Università degli Studi di Palermo

AREA QUALITÀ, PROGRAMMAZIONE E SUPPORTO STRATEGICO
SETTORE STRATEGIA PER LA RICERCA
U. O. DOTTORATI

UNIVERSITÀ DEGLI STUDI DI PALERMO

Dottorato in Ingegneria dell'innovazione tecnologica- Ciclo XXXIII
Dipartimento dell'Innovazione Industriale e Digitale
Laboratorio di interazione Uomo-macchina

Deep neural networks leveraging different arrangements of molecular fingerprints to define a novel embedding for virtual screening procedure

Ph.D Candidate:
Isabella Mendolia

Ph.D. Coordinator:
Prof. **Salvatore Gaglio**

Supervisor:
Prof. **Roberto Pirrone**

Co-Supervisor:
Dr. **Ugo Perricone**

CICLO XXXIII Anno Accademico 2019/2020

Contents

List of Figures

List of Tables

Nomenclature

Acronyms

ACK	Tyrosine kinase non-receptor protein 2
ADMET	Absorption, Distribution, Metabolism, Excretion and Toxicity
AI	Artificial Intelligence
ALK	ALK tyrosine kinase receptor
AUC	Area Under the Curve
*b*ACC	Balanced Accuracy
CADD	Computer-Aided Drug Design
CDKs	Cycline-Dependent Kinases
CDK1	Cycline-Dependent Kinase1
CDK2	Cyclin-dependent kinase 2
CDK6	Cyclin-dependent kinase 6
CETSA	cellular ThermoStabilty Assay
CHK1	Serine/threonine-protein kinase
CK2A1	Casein kinase II alpha
CLK2	Dual specificity protein kinase
cMaP	Connectivity Map
CNN	Convolutional Neural Network
DTBA	Drug target binding affinity
DL	Deep Learning
DNN	Deep Neural Network
DR	Drug Repositioning
DYRK1A	Dual-specificity tyrosine-phosphorylation regulated kinase 1A
ECFPs	Extended-connectivity fingerprints
EF	Enrichment Factor
EGFR	Epidermal growth factor receptor erbB1
EHR	Electronic Health Record
EMA	European Medicines Agency
ENL	Erythema nodosum leprosum
ERK2	MAP kinase
GSK3	Glycogen synthase kinase-3 beta
GWAS	Genome-Wide Association Study

IC50	Half Maximal Inhibitory Concentration
IFP	Interaction fingerprints
INSR	Insulin receptor
IRAK4	Interleukin-1 receptor-associated kinase 4
ITK	Tyrosine-protein kinase ITK/TSK
JAK2	Tyrosine-protein kinase JAK2
JNK3	c-Jun N-terminal kinase 3
MAP2K1	Dual specificity mitogen-activated protein kinase kinase 1
MCC	Matthews Correlation Coefficient
MELK	Maternal embryonic leucine zipper kinase
ML	Machine Learning
MLP	Multi Layer Perceptron
MPNN	Message passing neural networks
N	Negative
NN	Neural Network
P	Positive
PDK1	Pyruvate dehydrogenase kinase isoform 1
RAS	Rat sarcoma virus
ReLU	Rectified Linear Unit
RF	Random Forest
RNN	Recurrent Neural Network
SVM	Support Vector Machines
SMILES	Simplified Molecular Input Line Entry System
TN	True Negative
TP	True Positive
QSAR	Quantitative Structure-Activity Relationship
SHAP	SHapley Additive exPlanations
VAE	Variational Autoencoder
VS	Virtual screening
WCSS	Within Cluster Sum of Squares
WL	Weisfeiler-Lehman

Chapter 1

Introduction

1.1 Overview

Biological systems are complex sources of information. This information is now being systematically measured and extracted at unprecedented levels using a plethora of "omics" and intelligent technologies.
The advent of these approaches to biology and disease presents both challenges and opportunities for the pharmaceutical industry, whose goal is to identify plausible therapeutic hypotheses from which to develop drugs. However, recent advances in a number of factors have led to increased interest in the use of machine learning (ML) approaches within the pharmaceutical industry.

Coupled with infinitely scalable storage, the large increase in the types and sizes of data sets that can provide the basis for ML has allowed pharmaceutical companies to access and organize much more data. Data types can include images, textual information, biometric and other information from wearables, analytics information, and high-dimensional data.
In recent years, the field of artificial intelligence (AI) has moved from largely theoretical studies to real-world applications. Much of this explosive growth has to do with the wide availability of new hardware such as graphics processing units (GPUs) that make parallel processing faster, especially in numerically intensive computations. More recently, advances in new ML algorithms, such as deep learning (DL), have contributed to a tremendous increase in ML applications within pharmaceutical companies over the past 2

years.

My research activity, during the three years of my PhD course, has been conducted in concert with the team of the Human Computer Interaction Laboratory of the Engineering Department of the University of Palermo under the supervision of Prof. Roberto Pirrone and the Molecular Informatics Group of the Ri.MED Foundation directed by Dr. Ugo Perricone. The research has focused on the analysis and definition of deep neural networks using different organizations of molecular fingerprints; this activity has led to the definition of a new embedding that can be used in virtual screening (VS) procedures that consist in tasks of classification of small molecules with respect to their bio-activity on one or more protein targets.

The general objective of this thesis is the creation of a virtual screening procedure, aimed at identifying molecules potentially bioactive on one or more proteins belonging to the family of cyclin-dependent kinases (CDKs). To do this, I started from cyclin-dependent kinase 1 (CDK1) and then expand the study to other 19 proteins belonging to CDKs with the particularity of having a similarity of the binding site greater than 80%. The choice of this target is given by the previous experience of the research group at Fondazione Ri.MED in CDK1 modulators and activity because of the high structural similarity between the binding sites of the different kinases. Secondly I chose to start from CDK1 because it is the main regulator that guides cells through G2 phase and mitosis, so it is particularly involved in cancer development. Furthermore, the fact that, unlike other CDKs, loss of CDK1 in the liver confers complete resistance to tumor formation, revealing its role in cancer development, demonstrates its importance in carcinogenesis. My research project was aimed at finding an appropriate molecular descriptor for a convolutional neural classifier. In first phase of the study, oriented to the identification of the most performing fingerprint, I built networks to classify as active or inactive molecules on a single target, CDK1. The classification architectures I proposed are convolutional networks (CNNs) that use input tensors consist-

ing of combinations of molecular fingerprints.

Molecular fingerprints are vectors of bits of fixed length that are generated following well-defined rules. They are generated by analyzing each atom together with its neighborhood up to 6 or 7 bonds apart. For each atom the so called patterns are analyzed, i.e. type of atom, bond, presence of aromatic rings and so on. After enumerating all patterns, each of them is used as a seed for the hashing function that generates 4/5 position indexes where to allocate bits. The scientific literature reports several types of fingerprint to describe different aspects of both the structure and local properties of a molecule. The same fingerprint can also be designed with different dimensions. Specifically, I used Knime software for the generation of molecular fingerprints. In the early stages of the work, I selected the length and type of molecular fingerprints and I tested them using different classification techniques including *shallow* techniques such as Random Forest and Support Vector Machine.

I first tested individual fingerprints at different lengths using one-dimensional CNNs to identify the best performing length. The results of this study I presented at the 2019 *BITS, Bionformatics Italian Society bioinformatics* conference held in Palermo, Italy.

One of the most frequently asked questions by computational chemists, is whether it is better to have a model that recovers some false positives or lose some assets as false negatives. Depending on the stage of drug discovery, at the beginning of the drug discovery cascade, it may be beneficial to have some false positives instead of losing some putative hits. At a more mature screening stage (e.g., expansion of hits), it might be better, however, to have a more precise algorithm that prevents false positive discovery. Based on these considerations, the best compromise is a virtual screening model that can adapt to the drug design phase of the campaign.

In this part of the work, I present different CNN architectures for VS of candidate compounds with respect to their biological activity at first on the CDK1 target. The vector representation of the

candidate compounds is obtained using their molecular fingerprints.

The importance of the target lies in its validation as a pharmacological target. It is an archetypal kinase that acts as a central regulator guiding cells through G2 phase and mitosis. Its importance in tumorigenesis has been demonstrated by evidence that, unlike other CDKs, loss of CDK1 in the liver confers complete resistance against tumor formation, demonstrating its role in cancer development [32]. Initially, there were two very favorable points:

- molecular fingerprints are used as a suitable embedding to describe molecules;
- a unique neural architecture was designed and trained with several hyper-parameters to achieve good performance in both initial and mature screening.

In a second step, I performed tests on all possible combinations of fingerprints both in number and in type through two-dimensional CNNs that I trained on appropriate combinations of different fingerprints of equal size for the same compound, to take into account all the different information coming from these descriptors at once. Different types of fingerprints with different sizes are reported in the scientific literature to address different aspects of both the structure and local properties of a molecule [21]. In my work, I addressed seven of the most popular types of fingerprints: RDKit, Morgan, AtomPair, Torsion, Layered, FeatMorgan, ECFP4. A substantial portion of this work has been devoted to addressing the single fingerprint or combination of fingerprints that achieves the best accuracy in both a highly discriminative task (i.e., mature screening) and an active-only selection task (i.e., early screening).

A molecular fingerprint represents the corresponding molecule "as a whole" in a suitable vector form, i.e., it conveys information about the presence of a particular substructure, but not about its exact location or its repetition at different sites of the same molecule. Furthermore, I aimed to perform a binary classification between active and inactive compounds, and biological activity is mostly related to the presence/absence of particular substructures that in turn are adapted to bind to the target protein. Consequently, CNNs seem to

be the best architectural choice to classify molecular fingerprints. A single fingerprint may not make explicit the particular substructure that is responsible for binding to the target, and this is due to both its search strategy and hashing mechanism.

Several types of fingerprints are reported in the scientific literature to address different aspects of both the structure and local properties of a molecule. In the chapter 2.4, the comparison between algorithms and substructures of different fingerprint types has been reported. The key idea of this work is that many fingerprints used together to describe the same candidate compound can make explicit the features responsible for bioactivity. Moreover, a neural model with adequate capacity can accommodate the redundancy derived from having the same molecular pattern encoded in different fingerprints. In light of the above considerations, both 1D and 2D CNNs were trained to test the performance of each descriptor separately, along with all combinations of multiple descriptors for the same compound. I presented the results of this study at the 2019 *ICIAP, International Conference on Image Analysis and Processing* conference held in Trento, Italy.

Next, I focused my attention on the optimal use of fingerprint combinations to increase the performance of a neural classifier starting from the assumption that different fingerprints describe the same molecule as if they were different "spectra" of the molecule since they are generated following computationally similar procedures, but from information collected differently along the molecular structure.
The first architecture I proposed, called Tuned MLP-Out, uses a parameter sharing concept in which several parallel 1D CNN branches are trained on single fingerprints and then merged into a single network. This study was published in the Q1 computer science journal *BMC Bionformatics*.

At the end of my PhD course, I proposed EMBER, a novel embedding for molecular structures that allows for explainability in a multi-target VS task. EMBER is composed of seven molecular fingerprints arranged as the different channels of a one-dimensional

tensor and exploits the Depthwise Separable Convolution operator to reduce the computational complexity. In this way, the different fingerprints immediately act as different features of the input tensor.

To support the efficacy of the new embedding, I developed an architecture that performs multi-class multi-label classification of single molecules against 20 different targets that have been selected as the most structurally similar kinases to CDK1. More specifically, this network uses depth-separable layers that benefit from using channels to see different types of molecular fingerprints as different spectra of the same molecule.

As it is well known, Explainable AI (XAI) [60] aims to provide a description of how the model uses features to build its predictions, and this is a crucial argument to make feasible an extended use of neural models in the general context of life sciences. In view of the previous considerations, I focused on the prioritization of actives and the use of the well-known SHAP XAI framework to provide a deeper description of how the model uses fingerprints and to analyze my trained network to provide an explanation of the role of each fingerprint in my embedding. Moreover, from this study it was possible to assess which fingerprints affect mostly the classification either positively or negatively.

This work has been published in the Q1 computer science journal *International Journal of Molecular Sciences*.

Lastly, a drug repurposing problem for COVID-19 has also been treated. Especially I tried to use the same techniques utilized for virtual screening task. The technique seems to be promising despite of the shortage of public data (probably even private ones), and it doesn't perform very well. Preliminary results are reported in section 2.3.

Chapter 2

State of the art

Artificial intelligence (AI) is becoming more widely used in different sectors of society, including the pharmaceutical industry. In this chapter, I look at how AI is being used in various areas of the pharmaceutical industry, such as drug discovery and development, drug repurposing, increasing pharmaceutical productivity, and clinical trials, to name a few. This use reduces human workload while also achieving targets in a short amount of time. I also talk about how AI tools and methodologies interact, as well as current issues and solutions, and the future of AI in the pharmaceutical business. Different applications of AI in drug discovery are depicted in Figure 2.1 that show how AI can be used effectively in different parts of drug discovery, including drug design, chemical synthesis, drug screening, polypharmacology, and drug repurposing.

The number of available compound and biomedical activity data has increased significantly over the past decade [50],[76]. ChEMBL, a comprehensive life science information resource for public chemical facilities and biological activity databases, is one of the most commonly used databases. Another is PubChem, which is a public database of chemical information and biological activities. Both give a plethora of information for medication development and medical research.

Many medications have traditionally been discovered by chance; nevertheless, the development of new computer-assisted enabling technologies is maturing, and computational performance is getting more powerful. Computer-Aided Drug Design (CADD) has grown in importance as a method of drug development. CADD is a drug discovery technology that uses target structures, functional quali-

Figure 2.1: Role of artificial intelligence (AI) in drug discovery [77].

ties, and processes to find new drugs [43]. CADD employs in silico tests to compare the expected and actual activity of a medicine in a cost-effective and timely manner, with the data being utilized iteratively to improve the compound's attributes. In the last decade, several CADDs based on artificial intelligence technologies have been used to drive classical trials, but they are still costly and time-consuming. Because there is such a big amount of experiment data and biomedical data to compare. Deep learning techniques provide a more powerful and effective way to cope with the vast amounts of data generated by modern drug discovery methods.

Drug development and drug planning is a time-consuming process. It can be thought of as both pure scientific inquiry and industrial development. To acquire knowledge on existing molecules, the drug planner must visit vast public or private databases at various phases of the process. These questions can be of various types; for example, it is required to investigate how two molecules can bond together based on strictly chemical criteria while also taking into account biological limits imposed by the molecule's toxicity to the human body. From a data standpoint, these searches return a wide range of information, including the structure of the molecular graph as well as textual and numerical data.

Drug discovery is a lengthy and costly process that involves a number of processes such as drug target selection, target validation, virtual screening (VS), hit to lead generation, lead optimization, and so on. Furthermore, the development of a new drug costs more than 2 billion on average before taxes and takes roughly 10-15 years [93, 31] . Despite the significant time and financial investment, the projected success rate of clinical approval of new small compounds during the drug discovery process is around 13%, implying a relatively high probability of failure. Computational approaches are used to help with drug design at practically every stage. Yu and MacKerell [100] present a review of the drug discovery process and the computational drug design methods that go with it.
Computational approaches do not ensure a thorough examination of molecular attributes (e.g., bioactivity, ADMET qualities, selectivity, and physicochemical properties), but they do generate lead molecules with desirable properties in silico. Candidate clinical molecules chosen for drug discovery must have a profile that fits a number of criteria, including potency of effect, selectivity, safety, and ADMET characteristics. As a result, designing the best molecule is a multidisciplinary task encompassing several areas of chemistry and biology, which is addressed utilizing machine learning. The ability to access and mine huge data sets with diverse information is critical for the success of ML techniques in property prediction. Until recently, the most effective machine learning approaches were the so-called "shallow" ones, such as support vector machines (SVMs) and decision trees, particularly ensemble methods like Random Forest (RF). To improve model dependability and prediction power, all of these ML models should be iteratively modified using new experimental data.
The ability to detect compound-target interactions is a critical step in the drug development process. Predicting the interaction between chemicals and proteins, for example, can drastically lower the cost of developing new drugs. As a result, various in silico strategies for predicting compound-target interaction and facilitating novel drug development have been presented. VS, in particular, is a hot issue in chemoinformatics and medicinal chemistry, as well as a commonly used tool in pharmaceutical research. VS entails

searching huge databases of small compounds for bioactive chemicals that are relevant to the target of interest. By lowering the number of candidate molecules, the researcher can save money on the cost of testing thousands of chemicals in the lab. In the recent decade, as DL has matured as a subject, research in the field of VS has become increasingly relevant [37] . In this topic, there is a lot of scientific dispute over how to represent the chemical structures that the network needs to learn. Classical representations such as molecular fingerprints were used in the very early architectures [72] and SMILES notation [97].

Recent research has looked at molecular graphs in conjunction with neural embeddings, which are essentially a low-dimensional vector representation for discrete and/or categorical data that may be used to train a neural network in place of the original samples. In fact, neural embeddings are a technique of adapting the representation of input data to the numerical limitations imposed by a neural model's training operations. Because the protein kinase family, which I chose as my study's family target, is so diverse and contains so many proteins, it provides a plethora of data that is ideally suited to VS-oriented ML techniques for new kinase inhibitors. In Martin et al. [64], used a huge but poorly populated data matrix of over 100,000 chemicals to develop Bayesian models for Quantitative Structure-Activity Relationship (QSAR) models based on various kinases.Another case study used Random Forest to estimate kinase activity on hundreds of kinases using publically available datasets and in-house data. Random Forest models have shown stronger prediction reliability than other approaches in certain cases, however they perform worse than deep neural networks [71]. In most applications that try to detect patterns from training data and develop models to make predictions, Deep Neural Network (DNN), also known as Deep Learning (DL), has been proved to be superior than classic Machine intelligence techniques. Although DNNs have been employed in drug research to predict QSAR and ligand-based bioactivity, none of these models have profited from this powerful convolutional design.

The CNN (convolutional neural network) is a well-known deep learning model that uses a deep architecture to extract a set of spatial hierarchies of information at several levels of abstraction. Image categorization, object recognition, gene function prediction, pharmaceutical compound screening, drug discovery, and other applications have all used it.

A DNN is a learning algorithm that can automatically extract and learn relevant features from input data due to its "deepness" i.e., the very large number of layers containing atomic computational units called *neurons* as they mimic the computation of biological neurons. I chose to work with CNNs for several reasons, mainly because they automatically learn the relevant features and are therefore particularly suitable for complex classification tasks, such as mine, where it is not easy to define a priori which features the classification depends on. Secondly because convolution is a mathematical operation that handles well the type of data that I decided to use as input, the molecular fingerprints.

In fact, in recent years DL has been used in all fields of research related to life sciences: a review of DL techniques in computational biology is reported in [8], while a comprehensive report on DL for medical imaging is proposed in [9]. DL solutions have been proposed to support all phases of the drug design workflow [47], and in general, AI-based techniques such as Decision Support Systems and robotic platforms are expected to be in synergy with human medical chemist in the near future to perform drug discovery [85]. Virtual screening is undoubtedly one of the most investigated topics for DL applications. We refer in particular to [51] for structure-based approaches, and to [91] for ligand-based ones. The first DNN for QSAR prediction was a multi-task classifier presented in [26] where the same candidate was tested for its bioactivity on different assays.

Wallach in [96] presented *AtomNet*, which is considered the first CNN for structure-based screening. In [33] a CNN for learning circular fingerprints [72] from molecular graphs is proposed, and some experiments are performed to demonstrate their effectiveness in both solubility and drug efficacy prediction. In [78] *DeepVS* is presented: this CNN makes use of the notion of *context* of an atom in the protein-compound complex that is a vector representation of

the structural properties of its neighborhood. In [41] the SMILES notation [97] describing the compound is used to create a *feature* matrix where each column is a one-hot encoding of the presence of a particular SMILES symbol at a certain position. This representation is fed to a CNN to detect the "chemical motifs" that are relevant to the binding substructures. In Jimenez-Carretero et. al. [46] research in 2018, they used a deep convolutional neural network (CNN) to train the model to predict the toxicity of images of DAPI-stained cells pretreated with a group of drugs with different toxic mechanisms. Goh et. al. [36] developed "Chemception", a deep CNN for predicting chemistry, using only two-dimensional drawings of molecules. Although Chemception is slightly inadequate in terms of predicting toxicity, there is still room for improvement.

To improve model dependability and prediction power, all of these ML models should be iteratively modified using new experimental data. Deep learning techniques, particularly CNNs, have had a growing impact on drug and VS design in recent years as a result of the significant rise in prediction accuracy at any level of the process. When utilized on different targets, DNNs have been used to predict biological activity, ADMET characteristics, and physicochemical parameters, displaying reliable and robust prediction skills with high sensitivity [94, 34].

Several DNN architectures use Simplified Molecular Input Line Entry System (SMILES) as input data, described in detail in the 2.4.2, which are strings with a given length that are obtained from the molecular structure of each individual molecule by applying a few simple rules [18]. SMILES is actually a simple chemical language whose rules allow the construction of string descriptors that can represent both molecular structures and reactions.
The design of a deep architecture is done by combining both the intuition of the designer and numerous tests to identify the best architecture. For example, AtomNet was one of the first deep neural networks designed for drug discovery to predict the bioactivity of small molecules by applying convolutional [96] technology. This approach has been used to predict various properties such as the kinetic energy of hydrocarbons as a function of electron density

[99].

There are very few recent examples using deep learning and fingerprinting in a VS [81] workflow. An interesting approach is presented by Hirohara et al. who present a CNN that learns a suitable fingerprint from SMILES, and use that feature to classify both active and inactive compounds [41]. Regarding the proposed application, the literature reports very few recent approaches for Virtual Screening regarding Cyclin-Dependent Kinase proteins that do not use molecular fingerprints as a descriptor [78, 56]. Finally, DNN-VS is a very recent network for VS applied to tyrosine kinase using molecular descriptors [10]. In fact, molecular fingerprints are the most natural choice to describe compounds as inputs to a neural network because of their inherent structure of number vectors encoding all substructures within a molecule.

2.1 Chemoinformatics

Chemoinformatics is a well-established field that focuses on extracting, processing, and deriving useful information from chemical structures. Machine learning literacy has become a vital skill for medicine contrivers to mine chemical information from vast compound databases to create medicines with important biological features, thanks to the rapid expansion of chemical big data from HTS and combinatorial synthesis.

Machine learning is presently one of the most important and fleetly evolving motifs in computer-aided drug discovery [94]. Machine learning approaches use pattern recognition algorithms to discern fine connections between empirical compliances of small molecules and decide on them to predict chemical, natural, and physical properties of new composites, in contrast to physical models that calculate on unequivocal physical equations like amount chemistry or molecular dynamics simulations. Furthermore, as compared to physical models, machine learning methods are more effective and can easily be applied to large datasets without requiring a large amount of processing resources.
Helping experimenters identify and exploit links between chemi-

cal structures and their natural conditioning, or SAR (structure-activity relationship), is one of the key operation areas for machine learning in medicine discovery [7]. For case, given a hit compound from a drug screening campaign, we might wish to know how its chemical structure can be optimized to improve its binding affinity, natural responses or physiochemical properties. Fifty years ago, this type of problem could only be addressed through multitudinous expensive, time- consuming, labor-ferocious cycles of medicinal chemistry conflation and analysis.

Below I report the main techniques of machine and deep learning used with a discreet success in chemoinformatics.

Supervised Learning:

- Multiple regression analysis (A statistical procedure for determining links between dependent and independent variables),[63]
- k-nearest neighbor (A type of instance-based learning in which an item is classified using the majority rule among its k nearest neighbors, where k is an integer) [49]
- Naive bayes (A probabilistic technique that assumes feature independence and employs the probability prior and Bayes rule to forecast membership) [40]
- Random forest (Multiple decision trees and majority voting rules are used to create a categorization strategy)[42]
- Neural network and deep learning (Input layers, many hidden layers (for deep learning), and output layers make up a model-based learning system that learns from input data using layers of connected neurons), [13]
- Support vector machine (Using a nonlinear kernel, a statistical strategy for mapping data into high-dimensional space and identifying a lower-dimensional hyperplane that maximizes data separation. This is accomplished by maximizing the support vectors, which are the margins between hyperplanes),[90]

Unsupervised learning:

- k-means clustering (The minimization of within-group distances to the centroid is used to classify data into k groups)[61]
- Hierarchical clustering (A classification approach that uses agglomerative clustering, such as merging smaller clusters, or divisive clustering, such as breaking a large cluster into smaller ones, to create a hierachy of clusters) [61]
- Principal component analysis (Principal components are a statistical strategy for transforming a group of correlated traits into new independent variables using an orthogonal procedure) [6]
- Independent component analysis (A statistical strategy for separating statistically independent additive components from a multivariable output)[44]

In the chapter 4 I reported the results of preliminary experiments comparing my approach with that of classical machine learning techniques such as RF and SVM. Going to make a more in-depth study like the one I did in chapters 5 and 6 it is no more possible to make a real comparison between the models mentioned above and the approach that I am proposing in this thesis. First of all because the type of data that is used by combining the different molecular fingerprints is not usable with classical machine learning techniques, but I have reported, albeit not in a completely explicit way, comparisons with different deep learning techniques. For example using classic convolutional operators and dept separable ones.

Today, cutting-edge machine learning techniques may be used to model QSAR, or quantitative structure–property relationships (QSPR), and create artificial intelligence algorithms that can anticipate in silico how chemical alterations might affect biological activity. QSAR methods have been used to mimic a variety of physiochemical aspects of drugs, including toxicity, metabolism, drug-drug interactions, and carcinogenesis. Previously, QSAR models employed basic multivariate regression models to associate potency (logIC50) with substructure motifs and chemical parameters like solubility (logP), hydrophobicity, substituent pattern, and electronic variables, comparable to Hansch and Free – Wilson analysis.

These approaches, while revolutionary and successful, were eventually constrained by the lack of experimental data and the linearity assumption used in modeling. As a result, improved chemoinformatics and machine learning techniques capable of modeling nonlinear datasets, as well as big data methods capable of adding depth and complexity, are required. Despite its benefits, AI has substantial data difficulties, including the data's scale, growth, diversity, and ambiguity. Pharmaceutical companies' drug research data sets can contain millions of molecules, and typical machine learning algorithms may not be able to handle them. A computational model based on the quantitative structure-activity relationship (QSAR) can swiftly predict huge numbers of compounds or simple physicochemical characteristics like logP (Repartition coefficient) or logD (Distribution coefficient).

However, these models are a long way from predicting complicated biological features like chemical efficacy and side effects. Small training sets, experimental data error in training sets, and a lack of experimental validations are all issues that QSAR-based models confront. To address these issues, recently emerging AI technologies, such as deep learning (DL) and relevant modeling studies, can be used to evaluate the safety and efficacy of therapeutic compounds using big data modeling and analysis. For ADMET data sets of drug candidates, DL models exhibited significant predictivity as compared to traditional ML techniques. By depicting the distributions of molecules and their attributes, the virtual chemical space provides a geographical map of molecules. The goal of the chemical space illustration is to gather positional information about molecules inside the space in order to search for bioactive compounds, and therefore virtual screening (VS) aids in the selection of relevant molecules for further testing. ChemBL, PubChem, ChemBank, DrugBank, and ChemDB are just a few of the chemical spaces that are open to the public.
Drug development and discovery can take more than a decade and cost an average of 2.8 billion. Even still, nine out of ten medicinal molecules fail Phase II clinical studies and are not approved by regulators. VSs are developed using algorithms such as Nearest-

Neighbour classifiers, RFs, extreme learning machines, SVMs, and DNNs, which may predict in vivo activity and toxicity. Several biopharmaceutical companies, including Bayer, Roche, and Pfizer, have teamed up with IT firms to create a platform for medication discovery in areas including immuno-oncology and cardiovascular disease.
Physicochemical features of a drug, such as solubility, logP, degree of ionization, and intrinsic permeability, have an indirect impact on its pharmacokinetic qualities and target receptor family, and must be taken into account while developing a new medicine. Physicochemical properties can be predicted using a variety of artificial intelligence-based technologies. To train the software, ML, for example, employs massive datasets generated during compound optimization. Molecular descriptors, such as SMILES strings, measurements of potential energy, electron density around the molecule, and 3D coordinates of atoms, are used in drug design algorithms to construct viable molecules using DNN and then forecast their attributes.

In the subject of chemoinformatics, machine learning techniques have been widely used to find and build novel medications with improved biological activity. The development of a constellation of 2D or 3D chemical descriptors, which are packed as chemical fingerprints in a variety of machine learning models and predictive tasks, is made possible by mathematical mining of chemical graphs. The combination of big data and machine learning to predict a greater range of biological events is a prominent area of advancement in the discipline. For clinical medication safety, traditional drug design strategies based on simple ligand–protein interactions are no longer sufficient. Biological mechanisms and systematic reactions at higher levels are frequently involved in high drug attrition rates due to severe side effects.

As a result, AI plays a key role in drug development, predicting not just physicochemical qualities but also desirable bioactivity. The affinity of drug molecules for the target protein or receptor determines their efficacy. Drug molecules with no contact or affinity for the target protein will be unable to deliver a therapeutic re-

sponse. It's also possible that the medication compounds generated will interact with undesired proteins or receptors, causing harm. As a result, predicting drug-target interactions requires knowledge of drug target binding affinity (DTBA). By taking into account the qualities or similarities of the drug and its target, AI-based algorithms can calculate the medication's binding affinity. To determine carrier vectors, feature-based interactions recognize the chemistries of the medication and the target. In similarity-based interactions, on the other hand, the resemblance of the drug and the target is taken into account, and it is anticipated that comparable medications would interact with the same targets.

The widespread use of virtual screening for drug development has been facilitated by technological advancements over the last two decades. Virtual screening is an in silico method for searching vast databases of small compounds for bioactive chemicals. By decreasing the number of potential molecules to be tested to reasonable levels, the researcher can avoid the cost of experimentally testing hundreds or thousands of chemicals[21]. AI applications can be found in all aspects of drug development, owing to its versatility to a wide range of activities.

I focused my attention on virtual screening and on drug repurposing process. The difference between the two processes is very subtle. Both are screening of compounds, in the first case the task is to identify new compounds that can have high probability interactions with the chosen target while in the second case the screening occur on a already approved drugs for other targets database. In this second case I focused attention on protein targets responsible for the infection of SARS-COV-2. The choice of dealing with drug repurposing was dictated by the emergency situation of the world because of COVID-19 disease .

2.2 Virtual screening

Drug discovery is the very long and complex process leading to the development of a new medication, where several steps and loops are involved. Indeed, one of the most relevant parts in the drug dis-

covery cycle is *drug design* when one already knows the biological target the new compound has to bind to. In general, a biological target is an enzyme or a protein. In a modern drug design setup, many compounds are screened to assess the best matching ones as regards their ability of either inhibiting or activating the target associated to a particular disease. Such a process is also referred to as *Inverse Pharmacology* or *Target-based Drug Design.*

To collect knowledge on existing compounds, the drug designer must visit big public or private databases as part of his or her work. Such questions can take many forms; for example, it may be required to look into how two molecules can bond to each other based on strictly chemical criteria while also taking into account biological limits imposed by the molecule's toxicity to humans. In terms of data, these queries yield a wide variety of results, ranging from a full molecular graph representation to textual and numerical data.

VS is a computational technique used in drug discovery to search libraries of small molecules in order to identify those structures which are most likely to bind to a drug target, typically a protein receptor or enzyme.

VS stands for a computer-assisted process for identifying compounds that are likely to be active on a certain target. These strategies range from similarity searches to ML methods, and they all rely on numerical descriptors of the proposed compound's structure and its chemical properties.

In the context of machine learning, virtual screening may be thought of as a classification task; popular algorithms include Support Vector Machines and Random Forests. Convolutional and Recurrent Neural Networks are utilized for VS and other domains of pharmacology, including as chemical reaction prediction, in the Deep Learning era. The reader is referred to [47] for a recent and thorough review.

VS is a widely used computational approach for drug develop-

ment. However, due to the intricacy of the algorithms utilized in the screening effort, several concerns remain unknown, resulting in models with varying prediction reliability. Clinical candidate molecules chosen by drug detection must have a profile that responds to a variety of parameters, including effect potency, selectivity, safety, and so called ADMET characteristics. As a result, designing the best molecule is a multifaceted task combining several areas of chemistry and biology, which may be tackled with machine learning.
Computational strategies for virtual screening fall into two categories: ligand-based drug design (e.g. ligand similarity) and structure – based drug design (ligand docking). The ligand similarity methods take advantage of the fact that ligands that are similar to an active ligand are more likely to be active than random ligands, whereas protein-ligand docking attempts to predict the binding modes and affinities of ligands to the target protein using the three-dimensional (3D) protein structure of the target protein.

For both lead discovery and optimization, ligand-based virtual screening methods rely on the information contained in known active ligands rather than the structure of a target protein. When no 3D structure of the target protein is available, ligand-based approaches are the sole option. In practice, even if one doesn't know the protein structure of the target of interest, usually it is possible to assess if a group of ligands is active against it. As a result, ligand-based virtual techniques, such as identifying new ligands by comparing candidate ligands to known active molecules, can be used.

The extensive range of QSAR approaches is an excellent example of ligand-based drug design. Structure-based drug-design methods are regularly used in the drug development process when there is sufficient structural knowledge about the target protein, especially if a crystal structure is available.

Structure-based design is concerned with modelling the interactions between a ligand and a protein. These interactions are studied using molecular dynamics or Monte Carlo simulation-based free energy approaches, as well as protein–ligand docking. There are a

rising number of systems where both ligand and protein structural data are available, as the number of known protein–ligand crystal structures grows and more physicochemical and biological data on ligands is released. As a result, the hybrid technique, which combines ligand-based and structure-based drug creation on the same protein system, is becoming increasingly popular. These attempts might be as easy as running QSAR or pharmacophore studies on the same system and docking them, and there are examples of this in the literature[73, 58].

An effective similarity measure and a trustworthy scoring system are two crucial components of a ligand-based computational technique. Furthermore, the computational technique should be capable of accurately and quickly screening a large number of candidate ligands. As a result, the similarity measurements are made up of geometrical data from arbitrary objects defined on the structures. The classification of an object varies depending on the approach used, however it may be divided into three categories: pharmacophores, molecular shapes, and molecular fields. Pharmacophore-based approaches establish patterns of distances between predetermined molecular features such as aromatic systems or hydrogen-bond acceptors/donors, and then compare the patterns to calculate the similarity value. The goal of molecular shape techniques is to maximize shape overlap and calculate a similarity value based on that overlap. A scoring method for ligand-based screening should effectively discriminate active compounds from the inactive ones during the ranking phase and can be used to efficiently identify a small number of active compounds from a library including a large number of inactive compounds [39].

2.3 Drug repurposing

Drug repurposing (also known as drug repositioning, reprofiling, or retasking) is a strategy for discovering new uses for authorized or investigational medications that go beyond the original medical indication. This approach has several advantages over designing a

whole new medication for a specific indication. First and foremost, the risk of failure is lower; because the repurposed medicine has previously been found to be sufficiently safe in preclinical models and people if early-stage trials have been completed, it is less likely to fail in later efficacy trials, at least from a safety standpoint.
Second, because most preclinical testing, safety assessments, and, in some cases, formulation development will have already been accomplished, the time frame for drug development can be decreased.
Third, less investment is required, however this will vary substantially depending on the repurposing candidate's stage and development process.
For example, retrospective clinical experience was used to repurpose sildenafil citrate for erectile dysfunction, whereas serendipity was used to repurpose thalidomide for erythema nodosum leprosum (ENL) and multiple myeloma. Sildenafil was originally developed as an antihypertensive medicine, but when Pfizer repurposed it for the treatment of erectile dysfunction and sold it as Viagra, it had a market-leading 47 percent share of the erectile dysfunction drug market in 2012, with global sales totaling 2.05 billion [79].
Finding new therapeutic indications for medications that have already been approved is required when repurposing them. This is a growing technique for discovering new drugs, as it capitalizes on existing investments while reducing the risk of clinical trials. This method is appealing since we continue to encounter major gaps in the drug–target interaction matrix, as well as the need to collect safety and efficacy data throughout clinical trials. Collecting and making publicly available as much data as possible on medication target profiles opens the door to drug repurposing, although patent filings may limit commercial applicability. Because of significant disparities in side effect tolerance, certain clinical uses may be more feasible for repurposing than others. Relevance to the condition in question, as well as the intellectual property landscape, should be addressed when evaluating drug repurposing potential. These activities extend far beyond the discovery of new targets for existing medicines.

DR is an approach for discovering new applications for autho-

rized or late-stage medications that aren't covered by the original indication. Compared to designing an altogether novel medicine for a given indication, this technique has significant advantages

These advantages, when combined, have the potential to result in a less risky and faster return on investment in repurposed drug development, with lower average associated costs once failures are taken into account (in fact, the cost of bringing a repurposed drug to market has been estimated at $300 million on average, compared to an estimate of $2-3 billion for a new chemical entity [74]). Finally, repurposed drugs may uncover new targets and pathways that can be explored further [80].
Drug repurposing typically consists of 3 steps, before being able to identify a drug that can be reused on another disease. The first and most critical step is to identify a candidate molecule. The second step is to evaluate the drug's efficacy in preclinical models (in vivo or in vitro tests) which, if successful, lead to step 3, i.e. evaluation in phase II clinical trials. Since the first step is the most important, but also the most expensive, new approaches have been developed over the years to speed up this phase, and in particular computational methods, thanks to the possibility of analysing data of different nature (e.g. gene expression data, chemical structure, genotypic or proteomic data or EHR) have proved to be the strong point in the approach to this task.

One of the most popular computational approaches to date is *signature matching*, which is based on the comparison of a unique characteristic between drugs or the comparison of a drug with a diseased phenotype. The signature can be derived from three different types of data: transcriptomic, proteomic and metabolomic. Transcriptomic data can be used to perform a drug-disease or drug-drug comparison, in both cases to identify similarity. For two drugs, sharing a transcriptomic signature may imply a shared therapeutic application regardless of their structural similarity or sharing of similar chemical structures. Because of the effectiveness of this approach, the cMap (Connectivity Map) was created in 2006, which contains expression profiles generated by dosing more than 1,300

compounds in a range of cell lines [54].*Molecular docking* is a further computational strategy used for drug repurposing to predict the complementary binding site between ligands and targets. In contrast to the traditional approach described in the previous chapter, inverse docking is used in DR, where multiple receptor sites are interrogated against a specific drug in order to identify new interactions.

A further approach is based on *GWAS*, which is based on the search for variations associated with common diseases; this has been made possible by the technological leap forward that has been achieved with new genotyping techniques and the completion of the Human Genome Project [95, 55], the worldwide project for the complete sequencing and mapping of the human genome. The results obtained from this research are not always easy to interpret, especially in DD, which is why they are often associated with pathway analysis or network mapping, which provides information on the proteins involved in the signal cascade, also clarifying the result obtained through GWAS.
A further method is the Retrospective clinical analysis which is based on a systemic analysis of data that can be obtained from various sources, including EHR data and clinical trial data. EHRs contain a considerable amount of structured data such as diagnostic and pathophysiological data, including data obtained from laboratory results and unstructured data such as clinical descriptions of patient signs and symptoms or data imaging. Data belonging to this category are not always open access, often bound by ethical constraints and legal restrictions. In 2016, the EMA [17] started to give free access to data obtained from clinical trials submitted by pharmaceutical companies for the free use of the academic community.

In addition to computational methods, experimental approaches are also used in drug repurposing, the two most widely used being: i) Binding assays to identify target interaction using proteomic techniques such as chromatographic affinity and mass spectrometry. The CETSA technique, for example, has been introduced as

a method of mapping target engagement in cells using biophysical principles involving the thermal stabilisation of target proteins by drug-like ligands with the appropriate cellular affinity; ii) phenotypic screening can identify compounds that show disease-relevant effects in model systems without prior knowledge of the target(s) involved. In the context of drug repurposing, if the compounds screened are approved or in the process of being approved, this may indicate repurposing opportunities that can be readily seized.

A summary of all approaches to Drug Repurposing is shown in the figure 2.2. Various computational approaches can be used individually or in combination to systematically analyse different types of large-scale data to obtain meaningful interpretations for repurposing hypotheses.

2.4 Molecular representations

In this section I report the main molecular representations learned during the PhD period, focusing on those that contributed primarily to obtaining multi-fingerprint embeddings.

2.4.1 Molecular graph

The graph is the first kind of representation one identifies with molecules, and it is also the starting point for the building of the various chemical descriptors outlined below in computational chemistry. The molecular graph representation is based on mapping the atoms and bonds that make up a molecule into a set of nodes and arcs, often in a 2D structure that can be extended with 3D data (e.g. atomic coordinates, bond angles and chirality).
A graph is a data structure comprised of two parts: nodes (vertices) and edges (connections). A graph G can be defined as $G = (V, E)$ where **V** is the collection of nodes, and **E** are the edges between them. Edges are directed if there are directional dependencies between nodes. Otherwise, the edges are undirected. A graph is often represented by **A**, an adjacency matrix shown in figure 2.3. If a graph has **n** nodes, **A** has a dimension of $(n * n)$. Sometimes the nodes have a set of features. If the node has **f** numbers of features,

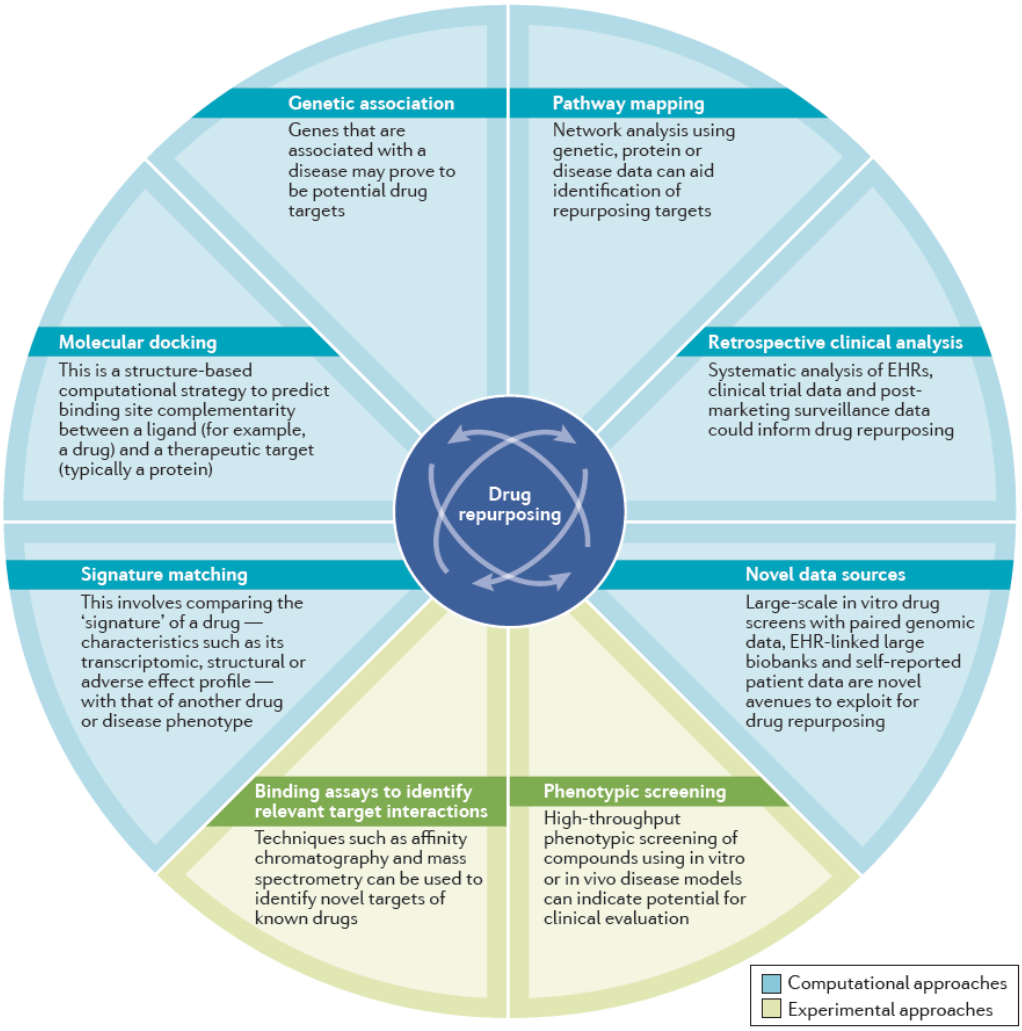
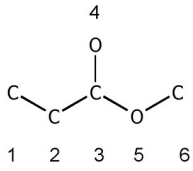


Figure 2.2: Approaches used in drug repurposing. Image taken from [80]

.

then the node feature matrix $\mathbf{X}$ has a dimension of $(n * f)$.

In a molecular graph, $\mathbf{V}$ is intuitively the set of all atoms in a molecule, and $\mathbf{E}$ is the set of all bonds linking the atoms, although this does not have to be the case. Molecular graphs are generally undirected, meaning that the pairs in $\mathbf{E}$ are unordered. To convert a graph from an abstract mathematical idea to a tangible representation that can be handled by a computer, the sets of nodes and edges must be converted to linear data structures; one popular method is to use matrices or arrays. To specify the connectedness of the nodes, linear data structures are required. Even while $\mathbf{V}$ and $\mathbf{E}$ are officially sets and the order of elements in sets is immaterial, a artificial node-ordering must first be created for encoding a molecule



G_1

$A(G_1) =$

	1	2	3	4	5	6
1	0	1	0	0	0	0
2	1	0	1	0	0	0
3	0	1	0	1	1	0
4	0	0	1	0	0	0
5	0	0	1	0	0	1
6	0	0	0	0	1	0

$D(G_1) =$

	1	2	3	4	5	6
1	0	1	2	3	3	4
2	1	0	1	2	2	3
3	2	1	0	1	1	2
4	3	2	1	0	2	3
5	3	2	1	2	0	1
6	4	3	2	3	1	0

Figure 2.3: The adjacency matrix (A) and distance matrix (D) for the hydrogen-suppressed graph (G1) of ethyl acetate [12]

using arrays. How the atoms in the molecule are connected, the identification of the atoms, and the identity of the bonds are all examples of information that can be mapped.

The two formats utilized in chemoinformatics field, closely related to the molecular graph representation are connection tables and the MDL (now BIOVIA) file format.

In figure 2.4 The MDL family of file formats are collectively known as CTfiles (chemical table files) as they are built upon connection tables (Ctab), shown at the top of the figure. The connection table is split into an atom and bond block, describing the atoms and their corresponding connectivity. The Ctab is built upon to form the Molfile for the description of single molecules, RGfile for handling queries, SDfile for structure and associated data, RXNfile for the description of single reactions, RDfile for either a series of molecules/reactions and their associated data, and the XDfile for the transfer of structure or reaction data based on the XML format[28]. A molecular graph conveys structural information through both categorical and numeric data at each node or edge: atoms and bonds types, presence of rings, aromaticity, formal and partial charge, and so on. As a consequence, the use of embeddings is a common practice when designing a GNN for molecular analysis. One of the most widely used GNNs belongs to

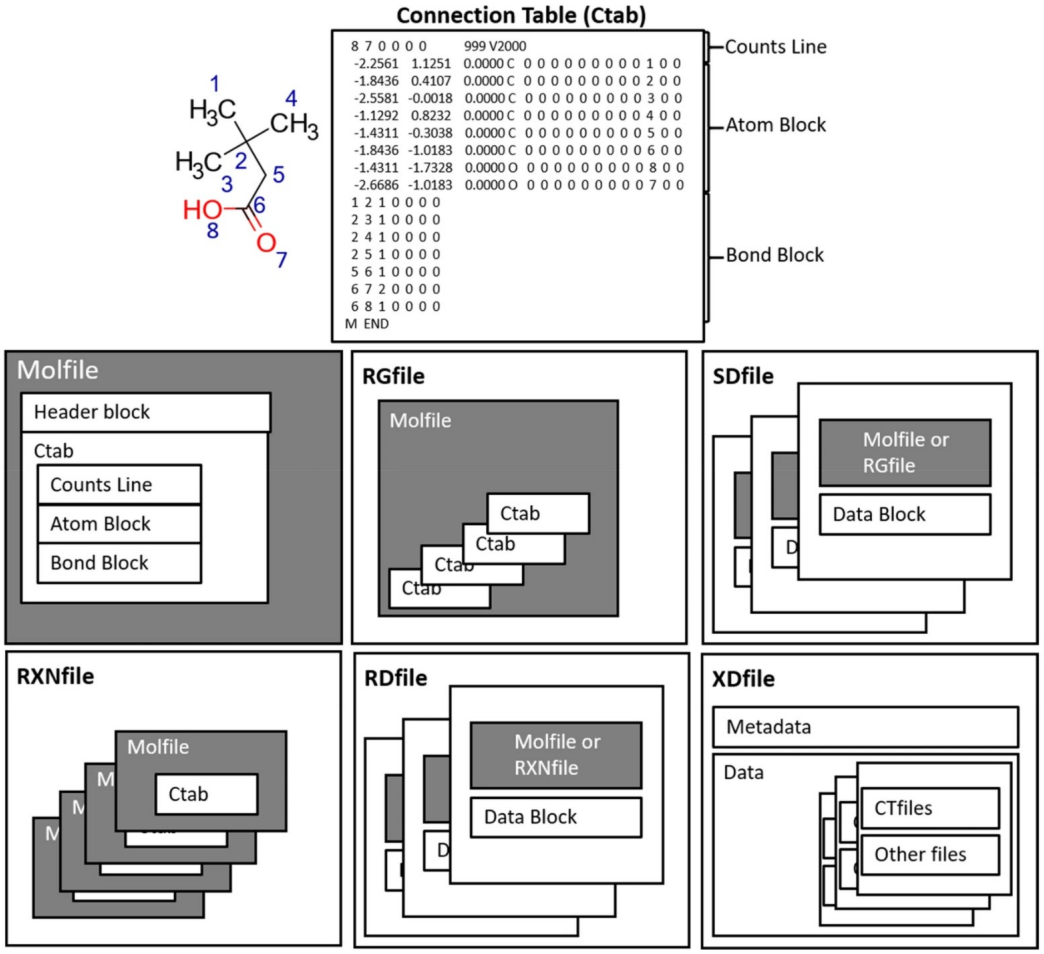


Figure 2.4: Example of Structure/Data file, containing both structural information and additional property data for any number of molecules.

the family of so-called Message Passing Neural Networks (MPNNs) that were introduced in depth in [35]. MPNNs perform learning at each state through a "message function" that passes information between nodes along the edges, followed by an "update function" that computes the new state at each node.

Another application of molecular graph are reported in Torng et al. work [92], where drug-target interaction is investigated through graph autoencoders. A neural embedding for target pocket features is learned through a graph variational autoencoder (VAE) that is a DNN trained to learn a latent representation of the inputs in an unsupervised way: two mirrored CNNs are coupled, and the overall network is trained with its inputs. The activations of the innermost layer form a low dimensional *latent representation* of the input space. The weights of the trained encoder in the Graph-VAE

are used to perform fine tuning in a target Graph-CNN that is trained in parallel to a ligand Graph-CNN. The two Graph-CNN are fed in parallel to a fully connected "interaction" layer, and then to the output binding classifier. In koge et al. work [52] a molecular embedding is proposed where hypergraph molecular representations are learned by VAEs based on RNNs along with a regression model for physical molecular properties so that anchor, positive and negative molecular samples w.r.t. a particular property have a latent representation that maintains similarity. Finally, in Ishiguro et al. work [45] the Weisfeiler-Lehman (WL) embedding of the molecular graph is proposed as the input for a MPNN. The WL embedding is a simple algorithm that enumerates the neighbors of each atom so that the input of the MPNN is formed by the atom label and the vector of its neighbors' labels.

2.4.2 SMILES

SMILES (The **S**implified **M**olecular **I**nput **L**ine **E**ntry **S**ystem), was developed in 1988 by Weininger et al. [97] and since then it is the most popular line notation because it can represent molecules in a simple way.

The SMILES notation system was later incorporated into the Daylight Chemical Information Systems toolkit [2]; the company is still maintaining it. The SMILES representation, which is non-unique and unambiguous, is obtained by assigning a number to each atom in the molecule and then traversing the molecular graph using that order as shown in Figure 2.5. The multiple representations of a molecule created by randomly selecting the starting node in the graph traversal algorithm, thus varying the order of the nodes traveled in the molecular graph, are referred to as randomized SMILES (still using depth-first search). The numbers represent the traversal order of the graph, with 1 being the first node (user defined). If we consider a to be the conventional representation of aspirin, b depicts a distinct arrangement of the molecule's atoms. The final SMILES is one of the randomized SMILES that can be created. The molecular graph is navigated using green arrows. Both SMILES strings depict the same molecule, but the generated SMILES strings differ due to the atom numberings.

SMILES did not originally encode for stereochemistry. Later on, a specification known as isomeric SMILES was established, and it is now the default SMILES in many software applications. SMILES may therefore encode isomeric specifications, configurations around double bonds (Z or E), configurations around tetrahedral centres, and a variety of additional chiral centres that are rarely supported (e.g. allene-like, octahedral). However, structures that cannot be easily represented using molecular graphs, such as organometallic compounds and ionic salts, are difficult to characterize using SMILES notation.

There are many varieties of molecules that the graph model cannot describe. Any structure with delocalized bonds, such as coordination compounds, as well as any molecule with polycentric bonds, ionic bonds, or metal–metal bonds, falls under this category. Organometallic compounds like metallocenes and metal carbonyl complexes, for example, are challenging to characterize using molecular graphs because their bonding scheme is not explained by valence bond theory. To put it another way, it would be difficult to define the bonds using solely atom-to-atom pairwise interactions.

Solutions to the handling of multi-valent bonds have been introduced via the use of hypergraphs; in a hypergraph, edges are sets of at least two atoms (hyperedges) instead of tuples of atoms [30]. However, the use of hypergraphs is not further discussed here as they are not currently widespread in the field.

The graph representation may not be useful for molecules whose atom arrangement is continually changing in 3D space, especially if pairwise bonds are breaking and forming or if the structure is frequently rearranging. That instance, for applications where a single static representation for a molecule that is actually rearranging on the timescale of the problem (e.g. tautomers), a single molecular graph representation would not be acceptable and could potentially be harmful to addressing the problem.

In deep learning models, many architectures use SMILES as the molecular representation [18], which as obtained by assigning a unique number to each atom in the molecule and then traversing the molecular graph using that order. Commonly, a canonical SMILES of each molecule, which is obtained by computing a unique

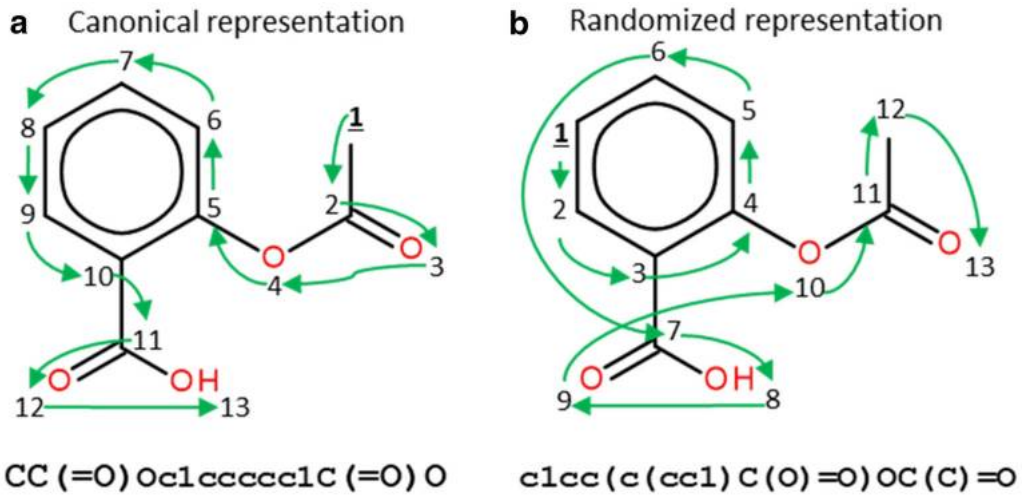


Figure 2.5: Canonical (a) and randomized (b) SMILES representations of aspirin. The original figure can be found in [27]

numbering for the molecules [97]. This representation has served as a way to uniquely identify molecules. However, most molecules can have more than one SMILES representation obtained by changing only the numbering of the atoms, which means that different SMILES start at different atoms of the molecule and traverse it in different ways (Figure2.5).

Randomized SMILES for the same compound can then be used for data augmentation. A major surge of interest in cheminformatics applications of deep learning has occurred in recent years when NNs have been used to generate molecules represented by SMILES strings ([75]; [38]; [88]). Recurrent NN (RNN) trained with a set of SMILES strings can generate molecules that are not present in the training set but have properties similar to the training samples.

Another deep learning model for chemical classification was created by Hirohara et al. They developed a distributed representation of compounds based on the SMILES notation, which depicts a compound structure linearly, and used the SMILES-based representation to a convolutional neural network in this manner. SMILES enables them to handle a wide range of compounds while combining a wide range of structure information, and CNN representation learning develops a low-dimensional representation of input features automatically.

Their solution beat conventional fingerprint methods in a benchmark experiment using the TOX 21 dataset, and was comparable to the TOX 21 Challenge winning model. Multivariate study indicated that the chemical space created by SMILES-based representation learning appropriately reflected a richer feature space that allowed for accurate compound classification. Not only key known structures (motifs) such as protein-binding sites, but also structures of unknown functional groups were found using motif detection with learnt filters.[41].

2.4.3 Molecular Fingerprint

The length and complexity of molecular fingerprints range from simple representations of restricted topological distances or functional group occurrences to complex multi-point 3D pharmacophore configurations. Surprisingly, in virtual screening studies, more complexity rarely resulted in better performance of fingerprints. Despite the fact that three-dimensional fingerprints have been used to assess compound similarity, two-dimensional (2D) similarities are still the most common method of recording small molecule properties in fingerprint bit strings. The speed and ease with which 2D fingerprints can be calculated is one factor. Figure 2.6 shows how chemical characteristics are transformed into bit strings in the case of Daylight fingerprints.

The most often used molecular fingerprints for similarity searches can be classified into the following categories:

- Topological fingerprints (e.g., Daylight [2], atom pairs [20]);
- Structural keys (e.g., MACCS [48], BCI [11], PubChem [4]);
- Circular fingerprints (e.g., Molprint2D [14], ECFP, ECFP [83]);
- Pharmacophore fingerprints (e.g., CAT descriptors [86], 3pt [66], [67] and 4pt [65] 3D fingerprints)
- Hybrid fingerprints (e.g., Unity 2D [5])
- Other fingerprints sometimes focusing for coding protein-ligand interactions (SMIfp [87],SIFFt [29], PLIF (MOE [3])

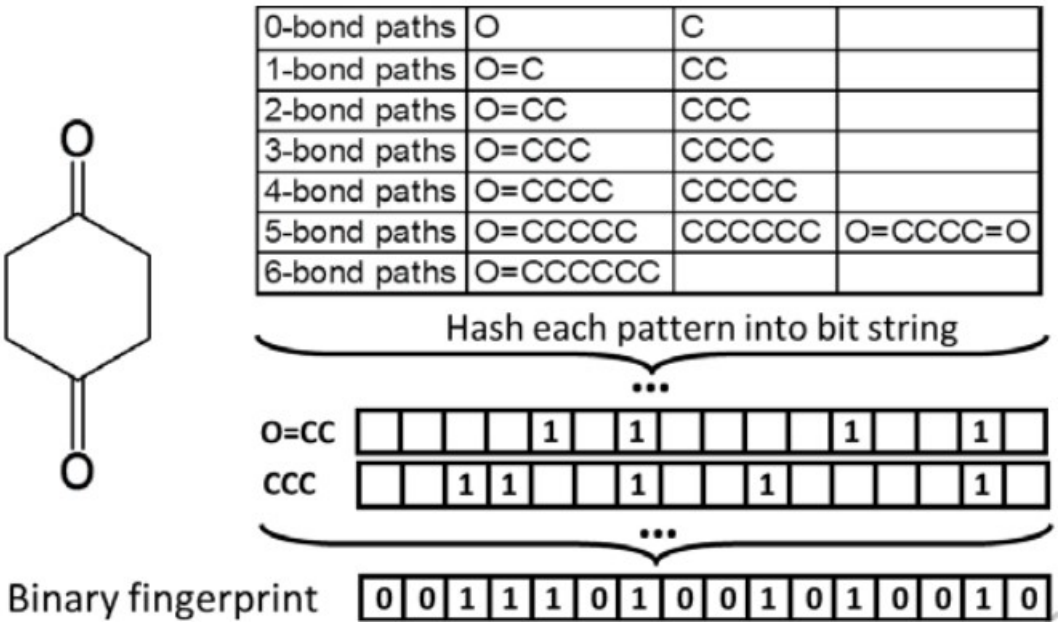


Figure 2.6: Generation of topological fingerprint using Daylight fingerprint as example [72].

Molecular fingerprints are a way of representing molecules as mathematical objects in fact they are representations of chemical structures originally designed to assist in the search for substructures of chemical databases but later used for analysis tasks, such as similarity searching[21], clustering, and classification [70].

A major principle of medicinal chemistry is that structurally equivalent molecules have similar biological activities. Molecular fingerprints are bit string comparisons that encode characteristics of small molecules and compute their similarity. Virtual screening is a procedure that uses molecular fingerprint approaches to identify additional compounds with a better chance of displaying similar biological activities against the same target based on their similarity to a biologically active template.

Similarity-search methods based on molecular fingerprints are an important tool for ligand-based virtual screening. There are a variety of fingerprints, and their effectiveness is mostly dependent by the validation data sets and similarity metric used. [82]. The majority of techniques generate fingerprints using only the 2D molecular graph and are referred to as 2D fingerprints, although some generate pharmacophoric fingerprints that include 3D information. The

three main strategies are substructure keys-based fingerprinting, topological or path-based fingerprints, and circular fingerprints.

Molecular fingerprints are created by examining each atom and its surroundings up to 6 or 7 bonds away. A series of predetermined molecular substructures, known as patterns, are looked for in such a neighborhood, such as atom types, bond types, the presence of rings, and so on. After enumerating all of the patterns in the molecule, each one is used as a seed for a hashing function that yields 4 to 5 index places in the "pattern fingerprint," with the associated bits set to 1; this fingerprint is bit-wise OR-ed to the molecular one. Actually, because the hashing function can induce a bit collision, we can't be sure that a pattern is present unless at least one of its bits is unique. A chemical substructure, on the other hand, is absent if all of its bits in the fingerprint are set to 0.

In this section were described in detail how are generated the different fingerprint types used in this study emphasizing the type of substructure analyzed.

Topological or pathway-based fingerprints work by analyzing all the fragments of the molecule that follow a pathway (usually linear) to a certain number of bonds, and then hashing each of these paths to create the fingerprint figure. 2.7. This means that any molecule can provide a relevant fingerprint, and the length of the imprint can be varied. These are hashed fingerprints, meaning that no one bit can be linked to a specific feature. Bit collision occurs when a single bit is set by many separate features. The most essential of these forms of fingerprints is the daylight fingerprint. It has a maximum length of 2048 bits and encodes all conceivable connection channels via a molecule up to that length while reducing bit collisions. A representation of a hypothetical 17-bit topological fingerprint is shown in the figure 2.7. The corresponding bit in the fingerprint is displayed for all fragments found from the beginning atom (circled in red). Only fragments and bits for a single starting atom are displayed; this process would be repeated for each atom in the molecule to obtain the whole fingerprint. Circular fingerprints take a similar strategy, but instead of linear fragments, they construct fragments within a radius of the initial atom.

The idea under fingerprint generation is to apply a kernel to a

molecule for generating a bit vector. Typical kernels extract features from the molecule, hash them, and use the hash to determine bits that should be set. Typical fingerprint size range is from 1K to 4K bits: in the cited work I used the 1024 bit size. As regards the fingerprint types I selected, they can be grouped in two classes: *pathway-based* also known as *topological*, and *circular*. Pathway-based fingerprints encompass RDKit, Atompair, Torsion and Layered. In this case the kernel is linear, and each fingerprint differs in atom types and bond types. For example, RDKit's atom types are set through atomic number and aromaticity. In Layered, both atom and bond types contribution are determined by the particular layers included in the fingerprint.

The process of creating a fingerprint is summarized in the following algorithm. The course and length of a fingerprint vary depending on the type of fingerprint. As you can see, the hashing function prohibits you from getting the same fingerprint for the same molecule, and an inverse research, that is, determining the presence of a certain path in the molecule from a single bit, is impossible.

Algorithm 1

FunctionMakeFingerprint(GraphMolecule, SizeD, IntLength)

```
1: fingerprint = initializeFingerprint(d)
2: paths = getPaths(molecule, length)
3: for each atom in molecule do
4:     for each path from atom do
5:         seed = hash(path)
6:         indices = random(seed)
7:         for each value in indices  do
8:             index = value mod d
9:             fingerprint[index] = TRUE
10:        end for
11:    end for
12: end for
13: return fingerprint
```

Four fingerprints used in this study belong to topological family:

- RDKit;
- Topological Torsion;
- Atom Pair;

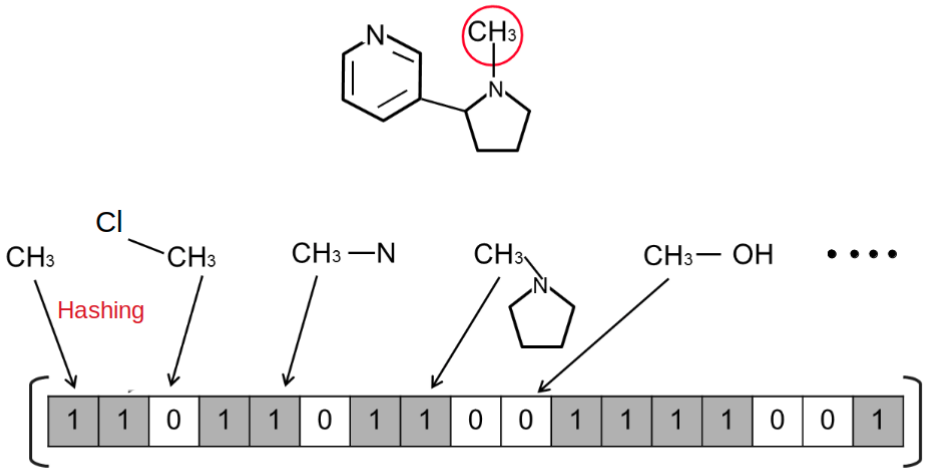


Figure 2.7: Fingerprint generation. Simplified fingerprint generation: the hashing function sets just 1 bit per pattern.

- Layered.

All the fingerprint that I utilize in this work are generated with the software *Knime*, more detail are reported in chapter 4.1.

Three types of circular fingerprints were used in this study:

- Morgans;
- ECFPs;
- FCFPs.

Morgan, Featmorgan, and ECFP4 are all circular fingerprints. In this scenario, the kernel is circular and takes into account each atom's neighborhood based on the radius chosen (usually from 1 to 3). Although it has been claimed that Morgan and EFCP fingerprints are identical, this is not totally true due to a considerable discrepancy in the aromatic groups. The ECFP algorithm, in particular, varies from the Morgan algorithm in two ways. First, rather than stopping when identifier uniqueness is achieved, the ECFP generation terminates after a predetermined number of iterations. The extended-connectivity fingerprint is determined by grouping the original atom IDs, as well as all subsequent identifiers, into a collection. The ECFP algorithm keeps the intermediate atom identifiers rather than discarding them. This means that the iterative

process does not have to go all the way to the end (that is maximum disambiguation) rather it is carried out for a fixed number of iterations. Second, algorithmic optimizations are available in ECFP because perfect accuracy is not necessary for disambiguation [83] .

The Extended-Connectivity Fingerprints (ECFPs), which are based on the Morgan technique and were specifically created for use in structural activity modeling, are the de facto standard circular fingerprints. They depict circular atom neighborhoods and produce variable-length fingerprints. They're most typically referred to as ECFP4 when they have a diameter of four.

Although some benchmarks have demonstrated slight performance differences between the two, a diameter of 6 (ECFP6) is also routinely utilized. In addition, there is a variant that keeps track of the ECFP feature frequency counts by recording each identifier as many times as it appears in the molecule rather than just once. ECFC is a common abbreviation for this variant. FCFPs (Functional-Class Fingerprints) are a version of ECFPs that are more abstracted in that they index the role of an atom in the environment rather than the atom itself. As a result, the fingerprint cannot discriminate between atoms or groups that have the same or comparable functions. They can now be employed as pharmacophoric fingerprints as a result of this. FeatMorgan is an FCFP that was employed in this research.

When compared to the others, the FeatMorgan fingerprint with RDikit and Layered had a significant impact on the prediction findings, as will be discussed in the results section. In FCFP, the ligand is defined by the functional descriptions of atoms that are directly related to its binding capabilities (for example, hydrogen donor/acceptor, polarity, aromaticity, and so on). When compared to the simple ECFP circular fingerprint, solely relative to atom type routes, such a ligand description is likely to outperform when used for such a classification, not just based on the chemical path, but also on the ligand capability to bind certain protein residues.

The primary problem in employing molecular fingerprints is the

embedding generation algorithm. It's impossible to search backwards for substituents and structures involved in the protein or enzyme binding process because of the bit collision and hashing functions. As a result, this type of descriptor can only be used for searching for similarities, not for studying binding affinity in general. These representations are used to evaluate chemical space coverage, molecular diversity, and similarity searching.

Instead than using traditional molecular descriptors, modern chemoinformatics approaches have emphasized the use of machine learning techniques applied to fingerprints. The reason for this is because fingerprints contain information on chemical groups and pathways, as well as complete information about molecular complexity, allowing for a more rigorous comparison between two or more structures than molecular descriptors. SMILES descriptors also transmit information about molecular structures, but their intrinsic string form necessitates cutting the cycles, and as no two molecules have the same description, a "SMILES canonicalization" is also required.

Similarity is a subjective concept that can be tested and the results interpreted in a variety of ways. The difficulty of the task, which is dependent on the complexity of the molecular representation utilized, is one of the most significant issues encountered while attempting to quantify the similarity between two compounds. Some level of simplification or abstraction is required to make computational comparisons between molecular representations easier. Molecular fingerprints are the most widely utilized of these abstractions, which involve converting a molecule into a sequence of bits that can then be easily compared between molecules. [22].

After that, the comparison must be expressed in a quantifiable manner. There are numerous methods for determining the similarity of two vectors. The Tanimoto coefficient, which is the same as the Jaccard coefficient for molecular fingerprinting, is calculated by dividing the number of common bits set to 1 in both fingerprints by the total number of bits set to 1 in both fingerprints. This means that the Tanimoto/Jaccard coefficient will always have a value between 1 and 0 as reported in Table 2.1 and that the similarity

between two fingerprints with a given Tanimoto coefficient will actually depend a lot on the type of fingerprint used, which makes it impossible to select a universal criterion to determine if two fingerprints are similar or dissimilar. However, the performance of molecular fingerprints could be improved by combining them with other similarity coefficients. Several similarity metrics that have been used with fingerprints are listed in Table 2.1. The presence or absence of features, then the binary association coefficients or similarity measures are based on the four terms where, given the fingerprints of two compounds, A and B, a equals the amount of bit set to 1 in A, b equals the amount of bits set to 1 in B and c equals the amount of bits set to 1 in both A and B and d equals the amount of bits set to 0 in both object A and object B.

VS procedures take advantage of the fact that fingerprints are not very sparse bit vectors from a computational standpoint. ML techniques employ well-known similarity measures like Tanimoto, Cosine, Dice, or Euclid to perform different search tactics. These measures are generated based on the amount of 1s counted in each fingerprint and the number of 1s in common between the two fingerprints. The most commonly used coefficients for similarity search were reported in table 2.1.

Fingerprints have been also learnt from molecular graphs using CNNs as reported in [33]. In Duvenaud et al. work, a single convolutional layer with softmax activation is used in place of the hashing function to produce the bits indexing of a atom neighborhood collected in the same way as circular fingerprints do. Authors report very good performance in predicting both solubility and toxicity from two purposely defined data sets, but the approach suffers from a high computational cost when compared with direct use of circular fingerprints.

Clustering has been described as "the art of finding groups in data" and is widely used in the pharmaceutical industry pharmaceutical industry to design different representative sets. The most common uses common of representative sets might be as training sets in the development of different structure-activity models and for screening in different biological screens. In both cases, I assume

that the centroid of the cluster is a good representative member of the corresponding cluster. It is therefore of great importance to be able to create homogeneous clusters consistently and to treat both small and very large sets equally well. In fact, at the beginning of my study, the K-means algorithm was used to identify activity thresholds in order to label the dataset. This technique, as as was explained in detail in section 4.1, was initially used to identify the correct IC50 thresholds to be used to label the downloaded data.

Table 2.1: Summary of the most used similarity metrics.

Measure	Range	Formula
Cosine	$0.0, 1.0$	$\frac{c}{\sqrt{(a+b)*(b+c)}}$
Dice	$0.0, 1.0$	$\frac{2.0*c}{((a+c)+(b+c)}$
Euclid	$0.0, 1.0$	$\sqrt{\frac{c+d}{a+b+c+d}}$
Forbes	$0.0,$	$\frac{c*(a+b+c+d}{((a+c)*(b+c)}$
Hamman	$-1.0, 1.0$	$\frac{(c+d)-(a+b)}{(a+b+c+d}$
Jaccard	$0.0, 1.0$	$\frac{c}{a+b+c}$
Kulczynski	$0.0, 1.0$	$0.5*(\frac{c}{a+c}+\frac{c}{b+c})$
Manhattan	$0.0, 1.0$	$\frac{(a+b)}{a+b+c+d}$
Matching	$0.0, 1.0$	$\frac{c+d}{a+b+c+d}$
Pearson	$-1.0, 1.0$	$\frac{(c*d)-(a*b)}{\sqrt{(a+c)*(b+c)*(a+d)*(b+d)}}$
Rogers-Tanimoto	$0.0, 1.0$	$\frac{c+d}{(a+b)+(a+b+c+d)}$
Russell-Rao	$0.0, 1.0$	$\frac{c}{a+b+c+d}$
Simpson	$0.0, 1.0$	$\frac{c}{min((a+c),(b+c))}$
Tanimoto	0.0,1.0	$\frac{c}{a+b+c}$
Yule	$0.0, 1.0$	$\frac{(c*d)-(a*b)}{(c*d)+(a*b)}$

Chapter 3

Target selection

My study is part of a broader research aimed at screening new compounds with anti-cancer properties. It is well known that protein kinases are key regulators of cellular function and constitute one of the largest and most functionally diverse families of proteins.
By adding phosphate groups to the substrate, they regulate the activity, localization, and overall function of many proteins, and serve to orchestrate the activity of nearly all cellular processes. Kinases are involved in signal transduction and coordination of complex functions such as the cell cycle. In particular, CDK1 is a central regulator that guides cells through G2 phase and mitosis.

Diril et al. generated a conditional-knockout mouse model to study the functions of CDK1 in vivo [32]. From this study, it was found that the low presence of CDK1 in the liver confers complete resistance against tumorigenesis induced by activated RAS and P53 (most frequently mutated tumor suppressor gene in human cancers) silencing. The choice of the target family was also guided by its importance in genetic mutations. In particular, one of the most extreme pathways for cancer development and progression is the mutation of various genes, including kinases. Mutated kinases can become constitutively active, and thus cause various cellular abnormalities, leading to the initiation or growth of cancer. Probably the best-known mutated kinase is BRAF (a human gene encoding a protein called B-Raf), which is frequently mutated on Val-600 (p.V600E) and is a driver mutation in several cancers, including colorectal cancer, melanoma, and thyroid cancer [25] .

Cyclin-dependent kinases (CDKs) are protein kinases marked by the need for a separate subunit - a cyclin - that provides essential domains for enzymatic activity. CDKs play important roles in the control of cell division and modulate transcription in response to various extra- and intracellular cues. Evolutionary expansion of the CDK family in mammals has led to the division of CDKs into three cell cycle-related subfamilies (CDK1, CDK4, and CDK5) and five transcriptional subfamilies (CDK7, CDK8, CDK9, CDK11, and CDK20). Unlike the prototype Cdc28 kinase in budding yeast, most of these CDKs bind one or a few cyclins, consistent with functional specialization during evolution. This review summarizes how, although CDKs are traditionally separated into cell cycle or transcriptional CDKs, these activities are often combined in many family members. Not surprisingly, deregulation of this family of proteins is a hallmark of several diseases, including cancer.

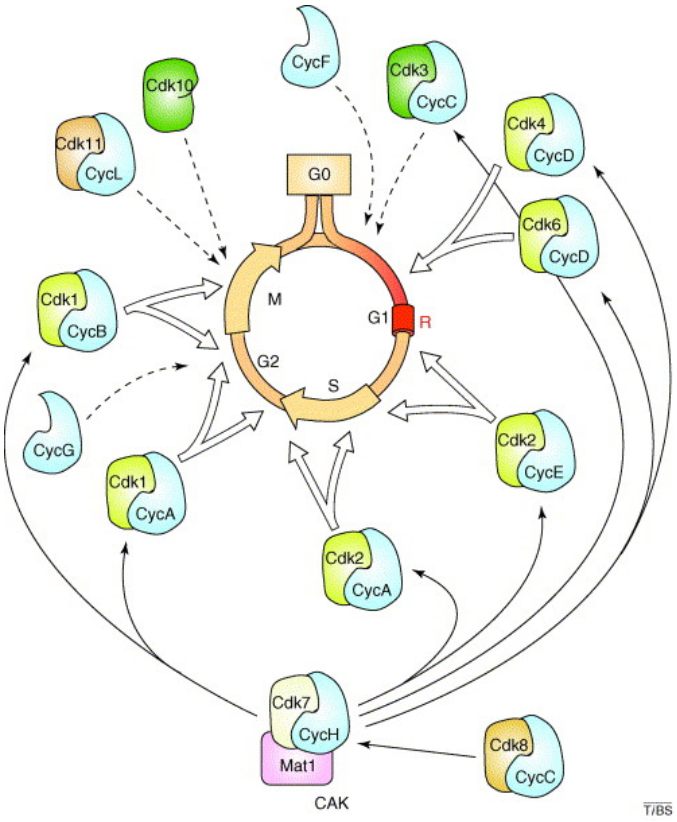


Figure 3.1: CDKs involved in cell cycle. Proposed roles of CDK–cyclin complexes in the mammalian cell cycle [62]

It is evident from the work of Lim et al. that CDKs, cyclins, and CKIs are more than simple regulators of the cell cycle [57]. They

are multifaceted proteins with important functions in processes that are distinct from the major events of cell division. However, rather than labeling these as "cell cycle-independent roles" one should appreciate that most of these emerging functions are closely intertwined with the cell cycle. For example, cell cycle regulators modify transcription to achieve differential expression of gene clusters appropriate to the proliferative state of the cell; preselect DNA repair mechanisms to utilize the most appropriate form of repair in accordance with the cell cycle period; control degradation to ensure timely destruction of cell cycle proteins; they activate methyltransferases to impart epigenetic marks on newly synthesized histones and DNA; they vary metabolic pathways to provide the level of energy needed to drive cell cycle events; and they target self-renewal or differentiation factors to dictate the outcome of cell division in stem cells.

In systems that are not directly related to the cell cycle, the characteristic fluctuation in the activities of cell cycle regulators can be reused for different purposes. For example, the changing activities of CDK/cyclin complexes are valuable for achieving orderly progression through the Pol II RNA-mediated transcription cycle. In light of the enormous amount of new information generated in recent years, the study of cell cycle regulators is certainly far from being a mature field, and the continuing quest toward understanding the full repertoire of their physiological functions is bound to reveal many more surprises along the way[57].

In the first step of my work, aimed at verifying the feasibility of the approach, I focused my attention on CDK1 as protein target for several reasons: first of all for the large amount of available bioactivity data, necessary to perform a training of a deep neural network; on the second and for the importance of the target, that is given by its validation as drug target. It is an archetypal kinase acting as central regulator that drives cells through G2 phase and mitosis. Its importance in tumorigenesis has been demonstrated by the evidence that, unlike other CDKs, loss of CDK1 in the liver confers complete resistance against tumor formation demonstrating its role in the cancer development.

Analysis of human tumor samples has revealed that the expression of several CDK and cyclins is often upregulated. However, several studies have indicated that none of the previously tested CDK can completely prevent tumorigenesis. Therefore, Diril et al. sought to determine whether loss of CDK1 can prevent cell transformation and tumor development in vivo.
From experiments performed in vivo, on mouse specimens, in which in a part of the colony CDK1 was inhibited and in the remaining part not, they could see that tumor proliferation could not occur in the absence of CDK1. They also assessed by histological analysis that the livers of mice in which CDK1 activity was inhibited appeared healthy and completely similar to those of healthy untreated mice.
Therefore, they concluded that cell proliferation in transformed cells is dependent on CDK1, and their results indicate that CDK1 is required for the growth of liver tumors in vitro and in vivo.
Validation of these results in vivo demonstrated that liver tumors do not form in the absence of CDK1. Because CDK1 appears to be essential for proliferation in every cell type and tissue tested, its inactivation would prevent tumor formation and propagation. This result is in contrast to other CDK whose inactivation had little effect on tumor formation. Thus, their results indicate the potential of CDK1 inhibitors in cancer therapy if we can prevent harmful side effects resulting from inadvertent interference with essential CDK1 functions in proliferative tissues[32].

Choosing CDK1 as a starting point, the colleagues at the Fondazione RI.MED, have selected other 19 proteins, reported in figure 3.1, among the most similar to CDK1 belonging to the family of kinases, with the method explained below. This method consisted of the IFPs Tanimoto Similarity calculation for protein swith high similarity to CDK1. The binding site similarity was calculated on both aminoacid sequence and interaction patterns with known ligands (experimental data of relative crystallography to the ligand-receptor interaction). I took the top nineteen protein with a similarity coefficient ≥ 0.80 . This was used to verify whether with the hypothesized embedding the model was able to generalize correctly and therefore to obtain a precise classification of activity on 20

different targets but with very similar binding site between them.

Several assumptions and hypotheses, reported in detail in the chapters of implementation of the datasets, led to the creation of two key databases that could be used to train machine learning and deep learning models. The first batch consisted of around 8000 molecules that were merely designated as active or inactive in relation to the CDK1 target. The second one consisted of roughly 90000 molecules that were tagged as active or inactive in relation to 20 target proteins that were identified as kinases with binding sites that were more similar to those of CDK1. The pipelines of the creation of the two datasets are detailed in the following sections, respectively in Section 5.1 and Section 6.2.

Protein name	Binding Pocket sequence	Best IFPs Tanimoto Similarity
CDK1	EKIGEGTYGVVYKVAMKKITAIREISLLKELRPNIVSLQDVYLI FEFLS_MDLKKYLDSFCHSRRVLHRDLKPQNLLILADFGLA	-
CDK2	EKIGEGTYGVVYKVALKKITAIREISLLKELNPNIVKLLDVYL VFEFLH_QDLKKFMDAFCHSHRVLHRDLKPQNLLILADFGLA	0.96
JAK2	QQLGKGNFGSVEMVAVKKLDFEREIEILKSLQDNIVKYKGVK LIMEYLPYGSLRDYLQKYLGTKRYIHRDLATRNILVIGDFGLT	0.87
ALK	RGLGHGAFGEVYEVAVKTLDFLMEALIISKFNQNIVRCIGVFI LLELMAGGDLKSFLREYLEENHFIHRDIAARNCLLIGDFGMA	0.86
MELK	ETIGTGGFAKVKLVAIKIMRIKTEIEALKNLRQHICQLYHVFM VLEYCPGGELFDYIISYVHSQGYAHRDLKPENLLFLIDFGLC	0.86
INSR	RELGQGSFGMVYEVAVKTVEFLNEASVMKGFT_HVVRLLGV LVVMELMAHGDLKSYLRSYLNAKKFVHRNLAARNCMVIGD FGM_	0.86
JNK3	KPIGSGAQGIVCAVAIKKLRAYRELVLMKCVNKNIISLLNVYL VMELMD_ANLCQVIQMHLHSAGIIHRDLKPSNIVVILDFGL_	0.86
ITK	QEIGSGQFGLVHLVAIKTIDFIEEAEVMMKLSPKLVQLYGVCL VFEFMEHGCLSDYLRTYLEEASVIHRDLAARNCLVVSDFGMT	0.83
ACK	EKLGDGSFGVVRRVAVKCLDFIREVNAMHSLDRNLIRLYGVK MVTELAPLGSLLDRLRKYLESKRFIHRDLAARNLLLIGDFGLM	0.83
CDK6	AEIGEGAYGKVFKVALKRVSTIREVAVLRHLEPNVVRLFDVT LVFEHVD_QDLTTYLDKFLHSHRVVHRDLKPQNILVLADFGL A	0.83
GSK3B	KVIGNGSFGVVYQVAIKKVFKNRELQIMRKLDCNIVRLRYFN LVLDYVP_ETVYRVARHYIHSFGICHRDIKPQNLLLLCDFGSA	0.82
IRAK4	NKMGEGGFGVVYKVAVKKLQFDQEIKVMAKCQENLVELLG FCLVYVYMPNGSLLDRLSCFLHENHHIHRDIKSANILLISDFGL A	0.82
PDK1	KILGEGSFSTVVLYAIKILYVTRERDVMSRLDPFFVKLYFTYF GLSYAKNGELLKYIRKYLHGKGIIHRDLKPENILLITDFGTA	0.82
MAP2K1	SELGAGNGGVVFKMARKLIQIIRELQVLHECNPYIVGFYGASI CMEHMDGGSLDQVLKKLREKHKIMHRDVKPSNILVLCDFGV S	0.82
ERK2	SYIGEGAYGMVCSVAIKKIRTLREIKILLRFRENIIGINDIYIVQD LME_TDLYKLLKTYIHSANVLHRDLKPSNLLLICDFGLA	0.82
EGFR	KVLGSGAFGTVYKVAIKELEILDEAYVMASVDPHVCRLLGIQ LIMQLMPFGCLLDYVREYLEDRRLVHRDLAARNVLVITDFGR A	0.82
CLK2	STLGEGTFGRVVQVALKIIAARLEINVLEKINNLCVQMFDWCI SFELLG_LSTFDFLKDFLHDNKLTHTDLKPENILFVVDFGSA	0.82
CHK1	QTLGE_____VQLVAVKIVNIKKEICINKMLNENVVKFYGHYL FLEYCSGGELFDRIEPYLHGIGITHRDIKPENLLLISDFGLA	0.81
CK2a1	RKLGRGKYSEVFEVVVKILKIKREIKILENLRPNIITLADIALVF EHVN_NTDFKQ__LYCHSMGIMHRDVKPHNVMILIDWGLA	0.81
DYRK1A	SLIGKGSFGQVVKVAIKIIQAQIEVRLLELMNYYIVHLKRHCL VFEMLS_YNLYDLLRNATPELSIIHCDLKPENILLIVDFGSS	0.80

Table 3.1: Kinases list chosen for the task with tanimoto similarity coefficient of the pocket.

Chapter 4

1D and 2D CNN for classification on CDK1

In this chapter, I report the results obtained from the study oriented to the identification of the best performing molecular fingerprints. I built networks to classify as active or inactive molecules on a single target(CDK1). The classification architectures I have proposed are convolutional networks (CNNs) that use input tensors consisting of combinations of molecular fingerprints. First, I tested individual fingerprints at different lengths using one-dimensional CNNs to identify the best performing length. Then, I tested all possible combinations of fingerprints as regards both their lengths and types using two-dimensional CNNs, each trained on a combination of different fingerprints with the same size for the same compound, to take into account all the different information coming from those descriptors at once.

In general, different patterns are searched for in each fingerprint kind, and also the same pattern is searched in different ways. Both networks consist of 4 convolutional layers with 512, 256, 128, 64 filters respectively with ReLU activation, each followed by a 2x2 Max Pooling, while they differ only in the convolutional kernel dimensions. Classification is achieved through a MLP with 1024, 512, and 256 ReLU units respectively, while the output is a sigmoidal unit.

4.1 Dataset implementation

The data used in my experiments where extracted from the well known CheMBL molecular database [1]. Biological activity of the tested compunds was measured using the *half maximal inhibitory concentration* parameter (IC_{50}) that is the amount of substance which is needed to inihibit the target protein (i.e. CDK1) by one half. A molecule has been considered active when $IC_{50} \leq 9\ \mu M$, otherwise it is inactive.

Data preparation was accomplished using the KNIME data analysis platform [16], and a workflow was implemented to prepare both the training and the test set. Activity data for 1830 compounds on the CDK1 target were taken from the *CHEMBL308* ID were CDK1 is considered as a single protein, and the *CHEMBL1907602* ID were it is considered as a protein complex.

Once all the data were obtained, using the Knime software, I implemented a workflow that was able to read the csv files containing the ChEMBL IDs and SMILES (Figure 4.1, and that could generate the seven types of molecular fingerprints used as input for the network.

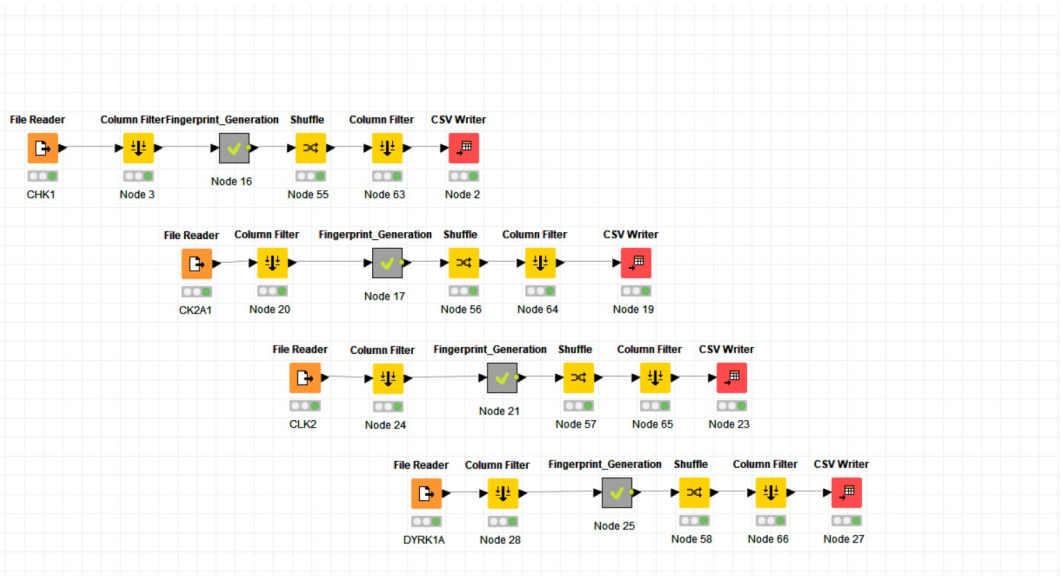


Figure 4.1: Knime workflow

In Figure 4.2 were reported in detail how 'Fingerprint generation' node works and the seven fingerprint generated. The first node on the left is *"Molecule Type cast"* that converts all cells of a chosen

string column into several one of several molecule types, such as Mol2, PDB, SDF, CML, HELM, SLN, Smiles, Smarts, or Rxn. This node is connected with *"RDKit Fingerprint Molecule"* that generates hashed bit-based fingerprints for an input RDKit Mol column and appends them to the table. Several fingerprint types are available, I choose RDKit, Morgan, Layered, Torsion, AtomPair, FeatMorgan and ECFP4 because in the early stages of the work I found them to be the best performing for the task . Not all settings are used for each type. these nodes will modify according to the selected fingerprint type and settings that are not supported by a fingerprint type will be disabled/hidden. The settings by which a fingerprint is generated are made available as column properties and can be viewed with the RDKit Interactive View node. The last fingerprint generation node always creates a new column containing a fingerprint for the molecules but this time the calculations are based on the CDK toolkit instead of RDKit's one.

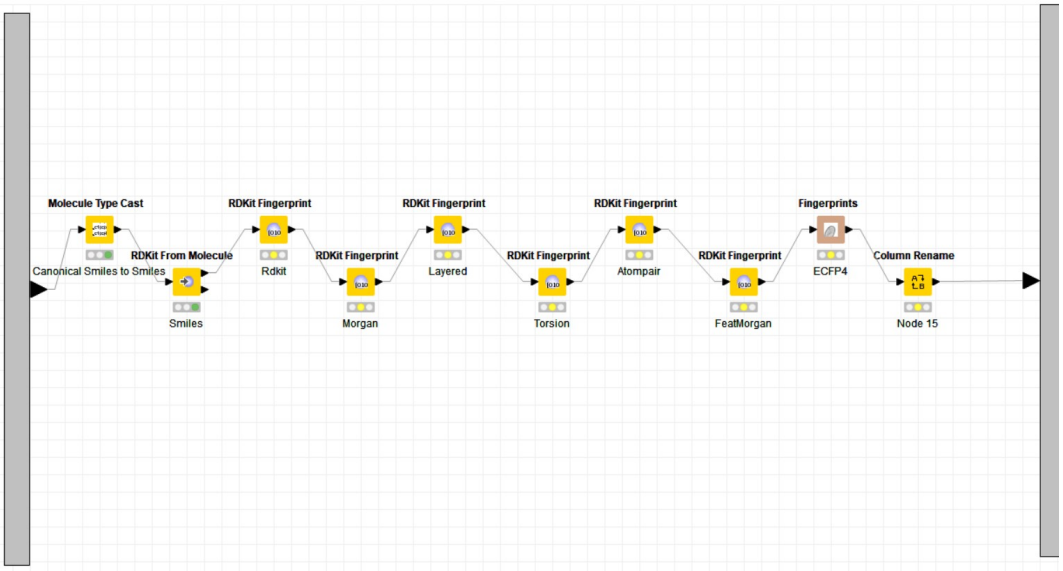


Figure 4.2: Fingerprint generation node

In figure 4.3 I see how in output from the fingerprint generation methanode I get a table that reports the molecule graph needed for fingerprint generation.

At first, incomplete data were deleted; yielding a total of 1707 molecules that were partitioned into a training set consisting of 1432 molecules with a perfect balancing between the two class labels ($1:1$ ratio of active to inactive). Particularly, 716 active samples

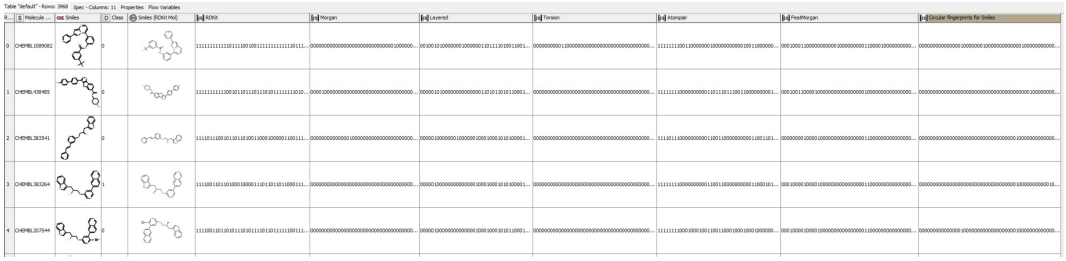


Figure 4.3: Output of fingerprint generation node

and 662 inactive ones were selected from the *CHEMBL308* ID, while 54 inactive samples were selected from the *CHEMBL1907602* ID. The test set was made as a whole by 175 inactive molecules coming from the *CHEMBL1907602* ID, and 100 active samples coming from the *CHEMBL308* ID.

Data in the two CheMBL IDs were searched for duplicates that were removed to avoid repetitions in both the training and the test set. However it is worth noting that in the same data set there may be two times the same molecule with very different IC_{50} value coming from two different biological assays. I have not used data augmentation because it is not possible to generate molecular fingerprints and predict whether they are active molecules or not in a specific biological assay but I have used 5-cross validation.

4.2 Proposed Architectures

The 1D CNNs were trained on single fingerprints; seven networks were trained, one for each tested fingerprint type. I selected only 1024 bit fingerprints as a good trade-off between compactness and expressivity. Low size fingerprints are too small to allow the network learning their features properly, while 2048 or 4096 bit fingerprints require models with very high capacity whose training is difficult. In the 2D CNNs each compound was represented by a combination of different fingerprints arranged as a bi-dimensional $\{1; -1\}$ matrix. The 2D CNNs have been trained on all possible combinations, ranging from fingerprint couples to a unique 7×1024 matrix enclosing all the fingerprints in a single descriptor. The intuition behind this architecture is that a fingerprint ensemble represents in a single tensor all the structural properties of a compound

that concur to its bioactivity on the target. On the other hand, a fingerprint combination can attain a high redundancy because the same pattern is encoded in several rows of the resulting matrix so I'm not guaranteed that the more fingerprints are present in the 2D descriptor the more accuracy I will obtain after training the network.

Both 1D and 2D networks consist of 4 convolutional layers with 128/64/32/16 filters per layer, and ReLU activation, each followed by a 2x2 Max Pooling, while they differ only in the convolutional kernel dimensions. Such networks have 512/256/128/64 filters respectively for each convolutional layer, while the number of filters per layer in the 1D CNNs used for direct classification are 128/64/32/16.

Classification is achieved through a MLP with 1024/512/256 ReLU units per layer respectively, while the output is a sigmoidal unit. The number of filters, kernel size, and number of MLP neurons were obtained by searching for hyperparameters. The 1D CNN architecture is shown in figure 4.4, while the 2D architecture is depicted in figure 4.5.

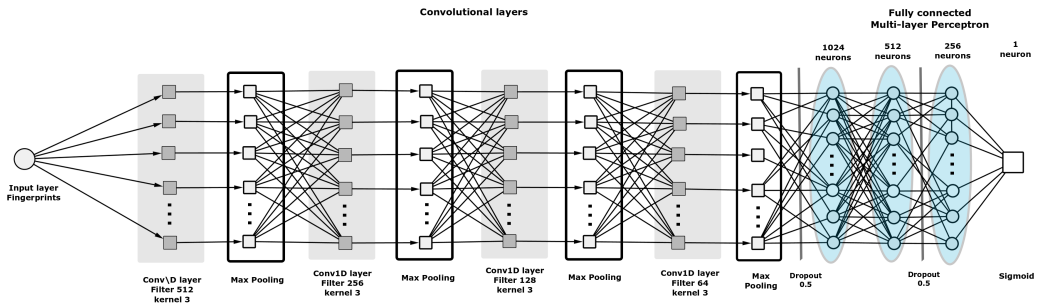


Figure 4.4: 1D CNN. One-dimensional convolutional architecture used to test fingerprints of different sizes (256/512/1024)

Molecular fingerprint generation acts as a transform on the molecular structure from the spatial domain to a suitable Vector Space

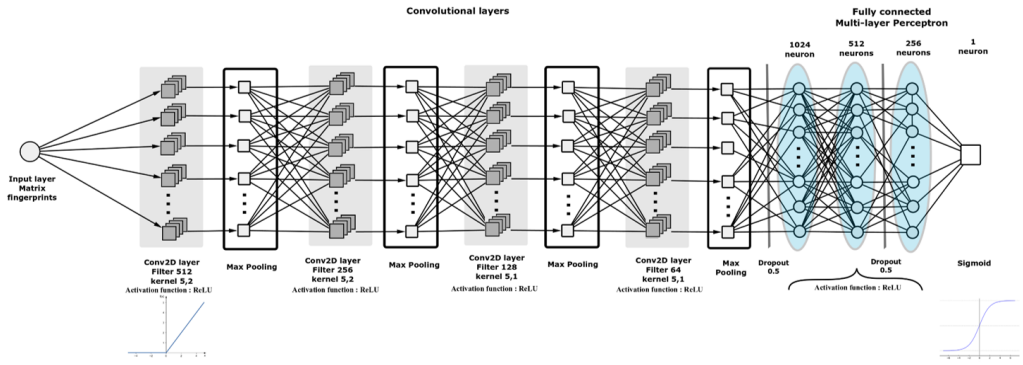


Figure 4.5: 2D CNN. Bi-dimensional convolutional architecture used to test fingerprints arrangement of different sizes (256/512/1024)

Representation. A fingerprint represents the corresponding molecule "as a whole" that is it conveys information about the presence of a particular substructure but not on its exact position or its repetition in different sites of the same molecule. Moreover, I wanted to perform a binary classification between active and inactive compounds, and biological activity is mostly related to the presence/absence of particular substructures which in turn are well suited to bind to the target protein. As a consequence, a CNN architecture appeared to be the best choice to classify molecular fingerprints.

Hyperparameters tuning was performed as a grid search in the following sets of values; Convolutional filters tested were [1024, 512, 256, 128, 64] in combination with all Keras padding value; learning rate were multiplied by 10 in the ranges $[10^{-6}, 1; 2 \cdot 10^{-5}, 0.2]$. Dropout probabilities where in the range $[0.2, 0.9]$ with step 0.1, all the available optimizers in Keras were tested. Bi-dimensional tested kernel sizes were [(20,2), (20,1), (15,2), (15,1), (5,2), (5,1), (4,2), (4,1), (3,2), (3,1)], while 1D tested kernels were $\{2, 3, 4\}$. Batch sizes were doubled in the range $[8, 128]$. Early sopping was used to devise training epochs. Table 4.1 shows the best choices for all the hyperparameters. Due to the low number of samples, small

Table 4.1: Hyperparameters setting, used in all experiments.

Optimizer	Learning rate	Dropout	Kernel size 2D	Kernel size 1D	Batch size	Epochs	Padding
Adamax	0.0002	0.5	(5,2), (5,1)	3	64	55	Same

size fingerprints were tested with a number of epochs greater than 55; retraining was performed with 70, 100 and 120 epochs, and the

minimum loss was achieved with 100 epochs. No overfitting was encountered with this setup. Hyperparameter optimization took about 150 hours to be accomplished on a GPU NVIDIA GTX1060 6 GB, 1280 CUDA Cores, while each experiment took about 20 minutes. These times are reasonable and acceptable for the proposed task.

4.3 Results and discussion

The firs set of experiments where devoted to devise the best performing fingerprint type/size in predicting biological activity, and 1D CNN was used. Table 4.2 reports the best test results for each fingerprint size along with its type. Here and in the following tables, best results are highlighted in bold. The table reports the achieved test accuracy, the F1-score, and the AUC value, which is used commonly when comparing two approaches in the drug design literature. Both a SVM and a Random Forest model were trained

Table 4.2: Results of 1D CNN on the test set

Length	Fingerprint	Accuracy	Loss	F1-score	AUC
1024	Layered	0.9100	0.54	0.8700	0.9453
512	**Layered**	**0.9272**	**0.4447**	**0.9000**	**0.9610**
256	Torsion	0.8654	0.5456	0.831	0.9481

on my data sets to validate the performance of my model. The results of such experiment are reported in figure 4.6. As it was expected, ML approaches have a very poor accuracy performance (SVM = 0.9081, RF = 0.9081) if compared to ours best architecture (0.9345), despite the better AUC value shown in figure 4.6.

The second round of experiments was aimed at devising the best fingerprint combination/size for biological activity prediction using 2D CNN. The idea behind this experiments is that different fingerprints for the same molecule contain many different patterns, which in turn describe different molecular substructures. Also different sizes correspond to patterns with variable length. As a consequence, a set of fingerprints arranged as a 2D matrix can act as a better descriptor for molecular substructures than a single one can do. Table 4.3 reports the overall results for different fingerprint

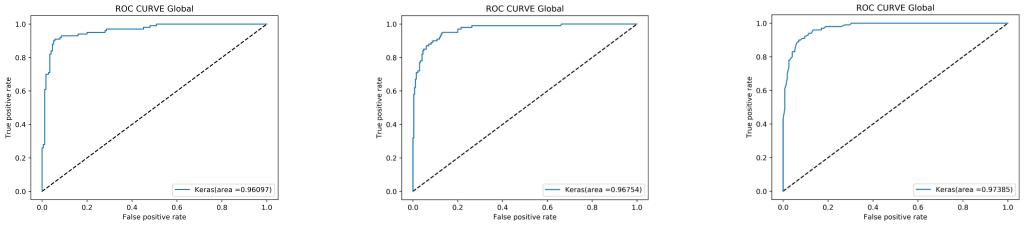


Figure 4.6: ROC Curves comparison of the proposed architecture with classical ML approaches; (a) best performing 1D CNN (L-512); (b) SVM; (c) Random Forest.

Fingerprints	Accuracy	Loss	F1-score	AUC
M,L	**0.9200**	**0.5600**	**0.8800**	**0.9563**
R,M,A	0.900	0.6800	0.8600	0.9527
M,A,L,F	0.9200	0.6000	0.8877	0.9444
R,M,A,L,F	0.9163	0.6082	0.8820	0.9513
R,M,A,T,L,F	0.8945	0.6280	0.8557	0.9494

(a) 1024 bit fingerprints

Fingerprints	Accuracy	Loss	F1-score	AUC
M,F	0.8981	0.4463	0.8679	0.9555
M,T,L	**0.9345**	**0.3900**	**0.9117**	**0.9685**
R,M,T,F	0.9418	0.4268	0.9001	0.94
R,A,T,L,F	0.9127	0.4052	0.8867	0.963
R,M,A,T,L,F	0.9236	0.3950	0.9004	0.9774

(b) 512 bit fingerprint

Fingerprints	Accuracy	Loss	F1-score	AUC
L,F	0.9090	0.4087	0.8792	0.9655
R,L,F	**0.9127**	**0.4734**	**0.8846**	**0.9606**
R,A,L,F	0.9054	0.4914	0.8749	0.9572
R,M,T,L,F	0.8909	0.5380	0.8623	0.9624
R,M,A,T,L,F	0.8981	0.5982	0.8679	0.9537

(c) 256 bit fingerprint

Table 4.3: Results of the 2D CNN on the test set with different fingerprint length. Fingerprint types: *(R)DKit*, *(M)organ*, *(A)tompair*, *(T)opological Torsion*, *(L)ayred*, and *(F)eatMorgan*

sizes.

As it is reported in tables 4.2 and 4.3, the best performance is achieved with the set of Morgan, Topological Torsion and Layered 512 bit fingerprints (MTL-512). Layered fingerprints are always among the best performing descriptor regardless their size. Moreover, 512 Layered is exactly the best performing descriptor in the 1D CNN architecture. Regarding the 256-bit results, I choose to

emphasize the RLF model since I want to favor the model that performs better overall. The highlighted model, in particular, has the highest F1-score and accuracy, but loss and AUC are close. It is trivial to say that 512 bit is the input data size that best suits to the network capacity as it is defined by its architecture. As regards the fingerprint types, it is difficult to devise an exact explanation of the results due to the random process involved in the generation of molecular fingerprints. It is not possible to devise precise patterns in precise positions that are mainly responsible for the network performance. Anyway, I can say that Layered fingerprints have a particular hashing scheme that allows accommodating substructure information with high level of detail so it is reasonable that 1D CNN achieved its best performance using this kind of fingerprint. As regards the 2D CNN's performance, it is worth noting that MTL-512 fingerprints together span all the diverse criteria to search for patterns so it seems quite reasonable that such a triple produced the best result.

I further validated my architecture against the DeepVS network, which is presented in [78], and deals with VS versus CDK proteins even if there are some differences with my work.

DeepVS was trained on the CDK2 protein only; the authors tested their network with a subset of the *CHEMBL301* data set, which is extracted from the DUD-E data set (798 active molecules and 28,329 decoys). At first, the entire *CHEMBL301* data set that consists of 1528 compounds (956 CDK2-active molecules, and 572 inactive ones) was used to test the MTL-512 2D CNN. In this experiment my network achieved AUC=0.8030 that is a very good result when compared with AUC=0.82 achieved by DeepVS, which was trained purposely for CDK2. As some compounds in *CHEMBL301* are also active on CDK1, I removed explicitly all of them to stress the network performance. As a result, I obtained AUC=0.678, which shows an obvious decrease; this still remains a satisfactory result if related to human performances in VS, and also classical ML approaches.

The main novelty of this research relies on performing Deep Learning based VS starting from molecular fingerprints for CDK1 that is a very important biological target for its direct implication

in the etiology of various cancerous forms. One qualifying point of my approach is that fingerprints capture molecular structures according to different criteria and are already accepted as molecular descriptors by the chemoinformatics society. Another novelty of the approach is the use of fingerprint matrices, in order to keep direct information from single fingerprint and indirect information from the combination of the same. Their shape already makes them an embedding that lends itself perfectly to the intended use. Fine tuning of hyperparameters has been carried on along with several experiments with different fingerprint types and sizes.

Chapter 5

Tuned-MLP-Out architecture for classification on CDK1

The main aim of the research reported in this chapter is to assess a way to obtain a tight interaction among the molecular fingerprints used to represent the input ligand. To this aim a multi-branch architecture has been proposed with *parameter sharing* regularization where seven 1D CNN branches extract features from different fingerprints, and they are then merged in a unique MLP classifier.

It is very important to emphasize that the training was directed to obtain a high discriminative power in our model. In fact, I measured the TP/P ratio that is the number of true actives predicted, in order of decreasing probability, in a fixed percentage of the test set. This parameter is very important because, usually, the screening is performed on very large data sets heavily biased towards inactive molecules. Maximizing this parameter at the expense of class-specific sensitivity of active molecules means that the classifier correctly prioritizes active molecules while incorrectly classifying several inactive molecules. Early screening needs to ensure that all bioactive compounds are ranked in the first positions despite of the number of false positives, while a second screening round is aimed at increasing the prediction accuracy. A novel CNN architecture is presented to this aim, which predicts bioactivity of candidate compounds on CDK1 using a combination of molecular fingerprints as their vector representation, and has been trained suitably to achieve good results as regards both TP/P parameter and accuracy in different screening modes (98.55% accuracy in active-only selection, and 98.88% in high precision discrimination).

I performed two types of experiments regarding the training procedure, and several measures were collected to conceive the performance in both tasks. The first training procedure (*training scheme 1*) makes use of a classical ML approach for training where the ratio between training, validation and testset is strongly unbalanced towards the training set, keeping the ratio between the two classes unaltered in all of them. This is the correct choice if one wants to maximize the discriminative power of the network. On the other hand, the second procedure (*training scheme 2*) takes into account the fact that the general population of a dataset containing candidate compounds to be screened is strongly skewed towards inactive candidates. Consequently, I stressed network performance by using balanced training with many active compounds, and testing with a $1 : 50$ active/inactive ratio. Both schemes make use of 10-fold cross-validation in the training phase.

5.1 Dataset implementation

The data used in my experiments were extracted from the well known CheMBL molecular database [1]. Biological activity of the tested compounds was measured using the IC_{50} introduced in 4.1. The literature does not report a precise threshold to be used for labeling a compound as *active* or *inactive*. A good rule of thumb is that $IC_{50} < 1.0\ \mu M$ implies good bioactivity, while $IC_{50} > 10.0\ \mu M$ indicates definitely no bioactivity. Our task is a binary classification so I needed a crisp threshold to divide my data in two classes. As a consequence, I followed a typical ML approach in this respect, that is I devised the threshold from the data using K-Means clustering. I didn't have any knowledge in advance about the distribution of the IC_{50} values in my data. At first, the so called *elbow method* was applied to assess the correct number of clusters. This heuristics consists in clustering the data points $\boldsymbol{x}$ with a variable number of clusters k, while plotting the *Within-Cluster Sum of Squares*:

$$WCSS = \sum_{i=1}^{k} \sum_{\boldsymbol{x} \in C_i} (\boldsymbol{x} - \boldsymbol{\mu_i})^2$$

where C_i is the i-th cluster, and μ_i is its centroid. The plot will exhibit an "elbow" in correspondence of the optimal value for k. Our analysis resulted in choosing two clusters ($k = 2$), as I expected. Then I ran the K-Means algorithm with two clusters, and I obtained a threshold value for $IC_{50} = 7.414\ \mu M$, so a molecule was labeled *active* when $IC_{50} \leq 7.414\ \mu M$. It is worth noticing that this value was used merely for splitting the data in two classes. There is no chemical relevance in this threshold. Actually, it is the value of the two class centroids' average. The K-Means algorithm reported also the following results about the shape of the clusters, that are coherent with the literature:

- Active molecules: centroid at $IC_{50} = 0.91762\ \mu M$, upper bound at $IC_{50} = 0.971\ \mu M$
- Inactive molecules: centroid at $IC_{50} = 13.91221\ \mu M$, lower bound at $IC_{50} = 13.338\ \mu M$

I used the KNIME data analysis platform [16], to implement a preprocessing workflow for both the training and the test set. Activity data for 1830 compounds on the CDK1 target were taken from the *CHEMBL308* ID were CDK1 is considered as a single protein, and the *CHEMBL1907602* ID were it is considered as a protein complex. At first, incomplete data were deleted; the resulting data set was then made by 1720 samples. The data set was expanded using some compounds, which are active on some kinases with very different structure from CDK1. Also these molecules were extracted from CheMBL. In particular, I selected 2422 active compounds on TRKA (Tropomyosin receptor kinase A, CHEMBL2815), 50 active compounds on RIPK1 (Receptor-Interacting Protein 1, CHEMBL5464), 2825 active compounds on AKT1 (AKT Serine/ Threonine Kinase 1, CHEMBL4282), and 199 active compounds on LIMk1 (LIM Domain Kinase 1, CHEMBL5932). Duplicates have been removed from the original 5496 records returned by the queries thus obtaining 5452 inactive compounds on CDK1.

Two different schemes have been used for training even if 10-fold cross-validation has been used in both procedures. In the scheme 1, I adopted a classic strategy with an approximate 80%:10%:10% split for training, validation, and test set respectively with a $1:10$

active/inactive ratio. Validation set has been used for hyperparameters grid search while Test set has been used to evaluate the overall performance. In the scheme 2, the same data set as above has been divided in two almost equal parts (48% training, and 52% test set). Moreover the training data were split in training and validation set with a 90%:10% ratio. In the training data 720 compounds out of 3440 samples were active molecules, while just 80 active molecules out of 3720 compounds were present on the test set.

Finally I turned each fingerprint's 0 value in -1 to cope with the inherent sparsity of such a vector representation. In this way I maintained the binary information conveyed by each fingerprint, while avoiding unwanted bias of the output of the convolutional units when they receive an almost zero input.

5.2 The proposed architectures

The 1D CNNs were trained on single fingerprints; seven networks were trained, one for each tested fingerprint type. I selected only 1024 bit fingerprints as a good trade-off between compactness and expressivity. Low size fingerprints are too small to allow the network learning their features properly, while 2048 or 4096 bit fingerprints require models with very high capacity whose training is difficult. In the 2D CNNs each compound was represented by a combination of different fingerprints arranged as a bi-dimensional $\{1; -1\}$ matrix. The intuition behind this architecture is that a fingerprint ensemble represents in a single tensor all the structural properties of a compound that concur to its bioactivity on the target. On the other hand, a fingerprint combination can attain a high redundancy because the same pattern is encoded in several rows of the resulting matrix so I'm not guaranteed that the more fingerprints are present in the 2D descriptor the more accuracy I will obtain after training the network.

1D networks consist of 4 convolutional layers with 128/64/32/16 filters per layer and ReLU activation, as already presented in the previous chapter. Each layr is followed by a 2x2 Max Pooling, while they differ only in the convolutional kernel dimensions. Such networks have 512/256/128/64 filters respectively for each convo-

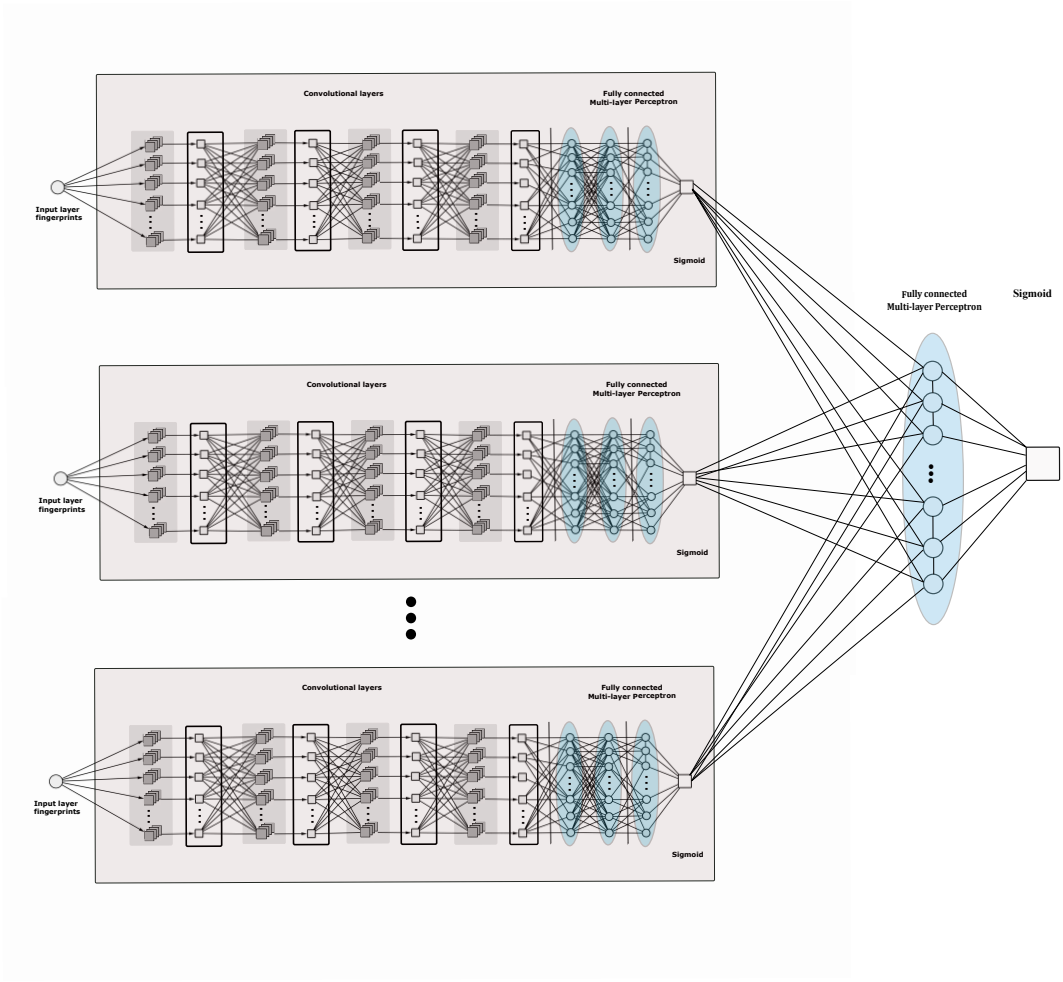


Figure 5.1: Tuned-MLP-Out. The complex architecture with MLP classifier.

lutional layer, while the number of filters per layer in the 1D CNNs used for direct classification are 128/64/32/16.

Classification is achieved through a MLP with 1024/512/256 ReLU units per layer respectively, while the output is a sigmoidal unit as I want binary classification. The number of filters, kernel size, and number of MLP neurons were obtained by searching for hyperparameters.

The architecture presented in this chapter are ensemble classifiers using the outputs of the 1D CNNs. Moreover, these networks exhibit a sort of inertia that is they attain both high sensitivity and balanced accuracy but it is not possible to stress their performance towards either mature or early screening task. This is mainly due to the intrinsic redundancy of the input fingerprints regardless the best performing combination. On the contrary, 1D CNN are more

flexible architectures than the 2D ones, reported in 4.2, and are implemented by low capacity models. Such networks suffer from the use of a single fingerprint, which might not encode properly the core bioactivity features of the compound due to its particular generation algorithm. As a consequence I resorted to two kinds of ensemble classifiers: the first one, which has been called *Voting*, is a pure voting mechanism where the output labels from the best performing 1D CNN for each fingerprint type are collected, and the final label is the one provided by the majority of the voting classifiers. The second scheme (called *Voting-MLP*) is a more refined version of the pure voting mechanism, where I trained again from scratch seven 1D CNNs, one for each fingerprint type, with 512/256/128/64 filters per layer, and the same MLP arrangement as regards their classification stage. All of them are connected in parallel as inputs of a unique MLP layer through the probability values associated to the sigmoidal outputs. Just three ReLU units were needed for the final classification layer. The whole Voting-MLP architecture is reported in figure 5.1.

Hyperparameters tuning was performed as a grid search in the following sets of values. Convolutional filters tested were [1024, 512, 256, 128, 64, 32, 16] in combination with all Keras padding value; learning rate were multiplied by 10 in the ranges $[10^{-6}, 1; 2 \cdot 10^{-5}, 0.2]$. Dropout probabilities where in the range $[0.2, 0.9]$ with step 0.1, all the available optimizers in Keras were tested. Bi-dimensional tested kernel sizes were [(20,2), (20,1), (15,2), (15,1), (5,2), (5,1), (4,2), (4,1), (3,2), (3,1)], while 1D tested kernels were $\{2, 3, 4\}$. Batch sizes were doubled in the range $[8, 128]$. Early sopping was used to devise training epochs and *model checkpoint* to saving best model after each epoch. Hyperparameter optimization took about 100 hours to be accomplished on a GPU NVIDIA TITAN Xp 12 GB, 3840 CUDA Cores, while each Voting-MLP experiment took about 2 hours.

5.3 Results and discussion

The balanced accuracy $bACC = (TP/P + TN/N)/2$ has been used for active/inactive discrimination, which is a binary classifi-

cation task. The $bACC$ value is the mean of sensitivity or *true positive rate* that is the ratio between the predicted positives TP and the labeled positives P, and specificity or *true negative rate* that is the ratio between the predicted negatives TN and the labeled negatives N. $bACC$ measures the performance in labeling each sample in the proper class. Also the *Matthews correlation coefficient* (MCC) was used as a discrimination measure. MCC is a well known index used for binary classification, that returns a value in $[-1; 1]$; for a 2×2 contingency table, that is a binary classifier's confusion matrix, MCC is related to the chi-square statistic as $\|MCC\| = \sqrt{\chi^2/n}$ where n is the number of observations. MCC thus measures the dependency of the predictions from the true (i.e. expected) labels. On the other hand, just sensitivity has been used in the active only selection task because I want to maximize correct prediction of active compounds in spite of accepting a relevant number of false positives.

Our models are compared with two state-of-the-art ML approaches for Virtual Screening that is Support Vector Machines (SVM), and Random Forests (RF) which form the baseline for my experiments. The parameters for both models where devised using a classical grid search. Particularly, a Radial Basis Function-SVM has been trained, and the best performing machines have $\gamma = 1$ for both training schemes, while the regularization parameter is $C = 5$ in the training scheme 1, while $C = 1$ and $\gamma = 0.1$ in the training scheme 2. SVM trained on *FeatMorgan* fingerprint performed the best in both the training schemes. The best performing RF used 100 estimators, and the *Gini* index for the training scheme 1 on the *FeatMorgan* fingerprint, while in training scheme 2 *Gini* index and 2 estimators.

The results of the best performing architecture for each task are reported in Table 5.1 and Table 5.2. The two tables show clearly that the reduced number of samples in the training scheme 2 along with the inherent class unbalancing in the data set reduce the absolute performance of the network due to an increase of false negatives. Even if both $bACC$ and sensitivity are acceptable, the *Loss* value doubles with respect to the training scheme 1. This reflects on all the global measures that are AUC, F_1 score, and MCC.

Table 5.1: Results for the active/inactive discrimination task, and Training scheme 1

Architecture	Bal. accuracy	Sensitivity	Loss	AUC	F1-score	MCC
Tuned-MLP-Out	**0.9880**	**0.9855**	**0.0405**	**0.9979**	**0.9510**	**0.9462**
Voting	0.9768	0.9710	0.2093	0.9920	0.8965	0.9033
CNN 1D (F)	0.9687	0.9710	0.0688	0.9904	0.8979	0.8813
CNN 2D (R-M-F)	0.9679	0.9565	0.0770	0.9912	0.8918	0.8817
Random Forest (F)	0.9510	0.8985	0.6405	0.9837	0.6065	0.8962
SVM (F)	0.9421	0.8985	0.7883	0.9868	0.8857	0.8731

Fingerprint types: *(R)DKIT,(M)organ, (F)eatMorgan, (L)ayered*

Table 5.2: Results for the active/inactive discrimination task, and Training scheme 2

Architecture	Bal. Accuracy	Sensitivity	Loss	AUC	F1-score	MCC
Tuned-MLP-Out	**0.9644**	**0.9625**	**0.0983**	0.9875	0.5519	0.5989
Voting	0.9639	0.9500	0.1523	**0.9889**	0.6379	**0.6694**
CNN 1D (F)	0.9579	0.9625	0.1398	0.9854	0.4709	0.5336
CNN 2D (T-L-E)	0.9525	0.9375	0.1054	0.9841	0.5192	0.5920
Random Forest (F)	0.8789	0.7750	0.6221	0.9541	0.6528	0.6540
SVM (F)	0.9208	0.8625	0.6221	0.9682	**0.6699**	0.6524

Fingerprint types: *(F)eatMorgan, (T)orsion, (L)ayered, (E)CFP4*

The winning architecture for both tasks is *Tuned-MLP-Out* because it takes into account all the fingerprint types, and manages the possible redundancies by training a shallow MLP classifier. Just one layer was always sufficient to achieve good classification, even if I tried different sizes for the hidden layers. Particularly, discrimination task was performed with 3 units, learning rate equal to 10^{-3}, and Adam optimizer, while active-only selection was accomplished using 5 units, learning rate equal to 10^{-4}, and Adamax optimizer. Active-only selection is achieved with a classifier with both higher capacity and lower learning rate than in the discrimination case. These values indicate a network that is more prone to overfitting than in the balanced case as it is less accurate. Also the 1D CNNs used by *Tuned-MLP-Out* have higer capacity than the best performing 1D CNNs alone.

The winning *Voting* architecture is the same for both tasks because it uses always the best 1D CNN for each fingerprint type. This network exhibits always the highest AUC value, which means that it tends to have a good balanced performance in every case. In fact, *Voting* has the lowest sensitivity when used for active-only selection, and it falls below the baseline, but it is one of the best ranked networks in terms of the $bACC$ value.

In Tables 5.3 and 5.4 I report an unconventional metric, which I used to identify the best models in terms of reliability of asset prediction. To measure explicitly the classifier's ability to prioritize the ligands, I reported also values of the ratio between the True Positives (TP) that is the number of correct predictions prioritized at the top $x\%$ of the test set, and the Positives (P) that is the total number of positives in the test set for each target. This parameter is crucial in VS procedures due to the huge number of candidates to be evaluated, so the drug designers require that a good VS procedure assigns the highest probability values to the very first candidates in the data set, in order to discard the remaining ones without further test. I compared my best performing architecture versus both SVM and RF also as regards the $TP/P\%$ value. Results are reported in table 5.3, and table 5.4 respectively for each training scheme. In particular, in training scheme 1 I were able to compute $TP/P\%$ from 1% to 10% because both training and test set were equally balanced. Just $TP/P1\%$, and $TP/P2\%$ were computed in the training scheme 2 because only 80 out of 3270 molecules were truly active that is a percentage of 2.4%. As a consequence, computing both $TP/P5\%$ and $TP/P10\%$ would have resulted in a artificial performance decrease by definition. Results are satisfactory. Our *Tuned-MLP-Out* network ranks at 100% in $TP/P1\%$ and $TP/P2\%$, just like RF and SVM. It is worth noticing that drug designers are more interested computing $TP/P\%$ for low percentages that implies screening very few candidates. For high percentages, the probability of a false positive prediction increases. Even if, my architecture misses just one active compound with respect to both SVM and RF in the case of $TP/P5\%$ both the shallow models exhibit a low $TP/P10\%$ value due to their reduced accuracy on the whole data set.

Experiments gave us some interesting insights on the use of different fingerprint types. The best performing 2D CNN is the one using the combination of *RDKit*, *Morgan*, and *FeatMorgan* fingerprints. Such a network has a good mix of accuracy, sensitivity, and a high AUC together with a very low Loss value. As already pointed out in the previous section, the network has a good general behaviour but it can not be pushed towards extreme performance

Table 5.3: TP/P parameter computed on the test set 1 (70 active molecules out of 701 compounds).

Architecture	TP/P 1%	TP/P 2%	TP/P 5%	TP/P 10%
Tuned-MLP-Out	7/7	14/14	34/35	65/69
Voting	7/7	14/14	34/35	61/69
CNN 1D (M)	7/7	13/14	33/35	62/69
CNN 2D (R-M-F)	7/7	12/14	32/35	61/69
RF(F)	7/7	14/14	35/35	63/69
SVM(F)	7/7	14/14	35/35	61/69

Fingerprint types: *(R)DKIT,(M)organ, (F)eatMorgan,*

Table 5.4: TP/P parameter computed on the test set 2 (80 active molecules out of 3720 compounds).

Architecture(Training 2)	TP/P 1%	TP/P 2
Tuned-MLP-Out	37/37	65/74
Voting	32/37	57/74
CNN 1D (F)	31/37	52/74
CNN 2D (T-L-E)	31/37	52/74
RF(F)	37/37	62/74
SVM(F)	32/37	55/74

Fingerprint types: *(F)eatMorgan, (L)ayered, (T)orsion, (E)CFP4*

Table 5.5: Performance of the *Tuned-MLP-Out* network on three data sets with 1%, 2%, and 5% active/inactive proportion respectively

Active/inactive	Bal.Accuracy	Sensitivity	Loss	AUC	F1-score	MCC
1%	0.7475	0.5000	0.5116	0.9700	0.5333	0.5289
2%	0.9671	0.9375	0.5114	0.9415	0.9009	0.8226
5%	0.9382	0.8780	0.0565	0.9991	0.9230	0.9196

Table 5.6: TP/P parameter computed on the test set with 1%, 2%, and 5% active/inactive proportion respectively.

Active/inactive	TP/P 1%	TP/P 2%	TP/P 5%
1%	4/8	-	-
2%	7/8	7/16	-
5%	4/8	8/16	20/41

in neither task I addressed in this work. Finally, 1D CNNs differ only in the fingerprint type used in the training phase. The *Layered* network showed the best *bACC*, while the *Morgan* one has the highest sensitivity. The reason for this difference in predictive ability lies in the different way of interpreting the molecular structure. The *Layered* fingerprint, using different layers of structural

analysis, seems to outperform in discriminating between active and inactive candidate compounds. The *Morgan/FeatMorgan* fingerprints represent a circular approach which uses either connectivity or feature invariants, and it has been outclassed by modern ECFP fingerprints as they are more accurate. Nevertheless, both 1D and 2D CNNs have the best performance when such descriptors are used to represent the candidate compounds. It is worth noting, in this respect, that in this work, I have tested the ability to recognize between active and inactive molecules, based on their IC_{50} value, and such a task requires a more discriminative power than other works in the literature in which fingerprints are compared on the basis of the distinction between actives and decoys.

Our approach has to be regarded as typical ligand-based one, while decoys are generated to validate docking-based algorithms. On the other hand, decoys are synthetic molecules whose mere structure could make them active on the target, and their use to train an active/inactive classifier could result in a poor discriminative power. Even if it is well known in the literature that a $1:10$ active/inactive ratio is a common value for *in silico* screening, I performed some stress tests on my best architecture that is *Tuned-MLP-Out* with training scheme 1. I aimed at devising its performance in a typical *in vivo* screening, where several problems can occur in essays, thus reducing the active/inactive ratio even to $1:100$. I resampled my data set to obtain three different data sets with varying active/inactive ratio: $1:20$, $1:50$ and $1:100$ respectively. The *Tuned-MLP-Out* network was trained using the scheme 1 on all of them. The results of active/inactive discrimination with these different proportions are reported in Table 5.5 while TP/P values from 1% to 5% are reported in Table 5.6. As expected, all the *bACC*, *Sensitivity*, and TP/P values decreased with respect to the results reported in Table 5.1. This is due to the extreme unbalancing between classes that is a hard issue for whatever learning algorithm. Nonetheless, the results are still positive, and this can be observed in Table 5.6. Apart from the $TP/P1\%$ for the 2% proportion data set that attains a 87.5% value, almost all the TP/P values drop to values that are close to 50%, but all the hits in wathever experiment ranked as the very first molecules in

terms of the output probabilities of the model so they still remain the first choice for the drug designer. Also in this case I did not compute TP/P values for test percentages greater than the true active/inactive proportion to avoid the computation of false low values due to the absence of active molecules.

Chapter 6

EMBER multi-fingerprint embedding

In very recent years, the debate in the field of the application of Deep Learning to Virtual Screening has focused on the use of neural embeddings w.r.t. classical descriptors to encode both structural and physical properties of ligands and/or targets. The attention on embeddings raised with the increasing use of Graph Neural Networks aimed at overcoming molecular fingerprints that are short range embeddings for atomic neighborhoods.
In this chapter, I show a deep convolutional architecture for evaluating the bioactivity of ligands on twenty protein kinases that have the most comparable binding sites to CDK1. The proposed architecture makes use of a suitable molecular embedding made by seven molecular fingerprints arranged as different "spectra" to describe the same molecule, see section 6.1, and it exploits the Depthwise Separable Convolution operator to reduce computational complexity. The data set is presented, and the architecture is explained in detail along with its training procedure. I report experimental results, and an explainability analysis to assess the contribution of each fingerprint to the different targets.

The major contributions of the presented work reported in this chapter can be resumed in the following.

- A suitable embedding is proposed that is made by multiple molecular fingerprints that have been generated using complementary ways to search for molecular substructures, and are stacked as the spectra of a sort of "molecular image"; such an embedding is aimed to exploit the ability of Convolutional

Neural Networks (CNN) in learning the proper features, as they do for images.

- A multi classifier has been developed to prove the previous claim, that performs very well in screening ligands on twenty protein kinases that are the ones with the most similar binding sites to CDK1; moreover my architectural design lowers the parameter number.
- A curated data set made by nearly 90000 ligands labeled as active/inactive against 20 Kinase target selected as the most similar to CDK1.

6.1 EMBER

A major contribution of my Ph.D. work is the introduction of EMBER, an embedding that is obtained using different molecular fingerprints bundled as the "channels" of the input tensor of a 2D CNN. This section is devoted to explain the motivations of my choice.
In my approach molecular fingerprints are regarded as different "spectra" of the same *molecular image* as it is possible to see in figure 6.1. To clarify, EMBER is the output of an input layer that arranges the seven fingerprints in tensor form, that is, as a multidimensional array, resulting in a tensor T_{ij}. In fact, different fingerprints collect information about atomic neighborhoods using heterogeneous criteria: moving along bond connected paths, exploring circular regions, encoding atom pairs and their bond distance, and so on. As a consequence, different fingerprints convey diverse structural information about the same molecule [69].

Different fingerprints collect information about atomic neighborhoods using heterogeneous criteria: moving along bond connected paths, exploring circular regions, encoding atom pairs and their bond distance, and so on. As a consequence, different fingerprints convey diverse structural information about the same molecule. In the research reported in the previous chapter I used seven fingerprint: *RDKit*, *Morgan*, *AtomPair*, *Torsion*, *Layered*, *FeatMorgan*, *ECFP4* as a result I verified that the different fingerprints are able to interact with each other, and such interaction was ex-

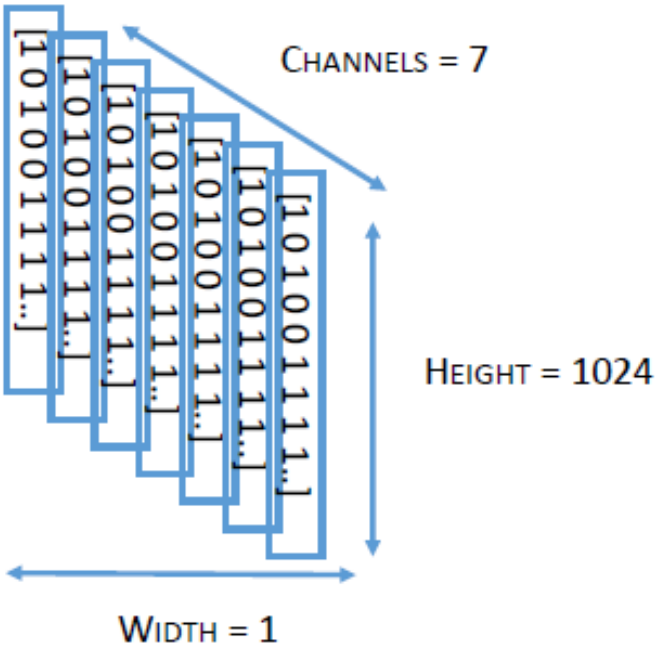


Figure 6.1: EMBER fingerprint channels of a molecule

ploited through *parameter sharing* regularization where seven convolutional branches are merged in a unique deep classifier, and the training procedure is in charge of merging the information conveyed by each single branch.

In this new work I started from the consideration that this kind of coupling is too loose, so I regarded the input fingerprints as the *features* of my molecular representation and used a CNN that is the ideal model for multi-channel image classification to perform my analysis. Molecular fingerprints have been widely used since many decades as a key technique in Virtual Screening, and they are no doubt an algorithmic embedding for molecular information with all the pros and cons of using such an approach. The fingerprint algorithm in two phases where at first information is collected from an atomic neighborhood, and then hashing is used to set the actual bits in the binary string, makes a molecular fingerprint "opaque" as regards direct explanation of the molecular graph, even if it retains global information about the presence of particular substructures in

the molecule. On the other hand, this algorithm ensures a similar computational process for each fingerprint family, and this enforces my claim about their use as channels of the same input tensor.
Finally, even if very recent research moves fast towards neural molecular embeddings, I wanted to use a solid reference framework for assessing the explainabilty of my approach due to the loss of explicit structural information induced by the use of fingerprints. In this respect, my multi-target classifier was analysed using well known frameworks for feature attribution reported in section 7.2, which is the standard approach in CNNs.

6.2 Dataset implementation

The targets considered in this task, were derived from the similarity approach reported below. This method consisted of the *IFPs Tanimoto Similarity* calculation for proteins with high similarity to CDK1 . The binding site similarity was calculated on both aminoacid sequence and interaction patterns with known ligands (experimental data of relative crystallography to the ligand-receptor interaction). I took the top twenty protein with a similarity coefficient ≥ 0.80 .

At the same time, to enrich inactive molecules library I used the opposite of the concept of similarity, *dissimilarity* (similarity coefficient < 0.1).

Of these twenty protein, I have extracted a portion of the data from CheMBL molecular database [68] where biological activity of the compounds was measured mainly using the IC_{50} described in 4.1. To identify the largest number of molecules I used all the other parameters available on CheMBL, like *inhibition constant* K_I(indirectly represents a measure of the affinity of a given substance with inhibitory activity for a given enzyme), and *dissociation constant* K_D that is a parameter that expresses the tendency of a compound to dissociate (i.e., to split to form other compounds consisting of molecules having a lower molecular weight than the molecules of the starting compound). A good rule of thumb used for both IC_{50} and K_I is that values less than 1.0 μM imply good bioactivity, while values greater than 10.0 μM indicate low or negligible

bioactivity. The literature does not report a precise K_D threshold to be used for labeling a compound as *active* or *inactive*. Therefore I clustered my data using the well known K-means algorithm with respect to the K_D value separately for each target, and devised a suitable threshold using the well known *elbow method* calculated with the *Within-Cluster Sum of Squares* (see 5.1). In this way, I obtained $k = 2$ for each target as it was expected, and I was also able to evaluate the centroids, and the extent of each cluster. Analyzing the clustering results, I obtained the value $K_D = 7\mu M$ as a good threshold to separate data correctly for each target.

Based on the available data in ChEMBL, inactive compounds for each protein evaluated in this study were too few to build any kind of model. Authors preferred not to use Decoys molecules for the inactives set, because of some known issues about their use, especially in DL methods. Madhavi Sastry et al. [84] had already reported a variable performance of decoys based on targets and method used for virtual screening in 2013. Then, more recent literature, mainly focused on the use of decoys data sets for DL has revealed some hidden biases when testing CNN virtual screening performance evaluation [23] . Moreover, Yang et al in their recent work [98] have pointed out the importance and at the same time the lack in publicly available DBs of sufficiently large and unbiased data sets to be used for robust AI models. Besides, the work enlightened once more how the use of decoys dataset to train the model presents some critical issues. On the light of these considerations and since this workflow is based on multi-target affinity approach, authors preferred to create their own dataset starting from ChEMBL database and exploiting dissimilarity metrics to enrich a diversity-based inactives DB. Therefore, in order to enrich the library of inactive compounds for each kinase, two different approaches were used. The first one was based on the collection of selective active ligands on targets presenting different ligand binding interaction pattern compared to the 20 reference ones in the study. The second one relied on the search for dissimilar compounds compared to co-crystalised kinase inhibitors.

Molecules retrieved by these two approaches were then evaluated to avoid the presence of duplicates. The advantage of using these

two different approaches, allowed the creation of a data set with a wide chemical space of active and inactive compounds.

Both methods are based on a workflow built with KNIME Analytics Platform [16] (Knime version 3.7.1).

In the first approach, the idea was to identify the kinases with less similar binding site compared to the 20 targets under investigation,for each of them, active and selective compounds were chosen. To perform this analysis, a workflow was built using the "3D-e-Chem - KLIFS" nodes, which return information on the entire human kinome from the "Kinase-Ligand Interaction Fingerprints and Structures" database [53] (KLIFS - release version 2.4, developed by the Pharmaceutical Chemistry Division - VU University Amsterdam).

In fact, this database contains detailed information about the structural kinase-ligand interactions relating to all the structures of the catalytic domains of the human protein kinases deposited in the Protein Data Bank.

The *Structures Overview Retriever* node was used to obtain the structure IDs of each reference kinase and all other human kinases (total 555). All the kinases data were processed as input by the *Interaction Finger print Retriever* node, to generate the protein-ligand interaction (IFP) fingerprints for subsequent chemoinformatics analysis. Additionally, this node corrects fingerprints for gaps and missing debris within the binding pockets, thus enabling free-for-all comparisons. Once the interaction fingerprints for each protein-ligand complex were obtained, a dissimilarity analysis was performed between each kinase's IFP, using the KNIME *Similarity Search* node. For this purpose, the Tanimoto coefficient was used as a method to calculate the distance (or dissimilarity) between each and all other human kinase IFPs. The results were also filtered, setting a coefficient range of [0 - 0.15]. For each kinase a list of proteins that satisfy this dissimilarity criterion was obtained and for each of them the compounds considered as actives in the literature was collected. In particular, for each kinase that was dissimilar to a reference one, only compounds with IC50 values $< 1.0\ \mu M$ were collected using ChEMBL Database v26.

Nevertheless, it was necessary to further expand the number

of inactive molecules using a second approach. This second approach consists of ligands dissimilarity search. Specifically, it was based on structurally different ligands compared to known active co-crystalised ligands for each protein used in this work.

The ligands in sdf format were downloaded from the Protein Data Bank[15, 19] (RCSB) for each 3D structure of the twenty proteins (see ligand code in table 6.1). Actives compounds were downloaded from crystal structures with resolution less than 2 angstrom.

In order to enlarge my dataset, 601810 small molecules were downloaded from the entire ChEMBL DB v26 and used for dissimilarity analysis with ligands obtained from the PDBs. All small molecules not relevant for classification purposes were removed according to the following criteria:

- molecular weight > 100
- number of carbon atoms > 10
- number of nitrogen atoms > 2
- number of oxygen atoms > 2
- at least one aromatic ring

Similarity analysis was conducted by calculating the Tanimoto coefficient using the ECFP4 fingerprints. Different compounds to kinase inhibitors previously downloaded were selected from the ChEMBL database considering a similarity coefficient between [0-0.1] in order to have a diverse chemotypes.

As result, the use of three different methods to enrich the inactive dataset allowed us to obtain a diverse set. The inactive set in fact was in the end mainly composed by the inactives found in ChEMBL to which other molecules actives on different proteins from the 20 selected and dissimilar from the PDB co-crystalised ligands of the 20 kinase of interest. Such an approach had two advantages. The first one was the possibility to have a large and diverse chemotypes space. Moreover, using these three different approaches, that is using different approaches to select molecules, I minimised the possibility to have analogue bias and artificial enrichment typical of the usage of decoys or not curated data sets.

Table 6.1: A summary of all proteins (active and inactive) obtained from preprocessing methods.

Target	PDB ID	Ligand Code*	Actives	Inactives
ACK	5ZXB	9KO	746	159775
ALK	6E0R	HKJ	1665	227247
CDK1	6GU2	F9Z	1241	124473
CDK2	6INL	AJR	1924	225087
CDK6	5L2S	6ZV	646	256561
INSR	5E1S	5JA	1423	195990
ITK	4RFM	3P6	1001	135007
JAK2	6M9H	J9D	5526	577409
JNK3	2B1P	AIZ	658	95252
MELK	6GVX	TAK	1215	246662
CHK1	6FC8	D4Q	2175	21763
CK2a1	6JWA	5ID	1053	10534
CLK2	6FYL	3NG	671	6800
DYRK1A	4YLK	4E2	1126	11274
EGFR	5GNK	80U	4757	47541
ERK2	6OPH	6QB	3525	35237
GSK3B	5F94	3UO	2578	25768
IRAK4	6EG9	OLI	2131	21282
MAPK2K1	4AN9	ACP; 2P7	1254	12508
PDK1	3NAX	MP7	1117	11166

* Most affine lingands

The overall data set was built starting from two separate sets. The first one was made by 64600 compounds that result inactive for all the targets. The second data set contains all the ligands that are active at least on one target. In the end, I merged the two data sets to obtain the final one that has a $1:100$ active/inactive rate, referred to the less abundant class (CDK6) (see table 6.1) .

This final data set consisted of 89373 molecules, and was separated into training set (68370 molecules), test set (13046 molecules), and validation set (7597 molecules), respectively.

The molecules were manipulated on Knime platform in order to generate the seven fingerprints used as the channels of my embedding used as input to the network.

Given the intrinsic sparsity of a molecular fingerprint I chose to transform the 0 bits in -1 in order to reduce the unwanted output bias of the convolutional units when they receive a zero input.

6.3 Proposed architecture

The proposed architecture for multitarget classification is based on a Depth Separable Convolution (DSC) operation as in GoogleNet or Inception networks. This kind of convolution consist on two different operation, a *depthwise convolution* i.e. a spatial convolution performed independently over each channel of an input, followed by a *pointwise convolution*, i.e. a 1x1 convolution, projecting the channels output by the depth-wise convolution onto a new channel space [24].

Training was conducted using both with and without 10-cross-validation in order to observe how the different distribution of data helped the learning phases in the classification task. I selected the 10-fold cross-validation training scheme, where a classic strategy was adopted with an approximate 80%:10%:10% split for training, validation, and test set respectively. A $1:100$ active/inactive ratio compared to the less abundant class (646 active compounds) was maintained in the three data sets.

The networks consist of nine Depth Separable Convolutional layers with 64/128− /256/512/512/256/128/64/32 filters per layer. The second and the last DSC layer are followed by a 2x2 Average Pooling layer. A Parametric ReLU activation has been used. This activation function adaptively learns the parameters of the rectifiers, and improves accuracy at negligible extra computational cost. PReLU (Parametric ReLU, that is ReLU with parameters) introduce a learnable parameter, different neurons can have different parameters, or a group of neurons can share one parameter.

$$\begin{aligned}\mathrm{PReLU}_i(x) &= \begin{cases} x & \text{if } x > 0 \\ \alpha_i x & \text{if } x \leq 0 \end{cases} \\ &= \max(0, x) + \alpha_i \min(0, x)\end{aligned}$$

If $\alpha_i = 0$, then PReLU degenerates to ReLU; if α_i is a small fixed value (such as $\alpha_i = 0.01$), then PReLU degenerates to Leaky ReLU (LReLU). In my work α_i has been set constant at 0.25. Classification is achieved through a MLP with 64/32/32 ReLU units per layer respectively, while the output is a 20 sigmoidal units because the probabilities of each class is independent from the other

class probabilities. The number of filters, kernel size, and number of MLP neurons were obtained by searching for hyperparameters. For this reason a *binary crossentropy* loss function has been used instead of the usual *categorical crossentropy*. This choice is reasonable because the network performs a “multi-label" “multi-class" classification task. The architecture is shown in Figure 6.2 and the summary of the model with the relative parameters is shown in Figure 6.3.

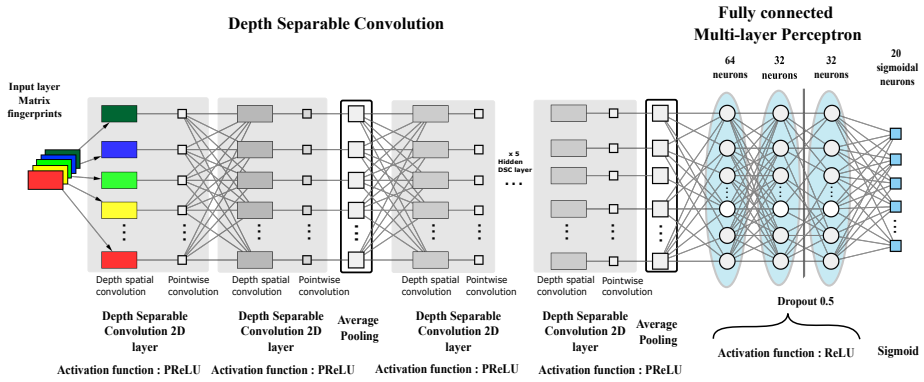


Figure 6.2: Depth Separable Convolutional architecture.

Hyperparameter tuning was performed as a grid search for the following values. Depth separable convolution filter [1024, 512, 256, 128, 64, 32, 16] with zero padding were tested. The learning rates tested were in the range $[10^{-5} - 10^{-1}]$ multiplied by 10. The batch sizes tested were in the range $[8 - 64]$. Early sopping was used to identify the optimal number of training epochs and model checkpoint was used to save the best model after each epoch. Hyperparameter optimization took 64 days and was performed using an NVIDIA TITAN Xp GPU, 3840 CUDA Cores. Notwithstanding the complexity of the architecture, each training session took about 6 hours due to the efficiency of the DSC convolution operation.

The architecture of my classifier is a deep CNN with nine layers using Parametric Rectified Linear Units (PReLU) for feature extraction, and a three-layers fully connected perceptron for actual classification. The network is trained on $7 \times 1024 \times 1$ input tensors that represent the seven 1024 long fingerprints stacked as the channels of a 1024×1 image. Multi-target bioactivity prediction is a *multi-class, multi-label* classification, that is my classifier has to

assess also if a ligand is active at the same time on different targets. As a consequence, the output is a *vector label* that is a binary vector where the 1s indicate bioactivity with respect to a particular target.

In line with the most recent CNNs, I implemented the convolutional layers using *Depthwise Separable Convolution* (DSC) [24] to reduce the network parameters, and lower the computational load. The classical convolution operator computes an element of the output tensor Y by applying a kernel K with spatial extent $s \times s$ and depth d to the input tensor X:

$$\mathsf{Y}_{i,j,k} = \sum_{l=1}^{s} \sum_{m=1}^{s} \sum_{n=1}^{d} \mathsf{X}_{i-l,j-m,k-n} \mathsf{K}_{l,m,n}$$

Here I'm using the proper index notation for convolution without kernel flipping. In DSC, d *spatial* kernels $\mathsf{K}^{S}_{(h)}$ with $s \times s$ size compute 1-depth convolutions, and a $1 \times 1 \times d$ *depth* kernel K^{D} gives the final convolution output.

$$\mathsf{Y}^{(h)}_{i,j} = \sum_{l=1}^{s} \sum_{m=1}^{s} \mathsf{X}_{i-l,j-m,h} \mathsf{K}^{S}_{(h)\ l,m}, \ h = 1 \dots d$$

$$\mathsf{Y}_{i,j,k} = \sum_{n=1}^{d} \mathsf{Y}^{(h-n)}_{i,j} \mathsf{K}^{D}_{n}$$

It can be shown that DSC can reduce the number of parameters by a factor $1/s^2$ for each layer: my network was built using just 2,252 million parameters, as it is reported in figure 6.3 where the overall architecture is detailed.

6.4 Results and discussion

Table 6.2 and Table 6.3 report the results of the proposed multi-classifier on the test set. In particular, Table 6.2 reports the accuracy and loss values obtained for single target. The overall performance of the network remains high in terms of global accuracy

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 7, 1024, 1]	7
Conv2d-2	[-1, 64, 1024, 1]	448
SeparableConv2d-3	[-1, 64, 1024, 1]	0
PReLU-4	[-1, 64, 1024, 1]	1
PReLU-5	[-1, 64, 1024, 1]	1
Conv2d-6	[-1, 64, 1024, 1]	64
Conv2d-7	[-1, 128, 1024, 1]	8,192
SeparableConv2d-8	[-1, 128, 1024, 1]	0
PReLU-9	[-1, 128, 1024, 1]	1
PReLU-10	[-1, 128, 1024, 1]	1
AvgPool2d-11	[-1, 128, 1024, 1]	0
Conv2d-12	[-1, 128, 1024, 1]	128
Conv2d-13	[-1, 256, 1024, 1]	32,768
SeparableConv2d-14	[-1, 256, 1024, 1]	0
PReLU-15	[-1, 256, 1024, 1]	1
PReLU-16	[-1, 256, 1024, 1]	1
Conv2d-17	[-1, 256, 1024, 1]	256
Conv2d-18	[-1, 256, 1024, 1]	65,536
SeparableConv2d-19	[-1, 256, 1024, 1]	0
PReLU-20	[-1, 256, 1024, 1]	1
PReLU-21	[-1, 256, 1024, 1]	1
AvgPool2d-22	[-1, 256, 1024, 1]	0
Conv2d-23	[-1, 256, 1024, 1]	256
Conv2d-24	[-1, 128, 1024, 1]	32,768
SeparableConv2d-25	[-1, 128, 1024, 1]	0
PReLU-26	[-1, 128, 1024, 1]	1
PReLU-27	[-1, 128, 1024, 1]	1
Conv2d-28	[-1, 128, 1024, 1]	128
Conv2d-29	[-1, 64, 1024, 1]	8,192
SeparableConv2d-30	[-1, 64, 1024, 1]	0
PReLU-31	[-1, 64, 1024, 1]	1
PReLU-32	[-1, 64, 1024, 1]	1
AvgPool2d-33	[-1, 64, 1024, 1]	0
Conv2d-34	[-1, 64, 1024, 1]	64
Conv2d-35	[-1, 32, 1024, 1]	2,048
SeparableConv2d-36	[-1, 32, 1024, 1]	0
PReLU-37	[-1, 32, 1024, 1]	1
PReLU-38	[-1, 32, 1024, 1]	1
Conv2d-39	[-1, 32, 1024, 1]	32
Conv2d-40	[-1, 32, 1024, 1]	1,024
SeparableConv2d-41	[-1, 32, 1024, 1]	0
PReLU-42	[-1, 32, 1024, 1]	1
PReLU-43	[-1, 32, 1024, 1]	1
Flatten-44	[-1, 32768]	0
Linear-45	[-1, 64]	2,097,216
Linear-46	[-1, 32]	2,080
ReLU-47	[-1, 32]	0
Dropout-48	[-1, 32]	0
Linear-49	[-1, 32]	1,056
ReLU-50	[-1, 32]	0
Linear-51	[-1, 20]	660
Sigmoid-52	[-1, 20]	0

Total params: 2,252,939
Trainable params: 2,252,939
Non-trainable params: 0

Input size (MB): 0.03
Forward/backward pass size (MB): 41.06
Params size (MB): 8.59
Estimated Total Size (MB): 49.68

Figure 6.3: Architecture summary.

when analysing each single target: this finding is confirmed by the high AUC values. In general, sensitivity values are low because the data set is strongly unbalanced to reflect the true operational screening conditions.

Table 6.2 reports also the values of the Matthews Correlation Coefficient (MCC) which is a well known index used for binary classification that returns a value in $[-1; 1]$, and can be related to the chi-square statistic for a 2×2 contingency table, that is a binary classifier's confusion matrix. In particular, the relation with chi-square statistic is expressed by $\| \mathrm{MCC} \| = \sqrt{\chi^2/n}$ where n is the number of observations so it measures the dependency of the predictions from the true (i.e. expected) labels. The form of this indicator is related to the results reported in Table 6.3, which is devoted to explain the actual screening capabilities of my model, and contains both the Enrichment Factors (EF) and the True Positives versus Positives (TP/P) ratio for each target at different percentages.

EF after $x\%$ of the focused library were calculated according to the following formula

$$EF = \frac{N_{\text{experimental}}{}^{x\%}}{N_{\text{expected}}\, x^0} = \frac{N_{\text{experimental}}{}^{x\%}}{N_{\text{active}}\, \cdot x\%}$$

where $N_{experimental}$ is the number of experimentally found active

Table 6.2: Accuracy metrics for all the targets. Best/worst values for each column are in bold/italic

Target	Acc.	Loss	Sensitivity	MCC	AUC	F1-score
ACK	0.9957	0.0226	0.5000	0.6742	0.9834	0.6463
ALK	0.9930	0.0402	0.6575	0.7913	0.9904	0.7804
CDK1	0.9910	0.0314	0.4537	0.6397	0.9850	0.6059
CDK2	0.9859	0.0431	0.5281	0.6338	0.9845	0.6287
CDK6	0.9966	0.0210	0.5865	0.7523	0.9895	0.7305
INSR	0.9893	0.0329	0.3779	0.5830	0.9858	0.5342
ITK	0.9945	0.0232	0.5886	0.7302	0.9905	0.7154
JAK2	0.9898	0.0472	**0.8474**	**0.9090**	**0.9950**	**0.9114**
JNK3	**0.9967**	**0.0154**	0.5905	0.7610	0.9901	0.7381
MELK	0.9957	0.0229	0.7081	0.8270	0.9897	0.8188
CHK1	0.9895	0.0512	0.6385	0.7650	0.9846	0.7565
CK2A1	0.9942	0.0253	0.5166	0.6944	0.9857	0.6667
CLK2	0.9936	0.0259	*0.2255*	*0.4137*	0.9771	*0.3485*
DYRK1A	0.9916	0.0321	0.4080	0.5987	0.9776	0.5591
EGFR	0.9845	*0.0604*	0.7536	0.8331	0.9874	0.8357
ERK2	0.9881	0.0563	0.7295	0.8292	0.9886	0.8272
GSK3	*0.9843*	0.0554	0.5827	0.6892	*0.9762*	0.6856
IRAK4	0.9936	0.0287	0.7611	0.8611	0.9938	0.8571
MAP2K1	0.9931	0.0319	0.5497	0.7184	0.9795	0.6954
PDK1	0.9945	0.0271	0.6310	0.7757	0.9875	0.7613

structures in the top $x\%$ of the sorted database, $N_{expected}$ is the number of expected active structures, and N_{active} is total number of active structures in database[14] . The EF computes the number of predicted true actives, in decreasing probability order, in a fixed percentage of the test set. Typical percentages are 5% and 10% but in this study I tested also the performance at 1%. Such a measure is intended to provide the number of times a particular screening procedure performs better than a pure random process.

EF values reported in Table 6.3 are considerably high, and drop to 9 only at 10%. This result is truly remarkable even though no drug designer takes into account such large test set percentages. Moreover, all such values are considerably higher than the ones considered sufficient for a good model[14] .

I calculated TP/P parameter at different percentages of the test set as described in section 5.3. The use of the TP/P indicator explains some controversial EF values. The worst EF values (less

Table 6.3: True Positives versus Positives ratio and Enrichment Factors computed on the entire test set.

Protein	TP/P 1%*	TP/P 2%*	TP/P 5%*	TP/P 10%*	EF 1%	EF 2%	EF 5%	EF 10%
ACK	72/106	84/106	95/106	101/106	68	40	18	10
ALK	131/254	202/254	229/254	247/254	52	40	18	10
CDK1	111/205	150/205	189/205	196/205	54	37	18	10
CDK2	118/303	194/303	264/303	289/303	39	32	17	10
CDK6	79/104	90/104	98/104	101/104	76	43	19	10
INSR	110/217	145/217	195/217	206/217	51	33	18	9
ITK	107/158	125/158	148/158	155/158	68	40	19	10
JAK2	134/832	268/832	669/832	804/832	16	16	16	10
JNK3	81/105	88/105	95/105	102/105	77	42	18	10
MELK	130/185	157/185	178/185	181/185	70	42	19	10
CHK1	134/343	233/343	300/343	324/343	39	34	17	9
CK2A1	100/151	117/151	141/151	146/151	66	39	19	10
CLK2	59/102	73/102	87/102	96/102	58	36	17	9
DYRK1A	97/174	126/174	152/174	162/174	56	36	17	9
EGFR	134/702	268/702	586/702	664/702	19	19	17	9
ERK2	133/525	267/525	471/525	505/525	25	25	18	10
GSK3	132/393	226/393	327/393	353/393	34	29	17	9
IRAK4	134/339	263/339	320/339	333/339	40	39	19	10
MAP2K1	118/191	142/191	167/191	178/191	62	37	17	9
PDK1	123/187	149/187	170/187	181/187	66	40	18	10

* percentage relative to the evaluated test set evaluated (13400 compounds), i.e 1% = 134 molecules

than 20 at every percentage) are obtained for the JAK2 and EGFR targets respectively. This result comes from the high abundance of active molecules in the test set that are much higher than the the number of ligands considered at each percentage. In fact, the TP/P ratio reported in the same table confirms that the classifier correctly prioritizes as many active molecules as the considered test set percentage for both the target proteins.

Finally, MCC values in Table 6.2 are in line with TP/P values as it was expected due to the very similar form of these indicators. In fact, the highest MCC is obtained exactly for the JAK2 target.

Chapter 7

Other research activities

In this chapter I report two small activities done during my PhD period.

7.1 Drug repurposing application

During the PhD program, I collaborated with the CLAIRE (Confederation of Laboratories for Artificial Intelligence in Europe) task force on the covid-19. The task of our team was to use the molecular fingerprint approach in drug repurposing. However, the problem was not a simple one because, as is well known, there are no recognized drugs active against covid-19 infection. The work was mainly focused on the search for data because drug repurposing, as already mentioned, is based on the reuse of drugs already approved or in advanced trials. Since SARS-Cov-2 is a virus of new origin, the identification of drugs that act directly on proteins involved in the infection was not trivial.

At the beginning a set of 41 proteins, including viral and host proteins, directly involved in the infection has been identified, selected and used to carry out a bioinformatic investigation. Starting from the amino acid sequences of each of these proteins was performed a query on Blast (Basic Local Alignment Search Tool), a tool provided by NCBI (National Center of Biotechnology Information) for the search of sequence similarity. The algorithm used is blastp and the search was performed on the "non-redundant protein sequence (nr)" database for the organisms: i) human (taxid:9606) and ii) virus (taxid:10239) excluding SARS-CoV-2 (taxid:2697049).

Class	*Precision(%)*	*Recall(%)*	*F1-Score(%)*
0	85	94	89
1	35	33	34

Table 7.1: Average performance measures by class in the DSC network. Class 0 indicates inactive compounds, and class 1 indicates active compounds.

Results were filtered based on Identity values number of identical AAs shortest sequence length and the Expect value (E value) a parameter describing the number of random hits that can be found when searching for a hit within a sequence of variable length.

The data thus obtained were cross-referenced with the DrugBank database, succeeding in identifying 5 approved drugs and 19 experimental ones. To enlarge the pool of active compounds a further screening analysis using the protein interaction dataset provided by the CLAIRE Task Force for COVID19, in which the "human-computer interaction" laboratory is actively participating, was conducted. This cross-reference analysis allowed the identification of all protein interactors (positive or negative), expanding the number of approved and investigational drugs to 100 and 133, respectively. Starting with these active drugs, the final dataset includes 1153 compounds, with an active to inactive ratio of 1:4 (233 active and 920 inactive). Despite great effort to identify molecules active against covid-19 infection, I could not detect enough active molecules to perform a deep network training.

Class	*Precision(%)*	*Recall(%)*	*F1-Score(%)*
0	91	83	87
1	50	68	58

Table 7.2: Average performance measures by class in the CNN network. Class 0 indicates inactive compounds, and class 1 indicates active compounds.

Despite efforts to obtain an adequate dataset for training the networks, I was unable to obtain sufficient numbers to carry out the training satisfactorily. The tables 7.1 and 7.2 show the results obtained as a result of the training performed with the available data. An attempt has been made but, as I expected, it did not lead to good results. In in table 7.1 were reported DSC results and table 7.2 CNN results.

7.2 SHAP study

An explainability analysis has been performed to assess the most relevant features for the classification task, and the results of this analysis confirm some very recent *in vitro* studies that outline the relevance of pharmacophore-like description fingerprints when addressing bioactivity classification for kinase inhibitors. In order to accomplish my commitment to explain the role of each fingerprint in my embedding, I used the well-known SHAP framework to analyze my trained network. SHAP stands for *SHapley Additive exPlanations* [60] , and it is a game theoretic approach that was proposed first by Lipovetsky and Conklin [59] . In this work, the relevance of each predictor in a linear regression model is measured using the *Shapley Value (SV) imputation* that is a method to rank the importance of each player in a multi-player game over all the possible combinations of players. The authors use the *SHAP values* as the unique measure for feature relevance in an additive feature attribution explainability model, that is defined by a linear combination of the features to be explained z_i weighted by some importance factors ϕ_i. The SHAP value for a feature z_i is estimated as the SV ϕ_i of a conditional expectation function $E[f(z)|z_i]$ describing the expected prediction over the entire feature set z conditioned to z_i. Both model agnostic linear explanation and model specific computation of SHAP values is proposed.

In my case, I adopted the so called *Deep SHAP* explanation model that is suited for CNN because it combines SHAP values with the recursive relevance scores computation proposed in *DeepLIFT* [89] . The DeepLIFT explainability model assumes that a difference $\Delta t = t - t_0$ in an output neuron between the actual activation t and a reference one t_0 is related to the activation difference Δx_i in whatever contributing neuron by the *summation-to-delta* property $\sum_i C_{\Delta x_i \Delta t} = \Delta t$ that is a constraint on the relevance scores $C_{\Delta x_i \Delta t}$. Deep SHAP applies the DeepLIFT approach to the expectation function $E[f(z)|z_i]$ reference value.

The results of my analysis are reported in figure 7.1; on the left I reported the SHAP values for each target and for each fingerprint averaged on the entire test set separately for each target, while on the right the CDK1 only analysis is reported as an example of the

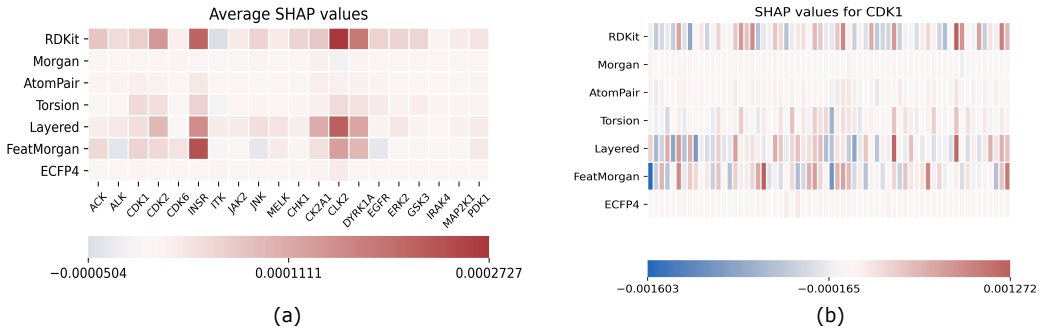


Figure 7.1: Explainability results using SHAP; (a) average SHAP values; (b) example of single target explainability analysis for CDK1

results obtained target by target. Here, each fingerprint has been grouped in 64 bins to enhance readability.
As I expected, SHAP values are arranged in a way that some fingerprints are relevant as a whole for predicting a target, while others have no contribution that is all the SHAP values are almost zero for each bit of the fingerprint. All the targets exhibit the same relevant fingerprints even if the actual SHAP values differ from each other.

FeatMorgan, Layered and RDKIT fingerprints, showed a major influence on the prediction results when compared to the others. I tried to rationalize this observation related to the fingerprint composition. In detail, FeatMorgan is a kind of FCFP circular fingerprint where the ligand is characterized by the functional description of atoms directly related to its binding capability (e.g. hydrogen donor/acceptor, polarity, aromaticity, and so on). Probably, for such a kind of classification, not merely based on the chemical path, but on the ligand capability to bind specific protein residues, such a kind of ligand description outperforms when compared to the simple ECFP circular fingerprint, only relative to atom type paths. RDKIT and Layered fingerprint are both based on substructure decomposition (e.g. aromatic rings). In a very recently published work by Zhu et al.[101] the authors ran a chemoinformatic analysis of 2139 Protein kinases inhibitors and found the majority of these molecules as "flat" with a very low fraction of sp^3 carbons and a high number of aromatic rings. From the study, it was also demonstrated that the average weighted hydrogen bond count was inversely proportional to the number of aromatic rings. In detail, it

seems like in the binding affinity to protein kinases, there is a correlated compensation between H-bond interactions and aromatic and non-bonded interactions. Such an inverse relationship strongly suggests the importance of the balanced presence of hydrogen bond donor and acceptors, and aromatic moieties within the ligand for the molecular recognition of Protein Kinases inhibitors.
In my opinion, the interpretation of the above-described interaction elements for kinase inhibitors, is better performed by the FCFP, RDKIT and Layered fingerprints compared to the other fingerprints mainly based on the mere description of chemical path, and not on the pharmacophoric role of the molecular elements.

Chapter 8

Conclusions

Summarizing, the work started with the identification of the group of fingerprints most suitable to be used simultaneously for classification using both machine learning and deep learning techniques, which brought common results. In the second step I tested the different lengths to get the right compromise between sparsity of the data (so high lengths) and bit collision (fingerprint rather short). After that I tested all the possible combinations of fingerprints to take into account all the different information coming from the various descriptors. Starting from the assumption that different fingerprints describe the same molecule as if they were different "spectra", because they contain the same information but collected in different ways along the molecular structure, two different approaches have been thought. In the first one I implemented a first architecture composed by 7 feature extractors placed in parallel and then merged in a single classifier that gave very good results. The second approach makes use of EMBER, an appropriate molecular embedding composed of 7 fingerprints arranged as the different channels of a one-dimensional tensor, which is used as input for a network that uses the operator Depthwise Separable Convolution to reduce computational complexity. The latter approach was initially tested on 10 target proteins, and given the excellent results I have expanded the number of targets to 20.

In conclusion, I presented a deep neural architecture for ligand multi classification as regards their bioactivity on twenty protein kinase targets. The innovation in my approach is the use of a neural embedding to represent the ligand's molecular structure made by several molecular fingerprints stacked as the channels of the

input tensor. The key idea behind this embedding is that molecular fingerprints are computed using the same algorithmic process, but using complementary information collected from the molecular structure so they can be regarded as the "spectra" of a sort of molecular image. I achieved very satisfactory results as regards the classification task, and in general I obtained a very high capacity model with a very small number of parameters.
Moreover, I presented an explainability analysis by feature attribution showing that just three molecular fingerprints play an active role in classification that are FeatMorgan, Layered and RDKIT. Our findings confirm very recent studies that outline the relevance of functional description Fingerprints (i.e. Pharmacophore-like) when addressing bioactivity classification, especially for kinase inhibitors.

From the results obtained with the explenability it is possible to see that the results reported in the table 4.3 have been confirmed. In fact my preliminary studies to identify the most performing fingerprint combinations have given as a result the same three fingerprints highlighted by SHAP (RDKit, Layered and FeatMorgan). This result was obtained with fingerprints of length 256 bits because the loss of chemical information was balanced by the computational efficiency of embedding.

Moreover, it is worth noting, how the model called Tuned-MLP-Out, which already obtained good results in terms of EF and TP/P, led us on the right track to obtain even better results with the use of fingerprints as different spectra of the same image. In the table 8.1 I show the comparison of the results obtained for CDK1 with the respective approaches, where I see the clear improvement.

Table 8.1: True Positives versus Positives ratio and Enrichment Factors computed on the entire test set.

Approach	TP/P 1%	TP/P 2%	TP/P 5%	TP/P 10%	EF 1%	EF 2%	EF 5%	EF 10%
Tuned-MLP-Out	7/83	14/83	35/83	69/83	8	8	8	8
DSC	111/205	150/205	189/205	196/205	54	37	18	10

The enrichment factor represents the number of times the model predicts better than a random model. As you can see the approach with DSC at 1% predicts more than 50 times better than a random model.

8.1 Interesting future challenges

Among the main goals I set for myself since the start of my PhD program, and which I believe I have achieved:

- to have deepened as much as possible the influence of each molecular fingerprint studied, on the prediction of biological activity and therefore on the accuracy of screening;
- to have designed a descriptor (EMBER) that can be understood also as an input layer, able to transmit relevant information in order to establish the bioactivity of a molecule on 20 protein targets;
- to have designed and implemented a multiclassifier to test EMBER with very good results.

Having said that, one of the future goals I'd like to achieve is the development of a bioactivity classifier for the complete kinase family. The main issues that will almost certainly arise are, first and foremost, data availability. Because of the way I structured the network of multiclassifiers, each molecule given as input must have bioactivity information on all targets chosen, which is understandable given the large number of proteins in the kinase family (about 500). To solve this difficulty, we could build a network with layers that can handle EMBER, as in the multiclassifier, and a descriptor that can precisely characterize a more or less tightly around the protein's pocket. Even here, the issues are diverse; among the most pressing are the development of a rigorous descriptor for the protein's pocket and the standardization of the data's size, because it is well known that working with fixed-size input is more convenient. Another critical point would be to investigate how an EMBER-like multispectral approach could apply to other vector representations.

Chapter 9

Acknowledgements

Acknowledgements are always the hardest part of the thesis to write. Mainly because you don't want to go overboard with the cheesiness and honey. In my case, however, it is a must, because during my PhD I was accompanied by a very patient tutor who spent a lot of time to make me reach goals that I would have never even imagined. So the first person I want to thank is definitely Prof. Pirrone, my "university father", who followed me from the first day, when he taught me how to turn on a PC, to the last day when I handed in my PhD thesis in computer engineering. He provided me a cultural background for which I will be always grateful. The second person I would like to say thank you to is my co-tutor Dr. Perricone who was always available for any clarification. He has always supported our hybrid work, half computer science and half chemoinformatics, with great ideas and great positivity, and thank to our teamwork has given great satisfaction. This last statement can only lead me to thank my companion of 'adventures' or as the prof says of 'mischief', Salvatore Contino, with whom I worked closely for almost all the PhD and with whom I shared all our successes and our disappointments. At the bottom of the list but not least is my family, the one that was there and the one that formed during my PhD for which I will never be grateful enough. Finally, I would like to thank the entire Computer – Human Interaction Lab for the good times we had together.

Bibliography

[1] CHeMBL Database. `https://www.ebi.ac.uk/chembl/`. Accessed: 24/09/2018.

[2] Daylight Chemical Information Systems. `https://www.daylight.com/`. Accessed: 24/01/2019.

[3] Molecular operating environment (moe).

[4] Pubchem substructure fingerprint.

[5] Unity 2d fingerprints.

[6] Lakshmi B. Akella and David DeCaprio. Cheminformatics approaches to analyze diversity in compound screening libraries. *Current Opinion in Chemical Biology*, 14(3):325–330, 2010.

[7] Syed M. Ali, Michael Z. Hoemann, Jeffrey Aubé, Gunda I. Georg, Lester A. Mitscher, and Lalith R. Jayasinghe. Butitaxel analogues: Synthesis and structureactivity relationships. *Journal of Medicinal Chemistry*, 40(2):236–241, Jan 1997.

[8] Christof Angermueller, Tanel Pärnamaa, Leopold Parts, and Oliver Stegle. Deep learning for computational biology. *Molecular Systems Biology*, 12(7):878, 2016. Publisher: John Wiley & Sons, Ltd.

[9] Syed Muhammad Anwar, Muhammad Majid, Adnan Qayyum, Muhammad Awais, Majdi Alnowami, and Muhammad Khurram Khan. Medical Image Analysis using Convolutional Neural Networks: A Review. *Journal of Medical Systems*, 42(11):226, October 2018.

[10] Meriem Bahi and Mohamed Batouche. Deep learning for ligand-based virtual screening in drug discovery. In *2018 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS)*, pages 1–5, 10 2018.

[11] J. M. Barnard, G. M. Downs, A. von Scholley-Pfab, and R. D. Brown. Use of markush structure analysis techniques for descriptor generation and clustering of large combinatorial libraries. *Journal of Molecular Graphics Modelling*, 18(4–5):452–463, Oct 2000.

[12] Subhash C. Basak, Brian D. Gute, and Denise Mills. Similarity methods in analog selection, property estimation and clustering of diverse chemicals. *Arkivoc*, 2006(9):157–210, Sep 2006.

[13] Igor I. Baskin, David Winkler, and Igor V. Tetko. A renaissance of neural networks in drug discovery. *Expert Opinion on Drug Discovery*, 11(8):785–795, 2016. PMID: 27295548.

[14] Andreas Bender and Robert C. Glen. A discussion of measures of enrichment in virtual screening: Comparing the information content of descriptors with increasing levels of sophistication. *Journal of Chemical Information and Modeling*, 45(5):1369–1375, Sep 2005.

[15] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242, January 2000.

[16] Michael R. Berthold, Nicolas Cebron, Fabian Dill, Thomas R. Gabriel, Tobias Kötter, Thorsten Meinl, Peter Ohl, Kilian Thiel, and Bernd Wiswedel. Knime - the konstanz information miner: Version 2.0 and beyond. *SIGKDD Explor. Newsl.*, 11(1):26–31, November 2009.

[17] I. Bighelli and C. Barbui. What is the european medicines agency? *Epidemiology and Psychiatric Sciences*, 21(3):245–247, Sep 2012.

[18] Esben Jannik Bjerrum. SMILES Enumeration as Data Augmentation for Neural Network Modeling of Molecules. *arXiv:1703.07076 [cs]*, May 2017. arXiv: 1703.07076.

[19] Stephen K. Burley, Charmi Bhikadiya, Chunxiao Bi, Sebastian Bittrich, Li Chen, Gregg V. Crichlow, Cole H. Christie, Kenneth Dalenberg, Luigi Di Costanzo, Jose M. Duarte, Shuchismita Dutta, Zukang Feng, Sai Ganesan, David S. Goodsell, Sutapa Ghosh, Rachel Kramer Green, Vladimir Guranović, Dmytro Guzenko, Brian P. Hudson, Catherine L. Lawson, Yuhe Liang, Robert Lowe, Harry Namkoong, Ezra Peisach, Irina Persikova, Chris Randle, Alexander Rose, Yana Rose, Andrej Sali, Joan Segura, Monica Sekharan, Chenghua Shao, Yi-Ping Tao, Maria Voigt, John D. Westbrook, Jasmine Y. Young, Christine Zardecki, and Marina Zhuravleva. RCSB Protein Data Bank: powerful new tools for exploring 3D structures of biological macromolecules for basic and applied research and education in fundamental biology, biomedicine, biotechnology, bioengineering and energy sciences. *Nucleic Acids Research*, 49(D1):D437–D451, January 2021.

[20] Raymond E. Carhart, Dennis H. Smith, and R. Venkataraghavan. Atom pairs as molecular features in structure-activity studies: definition and applications. *Journal of Chemical Information and Computer Sciences*, 25(2):64–73, May 1985.

[21] Adrià Cereto-Massagué, María José Ojeda, Cristina Valls, Miquel Mulero, Santiago Garcia-Vallvé, and Gerard Pujadas". Molecular fingerprint similarity search in virtual screening. *Methods*, 71:58 – 63, 2015. Virtual Screening.

[22] Adrià Cereto-Massagué, María José Ojeda, Cristina Valls, Miquel Mulero, Santiago Garcia-Vallvé, and Gerard Pujadas. Molecular fingerprint similarity search in virtual screening. *Methods*, 71:58–63, Jan 2015.

[23] Lieyang Chen, Anthony Cruz, Steven Ramsey, Callum J. Dickson, Jose S. Duca, Viktor Hornak, David R. Koes, and Tom Kurtzman. Hidden bias in the DUD-E dataset leads to

misleading performance of deep learning in structure-based virtual screening. *PLOS ONE*, 14(8):e0220113, August 2019.

[24] François Chollet. Xception: Deep Learning with Depthwise Separable Convolutions. *arXiv:1610.02357 [cs]*, April 2017. arXiv: 1610.02357.

[25] Jonas Cicenas, Linas Tamosaitis, Kotryna Kvederaviciute, Ricardas Tarvydas, Gintare Staniute, Karthik Kalyan, Edita Meskinyte-Kausiliene, Vaidotas Stankevicius, and Mindaugas Valius. KRAS, NRAS and BRAF mutations in colorectal cancer and melanoma. *Medical Oncology (Northwood, London, England)*, 34(2):26, 2017-02.

[26] George E. Dahl, Navdeep Jaitly, and Ruslan Salakhutdinov. Multi-task Neural Networks for QSAR Predictions. *arXiv:1406.1231 [cs, stat]*, June 2014. arXiv: 1406.1231.

[27] Laurianne David, Josep Arús-Pous, Johan Karlsson, Ola Engkvist, Esben Jannik Bjerrum, Thierry Kogej, Jan M. Kriegl, Bernd Beck, and Hongming Chen. Applications of deep-learning in exploiting large-scale and heterogeneous compound data in industrial pharmaceutical research. *Frontiers in Pharmacology*, 10:1303, Nov 2019.

[28] Laurianne David, Amol Thakkar, Rocío Mercado, and Ola Engkvist. Molecular representations in ai-driven drug discovery: a review and practical guide. *Journal of Cheminformatics*, 12(1):56, Sep 2020.

[29] Zhan Deng, Claudio Chuaqui, and Juswinder Singh. Structural interaction fingerprint (sift): a novel method for analyzing three-dimensional proteinligand binding interactions. *Journal of Medicinal Chemistry*, 47(2):337–344, Jan 2004.

[30] Andreas Dietz. Yet another representation of molecular structure. *Journal of Chemical Information and Computer Sciences*, 35(5):787–802, Sep 1995.

[31] Joseph DiMasi, Ronald Hansen, and Henry Grabowski. The price of innovation: New estimates of drug development costs. *Journal of health economics*, 22:151–85, 2003-04-01.

[32] M Kasim Diril, Chandrahas Koumar Ratnacaram, VC Padmakumar, Tiehua Du, Martin Wasser, Vincenzo Coppola, Lino Tessarollo, and Philipp Kaldis. Cyclin-dependent kinase 1 (cdk1) is essential for cell division and suppression of dna re-replication but not for liver regeneration. *Proceedings of the National Academy of Sciences*, 109(10):3826–3831, 2012.

[33] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015.

[34] Fahimeh Ghasemi, Alireza Mehridehnavi, Alfonso Pérez-Garrido, and Horacio Pérez-Sánchez. Neural network and deep-learning algorithms used in QSAR studies: merits and drawbacks. *Drug Discovery Today*, 23(10):1784–1790, October 2018.

[35] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural Message Passing for Quantum Chemistry. *arXiv:1704.01212 [cs]*, June 2017. arXiv: 1704.01212.

[36] Garrett B. Goh, Charles Siegel, Abhinav Vishnu, Nathan O. Hodas, and Nathan Baker. Chemception: A deep neural network with minimal chemistry knowledge matches the performance of expert-developed qsar/qspr models. *arXiv:1706.06689 [cs, stat]*, Jun 2017. arXiv: 1706.06689.

[37] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge, MA, USA, 2016. `http://www.deeplearningbook.org`.

[38] Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, Feb 2018.

[39] Adel Hamza, Ning-Ning Wei, and Chang-Guo Zhan. Ligand-based virtual screening approach using a new scoring function. *Journal of chemical information and modeling*, 52(4):963–974, Apr 2012.

[40] Jérôme Hert, Peter Willett, David J. Wilton, Pierre Acklin, Kamal Azzaoui, Edgar Jacoby, and Ansgar Schuffenhauer. New methods for ligand-based virtual screening: use of data fusion and machine learning to enhance the effectiveness of similarity searching. *Journal of Chemical Information and Modeling*, 46(2):462–470, 2006. PMID: 16562973.

[41] Maya Hirohara, Yutaka Saito, Yuki Koda, Kengo Sato, and Yasubumi Sakakibara. Convolutional neural network based on smiles representation of compounds for detecting chemical motif. *BMC Bioinformatics*, 19(19):526, Dec 2018.

[42] Tao Huang, Hong Mi, Cheng-yuan Lin, Ling Zhao, Linda L. D. Zhong, Feng-bin Liu, Ge Zhang, Ai-ping Lu, Zhao-xiang Bian, Shu-hai Lin, Man Zhang, Yan-hong Li, Dong-dong Hu, Chung-Wah Cheng, and for MZRW Group. Most: most-similar ligand based approach to target prediction. *BMC Bioinformatics*, 18(1):165, Mar 2017.

[43] Che-Lun Hung and Chi-Chun Chen. Computational approaches for drug discovery. *Drug Development Research*, 75(6):412–418, Sep 2014.

[44] A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4):411–430, 2000.

[45] Katsuhiko Ishiguro, Kenta Oono, and Kohei Hayashi. Weisfeiler-Lehman Embedding for Molecular Graph Neural Networks. *arXiv:2006.06909 [cs, stat]*, August 2020. arXiv: 2006.06909.

[46] Daniel Jimenez-Carretero, Vahid Abrishami, Laura Fernández-de Manuel, Irene Palacios, Antonio Quílez-Álvarez, Alberto Díez-Sánchez, Miguel A. del Pozo, and María C. Montoya. Tox(r)cnn: Deep learning-based nuclei

profiling tool for drug toxicity screening. *PLOS Computational Biology*, 14(11):e1006238, Nov 2018.

[47] Yankang Jing, Yuemin Bian, Ziheng Hu, Lirong Wang, and Xiang-Qun Sean" Xie. Deep learning for drug design: an artificial intelligence paradigm for drug discovery in the big data era. *The AAPS Journal*, 2018.

[48] Douglas R. Henry Joseph L. Durant, Burton A. Leland and James G. Nourse. Reoptimization of mdl keys for use in drug discovery. *J. Chem. Inf. Comput. Sci.*, 42:1273–1280, Nov 2002.

[49] Mohammad A. Khanfar and Mutasem O. Taha. Elaborate ligand-based modeling coupled with multiple linear regression and k nearest neighbor qsar analyses unveiled new nanomolar mtor inhibitors. *Journal of Chemical Information and Modeling*, 53(10):2587–2612, 2013. PMID: 24050502.

[50] Sunghwan Kim, Paul A. Thiessen, Evan E. Bolton, Jie Chen, Gang Fu, Asta Gindulyte, Lianyi Han, Jane He, Siqian He, Benjamin A. Shoemaker, Jiyao Wang, Bo Yu, Jian Zhang, and Stephen H. Bryant. Pubchem substance and compound databases. *Nucleic Acids Research*, 44(D1):D1202–D1213, Jan 2016.

[51] Talia B. Kimber, Yonghui Chen, and Andrea Volkamer. Deep learning in virtual screening: Recent applications and developments. *International Journal of Molecular Sciences*, 22(9), 2021.

[52] Daiki Koge, Naoaki Ono, Ming Huang, Md Altaf-Ul-Amin, and Shigehiko Kanaya. Embedding of Molecular Structure Using Molecular Hypergraph Variational Autoencoder with Metric Learning. *Molecular Informatics*, 40(2):2000203, 2021. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/minf.202000203.

[53] Albert J. Kooistra, Georgi K. Kanev, Oscar P.J. van Linden, Rob Leurs, Iwan J.P. de Esch, and Chris de Graaf. Klifs: a

structural kinase-ligand interaction database. *Nucleic Acids Research*, 44(D1):D365–D371, 2015.

[54] Justin Lamb, Emily D. Crawford, David Peck, Joshua W. Modell, Irene C. Blat, Matthew J. Wrobel, Jim Lerner, Jean-Philippe Brunet, Aravind Subramanian, Kenneth N. Ross, Michael Reich, Haley Hieronymus, Guo Wei, Scott A. Armstrong, Stephen J. Haggarty, Paul A. Clemons, Ru Wei, Steven A. Carr, Eric S. Lander, and Todd R. Golub. The connectivity map: Using gene-expression signatures to connect small molecules, genes, and disease. *Science*, Sep 2006.

[55] Eric S. Lander, Lauren M. Linton, Bruce Birren, Chad Nusbaum, Michael C. Zody, Jennifer Baldwin, Keri Devon, Ken Dewar, Michael Doyle, William FitzHugh, Roel Funke, Diane Gage, Katrina Harris, Andrew Heaford, John Howland, Lisa Kann, Jessica Lehoczky, Rosie LeVine, Paul McEwan, Kevin McKernan, James Meldrim, Jill P. Mesirov, Cher Miranda, William Morris, Jerome Naylor, Christina Raymond, Mark Rosetti, Ralph Santos, Andrew Sheridan, Carrie Sougnez, Nicole Stange-Thomann, Nikola Stojanovic, Aravind Subramanian, Dudley Wyman, Jane Rogers, John Sulston, Rachael Ainscough, Stephan Beck, David Bentley, John Burton, Christopher Clee, Nigel Carter, Alan Coulson, Rebecca Deadman, Panos Deloukas, Andrew Dunham, Ian Dunham, Richard Durbin, Lisa French, Darren Grafham, Simon Gregory, Tim Hubbard, Sean Humphray, Adrienne Hunt, Matthew Jones, Christine Lloyd, Amanda McMurray, Lucy Matthews, Simon Mercer, Sarah Milne, James C. Mullikin, Andrew Mungall, Robert Plumb, Mark Ross, Ratna Shownkeen, Sarah Sims, Robert H. Waterston, Richard K. Wilson, LaDeana W. Hillier, John D. McPherson, Marco A. Marra, Elaine R. Mardis, Lucinda A. Fulton, Asif T. Chinwalla, Kymberlie H. Pepin, Warren R. Gish, Stephanie L. Chissoe, Michael C. Wendl, Kim D. Delehaunty, Tracie L. Miner, Andrew Delehaunty, Jason B. Kramer, Lisa L. Cook, Robert S. Fulton, Douglas L. Johnson, Patrick J. Minx, Sandra W. Clifton, Trevor Hawkins, Elbert Branscomb, Paul Predki,

Paul Richardson, Sarah Wenning, Tom Slezak, Norman Doggett, Jan-Fang Cheng, Anne Olsen, Susan Lucas, Christopher Elkin, Edward Uberbacher, Marvin Frazier, Richard A. Gibbs, Donna M. Muzny, Steven E. Scherer, John B. Bouck, Erica J. Sodergren, Kim C. Worley, Catherine M. Rives, James H. Gorrell, Michael L. Metzker, Susan L. Naylor, Raju S. Kucherlapati, David L. Nelson, George M. Weinstock, Yoshiyuki Sakaki, Asao Fujiyama, Masahira Hattori, Tetsushi Yada, Atsushi Toyoda, Takehiko Itoh, Chiharu Kawagoe, Hidemi Watanabe, Yasushi Totoki, Todd Taylor, Jean Weissenbach, Roland Heilig, William Saurin, Francois Artiguenave, Philippe Brottier, Thomas Bruls, Eric Pelletier, Catherine Robert, Patrick Wincker, André Rosenthal, Matthias Platzer, Gerald Nyakatura, Stefan Taudien, Andreas Rump, Douglas R. Smith, Lynn Doucette-Stamm, Marc Rubenfield, Keith Weinstock, Hong Mei Lee, JoAnn Dubois, Huanming Yang, Jun Yu, Jian Wang, Guyang Huang, Jun Gu, Leroy Hood, Lee Rowen, Anup Madan, Shizen Qin, Ronald W. Davis, Nancy A. Federspiel, A. Pia Abola, Michael J. Proctor, Bruce A. Roe, Feng Chen, Huaqin Pan, Juliane Ramser, Hans Lehrach, Richard Reinhardt, W. Richard McCombie, Melissa de la Bastide, Neilay Dedhia, Helmut Blöcker, Klaus Hornischer, Gabriele Nordsiek, Richa Agarwala, L. Aravind, Jeffrey A. Bailey, Alex Bateman, Serafim Batzoglou, Ewan Birney, Peer Bork, Daniel G. Brown, Christopher B. Burge, Lorenzo Cerutti, Hsiu-Chuan Chen, Deanna Church, Michele Clamp, Richard R. Copley, Tobias Doerks, Sean R. Eddy, Evan E. Eichler, Terrence S. Furey, James Galagan, James G. R. Gilbert, Cyrus Harmon, Yoshihide Hayashizaki, David Haussler, Henning Hermjakob, Karsten Hokamp, Wonhee Jang, L. Steven Johnson, Thomas A. Jones, Simon Kasif, Arek Kaspryzk, Scot Kennedy, W. James Kent, Paul Kitts, Eugene V. Koonin, Ian Korf, David Kulp, Doron Lancet, Todd M. Lowe, Aoife McLysaght, Tarjei Mikkelsen, John V. Moran, Nicola Mulder, Victor J. Pollara, Chris P. Ponting, Greg Schuler, Jörg Schultz, Guy Slater, Arian F. A. Smit, Elia Stupka, Joseph Szustakowki, Danielle Thierry-Mieg,

Jean Thierry-Mieg, Lukas Wagner, John Wallis, Raymond Wheeler, Alan Williams, Yuri I. Wolf, Kenneth H. Wolfe, Shiaw-Pyng Yang, Ru-Fang Yeh, Francis Collins, Mark S. Guyer, Jane Peterson, Adam Felsenfeld, Kris A. Wetterstrand, Richard M. Myers, Jeremy Schmutz, Mark Dickson, Jane Grimwood, David R. Cox, Maynard V. Olson, Rajinder Kaul, Christopher Raymond, Nobuyoshi Shimizu, Kazuhiko Kawasaki, Shinsei Minoshima, Glen A. Evans, Maria Athanasiou, Roger Schultz, Aristides Patrinos, Michael J. Morgan, International Human Genome Sequencing Consortium, Center for Genome Research: Whitehead Institute for Biomedical Research, The Sanger Centre:, Washington University Genome Sequencing Center, US DOE Joint Genome Institute:, Baylor College of Medicine Human Genome Sequencing Center:, RIKEN Genomic Sciences Center:, Genoscope, CNRS UMR-8030:, Institute of Molecular Biotechnology: Department of Genome Analysis, GTC Sequencing Center:, Beijing Genomics Institute/Human Genome Center:, The Institute for Systems Biology: Multimegabase Sequencing Center, Stanford Genome Technology Center:, University of Oklahoma's Advanced Center for Genome Technology:, Max Planck Institute for Molecular Genetics:, Lita Annenberg Hazen Genome Center: Cold Spring Harbor Laboratory, GBF—German Research Centre for Biotechnology:, also includes individuals listed under other headings): *Genome Analysis Group (listed in alphabetical order, US National Institutes of Health: Scientific management: National Human Genome Research Institute, Stanford Human Genome Center:, University of Washington Genome Center:, Keio University School of Medicine: Department of Molecular Biology, University of Texas Southwestern Medical Center at Dallas:, US Department of Energy: Office of Science, and The Wellcome Trust:. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921, Feb 2001. Bandiera$_a$*btest* : $aCg_t type$: $NatureResearchJournalsnumber$: $6822Primary_a type$: $Researchpublisher$: $NaturePublishingGroup$.

[56] Jiazhong Li, Huanxiang Liu, Xiaojun Yao, Mancang Liu, Zhide Hu, and Botao Fan. Structure–activity relationship study of oxindole-based inhibitors of cyclin-dependent kinases based on least-squares support vector machines. *Analytica Chimica Acta*, 581(2):333–342, January 2007.

[57] Shuhui Lim and Philipp Kaldis. Cdks, cyclins and ckis: roles beyond cell cycle regulation. *Development*, 140(15):3079–3093, Aug 2013.

[58] Ting-Wan Lin, Melrose M. Melgar, Daniel Kurth, S. Joshua Swamidass, John Purdon, Teresa Tseng, Gabriela Gago, Pierre Baldi, Hugo Gramajo, and Shiou-Chuan Tsai. Structure-based inhibitor design of accd5, an essential acyl-coa carboxylase carboxyltransferase domain of mycobacterium tuberculosis. *Proceedings of the National Academy of Sciences*, 103(9):3072–3077, 2006.

[59] Stan Lipovetsky and Michael Conklin. Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*, 17(4):319–330, 2001. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/asmb.446.

[60] Scott M. Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems*, 30, 2017.

[61] John David MacCuish and Norah Elizabeth MacCuish. Chemoinformatics applications of cluster analysis. *WIREs Computational Molecular Science*, 4(1):34–48, 2014.

[62] Marcos Malumbres and Mariano Barbacid. Mammalian cyclin-dependent kinases. *Trends in Biochemical Sciences*, 30(11):630–641, Nov 2005.

[63] Keith A. Marill. Advanced statistics: Linear regression, part ii: Multiple linear regression. *Academic Emergency Medicine*, 11(1):94–102, 2004.

[64] Eric Martin, Prasenjit Mukherjee, David Sullivan, and Johanna Jansen. Profile-QSAR: A Novel *meta* -QSAR Method that Combines Activities across the Kinase Family To Accurately Predict

Affinity, Selectivity, and Cellular Activity. *Journal of Chemical Information and Modeling*, 51(8):1942–1956, August 2011.

[65] Jonathan S. Mason and Daniel L. Cheney. Library design and virtual screening using multiple 4-point pharmacophore fingerprints. In *Biocomputing 2000*, page 576–587. WORLD SCIENTIFIC, Dec 1999.

[66] Malcolm J. McGregor and Steven M. Muskal. Pharmacophore fingerprinting. 1. application to qsar and focused library design. *Journal of Chemical Information and Computer Sciences*, 39(3):569–574, May 1999.

[67] Malcolm J. McGregor and Steven M. Muskal. Pharmacophore fingerprinting. 2. application to primary library design. *Journal of Chemical Information and Computer Sciences*, 40(1):117–125, Jan 2000.

[68] David Mendez, Anna Gaulton, A. Patrícia Bento, Jon Chambers, Marleen De Veij, Eloy Félix, María Paula Magariños, Juan F. Mosquera, Prudence Mutowo, Michal Nowotka, María Gordillo-Marañón, Fiona Hunter, Laura Junco, Grace Mugumbate, Milagros Rodriguez-Lopez, Francis Atkinson, Nicolas Bosc, Chris J. Radoux, Aldo Segura-Cabrera, Anne Hersey, and Andrew R. Leach. ChEMBL: towards direct deposition of bioassay data. *Nucleic Acids Research*, 47(D1):D930–D940, January 2019.

[69] Isabella Mendolia, Salvatore Contino, Giada De Simone, Ugo Perricone, and Roberto Pirrone. Ember—embedding multiple molecular fingerprints for virtual screening. *International Journal of Molecular Sciences*, 23(4):2156, Feb 2022.

[70] Isabella Mendolia, Salvatore Contino, Ugo Perricone, Edoardo Ardizzone, and Roberto Pirrone. Convolutional architectures for virtual screening. *BMC Bioinform.*, 21-S(8):310, 2020.

[71] Benjamin Merget, Samo Turk, Sameh Eid, Friedrich Rippmann, and Simone Fulle. Profiling Prediction of Kinase Inhibitors: Toward the Virtual Assay. *Journal of Medicinal Chemistry*, 60(1):474–485, January 2017.

[72] Ingo Muegge and Prasenjit Mukherjee. An overview of molecular fingerprint similarity search in virtual screening. *Expert Opinion on Drug Discovery*, 11(2):137–148, 2016.

[73] Orazio Nicolotti, Teresa Fabiola Miscioscia, Andrea Carotti, Francesco Leonetti, and Angelo Carotti. An integrated approach to ligand- and structure-based drug design: development and application to a series of serine protease inhibitors. *Journal of Chemical Information and Modeling*, 48(6):1211–1226, Jun 2008.

[74] Nicola Nosengo. Can you teach old drugs new tricks? *Nature*, 534(7607):314–316, Jun 2016. $\text{Bandiera}_a btest : aCg_t ype : NatureResearchJournalsnumber : 7607Primary_a type : Newspublisher : NaturePublishingGroupSubject_t erm : Drugdiscovery; Healthcare; TherapeuticsSubject_t erm_i d : drug - discovery; health - care; therapeutics.$

[75] Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *Journal of Cheminformatics*, 9(1):48, Sep 2017.

[76] George Papadatos, Anna Gaulton, Anne Hersey, and John P. Overington. Activity, assay and target data curation and quality in the chembl database. *Journal of Computer-Aided Molecular Design*, 29(9):885–896, Sep 2015.

[77] Debleena Paul, Gaurav Sanap, Snehal Shenoy, Dnyaneshwar Kalyane, Kiran Kalia, and Rakesh K. Tekade. Artificial intelligence in drug discovery and development. *Drug Discovery Today*, 26(1):80–93, Jan 2021.

[78] Janaina Cruz Pereira, Ernesto Raúl Caffarena, and Cicero Nogueira dos Santos. Boosting Docking-Based Virtual Screening with Deep Learning. *Journal of Chemical Information and Modeling*, 56(12):2495–2506, December 2016.

[79] Sudeep Pushpakom, Francesco Iorio, Patrick A. Eyers, K. Jane Escott, Shirley Hopper, Andrew Wells, Andrew Doig, Tim Guilliams, Joanna Latimer, Christine McNamee, and et al. Drug repurposing: progress, challenges and recommendations. *Nature Reviews. Drug Discovery*, 18(1):41–58, Jan 2019.

[80] Sudeep Pushpakom, Francesco Iorio, Patrick A. Eyers, K. Jane Escott, Shirley Hopper, Andrew Wells, Andrew Doig, Tim Guilliams, Joanna Latimer, Christine McNamee, Alan Norris, Philippe Sanseau, David Cavalla, and Munir Pirmohamed. Drug repurposing: progress, challenges and recommendations. *Nature Reviews Drug Discovery*, 18(1):41–58, Jan 2019.

[81] Bharath Ramsundar, Steven Kearnes, Patrick Riley, Dale Webster, David Konerding, and Vijay Pande. Massively multitask networks for drug discovery, 2015.

[82] Sereina Riniker and Gregory A Landrum. Open-source platform to benchmark fingerprints for ligand-based virtual screening. *Journal of Cheminformatics*, 5(1):26, Dec 2013.

[83] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, May 2010.

[84] G. Madhavi Sastry, V. S. Sandeep Inakollu, and Woody Sherman. Boosting Virtual Screening Enrichments with Data Fusion: Coalescing Hits from Two-Dimensional Fingerprints, Shape, and Docking. *Journal of Chemical Information and Modeling*, 53(7):1531–1542, July 2013.

[85] Gisbert Schneider. Mind and machine in drug design. *Nature Machine Intelligence*, 1(33):128–130, Mar 2019.

[86] Gisbert Schneider, Werner Neidhart, Thomas Giller, and Gerard Schmid. "scaffold-hopping" by topological pharmacophore search: A contribution to virtual screening. *Angewandte Chemie International Edition*, 38(19):2894–2896, Oct 1999.

[87] Julian Schwartz, Mahendra Awale, and Jean-Louis Reymond. Smifp (smiles fingerprint) chemical space for virtual screening and visualization of large databases of organic molecules. *Journal of Chemical Information and Modeling*, 53(8):1979–1989, Aug 2013.

[88] Marwin H. S. Segler, Thierry Kogej, Christian Tyrchan, and Mark P. Waller. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Central Science*, 4(1):120–131, 2018.

[89] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning Important Features Through Propagating Activation Differences. *arXiv:1704.02685 [cs]*, October 2019. arXiv: 1704.02685.

[90] Vladimir Svetnik, Andy Liaw, Christopher Tong, J. Christopher Culberson, Robert P. Sheridan, and Bradley P. Feuston. Random forest: a classification and regression tool for compound classification and qsar modeling. *Journal of Chemical Information and Computer Sciences*, 43(6):1947–1958, 2003. PMID: 14632445.

[91] Dominique Sydow, Lindsey Burggraaff, Angelika Szengel, Herman W. T. van Vlijmen, Adriaan P. IJzerman, Gerard J. P. van Westen, and Andrea Volkamer. Advances and challenges in computational target prediction. *Journal of Chemical Information and Modeling*, 59(5):1728–1742, 2019.

[92] Wen Torng and Russ B. Altman. Graph Convolutional Neural Networks for Predicting Drug-Target Interactions. *Journal of Chemical Information and Modeling*, 59(10):4131–4149, October 2019. Publisher: American Chemical Society.

[93] J. Rick Turner. *New Drug Development: An Introduction to Clinical Trials: Second Edition*. Springer-Verlag, 2 edition, 2010.

[94] Alexandre Varnek and Igor Baskin. Machine Learning Methods for Property Prediction in Chemoinformatics: *Quo Vadis* ? *Journal of Chemical Information and Modeling*, 52(6):1413–1437, June 2012.

[95] J. Craig Venter, Mark D. Adams, Eugene W. Myers, Peter W. Li, Richard J. Mural, Granger G. Sutton, Hamilton O. Smith, Mark Yandell, Cheryl A. Evans, Robert A. Holt, Jeannine D. Gocayne, Peter Amanatides, Richard M. Ballew, Daniel H. Huson, Jennifer Russo Wortman, Qing Zhang, Chinnappa D. Kodira, Xiangqun H. Zheng, Lin Chen, Marian Skupski, Gangadharan Subramanian, Paul D. Thomas, Jinghui Zhang, George L. Gabor Miklos, Catherine Nelson, Samuel Broder, Andrew G. Clark, Joe Nadeau, Victor A. McKusick, Norton Zinder, Arnold J. Levine, Richard J. Roberts, Mel Simon, Carolyn Slayman, Michael Hunkapiller, Randall Bolanos, Arthur Delcher, Ian Dew, Daniel Fasulo, Michael Flanigan, Liliana Florea, Aaron Halpern, Sridhar Hannenhalli,

Saul Kravitz, Samuel Levy, Clark Mobarry, Knut Reinert, Karin Remington, Jane Abu-Threideh, Ellen Beasley, Kendra Biddick, Vivien Bonazzi, Rhonda Brandon, Michele Cargill, Ishwar Chandramouliswaran, Rosane Charlab, Kabir Chaturvedi, Zuoming Deng, Valentina Di Francesco, Patrick Dunn, Karen Eilbeck, Carlos Evangelista, Andrei E. Gabrielian, Weiniu Gan, Wangmao Ge, Fangcheng Gong, Zhiping Gu, Ping Guan, Thomas J. Heiman, Maureen E. Higgins, Rui-Ru Ji, Zhaoxi Ke, Karen A. Ketchum, Zhongwu Lai, Yiding Lei, Zhenya Li, Jiayin Li, Yong Liang, Xiaoying Lin, Fu Lu, Gennady V. Merkulov, Natalia Milshina, Helen M. Moore, Ashwinikumar K Naik, Vaibhav A. Narayan, Beena Neelam, Deborah Nusskern, Douglas B. Rusch, Steven Salzberg, Wei Shao, Bixiong Shue, Jingtao Sun, Zhen Yuan Wang, Aihui Wang, Xin Wang, Jian Wang, Ming-Hui Wei, Ron Wides, Chunlin Xiao, Chunhua Yan, Alison Yao, Jane Ye, Ming Zhan, Weiqing Zhang, Hongyu Zhang, Qi Zhao, Liansheng Zheng, Fei Zhong, Wenyan Zhong, Shiaoping C. Zhu, Shaying Zhao, Dennis Gilbert, Suzanna Baumhueter, Gene Spier, Christine Carter, Anibal Cravchik, Trevor Woodage, Feroze Ali, Huijin An, Aderonke Awe, Danita Baldwin, Holly Baden, Mary Barnstead, Ian Barrow, Karen Beeson, Dana Busam, Amy Carver, Angela Center, Ming Lai Cheng, Liz Curry, Steve Danaher, Lionel Davenport, Raymond Desilets, Susanne Dietz, Kristina Dodson, Lisa Doup, Steven Ferriera, Neha Garg, Andres Gluecksmann, Brit Hart, Jason Haynes, Charles Haynes, Cheryl Heiner, Suzanne Hladun, Damon Hostin, Jarrett Houck, Timothy Howland, Chinyere Ibegwam, Jeffery Johnson, Francis Kalush, Lesley Kline, Shashi Koduru, Amy Love, Felecia Mann, David May, Steven McCawley, Tina McIntosh, Ivy McMullen, Mee Moy, Linda Moy, Brian Murphy, Keith Nelson, Cynthia Pfannkoch, Eric Pratts, Vinita Puri, Hina Qureshi, Matthew Reardon, Robert Rodriguez, Yu-Hui Rogers, Deanna Romblad, Bob Ruhfel, Richard Scott, Cynthia Sitter, Michelle Smallwood, Erin Stewart, Renee Strong, Ellen Suh, Reginald Thomas, Ni Ni Tint, Sukyee Tse, Claire Vech, Gary Wang, Jeremy Wetter, Sherita Williams, Monica Williams, Sandra Windsor, Emily Winn-Deen, Keriellen Wolfe, Jayshree Zaveri, Karena Zaveri, Josep F. Abril, Roderic Guigó, Michael J. Campbell, Kimmen V. Sjolander, Brian

Karlak, Anish Kejariwal, Huaiyu Mi, Betty Lazareva, Thomas Hatton, Apurva Narechania, Karen Diemer, Anushya Muruganujan, Nan Guo, Shinji Sato, Vineet Bafna, Sorin Istrail, Ross Lippert, Russell Schwartz, Brian Walenz, Shibu Yooseph, David Allen, Anand Basu, James Baxendale, Louis Blick, Marcelo Caminha, John Carnes-Stine, Parris Caulk, Yen-Hui Chiang, My Coyne, Carl Dahlke, Anne Deslattes Mays, Maria Dombroski, Michael Donnelly, Dale Ely, Shiva Esparham, Carl Fosler, Harold Gire, Stephen Glanowski, Kenneth Glasser, Anna Glodek, Mark Gorokhov, Ken Graham, Barry Gropman, Michael Harris, Jeremy Heil, Scott Henderson, Jeffrey Hoover, Donald Jennings, Catherine Jordan, James Jordan, John Kasha, Leonid Kagan, Cheryl Kraft, Alexander Levitsky, Mark Lewis, Xiangjun Liu, John Lopez, Daniel Ma, William Majoros, Joe McDaniel, Sean Murphy, Matthew Newman, Trung Nguyen, Ngoc Nguyen, Marc Nodell, Sue Pan, Jim Peck, Marshall Peterson, William Rowe, Robert Sanders, John Scott, Michael Simpson, Thomas Smith, Arlan Sprague, Timothy Stockwell, Russell Turner, Eli Venter, Mei Wang, Meiyuan Wen, David Wu, Mitchell Wu, Ashley Xia, Ali Zandieh, and Xiaohong Zhu. The sequence of the human genome. *Science*, 291(5507):1304–1351, Feb 2001.

[96] Izhar Wallach, Michael Dzamba, and Abraham Heifets. Atomnet: A deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *CoRR*, abs/1510.02855, 2015.

[97] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 1988.

[98] Jincai Yang, Cheng Shen, and Niu Huang. Predicting or Pretending: Artificial Intelligence for Protein-Ligand Interactions Lack of Sufficiently Large and Unbiased Datasets. *Frontiers in Pharmacology*, 11:69, February 2020.

[99] Kun Yao and John Parkhill. Kinetic energy of hydrocarbons as a function of electron density and convolutional neural networks. *Journal of chemical theory and computation*, 2016.

[100] Wenbo Yu and Alexander D. MacKerell. Computer-aided drug design methods. *Methods in molecular biology (Clifton, N.J.)*, 1520:85–106, 2017.

[101] Yan Zhu, Saad Alqahtani, and Xiche Hu. Aromatic Rings as Molecular Determinants for the Molecular Recognition of Protein Kinase Inhibitors. *Molecules*, 26(6):1776, January 2021. Number: 6 Publisher: Multidisciplinary Digital Publishing Institute.