POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

# A High-throughput pose selection method for extreme scale virtual screening in drug discovery

Laurea Magistrale in Computer Science Engineering - Ingegneria Informatica

**Author: Gianmarco Accordi**

**Advisor: Prof. Gianluca Palermo**

**Co-advisor: Davide Gadioli, Alfonso Gautieri**

**Academic year: 2021-2022**

## 1. Introduction

Humans in history have always tried to increase their quality of life, thus their lifespan. An increase in a human lifetime duration usually coincides with a breakthrough in medicine: for example the discovery of a new treatment against a disease. Drug discovery is the identification process of suitable new treatments against a disease. Before computers, this process was done by researchers in vitro and in vivo: researchers were testing compounds that are likely to have effects on disease. This is a very long and costly process [1]. Part of this process can be anticipated and simulated with an in-silico stage named virtual screening, which aims at finding small molecules named ligands, that have a strong interaction with a target protein, usually named receptor. Ligands can be an ion or a molecule composed of tens of atoms. Receptors are a class of proteins that can bind to ligand molecules. Upon binding, ligand and receptor will change their conformations. Domain experts expect a beneficial effect from this interaction. Since this stage can be simulated in-silico, the number of evaluated ligands is limited only by the computational power.
Compounds, that are selected by the whole virtual screening pipeline, are passed to later stages of the drug discovery process, for further analysis. A virtual screening pipeline is usually executed on a supercomputer. Since a higher throughput leads to a greater number of screened ligands. In this way, a virtual screening pipeline execution is feasible in a reasonable amount of time [2]. The first studies for antivirals against HIV and Influenza were limited to testing a number of 100 compounds in a reasonable time frame. In the 90s the power of supercomputers has increased that number to roughly a million compounds tested in a reasonable time frame [2]. One of the most recent successes in the application of the virtual screening pipeline for drug discovery has been done on the Summit IBM supercomputer: researchers have been able to perform exhaustive docking of one billion ligands in under 24 hours [2]. Similar results have been obtained in europe with the Exscalate4Cov project [3].
In the last decades, the number of available compounds in a dataset of ligands is increased dramatically, requiring more computational power to test these ligands against a target. Increasing the throughput of virtual screening pipelines has become crucial to meet those requirements.

Up to now the miniaturization of computer components has led to an increase in the available computational power and efficiency while reducing costs. The decline of Moore's law has led researchers to think differently and to pursue new solutions. Approximate computing is used to trade computational complexity, thus computing time, with the accuracy of the results. Approximated computing can take advantage of some statistical properties of the data, to reduce the complexity. This thesis studies the possibility of increasing the number of compounds tested in the same amount of time, by using approximation techniques. In particular, the idea is to use approximate results as hints to drive the computation effort in a virtual screening pipeline.
The structure of this document starts with a brief introduction to the concepts required to understand the work done. It continues with the state of the art, which shows how others have approached similar problems. After that the proposed methodology for the solution is explained: the thesis has made use of precision scaling and memoization to increase the throughput of a virtual screening pipeline. Precision scaling reduces the size of the inputs space, making it feasible to make an exhaustive search in it. While memoization allows us to move most of the computation at precomputation time, making the stages of the virtual screening pipeline less computationally heavy. Once a solution has been identified the thesis explains how I've proceeded in implementing such a solution in a specific virtual screening pipeline, and the collected results I've obtained while designing the solution to validate the approach.

## 2. Background & State of the Art

The molecule is not a rigid body: a subset of the bonds between the atoms enable the molecule to change shape, without altering its physics-chemical properties. These bonds are named rotatable bonds. Since ligands are placed into a 3D space they have 6 degrees of freedom (3 translations, 3 rotations) and $r$ degrees of freedom based on the number of rotatable bonds in it. A ligand's conformation can change in space, this is called a pose: each ligand can have multiple poses.

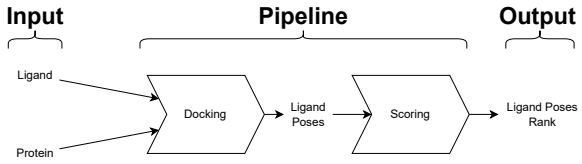


Figure 1: Virtual screening Pipeline

This thesis focuses its attention on *Virtual High-Throughput Screening* which uses computer-generated models to evaluate how a ligand binds to a receptor. Figure 1 shows a virtual screening pipeline. Among the virtual screening stages, the most important ones are docking and scoring. Docking analyzes the whole search space for all the possible orientations and conformations of the ligand and the receptor upon binding by using heuristic and docking techniques. The scoring stage instead scores the conformations obtained by the sampling stage. The highest scored poses might be selected as leads compounds for further analysis. The evaluations given to the poses are called scoring. The higher the score, the more stable the complex and the bond strength, will be. We would like to stress the fact that the number of poses scored by a scoring function is lower than the sampled ones. If we consider its implementation the inputs are the ligand and the protein. These inputs are passed to a molecular docking algorithm, which produces as output a set of ligands' conformation, also called poses. These poses will be evaluated by a scoring function, and based on the scores the poses are ranked. The rank will be the output of this pipeline. From the top-scored poses in the output, we can retrieve the lead compounds for further analysis in the drug discovery process.
We have chosen to apply our proposed solution to the LiGen pipeline, which is proprietary software owned by Dompè, used in the Exscalate4Cov project. E4C aims at using the EU's computing resources, to respond much faster to international pandemics.
Since docking algorithms produce as output a set of ligand poses, a metric that evaluates the goodness of a pose is needed. Different types of scoring functions exist based on how they consider the interactions between atoms. XSCORE [4] is the scoring function we have further


Figure 2: Input discretization process.

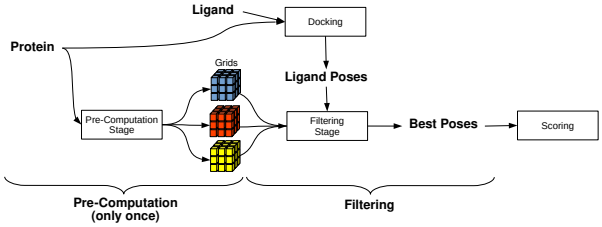


Figure 3: High throughput poses selection with pre-filtering implementation architecture.

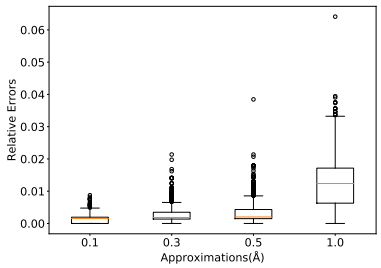


Figure 4: Distribution of the XSCORE errors with quantized space. On the y-axis the relative difference between the original values and the quantized ones are reported. While on the x-axis there are different grid resolution.

analyzed.

## 3. Proposed Methodology

The problem we are facing in this thesis consists in accelerating the virtual screening pipeline by applying approximated computing. To do so we have to move most of the scoring computation before the ligand is docked onto the receptor. Through a process of space gridification, the idea is to use fast lookup tables, which contain pre-computed values, to reduce the computation time of the pipeline's scoring stage. The first approximation technique we have applied is precision scaling to an atom's coordinates. Figure 2 shows how with precision scaling I've discretized the input space. In this way, I've obtained a finite number of input combinations. Figure 3 propose a modified version of the pipeline from Figure 1, highlighting the modification to the pipeline proposed in this work. Before launching the virtual screening pipeline, the pre-computation stage takes the protein's structure and computes the grids required by the filtering stage. The filtering stage takes the ligand's poses produced by the docking algorithm, and it evaluates them according to the approximated scoring function based on the pre-computation stage. We used this approximated score to select which are the poses of the molecule that needs to be re-scored using the original scoring function.
The implemented application has been designed with a multi-CPU and multi-GPU approach. The typical HPC infrastructure has a heterogeneous architecture with multiple-node, and with accelerators on each node (usually GPUs). In our case, we have targeted NVIDIA's GPUs as accelerators, so the adoption of CUDA to program is mandatory. While for the node communications we have used OpenMPI.

## 4. Scoring Approximations

### 4.1. Scoring Function Analysis

As anticipated we used a well known scoring function for the purposes of this document. The disassembling of XSCORE becomes essential, to find how we can pre-compute each of its components. The computation of the Van der Walls interactions, the hydrophobic contact term, and the hydrophobic matching term show dependencies on the atoms' radiuses and on the atom's coordinates. Hydrogen Bond computation requires the atoms' coordinates, atomic type, and the atoms' root coordinates. The root of an atom is at the geometric center of all its non-hydrogen neighboring atoms. The deformation effect computation requires the knowledge of the atom's rotatable bonds, also called rotors. While the hydrophobic surface term computation requires the identification of the solvent accessible surface (SAS).
Therefore, the information from the actual ligand required for the computation of XSCORE factors is the following: atom's radiuses, atom's coordinates, and atom's types. The atom's type refers both to the atom's element (nitrogen, hydrogen, etc.) and to the atom's capacity of accepting or donating an electron (donor, acceptor, donor-acceptor). The combined action of precision scaling and memoization led to the de-

cision of structuring pre-computed data in memory as grids. For each XSCORE component, a grid is pre-computed. Each XSCORE's component computation depends on different factors (atoms radiuses, types, coordinates). Grids are a 3D matrix in which each cell represents a position of the space, in which we place an atom probe for evaluating a function return value. Each point of the grid stores multiple results of the same function, based on the atom's probe properties. Atom's properties can change, in our case atom probe can change radius, type, or both. Since radiuses are a real value, they are discretized along with the atom probe position in space. The grids' dimensions are defined by the receptor bounding box, while the grids' spacing has to be choose based on the desired results' accuracy. Using lower spacing values will not reduce much the approximation error, while the memory requirements will be higher since the number of grids' points is higher.

### 4.2. XSCORE Approximations

Now that the approximations strategies used are defined, and the XSCORE implementation has been described, is time that we analyze how each XSCORE's components have been approximated. Precision scaling and memoization should be combined, to make feasible the precomputation stage, since it is required to reduce the time-to-solution of the pipeline. On the Van der Waals interactions, the hydrophobic contact term, and the hydrophobic matching term I've applied precision scaling on the atoms' coordinates and radiuses, and on all the possible inputs combinations I've computed the value of these factors and stored them in memory for later lookup. Hydrogen bond term cannot be pre-computed since the quantization of the atoms' root is not feasible, and the inputs space is too big for applying memoization. To resolve this problem I've instead used the hydrogen bond factor computation of the Autodock scoring function [5], which does not consider the atoms' root. The hydrophobic surface term cannot also be pre-computed because it requires the computation of the ligand's SAS, which is unavailable during this phase. Since a deeper analysis has shown how the hydrophobic surface term accounts for only 4% on the final score, I've disabled its computation, with also the SAS

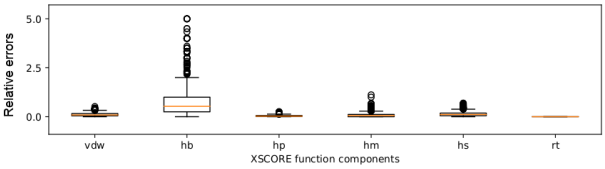


Figure 5: Relative error on VDW, HB, HP, HM, HS, RT when using precision scaling. The y-axis reports the relative errors. While on the x-axis there are the box plots of the single component.

computation. In this way, the scoring time is reduced by about 35%. Also, the deformation effect term has been disabled, because it is the same among all the poses of the same ligand, so it is the same as adding a constant factor to each score.

## 5. Results

The implementation of the proposed approximations has the purpose of increasing the throughput of a virtual screening pipeline. This section reviews the results collected while designing the proposed implementation. The objective is to demonstrate from the results how the proposed solution can approximate the functionalities of a scoring function, in terms of RMSD and scoring time. At first, the analysis is done by considering the approximated scoring function as a stand-alone module to compare against the original. Then, this section will evaluate the module as integrated into a virtual screening pipeline.
The first step in the development of the solution was to analyze the impact of precision scaling on XSCORE's factors. After analyzing the results obtained with different grid resolutions, going from 0.1 to 1.0 Å, we have decided to approximate each atom's coordinates using a grid spacing of 0.5 Å, since we have seen it is the best tradeoff between accuracy, memory requirements, and computational time, as shown by Figure 4. The analysis I've done shows that the number of radiuses clusters, which should be used is 5, since using more ranges will not reduce the approximation error, while it will increase memory requirements. Figure 5 shows the impact of precision scaling on the single components' values. Each box plot shows the distribution of differences between the original component's value and the approximated one. We want to remark the sensibility of HB to approxi-

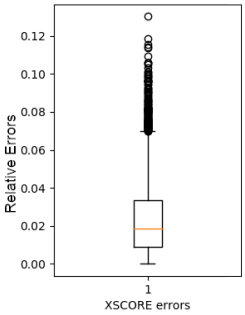


Figure 6: XSCORE values approximations when all the proposed approximations of Section 4.1 are active.

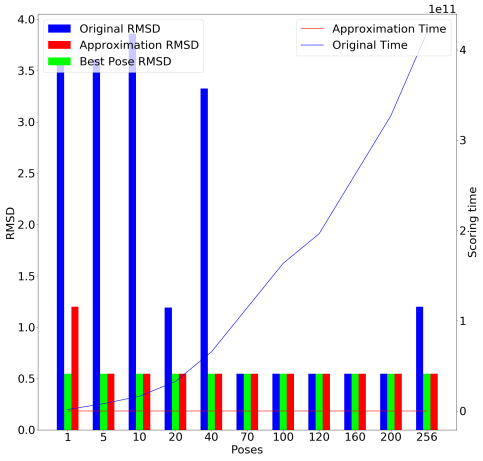


Figure 7: The relationship between the RMSD and scoring time, between the original and the approximated XSCORE versions.

mations on atoms' coordinates. In fact, is worth noticing that HB computation depends also on the atoms' root. not only on the distances. This analysis demonstrates that the sensibility of XSCORE to precision scaling is very limited. The approximation error is usually lower than 2%. The final considerations about the errors introduced are reported in Figure 6, which shows the relative errors of the original XSCORE value with the approximated one. It is possible to notice how the low error can be used to select the poses to evaluate with the original scoring function.

Indeed the next step is the analysis of the filter when deployed in a virtual screening pipeline. The next analysis aims at demonstrating how approximated computing techniques reduce the required scoring time, while will maintain an acceptable RMSD with XSCORE. Figure 7 highlights with bars the RMSDs between the golden

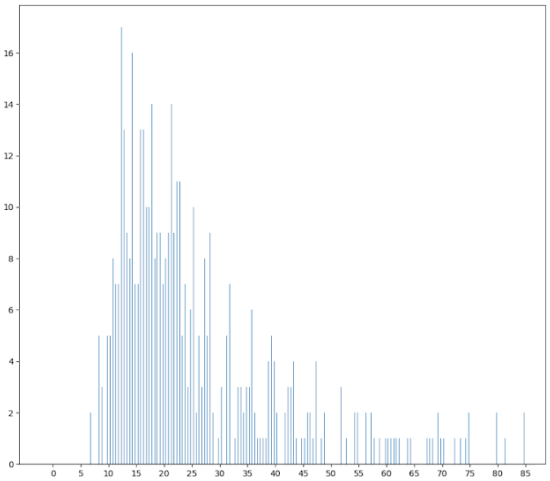

Figure 8: Speedups obtained when using the GPU version of the solution with respect to the original only CPU XSCORE version.

reference and, the best possible pose obtained from a docking algorithm, the best-scored pose with the original XSCORE version, and the best-scored pose with the approximated version of XSCORE. On the x-axis, there is the number of scored poses. For the original version, each value X on the x-axis is the number of poses taken from the docking's output, passed to XSCORE, then we compute the RMSD using the best-scored pose. For the proposed filter all the docking's output poses are scored then we select the first X poses, and we have reported the best RMSD among the first X poses. The lines instead compare the scoring time between the original and the approximated version of XSCORE. The lines refer to the scoring time, and they show how the approximated version is much faster than the original version. The approximated filter is much faster than XSCORE, it can be used to analyze all the poses in a fraction of the time, in this way also with a small value of X we can identify almost immediately the poses with the lowest RMSD. For example with X=1, we can see that in the same amount of time the approximated version can get a poses with the lowest RMSD, because it is able to scan the whole output of the docking stage, while the original version was able to score only one element.

Moreover, this section shows the speedup obtained with the usage of MPI and CUDA for parallelizing the proposed solution. Figure 8 has on the y-axis the number of occurrences of

| Protein Name | Only Scoring | | Filtering+Scoring | | Speedup |
|---|---|---|---|---|---|
| | Time(ms) | Best RMSD(Å) | Time(ms) | Best RMSD(Å) | |
| **1a0q** | 3076 | 3.04 | 240 | 3.04 | 12.8 |
| **1a0t** | 3588 | 5.6 | 312 | 5.6 | 11.5 |
| **1a1b** | 5322 | 10.6 | 401 | 10.6 | 13.2 |
| **1a3e** | 13065 | 7.27 | 1315 | 7.27 | 9.93 |
| **1a4h** | 5926 | 1.3 | 598 | 1.08 | 9.9 |
| **1a5g** | 7025 | 2.99 | 690 | 2.99 | 10.4 |
| **1a5h** | 4726 | 2.6 | 399 | 2.6 | 11.8 |
| **1a07** | 4748 | 5.7 | 381 | 5.7 | 12.4 |
| **1a7x** | 10832 | 2.85 | 2124 | 2.85 | 5.09 |
| **1a08** | 5925 | 10.7 | 438 | 10.7 | 13.5 |

Table 1: The table reports the comparison between the RMSD and scoring time when using the LiGen pipeline with and without the filtering stage with the proposed approximations (it selects 15 poses).

the speedup on the x-axis, for different ligand-receptor pairs. The performance improvement is very consistent: on average the approximated version of XSCORE gets a 26x speedup with respect to the original only CPU version of XSCORE in a single node.
In this thesis, I wanted to demonstrate that approximated computing can be used in a virtual screening pipeline with acceptable errors in the results. When used as pose filter, the average approximations are around 2%, which is an acceptable value concerning the speedup of 26x, as shown in Figure 8.

### 5.1. LiGen Integration

The analysis of the proposed approximations as stand-alone module is now integrated with its analysis when integrated into a real-world virtual screening pipeline, like the LiGen one. In Table 1 I've reported the results obtained with the filtering stage I've proposed in the LiGen pipeline. For each of the proteins, I've registered the RMSD of the golden reference with respect to the best pose obtained, in two different cases: in one case the filter has been placed in between the docking stage and the scoring stage (XSCORE) of the LiGen pipeline (Filtering+Scoring), in the other case the filter is not placed in between the two stages (Only Scoring). The filter will select the best 15 poses coming from the docking stage, and it will pass them to the scoring stage. I've also recorded the scoring time of each version. The last column reports the speedup obtained when using the modified pipeline. Table 1 shows how the modified version with the filtering module, with the proposed approximation before the scoring is much more efficient: the average speedup is equal to 11x, and the RMSD of the best pose is almost the same between the two versions.

## 6. Conclusions

The objective of this thesis was to analyze how to increase the throughput of a virtual screening pipeline for drug discovery. The thesis adopted precision scaling and memoization to approximate the scoring stage of the virtual screening pipeline. The results experimental results obtained with the integration of the approximated module inside a virtual screening pipeline, have demonstrated how this approach is feasible and leads to great results. Future developments on this topic concern a deeper optimization of the pre-computation and filtering stage, along with a deeper analysis of the advantages of the streams used in the CUDA implementation. Another interesting expansion of this work can be the analysis of the impact of the proposed methodology when processing different proteins at the same time, since the current proposed solution stress the memory a lot.

## References

[1] The cost of drug development: A systematic review. *Health Policy*, 100(1):4–17, April 2011. doi: 10.1016/j.healthpol.2010.12.002. URL `https://doi.org/10.1016/j.healthpol.2010.12.002`.

[2] Supercomputer-based ensemble docking drug discovery pipeline with application to covid-19. *Journal of Chemical Information and Modeling*, 60(12): 5832–5852, December 2020. doi: 10.1021/acs.jcim.0c01010. URL `https://doi.org/10.1021/acs.jcim.0c01010`.

[3] EXSCALATE: an extreme-scale in-silico virtual screening platform to evaluate 1 trillion compounds in 60 hours on 81 PFLOPS supercomputers. *CoRR*, abs/2110.11644, 2021. URL `https://arxiv.org/abs/2110.11644`.

[4] Further development and validation of empirical scoring functions for structure-based binding affinity prediction. *Journal of Computer-Aided Molecular Design*, 16(1):11–26, 2002. doi: 10.1023/a:1016357811882. URL `https://doi.org/10.1023/a:1016357811882`.

[5] Automated docking using a lamarckian genetic algorithm and an empirical binding free energy function. *Journal of Computational Chemistry*, November 1998. URL `https://doi.org/10.1002/(sici)1096-987x(19981115)19:14<1639::aid-jcc10>3.0.co;2-b`.