SHREC 2020: multi-domain protein shape retrieval challenge

Florent Langenfeld, Yuxu Peng, Yu-Kun Lai, Paul L. Rosin, Tunde Aderinwale, Genki Terashi, Charles Christoffer, Daisuke Kihara, Halim Benhabiles, Karim Hammoudi, et al.

Computers & Graphics (2020)

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

SHREC2020 track: Multi-domain Protein Shape Retrieval Challenge

Florent Langenfeld[a,1], Yuxu Peng[b], Yu-Kun Lai[c], Paul L. Rosin[c], Tunde Aderinwale[d], Genki Terashi[e], Charles Christoffer[d], Daisuke Kihara[d,e], Halim Benhabiles[f], Karim Hammoudi[g,h], Adnane Cabani[i], Feryal Windal[f], Mahmoud Melkemi[g,h], Andrea Giachetti[j], Stelios Mylonas[k], Apostolos Axenopoulos[k], Petros Daras[k], Ekpo Otu[l], Reyer Zwiggelaar[l], David Hunter[l], Yonghuai Liu[m], Matthieu Montès[a,1,]

[a] *Laboratoire de Génomique, Bio-informatique et Chimie Moléculaire (GBCM), Conservatoire National des Arts-et-Métiers, 2, rue Conté, Paris, 75003, France; HESAM Université*
[b] *School of Computer and Communication Engineering, Changsha University of Science & Technology, Changsha, 410114, Hunan Province, China*
[c] *School of Computer Science and Informatics, Cardiff University, Cardiff, CF24 3AA, UK*
[d] *Department of Computer Science, Purdue University, West Lafayette, IN, 47907, USA*
[e] *Department of Biological Sciences, Purdue University, West Lafayette, IN, 47907, USA*
[f] *Univ. Lille, CNRS, Centrale Lille, Univ. Polytechnique Hauts-de-France, Yncrea Hauts-de-France, UMR 8520 - IEMN, F-59000 Lille, France*
[g] *Université de Haute-Alsace, Department of Computer Science, IRIMAS, F-68100 Mulhouse, France*
[h] *Université de Strasbourg, France*
[i] *Normandie University, UNIROUEN, ESIGELEC, IRSEEM, 76000 Rouen, France*
[j] *Department of Computer Science, University of Verona*
[k] *Information Technologies Institute, Centre for Research and Technology Hellas, Greece*
[l] *Department of Computer Science, Aberystwyth University, Aberystwyth, SY23 3DB, UK*
[m] *Department of Computer Science, Edge Hill University, Ormskirk, L39 4QP, UK*

ARTICLE INFO

Article history:
Received May 4, 2020

ABSTRACT

Acknowledgments

Yuxu Peng was supported by the Young teachers growth plan project (2019QJCZ014) funded by Changsha University of Science & Technology.

Stelios Mylonas, Apostolos Axenopoulos and Petros Daras were supported by the ATXN1-MED15 PPI project funded by the GSRT - Hellenic Foundation for Research and Innovation.

Matthieu Montes and Florent Langenfeld were supported by the European Research Council Executive Agency under the research grant number 640283.

e-mail: florent.langenfeld@cnam.fr (Florent Langenfeld), pengyuxupjl@csust.edu.cn (Yuxu Peng), LaiY4@cardiff.ac.uk (Yu-Kun Lai), RosinPL@cardiff.ac.uk (Paul L. Rosin), taderinw@purdue.edu (Tunde Aderinwale), gterashi@purdue.edu (Genki Terashi), christ35@purdue.edu (Charles Christoffer), dkihara@purdue.edu (Daisuke Kihara), halim.benhabiles@yncrea.fr (Halim Benhabiles), karim.hammoudi@uha.fr (Karim Hammoudi), adnane.cabani@esigelec.fr (Adnane Cabani), feryal.windal@yncrea.fr (Feryal Windal), mahmoud.melkemi@uha.fr (Mahmoud Melkemi), andrea.giachetti@univr.it (Andrea Giachetti), smylonas@iti.gr (Stelios Mylonas), axenop@iti.gr (Apostolos Axenopoulos), daras@iti.gr (Petros Daras), eko@aber.ac.uk (Ekpo Otu), rrz@aber.ac.uk (Reyer Zwiggelaar), dah56@aber.ac.uk (David Hunter), liuyo@edgehill.ac.uk (Yonghuai Liu), matthieu.montes@cnam.fr (Matthieu Montès)

[1] Track organizers and corresponding authors

Computers & Graphics (2020)

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

SHREC2020 track: Multi-domain Protein Shape Retrieval Challenge

ARTICLE INFO

Article history:
Received July 25, 2020

Keywords: 3D Shape Analysis, 3D Shape Descriptor, 3D Shape Retrieval, 3D Shape Matching, Protein Shape, SHREC

ABSTRACT

Proteins are natural modular objects usually composed of several domains, each domain bearing a specific function that is mediated through its surface, which is accessible to vicinal molecules. This draws attention to an understudied characteristic of protein structures: surface, that is mostly unexploited by protein structure comparison methods. In the present work, we evaluated the performance of six shape comparison methods, among which three are based on machine learning, to distinguish between 588 multi-domain proteins and to recreate the evolutionary relationships at the *protein* and *species* levels of the SCOPe database.

The six groups that participated in the challenge submitted a total of 15 sets of results. We observed that the performance of all the methods significantly decreases at the *species* level, suggesting that shape-only protein comparison is challenging for closely related proteins. Even if the dataset is limited in size (only 588 proteins are considered whereas more than 160,000 protein structures are experimentally solved), we think that this work provides useful insights into the current shape comparison methods performance, and highlights possible limitations to large-scale applications due to the computational cost.

1. Introduction

Proteins are complex macro-molecular molecules with various shapes and sizes ranging from hundreds to millions of atoms [1]. The 3D arrangement of protein atoms is directly linked to specific functions that are mostly mediated through the protein surface. Protein surfaces are of great interest in drug discovery pipelines, adverse drug reaction or the characterization of cellular processes at the molecular level. However, challenges in protein surfaces comparison may arise from a) the dynamical, non-rigid nature of the proteins that allows protein conformational changes, i.e., surficial modifications and therefore specific functions, b) the intrinsic structure of multi-domain proteins, i.e., the fusion of multiple, individual domains into one protein throughout evolution, and c) the similarity between distinct protein structures and surfaces inherited from their evolutionary relationships.

The SHape REtrieval Challenges (SHREC) are time-restricted challenges, which aim to evaluate the effectiveness of 3D-shape retrieval algorithms. Typically, a challenge is opened by proposing a dataset of related shapes to participants while retaining the class membership. In the SHape REtrieval Challenge 2020 (SHREC2020) track on multi-domain protein shapes, the participants had 7 weeks from the dataset publication to send their results with a description of the methods used to generate the results (see Section 4). This SHREC2020 track on multi-domain protein shapes evaluates the current ability of shape comparison methods proposed by 6 different groups to tackle the protein surface comparison problem. The participants were asked to send their results in the form of matrices containing all-to-all dissimilarity scores. The results were analyzed and the overall retrieval performances are presented here.

The dataset includes 588 proteins consisting of two domains (the functional units of the proteins); only the corresponding triangulated meshes of their solvent-excluded surfaces (SES) [2] were provided as input to the participants. We then evaluated the retrieval performance of each method to retrieve the

evolutionary relationships between orthologous proteins (proteins that have the same function in different organisms), and to retrieve the different conformations of an individual protein. Here, we present the results of all the participants and methods, and briefly discuss the trade-off between performance in retrieval and computational cost of each method.

2. Dataset

Proteins are linear polymers (the so-called protein chains) made of amino-acid residues (up to several hundreds), which fold into a specific, well-defined 3D structure. Furthermore, many proteins need to form a complex of several chains to become functional. For instance, the human heamoglobin requires two α-globin and two β-globin chains to be fully functional. Domains define the functional units of the proteins, and are usually associated with a specific function and/or interaction; it is thus commonplace for two proteins to share one domain while their other respective domains differ. This characteristic led to the development of databases classifying proteins according to both their structure and the functions of their domains. The SHREC2020 track on multi-domain protein shapes dataset is devoted to the analysis of protein shapes generated from protein chains that comprise two domains.

Dataset creation. The SCOPe database [3, 4, 5] organizes the protein domains according to their structural (in the 2 top levels of the SCOPe tree) and evolutionary (for the 4 bottom levels) relationships. Protein domains in the SCOPe database originate from Protein Data Bank (PDB) experimental structures [6], and are characterized by their PDBId and chainId, allowing for filtering based on these parameters. From all entries implemented in the SCOPe tree (excluding entries from the 'Artifacts' and 'Low resolution protein structures' classes), we kept only the entries from X-ray crystallography PDB structures composed of two domains. When multiple copies of the same protein chain was present in the same PDB structure, we only kept one of those copies to limit redundancies. Finally, all proteins were required to have at least one orthologous protein, and classes with less than 10 members were discarded.

Table 1. Number of classes and number of shapes in the dataset, at the *protein* and the *species* levels.

Level	Number of classes	Shapes by class (min / max)
Protein	7	19 / 168
Species	26	12 / 63

Ground truth generation. The ground truth was generated using the resulting SCOPe tree of two-domains proteins. Only the biggest domain (highest number of amino-acid residues) was used to define two ground truth classifications, namely the *protein* and *species* levels, which reproduce the SCOPe tree classifications at the *protein* and *species* levels, respectively. These classifications were not provided to the track participants. By using this protocol, 588 protein chains were retrieved, from 26 orthologs (proteins having the same activity in different organisms such as the human and murine haemoglobin proteins) and 7 proteins (see Table 1). The solvent-excluded surfaces [2] were computed for all the entries using EDTSurf [7] (non-protein atoms were discarded) after protonation of the struture using propka [8, 9], and only the corresponding .off files were provided to the participants on the track website (`http://shrec2020.drugdesign.fr`). At the end of the track, the ground truths were published online as well. As the participants were not provided some important details about the dataset creation (two-domains proteins only, protonation and SES calculation parameters, ...), the reverse engineering of the memberships from the surfaces (.off files) would require to compare all the PDB entries of the SCOPe database to the dataset. While feasible in principle, this approach in practice would be difficult to carry out.

Compared to other known protein shapes datasets [10, 11], this dataset is exclusively composed by two-domains proteins while only one-domain proteins were included in [10, 11]. As multi-domain proteins are commonplace at the cellular level, the impact of additional domains on the protein shape retrieval performances need to be evaluated. Recently, another dataset of protein surface patches was published [12], encompassing both geometric and chemical features of proteins surfaces. That dataset gathers partially overlapping patches rather than complete proteins surfaces, and is currently limited to structures that display specific functionalities, namely the ability to bind selected small molecules or to form a protein-protein complex.

3. Evaluation

Analyses were performed with scikit-learn [13] and numpy [14], and Figures 4 and 5 were produced using matplotlib [15].

Nearest Neighbor, First-tier and Second-tier. These retrieval metrics measure the ratio of models that belong to the same class as the query. For Nearest Neighbor (NN), the first match only is considered (the identity is not considered), while the $|C|-1$ and $2*(|C|-1)$ first matches, where $|C|$ denotes the size of the query's class, are considered for First-tier (T1) and Second-tier (T2); the maximum value for the Second-tier is therefore 0.5.

Precision-Recall plot. Precision P refers to the ratio of results that are relevant and is computed as the number of models from class C retrieved within all objects attributed to class C, while Recall R represents the number of results correctly classified and is computed as the number of models from class C retrieved compared to the size $|C|$ of the class C.

Mean Average Precision. Given a query, its average precision is the average of all precision values computed when each relevant object is found. Given several queries, the mean average precision (MAP) is the mean of average precision of each query. It then gives in a single

value the overall retrieval performance of an algorithm.

All metrics were macro-averaged at the *protein* and *species* levels, as defined in the SCOPe database.

4. Participants & Methods

Six groups from five different countries registered for the track and submitted 15 dissimilarity matrices in the requested time (8 weeks) along with the description of their protocol. To ease the reading, we have assigned each group a short name for referencing in the following text.

1. CODSEQ by Author A, Author B, Author C, Author D, Author E (subsection 4.1),
2. 3DZ by Author F, Author G, Author H, Author I (subsection 4.2),
3. WKS/SGWS by Author J, Author K, Author L (subsection 4.3),
4. HAPT by Author M (subsection 4.4),
5. GraphCNN by Author N, Author O, Author P (subsection 4.5),
6. HAPPS by Author Q, Author R, Author S, Author T (subsection 4.6).

4.1. 3D Characterization of prOteins by Deep analysis of 2D view SEQuence (CODSEQ) — Author A, Author B, Author C, Author D, Author E

Table 2. Running times in seconds of each stage of the CODSEQ framework for one protein.

3D mesh size of one protein	247650 facets
2D views extraction $9 \times (312 \times 312)$	6.75
2D descriptor (9×512)	4
Compact 3D descriptor (1024)	2.46
Distance to all proteins dataset (588 proteins)	0.007

Table 3. Training times in seconds using GPU for the used CNN-based models.

CNN-based model	Training data size	Epochs	Training time (seconds)
Inception ResNet [16]	12798 images	5	1260
LSTM-RNN [17]	1422 sequences	40	13

The CODSEQ method is a deep learning based framework for indexing proteins. The approach consists of capturing surface details of the 3D proteins under the form of a set of 2D views. To this end, a classification architecture was tailored by exploiting a transfer learning strategy to extract relevant features from the considered views of proteins. The SHREC 2018 dataset [10] was exploited for the training stage as these protein surfaces share similar silhouettes with the protein surfaces proposed in the current contest (connected stretched shapes). The protein surfaces from SHREC 2019 dataset have not been exploited since their shapes visually seemed too different (compact shapes). Only protein classes represented by at least 19 (18 train and 1 test) different proteins (dominant classes) were considered, resulting in selecting 79 classes among 107 classes of the SHREC 2018 dataset. Noteworthy, the training stage has only been performed at the *protein* level since the SHREC 2018 dataset does not include the *species* level. The train/test methodology has been adopted thanks to the availability of the ground-truths.

Descriptors calculation.

Extraction of protein 2D views. In this stage, 3D meshes representing protein surfaces are simplified using the Quadric Error Metric Decimation [18]. By this way, the number of facets of each 3D mesh has been reduced to 20,000 facets (about 10% of the original surface) while maintaining the surface details. In the considered coordinate system related to the processing, each 3D mesh has its own position and size. These singular parameters are mainly due to the devices and conditions of acquisition that can vary from one protein to another. For normalizing the set of simplified 3D meshes of protein, each of them is recentered and rescaled with a sphere having a center of 0 and a radius of 1 as explained in [19]. This allows to obtain protein surfaces invariant to geometric affine transformations considering scale and translation. A sequence of 2D views (312×312 RGB images) is then extracted using a set of virtual cameras uniformly positioned around the bounding sphere of each protein. 9 views are enough for covering the whole surface of the protein.

Protein characterization based on a single 2D view (2D descriptor). The goal of this stage is to extract a feature descriptor from each 2D view using a transfer learning strategy. More precisely, an Inception ResNet architecture [16] pretrained on ImageNet dataset has been fine-tuned and trained on SHREC2018 dataset [10] in order to learn 79 protein classes. The trained model is used to return a 512-dimensional feature vector for each 2D view by getting the output of a penultimate layer (the one before the classes output).

Protein characterization based on a sequence of 2D views (Compact 3D descriptor). A bidirectional LSTM-RNN architecture [17] was trained on the SHREC2018 dataset to learn, as in the previous stage, 79 classes. The architecture has been fed with sequences of feature vectors obtained in the previous stage; each sequence is composed of 9 feature vectors associated to 9 views of a given protein. One of the strengths of RNN-based models is their ability to analyze data sequences (sequences of views in the current case) while keeping the most significant views to characterize protein classes. Indeed, a classification accuracy rate of 96% on the test data derived from

SHREC2018 dataset [10] was reached using this trained bidirectional LSTM-RNN model. This model was used to extract a 1024-dimensional feature vector for each protein of the present contest.

Dissimilarity distance calculation & runtimes. Dissimilarity matrices were generated by calculating the Euclidean distance between each pair of proteins using their associated 1024-dimensional feature vectors. Two matrices have been generated based on two training runs performed in the previous stage, namely CODSEQ1 with 0.96 and 0.18 of accuracy and loss, respectively, and CODSEQ2 with 0.94 and 0.14 of accuracy and loss, respectively.

This framework has been developed in Python 3.7.6 using Open3D 0.8.0.0, OpenCV 4.2.0 and Keras 2.2.4-tf on a TensorFlow-GPU 2.1.0 backend. The experiments have been conducted on an Intel Core i7-6700HQ CPU@2.60 GHz with 32 GB of memory and NVIDIA GeForce GTX 1070 GPU with 24 GB of memory. The running times in seconds of each stage performed on CPU are reported in Table 2 for one protein. Table 2, shows the training times of the used CNN-based models trained on GPU.

4.2. Network trained with encoded 3DZD and 3DZM (3DZ) — Author F, Author G, Author H, Author I

Three dissimilarity matrices of target protein surfaces were generated unsing three methods based on the 3D Zernike Descriptor (3DZD) or the 3D Zernike Moment (3DZM). 3DZM are the coefficients for representing a 3D shape function in terms of 3D Zernike-Canterakis polynomials [20]. 3DZD is the rotation-invariant shape descriptor derived from the 3DZM [21].

Descriptors calculation. Using the 3DZD or 3DZM as the feature of protein shape, a neural network was trained to output a score that measures the (dis)similarity between a pair of protein shapes. The framework is the same with the one in the SHREC2019 protein shape retrieval contest (see Section 4.3 in [11]). The network has an encoder, a feed-forward fully-connected neural network with an input layer and three hidden layers with a ReLU activation function. The network takes 3DZD or 3DZM of a protein shape as input. The three hidden layers have 250, 200, and 150 neurons, respectively, which are used for the encoding of an input 3DZD (or 3DZM). The encoder is connected to the feature comparator, a fully-connected network, which takes the 3DZD (or the 3DZM) of the two proteins, and the encodings from the three hidden layers, and four metrics that compare two vectors, the Euclidean distance, the cosine distance, the element-wise absolute difference, and the element-wise product, and the two features of the two protein shapes (the difference in the number of vertices and faces). In total, the number of the input features of the feature comparator is $2*121$ (or 1771 for 3DZM) $+2*(250+200+150)+2*4+2 =$ 1452 features (4752 features for 3DZM). The first term is the 3DZDs of order 20 (n=20), which is a 121 element vector, of the two protein shapes. The third term, 2 * 4 comes from the four comparison metrics applied to two representations of the two proteins, the original 3DZDs (or 3DZMs) and encodings, which concatenate the output of the input layer and the three intermediate layers of the encoder. The feature comparator outputs a score between 0 and 1 using a sigmoid activation function, which is the probability that the two proteins are in the same protein level classification in the SCOPe database [3]. The feature comparator network has an input layer of a 1452-dimensional feature vector, two intermediate layer of 100 and 50 neurons respectively, and one output neuron.

The network was trained on a dataset of 247,521 protein structures from the SCOPe 2.07 database. Proteins in Class I (Artifacts) were not included. To augment data for training the network for 3DZM, which is not rotation invariant, each protein was rotated with different random orientations. For each protein,EDTSurf [7] was used to generate the solvent excluded surface, which was then fed into the EM-Surfer pipeline [22] to compute 3DZM and 3DZD. The network was trained to correctly distinguish proteins in the same protein level category in SCOPe from the rest.

Dissimilarity distance calculation & runtimes. The first dissimilarity matrix submitted was computed with the network trained with 3DZDs. The second matrix was computed with the network trained on a vector of a size 1771, which was the absolute values of complex numbers in 3DZM. The distances in the third matrix were the average between the Euclidean distance of 3DZDs, and the distances in the first and the second matrices. Generating 3DZD and 3DZM takes ~8.00 seconds on average for each protein on an Intel(R) Xeon(R) CPU E5-2630 0 @ 2.30GHz. The 3DZD model took ~0.22 seconds on average to predict the dissimilarity between two proteins using TitanX GPU, while the 3DZM model took ~0.5 seconds on the same GPU. The Euclidean model took ~0.17 seconds on average per prediction and the averaging of the three matrices was almost instant and was negligible.

4.3. Wave Kernel Signature and SGWS based Shape Descriptor for Protein Retrieval (WKS/SGWS) — Author J, Author K, Author L

To reach robust and improved performance, a hybrid spectral feature descriptor is used which combines the benefits of features of wave kernel signature (WKS) [23] and spectral graph wavelet transform (SGWS) [24]. WKS is an isometric invariant descriptor that has been found to be effective for deformable 3D shape retrieval such as those of the dataset; in contrast to HKS, it focusses on the high-frequency information. SGWS is a generalisation of HKS and WKS, and provides a multiresolution local descriptor that is compact, easy to compute and combines the advantages of both band-pass and low-pass filters.

Data pre-processing. Meshes were simplified to reduce the number of faces to approximately 6000 using Qslim [18] which provides an effective compromise between the fastest algorithms and the highest-quality algorithms to reduce computing time. Then the mesh is fixed using the open source software meshfix [25] to convert a raw digitized polygon mesh to a clean mesh where all the occurrences of a specific set of “defects”

are corrected. Holes, self-intersections, degenerate and non-manifold elements are all replaced with valid configurations.

WKS descriptor calculation. The WKS feature vectors are computed from the eigenvalues and the eigenvectors of each protein mesh. Then the vocabulary is calculated using an improved vector-based k-means over 10% feature vectors of all proteins [26]. Finally, the WKS descriptor is normalized for the bag-of-features (BoF) for each protein using hard vector quantization. The lengths of the WKS feature vector and the descriptor are 50 and 1000 respectively.

SGWS descriptor calculation. The process of the SGWS descriptor is similar to that of the WKS descriptor. The SGWS feature vectors were computed first, and then the vocabulary and bag of feature were obtained. The lengths of the SGWS feature vector and the descriptor are 5 and 1000 respectively.

Hybrid spectral descriptor (WKS + SGWS). The hybrid spectral descriptor combines the normalized BoF of WKS and SGWS to form a long vector which is 2000-dimensional.

Dissimilarity matrices computation & runtimes. The procedure for model comparison consists of computing bags of features and measuring distances between shapes. For the similarity measure, the L1 distance $||X-Y||_1$ is used. The estimation of the descriptors takes 37 seconds on average, running on a laptop with an i5-5200U CPU, RAM 4GB, running Windows 10. The descriptor comparison time was negligible.

4.4. Histogram of Area Projection Transform (HAPT) — Author M

The method characterizes protein shapes with the Histograms of Area Projection Transform (HAPT) [27]. This descriptor, well suited for non-rigid shape retrieval, is based on a spatial map (Multiscale Area Projection Transform) [27] 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:

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

where S is the surface of the object (see Figure 1), $T_R(S, \vec{n})$, 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):

$$MAPT(x, y, z, R, S) = \alpha(R)\, APT(x, y, z, S, R, \sigma(R)) \qquad (2)$$

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

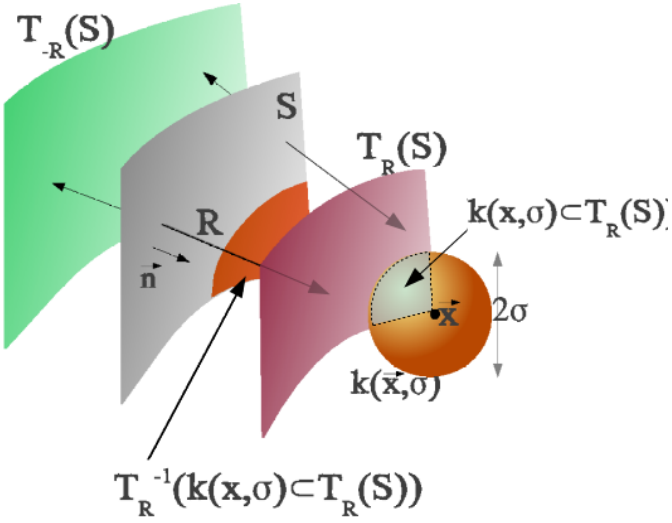


Fig. 1. APT measures the area of the part of the input surface that, projected along the normal at a selected distance, is included in a circular neigborhood of the point of interest (see subsection 4.4).

Descriptors calculation. A discrete MAPT is easily computed, for selected values of R, on a voxelized grid including the surface mesh, with the procedure described in [27]. 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 a unique descriptor. Voxel side and sampled radii were fixed for each run and chosen to represent the approximate radii of the spherical symmetries visible in the models.

Three different options were tested for the algorithm's parameters. In HAPT1, $s = 0.3$, the MAPT histograms were computed for 12 increasing radii starting from $R_1 = 0.3$ iteratively adding a fixed step of 0.3 for the remaining values, and c was set to 0.5. In HAPT2, $s = 0.3$, the MAPT histograms were computed for 8 increasing radii starting from $R_1 = 0.3$ iteratively adding a fixed step of 0.3 for the remaining values, and c was set to 0.5. In HAPT3, $s = 0.4$, 8 increasing radii (from $R_1 = 0.8$ and a fixed step of 0.4 for the remaining values) were used to compute the MAPT histograms, and c was set to 0.5.

Dissimilarity matrices computation & runtimes. The procedure for model comparison consists in concatenating the MAPT histograms computed at the different scales and measuring distances between shapes by evaluating the Jeffrey divergence [28] of the corresponding concatenated vectors. The estimation of the descriptors took 112 seconds on average for the run HAPT1, 47 seconds on average for the run HAPT2, and 17 seconds on average for the run HAPT3 on a laptop with an Intel CoreTM i7-9750H CPU running Ubuntu Linux 18.04. The descriptor comparison time was negligible.

Table 4. GraphCNN's network configuration.

Input	MLP	Encoder1 (ρ_1)	Encoder2 (ρ_2)	Encoder3 (ρ_3)	Encoder4	Output
3D points (10K, 3)	MLP (3, 32)	SPH3D(64, 64, 2) SPH3D(64, 64, 1) Pool(10K, 2500)	SPH3D(64, 64, 1) SPH3D(64, 128, 2) Pool(2500, 625)	SPH3D(128, 128, 1) SPH3D(128, 128, 1) Pool(625, 156)	G-SPH3D (128, 512)	FC(832, 512) FC(512, 256) FC(256, *C*)

4.5. Graph-based CNN (GraphCNN) for 3D shape retrieval — Author N, Author O, Author P

Following the recent tendency of addressing many scientific tasks by exploiting the existing vast amount of data, a data-driven approach was applied for the problem of 3D protein shape retrieval. Based on the fact that the provided input proteins are in the form of triangulated meshes, a transfer learning approach was applied. A method originally designed for the task of 3D point cloud classification and segmentation was adapted to the needs of the protein shape retrieval task, and trained on a relevant dataset of protein 3D point clouds in order to learn appropriate features (descriptors) for the representation of 3D molecular shapes.

Descriptors calculation. SPH3D-GCN [29], a graph-based CNN method equipped with a novel spherical convolution kernel, was employed as it has achieved state-of-the-art results on numerous computer vision tasks. The detailed architecture of the applied network is depicted in Table 4. From each triangulated protein surface, a number of 10000 points is uniformly sampled, since the network requires a constant number of input points. After transforming the input 3D coordinates to a higher dimensional space of 32 features with a multilayer perceptron (MLP), four encoder blocks are applied. Each encoder operates on a specific spatial range, which is denoted by ρ. Parameter ρ controls the radius of the applied spherical kernels and determines the spatial extent of the applied convolutions. SPH3D(α, β, γ) represents a separable spherical convolution that takes as input α channels, performs a depth-wise convolution with a multiplier γ and subsequently a point-wise convolution to generate the output β channels. At the end of each decoder, a pooling operation is applied, which reduces gradually the number of considered points. In Encoder4, a modified spherical convolution is applied in order to obtain a global representation of the whole point cloud. Finally, the output features of all the four encoders are concatenated and imported to a sequence of three fully connected (FC) layers. The proposed scheme was trained on the dataset from last years competition (SHREC2019 [11]), which comprises 5298 structures from 17 protein classes. The network was trained on a classification task aiming to assign each structure to each corresponding protein class. During the feature extraction step, the FC layers were dropped and the concatenated output of the four encoders were used as descriptors. Therefore, for each previously unseen input, a feature vector of 832 values is extracted.

Dissimilarity matrices computation & runtimes. 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 class. Among the three GraphCNN submissions, various sets of radius ρ were experimented. Specifically, the first one (GraphCNN1) corresponds to (ρ_1, ρ_2, ρ_3) = (0.05, 0.1, 0.2), the second one (GraphCNN2) to (0.05, 0.15, 0.45) and the third (GraphCNN3) to (0.1, 0.2, 0.4). The calculation of descriptors took on average 45 milliseconds per mesh sample on a GeForce GTX1070 GPU, while the training time is about 1 hour on the same GPU. The average comparison time between two descriptors is negligible (0.001 milliseconds on an Intel Core i7- 6700K CPU).

4.6. Hybrid Augmented Point Pair Signatures (HAPPS) — Author Q, Author R, Author S, Author T

Descriptors can be categorised into two main groups: ***local*** and ***global***. Combining two or more descriptors (e.g., *local-local*, *local-global*, or *global-global*) yields a third category, the ***hybrid*** descriptor - aimed at improving the resultant performance of the combined descriptors. The Hybrid Augmented Point Pair Signature (HAPPS) is a 3D shape descriptor in the third category, computed from a combination of two separate descriptors: ***local*** Augmented Point Pair Feature Descriptor (APPFD), and ***global*** Histogram of Global Distances (HoGD) or Multi-view 2D Projection (M2DP) [30] descriptors, each of which are computed using hand-crafted features extracted from 3D surface. Details of APPFD, HoGD, and M2DP descriptors are provided in the following sections.

HAPPS is an improvement over the APPFD, aimed at achieving better retrieval performances. Although the latter is capable of robustly representing 3D shapes, a closer inspection of protein shapes for this retrieval challenge reveals identical local surface characteristics and somewhat uniqueness in global appearances between the Protein shapes, hence the need to extend the capability of the APPFD and effectively capture both local and global characteristics of the Protein shapes. Therefore, two global 3D descriptors were separately combined: The Histogram of Global Distances (HoGD) and Multi-view 2D Projection (M2DP) with APPFD to derive two variants of hybrid descriptor: the Hybrid Augmented Point Pair Signatures (HAPPS), referred to as HAPPS-1 and HAPPS-2, i.e., hybrid descriptors formed by combining local APPFD with global HoGD and M2DP, respectively. Alongside the APPFD, the HAPPS algorithm was first introduced in [31] and recorded very high performance scores across several 3D benchmark datasets. Fig. (2) presents an overview of the HAPPS algorithms.

Augmented Point Pair Feature Descriptor (APPFD). The Augmented Point Pair Feature Descriptor (APPFD) is a 3D shape descriptor, which describes the ***local*** geometry around

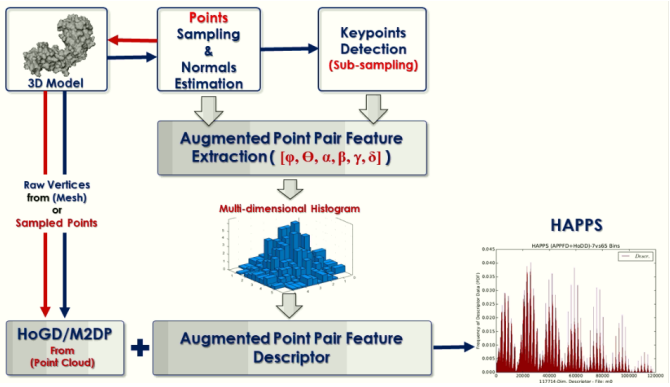


Fig. 2. Overview of HAPPS algorithm.

a point, $p = [p_x, p_y, p_z]$ or vertex, $v = [v_x, v_y, v_z]$ for 3D pointcloud or mesh datasets, respectively. Here, the pointcloud shape representation for this task was used, instead. Computing this descriptor involves the following stages: (i) pointcloud sampling and normals estimation, (ii) keypoint, p_{k_i} determination, (iii) local surface region (i.e., LSP), P_i selection, (iv) Augmented Point Pair Feature (APPF) extraction per LSP, and (v) final descriptor computation. The algorithms for stages (iv) and (v) are described in this section, and the reader is referred to the literature in [31] for more details on the other stages.

Feature Extraction. The first step of APPFD is to compute keypoints, $p_{k_i}, i = 1, 2, \cdots$, and locally extract four-dimensional Point Pair Feature (PPF), $f_1 = (\alpha, \beta, \gamma, \delta)$ as in [32] from r-nearest neighbourhood, $\{P_i, i = 1 : K\}$ of each keypoint $\{p_{k_i}, i = 1 : K\}$, where K is the number of keypoints for a given 3D shape. For every pair of points, p_i, p_j and their estimated normals, n_i, n_j i.e., oriented points, $[(p_i, n_i), (p_j, n_j)]$ $(i \neq j)$, in P_i where p_i is the ***origin*** w.r.t. the constraint in Equation (3) holding ***True***, a transformation-independent Darboux frame U, V, W is defined as: $U = n_i$, $V = U \times ((p_j - p_i)/\delta)$, $W = U \times V$.

$$|n_i \cdot (p_j - p_i)| \leq |n_j \cdot (p_j - p_i)| \tag{3}$$

Alternatively, p_j becomes the ***origin*** (i.e., point with the larger angle between its associated normal and the line connecting the two points) if the constraint in (3) is ***False***, and the variables in (3) are reversed. f_1 is then derived for the source point as follows:

$$\alpha = \arctan(W \cdot n_j, U \cdot n_j), \;\; \alpha \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \tag{4}$$

$$\beta = V \cdot n_j, \;\; \beta \in [-1, 1] \tag{5}$$

$$\gamma = U \cdot \frac{p_j - p_i}{\|p_j - p_i\|}, \;\; \gamma \in [-1, 1] \tag{6}$$

$$\delta = \|p_j - p_i\| \tag{7}$$

Secondly, $f_2(p_i, p_j) = (\phi, \theta)$ is extracted for every possible combination of points pair, p_i, p_j in P_i, because f_1 is not robust enough to capture the entire geometric information for a given LSP. In addition, the PPF approach opens up possibilities for additional feature space. Therefore, as illustrated in Figure 3, ϕ is the angle of the projection of the vector $\overrightarrow{S}$ onto the unit vector $\overrightarrow{V_2}$, while θ is geometrically the angle of the projection of the vector, $\overrightarrow{S}$ onto the unit vector $\overrightarrow{V_1}$, where $\overrightarrow{V_1} = p_i - p_c$, $\overrightarrow{V_2} = p_i - l$, and $\overrightarrow{S} = p_i - p_j$, with $p_c = \frac{1}{n_i}\sum_{i=1}^{n_i} p_{k_i}$ (i.e., LSP centroid), and $l = (p_j - p_c)$, the vector location of p_{k_i} *w.r.t.* its LSP. Note that p_i, p_j, p_c, and l are all points in $\mathbb{R}^3$ space, although l is a vector.

Basically, α, β, γ are the angular variations between (n_i, n_j), while δ is the spatial distance between p_i and p_j. In Euclidean geometry, each of the projections ϕ and θ is considered angle between two vectors. For example $\angle_1\langle\overrightarrow{S}, \overrightarrow{V_1}\rangle$ and $\angle_2\langle\overrightarrow{S}, \overrightarrow{V_2}\rangle$ are equivalent to θ and ϕ respectively. These angles are derived by taking the scalar products of $(\overrightarrow{S} \cdot \overrightarrow{V_1})$ for $\angle_1$, and $(\overrightarrow{S} \cdot \overrightarrow{V_2})$ for $\angle_2$ about a point p_i in a given LSP. Mathematically, scalar products defined in this manner are homogeneous (i.e., invariant) under scaling [33] and rotation [34]. For this reason, the two-dimensional local geometric features, ϕ and θ, are considered rotation and scale invariant for 3D shapes under rigid and non-rigid affine transformations.

Local APPF Descriptor. Lastly, for every possible combination, q of oriented point pair, $p_i, p_j = [(p_i, n_i), (p_j, n_j)]$ in an LSP, (P_i, N_i), $q(q-1)/2$ six-dimensional APPF: $f_3 = (f_2 + f_1)$ are locally obtained thus: $f_3(p_i, p_j) = (f_2(p_i, p_j), f_1(p_i, p_j)) = (\phi, \theta, \alpha, \beta, \gamma, \delta)$, then vertically stacked together and discretized into a multi-dimensional histogram with bins = 7 in each feature-dimension, flattened and normalized to give $7^6 = 117649$-dimensional single local descriptor (APPFD) per 3D shape.

In computing APPFD for this task, points and their normals, (P, N), where $|P| = 3500$ and 4200, were sampled from each 3D shape and K keypoints were computed, $\{p_{k_i}, i = 1 : K\}$), around which LSPs, $\{P_i, i = 1 : K\}$) and their corresponding normals, $\{N_i, i = 1 : K\}$) were extracted, within a specified radius, $r = 0.40 - 0.50$ for each p_{k_i}.

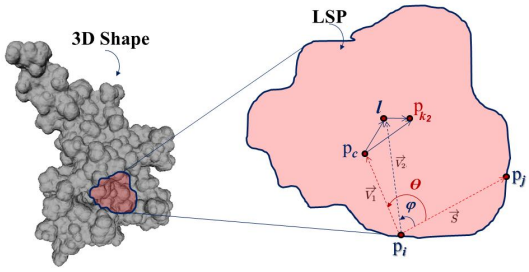


Fig. 3. Local Surface Patch (LSP), P_i with pairwise points (p_i, p_j) as part of a surflet-pair relation for (p_i, n_i) and (p_j, n_j), with p_i being the origin. θ and ϕ are the angles of vectors projection about the origin, p_i. θ is the projection angle from vector $\langle p_i - p_j\rangle$ to vector $\langle p_i - p_c\rangle$ while ϕ is the projection angle from vector $\langle p_i - p_j\rangle$ to vector $\langle p_i - l\rangle$. The LSP centre is given by p_c, keypoint is given as p_{k_i} where $i = 2$. Finally, l is the vector position of $p_{k_i} - p_c$.

Histogram of Global Distances (HoGD). Considering that a shape is represented by a discrete set of points, P on its surface which forms the external and internal contour of the shape, a set of normalized vectors $\delta_i = \|p_c - p_i\|$ was denoted between the centroid p_c of a given 3D shape to all other points on its surface, where $p_i \in P$. Such normalized vectors δ_i are regarded as global features whose distribution (histogram) is capable of expressing the configuration of the entire shape relative to its centroid, and is a rich description of the global structure of the shape. These global features were discretized into a histogram with $\sqrt{|P|} \approx 65$ bins, normalized to give HoGD, which is very fast and straightforward to compute - with $|P| = 3500$ and 4200, as in APPFD. Finally, HoGD is combined with APPFD to give HAPPS-1, with 117649 + 65 = 117714-dimensional final feature vector, **FV**. See Figure 2 for an overview of the HAPPS algorithm.

Multi-view 2D Projection (M2DP). The M2DP is a global descriptor for 3D point cloud applied for loop closure detection in [30]. It involves the projection of 3D cloud to multiple 2D planes from which density signature of points in each plane is computed and combined to produce 196-dimensional **FV**. This descriptor was adopted for HAPPS-2 due to its success and computational efficiency, and refer the reader to the literature in [30] for more details on M2DP. Again, using $|P| = 3500$ and 4200 as in previous cases, HAPPS-2 is a 117649 + 196 = 117845-dimensional **FV**.See Figure 2 for an overview of the HAPPS algorithm.

Shape Similarity Measurement. Overall, the L_2 or cosine distance metric between **FV**s are expected to give good approximations of the similarity between shapes in the SHREC2020 Protein dataset. The cosine metric was adpted in Equation (8), due to a slightly more improvement over the L_2 metric.

$$\cos(\mathbf{FV_1}, \mathbf{FV_2}) = \frac{\mathbf{FV_1} \cdot \mathbf{FV_2}}{\|\mathbf{FV_1}\|\|\mathbf{FV_2}\|} = \frac{\sum_{i=1}^{n} \mathbf{FV_1}_i \mathbf{FV_2}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{FV_1}_i)^2} \sqrt{\sum_{i=1}^{n} (\mathbf{FV_2}_i)^2}} \tag{8}$$

Dissimilarity matrices computation & runtimes. Two parameters of APPFD, r and vs (i.e., *voxel-size*, a parameter that determines how big or small an occupied voxel grid can be, during pointcloud down-sampling to yield keypoints [35]), influence the overall performances of the HAPPS retrieval algorithms. r is directly proportional to LSP size while vs is inversely proportional to the number of sub-sampled points (keypoints), which implies that increasing the values of r and vs increases the size of LSP and reduces the number of keypoints, and vice versa. Computational time and memory are affected by them, hence the configurations summarized in Table 5 were carefully selected for experimental run1, run2, and run3.

The HAPPS algorithms were implemented in Python 3.6.0 and all experiments were carried out under Windows 7 desktop PC with Intel Core i7-4790 CPU @ 3.60GHz, 32GB RAM. On average, it took 23 seconds and 45 seconds to compute HAPPS-1 and HAPPS-2, respectively, about 1 second to extract (P, N), and roughly 0.3 seconds each, to compute HoGD and M2DP per 3D shape.

Table 5. HAPPS settings for experimental runs 1, 2 and 3.

Expts.	Algorithms	Parameter Settings				
		P	r	vs	$bins_{appfd}$	$bins_{hogd}$
run-1	HAPPS-1	4200	0.40	0.20	7	65
run-2	HAPPS-1	3500	0.50	0.20	7	65
run-3	HAPPS-2	3500	0.50	0.20	7	-

5. Results & discussion

In this section, we assess quantitatively the performance of each method described in Section 4. We analyzed the performance at the *protein* (Fig. 4 and Table 6) and the *species* (Fig. 5 and Table 7) levels as described in Section 3.

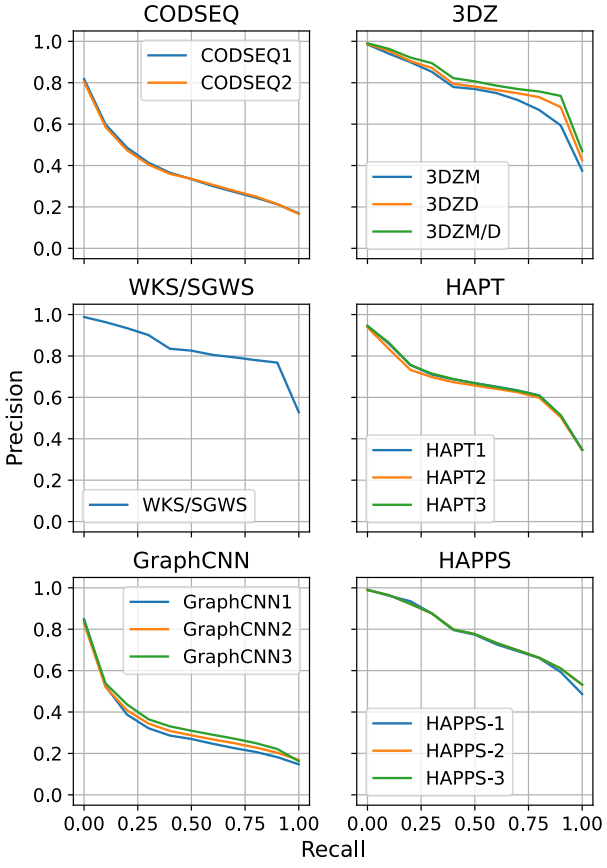


Fig. 4. Precision-Recall curves at the *protein* level.

Protein level. At the *protein* level, the 588 shapes were gathered into 7 classes of multi-domain orthologous proteins; among each class, all members share at least one common domain while the other domains are different.

Table 6. Evaluation metrics at the *protein* level. NN = Nearest Neighbor, T1 = First-tier, T2 = Second-tier, MAP = Mean Average Precision. For each metric, the highest value is in bold.

Method	NN	T1	T2	MAP
CODSEQ1	0.697	0.350	0.266	0.358
CODSEQ2	0.666	0.345	0.264	0.356
3DZD	0.978	0.753	0.428	0.797
3DZM	0.975	0.719	0.422	0.766
3DZD/3DZM average	0.980	0.789	0.436	0.823
WKS/SGWS	**0.985**	**0.818**	**0.438**	**0.840**
HAPT1	0.898	0.617	0.407	0.658
HAPT2	0.875	0.602	0.402	0.646
HAPT3	0.892	0.620	0.406	0.659
GraphCNN1	0.773	0.278	0.218	0.301
GraphCNN2	0.734	0.295	0.235	0.317
GraphCNN3	0.770	0.310	0.243	0.339
HAPPS-1	0.982	0.738	0.416	0.774
HAPPS-2	0.983	0.746	0.420	0.779
HAPPS-3	0.983	0.746	0.420	0.779

This feature allows the methods for having nearest-neighbor (NN) over the whole dataset ranging from 66.6 up to 98.5%, meaning that for a given query, the shape comparison algorithms were able to retrieve a query of the same class in at least two thirds of the cases. The performances vary largely between methods, as three methods (3DZM/D, WKS/SGWS and HAPPS1-3) achieve successful nearest-neighbor retrieval in more than 97.5% of the cases. For all the methods, the performances decrease as we consider further results, but the performance drops are different for each method, as illustrated by the differences in the precision-recall (PR) curves profiles (Fig. 4) and the First-tier (T1) and Second-tier (T2) values (Table 6). This results in a wide range of MAP values (from 0.301 to 0.840).

Species level. At the *species* level, the dataset contains 26 classes. Within each *protein* class, the *species* classes were evolutionary-related proteins. For instance, the *protein* class "T-cell antigen receptor" has two *species* child classes, the human and the murine orthologs, which display 71.5% of amino-acid sequence identity and a strong structural similarity.

Therefore, and similarly to the last two SHREC tracks on protein shape retrieval [10, 11], the performances of the shape comparison methods are significantly lowered at the *species* level compared to the *protein* level. The NN values range from 43.8 to 84.4%, while no method displays a T1 value greater than 0.46. The PR curve profiles are characterized by steepest slopes indicating lower precision values at the same recall values when compared to the *protein* level.

Machine learning approaches have recently been applied to protein surface patches [12]. In the present track, three of the six methods make use of learning approaches in their work-flows. Their performances are comparable to the performances of the other methods, showing no improvement in the retrieval results. Interestingly, the two learning-based methods trained on SHREC2018 and SHREC2019 tracks on protein shape retrieval (CODSEQ and GraphCNN, respectively) were outperformed by the learning-based method trained on the whole SCOPe dataset (3DZ). This latter training dataset encompasses multi-domains protein shapes while datasets from the last two SHREC tracks on protein shape retrieval only encompass one-domain protein shapes. Particularly, the CODSEQ method showed lower performances on the SHREC2020 multi-domain protein dataset compared to their training dataset (see section 4.1) for which the CODSEQ approach shows relatively high performance. This may originate from the specificites of these two sets (one-domain *versus* multi-domain proteins shapes); besides, the CODSEQ was only trained at the *protein* level. These remarks also stand for the GraphCNN method, which used the dataset from the SHREC2019 track on protein shape retrieval, another one-domain protein shapes dataset, to train their network.

Performance / Computation cost trade-off. The Protein Data Bank (PDB) is the most populated database for the protein structures. As of May 2020, more than 160,000 structures have been deposited and more than 11,000 new structures are deposited every year. Furthermore, the size of the proteins deposited is growing as the performance of experimental protein structure resolution methods are improving, and the number of multi-domain proteins follows this trend. The ability to screen such a large database in a reasonable time and with acceptable performances is therefore a challenge.

The two main steps of the shape comparison are the computation of a descriptor for each object, and the comparison between two descriptors. Depending on the algorithm, the descriptor computation times are ranging from 8 (3DZ) to 112 (HAPT1) seconds for descriptors computed on a CPU, and 45 milliseconds for the descriptors computed on a GPU (GraphCNN). The

Table 7. Evaluation metrics at the *species* level. NN = Nearest Neighbor, T1 = First-tier, T2 = Second-tier, MAP = Mean Average Precision. For each metric, the highest value is in bold.

Method	NN	T1	T2	MAP
CODSEQ1	0.438	0.173	0.125	0.180
CODSEQ2	0.447	0.172	0.124	0.179
3DZD	0.783	0.391	0.262	0.435
3DZM	0.722	0.369	0.256	0.402
3DZD/3DZM average	0.825	0.419	0.277	0.470
WKS/SGWS	**0.844**	**0.460**	**0.298**	**0.508**
HAPT1	0.595	0.286	0.209	0.313
HAPT2	0.572	0.264	0.200	0.289
HAPT3	0.608	0.286	0.209	0.313
GraphCNN1	0.513	0.177	0.117	0.178
GraphCNN2	0.533	0.175	0.120	0.186
GraphCNN3	0.499	0.181	0.122	0.186
HAPPS-1	0.757	0.407	0.272	0.432
HAPPS-2	0.772	0.400	0.269	0.430
HAPPS-3	0.768	0.400	0.269	0.430

Table 8. For each method, running times of descriptor computation for one protein, descriptors comparison and, when applicable, training times. Descriptors computation and comparison times are expressed in seconds, training time units are specified. The type of hardware (CPU = Central Processing Unit, GPU = Graphics Processing Unit) used is indicated in parenthesis. N/A = Not Applicable.

Method	Descriptor calculation	Descriptor comparison	Training Time
CODSEQ	13.21 (CPU)	0.007 (CPU)	2 hours (GPU)
3DZ	8.0 (CPU)	0.17-0.5 (GPU)	1 week
WKS/SGWS	37 (CPU)	negligible (CPU)	N/A
HAPT	17-112 (CPU)	negligible (CPU)	N/A
GraphCNN	0.045 (GPU)	negligible (CPU)	1 hour (GPU)
HAPPS	23-45 (CPU)	1.6 (CPU)	N/A

comparison between two descriptors are in the order of millisecond or below, except for the 3DZD and 3DZM descriptor comparisons which are in the range of 0.22-0.5 seconds on a GPU. Regarding the learning-based methods, the computation times are ranging from 20 minutes to 1 week. Carefully assessing the performance / computational cost ratio is therefore required if one aims to screen a large database as the computation cost may prove prohibitive for large-scale screening projects.

6. Conclusion

In the present work, we have presented a dataset of shapes from multi-domain proteins. Six groups, among which three used machine learning approaches in their respective workflows, submitted 15 sets of results. The performances were assessed at the *protein* and *species* levels of the SCOPe database.

Shape retrieval methods displayed high-quality results at the *protein* level. We observed a significant decrease in the performances of all the methods at the *species* level. These results indicate that comparing multi-domains proteins based on their shapes only remains challenging, especially for closely related proteins. It could be of interest to compare shape-retrieval methods to the reference methods used in the structural biology community.

Protein structures in the PDB are highly heterogeneous; missing (i.e., not solved by the experiment) atoms or residues at the surface of a protein is a very common phenomenon in PDB structures and can be considered as a noisy signal for the protein shape comparison. Their impact on the performances in retieval should be carefully evaluated.

It is common in drug design processes to compare proteins in order to find out a protein-specific feature against which to design a new, specific drug and limit drug adverse effects. In the upcoming years, we might propose similar tracks by adding other surficial properties such as the electrostatic potential to help determine whether combined methods (shape + surficial properties) improve the predictive power of the shape-only methods. It may also stimulate the development of new dedicated, protein-dedicated methods.

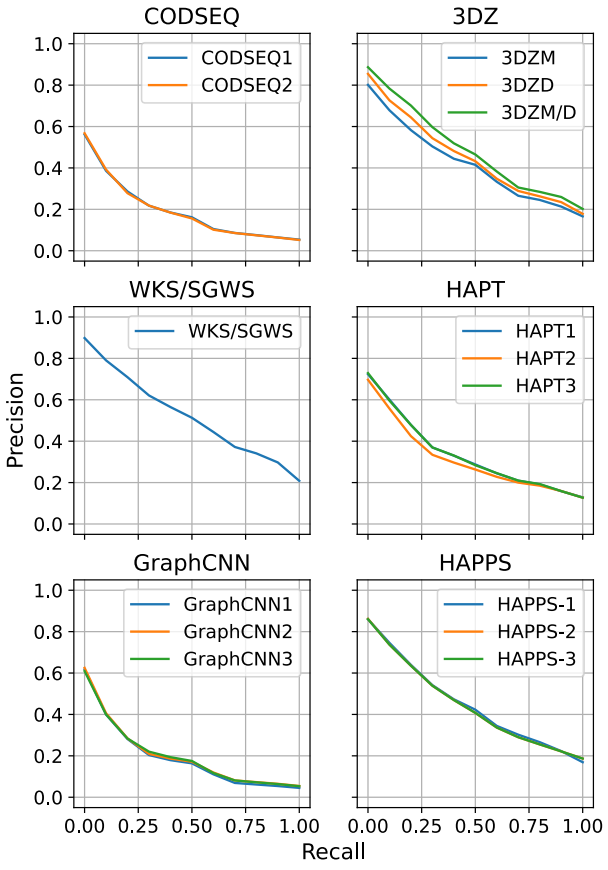


Fig. 5. Precision-Recall curves at the *species* level.

Acknowledgments

References

[1] Berg, J, Stryer, L, Tymoczko, J, Gatto, G. Biochemistry. 9 ed.; Macmillan Learning; 2019. ISBN 9781319114657.

[2] Connolly, ML. Analytical molecular surface calculation. Journal of Applied Crystallography 1983;16(5):548–558. doi:10.1107/S0021889883010985.

[3] Fox, NK, Brenner, SE, Chandonia, JM. SCOPe: Structural classification of proteins—extended, integrating SCOP and ASTRAL data and classification of new structures. Nucleic Acids Research 2013;42(D1):D304–D309. doi:10.1093/nar/gkt1240.

[4] Chandonia, JM, Fox, NK, Brenner, SE. Scope: Manual curation and artifact removal in the structural classification of proteins extended database. Journal of Molecular Biology 2017;429(3):348 – 355. doi:10.1016/j.jmb.2016.11.023; computation Resources for Molecular Biology.

[5] Fox, NK, Chandonia, JM, Brenner, SE. Scope: classification of large macromolecular structures in the structural classification of proteinsextended database. Nucleic Acids Research 2018;47(D1):D475–D481. doi:10.1093/nar/gky1134.

[6] Berman, HM, Westbrook, J, Feng, Z, Gilliland, G, Bhat, TN, Weissig, H, et al. The protein data bank. Nucleic Acids Research 2000;28(1):235–242. doi:10.1093/nar/28.1.235.

[7] Xu, D, Zhang, Y. Generating triangulated macromolecular surfaces by euclidean distance transform. PLoS ONE 2009;4(12):e8140. doi:10.1371/journal.pone.0008140.

[8] Sndergaard, CR, Olsson, MHM, Rostkowski, M, Jensen, JH. Improved treatment of ligands and coupling effects in empirical calculation and rationalization of pka values. Journal of Chemical Theory and Computation 2011;7(7):2284–2295. doi:10.1021/ct200133y.

[9] Olsson, MHM, Sndergaard, CR, Rostkowski, M, Jensen, JH. Propka3: Consistent treatment of internal and surface residues in empirical pka predictions. Journal of Chemical Theory and Computation 2011;7(2):525–537. doi:10.1021/ct100578z.

[10] Langenfeld, F, Axenopoulos, A, Chatzitofis, A, Craciun, D, Daras, P, Du, B, et al. Shrec 2018 protein shape retrieval. In: Eurographics Workshop on 3D Object Retrieval. 2018, p. 53–61. doi:10.2312/3dor.20181053.

[11] Langenfeld, F, Axenopoulos, A, Benhabiles, H, Daras, P, Giachetti, A, Han, X, et al. Protein Shape Retrieval Contest. In: Biasotti, S, Lavou, G, Veltkamp, R, editors. Eurographics Workshop on 3D Object Retrieval. The Eurographics Association; 2019,doi:10.2312/3dor.20191058.

[12] Gainza, P, Sverrisson, F, Monti, F, Rodolà, E, Boscaini, D, Bronstein, MM, et al. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. Nature Methods 2019;doi:10.1038/s41592-019-0666-6.

[13] Pedregosa, F, Varoquaux, G, Gramfort, A, Michel, V, Thirion, B, Grisel, O, et al. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 2011;12:2825–2830.

[14] Oliphant, T. NumPy: A guide to NumPy. USA: Trelgol Publishing; 2006–. URL: http://www.numpy.org/; [Online; accessed 2020-06-15].

[15] Hunter, JD. Matplotlib: A 2d graphics environment. Computing in Science & Engineering 2007;9(3):90–95. doi:10.1109/MCSE.2007.55.

[16] Szegedy, C, Ioffe, S, Vanhoucke, V, Alemi, AA. Inception-v4, inception-resnet and the impact of residual connections on learning. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence. AAAI17; AAAI Press; 2017, p. 42784284.

[17] Hochreiter, S, Schmidhuber, J. Long short-term memory. Neural Comput 1997;9(8):17351780. doi:10.1162/neco.1997.9.8.1735.

[18] Garland, M, Heckbert, PS. Surface simplification using quadric error metrics. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH 97; ACM Press/Addison-Wesley Publishing Co.; 1997, p. 209216. doi:10.1145/258734.258849.

[19] Benhabiles, H, Hammoudi, K, Windal, F, Melkemi, M, Cabani, A. A transfer learning exploited for indexing protein structures from 3D point clouds. In: Lepore, N, Brieva, J, Romero, E, Racoceanu, D, Joskowicz, L, editors. Processing and Analysis of Biomedical Information. Springer International Publishing. ISBN 978-3-030-13835-6; 2019, p. 82–89.

[20] Canterakis, N. 3D zernike moments and zernike affine invariants for 3D image analysis and recognition. In: In 11th Scandinavian Conf. on Image Analysis. 1999, p. 85–93.

[21] 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 2011;12(6):520–530. doi:10.2174/138920311796957612.

[22] Esquivel-Rodríguez, J, Xiong, Y, Han, X, Guang, S, Christoffer, C, Kihara, D. Navigating 3D electron microscopy maps with EM-SURFER. BMC Bioinformatics 2015;16(1). doi:10.1186/s12859-015-0580-6.

[23] Aubry, M, Schlickewei, U, Cremers, D. The wave kernel signature: A quantum mechanical approach to shape analysis. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops). IEEE Computer Society; 2011, p. 1626–1633. doi:10.1109/ICCVW.2011.6130444.

[24] Li, C, Hamza, AB. A multiresolution descriptor for deformable 3D shape retrieval. The Visual Computer 2013;29(6-8):513–524. doi:10.1007/s00371-013-0815-3.

[25] Attene, M. A lightweight approach to repairing digitized polygon meshes. The Visual Computer 2010;26(11):1393–1406. doi:10.1007/s00371-010-0416-3.

[26] Bronstein, AM, Bronstein, MM, Guibas, LJ, Ovsjanikov, M. Shape google. ACM Transactions on Graphics 2011;30(1):1–20. doi:10.1145/1899404.1899405.

[27] Giachetti, A, Lovato, C. Radial symmetry detection and shape characterization with the multiscale area projection transform. In: Computer Graphics Forum; vol. 31. Wiley Online Library; 2012, p. 1669–1678. doi:10.1111/j.1467-8659.2012.03172.x.

[28] Puzicha, J, Buhmann, J, Rubner, Y, Tomasi, C. Empirical evaluation of dissimilarity measures for color and texture. In: Proceedings of the Seventh IEEE International Conference on Computer Vision; vol. 2. 1999, p. 1165–1172. doi:10.1109/ICCV.1999.790412.

[29] Lei, H, Akhtar, N, Mian, A. Spherical kernel for efficient graph convolution on 3D point clouds. IEEE Transactions on Pattern Analysis and Machine Intelligence 2020;:1–1doi:10.1109/TPAMI.2020.2983410.

[30] He, L, Wang, X, Zhang, H. M2DP: A novel 3D point cloud descriptor and its application in loop closure detection. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE; 2016, p. 231–237. doi:10.1109/IROS.2016.7759060.

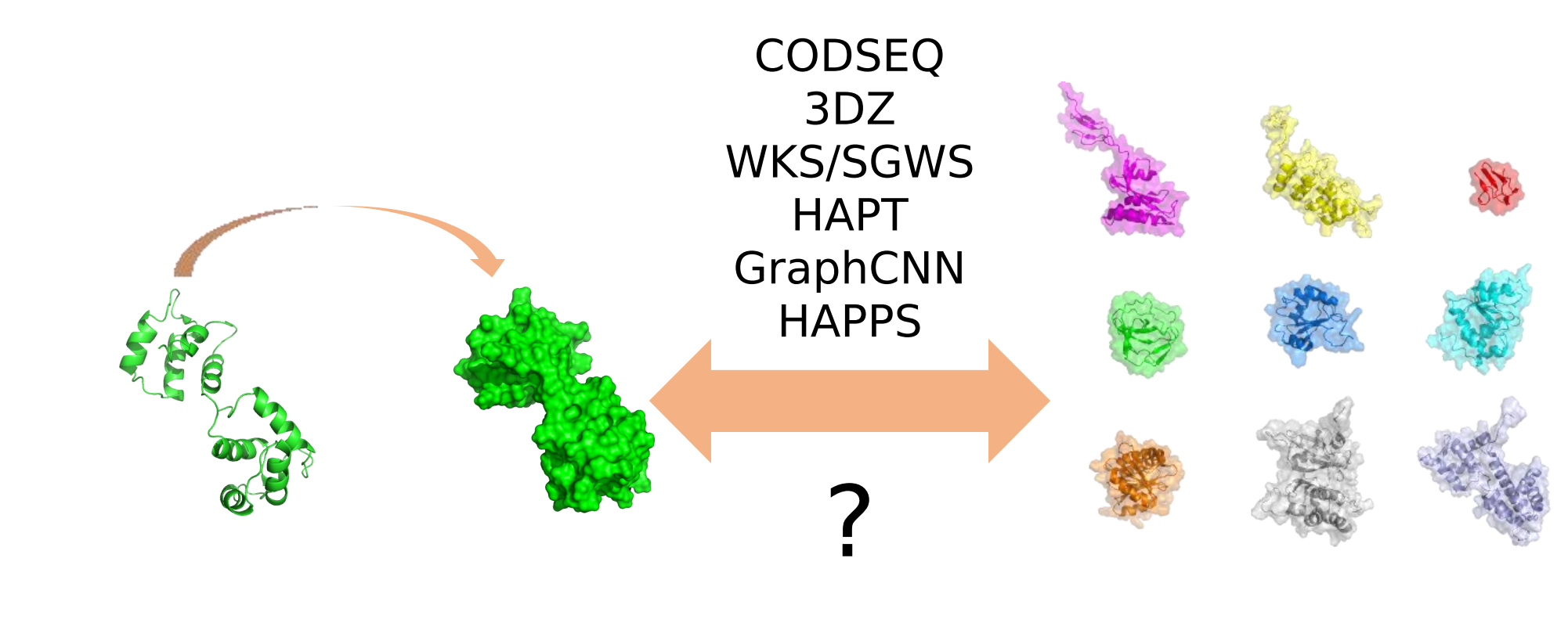
[31] Otu, E, Zwiggelaar, R, Hunter, D, Liu, Y. Nonrigid 3D shape retrieval with happs: A novel hybrid augmented point pair signature. In: 2019 International Conference on Computational Science and Computational Intelligence (CSCI). 2019, p. 662–668. doi:10.1109/CSCI49370.2019.00124.

[32] Wahl, E, Hillenbrand, U, Hirzinger, G. Surflet-pair-relation histograms: a statistical 3D-shape representation for rapid classification. In: Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings. 2003, p. 474–481. doi:10.1109/IM.2003.1240284.

[33] Wikipedia, . Dot Product. 2019. URL: https://en.wikipedia.org/wiki/Dot_product; [Online; accessed: 2019-10-15].

[34] Mathworld, W. Dot Product. 2019. URL: http://mathworld.wolfram.com/DotProduct.html; [Online; accessed: 2019-10-15].

[35] Zhou, QY, Park, J, Koltun, V. Open3D: A modern library for 3D data processing. arXiv:180109847 2018;.


CODSEQ
3DZ
WKS/SGWS
HAPT
GraphCNN
HAPPS
?