Improved Protein Decoy Selection via Non-Negative Matrix Factorization

Nasrin Akhter, Kazi Lutful Kabir, Gopinath Chennupati, Raviteja Vangara, Boian S. Alexandrov, Hristo Djidjev, and Amarda Shehu

Abstract—A central challenge in protein modeling research and protein structure prediction in particular is known as decoy selection. The problem refers to selecting biologically-active/native tertiary structures among a multitude of physically-realistic structures generated by template-free protein structure prediction methods. Research on decoy selection is active. Clustering-based methods are popular, but they fail to identify good/near-native decoys on datasets where near-native decoys are severely under-sampled by a protein structure prediction method. Reasonable progress is reported by methods that additionally take into account the internal energy of a structure and employ it to identify basins in the energy landscape organizing the multitude of decoys. These methods, however, incur significant time costs for extracting basins from the landscape. In this paper, we propose a novel decoy selection method based on non-negative matrix factorization. We demonstrate that our method outperforms energy landscape-based methods. In particular, the proposed method addresses both the time cost issue and the challenge of identifying good decoys in a sparse dataset, successfully recognizing near-native decoys for both easy and hard protein targets.

Index Terms—Decoy selection, non-negative matrix factorization, protein structure prediction, protein model quality assessment.

1 Introduction

Protein molecules are ubiquitous in the cell and participate in virtually all cellular processes. Their central role continues to motivate research in the wet laboratory in understanding protein function. Due to the central role that the three-dimensional/tertiary structure plays in governing the biological activity of a protein [1], determining biologically-active/native tertiary structures of a protein is often a first, critical step to decoding protein function [2]. Resolving tertiary protein structures in wet laboratories is challenging and costly, and manifests itself in a large disparity between millions of protein-encoding gene-sequences and the far lesser number of experimentally-resolved native structures ($147,193$ as of December 2019); this disparity has prompted complementary approaches in dry laboratories. Template-free protein structure prediction (PSP) methods address the most challenging setting of novel protein sequences with no known structural templates from homologous sequences [3].

Template-free PSP methods generate many low-energy three-dimensional/tertiary structures under the assumption that near-native structures are more likely to be associated with low energies. This assumption is based on seminal discoveries by Anfinsen et al [4], but the energy functions employed in silico are semi-empirical and inherently biased. Research has shown that energy is an unreliable indicator of nativeness [5]. For this reason, identifying one or more near-native structure(s) from an ensemble of decoys generated by a template-free method, a problem known as *decoy selection*, remains an outstanding challenge in protein modeling research and, broadly in computational structural biology.

In decoy selection methods that discriminate based on the energies are moderately effective; hence, independent estimation of the "near-nativeness" of decoys has become a necessity; the central question is what makes a decoy near-native? Accurate prediction of near-native decoys is essential in advancing the state of protein structure prediction, which prompted the community-wide experiment "Critical Assessment of Techniques for Structure Prediction" (CASP) to include decoy/model quality estimation, estimation of model accuracy (EMA), as an independent category in CASP 7 (2006). EMA methods are assessed using their ability to identify best model from several available. GDT-TS is one such popular metric [6], which we use among other metrics in this paper. To that end, we propose multi-model (clustering-based) methods to identify the best model.

Clustering-based (multi-model) methods have dominated the EMA category until recently. These methods employ the hypothesis that decoys are randomly distributed around the "true" answer (the native structure), and so a consensus method is likely to discover near-native decoys. The hypothesis is flawed, and its veracity is keenly tied to the quality of the distribution of decoys generated by a structure prediction method. In particular, the sparsity of good-quality decoys for a hard target makes it challenging for a clustering-based method to identify near-native decoys [7]. In response, single-model methods have emerged. These methods predict the quality of one decoy at a time using physics-based [8], knowledge-based [9] energetic properties of decoys, or a combination of both, and often employ machine learning techniques [10]. While steadily improving, the performance of single model methods may depend on the specific evaluation score that they are trained on. Consensus methods are more balanced in this regard [6].

• *N. Akhter, K.L. Kabir and A. Shehu are with the Department of Computer Science, George Mason University, Fairfax, VA, 22030, USA. G. Chennupati and H. Djidjev are with the Information Sciences (CCS-3) Group at Los Alamos National Laboratory, Los Alamos, NM, USA. R. Vangara and B. S. Alexandrov are with the Physics & Chemistry of Materials (T-1) Group at Los Alamos National Laboratory, Los Alamos, NM, USA. E-mail: amarda@gmu.edu, gchennupati@lanl.gov*

Since its inception in CASP7, EMA methods are steadily improving in accuracy estimation and best model selection. EMA methods in CASP13 showed clear progress. Several outperformed the best ones from previous CASPs; the average GDT-TS losses for the best consensus method and the best single model methods were 0.052 and 0.075, respectively [11]. One of the driving forces of this progress is the recent application of deep learning [6]. However, no EMA method is able to always select the best model. The top performing groups in the recent CASP still failed to precisely estimate the accuracy of models and select the top model for some hard targets [6]. The potential for more improvement is significant for hard targets. Additionally, the decoys generated by recent application of deep learning methods also pose a challenge for EMA methods [11].

A recent line of work leverages the concept of energy landscape for decoy selection via basin selection, using either unsupervised [12] or supervised learning [13]. Energy landscape-based methods are shown to outperform clustering-based methods that do not take into account decoy energies. However, they incur significant time costs associated with reconstructing the landscape and identifying the best basins/clusters which motivated us to investigate an alternative method for decoy clustering [14]. Inspired by the success of non-negative matrix factorization (NMF) [15] in various applications of computational biology, such as, clustering genes for molecular pattern discovery [16] as well as discovering mutational signatures (as latent variables) in human cancer [17], we embarked on using NMF to cluster decoys for decoy selection.

While various clustering-based methods have been employed to cluster decoys for decoy selection, the inherent property of NMF to extract latent structures and hidden patterns to aid decoy selection is yet to be explored. NMF is an unsupervised, parts-based learning methodology that has a number of applications, including pattern recognition, dimensionality reduction, feature extraction, text mining, sparse coding, multimedia data analysis, speech recognition, information retrieval, social network analysis, etc. Detailed reviews can be found in Ref. [18], [19].

In this paper, we propose a novel decoy selection framework using NMF. The key insight leveraged here is that, often, decoy selection methods use many energy-based, consistency-based, and contact-based decoy characteristics/features. These features are used to design machine learning approaches that identify the best decoy in an ensemble. The occurrence of interactions or overlaps between the features in intra- or inter-categories (energy-based, consistency-based, or contact-based) is a concern. A preliminary investigation in [20] demonstrated the promise of an NMF-based approach. Specifically, we showed that the latent features, ingrained in the feature matrix describes an ensemble of decoys, an effective tool to cluster decoys and discover near-natives. We showed that NMF extracts the latent features underlying the feature matrix of a decoy ensemble for decoy selection as a way to identifying the best group of near-native decoys as well as the best decoy.

Building on these initial findings, in this paper we propose a new strategy that exploits decoy-specific characteristics to select the best group of decoy, and the best decoy from the selected group. We also perform a statistical significance test to find the best decoy selection method. Detailed analysis of results are also carried out with the help of illustrations to provide more insights into the inner mechanisms of NMF for clustering and for subsequent results. Our analysis exhibits that NMF-based methods outperform energy landscape-based methods and a state-of-the-art EMA method MUFOLD-CL [21]. Encouraging results with regard to the quality of selected decoys indicate overall utility and promise of NMF in furthering decoy selection research.

2 Related Work

Protein model accuracy estimation, also known as decoy selection, has been a part of protein structure prediction since its infancy [22]. The energy function that the structure prediction methods optimize performs the initial estimation of model accuracy. The insufficient capability of purely energy-based functions in determining the "nativeness" of a decoy led to limited success in accurately predicting decoy quality [23]–[25]. Due to unsatisfactory decoy selection performance, multitudes of decoy selection methods emerged based on different types of features and learning strategies. These methods can be divided into three categories: single-model methods, multi-model methods, and quasi-single methods. Many of these methods utilize various unsupervised and supervised machine learning techniques. All of these methods explore features based on characteristics of primary amino acid sequence, secondary structures, and energy-evaluated three-dimensional structures (decoys).

Single-model decoy selection methods estimate quality of one decoy at a time [26]. These methods either develop a statistical scoring function or employ a machine learning technique for selecting the best decoy(s) from a decoy set. Both the scoring function and the machine learning model rely on a diverse collection of either physics-based [27]–[29] and/or knowledge-based [30]–[32] features with the later being more successful in discriminating good decoys (near-natives) from bad (non-natives) ones [33], [34].

Clustering-based methods are at the heart of multi-model methods [35]. These methods rely on the consensus in a decoy set. The general strategy is to group the decoys and select the best k decoy-groups as prediction. Multi-model methods dominated the accuracy estimation category of biennial assessment of CASP until recently [36]. Single-model methods progressed to a point where these methods are on par or more successful in selecting the best decoy from a decoy set [37]. Diverse and ever-increasing number of features and continually improving supervised machine-learning techniques are the key ingredients in the success of single-model methods. Quasi-single methods combine the approaches of single-model and multi-model methods [38], [39]. The general strategy is to select some high-quality decoys as a reference. The rest in the decoy set are then compared with the reference decoys for quality estimation [40]. Research shows that quasi-single methods can improve upon single and multi-model methods [41]–[43].

Machine learning techniques proved to aid both multi-model and single-model methods. Decoy selection literature shows the use of a variety of ML techniques such as Random Forest [44], Support Vector Machines [45], [46], Neural Networks [47], [48], and ensemble methods [49].

Recently, the works in [50], [51] show the application of supervised and unsupervised machine learning techniques for decoy selection exploiting the energy landscapes of the protein ensembles. Great success of deep learning models in a variety of research areas such image recognition inspired research in decoy selection to employ these models for decoy selection as well. For instance, deep convolutional neural network has been quite successful in a number of single-model methods [48], [52]–[55]. Despite the enormous success, deep learning-based methods need to fulfill some specific requirements such as access to a large amount of data for accurate prediction. Machine learning-based methods, specially the supervised learning models, need to deal with several challenges such as imbalanced data distribution and lack of enough labeled data.

NMF has seen success in computational biology [56], [57]. For instance, Greene et al. [58] used NMF for clustering protein-protein interactions and Brunet et al. [16] used NMF to elucidate cancer subtypes. NMF has been used in a variety of computational biology applications: identifying distinct genomic subtypes [59], discovering functionally related genes [60], for protein sequence motif discovery [61], as well as for successfully decomposing the largest available dataset of human cancers and identifying cancer mutational signatures [17]. To the best of our knowledge, we are the first to employ NMF for decoy selection. Our preliminary results of NMF for decoy selection [20] showed promising results on two sets of proteins.

3 METHODS

In this paper we address the *decoy selection* problem as follows. Given a decoy ensemble, we group the decoys, select a decoy group/cluster, then select a representative decoy from the selected decoy-group. Our aim is to find the best decoy group and the best decoy. The best group is populated with the maximum number of good decoys (true positives, near-natives) compared to the size of the group. A near-native is structurally similar to the known native structure. Section 3.6 explains how near-natives are defined. The best decoy is close to the native. We employ NMF to group the decoys into different clusters. NMF provides a parts-based representation of the original features. Due to the non-negativity property, the parts produced by NMF can be interpreted as a subset of of the elements that tend to occur together in a sub-part of the dataset [62]. This phenomenon potentially makes NMF a good candidate to build a clustering-based decoy selection method. We leverage this interpretation of NMF to devise a NMF-based consensus method for grouping decoys and find the decoy-group that represents the near-native.

Fig. 1 shows our framework for NMF-based decoy selection. The decoys are colored in red, green, and blue which indicate the energy levels of the decoys. Red decoy are of higher energy. In contrast, blue decoys are of lower energy. Green indicates a level somewhere in between red and blue. First, we extract features from the decoys and store them in an initial feature matrix, X. In the next step, NMF decomposes the feature matrix into two non-negative matrices, W and H. The first factor matrix W contains the basis patterns. Linear combinations of these basis patterns describe and reconstruct each decoy in the initial matrix. These basis patterns define different decoy-groups or clusters. We assign a decoy d to a decoy-group G if d is closest to the basis pattern representing decoy-group G. Next, we select a decoy-group by applying our proposed decoy selection methods and evaluate its "near-nativeness". To assess the "near-nativeness" of a group, we determine how many near-native decoys populate the group. Our aim is to select a group which is populated by mostly true-positives (near-natives) and very few to zero false-positives, thereby resulting in a high precision/purity. We explain the evaluation metrics in Section 3.7. Finally, we select a representative decoy from the selected cluster/group as an approximation of the best decoy in the decoy set. Our aim is to find the best decoy from the selected group.

3.1 Feature Extraction

First, we extract 39 features from the decoys of each target protein. Of these 39 features, 9 are potential energy-based, 17 are Rosetta REF2015 energy terms, 9 features are based on the consistency between the actual and predicted values of decoys, and 3 are contact-based scores. One more feature comes from Rosetta Score12 total energy. The potential energy-based and consistency-based terms are used in a Support-Vector-Machine-based single-model decoy selection method [63].

3.1.1 Features Based on Energy Functions

Twenty seven features fall under this category. Eighteen of these features are collected from Rosetta REF2015 and Score12 energy functions. We use raw values of 17 terms from Rosetta REF2015 energy function. We also use REF2015 and Score12 total energy. The total energy values are computed using the weighted energy terms.

The 9 potential energy terms are following. We use a side-chain orientation-dependent potential $RWplus$ [64]; a distance-dependent potential DFIRE [65]; dDFIRE [66] adds orientation-dependency to DFIRE; three features from GOAP [67] which is a distance and orientation-dependent all-atom potential. GOAP contains DFIRE and an angle-dependent term. We use the overall GOAP potential, the DFIRE term, and the angle-dependent term as three features. OPUS-PSP [68] is an orientation-dependent all-atom potential that includes an orientation-dependent packing energy term and a Lenard-Jones repulsive energy term. Both of these energy terms and the total energy make up three more features.

3.1.2 Features Based on Structural Consistency

We use nine features based on structural consistency. Four of these are secondary structure-based features. We use PSIPRED to compute the secondary structure of each target from their primary sequence. We extract the secondary structure of the decoys for each protein using DSSP. We keep track of the number of matches in secondary structure elements (beta sheets, alpha helixes, and coils) between PSIPRED and DSSP calculations. We normalize the match counts for the three secondary structure elements by the length of the corresponding sequence, and use the normalize match counts as features. The fourth feature is computed

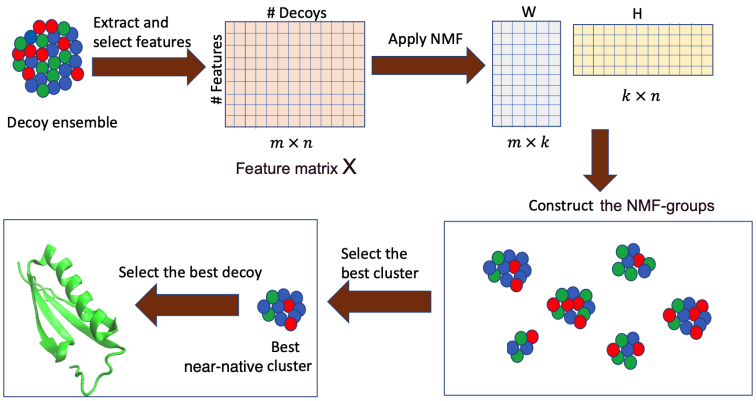

Fig. 1. Illustration: Framework of the proposed NMF-based decoy selection method.

as follows. When there is a match between the PSIPRED and DSSP calculations, we store the score of each secondary element computed by PSIPRED. We add these scores and use the total score as a feature.

We compute 5 features based on solvent accessibility. To determine the buried (B) or exposed (E) state of a residue (with respect to solvent accessibility), we use RaptorX, which is a Deep Convolutional Neural Fields (DeepCNF)-based web server [69]. We specify each residue as either buried or exposed based on the probabilities computed by RaptorX. We also calculate the relative solvent accessibility of a residue of each decoy using DSSP. To specify a residue as buried or exposed we divide the calculated solvent accessibility by the total solvent accessibility [70]. A cut-off value of 25% has been used in this process. Two more features are computed from the number of B and E matches computed by RaptorX and DSSP. We use the length of the corresponding sequence to normalize these features. When there is a mismatch between RaptorX and DSSP results, we note the corresponding probabilities. We use the combined probability as another feature. We also compute Pearson's correlation and cosine similarity between the number of secondary structure elements and solvent accessibility states computed from the primary sequence and from the decoys, and use them as two features.

3.1.3 Features Based on Residue Contacts

We use 3 features based on contact scores. We use relative contact order which is defined as the average sequence distance between all pairs of contacting residues and normalized by the total sequence length [71]. We extract two more features by following the process mentioned in [72]. We use RaptorX-contact to predict the contacts from amino-acid sequence. We treat the top 10 RaptorX-predicted contacts as references. We determine true positives (TP), false positives (FP), and False negatives (FN) from the decoys using the top 10 pairs of amino acids. If these 10 pairs are also found in contact in a decoy, we have a true positive. False negatives increase when the top 10 pairs are not found in contact in a decoy. Finally, false negatives are found if the contacts in a decoy are not found in the reference decoys. We calculate precision and recall and use them as features; precision is defined by TP/(TP+FP), and recall is defined as TP/(TP+FN).

Fig. 2 shows the importance of the employed features. Briefly, we calculate the importance of each feature using the feature_importances_ property of Random Forest algorithm implemented in scikit-learn as *RandomForestRegressor* [73]. As described in greater detail below in Section 3.6, we employ two datasets for evaluation, to which we refer as the *benchmark* dataset and the *CASP* dataset. Fig. 2**a** shows that features generated by the Rosetta template-free protocol play a crucial role in decoy-group selection for the benchmark proteins listed in Table 1. However, Fig. 2**b** shows that this is not the case for the CASP dataset listed in Table 2. In this case, the three most important features are the number of matches in the alpha helixes, the backbone-backbone hbonds distant in primary sequence, and the probability of an amino acid at ϕ/ψ.

3.2 Non-negative Matrix Factorization

Non negative matrix factorisation is a widely used unsupervised learning method for dimensionality reduction and feature extraction. The non-negative data, a matrix of dimensions features $\times$ samples, is factorized into two non-negative low-rank matrix factors, W and H with a small inner dimension K. For a given data $X \in \mathbb{R}_{+}^{F \times N}$ (features $\times$ samples), NMF approximates X with the product of W and H, by minimizing the Frobenius norm (indicated by $||.||_{\mathcal{F}}$),

$$\epsilon = min||X - WH||_{\mathcal{F}}^{2}$$

or, $X_{ij} = \sum_{s=1}^{K} W_{is}H_{sj} + \epsilon_{ij}$, where, ϵ_{ij} is the error of the approximation, which is normally distributed. In this way, each column of X (representing a sample) is expressed as a linear combination of the basis latent patterns (the columns of W) and its weights (the corresponding column of H), Fig. 3. The non-negativity forces NMF to learn local parts of the object (described in X) [74], hence, to extract easy interpretable and sparse latent features, which makes NMF a preferable technique when explainability is important.

NMF is underpinned by a statistical model of superimposed components (the number of these components is equal to the size of the small dimension K) that can be treated as latent features in Gaussian, Poisson, or other mixture model [75]. NMF minimization (with a specific distance metric $||...||_{dist}$) is equivalent to the expectation-minimization (EM) algorithm. In this probabilistic interpretation of NMF, the manifested variables are the columns

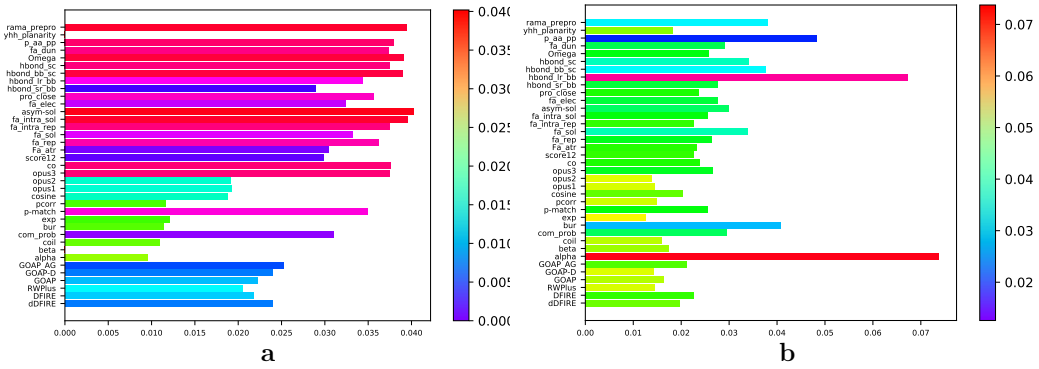

Fig. 2. X-axis shows feature importance values. Y-axis shows feature names. Higher value indicates higher importance. **a** shows feature importance on targets listed in Table 1 in Section 4, and **b** does so for targets listed in Table 2.

$d_1, ..., d_N$, of the matrix, X, generated by the latent variables, $h_1, ..., h_K$, that are the columns of the matrix, H. Specifically, each observable x_i is generated from a probability distribution with mean $\langle d_i \rangle = \sum_{s=1}^{K} W_{is} h_s$, where K is the number of the latent variables [74]. Thus, the influence of h_s on d_i is through the basis patterns represented by the columns of the matrix W, $w_1, ..., w_K$.

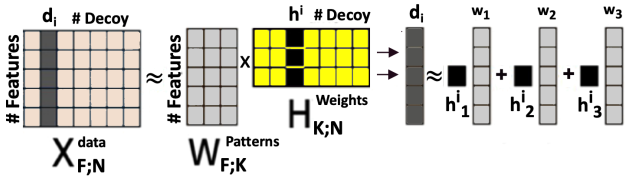

Fig. 3. Illustration of NMF decomposition of the feature matrix X. The decomposition produces two factor matrices W and H. The columns of W represent basis patterns. Each decoy d_i (column in feature matrix X) is expressed as a linear combination of the basis patterns with coefficients found in the corresponding column matrix H.

In our case, the basis patterns, represented by the columns of the matrix W, can be thought of as pseudo-decoys (not necessarily in the decoy ensemble) whose linear combinations span the entire ensemble space. Then each decoy is a linear combination of these pseudo-decoys with coefficients given by the corresponding columns of matrix H. The NMF optimization problem, $min||X - WH||_{dist}$, can be solved by various algorithms, such as the multiplicative update [74], block principle pivot [76] and projected gradient methods [77]. All these methods follow alternating non-negative least squares [78], where in each iteration one of the factors is fixed while updating the other factor.

3.3 NMF-Backed Decoy Groups

We construct decoy clusters/groups using the factors of NMF on the decoy matrix. The decoy matrix is essentially a feature matrix with features extracted from the decoys. In the decoy matrix X, the rows correspond to the features of decoys and the columns represent distinct decoys. Therefore, each cell $X(i, j)$ represents the i-th feature of j-th decoy. To satisfy the non-negativity, we shift the negative values of any feature in the feature matrix X into positive space, and then apply NMF to the feature matrix.

The basis patterns, i.e., the columns of the matrix W define the decoy-groups backed by NMF. To identify the basis pattern that a decoy d_i belongs to, we note the maximum value of the corresponding hidden variable h_s^i (see Fig. 3). A decoy d_i belongs to the group/cluster defined by the basis pattern w_i; basis pattern w_i is associated with the maximum value of the column h_s^i corresponding decoy d_i.

3.4 Decoy-Group Selection

Once the membership of each decoy to a given pattern w_i is established, we select a decoy group/cluster. The decoy-groups are characterized by their associated metrics. We compute two metrics. Each of these metrics are essential for our decoy (group) selection, and we use these metrics to name our methods. The first metric, median absolute deviation (MAD), measures the spread of data in a cluster. Lower MAD value indicates less variability. Another desirable property of MAD is its robustness against outliers. As our methods rely on the principle of consensus.

3.4.1 NMF-MAD

Median Absolute Deviation (MAD) measures the variability of data samples while not considering any application-centric characteristics of the data samples. This metrics is also resilient to outliers. Let C be a group and x_i be a decoy from C, then the MAD measure for C is defined as

$$\text{MAD}(C) = b \cdot median(\{dist(x_i, median(C)) \mid x_i \in C\}),$$

where b is a constant scale factor that depends on the probability density of the observed samples [79]. For each group C, we compute the MAD score for all decoys in the group and average the scores. The average MAD score is the group's characteristic. Groups are then ranked based on increasing MAD score and the top group is selected.

3.4.2 NMF-Rank

The second selection method is based on decoy-specific characteristics. The average of the energies of decoys of a group is defined as the *average energy* of the group. The *minimum energy* of a group is defined as the minimum of the energies of all decoys assigned to that group. The size of a group is the number of decoys populating the group. Three stages of rankings are performed on the groups to identify the group that represents the near-native. The groups are ranked based on increasing size and the top 5 groups are selected. We rank the selected groups again based on minimum energy and select the top 3 groups. Finally, these 3 groups are sorted in ascending order of average energy, where the top group is offered as prediction.

3.5 Decoy Selection

We use the concept of density score of a decoy [80] to select the best decoy from the top decoy-group. Let a decoy-group consists of n decoys and a decoy x_i belongs to this group. The density score S_i of decoy x_i is defined as following.

$$S_i = \frac{\sum_{j=1}^{n} r_{ij}}{n}$$

The term r_{ij} denotes the pairwise root-mean-squared-deviation (RMSD) between decoy x_i and decoy x_j ($1 \leq i, j \leq n$). We normalize the density scores so that the scores are in a range between -1 and 1. We compute the normalized density score $S_i^{'}$ as following.

$$S_i^{'} = \begin{cases} \frac{(S_i - S_{median})}{S_{median} - S_{min}} & \text{if } S_i < S_{median} \\ 0 & \text{if } S_i = S_{median} \\ \frac{(S_i - S_{median})}{S_{max} - S_{median}} & \text{if } S_i > S_{median} \end{cases}$$

The terms S_{min}, S_{max}, and S_{median} denote the minimum, maximum, and median density scores, respectively. We assign weight W_i to each decoy based on its normalized density score. The weight W_i is defined by $W_i = e^{-rS_i^{'}}$, where r is a constant. We use $r = 5$. We rank the decoys in decreasing order of their weights, and offer the top decoy as the best. If a decoy set is sparse decoy distribution, then a group might consist of only two decoys. In such a scenario, we offer the decoy with the lower energy as prediction.

3.6 Experimental Setup and Datasets

We evaluate our methods on two datasets. First, we evaluate on 17 benchmark proteins of different folds and lengths (number of amino acids, Table 1). We use Rosetta *template-free* protocol to generate $50,000$ to $60,000$ decoys per target. The 4-letter PDB id for each protein is in column 3. These proteins represent *easy*, *medium*, and *hard* cases for Rosetta. The difficulty levels ($easy$, $medium$, $hard$) are informed by the performance of an incremental clustering-based decoy selection. Details can be found in Ref. [12]. The size of decoy ensemble $|\Omega|$ for each protein is in column 6. Column 7 is the minimum distance, *min_dist*, between the decoys generated by Rosetta and a known native conformation in corresponding PDB entries. The *min_dist* informs about the varied performance that Rosetta achieves for each protein.

We set a threshold, $dist\_thresh$, to determine near-natives. We use least-root-mean-squared-deviation (lRMSD) to measure the distance between the decoys. All decoys with lRMSD from the known native structure within $dist\_thresh$ are considered near-natives. lRMSD removes differences due to rigid-body motions (translations and rotations in space) and reports the minimum RMSD by finding an optimal superimposition. The $dist\_thresh$ is set on a per-target basis as there are three different difficulty levels of proteins. For the easy cases ($min_dist < 0.7$Å), $dist\_thresh$ is set to 2Å. We set the range of $dist\_thresh$ to $\{2, 4.5\}$Å for medium-difficulty proteins. For hard proteins, we varied $dist\_thresh$ from 6Å to a value that ensures a non-zero number of near-natives in a cluster generated by an incremental clustering strategy [12]. Note, there are decoy sets where the best decoy is more than 4Å away from the native (e.g. hard targets). In such cases, we consider a decoy to be a near-native if it is within a certain distance threshold not too far away from the best decoy in the decoy set.

TABLE 1
Testing dataset (* denotes proteins with a predominant β fold and a short helix). The chain extracted from a multi-chain PDB entry is shown in parentheses.

Difficulty	#	PDB ID	Fold	Length	$\|\Omega\|$	min_dist (Å)
Easy	1	1ail	α	70	53,544	0.50
	2	1dtd(B)	$\alpha+\beta$	61	57,810	0.51
	3	1wap(A)	β	68	51,810	0.60
	4	1tig	$\alpha+\beta$	88	52,071	0.61
	5	1dtj(A)	$\alpha+\beta$	74	53,497	0.68
Medium	6	1hz6(A)	$\alpha+\beta$	64	57,449	0.72
	7	1c8c(A)	β*	64	53,297	1.08
	8	2ci2	$\alpha+\beta$	65	52,187	1.22
	9	1bq9	β	53	53,629	1.31
	10	1hhp	β*	99	52,128	1.52
	11	1fwp	$\alpha+\beta$	69	53,103	1.56
	12	1sap	β	66	51,182	1.75
Hard	13	2h5n(D)	α	123	51,450	2.05
	14	2ezk	α	93	50,167	2.56
	15	1aoy	α	78	52,189	3.27
	16	1cc5	α	83	51,666	3.95
	17	1isu(A)	*coil*	62	60,329	5.53

Besides, we consider 10 free modeling targets from CASP 12 and CASP 13 (Table 2). Several of these targets such as T0953s2, T0957s1, T1008 are determined as hard targets [6], [81].

TABLE 2
CASP dataset. CASP target IDs are shown in Column 2. PDB ID, Length, and Min RMSD over decoy dataset to corresponding native structure are shown for each target. Native structures only available in the CASP website [82] are marked by asterisks.

#	Target ID	PDB ID	Length	$\|\Omega\|$	min_dist (Å)
1	T1008-D1	6msp	77	55,000	1.54
2	T0886-D1	5fhy	69	55,000	4.92
3	T0953s1-D1	6f45	67	55,000	5.81
4	T0960-D2	6cl5	84	55,000	5.98
5	T0898-D2	**	55	43,435	6.0
6	T0892-D2	5nv4	110	36,860	6.62
7	T0953s2-D3	6f45	77	55,000	7.52
8	T0957s1-D1	6cp8	108	45,000	4.91
9	T0897-D1	**	138	25,000	8.30
10	T0859-D1	5jzr	113	40,000	9.06

We use the NMF ifrom scikit-learn with Non-negative Double Singular Value Decomposition (NNDSVD) initialization. We use 200 iterations with the default settings for other parameters such as Coordinate Descent solver and Frobenius norm. We vary k (number of components) from 5 to 38, then select the best-performing k. We use a publicly available code provided in [83] to calculate TM-Score and GDT-TS score. lRMSD loss of random selection is computed by averaging over results obtained from 100 runs. Both NMF-MAD and NMF-Rank take ~ 3 minutes to finish.

3.7 Evaluation Metrics

The evaluation of near-native group/cluster selection focuses on purity (p) metric, which keeps track of the number of near-native decoys relative to the size of a group. The purity of group/cluster C is

$$p_c = \frac{\text{number of near-natives in C}}{|C|}$$

where $|C|$ denotes the size of group C. Purity resembles the precision metric in machine learning. If we consider the near-natives in a group/cluster as true-positives (TP) and the non-native decoys as false-positives (FP), then purity is $\frac{TP}{TP+FP}$. Here we are less concerned about the false negatives as our objective is to maximize the possibility of selecting a near-native decoy from a random draw from a group, which entails maximizing true positives and minimizing false positives in a group. The metric p penalizes a group to the extent of the number of false positives present in that group. Therefore, a group/cluster populated with a large number of false positives will result in a low purity (p) regardless of number of true positive population present in that group. We also report the average lRMSD of all decoys in a group as an indication of group/cluster quality.

We use lRMSD loss to evaluate the performance of the best decoy selection. lRMSD loss is defined as the difference in lRMSD between the selected decoy and the best decoy in a decoy set. The best decoy is the one that is close to the native. We report GDT-TS loss and loss in TM-Score incurred by the selected decoy. GDT-TS is used in CASP to assess EMA methods [6]. TM-score is also popular among CASP participants. It eliminates protein size dependency in score calculation [84]. GDT-TS loss and TM-Score loss are defined similarly (the difference in score between the selected decoy and the best decoy in the decoy set).

4 Results

We present two sets of results. Decoy-group selection results are presented in the next section, followed by the results for the best decoy selection in terms of lRMSD loss, GTD-TS loss, and loss in TM-Score.

4.1 Decoy-Group Selection Results

First, we compare the NMF-based decoy selection methods, NMF-MAD and NMF-Rank, with four unsupervised basin-based methods presented in [12]. The concept of energy landscape has been used in [12] to construct decoy-groups. First, basins are extracted from the underlying energy landscape of a protein structure space, and then decoy selection is performed by ranking and selecting the basins based on their size (Basins-Select(S)), and size and energy (Basins-Select(S+E)). A Basin consists of decoys and is considered a decoy-group/cluster. Specifically, Basins-Select(S) ranks the basins based on decreasing basin-size and selects the top basin. Basins-Select(S+E) ranks the basins first by decreasing size and select top m basins where m is user-defined. Then, the m basins are further ranked based on increasing energy and the top basin is selected. Since the goals of obtaining lower energy and larger size pose two conflicting objectives, two Pareto-based selection methods are devised. Based on the concept of dominance [85], two measures Pareto rank (PR) and Pareto count (PC) are calculated. PR of a basin B denotes the number of other basins that dominate B. PC of a basin B denotes the number of other basins that B dominates. Basins are ranked based on increasing Pareto rank (Basins-Select(PR)) and the top basin is selected. Basins-Select(PR+PC) first ranks the basins based on increasing PR and then by decreasing PC, and selects the top basin. These four methods are shown to outperform a cluster-based decoy selection method in terms of the purity metric [12].

Fig. 4 compares NMF-MAD and NMF-Rank with four basin-based unsupervised decoy selection methods on 17 proteins (5 easy, 7 medium-difficulty, 5 hard). All methods perform comparably well on the easy test cases. For all the 5 test cases, NMF-MAD achieves 100% purity. NMF-Rank shows more than 90% purity in 4 test cases, and more than 80% purity for the remaining. The four basin-based methods achieve good purity scores (from 88% to 100%). However, one method, Basins-Select(S+E), shows poor performance (2.8% purity) even on an easy test case (*1dtd(B)*).

For medium-difficulty proteins, NMF-MAD performs better than basin-based methods in 4 out of 7 cases. The decoy sets for medium-difficulty proteins contain comparatively lower number of near-natives. We present two examples. For a decoy set of size $53,629$ (*1bq9*), only 1.6% of the decoys are near-natives. Similarly, we have only 2.5% near-natives among the decoys in a decoy set of size $52,128$ (*1hhp*). If we consider the near-natives as true positives, our datasets are imbalanced, a challenging problem in data mining and machine learning research. Our proposed decoy selection methods largely overcome this challenging problem and present us with reasonably good decoy selection results. For instance, the best that the basin-based methods achieve for the protein under PDB entry id *1bq9* is 80.4% purity, whereas NMF-MAD achieves 100% purity. As another example, NMF-Rank scores 74.1% for the protein under PDB entry id *1hhp*. For the same protein, the best purity score by any basin-based method is 53.6%. NMF-Rank and NMF-MAD both outperform the basin-based decoy (-group) selection methods.

The utility of NMF-based decoy selection methods can be better realized when we consider the hard test cases. The decoy sets for hard proteins exhibit the highest level of sparsity. The number of near-natives found in these decoy sets is the lowest (below 6%) among all three categories of difficulty level. Additionally, the best quality decoy that Rosetta could sample for these proteins is further away from the native compared to the best decoys under the category of easy and medium-difficulty proteins. For instance, for the hard protein with PDB entry id *1isu(A)*, the best decoy in the decoy set is 5.53Å from the native. In contrast, the best decoy in the decoy set for the easy protein with PDB entry id *1ail* is 0.50Å from the native. Template-free methods employs heuristics in their decoy generation process. Energy function designed for template-free methods contain inherent bias that may steer away the decoy generation process from an entire region of the decoy space that may contain near-native/native decoys [2], [86]–[88]. A lack of enough good quality decoys and the resulting sparsity in the decoy set for hard proteins make the task of decoy selection in template-free protein structure prediction more challenging. For these challenging cases, NMF-based methods significantly outperform basin-based methods in 4 out of 5 test cases. For instance, NMF-MAD achieves 100% purity for both the proteins with PDB entry ids *1cc5* and *1isu(A)*, whereas the basin-based methods can

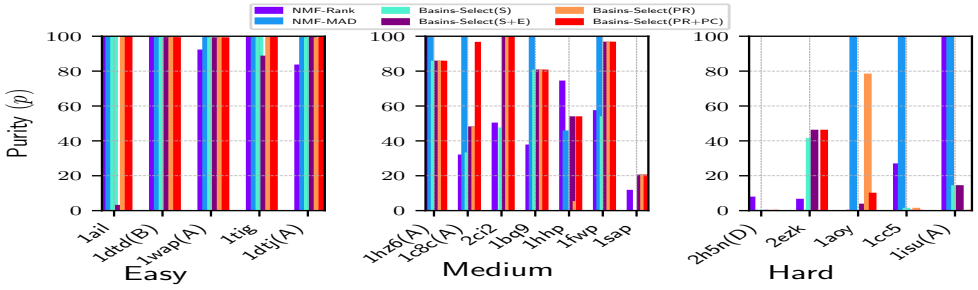

Fig. 4. Comparison of four unsupervised basin-based and two NMF-based decoy selection methods. y-axis tracks the purity of the top basin or group/cluster predicted by each selection method, while x-axis tracks the PDB entry id of each target protein.

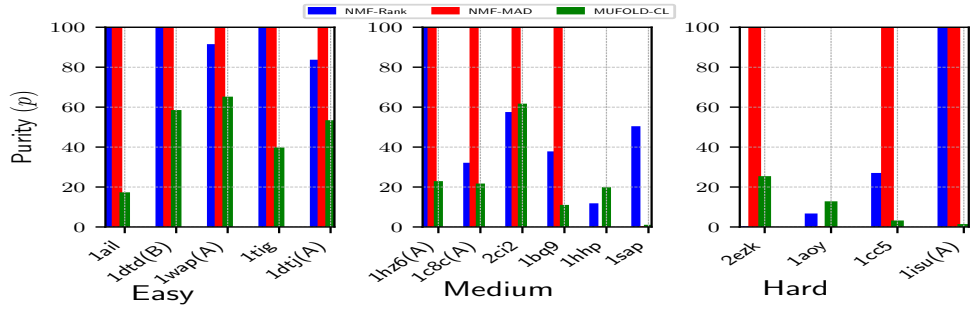

Fig. 5. Comparison of two NMF-based decoy selection methods and MUFOLD-CL. y-axis tracks the purity of the top basin/group/cluster predicted by each selection method, while x-axis tracks the PDB entry ID of each target protein.

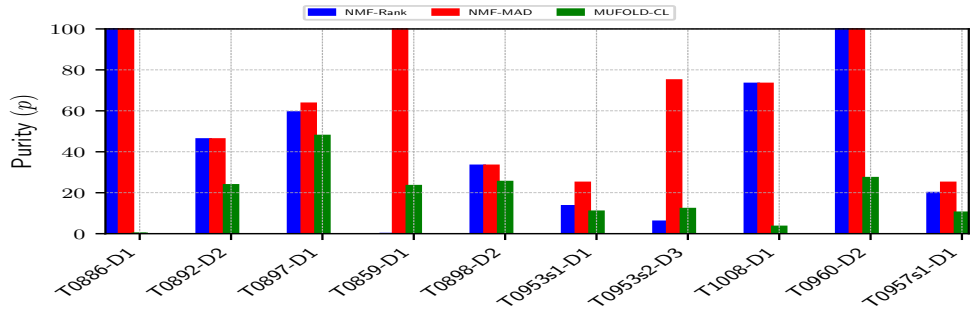

Fig. 6. Comparison of two NMF-based decoy selection methods and MUFOLD-CL on CASP targets. y-axis tracks the purity of the top basin/group/cluster predicted by each selection method, while x-axis tracks the ID of each target protein.

achieve 1.14% and 14.1% purity at best. On the hardest test case, the protein with PDB entry id *2h5n(D)*, NMF-Rank achieves 7.54% purity whereas the basin-based methods are unable to capture a single near-native (0% purity). Such an outstanding performance by NMF-based methods on the hard test cases emphasizes the utility of NMF for clustering/grouping decoys for decoy selection.

Fig. 5 compares NMF-MAD and NMF-Rank to a state-of-the-art EMA method MUFOLD-CL on the dataset listed in Table 1. MUFOLD-CL is a multi-model (clustering-based) method that clusters decoys and then selects cluster representatives [21]. Note that *1fwp* and *2h5n(D)* is absent in Fig. 5 because MUFOLD-CL was unable to successfully finish execution for these two targets. MUFOLD-CL is outperformed by NMF-MAD in $12/15$ and by NMF-Rank in $11/15$ test cases. Additionally, NMF-MAD acheives significantly better purity than MUFOLD-CL on 7 targets which implies that the possibility of selecting a good decoy from NMF-MAD-selected group is higher than that of MUFOLD-CL, in case one is interested in more than one decoys rather than only the best model/decoy.

Fig. 6 compares NMF-MAD, NMF-Rank and MUFOLD-CL on the CASP targets. NMF-MAD and NMF-Rank outperform MUFOLD-CL in $10/10$ and $8/10$ test cases.

4.2 Decoy Selection Results

We compare NMF-MAD and MUFOLD-CL for best model/decoy selection in Table 3 in terms of lRMSD loss, GDT-TS loss, and TM-Score loss. We report the loss incurred with random selection and the average lRMSD of the selected group as baselines. Table 3 shows that the lRMSD loss is below 1Å for 7 out of 17 test cases and below 4Å for 15 out of 17 test cases. One test case, a hard protein (*2h5n(D)*), NMF-MAD performs worse than random selection in terms of lRMSD loss. For the rest of the test cases, NMF-MAD-selected decoy is close to the best decoy available in the decoy set compared to the random selection. For instance, a randomly selected decoy for protein with PDB entry id *1ail* resulted in 6.2Å lRMSD loss, while NMF-MAD-selected decoy is only 0.65Å from the best decoy in the decoy set. For protein *1fwp* under the medium-difficulty category, lRMSD loss due to random selection is 6.0Å, whereas NMF-MAD-selected decoy is 0.72Å from the best decoy in the decoy set.

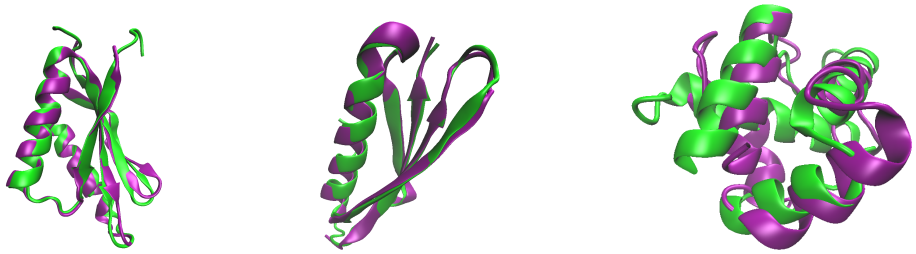

Fig. 7. Decoys under each difficulty category (easy, medium, hard) selected by NMF-MAD are shown superimposed over known wet-laboratory structures under PDB entry id 1tig, 1hz6(A), and 1cc5. The known native structure is colored in purple color, and the best decoy selected by NMF-MAD is colored green. RMSD loss due to NMF-MAD for each selected decoy is reported in parentheses.

For the hard protein *1cc5*, the decoy selected by NMF-MAD is 1.93Å from the best decoy in the decoy set, whereas the lRMSD loss due to random selection is 6.9Å. These results emphasize the excellence of NMF-MAD.

Column 6 of Table 3 shows the average lRMSD of the top clusters/decoy-groups for all test cases. The average lRMSD is below 4Å for 9 out of 17 test cases. This result implies that NMF-MAD is capable of selecting good clusters/decoy-groups that comprise mostly good decoys. For the easy proteins, the top decoy-group's average lRMSD is less than or equal to 1.5Å. The top decoy-groups for the medium-difficulty proteins show less than 4Å of average lRMSD in 4 out of 7 cases. For the proteins under hard category, the average lRMSD of the top decoy-groups is below 5Å in 6 out of 7 cases. These results show that NMF can be a promising tool for grouping good quality decoys.

Table 3 shows that out of 15 test cases, NMF-MAD is better than MUFOLD-CL in 8 cases for lRMSD loss, and in 10 cases for GDT-TS loss and TM-score loss. NMF-MAD outperforms MUFOLD-CL in 3 out of 4 hard targets in terms of lRMSD loss and all of 4 hard targets in terms of GDT-TS and TM-score loss, which shows promise of NMF-based methods for tackling hard targets.

Table 4 shows a quantitative comparison between NMF-MAD and MUFOLD-CL on the CASP dataset in terms of lRMSD loss, GDT-TS loss, and TM-score loss. NMF-MAD outperforms MUFOLD-CL in $8/10$ cases in terms of lRMSD loss, in $9/10$ cases in terms of GDT-TS loss. In terms of TM-score loss, MUFOLD-CL performs better than NMF-MAD in $5/10$ cases, NMF-MAD outperforms MUFOLD-CL in $4/10$ cases, while both are perform similar in the remaining case.

Fig. 7 shows NMF-MAD-selected decoys for each difficulty level (easy, medium, hard) superimposed over the known structures resolved in wet laboratory and deposited to Protein Data Bank. The native is colored purple and the decoy selected by NMF-MAD is colored green. The best decoy selected (see Fig. 7) by NMF-MAD for the easy protein with PDB entry id *1tig* and for the protein with PDB entry id *1hz6(A)* under medium-difficulty category are structurally similar to the known native. For the hard protein (PDB ID, *1cc5*), the selected decoy, albeit not quite close to the native as are the easy and medium-difficulty cases, does not deviate significantly from the native.

TABLE 3
Columns 2 and 3 show loss in lRMSD, GDT-TS, TM-score for the best model/decoy selected by NMF-MAD and MUFOLD-CL, respectively. Results due to random selection are shown in column 4. Column 5 records the average lRMSD of decoys populating the group selected by NMF-MAD. The '-' shown in two rows indicate that MUFOLD-CL was unable to return a result for the corresponding targets.

Targets	NMF-MAD lRMSD Loss (Å), GDT-TS loss, TM Loss	MUFOLD-CL lRMSD Loss (Å), GDT-TS loss, TM Loss	Loss Random (Å)	Average lRMSD (Å)
1ail	0.65, 0.54, 0.52	1.52, 0.27, 0.33	6.2	1.1
1dtdb(B)	0.55, 0.06, 0.04	0.94, 0.10, 0.09	5.9	1.1
1wap(A)	0.60, 0.21, 0.17	0.26, 0.75, 0.77	9.7	1.2
1tig	0.32, 0.07, 0.07	1.9, 0.03, 0.03	5.2	1.5
1dtj(A)	0.46, 0.07, 0.06	0.43, 0.10, 0.14	5.7	1.2
1hz6(A)	0.35, 0.08, 0.07	0.67, 0.20, 0.18	4.2	1.0
1c8c(A)	1.40, 0.62, 0.64	1.30, 0.23, 0.25	4.7	2.5
2ci2	3.80, 0.31, 0.30	2.5, 0.13, 0.16	6.0	8.2
1bq9	1.20, 0.13, 0.14	2.5, 0.31, 0.33	7.0	2.6
1hhp	3.30, 0.11, 0.12	2.6, 0.63, 0.64	11.6	4.6
1fwp	0.72, 0.11, 0.09	−, −, −	6.0	2.3
1sap	2.50, 0.42, 0.45	1.8, 0.25, 0.24	3.9	4.2
2h5n(D)	12.1, 0.11, 0.13	−, −, −	10.8	14.1
2ezk	6.30, 0.03, 0.02	4.2, 0.05, 0.07	6.4	8.8
1aoy	3.68, 0.34, 0.38	4.7, 0.46, 0.49	6.1	6.94
1cc5	1.93, 0.07, 0.10	5.5, 0.14, 0.14	6.9	7.0
1isu(A)	3.30, 0.08, 0.09	6.9, 0.40, 0.41	5.5	8.86

TABLE 4
Quantitative comparison results on CASP targets. Columns 2 and 3 show loss in lRMSD, GDT-TS, TM-score for the best model/decoy selected by NMF-MAD and MUFOLD-CL, respectively.

Targets	NMF-MAD lRMSD Loss (Å), GDT-TS loss, TM Loss	MUFOLD-CL lRMSD Loss (Å), GDT-TS loss, TM Loss
T0886-D1	2.77, 0.029, 0.03	8.51, 0.03, 0.02
T0892-D2	5.5, 0.007, 0.03	6.32, 0.025, 0.032
T0897-D1	3.5, 0.0, 0.003	6.5, 0.014, 0.017
T0859-D1	4.44, 0.011, 0.032	8.62, 0.022, 0.014
T0898-D2	5.2, 0.027, 0.01	5.1, 0.027, 0.007
T0953s1-D1	5.4, 0.011, 0.017	8.34, 0.019, 0.017
T0953s2-D3	4.88, 0.029, 0.039	4.2, 0.032, 0.024
T1008-D1	0.42, 0.012, 0.02	6.2, 0.019, 0.009
T0960-D2	3.51, 0.011, 0.016	5.6, 0.012, 0.02
T0957s1-D1	4.68, 0.008, 0.027	6.64, 0.074, 0.091

4.3 Statistical Significance Results

We report the results of Friedman statistical tests with Hommel's post-hoc [89] analysis on group purity in Table 5. The pair-wise non-parametric tests (such as Student's t-test or Mann Whitney U test) are simple to test statistical significance of two contending methods. The test becomes more complicated when there are multiple methods contending over multiple test cases. Friedman's test is ideal in such a case. It is a non-parametric test that helps to reject the null hypothesis. A null hypothesis here states that there is insignificant difference between the contending methods. Once the null hypothesis is rejected, we conduct a post-hoc analysis. There are a number of approaches (Nemenyi, Bonferroni, Holm, Hommel, Hochberg) to perform a post-hoc analysis, of which, Hommel analysis is considered relatively complicated, yet more powerful [89]. Therefore, we chose to perform Friedman tests with Hommel's post-hoc analysis as a more robust and accepted statistical procedure to justify the performance of our methods. The statistical tests are performed on all the seven different selection methods at $\alpha = 0.05$. The first column shows the methods, while the second column presents the average rank calculated from the Friedman's test [90], which rejects the null hypothesis. Upon the rejection of the null hypothesis, Hommel's post-hoc analysis helps to determine the statistical significance of the new techniques compared to others. The third and the fourth columns show the p-value and Hommel's critical value respectively. The lowest average rank shows the best (NMF-MAD) method, and is marked with an asterisk (*). A method is said to be significantly different from the best one if the p-value of the corresponding method is less than that of the p-Hommel at $\alpha = 0.05$, is in boldface. As shown in Table 5, NMF-MAD is the best and significantly outperforms the state-of-the-art basin-based decoy selection methods.

TABLE 5
Statistical significance of seven methods determined through Friedman tests with Hommel's post-hoc analysis at α=0.05. The best method is marked with an asterisk (*), while boldface presents the significance of the respective method when compared with the best method.

Method	Average Rank	p value	p Hommel
MUFOLD-CL	**6.0**	**1.83E-6**	**0.008**
Basins-Select(S+E)	**4.143**	**0.001**	**0.01**
Basins-Select(PR)	**3.929**	**0.004**	**0.0125**
Basins-Select(S)	**3.786**	**0.007**	**0.0167**
Basins-Select(PR+PC)	**3.679**	**0.012**	**0.025**
NMF-Rank	**3.571**	**0.018**	**0.05**
NMF-MAD*	1.893	-	-

5 Conclusion

In this paper, we have proposed two novel decoy selection methods, NMF-Rank and NMF-MAD. Both methods are based on non-negative matrix factorization. We compare these methods against four unsupervised, clustering-based methods and a state-of-the-art EMA method MUFOLD-CL in terms of purity (higher true positives and lower false positives), lRMSD loss, GDT-TS loss, and TM-score loss. Results show that NMF-MAD performs the best in selecting a good decoy group as well as in selecting the best decoy in a decoy ensemble. Moreover, NMF-MAD shows its superiority in tackling the harder cases. The time cost incurred by NMF-MAD is insignificant, as well.

These promising results shown by NMF-based methods open a new venue for furthering decoy selection research. In the future, we would like to investigate techniques to automatically determine the best value of *'k'*, i.e., the number of components for matrix decomposition. We would also like to discriminate the decoy features into categories and employ non-negative tensor decomposition to extract latent features that might effectively describe a protein structure and provide a marginalized summarization of tertiary structure. Moreover, the feature matrices of NMF are large scale, necessitating the use of parallelized NMF techniques. Recently developed distributed NMF [91] is shown to deal with 1TB input data, thus will employ these big data analytics in our future analysis. We can also use the extracted features in a supervised learning technique similar to [51], which we plan to further extend with heuristics (similar to [92], [93]) in improving the decoy selection.

Acknowledgment

This work is supported in part by NSF Grant No. 1900061 and resources provided by the Los Alamos National Laboratory Institutional Computing Program, which is supported by the U.S. Department of Energy National Nuclear Security Administration under Contract No. DE-AC52-06NA25396 and LANL laboratory directed research and development (LDRD) grant 20190020DR. Computations were run on Darwin, a research computing heterogeneous cluster (URL: `https://darwin.lanl.gov`). This material is additionally based upon work by AS supported by (while serving at) the National Science Foundation. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. We thank members of the Center for Advancing Human-Machine Partnerships (CAHMP) at George Mason University and the Shehu lab for useful feedback on this work.

References

[1] D. D. Boehr and P. E. Wright, "How do proteins interact?" *science*, vol. 320, no. 5882, pp. 1429–1430, 2008.

[2] T. Maximova, R. Moffatt, B. Ma, R. Nussinov, and A. Shehu, "Principles and overview of sampling methods for modeling macromolecular structure and dynamics," *PLoS computational biology*, vol. 12, no. 4, p. e1004619, 2016.

[3] D. Xu and Y. Zhang, "Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field," *Proteins: Structure, Function, and Bioinformatics*, vol. 80, no. 7, pp. 1715–1735, 2012.

[4] C. B. Anfinsen, "Principles that govern the folding of protein chains," *Science*, vol. 181, no. 4096, pp. 223–230, 1973.

[5] A. Verma, A. Schug, K. Lee, and W. Wenzel, "Basin hopping simulations for all-atom protein folding," *The Journal of chemical physics*, vol. 124, no. 4, p. 044515, 2006.

[6] J. Cheng, M.-H. Choe, A. Elofsson, K.-S. Han, J. Hou, A. H. Maghrabi, L. J. McGuffin, D. Menéndez-Hurtado, K. Olechnovič, T. Schwede *et al.*, "Estimation of model accuracy in casp13," *Proteins: Structure, Function, and Bioinformatics*, vol. 87, no. 12, pp. 1361–1377, 2019.

[7] J. Moult, K. Fidelis, A. Kryshtafovych, T. Schwede, and A. Tramontano, "Critical assessment of methods of protein structure prediction (casp)—round x," *Proteins: Structure, Function, and Bioinformatics*, vol. 82, pp. 1–6, 2014.

[8] T. Lazaridis and M. Karplus, "Discrimination of the native from misfolded protein models with an energy function including implicit solvation 1," *Journal of molecular biology*, vol. 288, no. 3, pp. 477–487, 1999.

[9] B. J. McConkey, V. Sobolev, and M. Edelman, "Discrimination of native protein structures using atom–atom contact scoring," *Proc. of the Natio. Aca. of Sci.*, vol. 100, no. 6, pp. 3215–3220, 2003.

[10] R. Cao, B. Adhikari, D. Bhattacharya, M. Sun, J. Hou, and J. Cheng, "Qacon: single model quality assessment using protein structural and contact information with machine learning techniques," *Bioinformatics*, vol. 33, no. 4, pp. 586–588, 2017.

[11] J. Won, M. Baek, B. Monastyrskyy, A. Kryshtafovych, and C. Seok, "Assessment of protein model structure accuracy estimation in casp13: Challenges in the era of deep learning," *Proteins: Structure, Function, and Bioinformatics*, vol. 87, no. 12, pp. 1351–1360, 2019.

[12] N. Akhter and A. Shehu, "From extraction of local structures of protein energy landscapes to improved decoy selection in template-free protein structure prediction," *Molecules*, vol. 23, no. 1, p. 216, 2018.

[13] N. Akhter, G. Chennupati, H. Djidjev, and A. Shehu, "Improved decoy selection via machine learning and ranking," in *8th International Conference on Computational Advances in Bio and Medical Sciences (ICCABS)*. IEEE, 2018, pp. 1–1.

[14] N. Akhter, W. Qiao, and A. Shehu, "An energy landscape treatment of decoy selection in template-free protein structure prediction," *Computation*, vol. 6, no. 2, p. 39, 2018.

[15] P. Paatero and U. Tapper, "Positive matrix factorization: A nonnegative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, 1994.

[16] J.-P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov, "Metagenes and molecular pattern discovery using matrix factorization," *Proc. of the Nation. Aca. of Sci.*, vol. 101, no. 12, pp. 4164–4169, 2004.

[17] L. B. Alexandrov, S. Nik-Zainal, D. C. Wedge, S. A. Aparicio, S. Behjati, A. V. Biankin, G. R. Bignell, N. Bolli, A. Borg, A.-L. Børresen-Dale *et al.*, "Signatures of mutational processes in human cancer," *Nature*, vol. 500, no. 7463, pp. 415–421, 2013.

[18] Y.-X. Wang and Y.-J. Zhang, "Nonnegative matrix factorization: A comprehensive review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 6, pp. 1336–1353, 2012.

[19] A. Cichocki, R. Zdunek, A. H. Phan, and S.-i. Amari, *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.

[20] N. Akhter, R. Vangara, G. Chennupati, B. S. Alexandrov, H. Djidjev, and A. Shehu, "Non-negative matrix factorization for selection of near-native protein tertiary structures," in *International Conference on Bioinformatics and Biomedicine, BIBM*. IEEE, 2019, pp. 70–73.

[21] J. Zhang and D. Xu, "Fast algorithm for population-based protein structural model analysis," *Proteomics*, vol. 13, no. 2, pp. 221–229, 2013.

[22] A. Elofsson, K. Joo, C. Keasar, J. Lee, A. H. Maghrabi, B. Manavalan, L. J. McGuffin, D. Ménendez Hurtado, C. Mirabello, R. Pilstål *et al.*, "Methods for estimation of model accuracy in casp12," *Proteins: Structure, Function, and Bioinformatics*, vol. 86, pp. 361–373, 2018.

[23] Y. N. Vorobjev and J. Hermans, "Free energies of protein decoys provide insight into determinants of protein stability," *Protein Sci*, vol. 10, no. 12, pp. 2498–2506, 2001.

[24] A. Verma and W. Wenzel, "Protein structure prediction by all-atom free-energy refinement," *BMC Struct Biol*, vol. 7, p. 12, 2006.

[25] K. Molloy, S. Saleh, and A. Shehu, "Probabilistic search and energy guidance for biased decoy sampling in ab-initio protein structure prediction," *IEEE/ACM Trans Comput Biol and Bioinf*, vol. 10, no. 5, pp. 1162–1175, 2013.

[26] K. Uziela and B. Wallner, "Proq2: estimation of model accuracy implemented in rosetta," *Bioinformatics*, vol. 32, no. 9, pp. 1411–1413, 2016.

[27] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. a. Swaminathan, and M. Karplus, "Charmm: a program for macromolecular energy, minimization, and dynamics calculations," *J Comp Chem*, vol. 4, no. 2, pp. 187–217, 1983.

[28] W. D. Cornell, P. Cieplak, C. I. Bayly, I. R. Gould, K. M. Merz, D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, and P. A. Kollman, "A second generation force field for the simulation of proteins, nucleic acids, and organic molecules," *J Amer Chem Soc*, vol. 118, no. 9, pp. 2309–2309, 1996.

[29] T. Lazaridis and M. Karplus, "Discrimination of the native from misfolded protein models with an energy function including implicit solvation," *J Mol Biol*, vol. 288, no. 3, pp. 477–487, 1999.

[30] S. Miyazawa and R. L. Jernigan, "An empirical energy potential with a reference state for protein fold and sequence recognition," *Proteins: Struct, Funct, and Bioinf*, vol. 36, no. 3, pp. 357–369, 1999.

[31] B. J. McConkey, V. Sobolev, and M. Edelman, "Discrimination of native protein structures using atom–atom contact scoring," *Proc Natl Acad Sci USA*, vol. 100, no. 6, pp. 3215–3220, 2003.

[32] K. T. Simons, I. Ruczinski, C. Kooperberg, B. A. Fox, C. Bystroff, and D. Baker, "Improved recognition of native-like protein structures using a combination of sequence-dependent and sequence-independent features of proteins," *Proteins: Struct, Funct, and Bioinf*, vol. 34, no. 1, pp. 82–95, 1999.

[33] B. Park and M. Levitt, "Energy functions that discriminate X-ray and near-native folds from well-constructed decoys," *J Mol Biol*, vol. 258, no. 2, pp. 367–392, 1996.

[34] A. K. Felts, E. Gallicchio, A. Wallqvist, and R. M. Levy, "Distinguishing native conformations of proteins from decoys with an effective free energy estimator based on the opls all-atom force field and the surface generalized born solvent model," *Proteins: Struct, Funct, and Bioinf*, vol. 48, no. 2, pp. 404–422, 2002.

[35] K. L. Kabir, L. Hassan, Z. Rajabi, N. Akhter, and A. Shehu, "Graph-based community detection for decoy selection in template-free protein structure prediction," *Molecules*, vol. 24, no. 5, p. 854, 2019.

[36] A. Kryshtafovych, A. Barbato, K. Fidelis, B. Monastyrskyy, T. Schwede, and A. Tramontano, "Assessment of the assessment: evaluation of the model quality estimates in casp10," *Proteins: Struct, Funct, and Bioinf*, vol. 82, pp. 112–126, 2014.

[37] A. Kryshtafovych, B. Monastyrskyy, K. Fidelis, T. Schwede, and A. Tramontano, "Assessment of model accuracy estimations in casp12," *Proteins: Struct, Funct, and Bioinf*, vol. 86, pp. 345–360, 2018.

[38] Z. He, M. Alazmi, J. Zhang, and D. Xu, "Protein structural model selection by combining consensus and single scoring methods," *PLoS ONE*, vol. 8, no. 9, p. e74006, 2013.

[39] M. Pawlowski, L. Kozlowski, and A. Kloczkowski, "MQAPsingle: A quasi single-model approach for estimation of the quality of individual protein structure models," *Proteins: Struct, Funct, and Bioinf*, vol. 84, no. 8, pp. 1021–1028, 2016.

[40] X. Jing, K. Wang, R. Lu, and Q. Dong, "Sorting protein decoys by machine-learning-to-rank," *Sci Reports*, vol. 6, p. 31571, 2016.

[41] A. Kryshtafovych, A. Barbato, K. Fidelis, B. Monastyrskyy, T. Schwede, and A. Tramontano, "Assessment of the assessment: evaluation of the model quality estimates in casp10," *Proteins: Structure, Function, and Bioinformatics*, vol. 82, pp. 112–126, 2014.

[42] Z. He, M. Alazmi, J. Zhang, and D. Xu, "Protein structural model selection by combining consensus and single scoring methods," *PloS one*, vol. 8, no. 9, p. e74006, 2013.

[43] M. Pawlowski, L. Kozlowski, and A. Kloczkowski, "Mqapsingle: A quasi single-model approach for estimation of the quality of individual protein structure models," *Proteins: Structure, Function, and Bioinformatics*, vol. 84, no. 8, pp. 1021–1028, 2016.

[44] B. Manavalan, J. Lee, and J. Lee, "Random forest-based protein model quality assessment (RFMQA) using structural features and potential energy terms," *PloS one*, vol. 9, no. 9, p. e106542, 2014.

[45] S. Chatterjee, S. Ghosh, and S. Vishveshwara, "Network properties of decoys and CASP predicted models: a comparison with native protein structures," *Molecular BioSystems*, vol. 9, no. 7, pp. 1774–1788, 2013.

[46] B. Manavalan and J. Lee, "SVMQA: support–vector-machine-based protein single-model quality assessment," *Bioinformatics*, vol. 33, no. 16, pp. 2496–2503, 2017.

[47] S. P. Nguyen, Y. Shang, and D. Xu, "DL-PRO: A novel deep learning method for protein model quality assessment," in *Int Conf Neural Networks (IJCNN)*. IEEE, 2014, pp. 2071–2078.

[48] R. Cao, D. Bhattacharya, J. Hou, and J. Cheng, "Deepqa: improving the estimation of single protein model quality with deep belief networks," *BMC bioinformatics*, vol. 17, no. 1, p. 495, 2016.

[49] S. Mirzaei, T. Sidi, C. Keasar, and S. Crivelli, "Purely structural protein scoring functions using support vector machine and ensemble learning," *IEEE/ACM Trans Comp Biol & Bioinf*, 2016.

[50] N. Akhter, G. Chennupati, K. L. Kabir, H. Djidjev, and A. Shehu, "Unsupervised and supervised learning over the energy landscape for protein decoy selection," *Biomolecules*, vol. 9, no. 10, p. 607, 2019.

[51] N. Akhter, G. Chennupati, H. Djidjev, and A. Shehu, "Decoy selection for protein structure prediction via extreme gradient boosting and ranking," *BMC Bioinformatics*, vol. 21, no. 1, 2019.

[52] J. Hou, R. Cao, and J. Cheng, "Deep convolutional neural networks for predicting the quality of single protein structural models," *bioRxiv*, p. 590620, 2019.

[53] G. Pagès, B. Charmettant, and S. Grudinin, "Protein model quality assessment using 3d oriented convolutional neural networks," *Bioinformatics*, vol. 35, no. 18, pp. 3313–3319, 2019.

[54] R. Sato and T. Ishida, "Protein model accuracy estimation based on local structure quality assessment using 3d convolutional neural network," *PloS one*, vol. 14, no. 9, 2019.

[55] J. Hou, T. Wu, R. Cao, and J. Cheng, "Protein tertiary structure modeling driven by deep learning and contact distance prediction in casp13," *Proteins: Structure, Function, and Bioinformatics*, 2019.

[56] K. Devarajan, "Nonnegative matrix factorization: an analytical and interpretive tool in computational biology," *PLoS computational biology*, vol. 4, no. 7, p. e1000029, 2008.

[57] L. B. Alexandrov, J. Kim, N. J. Haradhvala, M. N. Huang, A. W. T. Ng, Y. Wu, A. Boot, K. R. Covington, D. A. Gordenin, E. N. Bergstrom *et al.*, "The repertoire of mutational signatures in human cancer," *Nature*, vol. 578, no. 7793, pp. 94–101, 2020.

[58] D. Greene, G. Cagney, N. Krogan, and P. Cunningham, "Ensemble non-negative matrix factorization methods for clustering protein–protein interactions," *Bioinformatics*, vol. 24, no. 15, pp. 1722–1728, 2008.

[59] D. R. Carrasco, G. Tonon, Y. Huang, Y. Zhang, R. Sinha, B. Feng, J. P. Stewart, F. Zhan, D. Khatry, M. Protopopova *et al.*, "High-resolution genomic profiles define distinct clinico-pathogenetic subgroups of multiple myeloma patients," *Cancer cell*, vol. 9, no. 4, pp. 313–325, 2006.

[60] G. Wang, A. V. Kossenkov, and M. F. Ochs, "Ls-nmf: a modified non-negative matrix factorization algorithm utilizing uncertainty estimates," *BMC bioinformatics*, vol. 7, no. 1, p. 175, 2006.

[61] W. Kim, B. Chen, J. Kim, Y. Pan, and H. Park, "Sparse nonnegative matrix factorization for protein sequence motif discovery," *Expert Systems with Applications*, vol. 38, no. 10, pp. 13 198–13 207, 2011.

[62] A. Pascual-Montano, P. Carmona-Saez, M. Chagoyen, F. Tirado, J. M. Carazo, and R. D. Pascual-Marqui, "bionmf: a versatile tool for non-negative matrix factorization in biology," *BMC bioinformatics*, vol. 7, no. 1, p. 366, 2006.

[63] B. Manavalan and J. Lee, "Svmqa: Support–vector-machine-based protein single-model quality assessment," *Bioinformatics*, vol. 33, no. 16, pp. 2496–2503, 2017.

[64] J. Zhang and Y. Zhang, "A novel side-chain orientation dependent potential derived from random-walk reference state for protein fold selection and structure prediction," *PloS one*, vol. 5, no. 10, p. e15386, 2010.

[65] H. Zhou and Y. Zhou, "Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction," *Protein science*, vol. 11, no. 11, pp. 2714–2726, 2002.

[66] Y. Yang and Y. Zhou, "Specific interactions for ab initio folding of protein terminal regions with secondary structures," *Proteins: Structure, Function, and Bioinformatics*, vol. 72, no. 2, pp. 793–803, 2008.

[67] H. Zhou and J. Skolnick, "Goap: a generalized orientation-dependent, all-atom statistical potential for protein structure prediction," *Biophysical journal*, vol. 101, no. 8, pp. 2043–2052, 2011.

[68] M. Lu, A. D. Dousis, and J. Ma, "Opus-psp: an orientation-dependent statistical all-atom potential derived from side-chain packing," *Journal of molecular biology*, vol. 376, no. 1, pp. 288–301, 2008.

[69] S. Wang, W. Li, S. Liu, and J. Xu, "Raptorx-property: a web server for protein structure property prediction," *Nucleic acids research*, vol. 44, no. W1, pp. W430–W435, 2016.

[70] S. Miller, J. Janin, A. M. Lesk, and C. Chothia, "Interior and surface of monomeric proteins," *Journal of molecular biology*, vol. 196, no. 3, pp. 641–656, 1987.

[71] K. W. Plaxco, K. T. Simons, and D. Baker, "Contact order, transition state placement and the refolding rates of single domain proteins," *Journal of molecular biology*, vol. 277, no. 4, pp. 985–994, 1998.

[72] A. B. Zaman, P. V. Parthasarathy, and A. Shehu, "Using sequence-predicted contacts to guide template-free protein structure prediction," in *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*. ACM, 2019, pp. 154–160.

[73] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[74] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, p. 788, 1999.

[75] C. Févotte and A. T. Cemgil, "Nonnegative matrix factorizations as probabilistic inference in composite models," in *2009 17th European Signal Processing Conference*. IEEE, 2009, pp. 1913–1917.

[76] J. Kim and H. Park, "Fast nonnegative matrix factorization: An active-set-like method and comparisons," *SIAM Journal on Scientific Computing*, vol. 33, no. 6, pp. 3261–3281, 2011.

[77] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural computation*, vol. 19, no. 10, pp. 2756–2779, 2007.

[78] J. Kim and H. Park, "Toward faster nonnegative matrix factorization: A new algorithm and comparisons," in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 353–362.

[79] B. Boashash, *Time-frequency signal analysis and processing: a comprehensive reference*. Academic Press, 2015.

[80] K. Wang, B. Fain, M. Levitt, and R. Samudrala, "Improved protein structure selection using decoy-dependent discriminatory functions," *BMC structural biology*, vol. 4, no. 1, p. 8, 2004.

[81] L. A. Abriata, G. E. Tamò, B. Monastyrskyy, A. Kryshtafovych, and M. Dal Peraro, "Assessment of hard target modeling in casp12 reveals an emerging role of alignment-based contact prediction methods," *Proteins: Structure, Function, and Bioinformatics*, vol. 86, pp. 97–112, 2018.

[82] *Protein Structure Prediction Center*, Last Accessed: August 31, 2020. [Online]. Available: https://predictioncenter.org/

[83] Tm-score: Quantitative assessment of similarity between protein structures. [Online]. Available: https://zhanglab.ccmb.med.umich.edu/TM-score/

[84] Y. Zhang and J. Skolnick, "Scoring function for automated assessment of protein structure template quality," *Proteins: Structure, Function, and Bioinformatics*, vol. 57, no. 4, pp. 702–710, 2004.

[85] A. Santiago, H. J. F. Huacuja, B. Dorronsoro, J. E. Pecero, C. G. Santillan, J. J. G. Barbosa, and J. C. S. Monterrubio, "A survey of decomposition methods for multi-objective optimization," in *Recent Advances on Hybrid Approaches for Designing Intelligent Systems*. Springer, 2014, pp. 453–465.

[86] R. F. Alford, A. Leaver-Fay, J. R. Jeliazkov, M. J. O'Meara, F. P. DiMaio, H. Park, M. V. Shapovalov, P. D. Renfrew, V. K. Mulligan, K. Kappel *et al.*, "The rosetta all-atom energy function for macromolecular modeling and design," *Journal of chemical theory and computation*, vol. 13, no. 6, pp. 3031–3048, 2017.

[87] K. Molloy, S. Saleh, and A. Shehu, "Probabilistic search and energy guidance for biased decoy sampling in ab initio protein structure prediction," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 10, no. 5, pp. 1162–1175, 2013.

[88] R. Das, "Four small puzzles that rosetta doesn't solve," *PLoS One*, vol. 6, no. 5, p. e20044, 2011.

[89] S. Garcia and F. Herrera, "An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons," *Journal of Machine Learning Research*, vol. 9, pp. 2677–2694, 2008.

[90] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, vol. 7, no. Jan, pp. 1–30, 2006.

[91] G. Chennupati, R. ara, E. Skau, H. Djidjev, and B. S. Alexandrov, "Distributed non-negative matrix factorization with determination of the number of latent features," *The Journal of Supercomputing*, vol. 76, no. 9, pp. 7458–7488, 2020.

[92] G. Chennupati, J. Fitzgerald, and C. Ryan, "On the efficiency of multi-core grammatical evolution (mcge) evolving multi-core parallel programs," in *6th World Congress on Nature and Biologically Inspired Computing (NaBIC)*. IEEE, 2014, pp. 238–243.

[93] G. Chennupati, R. M. A. Azad, and C. Ryan, "Performance optimization of multi-core grammatical evolution generated parallel recursive programs," in *Proceedings of the Annual Conference on Genetic and Evolutionary Computation*. ACM, 2015, pp. 1007–1014.

Nasrin Akhter Nasrin Akhter is a PhD candidate in Computer Science at George Mason University, USA. She obtained her Master degree from Department of Computer Science and Engineering at University of Dhaka, Bangladesh. Her research focuses on Machine Learning and Data mining with applications in Computational Biology and Bioinformatics.

Kazi Lutful Kabir is a Ph.D. student in the Department of Computer Science at George Mason University, Fairfax, VA, USA. In 2019, he obtained his MS degree in Computer Science from George Mason University. His research interests include computational biology, machine learning, and molecular dynamics. Currently, he is working on the applications of machine learning algorithms in the domain of structural bioinformatics.

Gopinath Chennupati is a Computer Scientist in the Information Sciences (CCS-3) group at Los Alamos National Laboratory (LANL). Gopinath has obtained his PhD from University of Limerick, Ireland. Gopinath works on high performance computing (HPC), performance modeling, natural language processing (NLP), deep/machine learning and high performance linear algebra, etc.

Raviteja Vangara is a post-masters Graduate research assistant at Theoretical division in Los Alamos National Laboratory and Doctoral student in Chemical and Biological Engineering at University of New Mexico. Raviteja works on developing unsupervised machine learning techniques which involve cluster analysis, matrix and tensor factorization techniques for pattern recognition and latent feature extraction.

Hristo Djidjev is a Computer Scientist in the Information Sciences (CCS-3) group at Los Alamos National Laboratory (LANL). Before joining LANL as a scientist, Hristo worked as an Assistant Professor in Rice University, Senior Lecturer in Warwick University. He is currently a Research Adjunct Professor at Carleton University, Ottawa, Canada. Hristo holds an MSc in applied mathematics and a PhD in computer science from Sofia University, Bulgaria.

Alexandrov Boian is a scientist at the Theoretical Division in Los Alamos National Laboratory. He has MS in Theoretical Physics, a PhD in Nuclear Engineering and second PhD in Computational Biophysics. Alexandrov is specialized in Big Data analytics, Nonnegative Matrix and Tensor Factorization, Unsupervised Learning,and Latent Feature Extraction.

Amarda Shehu Dr. Amarda Shehu is a Professor in the Department of Computer Science in the Volgenau School of Engineering. She is Co-Director of the Center for Advancing Human-Machine Partnerships (CAHMP), a Transdisciplinary Center for Advanced Study at George Mason University. Shehu obtained her Ph.D. from Rice University in 2008. Her research focuses on novel algorithms in artificial intelligence and machine learning to bridge between computer and information science, engineering, and the life sciences. Her laboratory has made many contributions in bioinformatics and computational biology regarding the relationship between macromolecular sequence, structure, dynamics, and function. Shehu has published over 120 technical papers with postdoctoral, graduate, undergraduate, and high-school students. She is currently the chair of the steering committee of the IEEE/ACM Journal on Transactions in Computational Biology and Bioinformatics, where she is also an associate editor. Shehu is the recipient of an NSF CAREER Award, and her research is regularly supported by various NSF programs, including Information Integration and Informatics, Robust Intelligence, Computing Core Foundations, and Software Infrastructure, as well as various state and private research awards.