Computers in Biology and Medicine ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at SciVerse ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/cbm

A scalable and accurate method for classifying protein–ligand binding geometries using a MapReduce approach

T. Estrada [a,1], B. Zhang [a,1], P. Cicotti [b], R.S. Armen [c], M. Taufer [a,*]

[a] Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716, United States
[b] San Diego Supercomputer Center, La Jolla, CA 92093, United States
[c] Department of Pharmaceutical Sciences, Thomas Jefferson University School of Pharmacy, Philadelphia, PA 19107, United States

ARTICLE INFO

Article history:
Received 22 December 2011
Accepted 9 May 2012

Keywords:
High-throughput docking
Cross docking
Ensemble docking
Octree-based clustering
Volunteer computing

ABSTRACT

We present a scalable and accurate method for classifying protein–ligand binding geometries in molecular docking. Our method is a three-step process: the first step encodes the geometry of a three-dimensional (3D) ligand conformation into a single 3D point in the space; the second step builds an octree by assigning an octant identifier to every single point in the space under consideration; and the third step performs an octree-based clustering on the reduced conformation space and identifies the most dense octant. We adapt our method for MapReduce and implement it in Hadoop. The load-balancing, fault-tolerance, and scalability in MapReduce allow screening of very large conformation spaces not approachable with traditional clustering methods. We analyze results for docking trials for 23 protein–ligand complexes for HIV protease, 21 protein–ligand complexes for Trypsin, and 12 protein–ligand complexes for P38alpha kinase. We also analyze cross docking trials for 24 ligands, each docking into 24 protein conformations of the HIV protease, and receptor ensemble docking trials for 24 ligands, each docking in a pool of HIV protease receptors. Our method demonstrates significant improvement over energy-only scoring for the accurate identification of native ligand geometries in all these docking assessments. The advantages of our clustering approach make it attractive for complex applications in real-world drug design efforts. We demonstrate that our method is particularly useful for clustering docking results using a minimal ensemble of representative protein conformational states (receptor ensemble docking), which is now a common strategy to address protein flexibility in molecular docking.

1. Introduction and motivation

Cutting-edge distributed technologies, such as cloud and volunteer computing, provide scientists with an efficient and scalable way to perform computationally expensive simulations at a rate never seen before. However, this new capability to perform longer simulations presents new challenges for scientists who have to deal with the analysis, sorting, and selection of scientifically meaningful results from the massive amounts of data collected. Clustering techniques are an effective approach used to characterize scientific datasets. However, the state-of-the-art clustering requires comparing data with each other, many times in an iterative process. When speaking of massive datasets, even a small number of comparisons have a great impact on the efficiency of the clustering algorithm. In this paper, we propose a linear octree-based clustering that gets rid of explicit comparisons and relies on transforming the geometric space of data for their analysis. This transformation intrinsically organizes data points into a new space that better captures similarities. Contrary to other clustering techniques, the linear nature of our method makes it a logical fit for the MapReduce paradigm. By taking advantage of MapReduce, we can analyze very large datasets that cannot be explored by other, less-efficient, methods.

Specifically, this paper targets large datasets of protein–ligand conformations and presents how to apply our novel approach that is both accurate and scalable. The large datasets we use for illustrating our approach are produced by Docking@Home (D@H). D@H performs high-throughput docking simulations on computing resources donated by the public (e.g., idle computing cycles of desktops and laptops connected to the Internet) by docking small molecules, called ligands, to target proteins. In the docking process, the ligand docks into the protein binding site and plays an essential role in turning protein function on or off, which is the molecular basis of the therapeutic benefit of most

* Principal corresponding author. Tel.: +1 3028310071.
E-mail addresses: estrada@udel.edu (T. Estrada), bzhang@udel.edu (B. Zhang), pcicotti@sdsc.edu (P. Cicotti), roger.armen@jefferson.edu (R.S. Armen), taufer@acm.org, taufer@udel.edu (M. Taufer).
[1] T. Estrada and B. Zhang have contributed equally to this work.

0010-4825/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compbiomed.2012.05.001

drugs [11]. The computational simulation of this process requires that a large number of ligand conformations are generated, docked and, scored. In virtual screening, scientists select a reduced number of favorable drug-like small molecule candidates among the results to experimentally test them as possible new lead molecules.

A crucial step in this process is the accurate prediction of the binding geometry of a single ligand, which requires the evaluation of numerous possible predicted protein–ligand geometries. In evaluating an ensemble of possible protein–ligand binding geometries (for a single ligand), scientists typically rely on the traditional scoring approach based on a molecular mechanics energy function. This approach relies on the assumption that the conformations with lower energy are more likely to have a near-native conformation, which is correct given a perfect energy function that accurately describes the physics of the system. Note that a conformation is considered a near-native conformation if the root-mean square deviation (RMSD) of the heavy atom coordinates is smaller than or equal to two Angstroms (Å) from the experimentally observed conformation. Ideally the set of conformations with minimum energy and the set of near-native conformations should overlap. Unfortunately, we observed that this is not always the case, even for very large sets of ligand conformations. This is because in some cases the modeling of protein–ligand energies is still inaccurate even when using sophisticated and computationally expensive energy functions such as the Generalized Born implicit solvent model [19], which can provide a rigorous evaluation of electrostatic energetic components due to solvent. In other words, in most easy example protein–ligand complexes, ligand conformations that are geometrically close to near-native conformations exhibit the lowest energy scores. However, in more challenging protein–ligand complexes there are exceptions to this, which are typically due to non-native ligand conformations having a lower false positive energy scores compared to near-native conformations.

In preliminary work, we observed several scenarios in which the selection of tentative ligands based on their minimum energy failed in accurately predicting near-native ligand structures. Figs. 1 and 2 show two representative cases for the ligand 1gi6 in the Ligand-Protein Database or LPDB [27] (Fig. 1) and the ligand 1d4i in the same database (Fig. 2). In particular, Figs. 1a and 2a show the comparison of the geometrical 3D ligands of the crystal structure (black), the top structure sorted by energy (blue), and a ligand selected based on geometrical analysis of the whole dataset

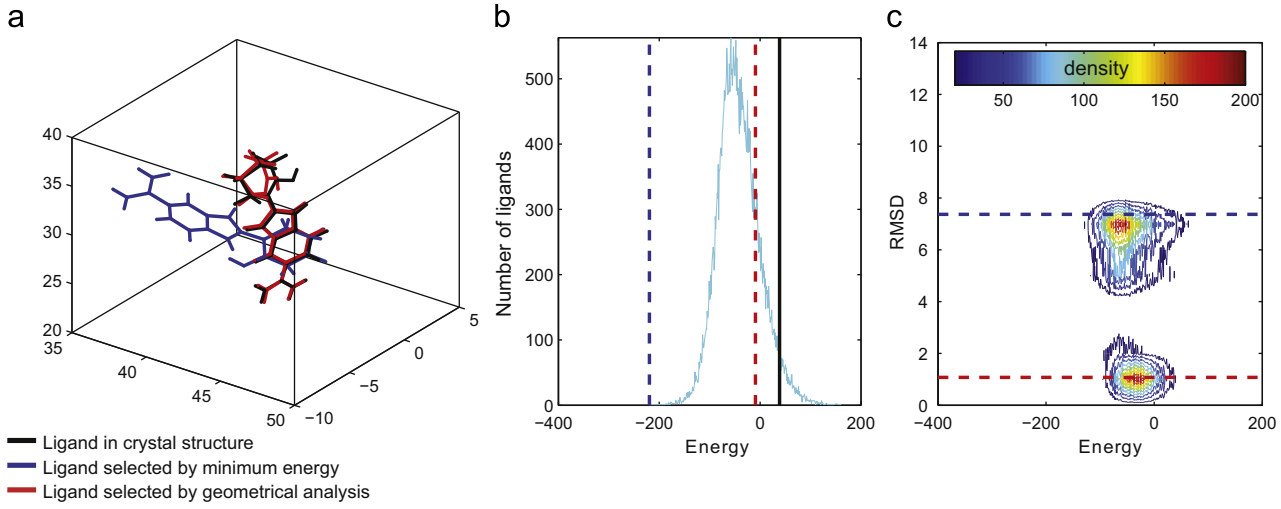

Fig. 1. Analysis of accuracy of ligand conformations for the LPDB ligand 1g16 when selected by minimum energy and selected by geometrical analysis. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

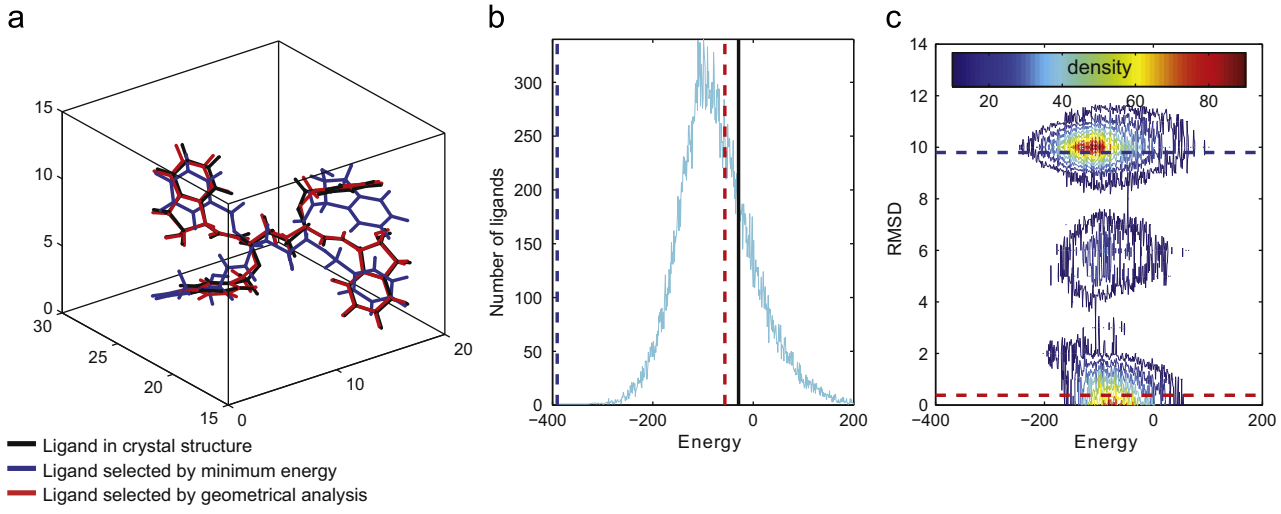

Fig. 2. Analysis of accuracy of ligand conformations for the LPDB ligand 1d4i when selected by minimum energy and selected by geometrical analysis. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

(red). Figs. 1b and 2b show an histogram of energy values for a dataset of 2 million docking attempts (docked ligand conformations). Vertical lines show the ligand in the crystal structure (black), the top structure sorted by energy (blue), and a ligand selected by a geometrical analysis (red). Finally, Figs. 1c and 2c show a contour map of ligand clusters given their energy (x-axis) and geometry expressed by their RMSD from the crystal structure (y-axis). Warmer colors (red, orange, yellow) represent higher densities, cooler colors (green, blue) represent lower densities. The cluster with smaller RMSD contains the near-native ligand structures, while other clusters that appear higher in the graph represent clusters of local minima. Horizontal lines show the relative RMSD for the ligand structure selected based on geometry analysis (red) and the top structure sorted by minimum energy (blue). The two figures point out how conformations scoring minimum energy over very large datasets produced by million docking attempts may be significantly different from the experimentally observed conformation while, if we compare the geometry of each ligand conformation in the large dataset, we can find several conformations that are close to the near-native conformation but do not necessary score the lowest energy. Note that when dealing with drug discovery, the crystal structure conformations are not always known a priori and therefore cannot be used to select candidate ligand conformations. At the same time, the simple analysis of ligand histograms is not powerful enough to find these dense clusters of geometrically similar ligands.

An alternative approach to analyzing the same dataset of predicted protein–ligand conformations would be to consider clustering the results into dominant clusters of putative protein–ligand binding geometries by using hierarchical clustering methods. From a statistical thermodynamic point of view, if the protein–ligand binding process can be described by a funnel shaped free energy landscape (as it has been proposed for the protein folding process), then given an ensemble of possible binding mode clusters, the near-native clusters should be associated with the lowest thermodynamic entropy [29,14]. Rigorous statistical thermodynamic theory [29,16] as well as empirical observations from clustering docking data [28] both support the hypothesis that the closer a cluster is to the native binding site the less entropy it should exhibit (both thermodynamic and Shannon entropy). Thus, the most densely populated cluster with the least positional variance should overlap with the native binding mode. Employing this paradigm, researchers have shown how hierarchical clustering methods can improve protein–ligand docking and virtual screening results compared to relying solely on the lowest energy conformation [36]. Although these powerful hierarchical clustering methods are able to select near-native poses with higher accuracy, unfortunately, for very large datasets, they scale poorly.

Our work presented in this paper is motivated by the need of clustering methods that are accurate and scalable at the same time. The contributions of this paper are: (1) a scalable and accurate method for classifying predicted protein–ligand binding geometries that encode conformation similarities and perform conformation alignments; (2) the integration of the method into the MapReduce paradigm and its implementation in Hadoop; and (3) the assessment of our method in terms of both its scalability and its accuracy for relevant scientific case studies. First, we show that our method is as accurate as traditional clustering methods based on ligand geometry. Then we show how, unlike traditional clustering methods, our method can also scale with the size of the dataset.

The rest of this paper is organized as follows: Section 2 provides an overview of background concepts; Section 3 presents our method and its implementation; Section 4 shows the scalability and accuracy for the different datasets; Section 5 gives a short overview of related work; and Section 6 concludes this paper.

2. Background

2.1. Protein–ligand docking, cross docking, and receptor ensemble docking

The computational search for potential drug-like lead molecules in virtual screening relies on molecular protein–ligand docking to simulate the docking of small molecules (also called ligands) into proteins involved in the disease process. Protein–ligand docking is a search with uncertainties in a very large space of potential docking conformations; this space is shaped by the protein, the ligand, the computational methods, and the degrees of freedom to be explored [17]. Given a protein, several different types of ligands can dock into one of the protein binding pockets. The same protein, once docked with the ligand, can assume different conformations (i.e., set of three-dimensional protein geometries). A protein conformation and the docked ligand form a complex. Complexes are available in databases such as the protein database (PDB) and the Ligand-Protein Database (LPDB) [27]), presenting the final crystal structure of both protein and docked ligand observed experimentally by X-ray crystallography. Protein–ligand docking simulations allow us to understand the docking process and the protein–ligand interatomic interactions (i.e., self-docking or simply docking). Computationally, a protein–ligand docking simulation is a search over a large space of conformations. In particular, given a protein and the ligand docking into the protein binding site (i.e., the protein–ligand complex), a docking simulation consists of a sequence of N independent attempts or trials. For each attempt, a random ligand conformation is generated and docked into the protein active site with randomly generated orientations. The number of docking attempts and related results in a large-scale docking simulation is $O(MN)$, where M is the number of ligands and N is the number of docking attempts for each ligand. Note that both N and M are normally very large numbers.

The docking algorithm normally deals with a flexible ligand; however, because of computing constraints, the protein can be simplified into a rigid 3D lattice, in which the interaction energy of probe atoms is calculated for every point on the lattice. This is mainly due to the computational cost associated to a flexible representation of the protein conformation as well as the computational and storage constraints of the computer systems on which the simulations have to be performed. Cross docking and receptor ensemble docking (RED) simulations can be used to assess the sensitivity of docking results to minor changes in protein conformations due to a flexible binding site, despite the rigid 3D lattice representation of the receptor [1].

In cross docking we consider the protein conformation originally observed experimentally for a single ligand together with all the other conformations (3D lattices) of the same protein determined in complex with other ligands. Given a specific ligand, we dock that ligand in each protein conformation in the dataset. For each protein conformation, we dock multiple randomly generated conformations of the same ligand, each with different randomly generated rotations. The number of cross docking attempts and related results, when trying to emulate the protein flexibility, are $O(M^2N)$, where M is the number of ligands and protein conformations and N is the number of docking attempts for each ligand. Note that both N and M are normally very large numbers.

The term receptor ensemble docking (RED) typically refers to a strategy where a minimal set of representative protein conformations are used to represent protein flexibility in a virtual screen

Please cite this article as: T. Estrada, et al., A scalable and accurate method for classifying protein–ligand binding geometries using a MapReduce approach, Comput. Biol. Med. (2012), http://dx.doi.org/10.1016/j.compbiomed.2012.05.001

over a large database of ligands. Here, the key difference between cross docking and ensemble docking is that if there were 24 protein–ligand complexes available for a given flexible protein target, the cross docking matrix would incorporate the entire matrix of 24 ligands docked into 24 different protein conformations, while the high throughput ensemble docking strategy may be to select as few as three of the 24 protein conformations to minimally represent the ensemble of conformational states. In this case, the full cross docking test can help to identify which conformations may be the best to use and shows how sensitive the docking results will be to protein flexibility.

The docking process is only one of the key steps, once the results (ligand conformations) are collected, they need to be evaluated to predict the near-native ligand geometry. As mentioned above, selecting the near-native ligand geometry based on energy alone may result in incorrect conclusions, and an alternative approach is to select the near-native geometry from clustering. However, taking this approach can result in extensive computing and storage needs. Ideally the clustering methods have to be scalable, efficient, and accurate, allowing the scientists to compare and select across a very large set of docking results.

2.2. Sampling large conformation spaces

Many computational molecular docking approaches for sampling large conformational spaces of ligands exist and have been used for virtual screening [26,2,24]. Typically a given docking method is evaluated with a selected number of experimentally determined protein–ligand complexes. In general, various docking methods differ from each other in the algorithm used in the conformational search [6,25], the scoring function [13] used to predict ligand geometries, and the scoring function used to rank compounds (or predict DGbinding).

Although the system software for sampling large conformation spaces is not the key contribution of this paper, here we briefly describe how we collect the docking data used in this paper using Volunteer Computing (VC). VC is a well-established paradigm for scientific projects. It is a form of distributed computing in which ordinary people volunteer processing and storage resources across the Internet to scientific simulations. The growing volunteer computing community includes the following: 65 scientific projects, over 2 million volunteers, and 6 million computers distributed across the world. Berkeley Open Infrastructure for Network Computing (BOINC) [3], a well-known VC middleware, supports Docking@Home (D@H),[2] the VC project producing the data used in this paper. D@H is an NSF-funded project in molecular docking that computationally searches for potential drug-like lead molecules against diseases, such as breast cancer and HIV. D@H generates a very large space of possible docking conformations. To extensively search this space, millions of independent docking attempts (jobs) are processed by the D@H server which distributes them for computation to clients across the Internet. Volunteers' computers perform the docking simulation and return results consisting of the docked 3D ligand conformation and its associated energy values. Currently, more than 22,000 volunteers and 58,000 computers worldwide support D@H.

More specifically in reference to how D@H explores the conformational space of the ligands, D@H considers a representation of the solvent by using two docking methods: (1) a implicit representation of water using a distance-dependent dielectric coefficient (low if the atoms are close and progressively larger as the inter-atom distance increases) and a more physically accurate implicit representation of water using a Generalized Born model [19]. The method based on the Generalized Born model is a more compute and memory intensive method. At the same time it provides a more physically accurate description of the potential energy of a ligand where part of the ligand conformation is exposed to solvent. In many situations where a large portion of the ligand is solvent exposed, the Generalized Born model should help significantly in providing better ligand conformations (e.g., when one orientation of a given ligand leaves a large bulky hydrophobic group exposed to solvent, this is penalized, where exposing a hydrophilic group like a hydroxyl OH group to solvent is much more favorable).

The molecular docking is performed using the CHARMM (Chemistry at HARvard Molecular Mechanics) molecular simulation package [5] and an intermediate accuracy all-atom force field. The CHARMM script describing the docking process considers a protein–ligand complex as a composition of a flexible ligand and a rigid protein structure (i.e., on a three dimensional lattice of regularly spaced points surrounding and centered on the active site of the protein, where each point on the grid stores the potential energy of a "probe" atom's interaction with the molecule). A D@H simulation consists of a sequence of independent trials (or jobs). For each trial, either a randomly generated conformation or a user-defined conformation for a ligand is used as an initial conformation. Random conformations are generated starting from the ligand crystal structure with random initial velocities on each ligand atom. Then the initial conformation is randomly rotated to produce a set of different orientations that are places into the active site of the protein or docking pocket (docking attempts).

Once the ligand is docked into the protein site, an MD simulation consisting of a gradual heating phase of 4000 1-femtosecond (1 fs) steps from 300 K to 700 K, followed by a cooling phase of 10,000 1 fs steps back to 300 K, is performed. To facilitate the penetration of ligands into protein sites and allow larger conformational changes, van der Waals (vdW) and electrostatic potentials with soft-core repulsions are utilized. A soft-core repulsion reduces the potential barrier at vanishing inter-atomic distances to a finite limit allowing ligands to pass between conformational minima with a relatively small potential barrier that would normally be very large and impossible to overcome with an unmodified standard potential. The detailed description of the docking method and its comparison with other docking codes is not in the scope of this paper and can be found in past work of the authors [34,33]. Once the results (ligand conformations) are collected, they need to be scored. Initially D@H used an energy-based scoring method; our previous work pointed out to us how this scoring approach can result in incorrect conclusions because energy values are approximated by the simplified methods used in the computational algorithms. The alternative approach that we pursue in this paper is to score ligands based on the geometry of their resulting conformations.

3. Methodology

We propose a novel method to identify 3D ligand conformations docked into one or multiple protein conformations and the method scalable implementation using MapReduce. The load balancing, fault tolerance, and scalability in MapReduce make the method attractive to exhaustively screen the large resulting space of ligand conformations which is difficult by traditional clustering methods.

3.1. Scalable octree-based clustering

Our method has three key steps: the first step, *geometry reduction*, encodes the geometry of a three-dimensional (3D) ligand conformation into a single 3D point in the space; the

[2] Docking@Home URL: http://docking.gcl.cis.udel.edu

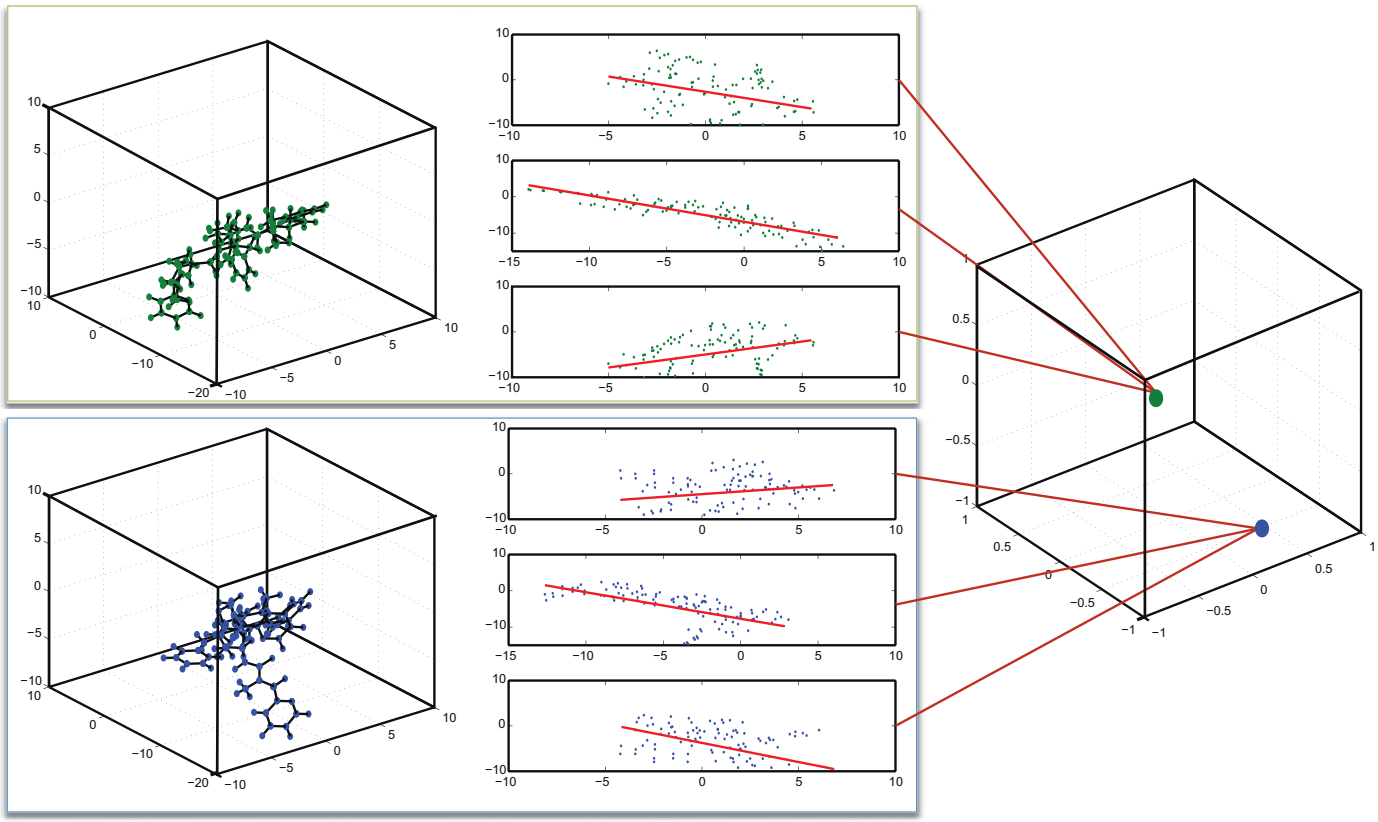

Fig. 3. Mapping 3D ligand conformations of ligand 1hvi into one-point representations.

second step, *octkey generation*, builds an octree by assigning an octant identifier, or octkey, to every single point in the space under consideration; and the third step, *octree-based clustering*, performs an octree-based clustering on the reduced conformation space and identifies the most dense octant. Each step is explained in detail below.

First step—geometry reduction: This step encodes ligand similarities and performs ligand alignments[3] by projecting each ligand conformation from a 3D structure of size $N_a \times 3$ to a single point of size 1×3 in the 3D space. Note that N_a is the number of atoms in the ligand and 3 is the three dimensions (x,y,z). The basic idea is to take the N_a atoms of the ligand conformation and generate a single 3D point that encodes the ligand shape. To this end, we project the 3D atoms of each conformation on each of the three planes: (x,y), (y,z), and (z,x). Each projection results in a set of 2D points on the associated 2D plane. For each projection, we compute the best-fit linear regression line of the 2D points. We calculate the slope of each line and use these slopes as the coordinates of the 3D point encoding the ligand geometry. Eq. (1) shows how we compute the slopes on any of the three planes

$$\alpha_x = slope(x,y) = \frac{N_a \sum_{j=1}^{N_a} (l_{i,k}(x,j)-\mu_x)(l_k(y,j)-\mu_y)}{(l_{i,k}(x,j)-\mu_x)^2}$$

$$\alpha_y = slope(y,z) = \frac{N_a \sum_{j=1}^{N_a} (l_{i,k}(y,j)-\mu_y)(l_k(z,j)-\mu_z)}{(l_{i,k}(y,j)-\mu_y)^2}$$

$$\alpha_z = slope(z,x) = \frac{N_a \sum_{j=1}^{N_a} (l_{i,k}(z,j)-\mu_z)(l_k(x,j)-\mu_x)}{(l_{i,k}(z,j)-\mu_z)^2} \quad (1)$$

[3] Here we refer to "alignment" in the context of molecular structures and not sequence alignment.

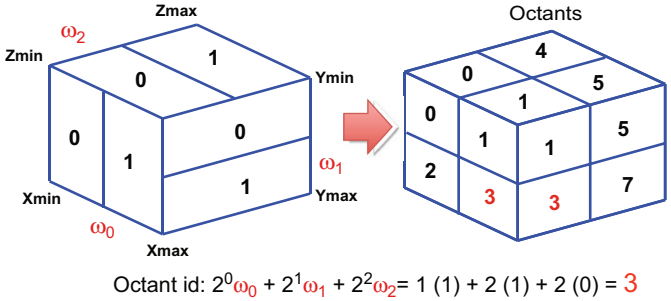

Fig. 4. 3D space partition and graphic representation of the octants given the formula $octantid = 2^0\omega_0 + 2^1\omega_1 + 2^2\omega_2$ in Algorithm 1.

In Eq. (1), $l_{i,k}$ is a 3D ligand conformation produced by D@H, N_a is the number of atoms, and the μ parameters are the mean values of the atomic coordinates in the 2D space considered. Fig. 3 shows an example of the process in which the geometry reduction is performed for two different conformations of the same ligand, 1*hvi*. Our geometry reduction not only compresses the ligand representation, but has the important advantage of projecting ligands with similar geometry close by in the 3D space. This advantage is the key to efficiently perform an octree-based clustering.

Second step—octkey generation: In the second step we assign an octkey (i.e., octant descriptor) to each 3D point representing a ligand conformation in the dataset. To build the octkeys, we initially determine the edge size (i.e., octant resolution) of the 3D space containing all the projected conformations. Empirically we observed that the x, y, and z edge sizes range from -4 to $+4$. We divide the initial 3D space into eight octants of the same size, half the original edge size. Every octant is given an unique identifier ranging from 0 to 7 based on its position in the 3D space. Fig. 4

Please cite this article as: T. Estrada, et al., A scalable and accurate method for classifying protein–ligand binding geometries using a MapReduce approach, Comput. Biol. Med. (2012), http://dx.doi.org/10.1016/j.compbiomed.2012.05.001

shows the procedure to determine how a point (conformation) is assigned to one of the eight octants, along with the graphic representation of the octants in the 3D space. For a given ligand conformation, if its one-point 3D projection falls into the octant with identifier (id) equal to i, then i is the first digit of the octkey of that ligand. For example, if 3 is the id of the octant in which the point falls, then "3" is the first digit of the octkey of the given ligand. For each octant hosting a ligand conformation, we further divide it into eight sub-octants. Then we perform the selection of the second digit based on the id of the sub-octant in which the one-point ligand projection falls. For example, if the considered ligand conformation falls into the sub-octant with id equal to 4, the id "4" is the second digit of the octkey and the octkey becomes "34". In other words, the octkey of a ligand conformation denotes the octant in which the one-point representation of the conformation resides. This process is repeated an arbitrary number of times to produce the complete octkey (octkey$[1 \ldots N_{key}]$), where N_{key} is the number of digits in octkey (see Algorithm 1). Across the large dataset of ligands with diverse degrees of flexibility, we empirically observed that opt-keys of length 15 ($N_{key} = 15$) are sufficient to capture cluster variability and that a larger number of digits does not introduce any improvement in accuracy. Thus in this work the max length of our octkey is 15. Fig. 5 shows the process described above, as well as the traversal of the associated octree, for a point in the space with octkey equal to "342".

Algorithm 1. Octkey computation for each ligand.

Input: 3D single-point coordinates $(\alpha_x, \alpha_y, \alpha_z)$
Output: octkey$[1 \ldots N_{key}]$
for $i = 1$ to N_{key} **do**
$x_{mid} = \lfloor \frac{1}{2}(x_{min} + x_{max}) \rfloor$
$y_{mid} = \lfloor \frac{1}{2}(y_{min} + y_{max}) \rfloor$
$z_{mid} = \lfloor \frac{1}{2}(z_{min} + z_{max}) \rfloor$
if $\alpha_x \geq x_{mid}$ **then**
$\omega_0 = 1$, $x_{min} = x_{mid}$
else
$\omega_0 = 0$, $x_{max} = x_{mid}$
end if
if $\alpha_y \geq y_{mid}$ **then**
$\omega_1 = 1$, $y_{min} y_{mid}$
else
$\omega_1 = 0$, $y_{max} = y_{mid}$
end if
if $\alpha_z \geq z_{mid}$ **then**
$\omega_2 = 1$, $z_{min} = z_{mid}$
else
$\omega_2 = 0$, $z_{max} = z_{mid}$
end if
octkey$[i] = 2^0\omega_0 + 2^1\omega_1 + 2^2\omega_2$
end for

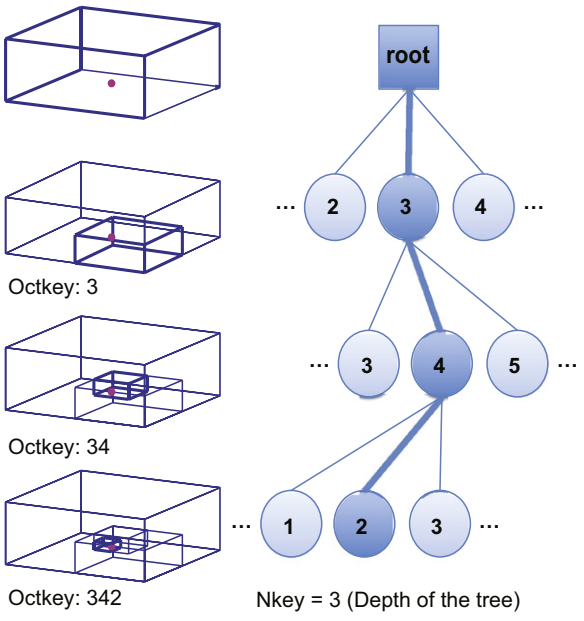

Fig. 5. Process of mapping one-point ligand conformation into a 3-digit octkey.

Third step—octree-based clustering: The last step of our method searches for the most dense octant (i.e., the deepest octant with a number of conformations larger than or equal to a threshold density T_{dens}). This octant represents the most reliable consensus obtained from the available data; we expect the near-native ligand conformations to reside in this octant. Our octree-based clustering performs a search looking for the most dense octant (see Algorithm 2). First, we start by exploring all the nodes (octants) of a certain octree depth δ, where δ is initially equal to $\lfloor \frac{1}{2}N_{key} \rfloor$. The method considers δ digits from each octkey (octkey$[1 \ldots \delta]$) and counts the number of conformations that start with octkey$[1 \ldots \delta]$. This is equivalent to counting the density of each sub-octant encoded with the δ digits. Two boundaries, $\min_{depth}$ and $\max_{depth}$, define the upper and lower levels of the search in the octree. $\min_{depth}$ is the minimum depth that can contain octants with density larger than T_{dens}, and $\max_{depth}$ is the maximum depth that can contain any octant with density smaller than T_{dens}. The search updates the two boundaries. The first time the search is performed, $\min_{depth}$ is equal to 1 and $\max_{depth}$ is equal to N_{key}. For each following iteration, the two boundary conditions are updated based on the following heuristic: the method tests if there is any octant encoded with δ-digits and density larger than the threshold density. If so, the method sets $\min_{depth} = \delta$; otherwise it sets $\max_{depth} = \delta$. Then the method continues by exploring depth $\delta = \lfloor \frac{1}{2}(\min_{depth} + \max_{depth}) \rfloor$. In this search, the octant with density larger than the density threshold in the deepest octree level is the result of our selection.

Algorithm 2. Octree-based clustering.

Input: octkey
Output: best_octant = octant with density $\leq T_{dens}$

$\min_{depth} = 1$
$\max_{depth} = N_{key}$
$\delta = \lfloor (\max_{depth} - \min_{depth})/2 \rfloor$
repeat
for each octant with depth δ **do**
count octant density
end for
if $\exists octant \mid density \leq T_{dens}$ **then**
$\min_{depth} = \delta$
else
$\max_{depth} = \delta$
end if
$\delta = \lfloor (\max_{depth} - \min_{depth})/2 \rfloor$
until $\min_{depth} = \max_{depth}$
best_octant = octant with largest density at deepest octree level

3.2. MapReduce implementation

For the sake of scalability, we rely on MapReduce to efficiently handle the large datasets of conformations. MapReduce is a

parallel programming model that facilitates the processing of large distributed datasets. It was originally proposed by Google to index and annotate data on the Internet [9]. In this paradigm, the programmer specifies two functions: map and reduce. The map function takes as input a key and value pair, performs the map function, and outputs a list of intermediate key and value pairs which may be different from the input

$$Map\langle k_1,v_1\rangle \rightarrow list\langle k_2,v_2\rangle \quad (2)$$

The runtime system automatically groups all the values associated with the same key and forms the input to the reduce function. The reduce function takes as input a key and values pair, performs the reduce function, and outputs a list of values. Note that the input values mean the list of all the values associated with the same key

$$Reduce\langle k_2,list(v_2)\rangle \rightarrow list\langle v_3\rangle \quad (3)$$

MapReduce is appealing to scientific problems, including the one addressed in this paper, because of the simplicity of programming, the automatic load balancing and failure recovery, and the ease of scaling nature. We implemented our framework in Hadoop, the Apache's free and open source implementation of MapReduce.

In our Hadoop implementation, the geometry reduction and octkey generation are implemented in one MapReduce job (MR I). The input dataset is built from the pdb files resulting from D@H and containing the coordinates of the docked ligands. The dataset is transformed into a text file containing the ligand conformations, one conformation per line with each line consisting of the 3D atom coordinates of a ligand conformation. Hadoop partitions the file into independent data splits, each containing a subset of conformations. Each mapper runs an identical map function on its local data split. The map function takes as input a $\langle id_{conf}, 3Dconformation\rangle$ pair in which id_{conf} is an unique identifier for each ligand conformation; transforms the 3D conformation to a single 3D point; encodes the point into a octkey with length N_{key}; and outputs the $\langle id_{conf},octkey[1\ldots N_{key}]\rangle$ pair as intermediate output. The Hadoop runtime system groups all the octkey associated with the same ligand conformation (same id_{conf}) across the multiple mappers and generates the input to the reduce function. The reduce function takes as input a $\langle id_{conf}, list(octkey[1\ldots N_{key}])\rangle$ pair. The output of the function is the same as the input. Note that the reduce function performs a pseudo-reduction by reducing each element into itself to generate the input dataset for the following chain of MapReduce jobs.

The octree-based clustering is implemented as a chain of MapReduce jobs (MR II) in which each job searches the δ level of the octree. Each mapper works with its local partial data by outputting the occurrence of each octant and builds a partial structure of the specific octree level. The reducer constructs the global structure of the specific tree level. The map function takes as input a $\langle id_{conf},octkey[1\ldots N_{key}]\rangle$ pair. For each input pair, it produces a $\langle octkey[1\ldots\delta],1\rangle$ pair where the first δ digits of the octkey and the associated occurrence equal to 1 are the intermediate output. Note that since each first δ digits serve as an identifier for a specific octant, the process is equivalent to tagging the conformations based on the octant they belong to, given a 3D space partitioned in 8^{δ} octants. The Hadoop runtime system groups all the occurrences associated with the same octkey$[1\ldots\delta]$ (octant) and forms the input to reduce. The reduce function takes as input a $\langle octkey[1\ldots\delta],list(1)\rangle$ pair and performs a sum over the list of occurrences. This is equivalent to computing the density of each octant tagged with the identifier $octkey[1\ldots\delta]$. A main function decides whether to run another job based on the density threshold T_{dens} and therefore to explore another level of the octree by using the heuristics presented in the previous section. Fig. 6 shows the overall framework of our MapReduce implementation.

3.3. Method complexity

Our method based on ligand encoding and octree searching analyzes ligand geometries with a linear computational complexity. In practice, the MapReduce implementation of the method runs on multiple machines in parallel. When discussing the method complexity we consider the fact that it runs in parallel, and discuss the parallel method complexity. In the first MapReduce job in Fig. 6 (MR I), the computational complexity of the map function is as follows:

$$O\left(\frac{N_a \times N}{nodes}+\frac{N_{key}*N}{nodes}\right)\approx O(N) \quad \text{for } N \gg N_a, \quad N_{key}$$

where N is the number of ligand conformations, N_a is the number of atoms per conformation, N_{key} is the number of digits for the octkeys, and *nodes* is the number of the Hadoop nodes. The computational complexity of the associated reduction function is $O(N)$.

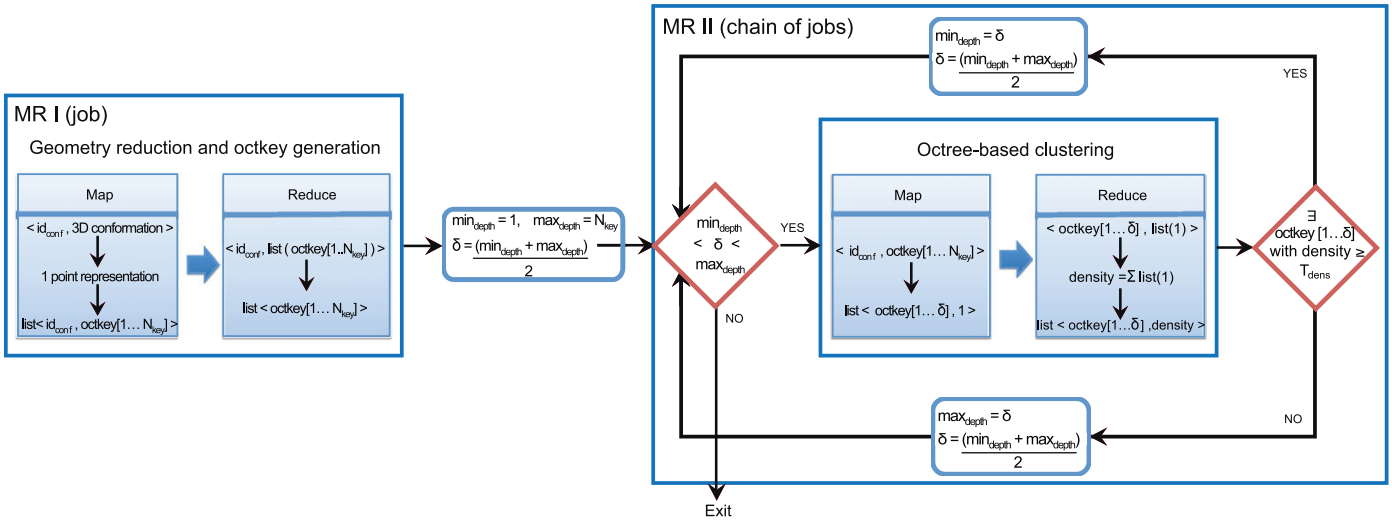

Fig. 6. MapReduce framework of our method implementation using Hadoop.

Please cite this article as: T. Estrada, et al., A scalable and accurate method for classifying protein–ligand binding geometries using a MapReduce approach, Comput. Biol. Med. (2012), http://dx.doi.org/10.1016/j.compbiomed.2012.05.001

In the chain of MapReduce jobs (MR II), the computational complexity of the mapping is as follows:

$$O\left(\frac{\log_2 N_{key} \times N}{node}\right) \approx O(N) \quad \text{with } N \gg 1$$

and the computational complexity of the associated reduction is as follows:

$$O(\log_2 N_{key} \times N) \approx O(N) \quad \text{with } N \gg 1$$

Since the comparison of ligand conformations is done by projecting each conformation onto the same 3D space, the first MapReduce job does not require any explicit data shuffling across nodes. For the chain of MapReduce jobs, Hadoop benefits from the octree to reduce the data shuffling. As described in Section 3.2, each reduce function processes a single octree level rather than the whole tree.

4. Results

4.1. Test set-up

We collect the dataset for testing our proposed method for classifying binding geometries by using the D@H project. On D@H, we ran docking trials for 23 protein–ligand complexes for HIV protease (an aspartic acid protease protein), 21 protein–ligand complexes for Trypsin (a serine protease protein), and 12 protein–ligand complexes for P38alpha kinase (a serine/threonine kinase protein). We also ran cross docking trials for 24 ligands, each docking into 24 protein conformations of the HIV protease. The data are used to assess both accuracy and scalability of our framework.

4.2. Accuracy

We validate the accuracy of our algorithm in the context of docking, cross docking, and ensemble docking (see Section 2.1 for the definitions).

Docking: For each protein–ligand complex, we consider 100,000 ligand conformations sampled with D@H. We compare the capability of our algorithm to capture near-native conformations by comparing its docking accuracy with the docking accuracy of our previous work based on probabilistic hierarchical clustering [11] and the naïve selection approach based on only the lowest conformation energy. For each protein, its docking accuracy is the number of captured near-native conformations over the total number of complexes for that protein. Note that a near-native conformation has a RMSD from the experimentally observed conformation that is smaller than or equal to two Å. For the octree-based clustering, we consider a density threshold equal to 100. In other words, we select the most dense octant with at least 100 ligand conformations. We capture a near-native conformation if the arithmetic median of the octant conformations is below or equal to two Å. The use of the median is preferred as the accuracy metric over the mean because less affected by extreme values [15], although the majority of our overall results are not very sensitive to if the median or mean is used for selection. For the probabilistic hierarchical clustering, the distance metric used to cluster each ligand is the RMSD of its atom coordinates versus all the other ligands already in the cluster. If a simulation converges, then the largest cluster with lower internal variance is likely the cluster that contains more near-native conformations. We capture a near-native conformation if the centroid of the selected cluster is a near-native conformation [11]. For the energy-based approach, we consider 100 D@H conformations selected based on their lowest energy versus the same crystal structure, which we denote the naïve approach. Here, we identify the near-native cluster, if the arithmetic median of the lowest energy conformations is below or equal to 2 Å. It is important to notice that we perform the clustering and selection of the near-native candidates without using any information on the crystal structures available for the complexes. The crystal structures played an important role only in the validation phase of our framework when, for each complex, we calculated the RMSD of the clustering candidate with respect to its crystal structure. Table 1 summarizes the docking accuracy for the three approaches.

Table 1
Comparison of the number of hits for different scoring approaches, i.e., our octree-based clustering performed with Hadoop, a probabilistic hierarchical clustering, and an energy-based scoring method.

Protein	Octree-based clustering (%)	Hierarchical clustering (%)	Min. energy selection (%)
HIV	**22**/**23** (95)	20/23 (86)	8/23 (34)
Trypsin	13/21 (61)	**16**/**21** (76)	5/21 (23)
P38alpha	**10**/**12** (83)	6/12 (50)	1/12 (0.8)
All	**45**/**56** (80)	42/56 (75)	14/56 (25)

For the HIV protease, the octree-based clustering captures 22 of the 23 near-native conformations (docking accuracy of 95%); the probabilistic hierarchical clustering method captures 20 of the 23 near-native conformations (docking accuracy of 86%); and the naïve approach is able to identify only eight of the 23 near-native conformations (docking accuracy of 34%). For Trypsin, the octree-based clustering captures 13 of the 21 near-native conformations (docking accuracy of 61%); the probabilistic hierarchical clustering method captures 16 of the 21 near-native conformations (docking accuracy of 76%); and the naïve approach identifies only five of the 21 near-native conformations (docking accuracy of 23%). For the P38alpha kinase, the octree-based clustering captures 10 of the 12 near-native conformation (docking accuracy of 83%); the probabilistic hierarchical approach captures six of the 12 near-native conformation (docking accuracy of 50%); and the naïve approach identifies one of the 12 near-native conformations (docking accuracy of 0.8%). The octree-based method outperforms the naïve approach for all the complexes and is as accurate as the probabilistic hierarchical clustering. In particular, the octree-based clustering outperforms the probabilistic hierarchical clustering for HIV and P38alpha but it has a lower docking accuracy for trypsin.

In Fig. 7 we show a case study for one of the HIV ligands docking into the HVI protease. For better visibility, the 3D space spans from -1 to 1. The figure shows the octants that were traversed by our algorithm using a density threshold equal to 100 (as defined in Section 3.1). Lighter dots (cyan) are conformations with RMSD larger than 2 Å and darker dots (red) are near-native conformations (i.e., conformations with RMSD smaller than or equal to 2 Å compared with the known crystal structure). The best conformations converge towards a dense octant in the octree as stated in out hypothesis. A similar behavior was observed for the other ligands were the near-native cluster was successfully identified in Table 1.

Cross docking (*application of octree-based clustering to each individual cross docking simulation*): When the dataset is very large, clustering methods such as those presented in [11] are not scalable. The MapReduce implementation of our algorithm allows us to accurately select near-native conformations across large datasets while benefiting from distributing data and computation among multiple nodes by using Hadoop and its distributed file

Please cite this article as: T. Estrada, et al., A scalable and accurate method for classifying protein–ligand binding geometries using a MapReduce approach, Comput. Biol. Med. (2012), http://dx.doi.org/10.1016/j.compbiomed.2012.05.001

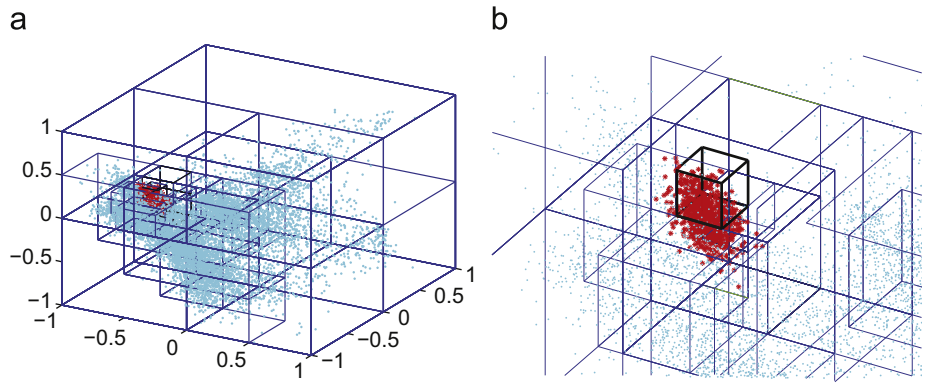

Fig. 7. Octree of the sampled conformations of ligand 1k1l. The most dense octant located deeper in the octree contains near-native ligand structures. (a) Entire octree for ligand 1k1l and (b) zoom in showing the dense octant. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

system. This is the case for the analysis of cross docking and ensemble docking results. As pointed out in Section 2.1, cross docking simulations are used to assess the sensitivity of docking results to minor changes in protein conformation due to a flexible binding site. The cross docking data we analyze here were generated with D@H by docking 24 different ligands from the HIV complex dataset into the 23 out of 24 different HIV protease protein conformations, excluding the protein conformation in which the ligand was previously crystallized. We show how our analysis with the octree-based algorithm significantly increases the accuracy of the cross docking and ensemble docking results compared to lowest-energy only scoring. Cross docking exercises, such as this one, are used to compare and contrast differences in docking methodologies in situations where docking results are known to be extremely sensitive due to differences in protein conformation. Here we compare the results using the metric of "overall cross docking accuracy", where each individual cross dock is considered an independent event that can be a docking success (ligand RMSD ≤ 2.0 Å) or a failure (ligand RMSD > 2.0 Å) compared to the native co-crystal structure conformation of the ligand. In this case we have 552 [(24×23)] individual cross docking simulations, where we ignore the results from the diagonal of the matrix that are "self-docking" of the ligand into its own co-crystal structure. Using the lowest energy pose to select the predicted binding mode, only 32 of 552 possible cross docks were successful for an overall cross docking accuracy of 5.8%. Using the median of the 100 lowest energy poses to select the predicted binding mode achieved very similar results (30 successful cross docks out of 552) for an overall cross docking accuracy of 5.4%. In comparison, using the octree-based clustering to select the predicted binding mode showed a large improvement in overall accuracy where 242 of 552 possible cross docks were successful for an overall cross docking accuracy of 43.8%. This large increase in accuracy is primarily due to cross docking situations where non-native ligand conformations exhibits a lower "false positive" energy scores compared to near-native conformations. The octree-based clustering is able to identify that these binding pose clusters are not as dense, which in a statistical thermodynamic context indicates that these "false positive" binding poses may be associated with a larger binding entropy, and/or may also be associated with minor enthalpic penalties for conformational change of the receptor. Both these thermodynamic quantities are difficult to approximate and include as improvements to energy functions. Therefore, the application of octree-based clustering can greatly improve accuracy compared to "lowest energy" only scoring in situations where protein flexibility may diminish docking results.

Ensemble docking (*application of octree-based clustering to pooled data over multiple receptor conformations*): Ensemble docking against several protein conformations is relevant for situations when the binding geometry for new ligands is unknown. In this case, a scalable clustering method is required to simultaneously analyze very large datasets. In the above sections, we outlined how octree-based clustering applied to each individual cross docking simulation can improve the overall docking accuracy. Here, we apply the octree-based clustering to pooled docking data for a single ligand docked against an ensemble of different protein conformations. We have re-analyzed the cross docking experiments reported above, but in this case, instead of applying octree-based clustering to each individual cross dock, we applied octree-based clustering to pooled datasets, where each dataset is composed of a ligand docking to either all but the original receptor conformation where the ligand was crystallized (Scenario a) or three receptor conformations that are representative of different conformational states of the protein (Scenario b).

Although numerous HIV protease conformations are considered for the full cross docking exercise, in practice, for an ensemble docking strategy, it would be beneficial to select a minimal number of representative protein conformations. Finding representative protein conformations can be done by determining the similarity of protein conformations. We analyzed protein conformations by considering their backbone structure as the comparison reference as well as considering important differences in protein side chain conformations. Thus, we grouped the same set of 24 protein–ligand complexes used for cross docking, by calculating the best-fit superposition of alpha carbon backbone atoms of the HIV protease for each pair of complexes. For each superposition, we calculated the C alpha RMSD (RMSD of the alpha carbon atoms) of the HIV protease backbone. Based on this analysis we identified three major backbone conformations (that for simplicity we denote *blue*, *green*, and *red*). Using this notation, if a ligand was taken from a blue protein conformation, and then also docked against a blue, green, and red conformations, it would be ideal if the method analyzing the pooled data could simultaneously identify the correct ligand binding pose bound to a blue conformation, compared to the other possibilities.

For the Scenario (a) with a dataset comprising of all but the original receptor conformation where the ligand was crystallized, we performed ensemble docking of each of the 24 ligands and the pooled data for 23 out of 24 receptors, excluding the protein conformation in which the ligand was previously crystallized. We define a docking success when the ligand RMSD is less than or equal to 2.0 Å and a failure if the ligand RMSD is greater than 2.0 Å compared to the native co-crystal structure conformation of

Please cite this article as: T. Estrada, et al., A scalable and accurate method for classifying protein–ligand binding geometries using a MapReduce approach, Comput. Biol. Med. (2012), http://dx.doi.org/10.1016/j.compbiomed.2012.05.001

the ligand. Using the lowest energy pose to select the predicted binding mode, only six of 24 ligands were successful for an overall ensemble docking accuracy of 25%. In comparison, using the octree-based clustering to select the predicted binding mode showed a large improvement in overall accuracy where 18 of 24 ligands were successful for an overall cross docking accuracy of 75%. Similar to cross docking above, this large increase in accuracy is primarily due to situations where non-native ligand conformations exhibit a lower "false positive" energy score compared to near-native conformations. When we extend this analysis to identify the receptor conformational state, our octree approach correctly identifies 19 of 24 receptors (79%), while the minimum energy approach identifies 17 of 24 (71%) as shown in Fig. 8(a) where each row reports either a success (yes) or a failure (no) and the color of the row identifies the receptor conformational state *blue*, *green*, or *red*.

For ensemble docking it makes sense to select a representative conformation from each of these three different conformational states to minimize computational expense for a high throughput virtual screen with a library of a large number of compounds (e.g., $N > 1000$). Thus, in Scenario B with a dataset comprising of three receptor conformations as representative of different conformational states of the protein, we selected one receptor conformation for each of the three different conformational states we identified above: 1d4h for *blue*, 1m0b for *green*, and 1dif for *red* based on what receptors provided the highest docking accuracy over the 24 different ligands in the cross docking. We performed ensemble docking for each of the 24 ligands and the pooled data for 1d4h,1m0b, and 1dif. Similar to above, now for each of the 21 ligands (24 total—three conformational representatives), a docking success is defined when the ligand RMSD ≤ 2.0 Å, and a failure when ligand RMSD > 2.0 Å. Results are shown in Fig. 8(b), where each row reports whether it was a success (yes) or a failure (no). Additionally, the color of the row denotes whether the receptor is a conformational representative *blue* (1d4h), *green* (1m0b), or *red* (1dif). In 10 out of the 21 cases (47%) our method found a near-native ligand conformation (RMSD ≤ 2.0 Å), and in 12 out of the 21 cases (57%) our method identified the more similar receptor conformation. If we perform the ensemble docking approach and but we select receptor conformations based on minimum energy, we identify also 12 out of 21 (57%) correct receptors but only three out of 21 (14%) near-native ligands. Thus, our method still outperforms the minimum energy approach.

In this scenario, poor accuracy may be due to the selected representative of the *green* conformation, whose ligand members seem to favor either the *blue* or the *red* conformations, but never the *green*. In this particular case, it may have been better to select the best green representative receptor that exhibited the highest docking accuracy for only green ligands, rather than for all 24 ligands. Thus, detailed analysis of cross docking results can help to identify which specific conformations are good representative members of a given subset of conformations. Careful selection of conformational representatives is important since minor differences in side chain conformations between two similar conformational states can have a dramatic difference in the overall docking accuracy of a specific conformation (e.g., the conformation may contribute to accurate docking of a larger number of ligands from *blue*, *red*, or *green* conformations).

4.3. Scalability

We study the scalability of our octree-based clustering by measuring how fast datasets can be analyzed as the number of computing resources increases; we consider two D@H datasets of 1 TBytes (TB) and 5 TB in size. The tests are run on (1) 192 cores of Gordon ION and (2) 1024 cores of Trestles, the two new flash-based systems at SDSC. On Gordon ION, the IO sub-system of the Gordon supercomputer, we analyzed strong and weak scalability

a

	Octree Prediction	Energy Prediction
1d4h	yes	no
1ajv	yes	no
1ajx	yes	yes
1c70	yes	no
1d4i	yes	no
1d4j	yes	no
1hsg	yes	no
1iiq	yes	yes
1ebw	yes	no
1ebz	yes	no
1ec1	yes	no
1g2k	yes	no
1g35	yes	no
1m0b	no	no
1gno	no	no
1hbv	no	no
1hps	yes	yes
1htf	no	no
1dif	yes	no
1hvi	no	yes
1hvj	yes	no
1hvk	yes	yes
1hvl	yes	yes
1ohr	no	no
Receptor	19/24 = 79%	17/24 = 71%
Ligand	18/24 = 75%	6/24 = 25%

b

	Octree Prediction	Energy Prediction
1d4h	Used as blue	representative
1ajv	no	no
1ajx	yes	no
1c70	yes	no
1d4i	yes	no
1d4j	yes	no
1hsg	no	no
1iiq	no	no
1ebw	yes	no
1ebz	yes	no
1ec1	yes	no
1g2k	no	no
1g35	no	no
1m0b	Used as green	representative
1gno	no	no
1hbv	no	no
1hps	no	no
1htf	no	no
1dif	Used as red	representative
1hvi	no	yes
1hvj	yes	no
1hvk	yes	yes
1hvl	yes	yes
1ohr	no	no
Receptor	12/21 = 57%	12/21 = 57%
Ligand	10/21 = 47%	3/21 = 14%

Fig. 8. Identification of near-native ligand conformations and receptor poses using Ensemble docking for two scenarios: (a) all ligands vs. all receptors (excluding self-docking) and (b) three major representative receptors. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

Please cite this article as: T. Estrada, et al., A scalable and accurate method for classifying protein–ligand binding geometries using a MapReduce approach, Comput. Biol. Med. (2012), http://dx.doi.org/10.1016/j.compbiomed.2012.05.001

of the 1 TB D@H dataset. ION is a set of IO nodes, some of which are currently available for production runs: currently, the 16 IO nodes (192 cores) available for production are connected via a QDR Infiniband switch. Each node hosts two hexa-core Intel Xeon running at 2.7 GHz with 48 GB of DDR3-1333 DRAM. Each node also has 16 Intel 710 drives with 300 GB of flash memory each, for high performance IO, aggregated with RAID 0 into a 4.8 TB unit. On Trestles, a cluster with 324 nodes, with four sockets each, we analyzed the weak scalability of a larger D@H dataset of 5 TB in size using 1024 core (32 nodes). Each node in Trestles hosts an octa-core AMD Opteron running at 2.4 GHz, for a total of 32 cores per node, and a total of 10,368 for the whole cluster. Each node also has 64 GB of DDR3-1333 DRAM as well as an Intel X25-M drive with 120 GB flash drive for high performance IO. Due to current limitations on the available resources for production we could use only 32 of the 324 nodes available on Trestles. The implementation of our clustering method is based on Hadoop 0.20.2.

In a first set of tests, we consider a 1 TB dataset of docking results consisting of 768 M ligand conformations, each 1.8 KB in size. The conformations are organized in 128 MB files (hdfs "block size"). For each test, we measure the total time (in seconds) of the MapReduce jobs shown in Fig. 6. For the map function in the MapReduce job performing geometry reduction and octkey generation (MR I) and the chain of map functions in the MapReduce jobs performing the octree clustering (MR II), we also analyze the performance in terms of map time, throughput, and parallel efficiency. The execution of reduce functions partially overlaps with the execution of map functions; thus the reduce functions behavior is captured by the total times.

For the *strong scalability study* of the 1 TB dataset, we start with a fixed number of ligand conformations (i.e., 768 M). As the number of cores increases from 12 to 192, we equally distribute the fixed number of conformations across the cores. Table 2 shows the strong scalability study of our MapReduce framework performing geometry reduction and octkey generation (MR I) and octree clustering (MR II) on the 1 TB dataset on Gordon ION. For the *weak scalability study* of the 1 TB dataset, we consider a fixed number of ligand conformations per core (i.e., 4 M ligand conformations) and we increase the number of cores that execute our clustering approach progressively, from 12 to 192. Table 3 shows the weak scalability study of our MapReduce framework performing geometry reduction and octkey generation (MR I) and octree clustering (MR II) on the 1 TBytes dataset on Gordon ION.

When dealing with strong scalability as shown in Table 2, as we increase the number of cores from 12 to 196, we observe an improvement in terms of execution time of two orders of magnitude; the running time decreases from about 5 h to less than 20 min. The main performance gain comes from the linear improvement of the Map throughput in MR I, which increases from 55.3 MB/s on 12 cores to 878.7 MB/s on 192 cores. The efficiency is always above 94%. After the geometry reduction performed by MR I, the map in the chain of MapReduce jobs in MR II deals with a smaller dataset (approximately 22 GB); thus the contribution of the job chain in MR II to the overall performance gain is modest compared to MR I and the efficiency drops to 58% for 196 cores. When dealing with weak scalability as shown in Tables 3, once again the main performance gain comes from the linear improvement of the Map throughput in MR I: its throughput

Table 2
Strong scalability study of our MapReduce framework performing geometry reduction and octkey generation (MR I) and octree clustering (MR II) on a 1 TB dataset on Gordon ION.

Nodes	1	2	4	8	16
Cores	12	24	48	96	192
# Ligand conf.	768×10^6	768×10^6	768×10^6	768×10^6	768×10^6
# Ligand conf./core	64×10^6	32×10^6	16×10^6	8×10^6	4×10^6
Total time MR I (s)	18,552	9365	4700	2396	1228
Map MR I:					
Time (s)	18,066	9030	4520	2265	1138
Throughput (MB/s)	55.3	110.7	221.2	441.5	878.7
Parallel effic. (%)	n/a	99.0	99.9	96.6	94.4
Total time MR II	2304	1211	669	391	247
Map MR II:					
Time (s)	2221	1137	608	339	173
Throughput (MB/s)	9.9	19.3	36.1	64.8	127.1
Parallel effic. (%)	n/a	95.1	86.0	73.6	58.2

Table 3
Weak scalability study of our MapReduce framework performing geometry reduction and octkey generation (MR I) and octree clustering (MR II) on a 1 TB dataset on Gordon ION.

Nodes	1	2	4	8	16
Cores	12	24	48	96	192
# Ligand conf.	48×10^6	96×10^6	192×10^6	384×10^6	768×10^6
# Ligand conf./core	4×10^6	4×10^6	4×10^6	4×10^6	4×10^6
Total time MR I (s)	1181	1181	1193	1211	1228
Map MR I:					
Time (s)	1128	1134	1136	1136	1138
Throughput (MB/s)	56.7	112.8	220.0	440.1	878.7
Parallel effic. (%)	n/a	100	98.9	97.5	96.9
Total time MR II	237	247	247	251	247
Map MR II:					
Time (s)	159	171	166	177	173
Throughput (MB/s)	8.8	15.7	32.5	62.1	127.1
Parallel effic. (%)	n/a	95.9	95.9	94.4	95.9

Please cite this article as: T. Estrada, et al., A scalable and accurate method for classifying protein–ligand binding geometries using a MapReduce approach, Comput. Biol. Med. (2012), http://dx.doi.org/10.1016/j.compbiomed.2012.05.001

Table 4
Weak scalability study of our MapReduce framework performing geometry reduction and octkey generation (MR I) and octree clustering (MR II) on a 5 TB dataset on Tresles.

Nodes	1	2	4	8	16	32
Cores	32	64	128	256	512	1024
# Ligand conf.	121×10^6	242×10^6	484×10^6	968×10^6	1936×10^6	3872×10^6
# Ligand conf./core	3.8×10^6	3.8×10^6	3.8×10^6	3.8×10^6	3.8×10^6	3.8×10^6
Total time MR I (s)	1150	1191	1256	1319	1767	2869
Map MR I:						
Time (s)	1000	1013	1063	1054	1349	1599
Throughput (MB/s)	160.0	315.8	602.0	1185.9	1853.2	3126.9
Parallel effic. (%)	n/a	96.6	91.6	87.2	65.1	43.8
Total time MR II	336	339	344	352	437	554
Map MR II:						
Time (s)	226	228	257	271	318	469
Throughput (MB/s)	15.4	30.7	54.4	103.3	176.1	202.1
Parallel effic. (%)	n/a	99.1	97.6	95.4	76.8	62.0

increases from 56.7 MB/s on 12 cores to 878.7 MB/s on 192 cores. The parallel efficiency of this map is always above 95%.

For larger datasets of 5 TB, the scalability performance of our clustering framework is well beyond the scalability capabilities present in traditional clustering algorithms, as shown in Table 4. Our clustering starts from 32 cores analyzing 121 M ligand conformations and scales to 1024 cores analyzing the full dataset of 3872 M ligand conformations. Table 4 shows the weak scalability study of our MapReduce framework performing geometry reduction and octkey generation (MR I) and octree clustering (MR II) on the 5 TB dataset on Tresles. The table shows the linearly scaling performance of the clustering framework for 3872 million ligand conformations (i.e., 5 TB dataset). Note that in all experiments each core processes 3.8 M ligands. The per-node parallel efficiency exceeds 85% up to 256 cores, and then drops to 43% for 1024 cores due to the cost of communication between map and reduce becoming the dominant factor. The degradation in performance suggests that the communication-to-computation ratio for the problem on 1024 cores has become too large. Similar behaviors of the MapReduce paradigm were observed in other scalability studies [35].

For both strong and weak scalability studies, the map time in the first MapReduce job dominates the overall time because it deals with the entire dataset. Thus, the main performance gain in our clustering comes from the improvement of the map throughput in MR I.

5. Related work

Exploring the search space of docking conformations has been approached using a variety of techniques including data analytics and clustering. Analytic approaches usually select one or multiple conformations that are likely to be near-native at runtime and then perform an extensive sampling around the predicted conformations. Important work in this direction includes Yang et al. [32] and Liang et al. [21]. These approaches improve the accuracy of docking methods and increase the probability of selecting near-native conformations, but do not provide automatic selection of near-native conformations.

Clustering methods are used to find a "reduced" set of near-native docking conformations and can be classified as semi-automatic and fully automatic. Semi-automatic clustering approaches include work of Lorenzen et al. [22], Bouvier et al. [4], and Chang et al. [7]. Lorenzen et al. [22] select near-native docking conformations by assuming that a bigger cluster is more likely to have better candidate conformations. As in our work, the selection based on cluster size outperforms the ranking based on the energy value. The clustering is driven by manually defined thresholds and can find docking conformations with an accuracy of about 5 Å. Bouvier et al. [4] use a Kohonen self-organizing map (SOM) that is trained in a preliminary phase using drug–protein contact descriptors. As in this paper, Bouvier's work describes the possibility of overcoming the inherent problems of scoring functions by using a statistical analysis of different properties of the docked conformations. Chang et al. [7] performed a simple cluster analysis of docking simulations and uses the size of the clusters to estimate the vibrational entropy of the resulting conformations. The conformation frequency provides information on the energy landscape of binding.

To the best of our knowledge, the only fully automatic clustering capable of identifying near-native ligand conformations is our previous work [11] based on a probabilistic hierarchical clustering and FCM. Similarly to our octree-based algorithm, the hierarchical algorithm also clusters ligand conformations based on their 3D geometry and reaches better accuracy compared to energy-only scoring methods. However, the hierarchical algorithm is not scalable and cannot handle very large datasets. Both semi-automatic and fully automatic clustering methods presented above require that each ligand conformation has to be compared to all the other conformations, therefore they are in the order of $O(N^2)$. For large datasets, such as cross docking datasets, the associated computational complexity becomes prohibitive. Moreover, data dependencies due to the algorithms' intrinsic all-to-all comparisons limit parallelism that can be exploited by these algorithms and make them specially unsuitable for the MapReduce paradigm.

Similar to our method, Milletti and Vulpetti [23] perform a geometry-based abstraction of a molecular structure. Specifically, they profile protein binding pockets by using a shape-context-based method. This method describes the coarse distribution of the pocket shape with respect to a given point in its geometry. The shape-context method is linear in the number of atoms considered to generate a pocket descriptor and, to perform the inhibition map prediction, it needs to calculate binding pocket similarity between a pocket and a target inhibitor. If there are N pockets and M targets, the method's complexity is $O(N \times M)$. The method to generate descriptors can efficiently reduce the space dimensionality of any molecular structure. However, the overall methodology presented in [23] is not practical to be used for the clustering of protein–ligand docking simulations, since it would require similarity comparison of all ligand conformations with each other; making this analysis unsuitable for very large datasets.

The MapReduce paradigm has been used in scientific applications. For example the paradigm is adapted to analyze long MD

Please cite this article as: T. Estrada, et al., A scalable and accurate method for classifying protein–ligand binding geometries using a MapReduce approach, Comput. Biol. Med. (2012), http://dx.doi.org/10.1016/j.compbiomed.2012.05.001

trajectories in parallel by the tool called HiMach [35]. The contribution in the paper includes the design and use of a MapReduce-style programming interface. The paper analyzes statistical data of long trajectories, while here we look at geometrical similarities of large conformational sets. In both the cases the MapReduce has to be adapted and in both cases it results in accurate and scalable solution to the scientific problem. In [20,10], well-known clustering methods such as the *k*-mean clustering and the hierarchical clustering were adapted to fit into the MapReduce framework. The resulting frameworks suffer from the limitations of the clustering algorithms integrated in the paradigm. A similar clustering approach based on the density of single points in an *N*-dimensional space is presented in work of Cordeiro and co-workers in [8]. Contrary to our work that considers the whole dataset and performs a single-pass analysis on it, the three algorithms presented in Cordeiro's paper rely on local clustering of sub-regions and the merging of local results into a global solution that can potentially suffer from accuracy issues. As in [8], we observed a direct correlation between space density and clustering efficiency as well as between space density and clustering accuracy. While it was not in the scope of this paper to study the impact of different space concentrations of optimal solutions and performance, an extensive study of these factors with synthetic data can be found in other work of the authors [12].

The scalability in our work is supported by powerful data structures such as octrees. Octrees have not been successfully used for clustering *N*-dimensional structures before because they deal only with a 3D space. In this work we propose a methodology that overcomes the octree limitations by introducing a geometry reduction step, enabling an efficient and accurate, general-purpose, octree-based clustering. The use of octrees in other scientific domains includes: octree-based information storage of ground motion simulations [31], octree-based parallelization of the kernel-independent fast multipole method (KIFMM) for hybrid computing [18], and octree-based parallel fast Gaussian transformations [30].

6. Conclusions

In protein–ligand docking, accurately ranking a series of ligand conformations (scoring) is important to successfully predict whether a given ligand will bind to one protein more favorably than others. It is acknowledged that energy-based scoring methods are error-prone and that traditional clustering methods based on geometries are not scalable. Still, protein–ligand docking simulations are delivering increasingly larger datasets of ligand conformations, and accurate solutions that are also scalable as the dataset grows are in need. In this paper, we present an accurate, scalable algorithm to deal with large datasets of ligand conformations resulting from docking and cross docking simulations. Our approach combines geometrical knowledge on conformation similarities with an efficient octree-based search across the encoded ligand shapes. We integrate the algorithm in a parallel and scalable MapReduce framework implemented using Hadoop. Tests on protein–ligand docking, cross docking, and ensemble docking show that our octree-based selection of near-native conformations is more accurate than energy-based methods and as accurate as traditional clustering approaches. At the same time we demonstrate that our method scales linearly with the number of ligand conformations, making it suitable for clustering massive datasets, a capability that not many clustering techniques share. Our scalability study shows that as the number of cores in our Hadoop system increases, the execution time of our algorithm decreases from 5 h to 20 min with a performance improvement of nearly two orders of magnitude. Even though our octree-based clustering was developed for selecting near-native ligand conformations, it can be used to analyze other molecular systems, including protein–protein binding geometries as well as protein conformations for protein structure prediction and protein folding.

Conflict of interest statement

Roger Armen's CoI:

PI's Collaborators and Co-Editors (Past 48 Months)

Collaborators: C.L. Brooks III (U Michigan), A. Mapp (U Michigan), M. Taufer (U Delaware), D.J. Doren (U Delaware), T.O. Chan (TJU), U. Rodeck (TJU), J.M. Pascal (TJU), J.Y. Cheung (Temple), A.M. Feldman (Temple), J.L. Benovic (TJU), C.P. Scott (TJU), R.A. Panettieri (U Penn), S.B. Liggett (U Maryland), R.B. Penn (U Maryland), B. Lu (TJU) A.P. Dicker (TJU) J.F. Zhang (TJU).

PI's Graduate Advisors

Valerie Daggett (University of Washington, Seattle), Roland Strong (Fred Hutchinson Cancer Research Center, Seattle), Bill Parsons, (University of Washington, Seatle), Patricia Campbell (University of Washington).

PI's Post Graduate Sponsors

Charles L. Brooks III (U Michigan, Ann Arbor).

Michela Taufer's CoI:

PI's Collaborators and Co-Editors (Past 48 Months)

Collaborators: D.P. Anderson (U Berkeley), R. Armen (TJU), C.L. Brooks III (U Michigan), J. Buhler (WUSL), R.D. Chamberlain (WUSL), E. Cochran (US Geological Survey), K.S. Decker (U Delaware), D.J. Doren (U Delaware), K. Ferreira (Sandia national Lab), O. Fuentes (UTEP), K.J. Johnson (UTEP), J. Lawrence (Stanford), M.-Y. Leung (UTEP), P. Ortoleva (Indiana U), S. Patel (U Delaware), R. Riesen (IBM), A. Rodrigues (Sandia National Lab), T. Solorio (UBA), D.M. Swany (Indiana U), P.J. Teller (UTEP), and D.G. Vlachos (U Delaware).

PI's Graduate Advisors

Thomas M. Stricker (Google) and Daniel A. Reed (Microsoft).

PI's Post-Graduate Sponsors

Andrew A. Chien (U Chicago) and Charles L. Brooks III (U Michigan, Ann Arbor).

Acknowledgments

This work was supported by the NSF IIS #0968350 entitled Collaborative Research: SoCS - ExSciTecH: An Interactive, Easy-to-Use Volunteer Computing System to Explore Science, Technology, and Health and by the NSF OCI Cooperative Agreement #0910847 entitled Flash Gordon: A Data Intensive Computer. We used Trestles and Gordon-ION resources of Teragrid and XSEDE that are provided by SDSC.

The authors thank Joshua Bernstein (Penguin Computing Inc.) for his help in installing and setting Hadoop on our cluster and the D@H volunteers for providing us with essential resources for our protein–ligand docking simulations.

References

[1] M. Totrov, R. Abagyan, Flexible ligand docking to multiple receptor conformations: a practical alternative, Curr. Opin. Struct. Biol. 18 (2) (2008) 178–184.

[2] R. Abagyan, M. Totrov, D. Kuznetsov, A new method for protein modeling and design: applications to docking and structure prediction from the distorted native conformation, J. Comput. Chem. J (1996) 488–506.

[3] D.P. Anderson, BOINC: a system for public-resource computing and storage, in: Proceedings of the Fifth IEEE/ACM International Workshop on Grid Computing, November 2004, pp. 4–10.

[4] G. Bouvier, N. Evrard-Todeschi, J.P. Girault, G. Bertho, Automatic clustering of docking poses in virtual screening process using self-organising map, Bioinf. Adv. Access (2009).
[5] B.R. Brooks, R.E. Bruccoleri, B.D. Olafson, D.J. States, S. Swaminathan, M. Karplus, CHARMM: a program for macromolecular energy minimization, and dynamics calculations, J. Comput. Chem. 4 (1983) 187–217.
[6] B.D. Bursulaya, M. Totrov, R. Abagyan, C.L. Brooks III, Comparative study of several algorithms for flexible ligand docking, J. Comp. Aided Mol. Des. 17 (2003) 755–763.
[7] M.W. Chang, R.K. Belew, K.S. Carroll, A.J. Olson, D.S. Goodsell, Empirical entropic contributions in computational docking: evaluation in APS reductase complexes, J. Comput. Chem. 29 (2008) 1753–1761.
[8] R.L.F. Cordeiro, C. Traina, Jr., A.J.M. Traina, J. López, U. Kang, C. Faloutsos, Clustering very large multi-dimensional datasets with MapReduce, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'11), August 2011, pp. 690–698.
[9] J. Dean, S. Ghemawat, MapReduce: simplified data processing on large clusters, in: Proceedings of the Sixth conference on Symposium on Operating Systems Design and Implementation, December 2004, pp. 107–113.
[10] A. Ene, S. Im, B. Moseley, Fast clustering using MapReduce, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'11), August 2011, pp. 681–689.
[11] T. Estrada, R. S. Armen, M. Taufer, Automatic selection of near-native protein–ligand conformations using a hierarchical clustering and volunteer computing, in: Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology (BCB'10), May 2010, pp. 204–213.
[12] T. Estrada, B. Zhang, P. Cicotti, R.S. Armen, M. Taufer, Reengineering high-throughput molecular datasets for scalable clustering using MapReduce, in: Proceedings of the 14th IEEE International Conference on High Performance Computing and Communications (HPCC'12), June 2012, pp. 1–10.
[13] P. Ferrara, H. Gohlke, D. Price, G. Klebe, C.L. Brooks III, Assessing scoring functions for protein–ligand interactions, J. Med. Chem. 47 (12) (2004) 3032–3047.
[14] M.K. Gilson, J.A. Given, B.L. Bush, J.A. McCammon, The statistical-thermodynamic basis for computation of binding affinities: a critical review, J. Biophys. 72 (3) (1997) 1047–1069.
[15] P.C.D. Hawkins, G.L. Warren, A.G. Skillman, A. Nicholls, How to do an evaluation: pitfalls and traps, J. Comp. Aided Mol. Des. 22 (2008) 179–190.
[16] V. Hnizdo, J. Tan, B.J. Killian, M.K. Gilson, Efficient calculation of configurational entropy from molecular simulations by combining the mutual-information expansion and nearest-neighbor methods, J. Comput. Chem. 29 (10) (2008) 1605–1614.
[17] A. Jain, Bias, reporting, and sharing: computational evaluations of docking methods, J. Comp. Aided Mol. Des. 22 (2008) 201–212.
[18] I. Lashuk, A. Chandramowlishwaran, H. Langston, T. Nguyen, R. Sampathl, A. Shringarpure, R. Vuduc, L. Ying, D. Zorin, G. Biros, A massively parallel adaptive fast-multipole method on heterogeneous architectures, in: Proceedings of the ACM/IEEE International Conference on High Performance Computing, Networking, Storage, and Analysis (SC'09), November 2009, pp. 1–12.
[19] M.S. Lee, M. Feig, F.R. Salsbury Jr., C.L. Brooks III, New analytic approximation to the standard molecular volume definition and its application to generalized Born calculations, J. Comput. Chem. 24 (2003) 1348–1356.
[20] H.-G. Li, G.-Q. Wu, X.-G. Hu, J. Zhang, L. Li, X. Wu, *K*-means clustering with bagging and mapreduce, in: Proceedings of the 44th Hawaii International Conference on System Sciences (HICSS'11), 2011, pp. 1–8.
[21] S. Liang, G. Wang, Y. Zhou, Refining near-native protein–protein docking decoys by local resampling and energy minimization, PROTEINS: Struct. Funct. Bioinf. 1 (2008) 309–316.
[22] S. Lorenzen, Y. Zhang, Identification of near-native structures by clustering protein docking conformations, PROTEINS: Struct. Funct. Bioinf. 68 (2007) 187–194.
[23] F. Milletti, A. Vulpetti, Predicting polypharmacology by binding site similarity: from kinases to the protein universe, J. Chem. Inf. Model. 50 (8) (2010) 1418–1431.
[24] G.M. Morris, D.S. Goodsell, R.S. Halliday, R. Huey, W.E. Hart, R.K. Belew, A.J. Olson, Automated docking using a lamarkian genetic algorithm and empirical binding free energy function, J. Comput. Chem. 19 (1998) 1639–1662.
[25] E. Perola, W.P. Walters, P.S. Charifson, A detailed comparison of current docking and scoring methods on systems of pharmaceutical relevance, Proteins 56 (2004) 235–249.
[26] M. Rarey, B. Kramer, T. Lengauer, G.A. Klebe, A fast flexible docking method using an incremental construction algorithm, J. Mol. Biol. 261 (1996) 470–489.
[27] O. Roche, R. Kiyama, C.L. Brooks III, Ligand–protein database: linking protein–ligand complex structures to binding data, J. Med. Chem. 44 (22) (2001) 3592–3598.
[28] A.M. Ruvinsky, Role of binding entropy in the refinement of protein–ligand docking predictions: analysis based on the use of 11 scoring functions, J. Comput. Chem. 28 (8) (2007) 1364–1372.
[29] A.M. Ruvinsky, A.V. Kozintsev, New and fast statistical-thermodynamic method for computation of protein–ligand binding entropy substantially improves docking accuracy, J. Comput. Chem. 26 (11) (2005) 1089–1095.
[30] R. Sampath, H. Sundar, S. Veerapaneni, Parallel fast Gauss transform, in: Proceedings of the ACM/IEEE International Conference on High Performance Computing, Networking, Storage, and Analysis (SC'10), November 2010, pp. 1–12.
[31] S. Schlosser, M. Ryan, R. Taborda, J. López, D. O'Hallaron, J. Bielak, Materialized community ground models for large-scale earthquake simulation, in: Proceedings of the ACM/IEEE International Conference on High Performance Computing, Networking, Storage, and Analysis (SC'08), November 2008, pp. 1–12.
[32] Y. Shen, I.C. Paschalidis, P. Vakili, S. Vajda, Protein docking by the underestimation of free energy funnels in the space of encounter complexes, PLOS Comput. Biol. 4 (10) (2008) e1000191.
[33] M. Taufer, R.S. Armen, J. Chen, P.J. Teller, C.L. Brooks III, Computational multiscale modeling in protein–ligand docking, IEEE Eng. Med. Biol. Mag. 28 (2) (2009) 58–69.
[34] M. Taufer, M. Crowley, D. Price, A.A. Chien, C.L. Brooks III, Study of an accurate and fast protein–ligand docking algorithm based on molecular dynamics, Concur. Comput.: Pract. Exp. 17 (14) (2005) 1627–1641.
[35] T. Tu, C.A. Rendleman, D.W. Borhani, R.O. Dror, J. Gullingsrud, M.O. Jensen, J.L. Klepeis, P. Maragakis, P. Miller, K.A. Stafford, S.E. Shaw, A scalable parallel framework for analyzing terascale molecular dynamics simulation trajectories, in: Proceedings of the ACM/IEEE International Conference on High Performance Computing, Networking, Storage, and Analysis (SC'08), November 2008, pp. 1–12.
[36] D. Wei, H. Zheng, N. Su, M. Deng, L. Lai, Binding energy landscape analysis helps to discriminate true hits from high-scoring decoys in virtual screening, J. Chem. Inf. Model. 50 (10) (2010) 1855–1864.

Please cite this article as: T. Estrada, et al., A scalable and accurate method for classifying protein–ligand binding geometries using a MapReduce approach, Comput. Biol. Med. (2012), http://dx.doi.org/10.1016/j.compbiomed.2012.05.001