BIOINFORMATICS

Vol. 00 no. 00 2005
Pages 1–8

Multi-view Methods for Protein Structure Comparison using Latent Dirichlet Allocation

S. Shivashankar, S. Srivathsan**, B. Ravindran, Ashish V Tendulkar*

Department of Computer Science and Engineering, IIT Madras, Chennai-600 036
**Department of CSE, Anna University, Chennai-600 025

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

ABSTRACT

Motivation: With rapidly expanding protein structure databases, efficiently retrieving structures similar to a given protein is an important problem. It involves two major issues: (i) effective protein structure representation that captures inherent relationship between fragments and facilitates efficient comparison between the structures (ii) effective framework to address different retrieval requirements. Recently, researchers proposed vector space model of proteins using bag of fragments representation (FragBag), which corresponds to the basic information retrieval model.

Results: In this paper, we propose an improved representation of protein structures using Latent Dirichlet Allocation (LDA) topic model. Another important requirement is to retrieve proteins, whether they are either close or remote homologs. In order to meet diverse objectives, we propose multi-viewpoint based framework that combines multiple representations and retrieval techniques. We compare the proposed representation and retrieval framework on the benchmark dataset developed by Kolodny and co-workers. The results indicate that the proposed techniques outperform state-of-the-art methods.

Availability: http://www.cse.iitm.ac.in/∼ashishvt/research/protein-lda/

Contact: ashishvt@cse.iitm.ac.in

1 INTRODUCTION

Following huge efforts from the structural genomics research community, protein structures databases are ever expanding. Whenever a new protein structure is determined, an important step is to identify its structural neighbours, which can provide important clues about its function and evolutionary linkages. Since the protein structure is relatively more robust than sequence during evolution, structure comparison methods can discover remote homologous proteins for a given protein. They also play a key role in understanding the diversity of structure space by analyzing the existing structure databases. In order to derive interesting scientific insights from the vast structure databases available now, more efficient methods for comparing protein structures are required.

The success of structure comparison methods can be measured based on their effectiveness in detecting closely and remotely homologous proteins [19]. The closely homologous proteins have similar structures with relatively less insertions and deletions. On the other hand, remote homologs possess significantly different structures. The similarity in these cases can be inferred based on similarity of structural fragments. Comparing protein structures at the fragment level has been shown to work well in practice. The first fragment based comparison method was proposed by Remington and Mathews, which performs rigid body superposition of fixed length backbone fragments from individual proteins [13]. The rigid body superposition was later used by Zuker and Somorjai to define distance between backbone fragments while comparing them using dynamic programming [22]. The fragment based structure comparison was also used in identification and ranking of local features [8]. For further details, the readers are referred to an excellent review paper by Taylor et al [19]. Recently researchers proposed an interesting vector space representation of protein structures using fragments as the bases [2]. The method, fragbag, appears to perform the task of retrieving similar structures efficiently and is the state-of-the-art method in fragment based structure comparison.

Fragbag represents each structure as a bag of fragments, which is the most basic model of retrieval proposed in text literature. The success of FragBag opens up many interesting avenues, where advanced language modeling techniques proposed in Information Retreival (IR)/Statistical Natural Language Processing (NLP) can be adopted for representing protein structures to achieve better performance in terms of efficiency and accuracy of identifying structural homologs. The paper focuses on two important problems in this context: (i) effective protein structure representation that captures inherent relationship between fragments and facilitates efficient comparison between the structures (ii) effective framework to address different retrieval requirements. We propose a new representation for protein structures based on latent dirichlet allocation (LDA) [1]. LDA models a collection of discrete objects as a mixture of latent topics, and has been shown to work remarkably well in text and image retrieval domain. The success of LDA in text domain is attributed to effective capturing of relationship between words. Drawing a parallel, here we demonstrate that LDA indeed models relationships between fragments in protein structures effectively and achieves a competitive performance with state-of-the-art structural methods at a fraction of the computation costs. Another important contribution of this work is that we propose multi-viewpoint homology detection framework to effectively find close, as well as, remote homologous proteins for a query protein structure. This is the first attempt to adopt advanced models from IR and statistical NLP for addressing protein structure comparison problem.

*to whom the correspondence should be addressed

2 RELATED WORK

Several methods have been proposed in literature for protein structue comparison. These methods compare a pair of protein structures, compute a quantitative measure of similarity and most often generate a structural alignment. Taylor et al [19] have compiled a comprehensive review describing challenges in protein structure comparison and its importance along with various proposed methods. The proposed methods differ on the following two broad points: choosing appropriate representation and the algorithm for efficient and accurate retrieval of homologous structures from the database.

The popular choices for representations include (i) complete three dimensional coordinate information or partial coordinate information of backbone atoms, (ii) representation of various elements using their properties such as ϕ-ψ angle, solvent accessibility, etc. The first type of representation preserves sequential and topological relationships between individual elements of the structure. The methods developed to compare the first type of representation are partitioned into two: the ones using dynamic programming (DP) [16, 18] and others not using DP [7, 17]. These methods are computationally expensive and do not scale well for large number of structures. Moreover, a large number of these comparisons do not yield results since the structures are not related. To overcome these problems, researchers proposed a two stage approach widely known as the filter and match paradigm. The first stage of this approach employs very fast filtering algorithm to obtain a small set of proteins which are most likely to be similar. These proteins are subjected to rigorous and computationally expensive structure alignment methods in the second stage. These methods achieve the desired speed in filtering stage by representing proteins as vectors and comparing them in the space spanned by appropriate descriptors or features. For instance, method proposed by Choi et al [3] and PRIDE makes use of the distance matrix to represent the structure. Rogen and Fein represented protein with 30 topological features of backbone using knot invariants [15]. Zotenko et al represented a protein structure as a vector of the frequencies of structural models, each of which is a spatial arrangement of triplets of secondary structure elements [21]. There are several other methods which proposed interesting and novel feature based structure representation and comparisons such as Friedberg et al [4], Tung et al[20], etc.

3 PROPOSED APPROACH

As mentioned in the introduction, the framework for protein structure comparison has two subproblems to be handled. In this section, we elaborate the proposed framework to address these subproblems. The proposed techniques draw a huge motivation from statistical NLP.

3.1 Representation of proteins in topic space

The key point of the proposed approach is to represent proteins as probability distributions over latent topics. Note that the topic is an abstract concept and is represented as a multinomial distribution over fragments. Given this representation, a collection of protein structures can be modeled using three-level hierarchical Bayesian generative model known as latent dirichlet allocation (LDA) [1]. Intuitively, this formalism clusters similar fragments into topics, which provides significant advantage over models that perform fragment to fragment comparision (expect identity) while comparing protein structures. We explain this concept with a simple example. Suppose we are interested in comparing two documents, one containing words **dog** and **cat** and the other containing **bark** and **mews**. Naive word level comparison of the two documents reveal that they are unrelated, when they actually talk about semantically related topics (dog-barking and cat-mews in this case). This example can be extended to protein structures, where fragments are entities equivalent to words in the document. The fragments are grouped into a topic in a probabilistic manner and the search for homologous proteins can be performed more accurately in the topic space. Before introducing formal aspects of the problem formulation, we describe the key ingredients:

1. A *fragment* f_i is the basic unit of protein structure. It is part of the fragment library of choice F. F=$\{f_1, f_2, \ldots, f_L\}$, where L is the size of fragment library F.
2. A *Protein* is a sequence of n fragments, denoted by $S = \{ f_i | f_i \epsilon F\}$. The protein structure is converted into a sequence of fragments using the method described in [2].
3. A *Universe* is a collection of N proteins, denoted by U=$\{s_1, s_2, \ldots, s_N\}$.

The graphical model representation of LDA is provided in Figure 1. It models the protein structure collection according to the following generative process:

1. Pick a multinomial distribution φ_z for each topic z from a Dirichlet distribution with parameter β.
2. For each protein s, pick a multinomial distribution θ_s from a Dirichlet distribution with parameter α.
3. For each fragment f_i in protein structure s, pick a topic $z \epsilon \{1,\ldots, \mathrm{K}\}$ with parameter θ_s.
4. Pick fragment f_i from the multinomial distribution φ_z.

According to the model, each protein is a mixture of latent variables z (referred to as clusters/topics), and each latent variable z_i is a probability distribution over fragments. Given N proteins, K topics, L unique fragments in the collection, we can represent $p(f|z)$ for the fragment f, with a set of K multinomial distributions φ over the L fragments, $P(f|z = j) = \varphi_f^{(j)}$. $P(z)$ is modeled with a set of N multinomial distributions θ over K topics. One way to achieve is to use Expectation Maximization to find the estimates of φ, and θ. It suffers from local maxima issues, and its hard to model an unseen protein since it does not assume anything about θ. LDA overcomes these issues by assuming a prior distribution on θ and φ to provide a complete generative model. It uses Dirichlet distribution [1] for choosing priors α for θ and β for φ.

The likelihood of generating a universe of protein structure collections is
$P(s_1, s_2, \ldots, s_N)$ =

$$\int\int \prod_{z=1}^{K} P(\varphi_z|\beta) \prod_{s=1}^{N} P(\theta_s|\alpha) \left(\prod_{i=1}^{N_p} \sum_{z_i=1}^{K} P(z_i|\theta) P(f_i|z,\varphi)\right) d\theta d\varphi$$

[1] Dirichlet prior is a conjugate prior of multinomial distribution

Exact inference is LDA model is intractable and hence a number of approximate inference techniques such as variational methods [1], expectation propagation [6], and Gibbs sampling [5, 6] have been proposed in literature. We use Gibbs sampling based inferencing to estimate φ and θ. From a sample, $\hat{\varphi}$ and $\hat{\theta}$ are approximated using following equations after a fixed number of iterations, which is commonly known as burn in period.

$$\hat{\varphi} \approx (n_{i,j}^{(w_i)} + \beta_{w_i}) / \sum_{v=1}^{V} (n_{i,j}^{(v)} + \beta_v) \quad (1)$$

$$\hat{\theta} \approx (n_{i,j}^{(s_i)} + \alpha_{z_i}) / \sum_{t=1}^{T} (n_{i,j}^{(s_i)} + \alpha_t) \quad (2)$$

Here, $n_{i,j}$ is the number of instances of fragment f_i, assigned to topic $z = j$. α and β are hyper-parameters that determine the smoothness of the distribution. $n_{i,j}^{(s_i)}$ is the number of fragments in protein s_i that belong to topic $z = j$. Thus, the total number of fragments assigned to topic $z = j$ is given by $\sum_{v=1}^{V} n_{i,j}^{(v)}$. The total number of fragments in protein s_i is given by $\sum_{t=1}^{T} n_{i,j}^{(s_i)}$. The terms, $\sum_{v=1}^{V} n_{i,j}^{(v)}$ and $\sum_{t=1}^{T} n_{i,j}^{(s_i)}$, are normalizing factors.

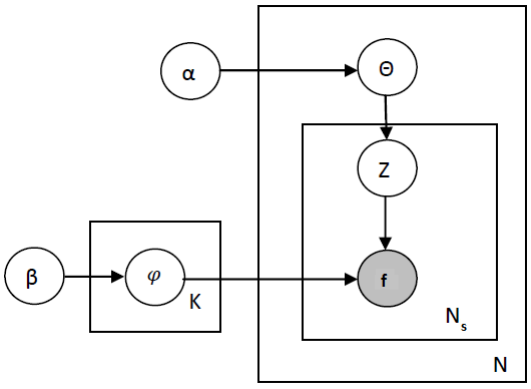

Fig. 1. Graphical representation of LDA; K is the number of topics; N is the number of protein structures; N_s is the number of fragments in protein structure s.

The workflow for building topic model is as follows:

1. We take collection of protein structures as an input. We process each structure and obtain the corresponding fragment by matching its substructures with the library. At the end of this process, we obtain a bag of fragments for each protein. This process is depicted in Figure 2.
2. We learn the topic model on the collection using the machinery described earlier in this section.
3. Each protein is then represented as a probability distribution over the latent topics discovered by LDA.

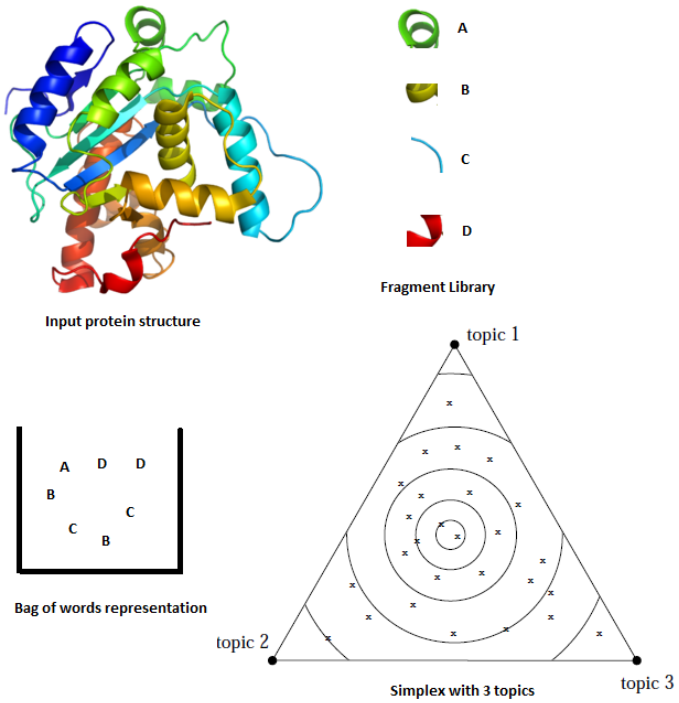

Fig. 2. Example protein structure with bag of fragments and topic space representations; built for a given fragment library. (a) shows an example protein structure and (b) shows a given fragment library. Each substructure in protein is compared against the fragment library and the closest matching fragment is used to represent the substructure. Thus, we obtain bag of fragments representation for protein structure as shown in (c). We model the structure as a probability distribution over latent topics. In (d) we have shown a toy representation using three topics, which forms a simplex.

3.2 Multi-viewpoint based Retrieval

A simple framework for protein retrieval is given in Figure 3, where the universe of proteins are modeled using the representation R chosen. In order to rank the proteins based on the structural similarity for a query protein, the query protein is modeled and transformed to the same representation space R. Once the transformation is done, the protein structures in the collection are ranked based on their structural similarity with the query protein using a retrieval technique. Most simplest technique would involve a boolean vector representation for each protein. Here, the fragments from the fragment library are matched against the protein structure, and a vector of size of the fragment library is built. The vector has 1 in the position of fragments that are present and 0 in the place of fragments that are absent. And retrieval can be based on Jaccard Coefficient [12]. It can be replaced by other IR techniques such as term frequency (TF), term frequency-inverse document frequency (TF-IDF), etc. The similarity metrics must be chosen according to the choice of representation [12]. We refer to this family of techniques as naive vector space models.

As mentioned earlier, the retrieval might have different objectives for different applications. For example, retrieving proteins that are similar, whether they are close homologs or remote homologs. Text

based IR researchers have shown that retrieval based on combination of multiple query representations, multiple representations of text documents, or multiple IR techniques provide significantly improved results compared to single representation based technique, especially when there are multiple retrieval requirements across users. These techniques are referred to as multi-viewpoint based IR in literature [14]. Schema of multi-viewpoint IR is given in Figure 4. The intuition behind doing this is: retrieval information about an author, publication or book would require exact keyword match, but querying based on topics, for example "sports news", must allow more than just keyword match. Motivated by the success of multi-viewpoint based text IR works, we propose a multi-viewpoint based retrieval system for protein structure collection. Protein structure similarity can be captured by not only matching fragments in the protein structure, but similar fragments (not just identity) must also be considered to help protein structure comparison. This is achieved by modeling the protein structure using LDA, which maps the fragments to a topic space using their co-ocurrence information. Protein structure comparison at topic space performs a soft matching by considering similar fragments too.

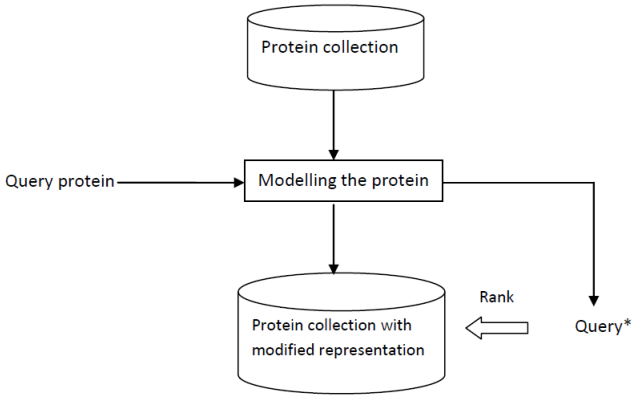

Fig. 3. Typical IR model

The proposed model combines the plain vector (boolean or frequency based) representation of fragments in protein structure and topic space representation using LDA. Query protein and proteins in the collection are transformed into a naive vector space model and LDA representation. The retrieval techniques for both the modeling methods are different. Let us assume a simple boolean representation and a cosine similarity metric for the naive vector space model. Cosine similarity between two protein structures represented using boolean vectors A, B is given below

$$FragSimilarity = cos(\theta) = \frac{A.B}{||A||||B||}$$

We refer to the similarity based on naive vector representation as $FragSimilarity$. LDA based representation uses the asymmetric KL divergence measure to rank proteins. Asymmeric KL divergence between two proteins represented by the topic distribution vector P and Q is given below

$$D_{KL}(P||Q) = \sum_i P(i) log \frac{P(i)}{Q(i)}$$

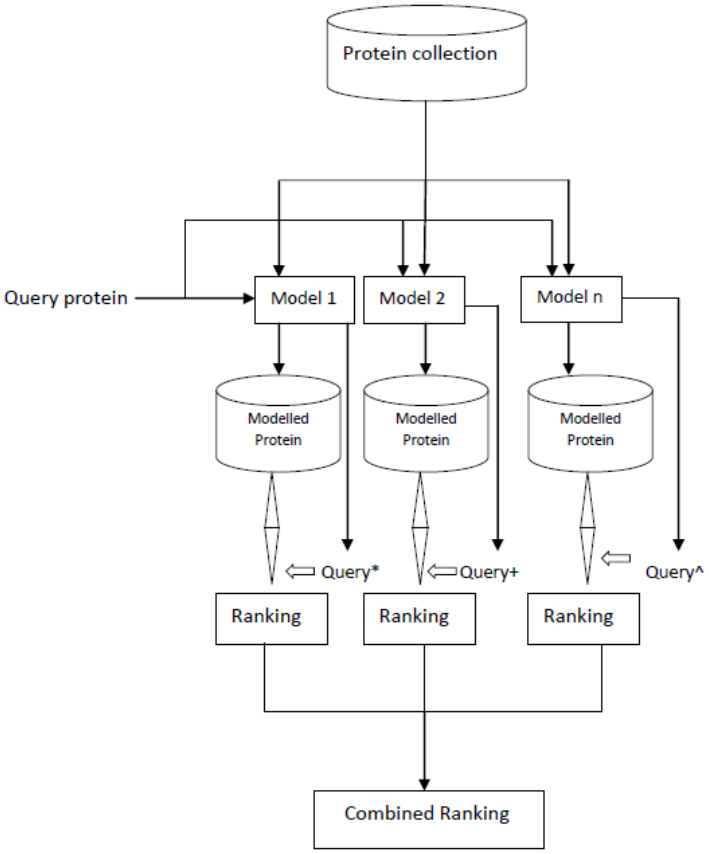

Fig. 4. Multi-viewpoint based IR

The ranking based on these two techniques are combined using a weighted combination of similarity values. KL divergence captures the distance and not similarity value as in the case of cosine similarity. The range of values for cosine similarity$(0, 1)$ and KL divergence $(-\infty, 0)$ are different. KL divergence values are normalized using min-max normalization to get normalized KL divergence measure D_{KL}^{norm}, and converted into similarity value by performing $1 - D_{KL}^{norm}$. Finally, the values are combined as follows

$$Similarity = \lambda_1 * FragSimilarity + \lambda_2 * (1 - D_{KL}^{norm})$$

λ_1 and λ_2 denote the relative weight for the retrieval schemes based on vector representation and LDA respectively. The model can also be extended to more representation schemes, where $0 \leq \lambda_i \leq 1$, and $\sum_i \lambda_i = 1$. Comparison of various vector representations and similarity metrics are given in the experimental results section.

4 EXPERIMENTAL RESULTS

The experiments are performed on the FragBag dataset [2]. Dataset has 2,930 sequence-nonredundant structures, and we treat each structure as a query. The gold-standard was built by [10, 2], by using a best-of-six structural aligner (using SSAP [18], STRUCTAL, DALI [7], LSQMAN [9], CE [17], and SSM); structural

neighbors of a query protein are the ones that are aligned to it with a structural alignment score (SAS) below a threshold T (for T =2 $\mathring{A}$, 3.5 $\mathring{A}$, and 5 $\mathring{A}$). We use area under the curve (AUC) of the ROC curve to measure the performance of ranking, and we average the AUC values across 2,930 queries. AUC ranges between 0 and 1, and higher the AUC indicates better the ranking mechanism. The protein structures from FragBag dataset are converted into BoW representation by matching the structures against the given libraries of fragments. We consider 7 libraries out of the 24 given by Kolodny et al [10]. For each fragment size, the library which was reported to have the best results are chosen for benchmarking in this work. The final libaries chosen include 100(5), 300(6), 250(7), 600(9), 600(10), 400(11), and 400(12). LDA is built on the BoW representation. The experimental results are given as follows

1. In FragBag, it was shown that the BoW representation performs best on 400(11) library. We use this library to choose the best similarity measure for the LDA's probability distribution vector. The experiments are performed with different distance measures $Dist$, such as Cosine distance (CO), Euclidean distance (EU) and KL divergence (KL). The results are shown for SAS=2, SAS=3.5 and SAS=5 in Table 1. It can be seen that KL and CO are the best for SAS=2, 5. KL is slightly better than CO for SAS=3.5. Further analysis are carried out using KL since it outperforms CO and EU in most choices of the number of topics.

	Topics								
Dist	10	100	150	200	250	300	400	500	SAS
KL	0.85	0.89	0.9	0.9	0.9	0.9	0.9	0.9	
EU	0.84	0.87	0.88	0.88	0.87	0.87	0.87	0.87	**2**
CO	0.85	0.89	0.89	0.89	0.9	0.9	0.9	0.9	
KL	0.71	0.77	0.77	0.78	0.77	0.78	0.78	0.77	
EU	0.69	0.71	0.73	0.73	0.73	0.73	0.72	0.72	**3.5**
CO	0.70	0.73	0.76	0.76	0.76	0.77	0.77	0.77	
KL	0.68	0.69	0.69	0.69	0.68	0.68	0.68	0.67	
EU	0.67	0.67	0.67	0.66	0.66	0.65	0.65	0.65	**5**
CO	0.68	0.69	0.69	0.69	0.69	0.69	0.69	0.68	

Table 1. Comparison of average AUC for different similarity measures using the LDA representation with 400(11) library.

2. Ideal number of topics was chosen based on the average AUC obtained for the similar proteins retrieval task. Experimental results for SAS threshold 2, 3.5 and 5 are given in Table 2. Further analyses are performed using the best number of topics for each library. It can be seen that 200-250 topics is the best choice across libraries.
3. As mentioned in Section 3.2, multi-viewpoint IR combines naive vector space model and LDA. In this section, we identify the best naive vector space model from the choices of term frequency (TF), term frequency inverse document frequency (TF-IDF) and boolean (Bool) vectors. Cosine similarity is chosen as the similarity metric for vector space model, since it has been shown to be the best in literature for these representations. The AUC score for different λ values are given in Table 5. The values are computed on 400(11) library, which gives the best results across different number of LDA topics (from Table 2). In Table 3, **I** refers to the multi-view model combining TF and LDA, **II** refers to combining TF-IDF and LDA, and **III** refers to combining Bool and LDA. Multi-viewpoint IR with TF and TF-IDF as the naive vector space model perform better than FragBag. Overall, combining TF and LDA gives the best results. The experiments are repeated for other libraries using the best model (TF and LDA). The results on other libraries are given in Table 4, 5, 6 for SAS threshold 2, 3.5 and 5 respectively. λ_1, the weight for LDA representation is shown in tables. The weight for naive vector space model λ_2, which is 1-λ_1 is not shown in tables.

	Topics								
Library	10	100	150	200	250	300	400	500	SAS
100(5)	0.83	0.85	0.86	0.88	0.88	0.87	0.86	0.84	
300(6)	0.84	0.85	0.86	0.88	0.88	0.88	0.87	0.85	
250(7)	0.85	0.87	0.88	0.89	0.9	0.89	0.89	0.89	
600(9)	0.85	0.89	0.9	0.9	0.9	0.9	0.89	0.88	**2**
600(10)	0.85	0.88	0.9	0.9	0.9	0.9	0.9	0.88	
400(11)	0.85	0.89	0.9	0.9	0.9	0.9	0.9	0.9	
400(12)	0.84	0.89	0.9	0.9	0.9	0.89	0.89	0.87	
100(5)	0.70	0.73	0.73	0.74	0.73	0.72	0.72	0.74	
300(6)	0.71	0.74	0.75	0.76	0.76	0.76	0.75	0.75	
250(7)	0.71	0.75	0.75	0.76	0.76	0.76	0.76	0.75	
600(9)	0.72	0.77	0.77	0.78	0.78	0.78	0.77	0.77	
600(10)	0.72	0.77	0.77	0.78	0.78	0.77	0.77	0.77	**3.5**
400(11)	0.71	0.77	0.77	0.78	0.77	0.78	0.77	0.77	
400(12)	0.71	0.77	0.77	0.77	0.76	0.76	0.76	0.76	
100(5)	0.67	0.68	0.68	0.68	0.67	0.67	0.67	0.67	
300(6)	0.66	0.67	0.68	0.68	0.68	0.68	0.68	0.67	
250(7)	0.66	0.69	0.69	0.68	0.68	0.68	0.67	0.67	
600(9)	0.68	0.69	0.7	0.69	0.68	0.68	0.68	0.67	
600(10)	0.67	0.69	0.69	0.69	0.68	0.68	0.68	0.66	**5**
400(11)	0.68	0.69	0.69	0.69	0.68	0.68	0.68	0.67	
400(12)	0.69	0.7	0.7	0.7	0.69	0.69	0.69	0.67	

Table 2. Comparison of the average AUC obtained with different number of LDA topics

Influence of the weights λ_1 and λ_2 on the retrieval performance is given in Figure 5. It can be seen that for SAS threshold 2 and 3.5, best performance is achieved with higher weight for LDA representation (λ_1). For SAS=5, best performance is achieved with higher weight for naive vector space model (or at least equal to LDA representation). SAS threshold 2, are for close homologs, and 5 denotes remote homologs. LDA based representation works fine to identify close homologs better than remote homologs. Since the fragment similarity is less as we move up the parent tree for a protein structure, exact match using naive vector space model performs well. Motivated by the fact that the best results are spanning libraries, an ensemble on ranking is attempted. For a query protein structure, similarity produced by a model, say using library 400 (11), and weights $\lambda_1 = 0.6$, $\lambda_2 = 0.4$ is treated as an independent hypothesis. Output of each model (combination of libraries and λ_1, λ_2 values) is treated as a hypothesis. The best 3 hypotheses

are chosen and are combined using $bucket\ of\ models$ strategy. For example, let model X and Y provide similarity values between query protein q and a protein s_d in database, denoted by $sim_X(q, s_d)$ and $sim_Y(q, s_d)$ respectively. Bucket of model chooses the similarity between q and s_d, using X and Y as given below

$$sim(q, s_d) = \max(sim_X(q, s_d), sim_Y(q, s_d))$$

This is referred to as $Combined\ Model$. We tested it by combining best models across SAS thresholds. The best models are 600(9) with weights (0.8, 0.2) for SAS=2; 400(11) with weights (0.7, 0.3) for SAS=3.5; 400(11) with weights (0.4, 0.6) for SAS=5. Results of the $Combined\ Model$ are given in the final results table (Table 9).

	SAS=2			SAS=3.5			SAS=5		
λ_1	**I**	**II**	**III**	**I**	**II**	**III**	**I**	**II**	**III**
0	0.89	0.87	0.8	0.77	0.72	0.64	0.75	0.73	0.68
0.1	0.89	0.87	0.81	0.78	0.73	0.65	0.75	0.73	0.69
0.2	0.9	0.88	0.81	0.78	0.74	0.66	0.75	0.73	0.69
0.3	0.9	0.88	0.82	0.79	0.75	0.67	0.75	0.74	0.7
0.4	0.91	0.89	0.83	0.79	0.76	0.69	0.77	0.74	0.7
0.5	0.91	0.9	0.85	0.8	0.77	0.7	0.75	0.74	0.71
0.6	0.91	0.9	0.86	0.8	0.78	0.72	0.75	0.73	0.71
0.7	0.91	0.9	0.88	0.8	0.78	0.75	0.75	0.73	0.71
0.8	0.91	0.91	0.89	0.8	0.79	0.77	0.74	0.72	0.71
0.9	0.91	0.9	0.9	0.8	0.78	0.77	0.72	0.7	0.7
1	0.9	0.9	0.9	0.78	0.78	0.78	0.68	0.68	0.68

Table 3. Comparing the average AUC for various Multi-viewpoint IR methods

λ_1	400(12)	600(10)	600(9)	250(7)	200(6)	100(5)
0	0.88	0.88	0.88	0.87	0.85	0.86
0.1	0.89	0.89	0.89	0.88	0.86	0.86
0.2	0.89	0.89	0.89	0.88	0.86	0.86
0.3	0.9	0.9	0.9	0.88	0.86	0.86
0.4	0.9	0.9	0.9	0.89	0.87	0.86
0.5	0.9	0.91	0.91	0.89	0.88	0.87
0.6	0.91	0.91	0.91	0.89	0.88	0.87
0.7	0.91	0.91	0.91	0.9	0.89	0.87
0.8	0.91	0.91	0.91	0.9	0.89	0.87
0.9	0.9	0.91	0.91	0.9	0.89	0.87
1	0.89	0.9	0.9	0.89	0.88	0.87

Table 4. Comparison of models built on different libraries for SAS threshold=2

4. In order to show the effectiveness of LDA based representation over BoW representation [2], we compare their performance on classification and clustering tasks. It can be seen that LDA representation performs better than the BoW for both the tasks,

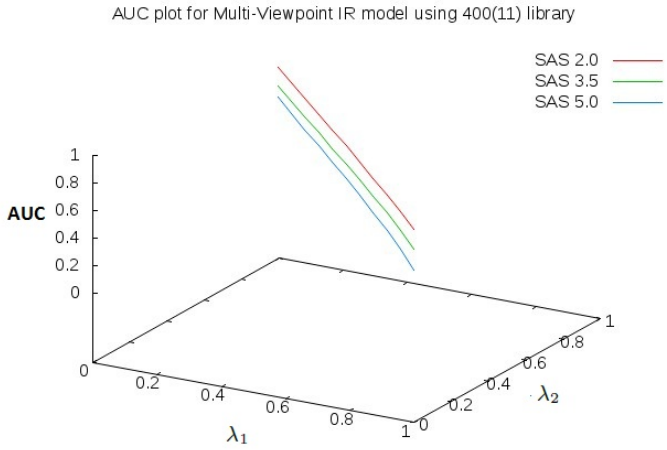

Fig. 5. Impact of weights in multi-viewpoint based IR model

λ_1	400(12)	600(10)	600(9)	250(7)	200(6)	100(5)
0	0.76	0.76	0.76	0.74	0.69	0.72
0.1	0.77	0.77	0.77	0.75	0.7	0.72
0.2	0.78	0.77	0.77	0.75	0.71	0.72
0.3	0.78	0.78	0.78	0.76	0.72	0.73
0.4	0.79	0.78	0.79	0.76	0.73	0.73
0.5	0.79	0.79	0.79	0.76	0.74	0.73
0.6	0.79	0.8	0.8	0.77	0.75	0.74
0.7	0.8	0.8	0.8	0.77	0.75	0.74
0.8	0.8	0.8	0.8	0.78	0.76	0.74
0.9	0.79	0.79	0.8	0.77	0.76	0.75
1	0.77	0.77	0.78	0.76	0.76	0.74

Table 5. Comparison of models built on different libraries for SAS threshold=3.5

λ_1	400(12)	600(10)	600(9)	250(7)	200(6)	100(5)
0	0.76	0.75	0.75	0.75	0.71	0.73
0.1	0.76	0.75	0.75	0.75	0.72	0.73
0.2	0.76	0.75	0.75	0.75	0.72	0.73
0.3	0.76	0.76	0.76	0.76	0.72	0.73
0.4	0.76	0.76	0.76	0.76	0.73	0.73
0.5	0.76	0.76	0.76	0.76	0.73	0.73
0.6	0.76	0.76	0.76	0.76	0.72	0.73
0.7	0.76	0.75	0.75	0.75	0.72	0.72
0.8	0.75	0.74	0.74	0.74	0.71	0.72
0.9	0.73	0.73	0.73	0.73	0.7	0.71
1	0.69	0.69	0.69	0.69	0.68	0.68

Table 6. Comparison of models built on different libraries for SAS threshold=5

in terms of time taken and standard measures for the tasks. Table 7 has the comparison results for classification at C level classes in CATH hierarchy (4 classes). Since the dataset chosen is sparse at other levels of CATH hierarchy (has less than 10 members for most classes at A, T, H levels), we perform classification only at C level. Radial Basis Function network (RBF) and Naive Bayesian(NB) classifiers are used for comparison. Results are compared in terms of RMSE (root mean squared error), ROC, and accuracy. The values are obtained by averaging results across 10 fold cross validation. Table 8 has the comparison results in terms of SSE (sum of squared error) for K Means algorithm using both BoW and LDA representations.

	BoW	LDA	BoW	LDA	BoW	LDA	BoW	LDA
	RMSE		ROC		Accuracy		Time (sec)	
RBF	0.25	0.23	0.93	0.95	83.9	85.7	6.84	2.5
NB	0.33	0.31	0.9	0.922	78.6	80.6	0.58	0.19

Table 7. Comparing the performance of BoW and LDA on the classification task

	BoW	LDA
K	SSE	
4	8556.336	3371.61
10	8531.03	3417.21
20	8154.35	3348.29
50	7872.79	3093.44
100	7455.93	2880.1

Table 8. Comparing BoW and LDA for the clustering task

The performance of LDA representation and retrieval based on asymmetric KL, multi-view retrieval using TF and LDA (multi-view model I) are compared against naive vector space model with cosine similarity on the seven libraries chosen. For multi-view based retrieval, the best weight combination (λ_1 and λ_2) for each library is chosen for the plot. The results are shown in Figure 6, 7, 8 for SAS threshold 2, 3.5 and 5 respectively. Table 9 has the overall ranking of structural and filter methods, which includes the relative positioning of proposed techniques. ([11]). The speed is given as average CPU minutes per query. If the processing time (after preprocessing of protein structure) for a query is less than 0.1s, then it is mentioned as $fast$. The proposed approaches are shown in bold. Multi-view model I refers to using TF and LDA, Multi-view model II refers to TF-IDF and LDA, and Multi-view model III refers to combining Boolean vector and LDA. It is clear that our method outperforms all the filter-and-match methods. We did a paired t-test, paired sign test with AUC values of each query obtained using proposed models and baseline state-of-the-art filter-and-match method (FragBag). We found that the performance results summarized in Table 9 are statistically valid with significance level above 95%. Our results are very competitive even with state-of-the-art structure comparison methods operating at the level of complete three dimensional representation. It must be noted that our method is much faster than these methods.

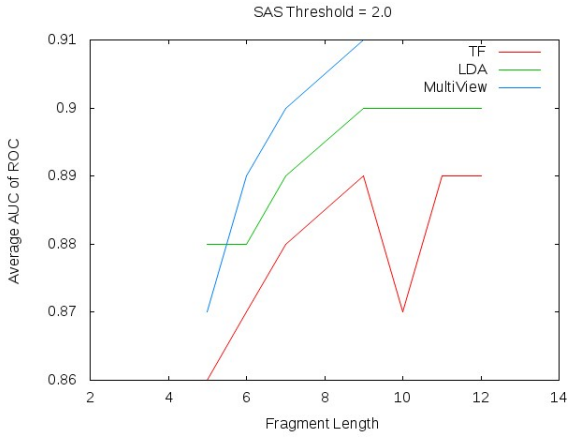

Fig. 6. LDA and TF based multi-view model's performance at SAS threshold 2.0

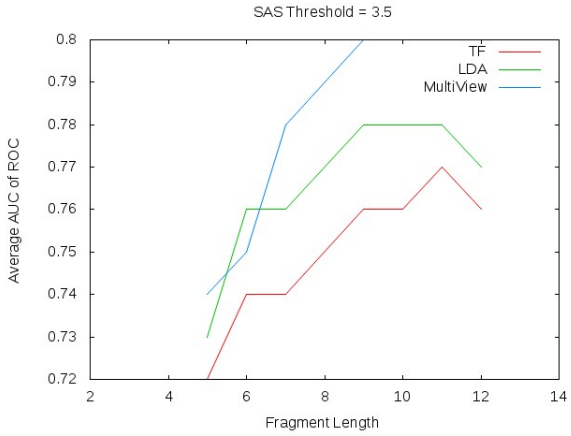

Fig. 7. LDA and TF based multi-view model's performance at SAS threshold 3.5

5 CONCLUSION

We proposed a novel framework for representation and comparison of protein structures. We demonstrated that our method outperforms most of the existing filter-and-match methods. Our results are very competitive even with the state-of-the-art structure comparison methods operating at the level of complete three dimensional representation. Moreover, our method is much faster than these methods. Kolodny and co-workers first proposed the use of IR techniques in protein structure comparison [2]. In this work, we have shown

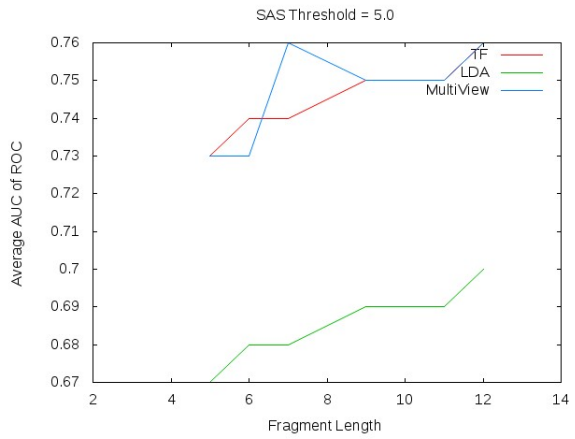

Fig. 8. LDA and TF based multi-view model's performance at SAS threshold 5.0

significant improvements by adapting more detailed models from statistical NLP literature to this task. We also take advantage of simpler models through the proposed multi-view framework and built a system that can be tuned to the retrieval objectives at hand. This work has firmly established that such fragment based models can be competitive with the structural methods. It also has opened the doors for deeper analysis, using techniques from statistical NLP, of the role that fragments play in determining the overall structure.

ACKNOWLEDGEMENT

Dataset and Tools : We thank Rachel for providing us the protein structure dataset and fragment libraries. Arun Konagurthu for fit3D code which matches a fragment with protein structure in three dimensional space.

Funding: AT's research is supported by Innovative Young Biotechnologist Award grant 2008 from Department of Biotechnology, Government of India.

REFERENCES

[1]David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[2]Inbal Budowski-Tal, Yuval Nov, and Rachel Kolodny. FragBag, an accurate representation of protein structure, retrieves structural neighbors from the entire PDB quickly and accurately. *Proc Natl Acad Sci U S A*, 107:3481–3486, 2010.

[3]In-Geol Choi, Jaimyoung Kwon, and Sung-Hou Kim. Local feature frequency profile: a method to measure structural similarity in proteins. *Proc Natl Acad Sci U S A*, 101:3797–3802, 2004.

[4]Iddo Friedberg, Tim Harder, Rachel Kolodny, Einat Sitbon, Zhanwen Li, and Adam Godzik. Using an alignment of fragment strings for comparing protein structures. *Bioinformatics*, 23:e219–e224, 2007.

[5]Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, November 1984.

[6]T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl. 1):5228–5235, April 2004.

[7]L. Holm and C. Sander. Mapping the protein universe. *Science*, 273:595–603, 1996.

[8]M. E. Karpen, P. L. de Haseth, and K. E. Neet. Comparing short protein substructures by a method based on backbone torsion angles. *Proteins*, 6:155–167, 1989.

[9]G. J. Kleywegt. Use of non-crystallographic symmetry in protein structure refinement. *Acta Crystallogr., Sect. D: Biol. Crystallogr*, 52:842–857, 1996.

[10]Rachel Kolodny, Patrice Koehl, and Michael Levitt. Comprehensive evaluation of protein structure alignment methods: scoring by geometric measures. *J Mol Biol*, 346:1173–1188, 2005.

[11]M. Kosloff and R. Kolodny. Sequence-similar, structure-dissimilar protein pairs in the PDB. *Proteins*, 71(2):891–902, May 2008.

[12]Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[13]B. W. Matthews, S. J. Remington, M. G. Grütter, and W. F. Anderson. Relation between hen egg white lysozyme and bacteriophage T4 lysozyme: Evolutionary implications. *J Mol Biol*, 147:545–558, 1981.

[14]Allison L. Powell and James C. French. The potential to improve retrieval effectiveness with multiple viewpoints. Technical report, Charlottesville, VA, USA, 1998.

[15]Peter Rogen and Boris Fain. Automatic classification of protein structure by using Gauss integrals. *Proc Natl Acad Sci U S A*, 100:119–124, 2003.

[16]A. Sali and T. L. Blundell. Definition of general topological equivalence in protein structures. A procedure involving comparison of properties and relationships through simulated annealing and dynamic programming. *J Mol Biol*, 212:403–428, 1990.

[17]I. N. Shindyalov and P. E. Bourne. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng*, 11:739–747, 1998.

[18]W. R. Taylor and C. A. Orengo. Protein structure alignment. *J Mol Biol*, 208:1–22, 1989.

[19]William R Taylor, Alex C W May, Nigel P Brown, , and Andras Aszodi. Protein structure: geometry, topology and classification. *Reports on Progress in Physics*, 64:517–590, 2001.

[20]Chi-Hua Tung, Jhang-Wei Huang, and Jinn-Moon Yang. Kappa-alpha plot derived structural alphabet and BLOSUM-like substitution matrix for rapid search of protein structure database. *Genome Biol*, 8:R31–R31, 2007.

[21]Elena Zotenko, Dianne P. O'Leary, and Teresa M. Przytycka. Secondary structure spatial conformation footprint: a novel method for fast protein structure comparison and classification. *BMC Struct Biol*, 6:12–12, 2006.

[22]M Zuker and R. L. Somorjai. The alignment of protein structures in three-dimensions. *Bulletine of Mathematical Biology*, 51:55–78, 1989.

Methods	SAS=2	SAS=3.5	SAS=5	Average	Rank	Speed
SSM using SAS score	0.94	0.9	0.89	0.91	1	13
Structal using SAS score	0.9	0.81	0.84	0.85	2	39
Combined Model	0.92	0.82	0.75	**0.83**	3	Fast
Structal using native score	0.87	0.77	0.83	0.823	4	39
Multi-view model I (400,11)	0.91	0.8	0.76	**0.823**	4	Fast
CE using native score	0.9	0.79	0.74	0.81	6	54
Multi-view model II (400,11)	0.9	0.78	0.73	**0.803**	7	Fast
FragBag Cos distance (400,11)	0.89	0.77	0.75	0.803	7	Fast
Multi-view model III (400,11)	0.89	0.77	0.7	**0.787**	9	Fast
CE using SAS score	0.84	0.72	0.75	0.77	10	54
FragBag histogram intersection (600,11)	0.87	0.73	0.7	0.767	11	Fast
SGM	0.86	0.71	0.68	0.75	12	Fast
FragBag Euclidean distance (40,6)	0.86	0.71	0.64	0.737	13	Fast
Zotenko et al (18)	0.78	0.64	0.66	0.693	14	Fast
Sequence matching by BLAST e-value	0.76	0.57	0.5	0.61	15	Fast
PRIDE	0.72	0.54	0.51	0.59	16	Fast

Table 9. AUCs of ROC Curves Using Best-of-Six Gold Standard