SHREC'19 Protein Shape Retrieval Contest

Florent Langenfeld[1,*], Apostolos Axenopoulos[2], Halim Benhabiles[3], Petros Daras[2], Andrea Giachetti[4], Xusi Han[5], Karim Hammoudi[6], Daisuke Kihara[5,7], Tuan M. Lai[7], Mahmoud Melkemi[6], Stelios K. Mylonas[2], Genki Terashi[5], Yufan Wang[8], Feryal Windal[3], and Matthieu Montes[1,*,@]

[1]Laboratoire GBCM, EA7528, Conservatoire National des Arts-et-Métiers, 2 rue Conté, 75003 Paris, France
[2]Visual Computing Lab (VCL) team, Information Technologies Institute, Centre for Research and Technology Hellas, Greece
[3]ISEN-Lille, Yncréa Hauts de France
[4]Department of Computer Science, Università di Verona, Strada le Grazie 15, 37134 Verona
[5]Department of Biological Sciences, Purdue University, West Lafayette, Indiana, USA
[6]Université Haute-Alsace, IRIMAS EA 7499, F-68100 Mulhouse
[7]Department of Computer Science, Purdue University, West Lafayette, Indiana, USA
[8]Beijing Computational Science Research Center, Beijing, China
[*]Track organizers
[@]Corresponding author: Matthieu Montes, matthieu.montes@cnam.fr

March 9, 2019

Abstract

This track aimed at retrieving protein evolutionary classification based on their surfaces meshes only. Given that proteins are dynamic, non-rigid objects and that evolution tends to conserve patterns related to their activity and function, this track offers a challenging issue using biologically relevant molecules. We evaluated the performance of 5 different algorithms and analysed their ability, over a dataset of 5,298 objects, to retrieve various conformations of identical proteins and various conformations of ortholog proteins (proteins from different organisms and showing the same activity).

1 Introduction

Proteins are complex macro-molecular molecules constituted of hundreds to millions of atoms, and are usually classified according to their function in the cellular environment. They display various motions reflecting 1) the relative motion of their atoms and 2) their ability to undergo small to large conformational changes in order to perform their cellular activities through surficial binding notably with other proteins (Protein-Protein Interaction, PPIs).Thus, protein structures are widely studied and relies on experimentally resolved structures stored in databases such as the Protein Data Bank [1]. Structures are represented as a set of 3D coordinates indicating the position of the atoms, considered individually as rigid shperes, and current algorithms analyses this cloud of points to extract biologically relevant data such as sepcific interactions between atoms, concerted displacements, ...

Another usufull way to represent proteins is to compute their surfaces, typically representing their solvent-excluded surface (SES) as defined by Connolly [2]. The SES is an abstraction of the surface that may form contacts with other molecules of the environment. Detecting similarities and/or dissimilarities between protein surfaces (all surfaces from all proteins of a cell, for instance) is of main importance in drug discovery pipelines, adverse drug event prediction and in the characterization of molecular processes and diseases. However, it remains very challenging to detect and characterize such variations as the SES surface may fluctuate for a given protein displaying many conformations, or may be very similar among different species as proteins may have the same role.

In this SHREC19' track, we used the Structural Classification of Proteins — extended (SCOPe) database [3–5] to generate a dataset of 5,298 proteins extracted from 211 PDB entries. The SCOPe classification was used to assess the participants' methods 1) at the *species* level (the ability to retrieve a conformations from the protein

of a given species) and 2) at the *proteins* level (the ability to retrieve a conformations from ortholog proteins, *i.e.* regardless of the species).Methods runtimes are also indicated when provided by the participants.

2 Data Set

The (mostly) manually-curated SCOPe database [3–5] elaborates a hirarchical classification of protein domains using structural and evolutionary relationships: the 2 top levels are structure-based (Class and Fold) while the 4 bottom levels are evolutionary-based (Superfamily, Family, Protein and Species, from top to bottom). The lowest level (Domain) links the SCOPe database to the protein structures from the PDB [1]. It therefore represents a useful representation of the protein domains relationships and protein domains surfaces.

We only kept SCOPe entries 1) from NMR structures whose conformers display the same number of atoms, 2) from 3 Classes: All alpha proteins, Alpha and beta proteins (a+b) and Alpha and beta proteins (a/b), 3) with at least 4 ortholog proteins.Finally, we randomly selected domains to decrease the data set size to 5,298 structures, representing 241 SCOPe entries extracted from 211 PDB entries. The SES surfaces were computed using EDTSurf [6]. At the *species* level, the data set is composed of 54 classes and at the *proteins* level, it is composed of 17 classes (Table 1).

Compared to the SHREC18' protein shape retrieval track, this year's track is focused on the evolutionary relationships between proteins shapes. As a consequence of the selection process of surfaces to be included in the dataset, the selected proteins are similar in terms of size (from 32 to 161 amino-acids) with most of the structures being 66 to 111-amino-acids long. The corresponding meshes are also of similar size, ranging from ˜54,000 to ˜270,000 points. Furthermore, because of the evolutionary relationships between them, the shapes from the same class at the *proteins* level (reflecting the variety of shapes from ortholog proteins) are expected to have a high level of similiratity in their overall shape as the share the same activities and functions in organisms that co-evolved from the same ancestor. For these reasons, the discrimination between shapes at the *species* level is expected to be more difficult compared to the *proteins* level.

3 Evaluation

3.1 Discounted Cumulative Gain

The Discounted Cumulative Gain (DCG) is a weighted statistics assuming that correct results associated with a higher rank should imply a gain in the performance rating as users will are more likely to consider these results. For a list R of correct results, a list G is generated, where G_i is 1 if element R_i is in the correct class (the ground truth class associated with element i GT_i), or 0 otherwise. The discounted Cumulative Gain is then computed using the following:

$$DCG_i = \begin{cases} G_1, & \text{if } i = 1 \\ DCG_{i-1} + \frac{G_i}{log_2(i)}, & \text{otherwise} \end{cases}$$

This value is then divided by the maximal value possible (*i.e.* the value obtained by the ground truth):

$$DCG = \frac{DCG_k}{1 + \sum_{j=2}^{|C|} \frac{1}{log_2(j)}}$$

where k is the number of objects in the data set and C the size of the classes. This value is a good summary of the performance when comparing algorithms.

3.2 Nearest Neighbor, First-tier and Second-tier

These paramters check the ratio of models thatscalescale belong to the same class as the query. For Nearest Neighbor, the first match only is considered (excluding the identity), while the $|C| - 1$ and $2 * (|C| - 1)$ are considered for First-tier and Second-tier parameters.

3.3 Precision-Recall plot and E-measure

Precision P represents the ratio of models from class C retrieved within all objects attributed to class C, while Recall R represents the ratio of models from class C retrieved compared to $|C|$.

The $E - measure$ is a composite parameter of both Precision and Recall:

$$E - measure = 1 - \frac{2}{\frac{1}{P} + \frac{1}{R}}$$

Table 1: Summary of the classes at the *species* and *proteins* levels

Level	Number of classes	Average class size	Largest class size
Species	54	98.11	1049
Protein	17	311.65	1160

4 Participants & Methods

4.1 Combined CNN-LDS framework (ConvLDSNet) for 3D shape retrieval - Stelios Mylonas, Apostolos Axenopoulos, Petros Daras

4.1.1 Problem definition

Following the recent tendency to address many scientific tasks by exploiting the existing vast amount of data, we propose a data-driven approach for the problem of 3D protein shape retrieval. A 3D neural network (NN) has been trained on relevant datasets to learn appropriate features (descriptors) for the representation of 3D molecular shapes. The input 3D model is the Solvent Excluded Surface (SES) of a protein molecule, which has been created from the molecules tertiary structure (PDB format) using the EDTSurf software. This software produces a high resolution watertight triangulated mesh, which is then transformed to a $32 \times 32 \times 32$ voxel model to be used as input in our method.

4.1.2 Input representation

Based on the approach of Wu *et al.* [7], we rasterize the protein 3D model to a binary voxel grid. The 3D models of the proteins are watertight, thus the parity count method is applied for binary voxelization. A voxel v is classified by counting the number of times that a line crossing the center of the voxel intersects polygons of the 3D model surface. Ray-casting the 3D model with parallel rays, all of the voxels along the ray are classified. For an odd number of intersections, voxel v is considered interior to the model, while for an even number, outside. For a $N \times N \times N$ voxel grid resolution, where $N = 32$, we cast $N \times N = 1024$ rays, with each ray passing through N-voxel centers.

4.1.3 Proposed method

The proposed architecture is depicted in Fig. 1. This network is an extension of our last year's proposal [8], which was based on the VoxNet CNN [9]. The new scheme consists of two branches, where two different operations are applied to the input voxels. The first convolutional branch is identical to VoxNet CNN and consists of 2 volumetric convolutional layers, 1 max-pooling layer and 1 fully connected (FC) layer. The second branch consists of two 3D-LDS modules and a FC layer. The 3D-LDS module is a novel operation which aims to simulate the behavior of Linear Dynamical Systems (LDS) and incorporate it as a NN layer. This operation was first introduced in [10] and is extended here to the 3D domain. The features obtained by the two branches after their fully connected layers are, then, concatenated and fed to a last FC layer, which provides the output feature vector.

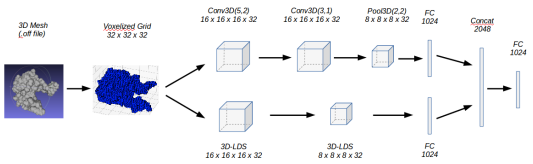

Figure 1: The proposed architecture consisting of a convolutional branch (top) and an LDS branch (bottom).

4.1.4 Training procedure

Two datasets have been used for training the proposed scheme; the dataset from last year's competition [8] and the MolMovDB [11] dataset. Among the three runs submitted, the first one (ConvLDSNet1) resulted from training the network on SHREC18, the second one (ConvLDSNet2) from training on MolMovDB and the third one (ConvLDSNet3) from training first on MolMovDB and then fine-tuning on SHREC18. Since both datasets contain classes of proteins, we added at the end of the network a Softmax layer and trained the network on a classification task.

In all cases, a three-stage training scheme has been employed: at the first stage, the convolutional part of the network is trained, by removing the LDS-branch, while, at the second step, the LDS branch is trained by removing the convolutional one. Finally, the two

branches are combined, freezing their weights and training separately the last part of the network. Subsequently, the Softmax layer is dropped and the architecture is used for feature extraction. For each previously unseen input, a feature vector is extracted. After the completion of the feature extraction, the Euclidean distance metric is used to measure the dissimilarity between two input models. Small distance values indicate that the corresponding feature vectors represent members of the same protein class.

The calculation of descriptors took on average 2.5 milliseconds per model on a GeForce GTX1070 GPU, while the average comparison time between two descriptors is 0.002 milliseconds on an Intel Core i7-6700K CPU.

Acknowledgement

The work was supported by the ATXN1-MED15 PPI project funded by the GSRT - Hellenic Foundation for Research and Innovation.

4.2 Protein Shape Retrieval using 3D Zernicke moments (3DZM) - Yufan Wang

3D Zernike moments are used as the shape descriptors. A nlm file is generated with Zernike moments' nlm coefficienst of the 5,298 proteins inside. Correlation Coefficient(cc) is used to evaluate the similarity of two shapes; the similarities are calculated between each pair, resulting in $5297 * 5298/2$ pairs. Fast Fourier Transformation (FFT) method is used to speed up the model rotation calculation to find the situation that maximises the overlap of the two proteins. The correlation coefficient cc is then computed as the similarity of the two shapes. The dissimilarity between each pair is defined as $(1 - cc)$.

Data preprocessing (transforming off files into pdb files) costs nearly 1.5 hours. Another 0.5 hour is required to get descriptors, namely, to generate the nlm file from the input 5,298 pdb files. To align and compute similarities between two shapes, about 0.5 second is required; therefore, the comparison calculations lasts 4 hours using 480 cpus.

4.3 3D Zernike Descriptors (3DZD) - Xusi Han, Tuan M. Lai, Genki Terashi, Daisuke Kihara

Our group has submitted three runs, all of which were based on 3D Zernike Descriptors (3DZD). We represented the protein global surface information with 3D Zernike Descriptors (3DZD) and quantified the similarity between 3DZDs by either the Euclidean distance or a similarity score from neural network. 3DZD is mathematical moment-based invariants of 3D functions [12], which has been demonstrated efficient for various biomolecular structure comparisons [13]. To calculate 3DZD for each protein, the surface triangulation of solvent excluded surface was mapped onto a 3D cubic grid, where each voxel (a cube defined by the grid) was assigned either 1 or 0: 1 for a surface voxel that locates closer than 1.7 grid interval to any triangle defining the protein surface, and 0 otherwise. This 3D grid with 1s and 0s was considered as a 3D function $f(x)$, from which 3DZD was computed. On average, 3DZD calculation takes 3.48 seconds per protein.

For the first submission (3DZD1), the global surface similarity between two proteins was quantified by the Euclidean distance of their 3DZDs. A small distance value indicates that two proteins share similar global surface. In this calculation, we took the triangulated surface (.off file) for each of the 5,298 proteins as the input to 3DZD computation and generated the 121-dimensional vector for each protein. The Euclidean distances between one query protein against all other 5297 proteins were calculated and put into each row in our first distance matrix.

For the second run (3DZD2), we built a deep learning based model to quantify the similarity between protein structures. The model was trained on all proteins in the SCOPe 2.07 database. We downloaded about 274,230 protein structures from the database for training. Solvent excluded surface of each protein was generated using the EDTSurf software [6]. The triangulated surface was then taken as the input to 3DZD computation, which produced 121-dimensional vector for each protein. On a high level, given a pair of protein structures, the deep learning model outputs a score between 0 and 1 indicating their similarity (the higher the score, the more similar the structures). The model consisted of an encoder whose role was to compute key features from a 3DZD vector. The encoder was a feed forward neural network consisting of three hidden layers. Each layer used ReLU as the activation function. Intuitively,

each hidden layer of the encoder computes a new level of representation of the original 3DZD vector. Given two protein structures as input, the model used the encoder to compute new features for each protein's 3DZD vector. The computed features and the original 3DZD vectors of the two structures were then compared using various operations such as the Euclidean distance, the Cosine similarity, the element-wise absolute difference, and the element-wise product. The comparision results as well as additional features such as the difference in number of vertices and the difference in number of faces were together fed into a final feed forward neural network that outputs a score between 0 and 1. We used techniques such as batch normalization and dropout to improve the training process. In the training data, if two protein structures had the same protein level, they were considered as being similar. When using a Titan X GPU, the neural network took about 0.1774 seconds on average to compare two proteins (given that 3DZD vectors have been pre-computed).

For the third run (3DZD3), we have used the same approach as in the second submission, except that in this case we trained the model to consider two protein structures to be similar only when the two structures have the same *species* level. In the second submission, if two protein structures have the same *proteins* level but different *species* level, we still considered them as being similar.

4.4 Histogram of Area Projection Transform (HAPT)- Andrea Gichetti

The method characterizes protein shapes with the Histograms of Area Projection Transform (HAPT) [14]. This descriptor, well suited for nonrigid shape retrieval and well behaving in SHREC 2018 contest on protein shape retrieval [8] is based on a spatial map (Multiscale Area Projection Transform) [14] that encodes the likelihood of the 3D points inside the shape of being centres of spherical symmetry. This map is obtained by computing, for each radius of interest, the value:

$$APT(\vec{x}, S, R, \sigma) = Area(T_R^{-1}(k_\sigma(\vec{x}) \subset T_R(S, \vec{n})))$$

where S is the surface of the object, $T_R(S, \vec{n})$, is the parallel surface of S shifted along the normal vector $\vec{n}$ (only in the inner direction) and $k_\sigma(\vec{x})$ is a sphere of radius σ centred in the generic 3D point $\vec{x}$ where the map is computed. Values at different radii are normalized in order to have a scale-invariant behaviour, creating the Multiscale APT (MAPT):

$$MAPT(x, y, z, R, S) = \alpha(R)APT(x, y, z, S, R, \sigma(R))$$

where $\alpha(R) = 1/4\pi R^2$ and $\sigma(R) = c \cdot R, \;\; 0 < c < 1$.

A discrete MAPT is easily computed, for selected values of R, on a voxelized grid including the surface mesh, with the procedure described in Giachetti & Lovato [14]. The map is computed in a grid of voxels with side s on a set of corresponding sampled radius values. For the proposed task, discrete MAPT maps were quantized in 12 bins and histograms computed at the selected scales (radii) were concatenated creating an unique descriptor. Voxel side and sampled radii were fixed set for each run and chosen to represent the approximate radii of the spherical symmetries visible in the models.

We tested three different options for the algorithm's parameters. In the first (HAPT1), we put $s = 0.5$ and we computed the MAPT histograms for 8 increasing radii starting from $R_1 = 0.5$ iteratively adding a fixed step of 0.5 for the remaining values and setting $\sigma = 0.25$. In the second (HAPT2), we put $s = 0.4$ and we computed the MAPT histograms for 10 increasing radii starting from $R_1 = 0.4$ iteratively adding a fixed step of 0.4 for the remaining values and setting $\sigma = 0.2$. In the third (HAPT3), we put $s = 0.3$ and we computed the MAPT histograms for 8 increasing radii starting from $R_1 = 0.3$ iteratively adding a fixed step of 0.3 for the remaining values and setting $\sigma = 0.15$.

The procedure for model comparison then simply consists in concatenating the MAPT histograms computed at the different scales and measuring distances between shapes by evaluating the Jeffrey divergence of the corresponding concatenated vectors. The estimation of the descriptors took 4.98 seconds on average for the first run, 11.1 seconds on average for the second run, 13.02 seconds on average for the third run on a laptop with an Intel® CoreTM i7-4720HQ CPU running Ubuntu Linux 18.04. The descriptor comparison time was negligible.

4.5 A Framework towards Protein Shape Singularity Characterization (Ft-PSSC) - Halim Benhabiles, Karim Hammoudi, Feryal Windal, Mahmoud Melkemi

4.5.1 Method description

Our proposed retrieval method aims at exploring three different feature extraction techniques in order to reach

Table 2: Running times in seconds of each step of the Ft-PSSC processing pipeline obtained respectively from the smallest and biggest protein of the dataset.

Point cloud size	Pre-processing	GASD	Local FPFH	FPFH feature selection	VLAD	Distance to all proteins of the dataset: Vector length 512	Vector length 4224	Vector length 512+512
53k	0.12	0.85	6.56	0.001	0.11	0.017	0.14	0.03
259k	1.23	4.35	128	0.009				

the best performance. The first technique relies on the combination of handcrafted local and global geometric descriptors for representing the shape of each protein point cloud. The local descriptor corresponds to the Fast Point Feature Histograms (FPFH) [15] which has shown in the literature its efficiency to characterize the local geometry around a surface point of a 3D object. The global descriptor is the Globally Aligned Spatial Distribution (GASD) [16] that is generally used for object recognition and pose estimation problems using point clouds. The combination of the two descriptors requires as a prior step a FPFH-based protein global representation. To this end, VLAD (Vector of Locally Aggregated Descriptors) technique [17] is exploited for gathering the FPFH associated to each point of the protein into a single descriptor. This global FPFH is then concatenated to the GASD resulting in a hybrid global descriptor. The final descriptor (FPFH+GASD) of dimension 1,024 is then used to measure the similarity between a protein query and the proteins of the dataset. For the similarity measure, we use a L2 distance. The two other techniques exploit the two descriptors separately.

4.5.2 Processing pipeline

- Data Preprocessing: it consists of a subsampling and a normalization of the protein point cloud. The subsampling allows to reduce the quantity of considered protein points to approximately 20% in order to go faster for further calculations. To this end, we use the simplification method proposed in Benhabiles *et al.* [18]. The main advantages of this method are its ability to preserve the global shape of the protein thanks to a uniform subsampling and its swiftness thanks to the use of advanced data structures. The normalization is performed by calculating the minimum bounding sphere [19] of the protein in order to rescale it into a unit sphere $S(c, r)$ where c is the center set to 0 and r is the radius set to 1. This makes each protein of the dataset invariant to geometric transformations including scale and translation.

- GASD descriptor calculation [16]: the descriptor consists of firstly estimating a reference frame of the point cloud using PCA (Principal Component Analysis) approach. The output of this step is three axis x, y and z calculated on the basis of the covariance matrix of the pre-processed point cloud and the associated eigenvectors. This reference frame is exploited to transform the point cloud into a canonical coordinate system making it pose invariant. The final global descriptor is then fitting points of the cloud with respect to a regular grid of 3D voxels. In our experiment the size of the resulting GASD vector is set to 512 dimensions.

- FPFH global representation (VLAD-FPFH): this stage goes through significant intermediate steps since the usual FPFH descriptor is only calculated for each point of the cloud and does not directly provide a global representation of the protein. In what follows, we describe the different steps that allow to build a global FPFH descriptor.

  - Local FPFH calculation [15]: the first step consists of calculating an FPFH descriptor for each point of the cloud (protein). To this end, the surface normals of all the points are calculated. Each point is then considered with its k nearest neighbors to calculate the angular variations between the normals of all possible pairs within the neighborhood. The angular variation is based on a Darboux frame construction. This result out into a 33 dimensions' feature vector for each point of the cloud.

- FPFH based relevant feature selection [15]: the goal of this step is to select for each protein the most relevant FPFH vectors from the ones calculated in the previous step. More precisely, many of the calculated FPFH vectors are redundant in all the dataset and thus do not reveal prominent characteristics that allow to distinguish between protein classes. To reach this goal an average FPFH vector is calculated over all the dataset and a distance distribution to this average vector is calculated for each protein. The obtained distribution for each protein which is close to a Gaussian is exploited to select the outlier vectors (vectors of interest). In our experiment, these vectors are those for which the distance is out of range $[d-\delta, d+\delta]$ where d is the mean distance within a protein and δ is set to a large value in such way to collect 0.5% of the total number of FPFH vectors. 64 FPFH vectors are retained for each protein.
- Vector of Locally Aggregated Descriptors (VLAD) [17]: this latter step allows to obtain a compact global vector for each protein. To this end, we first calculate an FPFH vector-based k-means over all the dataset. In our experiment we set k to 128 with centroids of 33 dimensions. Then we apply the accumulation process of local descriptors (the selected ones in previous step) for each protein to generate a global vector of 4,224 dimensions.

- Hybrid global descriptor (GASD + VLAD-FPFH): before to combine the global FPFH vector (VLAD-FPFH) with the GASD one, we apply a PCA (Principal Component Analysis) technique on each VLAD-FPFH descriptor to reduce its dimension from 4,224 to 512. This normalization aims at balancing the weights of both descriptors (GASD and VLAD-FPFH) in the final hybrid descriptor (GASD-VLAD).

4.5.3 Running time

The different steps described in the processing pipeline have been coded in C/C++ using PCL [20] on an Intel Core i7-6700HQ CPU@2.60 GHz with 32 GB of memory. The running times in seconds of each step are reported in table 2 for two protein point clouds having respectively the smallest and biggest size in the dataset. It is worth mentioning that the written code for each step has not been optimized to run in a parallel fashion.

5 Results

5.1 Overall results

Each team submitted one to three 5298×5298 dissimilarities matrix resulting in 13 different matrices from 5 teams to evaluate. Two teams were unable to produce results by the deadline. Tables 3 and 4 summarize the results, computed for each result matrix over all classes, at the *species* and *proteins* level, respectively. Precision-Recall curves are presented in figure 2. A quick overview show that all methods performed better at the *proteins* level than at the *species* level. Also, methods relying on the same descriptors (such as 3D Zernike descriptors) showed similar overall results while two different descriptor comparison methods can improve the retrieval performances (3DZD1 and 3DZD3 at the *proteins* level, for instance).

5.2 Precision-Recall curves

Figure 2 shows the retrival performance of each method at the *proteins* and *species* levels. The precision indicates the relevance of the results from a query, while the recall reflects the proportion of relevant results successfully retrieved.

3DZD, 3DZM and HAPT methods all display very good precision ($>$0.75) even for high recall values. Furthermore, these methods produced very similar performances when using different descriptors comparison methods (for 3DZD) or parameters (for HAPT). Using a training dataset of protein surfaces, the 3DZD method improve the precision at high recall values. The two training approach produce the same results at the *proteins* level, while the 3DZD2 approach, trained to discriminate surfaces at the *species* level, display better at the *species* level.

ConvLDSNet and Ft-PSSC displayed lower performances as evaluated by the precision-recall curves, with precision decreasing as the recall increases. From the three submitted set of results computed with the ConvLDSNet method, the ConvLDSNet2 variant (using the MolMovDB [11] only as a training set) show slighlty better results compared to the ConvLDSNet3 variant (trained using the MolMovDB and the SHREC18' protein shape retrieval dataset [8]). The ConvLDSNet1

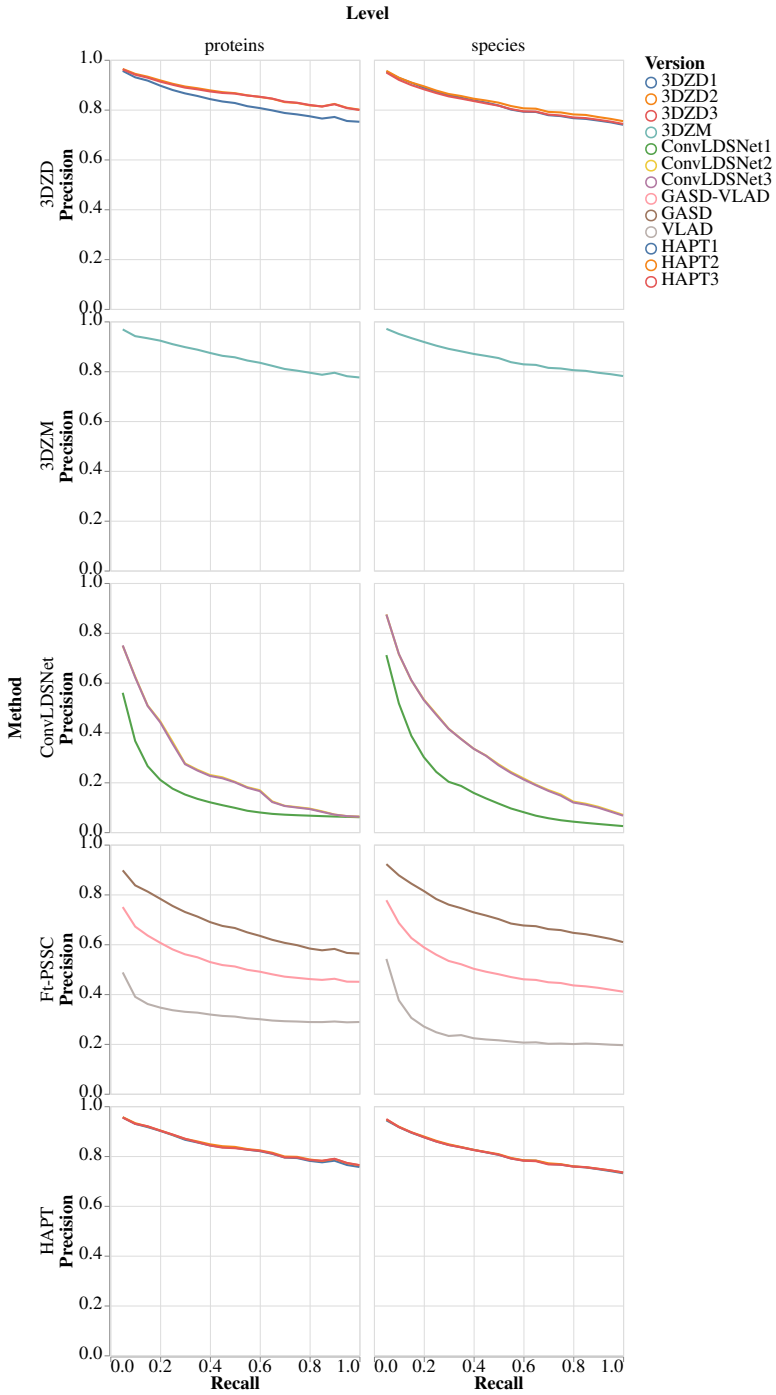

Figure 2: Precision-Recall curves for the *proteins* (left) and *species* (right) level. For each row, the precision-recall curves of all submitted results are shown.

variant (using the SHREC18' protein shape retrieval dataset only for training) show significantly lower performance. Regarding the Ft-PSSC method, using the VLAD (Vector of Locally Aggregated Descriptors)

technique [17] alone or in combination significantly damaged the retrieval performances.

5.3 Retrieval statistics

As one might expect, all retrieval statistics are increased at the *proteins* level compared to the *species*; this can be explained as the classes are significantly more populated in the *proteins* level (Table 1).

Similarly to the Precision-recall curves, 3DZD, 3ZDM and HAPT displayed better retrival performances. These methods showed really good performance at retrieving proteins at the *species* level, and were able, from a given protein shape, to retrieve a member of the same class at the *protein* level in more than 99% of the cases. Furhermore, these methods also showed high first-tier and second-tier values. The training used in the 3DZD2 and 3DZD3 approaches improve the 3DZD1 results for the first- and second-tier statistics, but not for the nearest-neigbor or the E-measure values. This training approach also allowed the 3DZD2 and 3DZD3 sets of results to outperform the 3DZM approach for the first- and second-tier statistics while the 3DZM approach display slightly better results for the nearest-neighbor metrics at the *species* level. The HAPT method displayed very similar results for the 3 sets of results. Interestingly, the 3DZD2-3 and HAPT1-3 are almost equivalent at the *species* level while the 3DZD2-3 methods produced better first-tier, second-tier, E-measure and DCG at the *proteins* level.

The Ft-PSSC method using the GASD descriptor only displayed the same performance as the previous methods regarding the Nearest-Neigbor statistics. This method displayed significantly lower performance as compared to the 3DZD, 3DZM and HAPT methods for the other statistics, however. The VLAD technique, when used, produced lower performances in all statistics. Regarding the ConvLDSNet method, the best results are produced when the MolMovDB dataset only: using the SHREC18' protein retrieval dataset only resulted in significantly lower performances.

5.4 Execution times

Table 5 summarizes, for each method, the execution times of the 3D descriptors calculations and the 3D descriptors comparison when provided by the authors. The 3D descriptors calculations are in the range of 3.95 to 13.02s, except for the ConvLDSNet method which uses GPUs to accelerate calculations (2.5ms). In comparison, descriptors comparisons consumed lesser time (from 0.002 ms for ConvLDSNet to 0.5s for 3DZM) for an individual calculation.

6 Discussion

- The SHREC19' dataset was highly homogenous (surface of similar size, only globular proteins from orthologs, *etc*)
- SHREC18' dataset did not improve methods' performance (strikingly examplified by ConvLDSNet) and even damaged ConvLDSNet performance when used as lone training set
- Need to use additional features to discriminate between similar proteins (electrostatics, *etc*) to improve overall preformance of the methods

References

1. Berman, H. M. *et al.* The Protein Data Bank. *Nucleic Acids Research* **28,** 235–242 (2000).
2. Connolly, M. L. Analytical molecular surface calculation. *Journal of Applied Crystallography* **16,** 548–558 (Oct. 1983).
3. Fox, N. K., Brenner, S. E. & Chandonia, J.-M. SCOPe: Structural Classification of Proteins—extended, integrating SCOP and ASTRAL data and classification of new structures. *Nucleic Acids Research* **42,** D304–D309 (2014).
4. Chandonia, J.-M., Fox, N. K. & Brenner, S. E. SCOPe: Manual Curation and Artifact Removal in the Structural Classification of Proteins – extended Database. *Journal of Molecular Biology* **429.** Computation Resources for Molecular Biology, 348–355 (2017).
5. Fox, N. K., Chandonia, J.-M. & Brenner, S. E. SCOPe: classification of large macromolecular structures in the structural classification of proteins—extended database. *Nucleic Acids Research* **47,** D475–D481 (Nov. 2018).
6. Xu, D. & Zhang, Y. Generating Triangulated Macromolecular Surfaces by Euclidean Distance Transform. *PLOS ONE* **4,** 1–11 (Dec. 2009).

Table 3: Macro-average values computed over each *species* class: Nearest-neighbor (NN), First Tier (1st tier), Second Tier (2d tier), E-measure and Discounted Cumulative Gain (DCG). The values are computed at the *species* level for each matrix submitted.

Method	Approach	NN	1st tier	2d tier	E-measure	DCG
3DZD	3DZD1	0.960	0.523	0.639	0.251	0.871
3DZD	3DZD2	0.965	0.581	0.694	0.250	0.884
3DZD	3DZD3	0.950	0.570	0.706	0.243	0.882
3DZM	3DZM	0.986	0.565	0.661	0.270	0.884
ConvLDSNet	ConvLDSNet1	0.458	0.173	0.256	0.168	0.536
ConvLDSNet	ConvLDSNet2	0.604	0.308	0.427	0.335	0.646
ConvLDSNet	ConvLDSNet3	0.602	0.308	0.421	0.332	0.645
Ft-PSSC	GASD-VLAD	0.671	0.260	0.378	0.116	0.715
Ft-PSSC	GASD	0.945	0.377	0.461	0.213	0.800
Ft-PSSC	VLAD	0.248	0.141	0.237	0.033	0.603
HAPT	HAPT1	0.951	0.549	0.693	0.240	0.875
HAPT	HAPT2	0.954	0.556	0.695	0.241	0.877
HAPT	HAPT3	0.945	0.557	0.695	0.243	0.878

Table 4: Macro-average values computed over each *proteins* class: Nearest-neighbor (NN), First Tier (1st tier), Second Tier (2d tier), E-measure and Discounted Cumulative Gain (DCG). The values are computed at the *protein* level for each matrix submitted.

Method	Approach	NN	1st tier	2d tier	E-measure	DCG
3DZD	3DZD1	0.994	0.588	0.742	0.129	0.919
3DZD	3DZD2	0.995	0.662	0.794	0.131	0.935
3DZD	3DZD3	0.990	0.673	0.810	0.129	0.936
3DZM	3DZM	0.996	0.585	0.708	0.135	0.919
ConvLDSNet	ConvLDSNet1	0.606	0.176	0.264	0.126	0.628
ConvLDSNet	ConvLDSNet2	0.806	0.267	0.354	0.248	0.710
ConvLDSNet	ConvLDSNet3	0.806	0.264	0.353	0.246	0.709
Ft-PSSC	GASD-VLAD	0.784	0.316	0.484	0.074	0.807
Ft-PSSC	GASD	0.982	0.365	0.500	0.113	0.845
Ft-PSSC	VLAD	0.407	0.231	0.386	0.030	0.734
HAPT	HAPT1	0.991	0.622	0.738	0.125	0.918
HAPT	HAPT2	0.993	0.631	0.741	0.126	0.920
HAPT	HAPT3	0.989	0.619	0.737	0.126	0.918

7. Wu, Z. *et al. 3D ShapeNets: A deep representation for volumetric shapes* in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2015), 1912–1920.

8. Langenfeld, F. *et al. Protein Shape Retrieval* in *Eurographics Workshop on 3D Object Retrieval* (eds Telea, A., Theoharis, T. & Veltkamp, R.) (The Eurographics Association, 2018).

9. Maturana, D. & Scherer, S. *VoxNet: A 3D Convolutional Neural Network for real-time object recognition* in *2015 IEEE/RSJ International Con-*

Table 5: Execution times and hardwares used to compute and compare 3D descriptors. NP: not provided by the participants.

Method	Descriptors calculation (hardware)	Descriptors comparison (hardware)
3DZM	2.95s (CPU)	0.5s (CPU)
ConvLDSNet	2.5ms (GTX1070)	0.002ms (i7-6700K)
3DZD	3.48s (NP)	0.1774s (Titan X)
Ft-PSSC		
HAPT	4.98s / 11.1s / 13.02s (i7-4720HQ)	NP

ference on Intelligent Robots and Systems (IROS) (Sept. 2015), 922–928.

10. Dimou, A., Ataloglou, D., Dimitropoulos, K., Alvarez, F. & Daras, P. LDS-Inspired Residual Networks. *IEEE Transactions on Circuits and Systems for Video Technology,* 1–1 (2018).
11. Echols, N., Milburn, D. & Gerstein, M. MolMovDB: Analysis and visualization of conformational change and structural flexibility. *Nucleic acids research* **31,** 478–82 (Feb. 2003).
12. Sael, L. *et al.* Fast protein tertiary structure retrieval based on global surface shape similarity. *Proteins: Structure, Function, and Bioinformatics* **72,** 1259–1273 (2008).
13. Kihara, D., Sael, L., Chikhi, R. & Esquivel-Rodriguez, J. Molecular Surface Representation Using 3D Zernike Descriptors for Protein Shape Comparison and Docking. *Current Protein & Peptide Science* **12,** 520–530 (2011).
14. Giachetti, A. & Lovato, C. *Radial symmetry detection and shape characterization with the multiscale area projection transform* in *Computer Graphics Forum* **31** (2012), 1669–1678.
15. Rusu, R. B., Blodow, N. & Beetz, M. *Fast Point Feature Histograms (FPFH) for 3D registration* in *2009 IEEE International Conference on Robotics and Automation* (May 2009), 3212–3217.
16. D. M. Lima, J. P. S. & Teichrieb, V. *An Efficient Global Point Cloud Descriptor for Object Recognition and Pose Estimation* in *2016 29th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)* (Oct. 2016), 56–63.
17. Jégou, H. *et al.* Aggregating Local Image Descriptors into Compact Codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34,** 1704–1716 (Sept. 2012).
18. Benhabiles, H., Aubreton, O., Barki, H. & Tabia, H. *Fast simplification with sharp feature preserving for 3D point clouds* in *2013 11th International Symposium on Programming and Systems (ISPS)* (Apr. 2013), 47–52.
19. Welzl, E. *Smallest enclosing disks (balls and ellipsoids)* in *New Results and New Trends in Computer Science* (ed Maurer, H.) (Springer Berlin Heidelberg, Berlin, Heidelberg, 1991), 359–370.
20. *Point Cloud Library* `http://pointclouds.org`. Accessed: 2019-03-08.