauchamp KA, Harrigan MP, Klein C, Swails JM, Hernández CX, et al. MDTraj: A Modern Open Library for the Analysis of Molecular Dynamics Trajectories. Biophys J. 2015;109(8):1528-32.
300. CRIANN. MYRIA computational power in 2020. Available from: https://www.criann.fr/docs/0/pres/Fiche-Myria-2020.pdf.
301. IDRIS. Jean Zay computational power in 2022.
302. Anaconda Software Distribution. 2012. Available from: https://anaconda.com.
303. Merkel D. Docker: lightweight Linux containers for consistent development and deployment. Linux J. 2014;2014(239):Article 2.
304. Kurtzer GM, Sochat V, Bauer MW. Singularity: Scientific containers for mobility of compute. PLOS ONE. 2017;12(5):e0177459.
305. Yoo AB, Jette MA, Grondona M. SLURM: Simple Linux Utility for Resource Management. Proceedings of the Job Scheduling Strategies for Parallel Processing; 2003 2003//; Berlin, Heidelberg: Springer Berlin Heidelberg.
306. Nvidia docker. Available from: https://developer.nvidia.com/blog/gpu-containers-runtime/.
307. Github. Available from: https://github.com/.

308. Yadan O. Hydra - A framework for elegantly configuring complex applications. Github. 2019. Available from: https://github.com/facebookresearch/hydra.
309. Zaharia M, Chen A, Davidson A, Ghodsi A, Hong SA, Konwinski A, et al. Accelerating the machine learning lifecycle with MLflow. IEEE Data Eng Bull. 2018;41(4):39-45.
310. Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980. 2014.
311. Loshchilov I, Hutter F. Decoupled weight decay regularization. arXiv preprint arXiv:171105101. 2017.
312. Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. Journal of machine learning research. 2011;12(7).
313. Huang G, Li Y, Pleiss G, Liu Z, Hopcroft JE, Weinberger KQ. Snapshot ensembles: Train 1, get m for free. arXiv preprint arXiv:170400109. 2017.
314. Guterres H, Im W. Improving Protein-Ligand Docking Results with High-Throughput Molecular Dynamics Simulations. Journal of chemical information and modeling. 2020;60(4):2189-98.
315. Liu K, Kokubo H. Prediction of ligand binding mode among multiple cross-docking poses by molecular dynamics simulations. Journal of Computer-Aided Molecular Design. 2020;34(11):1195-205.
316. Volkov M. Design and application of deep learning methods to structure-based drug design 2023.
317. Fraccalvieri D, Pandini A, Stella F, Bonati L. Conformational and functional analysis of molecular dynamics trajectories by Self-Organising Maps. BMC Bioinformatics. 2011;12(1):158.
318. Mallet V, Checa Ruano L, Moine Franel A, Nilges M, Druart K, Bouvier G, et al. InDeep: 3D fully convolutional neural networks to assist in silico drug design on protein–protein interactions. Bioinformatics (Oxford, England). 2021;38(5):1261-8.
319. Olsson MHM, Søndergaard CR, Rostkowski M, Jensen JH. PROPKA3: Consistent Treatment of Internal and Surface Residues in Empirical pKa Predictions. Journal of Chemical Theory and Computation. 2011;7(2):525-37.
320. Gokcan H, Isayev O. Prediction of protein pKa with representation learning. Chemical Science. 2022;13(8):2462-74.
321. Zankov DV, Matveieva M, Nikonenko AV, Nugmanov RI, Baskin II, Varnek A, et al. QSAR Modeling Based on Conformation Ensembles Using a Multi-Instance Learning Approach. Journal of chemical information and modeling. 2021;61(10):4913-23.
322. Weiler M, Geiger M, Welling M, Boomsma W, Cohen T. 3D Steerable CNNs: Learning Rotationally Equivariant Features in Volumetric Data. Proceedings of the NeurIPS; 2018.

**Pierre-Yves LIBOUBAN**

# Prédiction de l'affinité de liaison des complexes protéine-ligand en combinant des simulations de dynamique moléculaire avec des algorithmes d'apprentissage profond

Résumé :

Les interactions des petites molécules avec leurs protéines cibles sont essentielles à la recherche pharmaceutique. L'affinité de liaison des complexes protéine-ligand peut être mesurée par des expériences *in vitro*, mais ces tests sont couteux en argent et en temps. Aujourd'hui, les réseaux de neurones profonds utilisant les structures en trois dimensions des complexes sont capables de prédire cette affinité de liaison. Cependant, des limitations persistent malgré l'implémentation de nouveaux réseaux de neurones. Ceci est principalement dû au manque de données structurales, qui nécessitent un travail expérimental conséquent pour être déterminées. Ce projet vise à améliorer notre capacité à prédire l'affinité des complexes protéine-ligand en combinant des approches d'apprentissage profond avec des simulations de dynamique moléculaire. Ainsi, il est possible d'augmenter la quantité de données utilisées lors de l'apprentissage des modèles statistiques, en extrayant des structures supplémentaires des simulations de dynamique moléculaire. En outre, celles-ci fournissent des informations temporelles sur les interactions protéine-ligand qui peuvent être utilisées pour améliorer les modèles. Nous avons créé un ensemble de données de 63 000 simulations, obtenues à partir de 6 300 complexes. Puis nous avons développé des réseaux de neurones, tel que le LSTM à convolutions, capables d'analyser à la fois les informations spatiales et temporelles issues des simulations. Ces réseaux combinent un réseau de neurones à convolutions capable d'extraire l'information spatiale des structures en trois dimensions à chaque pas de temps, tandis que le LSTM suit l'évolution de cette information sur l'ensemble de la simulation. En utilisant les simulations de dynamique moléculaire en tant qu'augmentation de données, nos modèles obtiennent des résultats prometteurs.

Mots clés : apprentissage profond, dynamique moléculaire, affinité de liaison, intelligence artificielle, complexe protéine-ligand, conception de médicament

# Protein-Ligand binding affinity prediction using combined molecular dynamics simulations and deep learning algorithms

Summary:

Interactions of small molecules with their target proteins are essential to pharmaceutical research. *In vitro* experiments were developed to measure the binding affinity of protein-ligand complexes, but they remain long and expensive. Nowadays deep neural networks (NN) use the three-dimensional structures of complexes to predict their binding affinity. However, limitations persist despite the implementation of new NN, primarily due to scarce structural data, necessitating extensive experimental work. This project aims to improve our ability to predict the affinity of protein-ligand complexes by combining deep learning and molecular dynamics (MD) simulations. Data augmentation can be achieved by extracting additional structures from MD simulations. Furthermore, MD simulations provide temporal insights into protein-ligand interactions that can be used to improve models. We created a dataset of 63,000 simulations, obtained from 6,300 complexes. To create efficient statistical models by learning from these MD simulations, we developed NNs able to analyse both spatial and temporal information, like the convolutional LSTM. These NNs combine a convolutional NN able to extract the spatial information from the three-dimensional structures at each time step, while the LSTM keeps track of the evolution of this information over the whole simulation. Using molecular dynamics data augmentation, we are obtaining promising results.

Keywords: deep learning, molecular dynamics, binding affinity, artificial intelligence, protein-ligand complexes, drug design

**Institut de Chimie Organique et Analytique**
**UMR 7311 CNRS Université d'Orléans**
**Pôle de chimie, rue de Chartres**
**45067 Orléans**