Selecting Molecular Docking Sites by Neighbor Selection and Various Factors

Chen-En Hsieh*[1], Shiahuy Chen†[1], Pei-Sheng Hsu‡[2], Chun-Jung Chen§[3], and Yaw-Ling Lin¶ [1,2]

[1]Department of Applied Chemistry, Providence University, Taichung, Taiwan
[2]Dept of Computer Science and Information Engineering, Providence University
[3]Department of Medical Education and Research, Taichung Veterans General Hospital, Taichung, Taiwan

Abstract

Methods for finding molecular sites in molecular docking simulation is proposed in the paper. The method distinguishes the surface/inside atoms of the receptor by selecting a suitable distance maximizing the standard deviation of corresponding neighboring degrees of the molecules. With various considerations and different set ups of the underlying parametric spaces, the searching space for the docking simulation problem can be significantly reduced.

The method is implemented upon the widely employed automated molecular docking simulation software package, AutoDock. Experiments are set up to test upon Japanese encephalitis related biomolecules in virology research. In average, the proposed k-gridbox algorithm is about 2.3 flods faster. Hadoop MapReduce frameworks are used in our experiments to parallelize the underlying massive computation works corresponding to ligand-receptor pairs examined under the experiment. The experiment shows that the proposed method is much more efficient comparing to the general parametric set ups.

Keywords: bioinformatics, algorithm, molecular docking, drug design, AutoDock, Hadoop, MapReduce.

*g1026011@pu.edu.tw
†grace@pu.edu.tw
‡g1010541@pu.edu.tw
§cjchen@vghtc.gov.tw
¶Corresponding author. yllin@pu.edu.tw. This work is partly supported by grants from the Taichung Veterans General Hospital and Providence University (TCVGH-PU1038104) and by the National Science Council (NSC-99-2632-E-126-001-MY) Taiwan, Republic of China.

1 Introduction

The large number of structural investigations on medically relevant proteins [1] reflects the general recognition that the structure of a potential drug target is very precious knowledge; however, designing a new drug poses a great challenge. Computer-aided drug design techniques, especially the molecular docking simulation, can now be effective in reducing costs and speeding up drug discovery [2, 7].

Progress in functional genomics and structural studies on biological macromolecules are producing more and more potential therapeutic targets, but also increases the importance of small molecule docking and virtual screening of candidate compounds algorithms. [24, 3]. Usually, the first step in the molecular docking is to find the position of the space and the conformation matched. In molecular docking, the receptor is possibly a biological protein or biomolecule, and the ligand is possibly a different protein, medicine or compound. Molecular docking simulation is often used as a method for virtual screen by setting a protein to match a group of compounds, and report the final best compound [2].

Protein structures play critical roles in vital biological functions [16]. To date, there are more than 98,300 protein structures [1] determined by the advances in X-ray crystallography and NMR spectroscopy to date, molecular biologists these days proceed in the direction of analyzing and classifying these protein structures in order to discover the interaction with ligand and receptors.

In 1894s, Fischer [14] proposed “lock and key” model. The enzyme and the substrate has specific geometric shapes that will fit exactly into one an-other. This mode explains enzyme specificity, but fails to explain the transition of the enzyme. In 1958s, Koshland [25] proposed molecular recognition process concept of induced fit, when receptor and ligand combined with each other, receptor will selected an optimal binding conformation with ligand.

Molecular docking simulation is a method for computer-aided drug design (CADD). It simulate the interaction between a protein receptor and a drug ligand by calculating the energy of interaction between them, and then search the optimal binding sites in most stable state.

There have been several Public domain packages proposed in molecular docking simulation, include AutoDock [18], DOCK [26], Flex [9], Glide [15], GOLD [37], RosettaDock [11], SLIDE [39], Surflex [22] and AutoDock Vina [36]. Some analysis and visualization methods of molecular docking are importance step for Evaluation results, include AutoDockTool [24], DockingServer [3], POLYVIEW-MM [33], ViewDock [32], vsLab [2, 7].

Blind docking is a docking strategy when the binding site is unknown, it is necessary to set up big gridbox to put entire macromolecule. Rigid-body docking does not change the shape of the ligand and receptor; it only change the position and rotate angle, spending much less compute time then flexible docking. Flexible docking permit conformation change, it is more accurate to observation the interaction of the ligand and receptor. However, Flexible docking cost huge time for computation.

Hadoop [38] is a software framework intended to support data-intensive distributed applications. It is able to process petabytes of data with thousands of nodes. Hadoop supports MapReduce programming model [35] for writing applications that process large data set in parallel on Cloud Computing environment. Recently, Hadoop has been applied in various domains in bioinformatics [34, 12].

2 Method and Materials

2.1 Molecular Docking Simulation

The main idea of molecular docking algorithm for finding a lowest energy (LE) between two sets of points before utilizing the scoring function procedure to fine-tune the final result is by adjusting the suitable parameter sets by ways of searching the underlying parametric space.

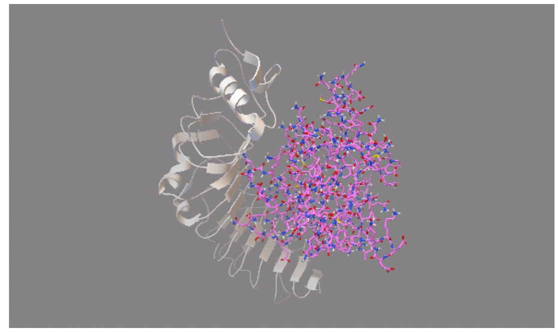

Figure 1: An illustration of the docking result of ligand, IL-1β, and receptor, TLR4, (with PDB ID: 6I1B and 2Z62) by AutoDockTool [24].

AutoDock

AutoDock is a automated molecular docking simulation software package since 1990. AutoDock uses Genetic Algorithm (GA) [17, 20], Lamarckian Genetic Algorithm (LGA) [31] in finding the lowest energy, and the used the Amber molecular force field scoring function [13]. AutoDock is now version 4.2, include two program with autodock and autogrid. Autogrid pre-calculates a set of grids which describing the target protein and autodock performs the docking of the ligand to these grids. Figure 1 shows the example of docking result. The force field evaluation function of AutoDock4.2 is described as the following: [30, 21]

$$V = W_{vdw} \sum_{i,j} \left(\frac{A_{ij}}{r^{12}} - \frac{B_{ij}}{r^6} \right) + W_{\text{elec}} \sum_{i,j} \frac{q_i q_j}{\varepsilon(r_{ij}) r_{ij}} + W_{\text{hbond}} \sum_{i,j} E(t) \left(\frac{C_{ij}}{r^{12}} - \frac{D_{ij}}{r^{10}} \right) + W_{\text{sol}} \sum_{i,j} (S_i V_j + S_j V_i) e^{\left(\frac{-r_{ij}^2}{2\sigma^2}\right)}$$

AMBER force field

In molecular docking simulation, molecular force field is an important part to evaluate the docking result. The traditional force field include AMBER force field [13, 21], CHARMM force field [6, 5, 29] and MMFF94 [19].

AMBER force field is one the most widely used force field functions, which is suitable for the treatment of biological macromolecules. Many mainstream molecular modeling software use the AMBER force field, and AutoDock use AMBER force field as prototype of the force field function. The AMBER force field function is described as the following: [13, 10]

$$E_{\text{total}} = \sum_{\text{bonds}} k_b(b - b_{eq})^2 + \sum_{\text{angle}} k_\theta(\theta - \theta_{eq})^2 + \sum_{\text{dihedrals}} \frac{V_n}{2}[1 + \cos(n\phi - \gamma)] + \sum_{i<j} \left[\frac{A_{ij}}{R_{ij}^{12}} - \frac{B_{ij}}{R_{ij}^{6}} + \frac{q_i q_j}{\varepsilon R_{ij}} \right]$$

The first term is the bonds (bond strength), the interaction between the bond atoms. The second term is angle (bond angle), the total value of all bond angle of the atoms. The third term is dihedrals, the energy change when bond rotation. The forth term is non-bonded interaction (non-bonded), the interaction of all atom of the bond distance at least three atom of both molecular; use the Coulomb potential [13] to describe electrostatic interactions, and using Lennard-Jones potential [28] describe van der Waals forces

GA

Genetic algorithm (GA) [17, 20] is a search heuristic that mimics the process of natural selection. This heuristic (also sometimes called a metaheuristic) is routinely used to generate useful solutions to optimization and search problems. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover.

LGA

LGA [31] use to search the best binding site with ligand and receptor. LGA is an algorithm which is compose by GA and Local search(LS). GA is responsible for global search and then optimization of energy through LS.

Rigid-body Docking

In the rigid-body docking, the molecular conformation does not change. Only change in the spatial position and rotate angle of the molecule. rigid-bosy docking is the most simplification method and the computation is relatively small. Useful on macromolecule docking.

Semi-flexible Docking

In the semi-flexible docking, allow the small molecule change their conformation in the docking process, but it is usally fixed the conformation of the macromolecule. The small molecule usually fixed bond lengths, bond angles of some non-critical parts. Semi-flexible docking method consider the computation and the predictive ability of the model, is one of the widely used docking method.

Flexible Docking

In the flexible docking, allow both molecules (ligand-receptor) change their conformation freely in the docking process, the computation growing exponentially due to the atom numbers. It is huge computation in the flexible docking. Flexible docking is useful in accurate identification the interaction of molecular docking and the consideration is the huge computation time.

Protein-Protein Docking

In the protein-protein docking, ligand and receptor are both macromolecule. It is really hard to perform a flexible docking on protein-protein docking. The rigid-body docking is suitable for the protein-protein docking.

2.2 Global-grid method

Given the requested ligand-receptor pair, the idea is to consider the *whole space* outside the receptor in order to obtain a reasonable final docking position for placement of the corresponding ligand. Thus, the general method Global-grid, is to build a global grid frames outside the given receptor. That is, Global-grid method builds a large enough box containing receptor and leave enough space for further possible ligand movement. The Global-grid method is to make sure user obtaining the full examined result. See Figure 2 for an illustration of the Global-grid algorithm.

2.3 k-gridbox algorithm

Here we propose method that reduces the searching space for the docking simulation process by finding subspaces of the surface of the given receptor. The idea is to distinguish the surface/inside atoms of the receptor, and identifying *hot spots* relevant to the given ligand. Please refer to Figure 3 for an illustration of the k-gridbox

GLOBAL-GRID(L, R, r, p, s)

Input: L, R is the ligand and receptor.
r : The resolution of these grids
p : The probe number
s : The magnification factor of the ligand size
Output: pos : The resulting position (conformation) of the ligand
E : The lowest energy of the result

Let box $b = (s \cdot x_L + x_R, s \cdot y_L + y_R, s \cdot z_L + z_R)$, where (x_L, y_L, z_L), (x_R, y_R, z_R) being bonding box of L and R
Execute `preparegpf.py` to prepare the grid parameter file `.gpf` file by parameters (L, R, r, p, b)
Execute `preparedpf.py` to prepare the docking parameter file `.dpf` file by parameters (L, R)
Execute `autogrid` on the `.gpf` ▷ the output of `autogrid` is the `.fld`, `.map` and `.xyz` files.
Execute `autodock` on the `.dpf` ▷ the output of `autodock` is the reslut file `.dlg`.
Obtain the best (pos, E) from the `.dlg` file, and **return** (pos, E)

Figure 2: The GLOBAL-GRID method.

algorithm. The idea is to selecting a suitable distance maximizing the standard deviation of corresponding neighboring degrees of the molecules. With various considerations and different set ups of the underlying parametric spaces, the searching space for the docking simulation problem can be significantly reduced.

Detecting surface atoms

The idea is to partition atoms of a receptor into *surface points* and *inside points* so that it is possible to intelligently place the corresponding ligand into suitable places nearing to surfaces of a receptor. The inside atoms are those *crowded* atoms with sufficient number of neighboring atoms, while surface atoms are those having fewer neighboring atoms. However, the neighboring relation is defined by a suitable distance. Setting the cut-off distance by an extremely small meaning that every atom is isolated, while an overly large distance resulting every atom pair being neighbor to each other. The trick is to pick the *right* distance that produces the most *informative* neighboring numbers.

Let A be the set of atoms of a receptor. Given a distance $d \in R^+$ and $x \in A$, denote the *neighbor number* of x by $N_d(x) = |\{y \mid |x - y| \leq d, y \in A\}|$. The *neighbor number list* of A is denoted by $G(d, A) = \langle N_d(x) | x \in A \rangle$, and let $\sigma(G(d, A))$ be the standard deviation of $G(d, A)$.

Here we consider the distance maximizing the standard deviation of corresponding neighboring degrees of molecules of the receptor. Let d^* be the distance such that

$$d^* = \underset{d}{\operatorname{argmax}}\{\sigma(G(d, A))\};$$

then we can use d^* to obtain neighbor number list $G(d^*, A)$; the list is used to to distinguish whether an atom is a surface or inside atom.

We use standard deviation (σ) to distinguish the surface/inside atoms of a given receptor, σ shows the dispersion or variation from the average exists [4]. A low σ indicates these values are similar and a high σ indicates these values are less similar.

Once the distance d^* is set, the neighbor number for each atom is decided. An arbitrary percentile value, 10%, is set to be the cut-off value for deciding the inside atoms. That is, x is an inside atom if and only if $N_{d^*}(x)$ is ranked among the top (highest) 10% among the neighbor number list $G(d^*, A)$. Let the set of surface (inside) atoms be S (D); we define '$(S, D) \leftarrow$ GETSURFACE(R)', as depicted in in Figure 3.

Gridbox placement

The k-GRIDBOX algorithm is to find a suitable docking site nearing to a surface atom $x \in S$ and build its corresponding gridbox. First step, use the ligand box multiply β; β' : $(\beta x_L, \beta y_L, \beta z_L)$ to choose the atom s_i which has the most neighbor s_j atoms in the box β', and then delete the (s_i, s_j) atoms for next box. Second step, use top ten percent of D atoms who is most close to s_i atom to push out the point by α value (Figure 4) and set as the gridcenter. Do the first and second step k times to get k boxes. Last step, set a gridbox g on these gridcenter and start preparing these gridcenter by `preparegpf.py` and then start `autogrid` and `autodock`.

3 Experiments

In order to tuning the best parameters (k, β, g, α) (Figure 4), we use the MapReduce

k-GRIDBOX$(L, R, r, k, \beta, g, \alpha, p)$

Input: L, R is the ligand and receptor.
r : The resolution of these grids
k : The limit of the box generate
β : The magnification factor of the ligand size use to find the maximun degree candidate
g : The magnification factor of the ligand size use to build the grid box
α : The distance factor use to push the grid box
p : The probe number

Output: pos : The resulting position (conformation) of the ligand
E : The lowest energy of the result

Get the surface/inside atom list: $(S, D) \leftarrow$ GETSURFACE(R).
Let the box size of L be $b_L = (x_L, y_L, z_L)$
Use the S with $\beta \cdot b_L$ to find the maximum degree box and delete those s_i's being inside the box k times
Get the top three atoms closest to the box from D, and use these atoms to push the grid-box by α factor
Set up k gridboxes, $B = \{b_1, b_2 ..., b_k\}$, each with size $g \cdot b_L$
Prepare `.gpf` and `.dpf` for each box $b_i \in B$ by `preparegpf.py` and `preparedpf.py`, and then start `autogrid` and `autodock`
The result is inside k `.dlg` file; get the best (pos, E) from k `.dlg`
return (pos, E)

GETSURFACE(R) is the function that use to distinguish the surface/inside atoms by σ strategy.

Figure 3: The k-GRIDBOX algorithm.

framwork to reduce the time of waiting each result. It it cost a lot of time. Tuning each parameter we need at least 4 hours for 32 nodes, the total computation time is 128 hour. The ligand-receptor pair is 36 set of 6 ligand (1IL6, 1ITB, 2LY4, 2TUN, 2YRQ, 6I1B) [8, 23] of TNF-α, IL-1β, IL-6 and HMGB1, and 6 receptor (1QU6, 1ZIW, 2A0Z, 2Z62, 2Z7X, 2Z80) [27], toll-like receptor (TLR), double-stranded RNA-activated protein kinase (PKR) are member of pattern recognition receptors [27]. Both ligand and receptor is macromolecule (protein). In tuning parameter experiments, we fixed resolution $r = 1$ and probe number $p = 50000$.

3.1 Tuning environment and data source

The experiments are performed on one NFS server and four IBM blade server in the Providence University Cloud Computation Laboratory. Each server is equipped with two Quad-Core Intel Xeon 2.26GHz CPU, 24G RAM, and 296G disk under the Ubuntu version 12.4 with the virtualization platform KVM/QEMU. Under the current system environment, we create 32 virtual machines by KVM; each virtual machine is set to one core CPU, 1G RAM, and 10G disk running under the O.S. Ubuntu version 12.04 with Hadoop version 1.21 MapReduce platform. Each virtual machine is responsible for one map operation and one reduce operation. The total number of the map/reduce operations is up to 32 respectively.

The Protein structure data sources are gathered from the Protein Data Bank [1]; it maintains this single archive of macromolecule structural data freely and publicly available to the global community. These protein structure data are downloaded from the wwPDB's ftp server (ftp://ftp.wwpdb.org/), where the number of protein structure data is 97,980. We download 6 ligand and 6 receptor as 36 set data are treated as the testing data for our experiments.

3.2 Hadoop MapReduce framework

Hadoop is a software framework for coordinating computing nodes to process distributed data in parallel. Hadoop adopts the map/reduce parallel programming model, to develop parallel computing applications. The standard map/reduce mechanism has been applied in many successful Cloud computing service providers, such as Yahoo, Amazon EC2, IBM, Google and so on. An application developed by Map/Reduce is composed of Map stage and Reduce stage (optionally). Figure 5 illustrates the Map/Reduce framework. Input data will be split into smaller chunks corresponding to the number of Maps. Output of Map stage has the format of $\langle key, value\rangle$ pairs. Output from all Map nodes, $\langle key, value\rangle$ pairs, are classified by key before being distributed to Reduce stage. Reduce stage combines value by key. Output of Reduce stage are $\langle key, value\rangle$ pairs where each key is unique.

Hadoop cluster includes a single master and multiple slave nodes. The master node consists of a jobtracker, tasktracker, namenode, and datan-

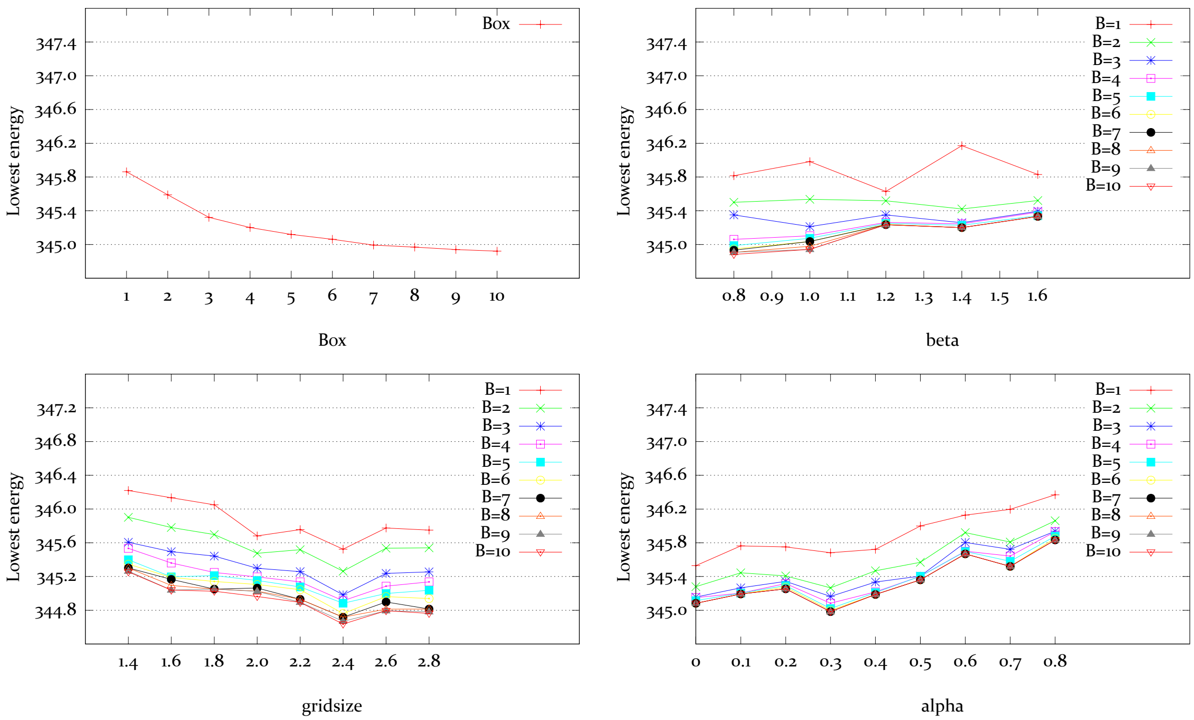

Figure 4: Experiments tested on various settings of parameters $\{k, \beta, g, \alpha\}$. Results suggest that the ligand box magnifying factor β shall be reasonably set around 0.8, and the pushing distance for the ligand grid-box shall be set to 0.3. The CPU time needed for these experiments takes approximately 430 machine-hours.

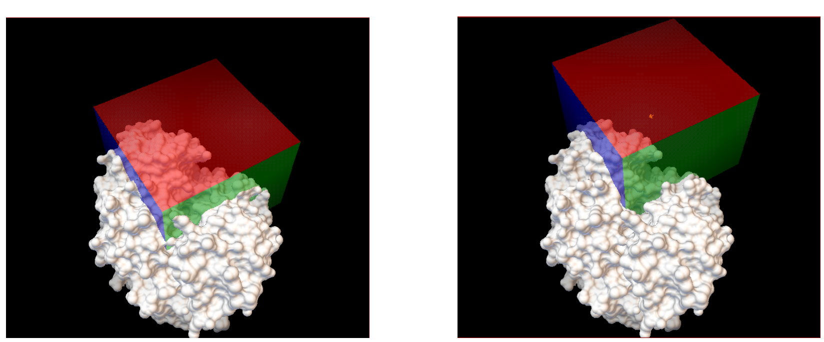

Figure 5: The left figure shows the orignal grid-box position; the figure on the right shows the grid-box is pushed out of the molecular surface by α factor.

ode. A slave node, as computing node, consists of a datanode and tasktracker. The jobtracker is the service within Hadoop that farms out Map/Reduce tasks to specific nodes in the cluster, ideally the nodes that have the data, or at least are in the same rack. A tasktracker is a node in the cluster that accepts tasks; Map, Reduce and Shuffle operations from a jobtracker.

Hadoop Distributed File System (HDFS) is the primary file system used by Hadoop framework. Each input file is split into data blocks that are distributed on datanodes. Hadoop also creates multiple replicas of data blocks and distributes them on datanodes throughout a cluster to enable reliable, extremely rapid computations. The namenode serves as both a directory namespace manager and a node metadata manager for the HDFS. There is a single namenode running in the HDFS architecture.

3.3 Time analysis on Map/Reduce framework

The MapReduce framework for the molecular docking analysis. Assume that the number of compute node is n and the number of ligand-receptor pairs lines is p in the ligand-receptor pairs list file, the p lines will become p map tasks and send to hadoop by streaming program. Each computing node receives a map and then performs the analysis work. When a node completes the map task, the node passes the score to the Reduce and receives next map task to compute unless the total map are finished. Generally, each node will be assigned to deal with p/n maps.

Therefore, Reduce will have p evaluation scores. The evaluation score is described in the previous section 2.1. The reduce operation writes the molecular docking pair evaluation scores line by line to the output file on NFS.

3.4 Performance

It is shown in Figure 6 that the experiments testing the cost/performance of k-GRIDBOX algorithm under various parametric settings (k, β, g, α) versus the GLOBAL-GRID method. The total CPU time needed for these experiments takes approximately 684 machine-hours. The experiment suggests that the k-GRIDBOX algorithm spends significantly less computation time comparing to the general GLOBAL-GRID method by reduction in the grid-box searching space. In average, the proposed k-GRIDBOX algorithm is about 2.3 folds faster comparing to the general GLOBAL-GRID method.

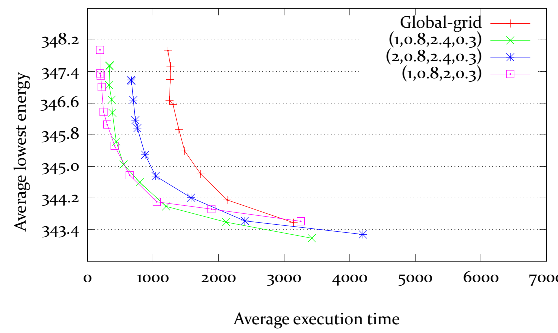

Figure 6: Performance of the k-GRIDBOX algorithm under various parametric settings (k, β, g, α) versus the GLOBAL-GRID method. These results are tested upon 36 different ligand-receptor pairs. Points with the same color are linked with different setting by various probe numbers. The CPU time needed for these experiments takes approximately 684 machine-hours.

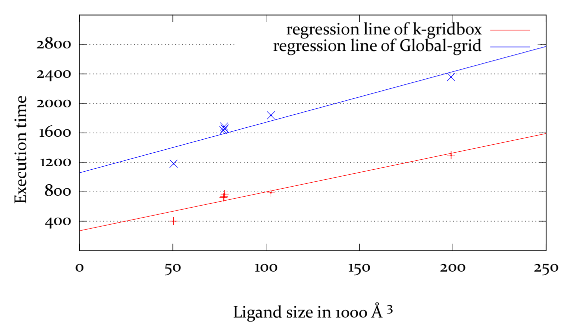

Figure 7: Execution times needed for k-GRIDBOX$(1, 0.8, 2.4, 0.3)$ (red) and GLOBAL-GRID (blue) that achieves similar lowest energy values. In average, the proposed k-GRIDBOX algorithm is about 2.3 folds faster.

To avoid the time delay, our web service provides user access keys for user to check the result later on after they submit the desired query. User can come back and check the result once tasks are finished by the system.

4 Concluding Remarks

In this paper, we proposed an algorithm to distinguish the surface/inside atoms (S, D), and then choose the best s_i who has the most neighbor s_j in box $\beta'$to place the gridbox. While we set a good parameter (k, β, g, α), we will get a better lowest energy value competitive with the GLOBAL-GRID method. In average, the proposed k-GRIDBOX algorithm is about 2.3 folds faster.

In the future, we will perform more experiments to find better algorithms for better results in the same computation time and investigate algorithmic methods for discovering bioinformatics functions, and try to parallelize these methods to provide more perspectives for biologists to improve the performance of these computation frameworks on for analyzing the increasingly huge bioinformatics data.

5 Acknowledgment

This research was partially supported by the National Science Council under the Grants NSC-99-2632-E-126-001-MY.

References

[1] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000.

[2] J. Biesiada, A. Porollo, P. Velayutham, M. Kouril, and J. Meller. Survey of public domain software for docking simulations and virtual screening. *Human genomics*, 5(5):497, 2011.

[3] Z. Bikadi and E. Hazai. Journal of cheminformatics. *Journal of cheminformatics*, 1:15, 2009.

[4] J. M. Bland and D. G. Altman. Statistics notes: measurement error. *Bmj*, 312(7047):1654, 1996.

[5] B. R. Brooks, C. L. Brooks, A. D. MacKerell, L. Nilsson, R. J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, et al. Charmm: the biomolecular simulation program. *Journal of computational chemistry*, 30(10):1545–1614, 2009.

[6] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus. Charmm: A program for macromolecular energy, minimization, and dynamics calculations. *Journal of computational chemistry*, 4(2):187–217, 1983.

[7] N. Cerqueira, J. Ribeiro, P. Fernandes, and M. Ramos. vslaban implementation for virtual high-throughput screening using autodock and vmd. *International Journal of Quantum Chemistry*, 111(6):1208–1212, 2011.

[8] C.-J. Chen, Y.-C. Ou, S.-Y. Lin, S.-L. Raung, S.-L. Liao, C.-Y. Lai, S.-Y. Chen, and J.-H. Chen. Glial activation involvement in neuronal death by japanese encephalitis virus infection. *Journal of General Virology*, 91(4):1028–1037, 2010.

[9] H. Claußen, C. Buning, M. Rarey, and T. Lengauer. Flexe: efficient molecular docking considering protein structure variations. *Journal of molecular biology*, 308(2):377–395, 2001.

[10] W. D. Cornell, P. Cieplak, C. I. Bayly, I. R. Gould, K. M. Merz, D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, and P. A. Kollman. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *Journal of the American Chemical Society*, 117(19):5179–5197, 1995.

[11] I. W. Davis and D. Baker. Rosettaligand docking with full ligand and receptor flexibility. *Journal of molecular biology*, 385(2):381–392, 2009.

[12] J. Dean and S. Ghemawat. Mapreduce: a flexible data processing tool. *Communications of the ACM*, 53(1):72–77, 2010.

[13] Y. Duan, C. Wu, S. Chowdhury, M. C. Lee, G. Xiong, W. Zhang, R. Yang, P. Cieplak, R. Luo, T. Lee, et al. A point-charge force field for molecular mechanics simulations of proteins based on condensed-phase quantum mechanical calculations. *Journal of computational chemistry*, 24(16):1999–2012, 2003.

[14] E. Fischer. Einfluss der configuration auf die wirkung der enzyme. *Berichte der deutschen chemischen Gesellschaft*, 27(3):2985–2993, 1894.

[15] R. A. Friesner, J. L. Banks, R. B. Murphy, T. A. Halgren, J. J. Klicic, D. T. Mainz, M. P. Repasky, E. H. Knoll, M. Shelley, J. K. Perry, et al. Glide: a new approach for rapid, accurate docking and scoring. 1. method and assessment of docking accuracy. *Journal of medicinal chemistry*, 47(7):1739–1749, 2004.

[16] M. Gerstein, R. Jansen, T. Johnson, J. Tsai, and W. Krebs. Studying macromolecular motions in a database framework: from structure to sequence. In *Rigidity theory and applications*, pages 401–420. Springer, 2002.

[17] D. E. Goldberg and J. H. Holland. Genetic algorithms and machine learning. *Machine learning*, 3(2):95–99, 1988.

[18] D. S. Goodsell, G. M. Morris, and A. J. Olson. Automated docking of flexible ligands: applications of autodock. *Journal of Molecular Recognition*, 9(1):1–5, 1996.

[19] T. A. Halgren. Merck molecular force field. i. basis, form, scope, parameterization, and performance of mmff94. *Journal of computational chemistry*, 17(5-6):490–519, 1996.

[20] J. H. Holland. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence.* U Michigan Press, 1975.

[21] R. Huey, G. M. Morris, A. J. Olson, and D. S. Goodsell. A semiempirical free energy force field with charge-based desolvation. *Journal of computational chemistry*, 28(6):1145–1152, 2007.

[22] A. N. Jain. Surflex: fully automatic flexible molecular docking using a molecular similarity-based search engine. *Journal of medicinal chemistry*, 46(4):499–511, 2003.

[23] J. H. Jung, J. H. Park, M. H. Jee, S. J. Keum, M. S. Cho, S. K. Yoon, and S. K. Jang. Hepatitis c virus infection is blocked by hmgb1 released from virus-infected cells. *Journal of virology*, 85(18):9359–9368, 2011.

[24] P. Kolb, R. S. Ferreira, J. J. Irwin, and B. K. Shoichet. Docking and chemoinformatic screens for new ligands and targets. *Current opinion in biotechnology*, 20(4):429–436, 2009.

[25] D. Koshland Jr. Application of a theory of enzyme specificity to protein synthesis. *Proceedings of the National Academy of Sciences of the United States of America*, 44(2):98, 1958.

[26] P. T. Lang, S. R. Brozell, S. Mukherjee, E. F. Pettersen, E. C. Meng, V. Thomas, R. C. Rizzo, D. A. Case, T. L. James, and I. D. Kuntz. Dock 6: Combining techniques to model rna–small molecule complexes. *Rna*, 15(6):1219–1230, 2009.

[27] M. S. Lee and Y. Kim. Pattern-recognition receptor signaling initiated from extracellular, membrane, and cytoplasmic space. *Molecules and cells*, 23(1):1, 2007.

[28] J. E. Lennard-Jones. Cohesion. *Proceedings of the Physical Society*, 43(5):461, 1931.

[29] A. D. MacKerell, B. Brooks, C. L. Brooks, L. Nilsson, B. Roux, Y. Won, and M. Karplus. Charmm: the energy function and its parameterization. *Encyclopedia of computational chemistry*, 1998.

[30] G. Morris, D. Goodsell, M. Pique, W. Lindstrom, R. Huey, S. Forli, W. Hart, S. Halliday, R. Belew, and A. Olson. User guide autodock version 4.2, 2012.

[31] G. M. Morris, D. S. Goodsell, R. S. Halliday, R. Huey, W. E. Hart, R. K. Belew, and A. J. Olson. Automated docking using a lamarckian genetic algorithm and an empirical binding free energy function. *Journal of computational chemistry*, 19(14):1639–1662, 1998.

[32] E. F. Pettersen, T. D. Goddard, C. C. Huang, G. S. Couch, D. M. Greenblatt, E. C. Meng, and T. E. Ferrin. Ucsf chimeraa visualization system for exploratory research and analysis. *Journal of computational chemistry*, 25(13):1605–1612, 2004.

[33] A. Porollo and J. Meller. Polyview-mm: web-based platform for animation and analysis of molecular simulations. *Nucleic acids research*, 38(suppl 2):W662–W666, 2010.

[34] M. C. Schatz. Cloudburst: highly sensitive read mapping with mapreduce. *Bioinformatics*, 25(11):1363–1369, 2009.

[35] R. C. Taylor. An overview of the hadoop/mapreduce/hbase framework and its current applications in bioinformatics. *BMC bioinformatics*, 11(Suppl 12):S1, 2010.

[36] O. Trott and A. J. Olson. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010.

[37] M. L. Verdonk, J. C. Cole, M. J. Hartshorn, C. W. Murray, and R. D. Taylor. Improved protein–ligand docking using gold. *Proteins: Structure, Function, and Bioinformatics*, 52(4):609–623, 2003.

[38] T. White. *Hadoop: The Definitive Guide: The Definitive Guide*. O'Reilly Media, 2009.

[39] M. I. Zavodszky, P. C. Sanschagrin, L. A. Kuhn, and R. S. Korde. Distilling the essential features of a protein surface for improving protein-ligand docking, scoring, and virtual screening. *Journal of computer-aided molecular design*, 16(12):883–902, 2002.