LMDS-based Approach for Efficient Top-k Local Ligand-Binding Site Search

Sungchul Kim

Dept. of Computer Science and Engineering,
POSTECH,
Pohang, Korea
E-mail: subright@postech.ac.kr

Lee Sael

Dept. of Computer Science,
Stony Brook University,
Stony Brook, USA
Dept. of Computer Science,
SUNY Korea, Korea
E-mail: sael@{cs.stonybrook.edu, sunykorea.ac.kr}

Hwanjo Yu

Department of Computer Science and Engineering,
POSTECH,
Pohang, Korea
E-mail: hwanjoyu@postech.ac.kr

Abstract: In this work, we propose a LMDS-based binding-site search for improving the search speed of the Patch-Surfer method. Patch-Surfer is efficient in recognition of protein-ligand binding partners, further speedup is necessary to address multiple-user access. Futher speedup is realized by exploiting LMDS. It computes embedding coordinates for data points based on their distances from landmark points. When selecting the landmark points, we adopt two approaches - random and greedy selection. Our method approximately retrieves top-k results and the accuracy increases as we exploit more landmark points. Although two landmark selection approaches show comparable results, the greedy selection shows the best performance when the number of landmark points is large. Using our method, the searching time is reduced up to 99%, and it retrieves almost 80% of exact top-k results. Additionally, LMDS-based binding-site search+ improves the retrieval accuracy from 80% to 95% while sacrificing the speedup ratio from 99% to 90% compared to Patch-Surfer.

Keywords: structure-based function prediction;protein surface;ligand binding pocket;3D Zernike descriptor;binding-site comparison

1 Introduction

Increasing number of protein structures of unknown function necessitates computational methods for characterizing protein tertiary structures. When the sequence similarity of protein of unknown functions and those of known function are low, typical sequence database searches are not enough to predict the function of unknown protein structures (Abascal & Valencia, 2003; Lee *et al.*, 2007; Hawkins *et al.*, 2008; Hawkins & Kihara, 2007). As an alternative approach, structural information of proteins is used as a legitimate analysis strategy (Gibrat *et al.*, 1996; Shindyalov & Bourne, 1997; Singh & Brutlag, 2008). One approach of characterizing proteins with structural information is through prediction of which ligands are likely to bind to a protein, which is a major task of molecular function of proteins (Venkatraman *et al.*, 2009). (Figure 1 shows the examples of protein with ligand-binding site). However, the complex nature of protein ligand interactions makes it difficult to predict whether a ligand molecule binds to a protein or not. In the previous work, Sael and Kihara (Sael & Kihara, 2010), have observed geometric and physicochemical complementarity between the ligand and its binding site in multiple cases. Accordingly, they proposed a method for finding ligand molecules which bind to a local surface site of a protein by finding similar local pockets of known binding ligands in the protein structure database (Chikhi *et al.*, 2010).

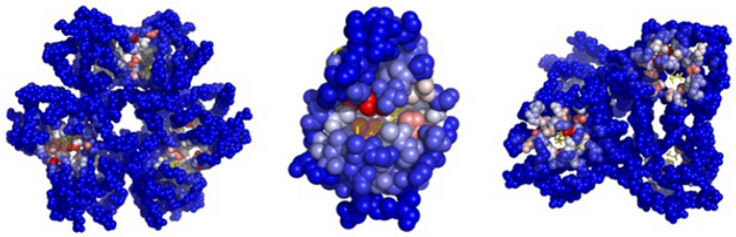

Figure 1 The examples of protein with its ligand binding site

Sael and Kihara (Sael & Kihara, 2012b) also suggested a local surface comparison method, Patch-Surfer, for more accurate prediction of whether a ligand molecule binds to a query protein (Sael & Kihara, 2012b). It represents a binding pocket as a combination of segmented surface patches, each of which is characterized by four representative features: 1) geometrical shape, 2) electrostatic potential, 3) hydrophobicity, and 4) concaveness. The shape and the physicochemical properties of surface patches are represented using the 3D Zernike Descriptors (3DZDs) (Canterakis, 1999). To compare two pockets, patches of given pockets are matched by a modified weighted bipartite matching algorithm and their similarity are evaluated based on a distance function which computes and combines the Euclidean distances of the patch surface features in terms of the four characteristics. However, computing the similarity of pockets is complex, thus finding the local ligand-binding sites based on the distance function takes nontrivial amount of time.

In this paper, we propose an efficient local binding-site search that improves the search speed of the Patch-Surfer, called LMDS-based binding-site search and LMDS-based binding-site search+. We exploit Landmark Multi-Dimensional Scaling (LMDS), which is an efficient version of MDS that is popularly used for efficiently representing

high-dimensional datasets. It computes embedding coordinates for data points based on their distances from a subset of data, landmark points. When selecting the landmark points in LMDS-based binding-site search, we adopt two approaches - random selection and greedy selection. According to our experimental result, LMDS-based binding-site search approximately retrieves top-k results and the accuracy increases as we exploit more landmark points. Although two landmark selection approaches show comparable results, the greedy selection shows the best performance when the number of landmark points is large. Using LMDS-based binding-site search with random or greedy selection, the searching time is reduced up to 99%, and it retrieves almost 80% of exact top-k results. We additionaly propose LMDS-based binding-site search+, which further improves the retrieval accuracy by compromising the efficiency. LMDS-based binding-site search+ retrieves k' nearest neighbors ($k' > k$) and finds k nearest neighbors among them. This approach improves the retrieval accuracy from 80% to 95% while sacrificing the speedup ratio from 99% to 90% compared to Patch-Surfer.

This paper is organized as follows. We briefly discuss related works about protein structure search including the prediction of ligand-binding site (Section 2). Then, we provide details on the proposed method, LMDS-based binding-site search, and its extension, LMDS-based binding-site search+ (Section 3). Finally, we provide experimental results to verify the efficiency of our approaches (Section 4), and conclude with future work (Section 5).

2 Related works

Binding ligand prediction: Ligand binding plays an important role of proteins in a cell and thus provides evidences of protein function in the Gene Ontology (GO) categories (Ashburner *et al.*, 2000). Also, binding ligand prediction has various application such as computational drug discovery (Rosenberg & Goldblum, 2006) and protein design (Samish *et al.*, 2011).

The binding partner of an uncharacterized protein can be predicted by evaluating the similarity of whole protein structure (Skolnick & Brylinski, 2009) or finding the binding pocket to those of known proteins-ligand interactions in the database (Morris *et al.*, 2005; Sael & Kihara, 2012b; Kim *et al.*, 2012). The pocket comparison approach is beneficial in that it can detect similar pockets independent of homologous relationship of proteins. There have been several works for pocket comparison, such as Catalytic Site Atlas (Porter *et al.*, 2004), AFT (Arakaki *et al.*, 2004), and SURFACE (Ferre *et al.*, 2004). Binding pockets are often represented by the positions of the residues or psedocenters in the pockets, and often the root mean square deviation (RMSD) of the residue positions of the searched pocket and the query protein is used to evaluate their similarity. There are also works that exploit geometric hashing to compare conformation of pseudocenters of ligand-binding sites (Gold & Jackson, 2006; Shatsky *et al.*, 2006).

Surface representation of a binding pocket: As an alternative approach to the residue/pseudocenter representations, binding pockets can be represented by protein surface (Sael & Kihara, 2009; Kahraman *et al.*, 2010; Chikhi *et al.*, 2010). The surface representation describes geometrical and physicochemical properties of a pocket on a continuous surface. The eF-seek method (Kinoshita *et al.*, 2007) represents a protein surface as a graph with nodes charactering local geometry and the electrostatic potential. Spherical harmonics based representation is also popular (Morris *et al.*, 2005; Kahraman *et al.*, 2010). Pocket-Surfer is a rotation invariant binding pocket comparison method which represents global surface

shape and the electrostatic potential of binding pockets using 3DZD (Chikhi *et al.*, 2010; Kim *et al.*, 2012). It allows efficient pocket database search. However, binding pockets of the same ligand often shows some variation in their shape and physicochemical properties. To resolve this problem, an extension of Pocket-Surfer, called the Patch-Surfer, was proposed. It has shown to have better performance in accuracy, but with sacrifice of the speed (Sael & Kihara, 2012b). We will provide more details in the next section.

Multi-Dimensional Scaling (MDS): Metric-preserving dimensionality reduction has been an important task in data analysis and machine learning. Given proximity data which consists of dissimilarity information for all pairs of objects, Multi-Dimensional scaling (MDS) (Cox & Cox, 2000) embeds the objects as points in a low-dimensional Euclidean space, while preserving the geometry as precisely as possible. The time complexity of solving MDS is approximately $O(kN^2)$ where N is the number of data points and k is the dimension of the embedding. Using MDS, we can represent protein database based on predefined distance function where they are represented as a complex format such that calculation of distance between queries to all feature vectors in the database is not necessary. Figure 2 is the plotting result of 100 randomly selected proteins' pockets where the dimension of the embedded coordinates is two and three. Landmark MDS (Platt, 2005) is a variant of classical MDS algorithm based on Nymstrom algorithm, to deal with a large number of data points. It preserves all of the attractive properties of the classical MDS algorithm, but is more efficient than the classical MDS algorithm. The time complexity of Landmark MDS is essentially linear in the number of data points.

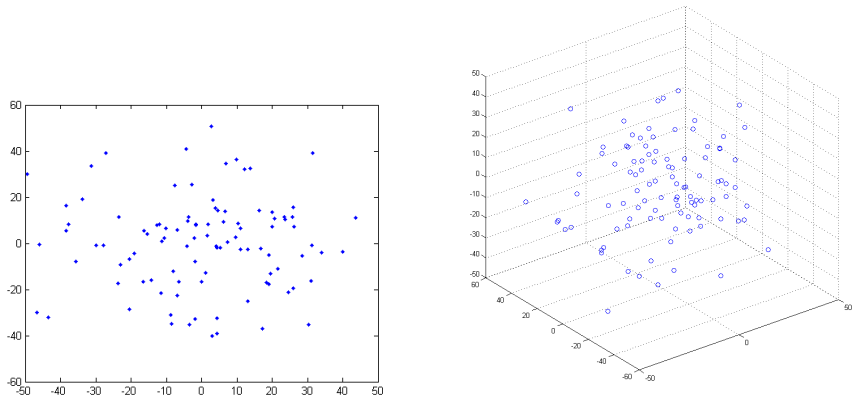

Figure 2 The plotting result of sample dataset; x-axis and y-axis is the first two and three dimensions.

3 Methods

In this section, we first provide a brief description of Patch-Surfer, which searches a database of pockets with known binding partners and finds similar pockets to the query. Then, the description of Landmark MDS and our algorithm, LMDS-based binding-site search, for more efficient top-k retrieval is provided.

3.1 Patch-Surfer method

Patch-Surfer (Sael & Kihara, 2012b) is local surface comparison method that does not require pre-alignment of query pockets to the pockets in the database. With the search results, Patch-Surfer predicts which ligand molecule is likely to bind to the query protein. Given a query protein structure, first surface region of the binding pocket is extracted. If the binding pocket of the query protein is unknown, binding pocket prediction methods can be used. The selected pocket is divided into surface patches represented as four features: geometrical shape, electrostatic potential, hydrophobicity, and concaveness (Sael & Kihara, 2012b). For representation of the four features, 3D-Zernike Descriptor (3DZD) is used (Canterakis, 1999; Venkatraman *et al.*, 2009). Thus, each pocket is represented as a set of surface patches (Sael & Kihara, 2009).

The 3DZD is a series expansion of a 3D function allowing compact and rotationally invariant representation of a 3D object. According to Sael and Kihara (Sael & Kihara, 2012b), to obtain the 3DZDs for a patch, a patch is mapped on a 3D grid and grid points overlapping with the patch are filled with either one for indicating the geometrical shape or their physicochemical measurement values on the particular location. After that, the grid with the assigned values is considered as a 3D function which is expanded into a series in terms of Zernike-Canterakis basis defined as follows (Canterakis, 1999):

$$Z_{nl}^{m}(r,\vartheta,\varphi) = R_{nl}(r)Y_{l}^{m}(\vartheta,\varphi) \tag{1}$$

where $-l < m < l, 0 \leq l \leq n, (n-l)$ is even, $Y_{l}^{m}(\vartheta,\varphi)$ are spherical harmonics, and R_{nl} are radial functions constructed to convert $Z_{nl}^{m}(r,\vartheta,\varphi)$ to polynomials in the Cartesian coordinates, $Z_{nl}^{m}(x)$. To obtain the 3DZD of $f(x)$, 3D Zernike moments need to be computed first. They are defined by expanding the orthonormal bases as follows:

$$\Omega_{nl}^{m} = \frac{3}{4\pi}\int_{|x|\leq 1} f(x)\bar{Z}_{nl}^{m}(x)dx \tag{2}$$

Then, the 3DZD, F_{nl}, is computed by normalizing Ω_{nl}^{m} as follows:

$$F_{nl} = \sqrt{\sum_{m=-l}^{m=l}(\Omega_{nl}^{m})^{2}} \tag{3}$$

where n is the order of 3DZD determining the resolution of the descriptor. The norms of the moments make the descriptor invariant to rotation. For each pair of n and l, 3DZD has a series of invariants, the numbers in the vector of 3DZD, where n is ranged from 0 to the predefined order.

After generation of descriptors for each of the patches, the query pocket is compared to the known pockets in the database where each pocket is a set of surface patches. To compare the query pocket and a pocket in the database (Figure 3), similar patches between the two pockets are determined by a modified bipartite matching algorithm (Step. B). Next, using four features represented as 3DZDs with assigned values, the weighted average distance is computed between matched pairs of patches (Step. C). Then, the relative position difference is computed based on patch distances (Step. D). Finally, the weighted sum of the result scores from step C and D is used as their final distance.

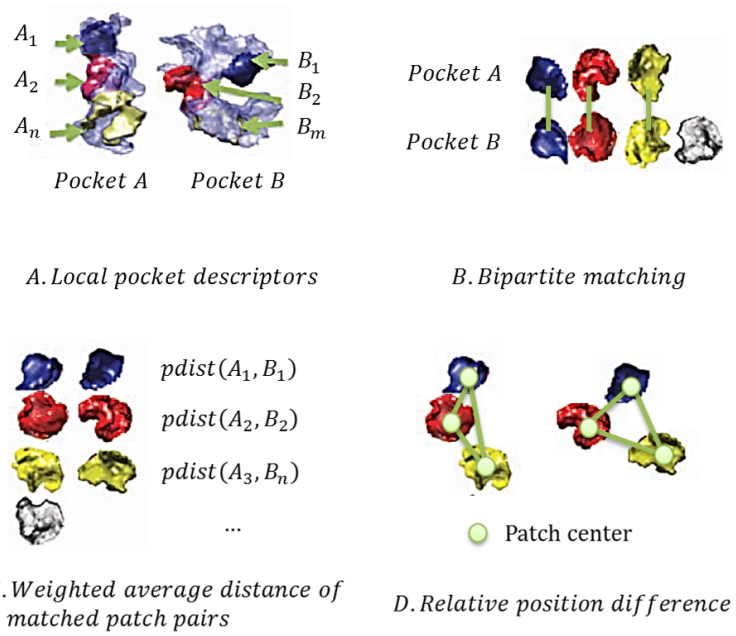

Figure 3 The process of computing similarity between proteins in Patch Surfer. Reproduced from figure 2 (Sael & Kihara, 2012b).

The Patch-Surfer retrieves top-k similar proteins in terms of the four characteristics of local binding-sites which are known to be important in recognition of binding partners. However, this introduces complexity in the computation. First of all, a protein is represented a set of 3DZDs (or vectors), which make comparison much more complex than comparing two vectors as in Pocket-Surfer. Also, the number of available protein structure continues to grow.

3.2 Landmark MDS

As aforementioned, the computational complexity of classical MDS is expensive, since distance matrix is usually not sparse and the computational complexity of the Eigen value decomposition is $O(n^3)$ where the scalability of dataset is n. To resolve this problem, Landmark MDS (LMDS) can be used. LMDS preserves all properties of classical MDS and allows efficient computation as well. Based on a dissimilarity matrix D of l data points, the goal is to embed them in m-dimensional Euclidean space. Specifically, Landmark MDS works in four steps:

1 Select l landmark points.

2 Apply classical MDS to find an embedding of the l landmark points in $\mathbb{R}^m$.

3 Compute the embedding coordinates of the remaining points based on distances to the embedded landmark points.

4 Apply PCA normalization.

Here are more details of each step (Algorithm 1).

Step 1. For selecting landmark points, we take two approaches: 1) random selection and 2) greedy selection. For random selection, a set of points are randomly selected without any duplication. For greedy selection, we first randomly select a small set of points as seed points. After that, landmark points are selected one at a time from all unused data points. At each selection, each new landmark point maximizes the minimum distance to any of the existing landmark points.

Algorithm 1: Greedy selection

```
Data: All data points D
Result: Landmark points L
L = φ
L = L ∪ Rand(l_init, [0, n])
while |L| < l do
    for i = 1 : n do
        d[i] = Min(L, i)
    end
    x = arg max_j(d[j])
    L = L ∪ x
end
return L
```

where n is the number of data points, l is the number of landmark points, l_{init} is the number of seed points, $Rand(l_{init}, [0, n])$ returns indexes of l_{init} randomly selected points from the range of $[0, n]$ as seeds and $Min(L, x)$ returns the minimum distance from any item in L to given point x.

Step 2. This step follows the classical MDS. Given the dissimilarity matrix Δ_l^2 is computed as follows:

$$B_l = -\frac{1}{2} H_l^T \Delta_l^2 H_l \tag{4}$$

where H_l is the mean-centering matrix. Then, the m largest positive eigenvalues of B_l with their eigenvectors are computed, $B_l = V \Lambda V^T$. Lastly, the coordinates of l landmark points in the m-dimensional Euclidean space are given by

$$Y_l = \Lambda^{1/2} V^T \in \mathbb{R}^m \tag{5}$$

Step 3. Based on the embedded landmark points in $\mathbb{R}^m$, the next step is to obtain embedding coordinates of the remaining data points based on their distances from the landmark points. Given a point x, the embedding $\vec{y}_x$ is computed as:

$$\vec{y}_x = \frac{1}{2} Y_l^{\#} (\vec{\delta}_x - \vec{\delta}_\mu) \tag{6}$$

where $\vec{\delta}_x$ is a vector of squared distances between the point x and the l number of landmark points and $Y_l^{\#}$ is pseudo-inverse transpose of Y_l. $\vec{\delta}_\mu$ $(= \vec{\delta}_1 + \vec{\delta}_2 + \cdots + \vec{\delta}_l)/l)$ is the mean

of $\vec{\delta}_i$s which are the vectors of squared distances from the i-th landmark to all the landmark points.

Step 4. The PCA step is not an optional step that, normalizes the embedding coordinates.

Note that the computational complexity of LMDS is $O(nlk + l^3)$ which is much more efficient than the classical MDS.

Algorithm 2: LMDS-based binding-site search

Data: A query protein q, and a dissimilarity matrix D
Result: Top-k result X_k
1 $X_k = \phi$
2 Select l landmark points, L
3 Apply classical MDS and obtain the embedding coordinates of landmark points, Y_l
4 Compute embedding coordinates of the remaining points, Y
5 Compute an embedding coordinate of the query point, y_q
6 $X_k = GetTopK(k, y_q, Y_l[:])$
7 **return** X_k

3.3 LMDS-based binding-site search

Based on Patch-Surfer, we suggest an efficient binding-site search algorithm, called LMDS-based binding-site search, by exploiting LMDS. Given a query protein, q, our algorithm first proceeds LMDS to obtain embedding coordinations of landmark points and other remaining points, $Y = \{y_1, y_2, \cdots, y_n\}$. To select landmark points (Line 1), we take two different approaches. After which, the embedding coordinates of q is computed as Eqn. 6. Using an embedding coordinate of the query, we can find top-k similar proteins by computing the Euclidean distance from all candidate points in database (GetTopK() at Line 5). The detailed algorithm is presented in Algorithm 2.

Although this approach still visits every data point (or it cannot reduce the evaluation ratio), the computation of the Euclidean distance between the embedding coordinates is much more efficient than computing similarity between pockets using the Patch-Surfer alone. Figure 4 is the processing time of Patch-Surfer and LMDS-based binding-site search where k is 25, the number of landmark points is 10, and 100 target queries are randomly selected.

According to the result, the LMDS-based binding-site search reduces the processing time for top-k retrieval. However, accuracy depends on the number of landmark points as well as representativity of the selected landmark points. The accuracy of a query differs when different landmark points are selected as well as when the landmark points are fixed and queries differed (Figure 5).

3.4 LMDS-based binding-site search+

Unfortunately LMDS-based binding-site search did not have high agreement with the original Patch-Surfer in the top-k proteins retrieved. To enhance the accuracy of LMDS-based binding-site search, we add one more step to the previous algorithm. In algorithm 3, k' nearest neighbors are retrieved where $k' > k$ (Line. 6), and among them k nearest neighbors

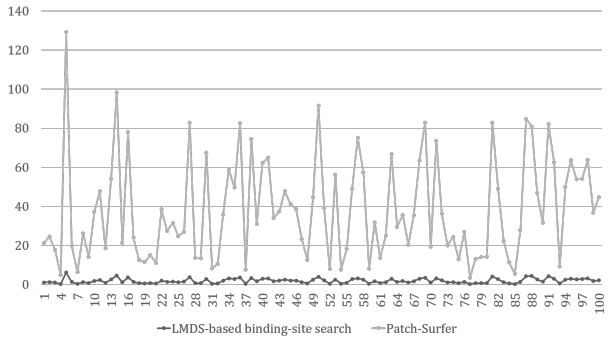

Figure 4 Comparison of Patch-Surfer and LMDS-based binding-site search in the processing time (sec.); x-axis is queries and y-axis is the processing time

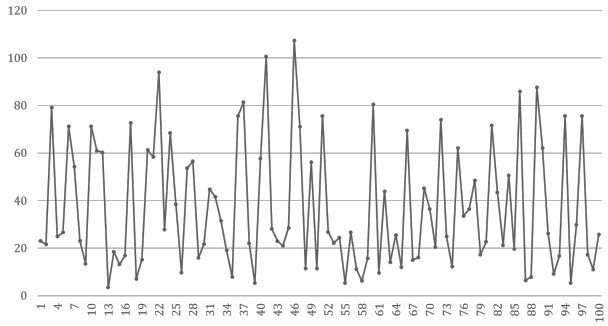

Figure 5 Accuracy of Patch-Surfer; x-axis is queries and y-axis is the number of proteins which are accurately where k is 50

Algorithm 3: LMDS-based binding-site search+

Data: A query protein q, a raw data X, and a dissimilarity matrix D
Result: Top-k result X_k

1 $X_k = \phi$
2 Select l landmark points, L
3 Apply classical MDS and obtain the embedding coordinates of landmark points, Y_l
4 Compute the embedding coordinates of the remaining points, Y
5 Compute the embedding coordinates of the query point, y_q
6 $X'_k = GetTopK(k', y_q, Y_l[:])$
7 $X_k = GetTopK(k, q, X[X'_k])$
8 **return** X_k

are retrieved by computing the original Patch-Surfer distances from query point to candidate nearest neighbors (Line. 7). It retrieves more accurate result than LMDS-based binding-site search does. However, the processing time is increasing since it has to compute the original Patch-Surfer distances. For efficient use of the second method, we have to find sufficient but small k' number for each k of interest.

4 Results

In this section, we provide experimental results for verifying the effectiveness of our methods on top-k search of ligand binding sites. Sael and Kihara (Sael & Kihara, 2012b,a) showed that Patch-Surfer can identify correct pockets even in the absence of known homologous structures in the database. Therefore, we evaluate our algorithm in terms of the efficiency considering the top-k result of the Patch-Surfer as golden standard. The experiments were conducted on the machine, Intel Core(TM) i7 CPU (3.40GHz), and 16 GB memory.

4.1 Dataset

For experiment, we used the same dataset as (Sael & Kihara, 2012a) where it contains 9393 representative pockets with 2707 different ligand types extracted from the Protein Data Bank (PDB). According to the previous work (Sael & Kihara, 2012a), ligand binding-site is represented as the 3DZDs of four features. In this section, we will briefly describe data format. Each data instance consists of six lines as follows:

- 1. The number of patches in the protein, and the dimensions of four 3DZDs
- 2. The weight of four 3DZDs based on their mean and standard deviations. They are used for computing distance and weighted sum for the final score.
- 3. The 3DZD of shape information (or typical 3DZD representation)
- 4. The 3DZD of hydrophobicity
- 5. The 3DZD of electrostatic potential
- 6. The 3DZD of visibility

4.2 Performance dependency on the number of landmark points

In this section, we inspect how the accuracy of LMDS-based binding-site search changes when the number of landmark points, l, is increased. Since our method is an approximation method, the accuracy is measured by how many pockets in the top-k results are retrieved consistently with the Patch-Surfer. More specifically, the accuracy is computed as follows:

$$\text{Accuracy} = \frac{|\hat{X}_k \cap X_k|}{|X_k|} \tag{7}$$

where $\hat{X}_k$ is a set of top-k nearest neighbors retrieved by LMDS-based binding-site search and X_k is a set of true top-k nearest neighbors. According to the result (Fig. 6), our method retrieves more accurate k nearest neighbors with more landmark points, since we can preserve the distance relationship of data points better with more landmark points. Fig. 7 shows the result of speedup ratio between Patch-Surfer and LMDS-based binding-site search. The speedup ratio is computed as follows:

$$\text{Speedup ratio} = \frac{|t_{ps} - t_{lmds}|}{|t_{ps}|} \tag{8}$$

where t_{lmds} is the processing time of LMDS-based binding-site search and t_{ps} is the processing time of Patch-Surfer. Intuitively, the speedup ratio of LMDS-based binding-site search decreases as the number of landmark points increases. This is due to the cost of computing Eigen value decomposition and embedding coordinates which both rely on the number of landmark points. However, LMDS-based binding-site search still retrieves faster than the Patch-Surfer (the speedup ratio always more than 95%).

Also, two landmark selection approaches are comparable in performance. When the number of landmark points is 10, however, random selection works better than greedy selection (Figure 6). Examination of greedy selection shows that when the number of landmark points is small, it tends to select from a small region on the input space. When a new query point is introduced, due to this bias in the landmark point distribution, it is likely that the similarity of query point to all other points have slight differences. However, if we select enough number of landmark points, both the landmark selection approaches are widely dispersed and small differences between positions are sufficiently preserved.

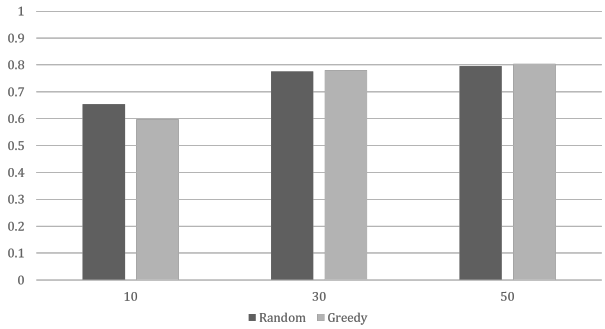

Figure 6 Accuracy of LMDS-based binding-site search; x-axis is the number of landmark points, l, and y-axis is the accuracy where k is 50

Generally, for landmark point selection, greedy selection takes 8.87 seconds on average with standard deviation of 0.98. However, since the landmark selection can be done offline, it does not affect actual the search speed. We do not present the result of speedup ratio of greedy selection since there is no reason to believe it would be different from random selection as in Fig. 7.

4.3 Performance dependency on the number of nearest neighbors

In this section, we observe how the performance are changed dependent on the number of retrieved pockets, k, when the number of landmark points is fixed to 50 (Fig. 8). In most cases, the result of greedy selection has higher accuracies than that of the random selection. It is also interesting to note that the difference between the accuracy of two landmark selection approaches is small when k is large. For example, the difference between k of 50 and 100 is ignorable (both of them are close to 80%). The result of the processing time is not presented since most of the cost depends on Eigen value decomposition of the distance matrix of landmark points so that it does not vary as k differs.

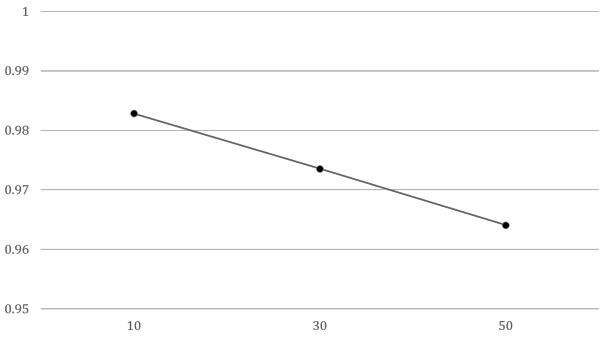

Figure 7 Speedup ratio of LMDS-based binding-site search (Random selection); x-axis is the number of landmark points, l and y-axis is the speedup ratio where k is 50

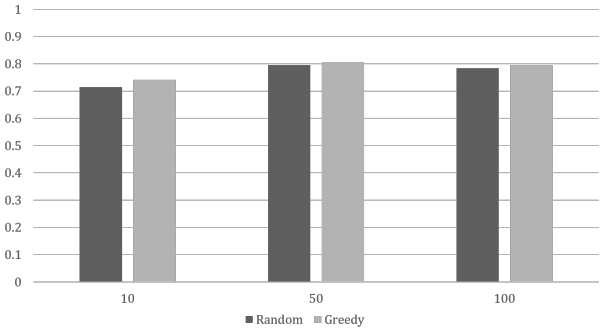

Figure 8 Accuracy of LMDS-based binding-site search with 50 landmark points; x-axis is the number of result, k and y-axis is the accuracy

4.4 Performance of LMDS-based binding-site search+

This section provides the experimental result that shows enhancement of LMDS-based binding-site search+. Previously, LMDS-based binding-site search retrieves only 80% of true top-k nearest neighbors. In this experiment, k is 10 and k' varies from 10 to 50. When k' is 10, the process is same as that of LMDS-based binding-site search. According to the result (Fig. 9), if we set k' to more than $k \times 3$, the result contains more than 90% of exact top-k nearest neighbors. As more nearest neighbors are retrieved, the accuracy becomes even better (about 95%). However, the speedup ratio decreases linearly (Fig. 10), since we have to compute the original Patch-Surfer distances between k' nearest neighbors to find exact top-k nearest neighbors. It shows that there is a tradeoff between the accuracy and the speedup ratio. To summarize, when k is 10 and k' is 30, LMDS-based binding-site search+ retrieves 95% of true top-10 result with 90% speedup ratio compared to the Patch-Surfer.

5 Conclusion

In this work, we proposed a new local binding site search system, called LMDS-based binding-site search. We exploit Landmark Multi-Dimensional Scaling (LMDS), which is an efficient version of MDS being popularly used for representing high-dimensional dataset.

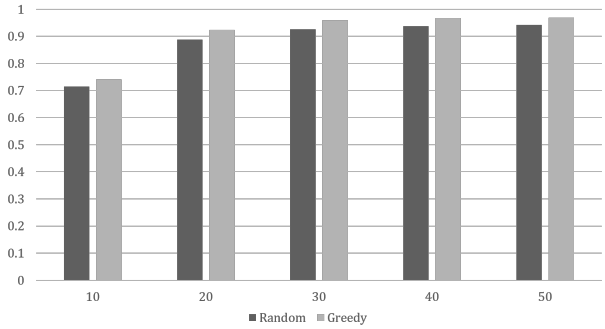

Figure 9 Accuracy of LMDS-based binding-site search with k'; x-axis is k' and y-axis is the accuracy where k is 10, l is 50

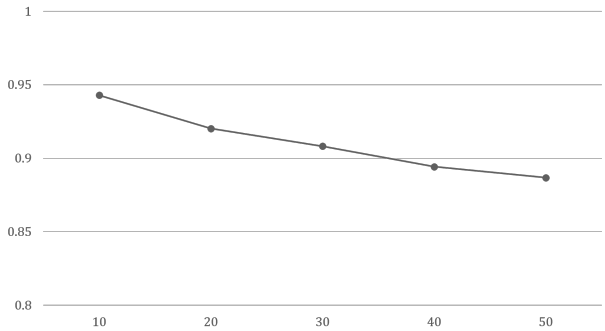

Figure 10 speedup ratio of LMDS-based binding-site search with k'; x-axis is k' and y-axis is the speedup ratio where k is 10, l is 50

We take two approaches for the selection of landmark points: 1) random selection and 2) greedy selection. According to the result, LMDS-based binding-site search approximately retrieves top-k nearest neighbors and its accuracy increases as we exploit more landmark points. Although two landmark selection approaches show comparable result, greedy selection shows the best result when the number of landmark points is sufficiently large. Specifically, using LMDS-based binding-site search with random or greedy selection, it retrieves almost 80% of true k nearest neighbors with upto 99% of the speedup ratio. LMDS-based binding-site search+ retrieves k' nearest neighbors ($k' > k$) with the LMDS-based binding-site search and finds k nearest neighbors using the original Patch-Surfer distances between the k' pockets and the query. It enhances the accuracy upto 95%. However, retrieving more neighbors and computing actual distances between them hurts the efficiency.

Acknowledgement

This work was partly supported by the ICT R&D program of MSIP/IITP [14-824-09-014, Basic Software Research in Human-level Lifelong Machine Learning (Machine Learning Center)], Next-Generation Information Computing Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education Science and Technology (No. 2012M3C4A7033344), Mid-career Researcher Program through NRF

Grant funded by the MEST (NRF-2013R1A2A2A01067425) and, the Industrial Core Technology Development Program (10049079, Development of Mining core technology exploiting personal big data) funded by the Ministry of Trade Industry and Energy (MOTIE, Korea).

References

Abascal, Federico, & Valencia, Alfonso. 2003. Automatic annotation of protein function based on family identification. *Proteins*, **53**, 683–692.

Arakaki, Adrian K., Zhang, Yang, & Skolnick, Jeffrey. 2004. Large-scale assessment of the utility of low-resolution protein structures for biochemical function assignment. *Bioinformatics*, **20**(7), 1087–1096.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., & Sherlock, G. 2000. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature genetics*, **25**.

Canterakis, N. 1999. 3D Zernike Moments and Zernike Affine Invariants for 3D Image Analysis and Recognition. *Pages 85–93 of: In 11th Scandinavian Conf. on Image Analysis*.

Chikhi, Rayan, Sael, Lee, & Kihara, Daisuke. 2010. Real-time ligand binding pocket database search using local surface descriptors. *Proteins*, **78**(9), 2007–2028.

Cox, Trevor F., & Cox, M.A.A. 2000. *Multidimensional Scaling, Second Edition*.

Ferre, Fabrizio, Ausiello, Gabriele, Zanzoni, Andreas, & Helmer-Citterich, Manuela. 2004. SURFACE: a database of protein surface regions for functional annotation. *Nucleic Acids Research*, **32**, 240–244.

Gibrat, Jean-Francois, Madej, Thomas, & Bryant, Stephen H. 1996. Surprising similarities in structure comparison. *Curr. Opi. Struct. Biol.*, 377–385.

Gold, N.D., & Jackson, R.M. 2006. Fold independent structural comparisons of protein-ligand binding sites for exploring functional relationships. *J Mol Biol*, **355**(5), 1112–24.

Hawkins, T., Chitale, M., & Kihara, D. 2008. New paradigm in protein function prediction for large scale omics analysis. *Mol Biosyst*, **4**(3), 223–31.

Hawkins, Troy, & Kihara, Daisuke. 2007. Function prediction of uncharacterized proteins. *Journal of bioinformatics and computational biology*, **5**(1), 1–30.

Kahraman, Abdullah, Morris, Richard J., Laskowski, Roman A., Favia, Angelo D., & Thornton, Janet M. 2010. On the diversity of physicochemical environments experienced by identical ligands in binding pockets of unrelated proteins. *Proteins*, **78**(5), 1120–1136.

Kim, Sungchul, Lee, Sael, & Yu, Hwanjo. 2012. Indexing methods for efficient protein 3D surface search. *Pages 41–48 of: In Proc. of the ACM 6th international workshop on Data and text mining in biomedical informatics*. DTMBIO '12.

Kinoshita, Kengo, Murakami, Yoichi, & Nakamura, Haruki. 2007. eF-seek: prediction of the functional sites of proteins by searching for similar electrostatic potential and molecular surface shape. *Nucleic Acids Research*, **35**, 398–402.

Lee, David, Redfern, Oliver, & Orengo, Christine. 2007. Predicting protein function from sequence and structure. *Nat Rev Mol Cell Biol*, **8**(12), 995–1005.

Morris, R. J., Najmanovich, R. J., Kahraman, A., & Thornton, J. M. 2005. Real spherical harmonic expansion coefficients as 3D shape descriptors for protein binding pocket and ligand comparisons. *Bioinformatics*, **21**(10), 2347–2355.

Platt, John C. 2005. Fastmap, metricmap, and landmark MDS are all nystrom algorithms. *Pages 261–268 of: In Proc. of 10th International Workshop on Artificial Intelligence and Statistics*.

Porter, Craig T., Bartlett, Gail J., & Thornton, Janet M. 2004. The Catalytic Site Atlas: a resource of catalytic sites and residues identified in enzymes using structural data. *Nucleic Acids Research*, **32**, 129–133.

Rosenberg, M., & Goldblum, A. 2006. Computational protein design: a novel path to future protein drugs. *Curr. Pharm. Des.*, 3973–97.

Sael, L., & Kihara, D. 2009. *Protein surface representation and comparison: New approaches in structural proteomics.*

Sael, L., & Kihara, D. 2010. Binding ligand prediction for proteins using partial matching of local surface patches. *Int J Mol Sci*, **11**, 5009–26.

Sael, Lee, & Kihara, Daisuke. 2012a. Constructing patch-based ligand-binding pocket database for predicting function of proteins. *BMC bioinformatics*, **13 Suppl 2**(Suppl 2), S7.

Sael, Lee, & Kihara, Daisuke. 2012b. Detecting local ligand-binding site similarity in nonhomologous proteins by surface patch comparison. *Proteins*.

Samish, Ilan, MacDermaid, Christopher M., Aguilar, Jose Manuel Perez, & Saven, Jeffery G. 2011. Theoretical and Computational Protein Design. *Annual Review of Physical Chemistry*, **62**, 129–149.

Shatsky, M., Peleg, Shulman A., Nussinov, R., & Wolfson, H. J. 2006. The multiple common point set problem and its application to molecule binding pattern detection. *J Comput Biol*, **13**(2), 407–28.

Shindyalov, I. N., & Bourne, P. E. 1997. Protein structure alighment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng.*, 739–747.

Singh, A. P., & Brutlag, D. L. 2008. Hierarchical protein structure superposition using both secondary structure and atomic representations. *Pages 1013–1022 of: Intl. Syst. for Mol. Biol. (ISMB)*.

Skolnick, Jeffrey, & Brylinski, Michal. 2009. FINDSITE: a combined evolution/structure-based approach to protein function prediction. *Briefings in Bioinformatics*, **10**(4), 378–391.

Venkatraman, Vishwesh, Chakravarthy, Padmasini Ramji R., & Kihara, Daisuke. 2009. Application of 3D Zernike descriptors to shape-based ligand similarity searching. *Journal of cheminformatics*, **1**.