An Isometry-Invariant Spectral Approach for Protein-Protein Docking

Dela De Youngster*, Eric Paquet†*, Herna Viktor* and Emil Petriu*
*School of Electrical Engineering and Computer Science
University of Ottawa, Ontario, Canada
Email: {ddeyo084, hviktor}@uottawa.ca, petriu@eecs.uottawa.ca
†Information Technology and Communication
National Research Council of Canada, Ontario, Canada
Email: eric.paquet@nrc-cnrc.gc.ca

***Abstract*—The protein docking problem refers to the task of predicting the appropriate matching of one protein molecule (the *receptor*) to another (the *ligand*), when attempting to bind them to form a stable *complex*. Research shows that matching the three-dimensional geometric structures of proteins plays a key role in determining a so-called docking pair. However, the active sites which are responsible for the binding do not always present a rigid-body shape matching problem. Rather, they may undergo deformations when docking occurs, which complicates the process. To address this issue, we present an isometry-invariant and topologically robust partial shape descriptor method for finding complementary protein sites. Our method employs Heat Kernel Signature shape descriptors which are based on the diffusion of heat on surfaces. Our experimental results against the Protein-Protein Benchmark 4.0 demonstrate the viability of our approach.**

I. Introduction

Proteins are essential to the sustenance of living organisms. They are present in all aspects of life processes performing a multitude of different roles. This ranges from playing the role of enzymes (biochemical catalysts) which rapidly accelerate biological reactions, to being structural components of cells and organisms providing support and also facilitating the passage of certain molecules between different regions of cells [1]. Some key protein functions that are of major interest are their ability to regulate biochemical activities in target cells, to serve as receptors for hormones and various ligands, and to act as modifiers in cell-to-cell interactions. These capabilities are of principal significance because, they are essential in the drug design process when aiming to find cures for diseases and ailments, such as designing antibodies that defend against infections [1].

One specific form of protein-protein interaction which is under active research is *protein-protein docking*, (also known as *protein-ligand docking*, or generically called *"the protein docking problem"*) [1], [2]. The protein docking problem refers to the method of predicting the appropriate pairing and alignment of a protein molecule (*receptor, host* or *lock*) with respect to another protein (*ligand, guest* or *key*) when bound to each other on their *active sites* to form a stable *complex* or *dimer*. Finding two binding proteins is a key step in identifying prospective drug candidates during the process of drug design. The problem of finding appropriate matches is made difficult by the complex geometric structure of proteins and the several thousands of protein sequence entries to be considered. Also, other complicating factors are the several degrees of freedom in terms of alignment and orientation, and the possibility of flexible deformations the active sites may undergo [1], [2]. Fig. 1 gives a simplified illustration of protein docking.

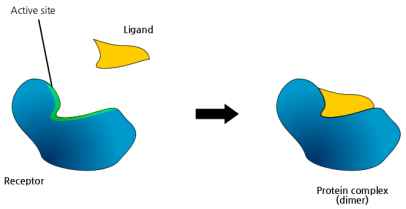

Fig. 1: A simplified illustration of a ligand *(yellow)* docking to a receptor protein *(blue)* to form a stable complex.

II. Related Works

Current techniques that address the protein docking problem fall within two main areas, namely *biological* and *computational*. Biological methods mainly deal with in vitro (laboratory) experimentations and mostly provide definitive results on docking pairs. However, they require very expensive laboratory equipment, appreciable time and human resources. Computational methods, on the other hand, seek to abate such costly necessities by using biological principles, mathematical theories and computing applications to address the protein docking problem [3].

Computational methods may be placed under two broad categories, namely *matching methods* and *docking simulation methods*. Matching methods aim to dock a target ligand structure into a created model of the receptor active site, by comparing their structural geometries. The docking methods, on the other hand, attempt to model the docking process by randomly exploring different translations, orientations and conformations of a target ligand to a receptor protein, with the aim of finding an ideal docking site [4].

Autodock, a widely used approach, is the result of the work of Morris et al [5]. They use three stochastic search methods, namely Lamarckian, traditional genetic algorithms and Monte Carlo simulated annealing, for predicting the bound

978-1-4799-3163-7/13/$31.00 ©2013 IEEE

conformations. Atilgan et al [2] introduce AutoDockX which extends the work of [5] to address the deficiencies arising from local optima and premature convergence issues associated with simulated annealing and traditional genetic algorithms. They observe that these issues are usually offset by doing multiple runs in order to obtain reasonable results, but end up being computationally more expensive and consequently time consuming. They present the Age-Layered Population Structure (ALPS) algorithm, a so-called sustainable genetic algorithm, to address the shortcomings of the core algorithms implemented in AutoDock [5].

Matching approaches, such as the technique developed by Axenopoulos et al [6] adopt a geometry-based shape matching technique, based on so-called Shape Impact Descriptors (SID). This is a rotation-invariant 3D shape descriptor which alleviates the need for repeated searches for initial alignment. The removal of this requirement reduces the computational needs as compared to the relatively more exhaustive free-energy optimization search methods. However, recall that the structures of active sites do not remain rigid, but may bend or contort to enable binding. The applicability of the SID approach is therefore limited, as it is only suitable for rigid body comparisons. The research as presented in our paper addresses this shortcoming, as will be discussed next. To this end, we present our ProtoDock algorithm, which is an isometry-invariant, topologically robust partial deformable shape matching method.

III. ProtoDock Algorithm

Our ProtoDock algorithm performs the partial shape matching in three (3) main phases. First, we divide a given three-dimensional object mesh (corresponding to a given protein) into salient segments. We achieve this process using a spectral segmentation method which performs this partitioning not directly on the three-dimensional mesh, but on its spectra i.e. its eigenvalues and eigenfunctions. Secondly, once the segments are obtained, we create an isometry-invariant, topologically robust shape descriptor for each segment. This descriptor is based on the *Heat Kernel Signature (HKS)* [7]. Thirdly, three different methods—*Bag of Features (BoF)*, *Closest Medoid Set (CMS)* and *Medoid Set Average (MSA)*—are used to obtain the final descriptor vectors from the representative vectors (medoids) after clustering the HKS values. The computed descriptor vectors are used in the final stage of our ProtoDock algorithm to find the matching (and possible docking) pairs between the segments by calculating the distance between the descriptor vectors.

A. Mesh Segmentation using Spectral Analysis

We begin our ProtoDock algorithm by first performing a segmentation on the three-dimensional mesh representation of the proteins by partitioning the mesh into disjoint regions of connected components (i.e. sets of vertices or faces). We employ a spectral analysis approach which performs the segmentations on the eigenvalues and eigenvectors of the underlying mesh. We proceed to obtain the eigensystem by first computing a *Laplacian* L from the graph G induced by the input three-dimensional mesh M. Reduced to a geometric space-partitioning problem on the Laplacian, the segmentation is obtained by embedding the graph G into the space R^k by clustering (using the K-Means algorithm) the first k eigenvectors of the Laplacian.

The continuous definition of the Laplacian or *Laplace-Beltrami Operator* is obtained by letting S be a smooth manifold with a Riemannian metric and with boundary, ∇ denote the gradient. The Laplace-Beltrami operator Δ (or ∇^2) of a given twice continuously differentiable function $f \in C^2$, is the divergence (div) of the gradient (grad) of S in Euclidean space [9]

$$\Delta f = \nabla^2 f = \text{div grad } f = \nabla \cdot \nabla f = \nabla^2 f \tag{1}$$

Consider a discrete domain M as a triangular mesh with n vertices denoted by $M = (V, E, F)$, let V be the set of vertices, with each vertex $i \in M$ denoted in absolute Cartesian coordinates as, $v_i = (x_i, y_i, z_i)$, E be the set of edges, and F be the set of faces. The discrete solution to (2) is approximated by a piecewise function over the triangular mesh $f : M \rightarrow R$. The function f linearly interpolates values of $f(v_i)$ over the vertices of M. The discrete Laplacian is therefore often represented as [10]

$$\Delta f(v_i) = \frac{1}{d_i} \sum_{j \in N(i)} w_{ij}[f(v_i) - f(v_j)] \tag{2}$$

where, $N(i)$ are the members of the immediate neighbourhood of vertex v_i (i.e. the degree or valence of vertex v_i), d_i is the associated mass assigned to vertex v_i, and w_{ij} is the symmetric weight assigned to the corresponding edge between vertex v_i and v_j. A subsequent matrix representation of (3) defines a vector of the function for all the vertices v_i to v_n as $\mathbf{f} = [f(v_i), \ldots, f(v_n)]^{\mathrm{T}}$.

A *weighted adjacency matrix*, $W = (w_{ij})$, contains all the corresponding neighbour edge weights for all the vertices. This matrix is usually symmetric and sparse. Also, consider a *volume matrix* $U = \text{diag}(u_1, ..., u_n)$, which is a diagonal matrix with elements on its leading diagonal u_i defined as $u_i = \sum_{j \in N(i)} w_{ij}$. Given the weighted adjacency matrix W and the volume matrix U, we define the *stiffness matrix* A as $A = U - W$, and the *lumped mass matrix* D as $D = \text{diag}(d_1, ..., d_n)$. Finally, the Laplacian L is defined with respect to the stiffness and lumped mass matrices as

$$L = D^{-1}A \tag{3}$$

We employ the Laplace-Beltrami operator as presented by Desbrun [8] which has been shown to be stable, and also adequately approximates the continuous operator. It is a variation of the cotangent scheme in that, the edge weights w_{ij} are assigned by computing

$$w_{ij} = \frac{\cot(\alpha_{ij}) + \cot(\beta_{ij})}{2} \tag{4}$$

with a corresponding vertex weighting obtained by dividing the area of the neighbouring triangles of a vertex by 3 (i.e. $d_i = \sum_{N(i)} \frac{A_i}{3}$), where A is the area of a given triangle, and $N(i)$ is the neighbourhood of vertex i.

With the Laplacian matrix defined in (4), the Laplacian eigenvalue problem can be written as

$$L\mathbf{f} = \lambda\mathbf{f} \tag{5}$$

Equation (5) can also be expressed as a generalized symmetric eigen decomposition problem with respect to the stiffness and mass matrices as

$$A\mathbf{f} = \lambda D\mathbf{f} \tag{6}$$

With the first n_{eig} smallest eigensystem of the Laplacian computed, we represent its eigenvectors as a column-major eigenvector matrix E_{vec} by stacking the eigenvectors into an $n \times n_{\text{eig}}$ matrix with each row normalized. We derive a segmentation on mesh M by using K-Means clustering to obtain k clusters from E_{vec}, where each row of E_{vec} is treated as a point in n_{eig}–dimensional space. We then map the cluster indices to each vertex v in the mesh. The k set of vertices, each with elements obtained from the connected vertices v assigned to a particular cluster, form a corresponding single object segment.

B. Segment Description using HKS

After obtaining the segments, our algorithm constructs a shape descriptor for each of the individual segments. We employ a feature-based shape descriptor based on the Heat Kernel Signature (HKS) [7]. The underlying concept that serves as the basis for the HKS is the principle of heat diffusion (or propagation) over the surface of three-dimensional objects as completely described by the *heat kernel* associated with the object's Laplace-Beltrami Operator. The HKS serves as a highly informative pointwise descriptor obtained by restricting the heat kernel to the temporal domain over the object. It has been shown to possess several useful properties. These include providing an efficient multi-scale organization of intrinsic geometric information of a given object or shape, being concise and commensurable, and being stable and robust to shape perturbations [7]. The heat propagation of a compact Riemannian manifold M, possibly with boundary, is governed by the heat equation

$$\Delta_M u(x,t) = -\frac{\partial u(x,t)}{\partial t} \tag{7}$$

where Δ_M is the Laplace-Beltrami operator of the manifold M, and u is a continuous smooth function. The Dirichlet boundary condition $u(x,t) = 0,\ \forall x \in \partial M, \forall t$, will have to be satisfied for M with boundaries. By considering only the temporal domain for any given point x on the manifold M, and given the Laplace-Beltrami operator of the manifold has a discrete eigen decomposition of the form $\Delta_M \varphi_i = \lambda_i \varphi_i$, where λ_i and φ_i for $i = 0, 1, 2, \ldots$ are the eigenvalues and eigenfunctions respectively, then the heat kernel (the fundamental solution of the heat equation) can be written as

$$K_t(x,y) = \sum_{i=0}^{\infty} e^{-\lambda t} \varphi_i(x) \varphi_i(y) \tag{8}$$

The subsequent Heat Kernel Signature (HKS) for the given point x can then be given as a compact n-dimensional descriptor vector $\mathbf{p}(x) = (p_1(x), ..., p_n(x))^{\mathrm{T}}$, which contain the elements [11]

$$p_i(x) = c(x) K_{t_i}(x,x) \tag{9}$$

where $c(x)$ is a normalization constant such that $\|\mathbf{p}(x)\|_2 = 1$.

Our algorithm constructs the Heat Kernel Signatures for each of the prior obtained k segments considered as an independent mesh M_i, by first computing the Laplacian and its corresponding eigensystem of each mesh. A *heating times* list $\{t\}$ of time intervals t_{int} starting from an initial value to a set t_{max} is computed. Now, given the obtained eigenvalues, their associated eigenvectors, and the heating times $\{t\}$, for each vertex point x in each segment mesh, we compute the normalized t_{int}-dimensional Heat Kernel Signature descriptor vector $\mathbf{p}(x)$.

C. Descriptor Vector Computation

The HKS for a mesh represents a considerable amount of information, since each point of the mesh is characterised by its own HKS. In order to obtain a more compact descriptor, we proceed by first clustering each HKS. Each cluster is characterised by its medoid, which is the HKS associated with the point that is the closest to the cluster centre as resulting from the application of the K-Means clustering algorithm [11]. The set of the HKSs associated with the medoids $P = \{\mathbf{p}_1, \ldots, \mathbf{p}_l\}$ thus forms the basis for obtaining the shape descriptors. These medoids are used in three different ways to construct three different types of descriptors, namely the Bag of Features, the Closest Medoid Set and the Medoid Set Average methods.

1) ***Bag of Features (BoF):*** For the Bag of Features (BoF) method [11], we consider the medoid set (also called a vocabulary set of "*geometric words*") of size l. For each point x on the segment mesh M, with its corresponding HKS $\mathbf{p}(x)$, we compute the *feature distribution* $\mathbf{\Theta}(x) = (\theta_1(x), \ldots, (\theta_l(x))^{\mathrm{T}}$, an $l \times 1$ vector which is defined as

$$\theta_i(x) = c(x) e^{-\frac{\|\mathbf{p}(x) - \mathbf{p}_i\|_2}{2\sigma^2}} \tag{10}$$

where $c(x)$ is a normalization constant such that $\|\theta(x)\|_2 = 1$, and σ is set to the median of the geometric words.

A final $l \times 1$ feature descriptor vector $\mathbf{J}$ is then obtained by integrating over the entire segment mesh M as $\mathbf{J} = \int_M \mathbf{\Theta}(x) da(x)$. We accomplish this for each segment mesh by first stacking our computed feature distributions $\mathbf{\Theta}(x)$ in row-major order and summing up all the columns of the matrix.

We proceed to obtain possible docking sites by performing a segment matching from different segments of other protein structures by calculating the Euclidean distance between their descriptor vectors. For example, given the descriptor vectors, $\mathbf{J}(M)$ and $\mathbf{J}(N)$ for two segments M and N, the similarity is equal to

$$d_{\text{BoF}}(M,N) = \|\mathbf{J}(M) - \mathbf{J}(N)\|_2 \tag{11}$$

2) ***Closest Medoid Set (CMS):*** We introduce the first of our two novel descriptor methods which we refer to as Closest Medoid Set (CMS). Unlike the Bag of Features method, the CMS method employs the $l \times t_{int}$ normalized medoid set of a given segment as its descriptor, where t_{int} is the time intervals or the number of time steps. The motivation behind employing only the medoid set comes from the observation that, the medoids are representative of the entire mesh to which they belong, given that we have a sufficient number of time intervals and medoids. To this end, we proceed to find the sum of the smallest distances between each pair of the medoids.

Here, the similarity between two segments M and N, each with their respective medoid sets $P_M = \{\mathbf{p}_{M1}, \ldots, \mathbf{p}_{Ml}\}$ and $P_N = \{\mathbf{p}_{N1}, \ldots, \mathbf{p}_{Nl}\}$ is defined by

$$d_{\text{CMS}}(M, N) = \sum_{i}^{l} \frac{\min_{j \in [1,l]} \|\mathbf{p}_{Mi} - \mathbf{p}_{Nj}\|_2}{l^2}. \quad (12)$$

*3) **Medoid Set Average (MSA)**:* The descriptor obtained by the MSA method is closely related to that of the CMS method in that it also employs the medoid set. However, after normalization, we form our final $1 \times t_{int}$ descriptor vector $\mathbf{J}$, from the medoid sets of a given segment mesh by finding the column-wise average of the row-major $l \times t_{int}$ medoid set matrix. The similarity measure between two segments M and N is then calculated by the Euclidean distance between the two descriptors, as in

$$d_{\text{MSA}}(M, N) = \|\mathbf{J}(M) - \mathbf{J}(N)\|_2 \quad (13)$$

This subsection described the three HKS-based methods that we employ to create segment descriptors. Next, we discuss our experimental evaluation.

IV. Results

We conducted a number of experiments to assess the success of our ProtoDock algorithm. We used protein structure data obtained from the Protein Data Bank [12] and the Protein-Protein Benchmark Version 4.0, as presented by Hwang et al [13]. This benchmark contains a total of 124 known protein docking pairs and their subsequent complexes. The benchmark also provides a classification derived from the measure of "*difficulty*" of predicting the pairings for docking algorithms. These three (3) proposed levels of difficulty—*Rigid body conformations, Medium difficulty, and Difficult*— are based on whether, and by how much the active sites of a given pair of proteins undergo flexible deformations after docking. That is, the difficulty is measured by the structural difference between the bound and the unbound forms of the pairing proteins, in terms of an I-RMSD value. I-RMSD is defined as the Root Mean Square Distance/Deviation (RMSD) between superimposed bound and unbound structures calculated using the interface residue $C\alpha$ atoms of both binding proteins. Rigid body conformations (i.e. the least difficult) are noted as having an I-RMSD $\leq 1.5\mathring{A}$ ($\mathring{A}$ is a unit length of 0.1 nanometers), with Medium difficulty conformations having $1.5\mathring{A} <$ I-RMSD $\leq 2.2\mathring{A}$, and Difficult ones having I-RMSD $> 2.2\mathring{A}$.

For our experiments, we consider complexes from all three of the aforementioned difficulty categories. Our algorithm utilizes three-dimensional meshes obtained from the *molecular surface* (also known as the *Solvent Excluded Surface (SES)*) representation of the proteins using BALLView [14]. The molecular surface is computed using the *rolling ball* algorithm with employs the use of a sphere of a certain radius to probe along the surface of the Van Der Waal atoms of a given molecular/protein structure. The trace of the closest point of the solvent probe when rolling along the surfaces of the atoms forms the molecular surface or SES. Figure 2 shows a molecular surface representation of the protein 1ZM8 (Nuclease A). Also, as noted above, we employ the discrete

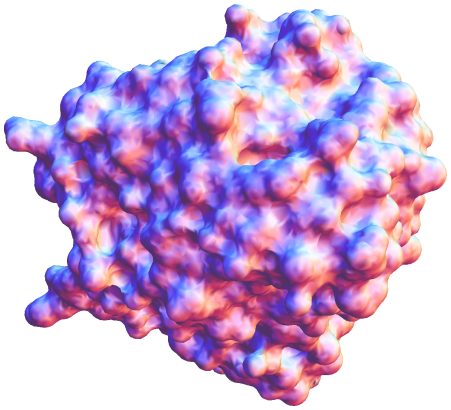

Fig. 2: A molecular surface (Solvent-Excluded Surface (SES)) representation of the 1ZM8 (Nuclease A) protein structure.

TABLE I: Protein structures and their known docking pairs

DIFFICULTY LEVEL	COMPLEX	RECEPTOR	LIGAND
Rigid Body	1GL1	1K2I	1PMC
	1JTG	3GMU	1ZG4
	2ABZ	3I1U	1ZFI
Medium	1MQ8	1IAM	1MQ9
	1SYX	1QGV	1L2Z
	1R6Q	1R6C	2W9R
Difficult	1F6M	1CL0	2TIR
	2IDO	1J54	1SE7
	2O3B	1ZM8	1J57

Laplace-Beltrami Operator as presented by Desbrun [8]. This method uses the area of neighbouring triangles for vertex weighting, and the cotangent scheme for edge weighting in constructing the corresponding stiffness and mass matrices. We select the first 100 eigenvalues and their associate eigenvectors. A time interval of 5 steps with a maximum time of 100 is chosen when computing the Heat Kernel Signatures. Also, medoid set sizes of 50 and 80 are selected for the segment mesh sizes of 3,500 and 5,000 vertices, respectively. These values were set by inspection.

Table I shows the list of protein structures used in our experimentation. We select three (3) complexes each from the three (3) difficulty groups. We also consider a BALLView meshing resolution of 3.5 when generating the molecular surface meshes. We first provide a visual illustration of a generated pairing showing the segment meshes that were matched. Figure 3 shows the pairings obtained from 1ZM8-2TIR and 3GMU-1ZG4 respectively, which both appropriately return the first closest matching segment from the benchmark.

Table II and Table III show the ranking of the closest matching pairs obtained from our experimentation for segments of average vertex count of 3,500 and 5,000 for the three (3) descriptor methods, namely the Bag of Features (BoF), Closest Medoid Set (CMS) and Medoid Set Average (MSA). We show the number of segment comparisons performed and also rank the closest match for each of the receptor-ligand proteins for the three descriptor methods. Note that the number of segment comparisons range from 521 to 832. Recall that the distance measure used for computing the similarity is the

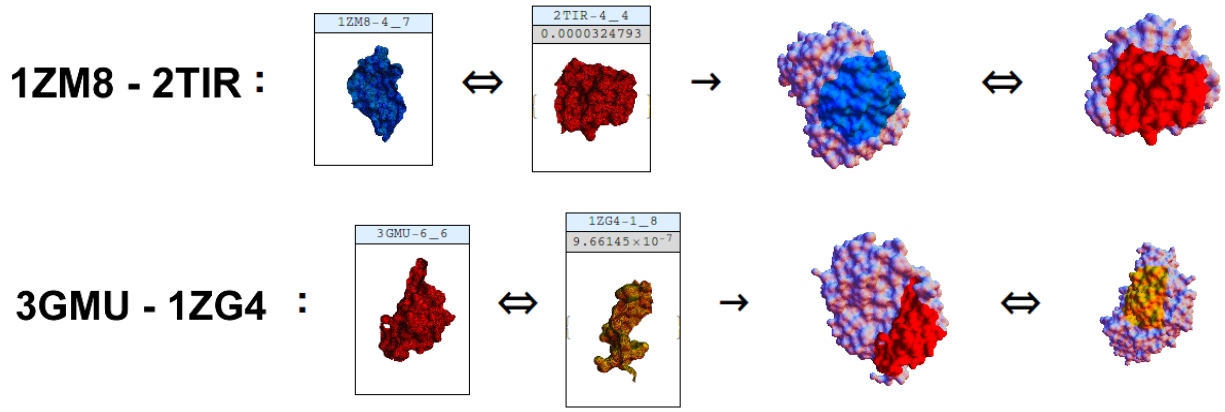

Fig. 3: Closest matching segments for the 2O3B and 1JTG complexes, from the pairings of 1ZM8 to 1J57 and 3GMU to 1ZG4.

TABLE II: Protein-protein Segment Matching at Average Mesh Vertex Count of 3,500

DIFFICULTY	PAIRING	SEG. COMPARISONS	BoF	CMS	MSA
Rigid Body	1K2I-1PMC	704	9	20	22
	3GMU-1ZG4	576	2	1	4
	3I1U-1ZFI	832	18	3	39
Medium	1IAM-1MQ9	704	1	16	1
	1QGV-1L2Z	512	1	1	2
	1R6C-2W9R	576	11	5	7
Difficult	1CL0-2TIR	640	10	12	8
	1J54-1SE7	576	7	17	8
	1ZM8-1J57	640	8	3	6

TABLE III: Protein-protein Segment Matching at Average Mesh Vertex Count of 5,000

DIFFICULTY	PAIRING	SEG. COMPARISONS	BoF	CMS	MSA
Rigid Body	1K2I-1PMC	308	8	9	12
	3GMU-1ZG4	264	5	1	1
	3I1U-1ZFI	396	1	34	22
Medium	1IAM-1MQ9	352	11	12	22
	1QGV-1L2Z	220	1	4	24
	1R6C-2W9R	264	11	6	4
Difficult	1CL0-2TIR	770	36	8	16
	1J54-1SE7	308	4	20	13
	1ZM8-1J57	308	10	4	6

Euclidean distance. Figure 4 shows the corresponding line graphs for Table II and Table III.

The results in Table II show that the three descriptor methods are all able to rank the known pairings. Specifically, for the matching of the six pairs that are considered difficult or moderately difficult, the Bag of Features (BoF) and Medoid Set Average (MSA) methods correctly rank the matches within the first 11 closest pairing. For example, for the 1QGV-1L2Z pairing, there were 512 segment comparisons performed, and both the BoF and CMS methods ranked the correct segments from the receptor and the known ligand as the closest match. The MSA technique also performs well, by e.g. ranking the correct pair for 1IAM-1MQ9 first.

From Table III, we observe complimentary retrieval results when the average vertex count is increased to 5,000 with a subsequent surface area per segment of about 6.25×10^8 Å^2. That is, in 63.0% (17/27) of the cases, the retrieval rates when using an average vertex count of 3,500 are equal, or higher, to the results as presented in Table III. However, given the variations in docking sites for certain pairing proteins, we note that the rankings for e.g. the 1J54-1SE7 and 1CL0-2TIR pairings, show closer matches for this larger segment size comparisons for the BoF and CMS methods, respectively. This indicates that the resultant area of about 4.37×10^8 Å^2, forms an adequate segment size for comparing the segments. This also suggests that an increase in the number of segment sizes may produce more insights into the possible matching segments, given that it will sufficiently cover the variations in the sizes of the docking sites for different protein complexes.

In summary, our results show that our ProtoDock algorithm is able to accurately find the correct matches. The three variations of the descriptor methods complement one another well. Given the varying sensitivity of the different descriptor methods, consideration may be given to obtaining a single rank value for each segment matched, based on the different ranks returned by each descriptor method. In this way, the final cumulative rank may be guided by considering the confidence assigned to each descriptor method, similar to ensemble-based procedures frequently used in Machine Learning [16].

V. Conclusion

We have presented our isometry-invariant deformable shape matching algorithm for addressing the protein-protein docking problem. Our approach employs Heat Kernel Signature descriptors which are based on the heat diffusion on surfaces. Our results show the viability of our method, especially being used fruitfully as a pre-filtering technique to the more computationally expensive but exhaustive methods. Future work will concentrate on finding the final geometric conformation between retrieved matching segments. We are also interested

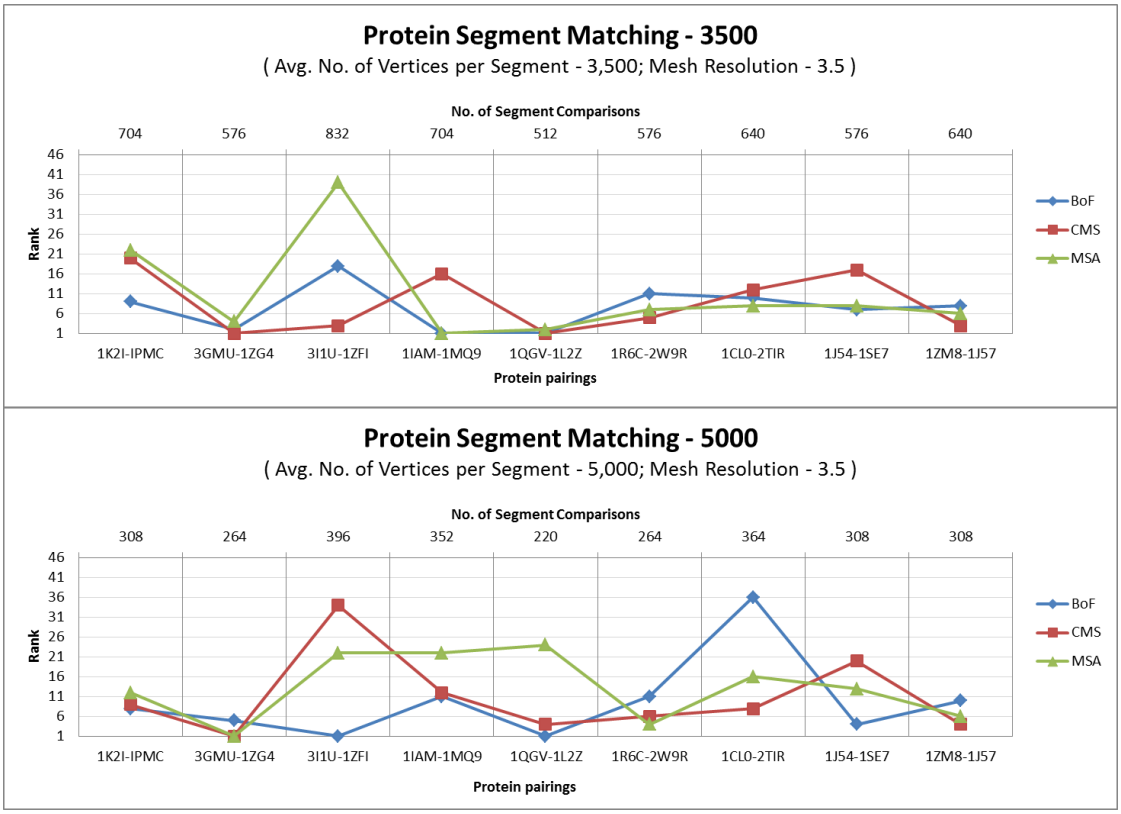

Fig. 4: A graph showing the rank of the closest matching segments for each known protein pairing (and the number of segment comparisons) with average vertex count of 3,500 (first) and 5,000 (second) for BoF, CMS and MSA descriptor methods.

in investigating the appropriateness of ensemble-based approaches [16]. Here, the general idea would be to use a form of majority or weighted voting during ranking. Subsequently, the final cumulative rank is influenced by the confidence assigned to each descriptor method. Interest also lies in finding intelligent computational methods for selecting the different input parameters (e.g. heating time intervals and the word count) when computing the Heat Kernel Signatures. Also, different Laplace-Beltrami Operators and distance measures will be considered.

REFERENCES

[1] Moran, L., Horton, R., Scrimgeour, G., Perry, M.: *Principles of Biochemistry*, 5th Edition. Publication Date: Sep 1 2011, ISBN-10: 0321707338, ISBN-13: 978-0321707338, Edition: 5 (2011)

[2] Atilgan, E., Jianjun H.: *Efficient Protein-ligand Docking using Sustainable Evolutionary Algorithms*, Hybrid Intelligent Systems (HIS), 2010 10th International Conference, vol., no., pp.113,118, 23-25 Aug. doi: 10.1109/HIS.2010.5600082, (2010)

[3] Lengauer, T., Rarey, M.: *Computational Methods for Biomolecular Docking*. Curr Opin Struct Biol, Vol. 6, No. 3., pp. 402-406 (1996)

[4] Rosenfeld, R., Vajda, S., DeLisi, C.: *Flexible Docking and Design*. Annu. Rev. Biophys. Biomol. Struct., 24, 677 (1995)

[5] Morris, G. M., Goodsell, D. S., Halliday, R. S., Huey, R., Hart, W. E., Belew, R. K., Olson, A. J.: *Automated Docking using a Lamarckian Genetic Algorithm and Empirical Binding Free Energy Function*. Journal of Computational Chemistry 19 (14): 1639-1662. (1998)

[6] Axenopoulos, A., Daras, P., Papadopoulos, G., Houstis, E.: *3D Protein-protein Docking Using Shape Complementarity and Fast Alignment*. Image Processing (ICIP), 2011 18th IEEE International Conference, vol., no., pp.1569,1572, 11-14 Sept. (2011)

[7] Sun, J., Ovsjanikov, M., Guibas, L.: *A Concise and Provably Informative Multi-scale Signature Based on Heat Diffusion*. In Proceedings of the Symposium on Geometry Processing, pp. 1383-1392. (2009)

[8] Desbrun, M., Meyer, M., Schrder, P., Alan, H. Barr.: *Implicit Fairing of Irregular Meshes using Diffusion and Curvature Flow*. In Proceedings of the 26th Annual Conference on Computer graphics and interactive techniques (SIGGRAPH '99). (1999)

[9] Wardetzky, M., Mathur, S., Klberer, F., Grinspun, E.: *Discrete Laplace operators: No Free Lunch*. ACM SIGGRAPH ASIA 2008 courses, pp. 1-5, doi:10.1145/1508044.1508063. (2008)

[10] Reuter, M., Biasotti, S., Giorgi, D., Patan, G., Spagnuolo, M.: *Discrete LaplaceBeltrami Operators for Shape Analysis and Segmentation*. Computers & Graphics, Vol. 33, No. 3. June, pp. 381-390, doi:10.1016/j.cag.2009.03.005. (2009)

[11] Ovsjanikov, M., Bronstein, A., Bronstein, M., Guibas, L.: *Shape Google: A Computer Vision Approach to Isometry Invariant Shape Retrieval*. pp. 320-327, doi:10.1109/iccvw.2009.5457682. (2009)

[12] RSCB Protein Data Bank (PDB), http://www.pdb.org

[13] Hwang, H., Vreven, T., Janin, J., Weng, Z.: *ProteinProtein Docking Benchmark Version 4.0*. Proteins, Vol. 78, No. 15, pp. 3111-3114, doi:10.1002/prot.22830. (2010)

[14] Moll A, Hildebrandt A, Lenhof HP, Kohlbacher O. J.: *BALLView: An Object-oriented Molecular Visualization and Modeling Framework.* Comput Aided Mol Des. Nov;19(11):791-800. (2005)

[15] Bronstein, A. M., Bronstein, M. M., Kimmel, R., *Calculus of non-rigid surfaces for geometry and texture manipulation*, IEEE Trans. Visualization and Computer Graphics, Vol 13/5, pp. 902-913, September-October (2007)

[16] Dietterich, T.G.: *Ensemble Methods in Machine Learning*, In J. Kittler and F. Roli(Ed.) First International Workshop on Multiple Classifier Systems, pp.1-15. Springer Verlag (2000)