POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

A Recommender System Approach for in-silico drug discovery using HPC architectures

Laurea Magistrale in Computer Science and Engineering - Ingegneria Informatica

Author: Alessio Russo Introito

Advisor: Prof. Gianluca Palermo

Co-advisors: Ph.D Davide Gadioli, Ph.D Stefano Cereda

Academic year: 2021-2022

1. Introduction

In recent years, *drug development* has undergone a strong growth, mainly pushed by the Covid-19 pandemic and the resulting research for new treatments. The process of bringing new drugs on the market is still characterized by large capital investments and complex pipelines of execution, which last 10-15 years on average. However, these rising costs don't find a proper correspondence on the *discovery* of therapeutic medications since still just a small portion (12%) of clinical trials eventually becomes an approved medicine. Therefore, *drug discovery* remains one of the biggest and most expensive challenges in the pharmaceutical industry, attracting numerous academic and industrial researchers in finding advanced methodologies to increase the efficiency of its pipelines.

The search for new medications pursued in drug discovery techniques consists in detecting the compounds that show a biological activity with the "target" molecule they bind on: in general, the "target" is represented by large macro-molecules such as *proteins* or nucleic acids, whereas the binding compounds are much smaller in size and they are usually called *ligands*. Finding the good candidates for a protein involves the screening of large chemical libraries, which can be expensive when Molecular Docking techniques are employed: the latter includes a collection of methodologies that exploit the protein's three-dimensional structure to evaluate its affinity with a ligand. However, these techniques are time-consuming and computational-intensive and require costly HPC solutions to perform large-scale analysis. Moreover, there is no preference in the choice of the compounds to submit to the docking evaluation, which leads to a poor exploration of the chemical space.

In this work, we address this problem by proposing a prioritization of the molecules that are likely to be a good fit for the protein into account. To achieve this, we apply a Recommender Systems approach capable of utilizing previously computed docking evaluations to recommend the most promising molecules to a new protein. To validate our work, we emulate a real-case scenario characterized by an *iterative* procedure where compounds are evaluated in batches. The results obtained show how even simple Recommender Systems models are able to efficiently select the relevant portion of compounds to be tested, reducing the costs and the time required to evaluate the most promising molecules of a

protein.

2. Background

In this thesis, we are going to focus on two main topics: *Virtual Screening*, which aims to employ in-silico experiments reducing the chemical space to a small set of possible candidates for drug-discovery, and *Recommender Systems*, which include a number of techniques for providing suggestions on *items* based on the preferences of a set of *users*.

2.1. Virtual Screening

In cheminformatics, we can refer to *chemical space* as the space containing all possible molecules and chemical compounds. The amount of chemical entities that lies in this space is extremely huge, composed of over 10^{63} possible elements. Because of its size, a complete and exhaustive exploration of the entire space is theoretically unmanageable. Consequently, several techniques have been developed to address this problem and scale down the number of compounds to be considered for synthesis and testing purposes. In the last decade, Virtual Screening is the one that has aroused the most interest among academic researches, driven by the desire of developing new methods to improve the screening quality at the early stages of a drug-discovery pipeline. *Virtual Screening* can be interpreted as the set of methodologies that rely upon computational resources to process large chemical libraries, pursuing the goal of finding potential chemical candidates that are most likely to fit the protein's shape.
Such approaches allow to evaluate billions of compounds in a reasonable amount of time, leveraging the parallelism powered by *High-Performance-Computing* infrastructures: HPC systems are composed of a cluster of interconnected nodes placed in a distributed environment and characterized by high computing capabilities.
Virtual screening techniques can be categorized as *ligand-based* and *structure-based*. The former focuses only on a set of ligands whose binding activity with the target is known a-priori, so that *similarity* measures or *Machine Learning* models can be employed to extract structurally similar compounds or classify them based on their activity/non-activity status. *Structure-based* approaches rely upon the knowledge of the three-dimensional structure of the target protein to perform a more detailed analysis of protein-ligand interaction. Thanks to the advent of X-ray crystallography and Protein-Nuclear-Magnetic resonance (NMR) spectroscopy that escalate the availability of 3D-representation of protein's structures, complex analysis can be addressed by exploiting a structure-based technique known as **Molecular Docking** (Figure 1). This kind of approach operates following two different phases:

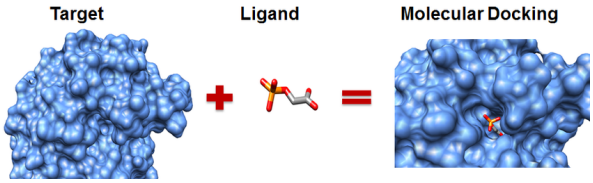

Figure 1: Molecular Docking [5]

1. Searching Algorithm: it searches in the 3D space the best conformation and orientation of a ligand when it binds to the protein
2. Scoring Function: it assigns a score to the best pose found that measures the affinity of the protein-ligand pair

Given the large amount of possible poses to fit a ligand into the protein's binding sites, Molecular Docking algorithms can lead to computationally intensive tasks when large sets of chemical compounds need to be screened.

2.2. Recommender Systems

Recommender Systems can be described as the collection of methods that allow discovering the preferences of users based on their past interactions with a set of items. The meaning behind "users" and "items" depends on the context we are dealing with, but in general, *users* are the active agents that interact with some *item*. The relationship between them defines the so-called *rating*, which can express the existence (*implicit*) or the goodness (*explicit*) of the interactions themselves. These concepts are grouped together into a *User Rating Matrix* (URM) that represents the key data structure in RS problems. Moreover, in most of the domains, additional information regarding properties of users and items is available, shaped into two different matrices that show for each user/item the corresponding

set of observed *features*: *User-Content Matrix* (UCM) and *Item-Content Matrix* (ICM). These data structures can be used to feed RS models in order to extract recommendations.
RS models can be divided into two main classes, namely *Collaborative* and *Content-based* approaches. The former uses the preferences users gave to items to generate predictions for another user, thus only the information in the URM is taken into account. Instead, the latter leverages the *features* to discover items similar to the ones in the user's profile. In both cases, it could be necessary to define a pairwise *similarity* measure between users or items to be applied during the prediction phase.

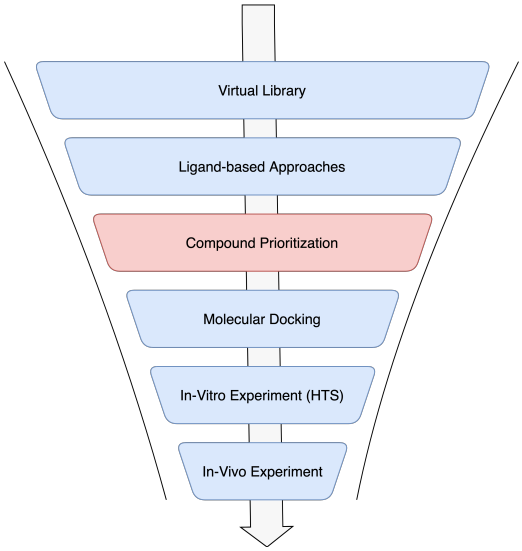

Figure 2: Proposed Virtual Screening Funnel

3. Contribution

State-of-the-art Molecular Docking software efficiently explores the conformational space to find the best poses and analyze the attractive forces that regulate a protein-ligand interaction. This software is integrated into Virtual Screening pipelines to detect the most peculiar molecules that bind a particular protein. However, the vastness of the chemical space makes impossible an evaluation in its entirety, but only a portion of it can be taken into account. In addition, the screening of large chemical libraries, composed of billions of compounds, via Molecular Docking techniques is computationally demanding and requires elevated costs to employ the appropriate resources in an HPC environment. Simulations in HPC infrastructures are usually subjected to strict temporal and resource constraints. As a consequence, in order to improve the quality of the screening pipeline, a subset of ligands to be tested should be properly chosen to increase the probability of finding promising molecules and minimize the time required for the simulation.
To achieve this goal, we propose an approach to prioritize (Figure 2) the evaluation of molecules that are most likely to fit the protein's binding site with high affinity, exploiting a set of previously computed protein-ligand interactions. The prioritization is addressed with the application of Recommender Systems algorithms, which study the preferences of proteins, trying to recommend them the *most promising* molecules. Accordingly, the idea is to guarantee a sorted sequence of compounds to be submitted to the next phases of a drug-discovery pipeline, such that the promising molecules are evaluated first, while the others are postponed: this approach allows us to efficiently manage expensive HPC simulations in which constraints on the cluster usage limit the exploration of ligands.
In this scenario, proteins take the role of *users*, ligands replace the *items*, whereas the *ratings* express a value of their binding affinity. Since the use of Recommender Systems in Molecular Docking screening is restricted to very few researches, no data is currently available to continue further our analysis. To face this problem, we decide to gather the data on our side, implementing a docking procedure to extract a collection of protein-ligand interaction scores.

3.1. Data Generation

As previously stated, the application of Molecular Docking in a Virtual Screening context requires a computationally intensive effort that can be offered by HPC architectures. We are able to implement our simulation thanks to the collaboration with IT4Innovations' National Supercomputing Center, which makes available one of the most powerful supercomputing systems in Europe. The access to its clusters is simplified by offering solutions of *HPC-as-a-Service*, thus providing high-level functionalities that relieve the customer from any additional duty. In particular, the supercomputing power is served through a framework called *High-End Applica-*

tion Execution Middleware [1] (HEAppE), which allows us to easily handle authentication policy, data transfer and job operations (submission, monitoring, management).
Since the procedure of docking a single ligand to a protein is not correlated to any of the others, the embarrassing parallelism can be tackled by splitting the work among different computing nodes in the distributed environment. Accordingly, the parallelism is achieved at two-level basis:

- **Inter-node**: different nodes communicate with each other following a *Message Passing* (MPI [2]) paradigm in order to distribute the workload and coordinate operations.
- **Intra-node**: the multiple CPU-cores belonging to each machine allow the application of *multi-threading* strategies to run multiple I/O tasks simultaneously.

The simulation starts by considering a large library of compounds and a set of proteins on which apply docking and scoring algorithms. In our configuration, the two phases involved in the Molecular Docking procedure use **GeoDock**[3] to explore the three-dimensional space to find the best pose (*searching algorithm*), and **X-Score**[6] to eventually establish an interaction score for the best poses found in the previous phase.
The chemical library is stored in a memory shared among all the machines, thus the load can be split assigning to each node a particular portion of the dataset to process. The real core of the elaboration resides at the node level, when a sub-routine, named **Executor**, is called. This routine is in charge of efficiently exploiting the *intra*-node resources to scale up the number of molecules that can be processed. For this purpose, multiple actors are considered. In particular, different *threads* in the same node are able to communicate with each other through a shared memory data structure known as *Queue*, which represents a FIFO queue where threads can fetch or push data. Moreover, the spawned threads can be summarized as follows:

- Reader: it reads a ligand from the library
- Worker: it processes a ligand-protein pair
- Writer: it aggregates the results in a file

The Executor procedure is developed following the *Consumer-Producer* paradigm. Initially, the Reader thread parses the dataset to locate the ligands which are accumulated into a *Task Queue*. Then, a Pool of Workers fetches a set of ligands from the same queue and applies the docking and scoring pipeline to extract a value of the binding affinity. Their results are sent to the Writer that groups them into the same file. When all the nodes complete the screening of their portion of the dataset, the results are gathered and the procedure terminates.

3.2. RS for Virtual Screening

The pipeline discussed in the previous chapter lets us process around 8.5 million molecules against 39 proteins and collect the corresponding interaction scores. These data can be used to serve Recommender Systems models, shaping this set of protein-ligand scores into a URM. The main difference with classical RS problems is related to the fact that our resulting URM is extremely *dense* because an affinity score is provided to each protein-ligand pair; in addition, an implicitization procedure to transform our *explicit* data into implicit one cannot take place, since the generated affinity scores represent a measure of the atomic binding forces, thus also the lowest can indicate the presence of chemical activity.
In this work, we focus on classical RS approaches to recommend the *most promising* molecules to a new protein when the latter is subjected to a screening analysis. In particular, we consider two *collaborative* approaches: a Top Popular model that tries to recommend the molecules with the highest average scores, and a *memory-based user-user* Collaborative Filtering[7] model, which instead uses the *Spearman Correlation Coefficient* as a metric to outline the similarities among the interaction scores of the proteins. Finally, the presence of a set of features describing some properties of the compounds gives us the ability to introduce an *item-item KNN* Content-based[4] model, which aims at suggesting molecules similar to those available in the protein's profile, using the *Cosine* similarity metric.

4. Result

To assess the performance of our models, we emulate a real drug-discovery scenario in which batches of evaluations are performed at time, fol-

lowing an *iterative* process to explore as many ligands as possible. Ideally, at each *round* of evaluation:

1. A model is trained with the current binding affinity scores
2. A set of recommendations is extracted for the new protein
3. The recommendations are evaluated in-vitro or in-vivo to determine if the protein-ligand complexes can be synthesized
4. The affinity scores of the recommendations are added to the dataset

However we cannot perform any in-vitro or in-vivo synthesization to test the activity of the complex, so we use the docking pipeline previously described to measure the interaction strength. The goal is to **minimize** the number of rounds required to discover all the *promising molecules*. In our testing scenario, we integrate this iterative procedure with a *cross-validation* approach in which each validation fold corresponds to a single protein, while the rest of the proteins are used for training (*leave-one-protein-out*). Hence, the tested protein initially starts without any available interaction score (new protein), such that at each *round* a set of recommendations is added to its profile: to address the well-known *cold-start* problem, i.e. the difficulty in extracting recommendations when no previous interactions are available, we adopt the Top Popular to compute suggestions in the first round of evaluations.

To capture the quality of the models over the multiple *rounds* of evaluations, we use a *Cumulative Recall*, which tracks the number of *promising molecules* detected up to a certain round, and, to scope the overall results, we compute the *Area-Under-the-Curve* (AUC) that Cumulative Recall describes.

The Recommender Systems models are evaluated with respect to an initial baseline: since state-of-the-art Molecular Docking techniques do not provide any preferential choice of the ligand, the baseline model emulates this behavior by carrying out a **random** selection of the molecules.

Figure 3 shows the immense gap between the performance of a Random and the Top Popular models in recommending relevant items. The results of a random selection are justified by the fact that it is extremely hard to discover the

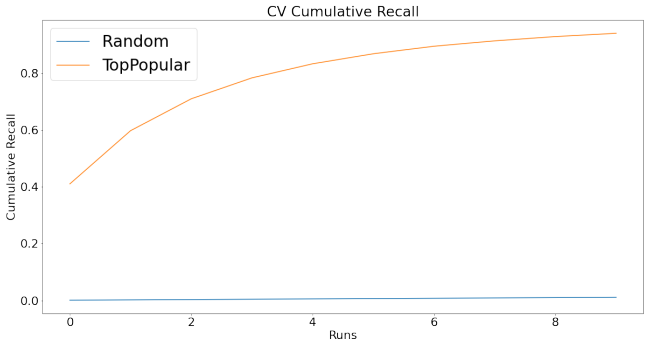

Figure 3: Random and Top-Popular CumRecall

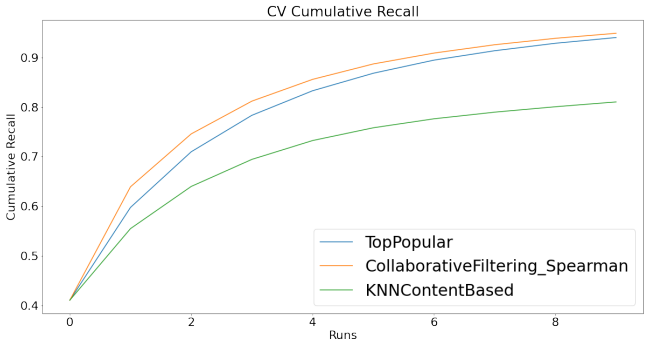

Figure 4: CumRecall of RS Models

most promising molecules randomly picking a subset of the compounds, especially if the size of the dataset increases. On the other hand, the results of the Top Popular reveal how a simple recommender can efficiently address this exploration problem, such that after 10 rounds almost all the relevant molecules are detected. In addition, since all the interaction scores lay in a limited range of values and our Top Popular recommends the items having the largest average score, its ability to discover relevant items implies that there are molecules that are a good fit (i.e. high score) for multiple proteins.

This behavior motives us in exploring other *collaborative* techniques; in fact, the results of *the Collaborative Filtering* model, depicted in Figure 4, confirms the ability of these kinds of approaches to face the problem, increasing the steepness of the curve and the corresponding AUC (Table 1)

Figure 4 also shows the Cumulative Recall of the Content-based, which turns out to be much better than a random exploration, but its results are not as good as the collaborative approaches. A more detailed analysis of the features used by this model reveals how the X-Score, one of the most used scoring functions, is **strongly biased**

Model	AUC
Random	0.00569
Top Popular	0.7204
Collaboartive Filtering	**0.7393**
Content-Based	0.6357

Table 1: Area Under The CumRecall Curve

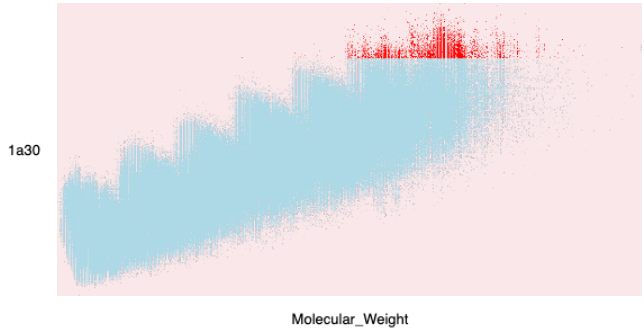

Figure 5: Distribution of Scores w.r.t. Molecular Weight

toward molecules with a high molecular weight. This can be immediately noticed in Figure 5 by plotting the weights of the molecules with respect to their scores extracted for the protein *1a30*. Among all the molecules (blue points), the set of its most promising molecules is highlighted in red. The linear correlation between the two observations is clear in the Figure 5, such that as the weight increases, the score increases as well.

5. Conclusion

In this thesis, we reviewed the state-of-the-art methods in Virtual Screening, giving special attention to Molecular Docking techniques. These techniques aim at finding the best reciprocal pose with which a protein and a ligand can interact. However, Molecular Docking approaches are quite expensive when a large chemical library has to be screened. As a consequence, we propose the application of RS models to prioritize the evaluation of ligands, restricting the computation to just the relevant portion of data. The results show how well the RS models are able to detect promising candidates in comparison to a random selection.
Future works can investigate the performance of applying different scoring functions or advanced RS models.

References

[1] IT4Innovations National Supercomputer Center. Heappe framework.

[2] Message P Forum. Mpi: A message-passing interface standard. Technical report, USA, 1994.

[3] Davide Gadioli, Gianluca Palermo, Stefano Cherubin, Emanuele Vitali, Giovanni Agosta, Candida Manelfi, Andrea R. Beccari, Carlo Cavazzoni, Nico Sanna, and Cristina Silvano. Tunable approximations to control time-to-solution in an HPC molecular docking mini-app. *CoRR*, abs/1901.06363, 2019.

[4] Simon Philip, Peter Shola, and Ovye Abari. Application of content-based approach in research paper recommendation system for a digital library. *International Journal of Advanced Computer Science and Applications*, 5, 10 2014.

[5] Yair Tenorio, Alejandra Hernandez-Santoyo, Victor Altuzar, Hector Vivanco-Cid, and Claudia Mendoza-Barrera. *Protein-Protein and Protein-Ligand Docking*, page 187. 05 2013.

[6] R. Wang, L. Lai, and S. Wang. Further development and validation of empirical scoring functions for structure-based binding affinity prediction. *J Comput Aided Mol Des*, 16(1):11–26, Jan 2002.

[7] Ruisheng Zhang, Qi-dong Liu, Chun-Gui, Jia-Xuan Wei, and Huiyi-Ma. Collaborative filtering for recommender systems. In *2014 Second International Conference on Advanced Cloud and Big Data*, pages 301–308, 2014.