2-1-2015

A Novel Scoring Based Distributed Protein Docking Application to Improve Enrichment

Prachi Pradeep
Marquette University, prachi.pradeep@marquette.edu

Craig Struble
Aria Diagnostics, Inc

Terrence Neumann
Texas Wesleyan University

Daniel S. Sem
Concordia University - Wisconsin

Stephen Merrill
Marquette University, stephen.merrill@marquette.edu

Accepted version. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Vol. PP, No. 99 (2015). DOI.

IEEE/ACM Transactions on Computationsal Biology and Bioinformatics, Vol. 12, No. 6 (Nov-Dec 2015): 1464-1469. DOI.

A Novel Scoring Based Distributed Protein Docking Application to Improve Enrichment

Prachi Pradeep
Department of Mathematics, Statistics, and Computer Science, Marquette University, WI

Craig Struble
Aria Diagnostics, Inc., San Jose, CA

Terrence Neumann
Department of Chemistry and Biochemistry, Texas Wesleyan University, TX

Daniel S. Sem
School of Pharmacy, Concordia University Wisconsin, Mequon, WI

Stephen J. Merrill
Department of Mathematics, Statistics, and Computer Science, Marquette University, WI

ABSTRACT

Molecular docking is a computational technique which predicts the binding energy and the preferred binding mode of a ligand to a protein target. Virtual screening is a tool which uses docking to investigate large chemical libraries to identify ligands that bind favorably to a protein target. We have developed a novel scoring based

distributed protein docking application to improve enrichment in virtual screening. The application addresses the issue of time and cost of screening in contrast to conventional systematic parallel virtual screening methods in two ways. Firstly, it automates the process of creating and launching multiple independent dockings on a high performance computing cluster. Secondly, it uses a Nàive Bayes scoring function to calculate binding energy of un-docked ligands to identify and preferentially dock (Autodock predicted) better binders. The application was tested on four proteins using a library of 10,573 ligands. In all the experiments, (i). 200 of the 1,000 best binders are identified after docking only ~14 percent of the chemical library, (ii). 9 or 10 best-binders are identified after docking only ~19 percent of the chemical library, and (iii). no significant enrichment is observed after docking ~70 percent of the chemical library. The results show significant increase in enrichment of potential drug leads in early rounds of virtual screening.

SECTION 1, INTRODUCTION

Modern drug discovery is a “lengthy, expensive, difficult, and inefficient process” with low rate of new therapeutic discovery [1]. Currently, the research and development cost of each new molecular entity is approximately US $1.8 billion [2]. Conventional experimental procedures such as medicinal chemistry and high throughput screening (HTS) are still the most accurate methods for rapid identification of drug leads. However, there is an enormous growth in commercial and publicly available chemical structure libraries of potential drug compounds (ligands), such as the ZINC database [3], [4] (contains over 21 million compounds), which require more efficient techniques for screening. In this context, computational methods are now being used to enhance the drug development process [5] , [6].

Molecular docking is a computational technique which predicts the interaction between a protein and a potential drug compound [7], [8], [9]. Virtual screening, the use of high-performance computing (HPC) clusters to analyze large databases of ligands, is a well established and cost-effective method for identifying possible drug leads against a target protein [10], [11] , [12], [13]. Virtual screening utilizes docking to simulate protein-ligand interaction to prioritize potential ligands for experimental validation. There exist standard docking protocols like DOCK [14], AutoDock [15], [16], GOLD [17] and FlexX [18] which predict if a ligand is a good binder and a potential drug lead for a given target protein.

Independent nature of these docking simulations allows for implementation of distributed protein docking, where a feasible number of these docking processes can be run simultaneously on a computing cluster. Currently, there a variety of software, including Docking@Home [19], DOVIS [20] and DockFlow [21], which automate the parallelization of virtual screening process to scale large chemical libraries. However, the selection of aforementioned *N* ligands is systematic or pre-defined in nature. Consequently, the entire chemical library needs to be docked in order to find the best binders which is time consuming for large chemical libraries.

To further reduce the time and cost of virtual screening, parallelization can be implemented using a mechanism to select potential binders from the remaining chemical library based on the docking results and the chemical nature of previously docked ligands. This set of potential binders needs a thoughtfully compiled sample of ligands for screening. These potential binders can then be queued for the docking process, preferentially over the others. Such an implementation eliminates the necessity of docking all the ligands in a chemical library, thereby, optimizing the virtual screening process.

The pharmaceutical industry heavily relies on Christopher Lipinski’s rule-of-five analysis for assessing if a compound is likely to be bioavailable [22], [23]. The rule establishes that certain compound properties (viz. molecular weight, lipophilicity, number of hydrogen bond donors and acceptors), if below threshold values, are highly correlated with a drugs having good bioavailability [24]. These properties are familiar to and routinely calculated by pharmaceutical researchers. We have, in our previous work, shown the utility of Lipinski properties as attributes in a neural net-based prediction of binding affinity, with an accuracy of 86 percent [25]. We, therefore, propose the use of Lipinski properties of ligands as attributes in a scoring function to predict the

binding energy of un-docked ligands much more quickly than via full Autodock calculations, which are not viable for large datasets like ZINC.

In this article, we present an application which performs supervised learning using binding energy of previously docked ligands and their similarity with un-docked ligands in terms of their Lipinski properties in a Nàive Bayes analysis, to make a prediction of binding energy of un-docked ligands. We present the performance of this application on four receptor proteins using an in-house library of 10,573 ligands which we have used in our previous docking studies [25], [26].

SECTION 2. METHODS

2.1 Setup

2.1.1 System Architecture

The application was originally developed and deployed on Marquette University's P *e*`re Cluster. The cluster is composed of 128 nodes, 2 × quad core Intel Xeon $X5550$, total of $1{,}024$ cores. The processors feature 24 GB RAM per node, DDR Inifiband backbone, 20 Gb/s, Red Hat Enterprise Linux 5.3.

2.1.2 Tools and Software

The scoring based virtual screening application was developed using several tools, modules, and a docking software. A list of all these resources is as follows:

HTCondor. It is a resource management and scheduling system for executing computation-intensive jobs harnessing idle compute power [27], [28], [29]. The Directed Acyclic Graph Manager (DAGMan) is a meta-scheduler for HTCondor jobs [30]. It is specially designed to provide a scheduling mechanism for jobs which have a dependency on each other. A DAGMan serves as a very handy tool in managing jobs that are components of a large workflow. In this work, HTCondor $7.4.4$ for X86_64-LINUX_RHEL5 is used a resource scheduler.

AutoDock. It is a widely used open source software for protein docking, which predicts how ligands bind to the pre-calculated docking area (grid) on the target protein [15] .These grids aid the physical description of the docking site and binding [16] . Lower predicted binding energy implies better ligand affinity. In this work, all dockings were performed using Autodock4.

MGLTools. The proteins and ligands used in this work were processed and using the MGLTools [31].

Python. Python $2.4.3$ was used as the programming language for code development [32].

2.2 Implementation

The HTCondor DAGMan is used to implement incremental docking using an X-DAG structure as shown in Fig. 1a. With this kind of dependency, jobs B, C, D will not start until job A is completed; job E will not start until jobs B, C, and D are completed and so on. Fig. 1b shows the submission file for such a DAG. Each job is defined by the keyword *JOB* and the relationship between jobs is defined by recursive usage of keywords *PARENT*and *CHILD* [33]. Each CHILD job in the X-DAG represents a ligand-protein docking simulation and each PARENT job represents the execution of the scoring function.

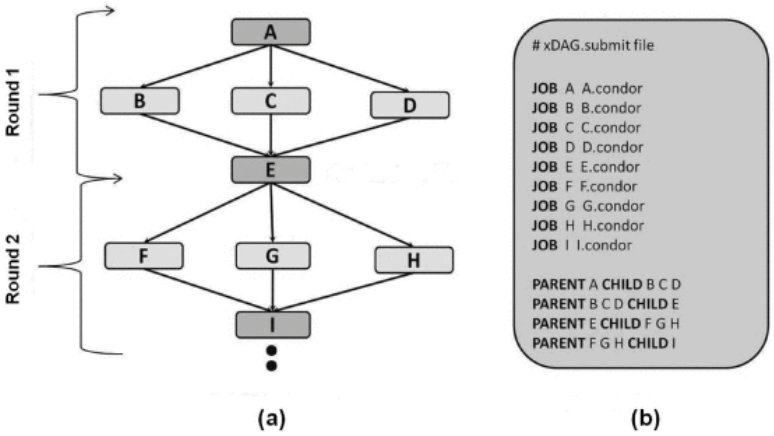


Fig. 1.
Implementation of X-Dag architecture: (a). The X-DAG workflow: Jobs B, C, D are launched when job A is completed. Job E is launched when jobs B, C, D are completed and so on, (b). A sample HTCondor DAGMan input file for a X-DAG. Each job is defined by the keyword *JOB*, jobs at the nodes are defined by the of keyword *PARENT*, and jobs at the leaves are defined by the keyword *CHILD*, illustrating the dependence relationship.

Fig. 2 shows the basic framework of our scoring based virtual screening application. The functionality can be broken down into three distinct steps:

1. *Data partitioning*. The application takes as input a data file (.tar format) which consists of the pre-prepared ligands, proteins and the supporting files in a ready to dock format. A dag submission file is created which contains the job definitions for determining the order in which the ligands are docked by the HTCondor *worker nodes. N*ligands are selected randomly for the first round of dockings to initiate the cycle.
2. *Job submission and control*. The result of each round of docking is sent back to HTCondor's *central manager*. At the end of each round of docking (i.e., at points A, E, I in Fig. 1a and so on) the scoring function predicts the next *N* best binders, which are then dynamically updated in the dag file and docked in the next round. This process is continued *K* times such that all the ligands are docked. Fig. 3 shows the implementation of the scoring function.
3. *Aggregation of results*. Once all the ligands from the given chemical library have been docked and results are obtained by the central manager, the last step in the X-DAG is a summarize step implemented in by HTCondor's *post script*. In this step, the ligands are sorted on the basis of their Autodock predicted binding energy and the final output file is created. This step is useful in deciding how many rounds of docking are essential to discriminate the potential binders from non-binders.

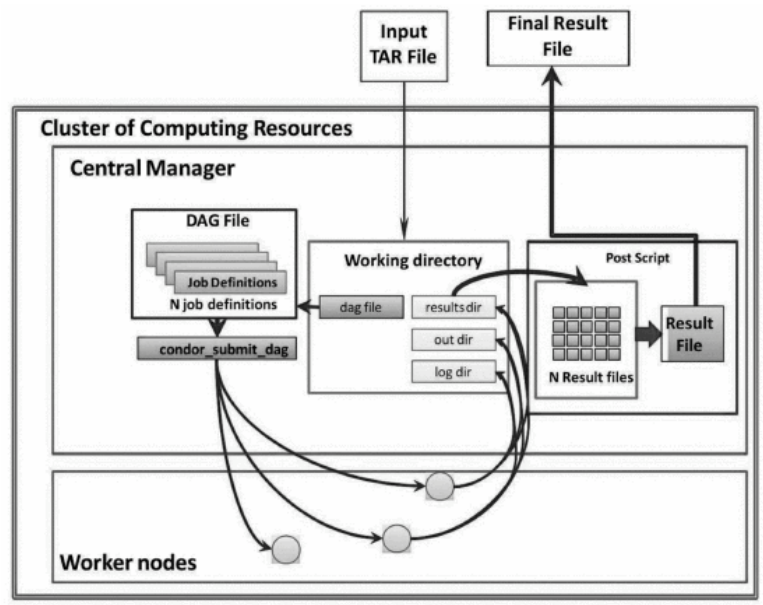


Fig. 2.
System architecture: Workflow of the scoring based distributed docking application. Multiple docking jobs are created and the dockings are implemented in an incremental fashion. The set of ligands for each round of docking is determined by the scoring function.

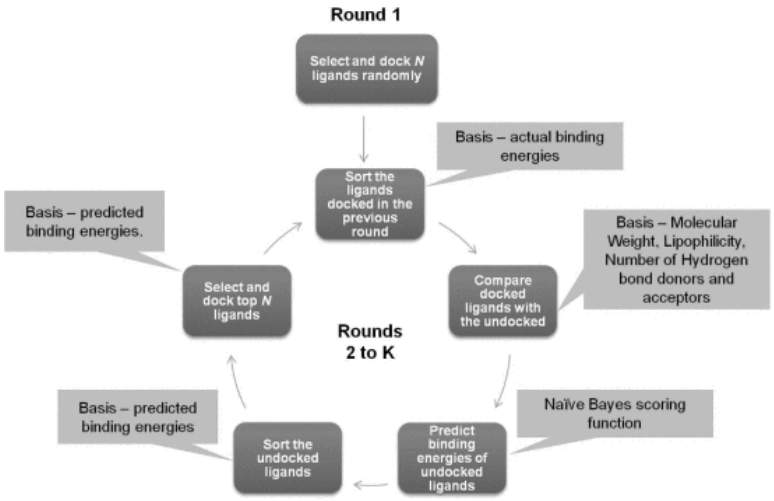


Fig. 3.
Implementation of Nȧive Bayes scoring function. *N*ligands are selected randomly for docking in round 1. The results of docking and the Lipinski properties of ligands are used to make a selection of the ligands to be docked in the next cycle. *K* rounds of docking are performed such that all the ligands in the library are docked for performance evaluation of the system.

2.3 Naive Bayes Scoring Function

Lipinski's Rule-of-Five establishes the importance of four different physico-chemical properties which are correlated with a chemical compound having good bioavailability viz. Molecular Weight, AlogP, Number of Hydrogen Acceptors and Number of Hydrogen Donors [22]. The expected Autodock predicted binding energy of a ligand is calculated based on the calculated binding energy of docked ligands and the Lipinski properties of ligands as attributes in a Nȧive Bayes analysis.

Each Lipinski property is grouped into ten equal sized bins based on the range of values of each property. For example, if the molecular weight of the ligands in the chemical library varies between 80 and 99 Daltons (range = 20), then the bins would be 80-81, 82-83, and so on up to 98-99. Each ligand in the chemical library is assigned to a bin for each property. So a ligand, L, can be represented as a vector of four property bins corresponding to each Lipinski property, l_i (i = 1-4).

Similarly, binding energy is grouped into five equal sized bins based on the range of binding energy values for the *N*ligands docked in each round. Each docked ligand is then assigned to one of the five energy bins, E_k (k = 1-5). This information along with the ligand property is then used to find the probability of an un-docked ligand (L) having binding energy in each of the five bins. The probability of a ligand having a binding energy can be calculated using the Bayes theorem:

$$P(E_k|L) \quad = \frac{P(L|E_k)P(E_k)}{P(L)}, \qquad (1)$$

where, $P(E_k|L)$ is the probability that the ligand will have a binding energy in bin E_k, $P(L|E_k)$ is the probability of an energy bin given a ligand $P(E_k)$ is the probability of any energy bin, and $P(L)$ is the probability of occurrence of any ligand. Assuming that each ligand is equally likely to occur in a chemical library, $P(L)$ is assigned a constant value. Also, $P(E)$ is the ratio of the number of ligands in each energy bin and the number of ligands docked so far (N), which is a constant. So, $P(L|E_k)$ can be re-written as:

$$P(L|E_k) \quad = \pi_{i=1}^{4} P_L(ld_i|E_k). \qquad (2)$$

Making the Nȧive assumptions of independence of ligand properties and representing the Lipinski bins for the docked ligands as $ld_i(i = 1 - 4)$, $P(L|E_k)$ can be written as a product of $P_L(ld_i|E_k)$, which is the probability of a ligand having a Lipinski property ld_i if it had a binding energy in energy bin E_k:

$$P(L|E_k) \quad = \pi_{i=1}^{4} P_L(ld_i|E_k). \qquad (3)$$

$P_L(ld_i|E_k)$ is the ratio of number of un-docked ligands with the same Lipinski property bin as the docked ligands with an energy bin E_k and the total number of ligands with an energy bin E_k:

$$P_L(ld_i|E_k) \quad = \frac{N_{(ld_i=l_i,E_k)}}{N_{E_k}}. \qquad (4)$$

If the number of ligands docked or the number of ligands in an energy bin is *zero*, the above probabilities are calculated by introducing a smoothing factor $\lambda = 0.1$ such that the new probabilities are:

$$P_L(ld_i|E_k) \quad = \frac{N_{(ld_i=l_i,E_k)}+\lambda}{N_{E_k}+4\lambda}, \qquad (5)$$

$$P(E_k) \quad = \frac{N_{E_k}+\lambda}{N+4\lambda}. \qquad (6)$$

Finally, the energy bin with the highest probability is the predicted binding energy (E_L) for each un-docked ligand:

$$E_L = \arg\max_{E_k} P(E_k|L). \quad (7)$$

Similarly, binding energy bins for all the un-docked ligands are estimated. *N* ligands with the lowest predicted energies are then selected for docking in the next round.

2.4 Datasets

Proteins. Four experiments on three distinct proteins were performed to evaluate the performance of application. The first set of proteins is Dihydrofolate Reductase (DHFR) [PDB:1DF7] and Dihydrodipicolinate reductase (DHPR) [PDB:1C3V] which are targets for the disease Tuberculosis [33], [34]. The other protein drug target is Human Dual Specificity Phosphatase 5 (DUSP5), an enzyme in humans encoded by the DUSP5 gene [35], [36]. DUSP5 protein has two domains and each of these domains participate in ligand binding. For this study, each of these domains were tested individually and are referred to as DUSP5C [PDB:2G6Z] and DUSP5R. The protein structure for DUSP5R was based on a homology model for a related mitogen-activated protein kinase phosphatase, MDP-3. The crystal structure of the proteins was obtained from the Protein Data Bank [37]. The details are included in the supplemental file 2, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TCBB.2015.2401020.

Chemical library. An in-house physical collection of $10{,}573$ chemical ligands in the Center for Structure-based Drug Design and Development (CSD3) was used [25], [26], [38]. The library contains drug-like molecules, selected on the basis of their predicted binding to dehydrogenases and kinases, a general compliance with the Lipinski Rule of Five, and other drug-like filters. The ligands were converted into a ready-to-dock format using Autodock tools [39]. The applicability and performance on of this chemical library has been demonstrated in our previous docking studies [25], [26]. The steps for the preparation of the files for the experiment are provided in supplemental file 1, available online.

2.5 Performance Metrics

Autodock computes the possible interaction points in the binding site of the protein and then docks each ligand to a protein target allowing the ligand to adopt many different conformations or *poses*. The docking simulation output consists of clusters of similar poses and the calculated binding energy for each docking pose within each cluster. For our experiments, we choose the cluster with the highest number of poses and then select the pose with the lowest predicted energy as the most favorable pose. The binding energy of this pose is used as the final predicted binding energy for a particular protein-ligand complex and is defined as the *binding energy of the ligand*. Lower binding energy of a protein-ligand complex is an indication of its binding affinity; lower the binding energy more stable the complex.

The objective of introducing the scoring function in the HPC framework is to enable the identification of better binders allowing for enrichment in early rounds of docking. To evaluate the performance of the application all the ligands were docked against the target proteins and 1,000 best binding ligands were identified for each protein based on their binding energy. Since the predictions are not binary in nature (i.e., a strong binder and not a strong binder), we do not measure performance in terms of ROC and AUC. We rank the ligands based on their predicted binded energy and measure the performance of the algorithm in terms of simpler and more commonly used metrics: average energy, ligand enrichment and cumulative ligand enrichment. Average energy is the average binding energy (as described above) of all the ligands docked in a round, ligand enrichment is the concentration of ligands with low binding energies in each round compared to their concentration throughout the docking cycle, and cumulative ligand enrichment is the concentration of ligands with low binding energies up to each round compared to their concentration throughout the docking cycle [9], [40], [41], [42]. We have evaluated the performance of the application based on enrichment observed for the 1,000 best binders in each round of incremental docking and is calculated as:

$$AverageEnergy = \frac{\sum E_i}{N_R}, \quad (8)$$

$$LigandEnrichment = N_{R1,000}, \quad (9)$$

$$CumulativeLigandEnrichment = N_{T1,000}, \quad (10)$$

where, E_i is the binding energy of the *i*th ligand docked in a round, N_R is the number of ligands docked in a round, $N_{R1,000}$ is the number of best 1,000 binders docked in a round and $N_{T1,000}$ is number of 1,000 best binders docked so far.

SECTION 3. RESULTS

500 ligands were docked in each round of incremental docking with a total of 22 rounds. If the ligands docked in each round were to be selected in a random fashion, it would be expected that each round is enriched with 50 out of 1,000 best binding ligands.

Fig. 4 shows the plot between the average energy of all ligands docked in a round versus round number. Average energy decreased substantially after the very first round of Bayesian selection of potential binders. For all four proteins (i). average energy in round 2 is the lowest across all rounds. (ii). average energy starts to increase after an initial decrease and attains a value higher than in round 1 towards the end of virtual screening experiments. These results demonstrate the earlier rounds are enriched with lower binding energy ligands (better binders) and later rounds have lesser number of potential binders.

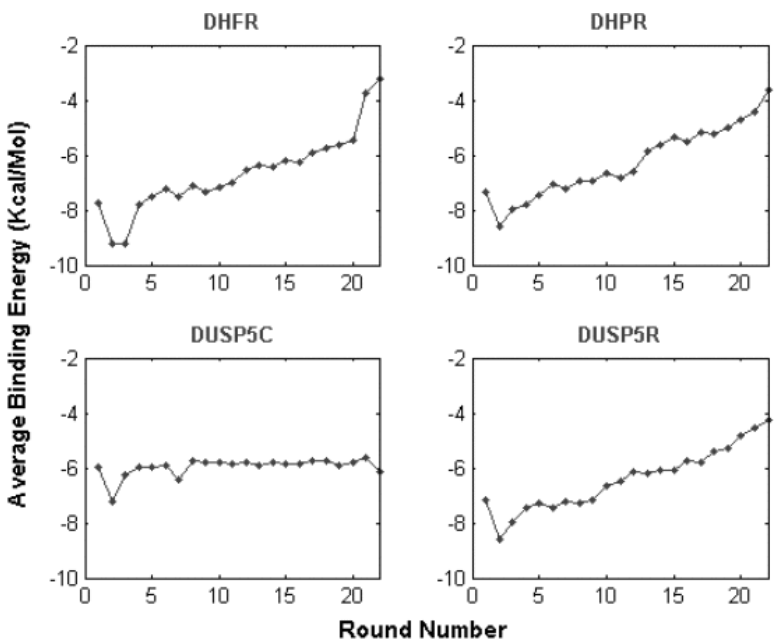


Fig. 4.
Plot between average binding energy of a round and round number. Lower binding energy indicates better protein-ligand complexes. Rounds 2 and 3 show a significant drop in average energy for all four proteins indicating significant enrichment in better binding ligands.

To verify that the binding actually occurs in the naturally occurring binding site, a visual representation of the protein-ligand complex is generated using Pymol [43]. Fig. 5 shows the predicted docking complex of the four proteins with the ligand with lowest binding energy (best binder) in the second round of docking. During docking

preparation, co-crystallized ligands were removed from the PDB coordinates allowing ligands to occupy the known binding sites. It is observed that the ligand (represented by stick model) is strategically placed in the natural substrate binding site of the protein molecules and as seen in the inset images. For DHFR and DHPR, the predicted inhibitors were predicted to bind in the $NADP^+$ pocket for each enzyme (see Supplemental Fig. 1, available online). The predicted inhibitor for DUSP5C was predicted to bind adjacent to the known active nucleophile for the protein, Cys-263 (see Supplemental Fig. 2). This structure is structurally similar to a group of molecules recently published [44]. As DUSP5R is based on a homology model, further investigation to probe the binding site of this regulatory domain is required.

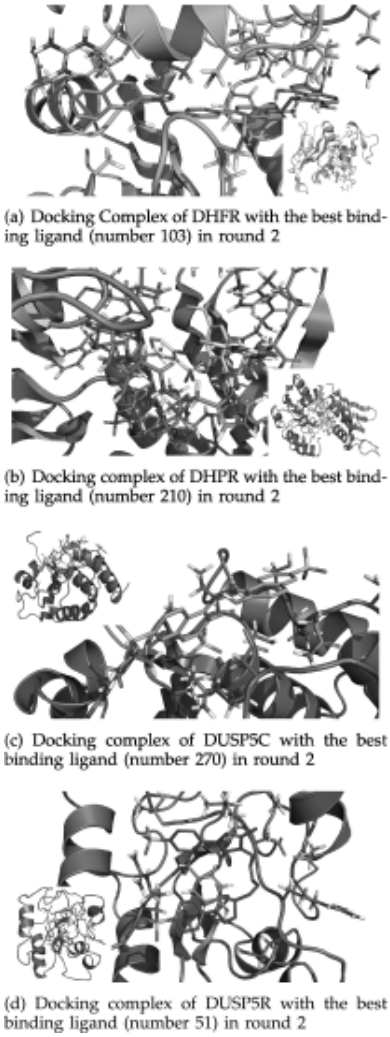
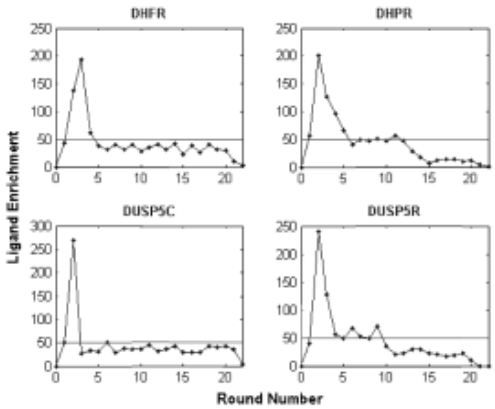


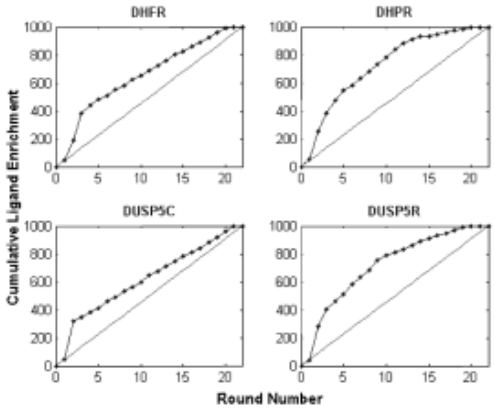
Fig. 5.
Focused image of the protein-ligand complex at the binding site. The inset shows full image of the protein-ligand complex.

Figs. 6a and 6b show ligand enrichment and cumulative ligand enrichment for each round, respectively. A sharp spike at rounds 2 and 3 in the ligand enrichment plot shows the effectiveness of the scoring function in identifying potential binders. The steep slope of the curve in the cumulative ligand enrichment plot between rounds 2 and 3 is an indicative of higher rate of enrichment in earlier rounds. For all proteins, (i). 200 of 1,000 best binding ligands are identified by docking only 14 percent of the chemical library, (ii). 500 of 1,000 best binding ligands are identified by docking only 28 percent of the chemical library, (iii). there is no significant gain

in enrichment after docking almost 70 percent of the chemical library, and (iv). nine or 10 best binding ligands are identified within docking 19 percent of the chemical library.



(a) Ligand enrichment observed in the 1000 best binders versus round number (blue). The horizontal line (red) depicts the ligand enrichment expected by a random selection.



(b) Cumulative enrichment of ligands with progressing rounds with Bayesian selection (blue) is much higher than the enrichment expected by a random selection (red).

Fig. 6.
Enrichment observed in terms of the 1,000 best binders in each round of incremental docking.

In all the four experiments, 1,000 best binders were docked in rounds 2 and 3 with p-value < 0.001. The results suggest that virtual screening for any protein can be considered complete after round 5 by docking only 30 percent of the chemical library. These results demonstrate that our approach has a selective preference for better binding ligands and provides better enrichment as compared to a completely random parallel virtual screening application. Additionally, it offers time and cost benefits by reducing the need to dock the entire chemical library to identify potential Autodock predicted drug leads.

SECTION 4. CONCLUSION

Virtual screening is a computational method to identify potential drug leads from a large chemical library. Ligand enrichment, thus, forms the essence of virtual screening. High performance computing clusters strengthen the capabilities of virtual screening process by further gain in time and cost. However, the growing size of available chemical libraries and the aspiration for exhaustive search for a potential drug from the entire virtual chemical pool necessitates a new methodology to allow for faster discrimination of binders from the non-binders.

We present an optimization to the virtual screening workflow allowing for large throughput of results in smaller time scale. We have implemented a Nàive Bayes scoring function which performs supervised learning using Lipinski properties and binding energy of ligands for a given target protein to predict the binding energy of unknown ligands. The application harnesses HTCondor's capability as a resource management system to automatically schedule parallel and distributed protein dockings. The results of using this application to isolate binders for four target proteins suggest that potential drug leads can be isolated by examining not more than 30 percent chemicals in a large chemical, saving the need to investigate the entire chemical library.

The application is compatible with and can be deployed on different computing clusters with slight or no modification. We have tested the performance of the application after porting and integration with other available grids like BOINC [45] and TeraGrid [46] . The application can also be implemented on a commercial cloud. We have successfully migrated it to the Amazon EC2 cloud to assess the feasibility of such an implementation. A detailed performance analysis and comparison was also done to validate against local high performance computing resources. It was found that the application can be implemented on the cloud as there is no overhead required to set up an in-house cluster or grid, software requirements are inexpensive or free, and computing time is rapid based on the number of resources purchased on the cloud.

However, the performance of this framework at the level of ZINC is still to be tested. It remains to be seen that enrichment and scaling obtained in this study can be maintained at the size of ZINC. Nonetheless, the early results are promising and further investigations to see if this approach scales appropriation are needed.

ACKNOWLEDGMENTS

The experiments were performed on the P ère cluster funded by National Science Foundation awards OCI-0923037 "MRI: Acquisition of a Parallel Computing Cluster and Storage for the Marquette University Grid (MUGrid)" and CBET-0521602 "Acquisition of a Linux Cluster to Support College-Wide Research & Teaching Activities." Daniel S. Sem is partly supported by NIH grants AI101975 and HL112639. Prachi Pradeep is the corresponding author.

SUPPLEMENTARY MATERIALS

Supplement 1: Steps for preparation of input files

1. STEPS FOR PREPARING THE LIGAND MOLECULE

(i) Load the ligand into the viewer (Ligand > Input> Open Ligand.pdb).
(ii) Determine which atom fits its idea of the best root (Ligand > Torsion Tree > Detect Root).
(iii) Determine which rotatable bonds to be active (Ligand > Torsion Tree > Choose Torsions).
(iv) Set the numbers of bonds to be active (Ligand > Torsion Tree > Set Number of Torsions).
(v) Save the prepared Ligand (Ligand > Output > Save as Ligand.pdbqt).
(vi) Remove from viewer (Edit > Delete Molecule).

2. STEPS FOR PREPARING THE PROTEIN MOLECULE

(i) Edit Protein
    a. Open the Protein structure file (File > Read Molecule .pdb).
    b. Color atoms by chemical element (Color > By Atom Type > All Geometries ¿ OK).
    c. Add Polar Hydrogen Atoms, choose default (Edit > Hydrogens > Add > Select All Hydrogens, no Bond Order, yes > OK).
    d. Save the Macromolecule (File > Save > Write PDB).
    e. Delete all molecules.
(ii) Prepare Macromolecule File

a. Add the charges and solvation parameters, choose defaults (Grid > Macromolecule > Open .pdb file saved in saved in 1.e).
b. Save the file (Save as .pdbqt file).

(iii) Prepare the Grid Parameter File
a. Specify the type of Maps used, Add A, Br, C, Cl, F, H, HD, I , N, NA, OA, P, SA, S (Grid > Set Map Types > Open Ligand).
b. Select the grid for Docking (Grid > Grid Box > Choose to center on ligand or atom, Make the grid box bigger to encapsulate the entire ligand > Delete Molecule > Close saving current).
c. Save the Grid Parameter File (Grid > Output > Save GPF).
d. Start Autogrid4 in the directory where .gpf file is located (autogrid4 p grid.gpf l grid.glg &).
e. Delete Molecule from ADT.

(iv) Prepare Docking Parameter File (dpf): This step is to be done for all the ligands in the ligand library. In this work, this step is scripted to prepare the dpf file for all ligands at once.
a. Open the edited Macromolecule (Docking > Macromolecule > Set Rigid Filename).
b. Select the current Ligand (Docking > Ligand > Choose Ligand).
c. Select Docking Algorithm, choose 50-100 GA Runs (Docking > Search Parameters > Genetic Algorithm).
d. Select Default docking parameters (Docking > Docking Parameters).
e. Choose the Algorithm (Docking > Output > Lamarckian GA).
f. Save as .dpf

3. COMBINING THE INPUT DATA TO CREATE THE INPUT TAR FILE

(i) Copy all the prepared ligands and the protein in one directory.
(ii) Prepare the dpf file for all the ligands by scripting step 4 in protein preparation.
(iii) Copy all the map files to be used for docking.
(iv) Tar the directory.
(v) Repeat these steps for all the proteins used.

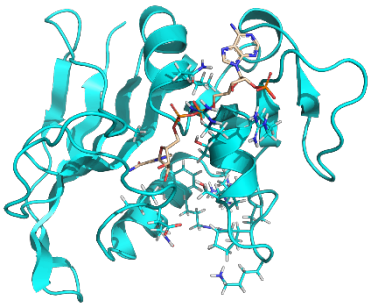
Supplement 2: Protein Structure and Supplemental Figures

1 Protein structure source

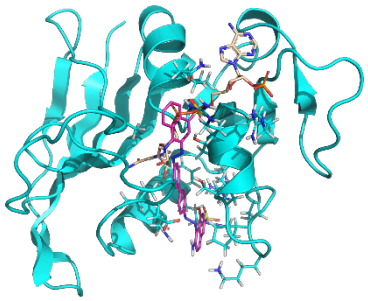
DHFR with NADP bound: http://www.rcsb.org/pdb/explore/explore.do?structureId=1df7
DHPR with NADP bound: http://www.rcsb.org/pdb/explore/explore.do?structureId=1C3V
DUSP5C: http://www.rcsb.org/pdb/explore/explore.do?structureId=2g6z
DUSP5R: Homology model based on 1HZM (25

Website link last assessed: 10/14/2014

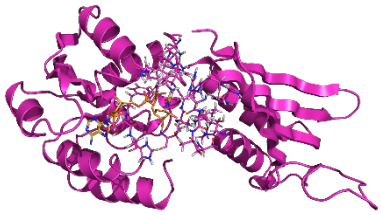
2 Supplemental Figures



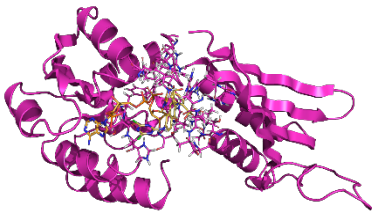
(a) Co-crystallized DHFR with natural ligand NDP^+



(b) DHFR with natural ligand and superimposed docked ligand



(c) Co-crystallized DHPR with natural ligand NDP_+



(d) DHPR with natural ligand and superimposed docked ligand

Figure 1: Natural co-crystallized ligands and the overlayed image of the best docked ligand in round 2 of incremental docking.

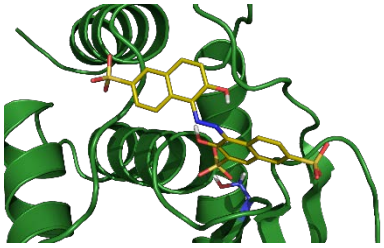


Figure 2: Natural co-crystallized DUSP5C-nucleophile complex

REFERENCES

1. B. D. Anson, J. Ma, J.-Q. He, "Identifying cardiotoxic compounds", *Genetic Eng. Biotechnol. News*, vol. 29, pp. 34-35, 2009.
2. S. M. Paul, D. S. Mytelka, C. T. Dunwiddie, C. C. Persinger, B. H. Munos, S. R. Lindborg, A. L. Schacht, "How to improve R&D productivity: The pharmaceutical industry's grand challenge", *Nature Rev. Drug Discovery*, vol. 9, pp. 203-214, 2010.
3. Zinc homepage [Online]. Available: http://zinc.docking.org/, 2011.
4. J. J. Irwin, B. K. Shoichet, "Zinc—A free database of commercially available compounds for virtual screening", *J. Chemical Inform. Model.*, vol. 45, no. 1, pp. 177-182, 2005.

5. B. Waszkowycz, D. J. Perkins, R. A. Sykes, J. Li, "Large-scale virtual screening for discovering leads in the postgenomic era", *IBM Syst. J.—Deep Comput. Life Sci.*, vol. 40, pp. 360-376, 2001.
6. J. Alvarez, B. Shoichet, J. Alvarez, B. Shoichet, Virtual Screening in Drug Discovery, Boca Raton, FL, USA:CRC Press, 2005.
7. C. E. P. Andrew, R. Leach, B. K. Shoichet, "Docking and scoring", *J. Med. Chemistry*, vol. 49, no. 20, 2006.
8. J. D. J. Blaney, "A good ligand is hard to find: Automated docking methods", *Perspect. Drug Disc. Des.*, vol. 1, pp. 301-319, 1993.
9. A. R. L.each, B. K. Shoichet, C. E. Peishof, "Prediction of protein-ligand interactions docking and scoring: Successes and gaps", *J. Med. Chemistry*, vol. 49, pp. 5851-5855, 2006.
10. W. Walters, M. Stahl, M. Murcko, "Virtual screening—An overview", *Drug Discovery Today*, vol. 3, pp. 160-178, 1998.
11. A. S. Reddy, S. P. Pati, P. P. Kumar, H. Pradeep, G. N. Sastry, "Virtual screening in drug discovery—A computational perspective", *Current Protein Peptide Sci.*, vol. 8, pp. 329-351, 2007.
12. I. Muegge, S. Oloff, "Advances in virtual screening.", *Drug Discovery Today: Technol.*, vol. 3, pp. 405-411, 2006.
13. B. K. Shoichet, "Virtual screening of chemical libraries", *Nature*, vol. 432, pp. 862-865, 2004.
14. I. D. Kuntz, J. M. Blaney, S. J. Oatley, R. Langridge, T. E. Ferrin, "A geometric approach to macromolecule-ligand interactions", *J. Molecular Biol.*, vol. 161, pp. 269-288, 1982.
15. Autodock homepage [Online]. Available: http://autodock.scripps.edu/, 2011.
16. G. M. Morris, D. S. Goodsell, R. S. Halliday, R. Huey, W. E. Hart, R. K. Belew, A. J. Olson, "Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function.", *J. Comput. Chemistry*, vol. 19, pp. 1639-1662, 1998.
17. G. Jones, P. Willett, R. C. Glen, "Molecular recognition of receptor sites using a genetic algorithm with a description of desolvation", *J. Molecular Biol.*, vol. 245, pp. 43-53, 1995.
18. M. Rarey, B. Kramer, T. Lengauer, G. Klebe, "A fast flexible docking method using an incremental construction algorithm.", *J. Molecular Biol.*, vol. 261, pp. 470-489, 1996.
19. Docking@home [Online]. Available: http://docking.cis.udel.edu/, 2011.
20. S. Zhang, K. Kumar, X. Jiang, A. Wallqvist, J. Reifman, "DOVIS: An implementation for high-throughput virtual screening using autodock", *BMC Bioinform.*, vol. 9, pp. 126, 2008.
21. N. Azam, M. Ghanem, D. Kalaitzopoulos, A. Wolf, V. Kasam, Y. Wang, M. Hofmann-Apitius, "Dockflow: Achieving interoperability of protein docking tools across heterogeneous grid middleware", *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 6, pp. 235-251, 2010.
22. C. Lipinski, F. Lombardo, B. Dominy, P. Feeney, "Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings", *Adv. Drug Del. Rev.*, vol. 23, pp. 3-25, 1997.
23. C. A. Lipinski, "Lead- and drug-like compounds: The rule-of-five revolution", *Drug Discovery Today: Technol.*, vol. 1, no. 4, pp. 337-341, 2004.
24. A. K. Ghose, V. N. Viswanadhan, and J. J. Wendalaski, "A knowledge-based approach in designing combinatorial or medicinal chemistry libraries for drug discovery. 1. a qualitative and quantitative characterization of known drug databases", *J Comb Chem.*, vol. 1, no. 1, pp. 55-68, 1999.
25. P. S. Bazeley, S. Prithivi, C. A. Struble, R. J. Povinelli, D. S. Sem, "Synergistic use of compound properties and docking scores in neural network modeling of CYP2D6 binding: Predicting affinity and conformational sampling", *J. Chemical Inform. Model.*, vol. 46, no. 6, pp. 2698-2708, 2006.
26. P. Boonsri, T. S. Neumann, A. L. Olson, S. Cai, T. J. Herdendorf, H. M. Miziorko, S. Hannongbua, D. S. Sem, "Molecular docking and NMR binding studies to identify novel inhibitors of human phosphomevalonate kinase", *Biochemical Biophysical Research Commun.*, vol. 430, no. 1, pp. 313-319, 2013.
27. "Condor—A hunter of idle workstations", *Proc. IEEE Conf. Distrib. Comput. Syst.*, pp. 104-111, 1988.
28. Condor project [Online]. Available: http://research.cs.wisc.edu/htcon-dor/, 2011.
29. D. Thain, T. Tannenbaum, M. Livny, "Distributed computing in practice: The condor experience", *Concurrency Comput.: Practice Experience*, vol. 17, no. 2–4, pp. 323-356, 2005.

30. DAGMan applications [Online]. Available: http://research.cs.wisc.edu/htcondor/dagman/dagman.html, 2011.
31. MGLTools website [Online]. Available: http://mgltools.scripps.edu/,2011
32. Python software foundation. python language reference, version 2.4.3[Online]. Available: http://www.python.org, 2011.
33. X-Dag [Online]. Available: http://www.cs.wisc.edu/condor/manual/v7.4/2_10DAGMan_Applications.html, 2011.
34. Tuberculosis [Online]. Available: http://en.wikipedia.org/wiki/-Tuberculosis, 2011.
35. Dual specificity protein phosphatase 5 [Online]. Available: http://en.wikipedia.org/wiki/DUSP5, 2011.
36. S. P. Kwak, J. E. Dixon, "Multiple dual specificity protein tyrosine phosphatases are expressed and regulated differentially in liver cell lines", *J. Biol. Chemistry*, vol. 270, 1995.
37. F. C. Bernstein, T. F. Koetzle, G. J. Williams, E. F. Meyer, M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, M. Tasumi, "The protein data bank", *Eur. J. Biochemistry*, vol. 80, no. 2, pp. 319-324, 1977.
38. Center for structure-based drug design and development [Online]. Avail-able: http://www.csddd.org/, 2011.
39. M. F. Sanner, "Python: A programming language for software integration and development", *J. Mol. Graphics Mod.*, vol. 17, pp. 57-61, 1999.
40. G. Klebe, "Virtual ligand screening: Strategies perspectives and limitations", *Drug Discovery Today*, vol. 11, no. 13, pp. 580-594, 2006.
41. E. Kellenberger, J. Rodrigo, P. Muller, D. Rognan, "Comparative evaluation of eight docking tools for docking and virtual screening accuracy", *Proteins: Struct. Function Bioinform.*, vol. 57, no. 2, pp. 225-242, 2004.
42. M. D. Cummings, R. L. DesJarlais, A. C. Gibbs, V. Mohan, E. P. Jaeger, "Comparison of automated docking programs as virtual screening tools", *J. Med. Chemistry*, vol. 48, no. 4, pp. 962-976, 2005.
43. The pymol molecular graphics system, version 1.2r3pre, Schrddinger, llc[Online]. Available: http://www.pymol.org/, 2011
44. T. Neumann, E. Span, K. Kalous, A. Gastonguay, R. Kutty, J. Nayak, C. Bohl, R. Lange, M. Sarker, M. Talipov, R. Rathore, R. Ramchandran, D. Sem, "Identification of polysulfonated inhibitors related to suramin that target dual specificity phosphatase 5 and provide new insights into the binding requirements for dual-phosphate substrate pockets", *Proteins: Struct. Funct. Bioinf.*.
45. D. P. Anderson, "Boinc: A system for public-resource computing and storage", *Proc. 5th IEEE/ACM Int. Workshop Grid Comput.*, pp. 4-10.
46. C. Catlett, L. Grandinetti, TeraGrid: Analysis of Organization System Architecture and Middleware Enabling New Types of Applications HPC Grids Action, Amsterdam, The Netherlands:IOS Press, 2007.

KEYWORDS

IEEE Keywords

Proteins , Chemicals , Drugs , Computational biology , Bioinformatics

INSPEC: Controlled Indexing

Bayes methods , binding energy , biology computing , molecular biophysics , proteins

INSPEC: Non-Controlled Indexing

Naive Bayes scoring function , high performance computing cluster , chemical libraries , virtual screening , ligand, binding energy , molecular docking , enrichment , scoring based distributed protein docking

Author Keywords

Virtual screening , high performance computing , distributed protein docking , HTCondor , Nàive Bayes , Scoring function

MeSH Terms

Algorithms , Bayes Theorem , Binding Sites , Computer Simulation , Models, Chemical , Molecular Docking Simulation , Protein Binding , Proteins Virtual screening , high performance computing , distributed protein docking , HTCondor , Näive Bayes , Scoring function