



A bioinformatics approach to 2D shape classification[☆]



Manuele Bicego*, Pietro Lovato

Dipartimento di Informatica, Università di Verona, Strada le grazie 15, 37134 Verona, Italy

ARTICLE INFO

Article history:

Received 10 June 2015

Accepted 21 November 2015

Available online 27 November 2015

Keywords:

2D shape classification

Bioinformatics

Sequence alignment

Visualization

ABSTRACT

In the past, the huge and profitable interaction between Pattern Recognition and biology/bioinformatics was mainly unidirectional, namely targeted at applying PR tools and ideas to analyse biological data. In this paper we investigate an alternative approach, which exploits bioinformatics solutions to solve PR problems: in particular, we address the 2D shape classification problem using classical biological sequence analysis approaches – for which a vast amount of tools and solutions have been developed and improved in more than 40 years of research. First, we highlight the similarities between 2D shapes and biological sequences, then we propose three methods to encode a shape as a biological sequence. Given the encoding, we can employ standard biological sequence analysis tools to derive a similarity, which can be exploited in a nearest neighbor framework. Classification results, obtained on 5 standard datasets, confirm the potentials of the proposed unconventional interaction between PR and bioinformatics. Moreover, we provide some evidences of how it is possible to exploit other bioinformatics concepts and tools to interpret data and results, confirming the flexibility of the proposed framework.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

Research in Computational Biology and Bioinformatics experienced an unprecedented growth in the last years, mainly due to the fruitful interaction with many disciplines and fields of computer science. Among others, Pattern Recognition/Machine Learning techniques have been successfully exploited in this context [1], for many different reasons: it is possible to “learn from examples”, derive quantitative models, handle non vectorial data, and deal with many classification, clustering and detection problems commonly encountered in life sciences. In many cases the particular Pattern Recognition model has not been applied “as is”, but has been adapted and modified to take into account biological constraints and needs. Sometimes, this produced approaches that are very different from original methodology – a clear example is the profile-HMMs [2].

To some extent, it can be stated that this tight interaction has been mainly unidirectional, with biology/life science gaining the largest benefit¹. In this paper, we explore an alternative direction, trying to answer the following question: *can we reverse the typical direction of interaction between Pattern Recognition and Bioinformatics?* Or, in

other words, *can we exploit advanced bioinformatics models and solutions to solve pattern recognition tasks?*

To the best of our knowledge, this perspective is rather new in the literature – the only relevant example is the video-genome project² [4] – and it seems a promising direction for two different reasons. First, if we are able to encode the Pattern Recognition problem in *biological terms* then we can exploit the huge range of effective, optimized, and interpretable bioinformatics tools developed by more than 40 years of research. These tools heavily rely on the solution of general pattern recognition tasks such as matching, classification, retrieval, clustering, distance computation and so on. For example, in the video-genome project [4], authors established an analogy between biological sequences and videos, defining the so called “video-DNA”, a way to map features extracted from video frames into nucleotidic biological sequences. Having encoded the problem in *biological terms*, authors were then able to address the video retrieval task by using the famous BLAST [5] – an extremely fast and effective heuristic-driven algorithm for biological sequence retrieval. Second, and more important, the main goal in bioinformatics research is to derive knowledge from biological data: therefore, the interpretability of methods and solutions is a key feature, and many visualization, inspection and interpretation tools are available in the literature. These tools may be very useful also in the Pattern recognition scenarios, to better understand the different aspects of the data for a given

[☆] This paper has been recommended for acceptance by Sven Dickinson.

* Corresponding author. Fax: +390458027068.

E-mail address: manuele.bicego@univr.it (M. Bicego).

¹ In different cases bioinformatics issues have led to novel pattern recognition methodological challenges – the most famous example being the biclustering problem [3].

² See <http://v-nome.org/about.html>

problem: actually, in recent years interpretability has become a stringent need in Pattern Recognition [6].

This paper makes another step in this direction, providing some further evidence on the effectiveness and interpretability of bioinformatics approaches for Pattern Recognition problems. In particular, in this paper, we propose and discuss a bioinformatics approach to 2D shape classification. Analysis of 2D shapes represents an important and vibrant research area (often paving the way for 3D object classification). Many approaches appeared in the literature (see for example the reviews [7,8]): very often, the 2D shape is encoded by the contour, which proved to be an effective and natural choice in many applications. Here we propose some methods to encode the shape contour as a biological sequence, employing tailored bioinformatics tools to perform classification. In the huge literature related to 2D shape analysis, many approaches exploit sequence alignments tools to perform shape matching ([9–13], just to cite a few) – some sequence matching-based approaches which start from shape-skeletons have also been proposed [14–16]. Focusing on our main target, i.e. to use *biological* sequence alignment tools, it should be noted that few approaches exist that employ techniques developed for biological sequences to perform shape classification or matching [17,18]. Nevertheless, these approaches propose a very different perspective with respect to our approach (and the video genome project), where the main goal is to encode the PR problem in biological terms, hence exploiting tools developed for biological sequence analysis. In other words, to exploit Bioinformatics tools for Pattern Recognition, one can consider two main steps: (i) encoding the PR problem in biological terms; (ii) applying bioinformatics tools to solve the problem. From this point of view, the approaches in [17,18] are rather poor, employing one particular technique for one particular purpose, and not considering a biological encoding which would allow the use of a wide class of algorithms for sequence analysis.

In this paper we do explicitly consider this aspect: first, we establish an analogy between 2D shapes and biological sequences, this motivating the employment of bioinformatics tools. Then we propose three ways for transforming a silhouette, encoded with the 8-directional chain code [19], into an aminoacidic sequence; given that, we can compute the similarity between shapes by using established biological sequence alignment tools. Such similarity is then exploited for classification in a K-nearest-neighbor setting. Finally, we show that other biological tools and concepts (such as multiple sequence alignment, conserved domains and locality and quality of alignment) can be used for a deeper analysis of the results. We performed different experiments with five standard shape datasets; on one hand, we show that classification results are very competitive with the state-of-the-art. On the other hand, we show that poor results we obtained on a retrieval case can be analysed in a deeper way by exploiting other biological sequence mining tools.

2. Background

This section briefly summarizes the bioinformatics tools exploited in our analysis. First, we present a preliminary overview of biological sequence alignment, so to clarify notations and terminology. Then, we present the tools employed for pairwise sequence alignment and multiple sequence alignment, trying to highlight specific aspects which are useful for our task.

2.1. Biological sequence alignment

Understanding and modelling the behavior of living cells is strongly dependent on the analysis of biological sequences, both nucleotide sequences – i.e. strings made with the 4 symbols of DNA, namely ATCG – and aminoacid sequences – i.e. strings with symbols

coming from a 20 letter alphabet. The most important basic operation is *sequence alignment*, which is a crucial step in many computational biology and bioinformatics analyses. The alignment of a pair of sequences aims at finding the best registration between them. This is done by taking into account the biological nature of the input sequences, so that biological (usually evolutionary) events, such as mutations and rearrangements, are clearly expressed [20].

From a practical point of view, alignment is obtained by inserting spaces inside the sequences (the so called gaps) so to maximise the point-wise similarity between them – a graphical example can be seen in Fig. 1. Such maximization relies on two important parameters. The first one is the so-called *substitution matrix* $B(i, j)$ which indicates the penalty to be paid for a mismatch between symbols i and j . This encodes the fact that in nature substitutions between aminoacids/nucleotides are not all equally likely. Different alternatives exist (such as the PAM [21] and the BLOSUM [22] matrices), each one exploiting biological a priori knowledge such as chemical properties of aminoacids. The second parameter is called the *gap penalty pair*, which is a pair of numbers specifying the cost of *inserting* and the cost of *extending* a gap in one of the sequences (in biology, inserting a new gap has a different impact with respect to extending an existing one).

2.2. Pairwise sequence alignment

The simplest instance of sequence alignment aims at finding the best registration between two sequences. In this case the approaches can be divided into global and local: global methods try to find an alignment between the entire strings, whereas local approaches aim at finding short regions of high similarity. Historically, the most famous pairwise sequence alignment algorithms are the Needleman-Wunsch [23] (which operates globally) and the Smith-Waterman [24] (which is local); both methods rely on dynamic programming to solve the problem efficiently. In particular, they both have a time complexity of $O(MN)$, with M and N being the lengths of the two sequences. We chose these two established tools, dating back to 70s/80s, in order to be as basic as possible; however large margins of improvements exist, since many advanced algorithms appeared in the last 30 years; one clear example is the popular BLAST (Basic Local Alignment Search Tool) [5], which implements a set of simple but effective heuristics to drastically reduce the time complexity of the alignment.

A by-product of the alignment process is the alignment similarity score: such quantity measures how “well aligned” the two sequences are. This score can be reasonably intended as a similarity measure between two sequences.

2.3. Multiple sequence alignment

When the goal of sequence analysis is to infer evolutionary events from a set of sequences, rather than reasoning in terms of pairwise alignments, the best option is to simultaneously align all the sequences, performing the so called multiple sequence alignment (MSA - [25]). In this context, the most widely used approach employs a heuristic search known as progressive technique, which builds up the final alignment by combining pairwise alignments – starting from the most similar pair and progressing to the most unrelated. In this scenario, the most famous tool employed by researchers is ClustalW³ [26].

Given a multiple alignment, different information can be inferred. For our scope, we will exploit two aspects:

1. The quality of the multiple alignment, which can be used to understand the local reliability of the sequence alignment (i.e. where

³ <http://www.ebi.ac.uk/Tools/msa/clustalw2/>

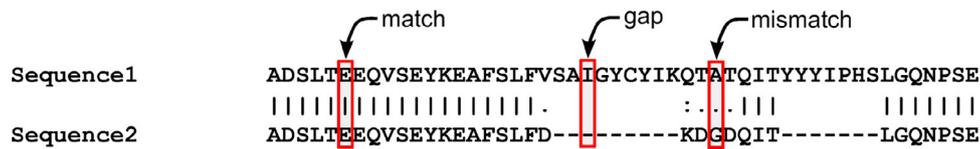


Fig. 1. Pairwise sequence alignment. A simple alignment is shown, highlighting the differences between matches, mismatches and gaps. As a convention, a vertical line connecting two symbols indicates a perfect match; a colon indicates a mismatch which is likely to occur in nature; finally, a dot corresponds to a very unlikely mismatch.

and how well the sequences are aligned). This can be assessed with publicly available tools such as the CORE web service⁴ [27].

2. The conserved domains, namely regions which are very conserved in the whole set of sequences: in biology, these regions are of crucial importance for the correct function of the molecules, since a mutation in such zone can have drastic effects. Different simple or advanced tools are available to discover such conserved domains, such as those implemented in the software ConFind⁵ [28]: in this case conserved domains are defined in terms of the maximum entropy allowed per position, the number of exceptions to the maximum entropy allowed and the minimum region length.

3. The proposed approach

In this section we present our approach: in particular, we first link 2D shapes and biological sequences, which may motivate the employment of bioinformatics tools in this context. Then we introduce the three methods used to encode shapes into biological sequences; finally, we detail how to transform alignments into a classification scheme.

3.1. 2D shapes and biological sequences

In this part we will describe some characteristics of biological sequences which can be present also in 2D silhouettes; such aspects are explicitly or implicitly taken into account by many biological sequence analysis tools, and can motivate and support the usage of such tools for the analysis of 2D shapes.

The first and most obvious similarity concerns the observation that 2D shapes can be effectively represented as sequences, similar to what is present in the biological case. From a general point of view, proteins represent the basic elements of every living cell, each one responsible for a different function of the cell. Proteins are 3D structures, which are directly mapped (through a projection) to a 2D discrete string (the aminoacid sequence). In the same way, a 3D object can be projected to the contour of one of its corresponding 2D aspects. In both cases there is a loss of information: it is difficult to recover the 3D object from the contour, as well as in biology it is difficult to recover the 3D structure of the protein from the sequence. Nevertheless, such projections are very useful: in both domains, similarities between 3D structures can be inferred by looking at similarities in the sequence domain.

In the 2D shape domain a classical contour representation exists, which shares many similarities with the biological representation: the chain code scheme [19]. This approach describes each point of the contour with a symbol taken from a 4- or 8-symbols alphabet, similarly to the biological case which uses a 4- or 20-letter alphabet. While being simple, such representation is very descriptive, allowing to encode all possible contours of all possible 3D objects – as the biological sequence can describe all possible proteins. Moreover, such representation is also useful to discriminate between shapes, as it has been shown in different successful cases [8,29].

Another aspect concerns occlusions, which can represent a problem when analyzing 2D shapes: part of the object is not visible or, worse, is covered by another object. This results in a contour which may have missing parts or “replaced” parts. This situation has a clear counterpart in biological sequences, and corresponds to sequence insertions, deletions or mutations. Many sequence analysis tools have to deal with this problem. Actually, local alignment tools (such as the classic Smith-Waterman [24]) rank alignments not on the basis of the whole sequence, but only using a set of highly similar small segments, which can prevent small occlusions from becoming problematic.

It is also important to note that there is a crucial difference between the two domains: 2D shapes tend to be closed, whereas biological sequences are not concerned with periodicity. This aspect is closely related to the choice of the starting point, which often represents an issue in the contour-based 2D shape classification. In all our experiments, a good starting point was reasonably determined by a simple pre-alignment of the shapes.

A final comment: as already mentioned in the introduction, it is worth considering that the main goal in biology/bioinformatics is to recover information which can help the understanding of the complex behavior of living organisms: therefore, a great importance has been always given to the interpretability of methods and solutions – with many visualization, inspection and interpretation tools available in the literature. These tools may be very useful also in Pattern Recognition, where interpretability has become a stringent need in recent years [6]. This aspect has been largely investigated in this paper: by linking 2D shapes and biological sequences we can then leverage a wide variety of tools, developed to address the great amount of different problems faced by Bioinformatics tools, where matching is only one of the possible tasks. In fact, more than showing that a very competitive shape classification tool can be devised, we provide evidence on possible exploitations of other bioinformatics tools, not strictly devoted to matching, usable to perform general mining and knowledge extraction from 2D shapes. Exploiting concepts such as high similarity segments, conserved domains and quality of multiple sequence alignments, we were able to show the effectiveness of our approach, providing a better understanding of some of the results. Remarkably, all these analyses have been carried out by using classical web tools that bioinformatics experts are using every day (ClustalW, Tcoffee, ConFind).

3.2. Encoding schemes

In the following we describe some methods to encode 2D shapes into biological sequences, in particular exploiting the 20-symbols alphabet of aminoacids.

Encoding scheme 1: Single. We start by describing every 2D shape with the chain code representation: such codes are used to represent a boundary by a connected sequence of straight-line segments with a specified length and direction [19]. Typically, this representation is based on 4- or 8- connectivity of the segments, as shown in the left part of Fig. 2: first we select an initial point, then we follow the contour in a clockwise order, and we select the code based on the direction taken towards the next pixel. Clearly, this descriptor is invariant to translation, but it is not invariant to rotations and scale

⁴ <http://www.igs.cnrs-mrs.fr/Tcoffee/tcoffee.cgi/index.cgi?stage1=1&daction=CORE::Regular>

⁵ <http://www.colorado.edu/chemistry/RGHP/software/>

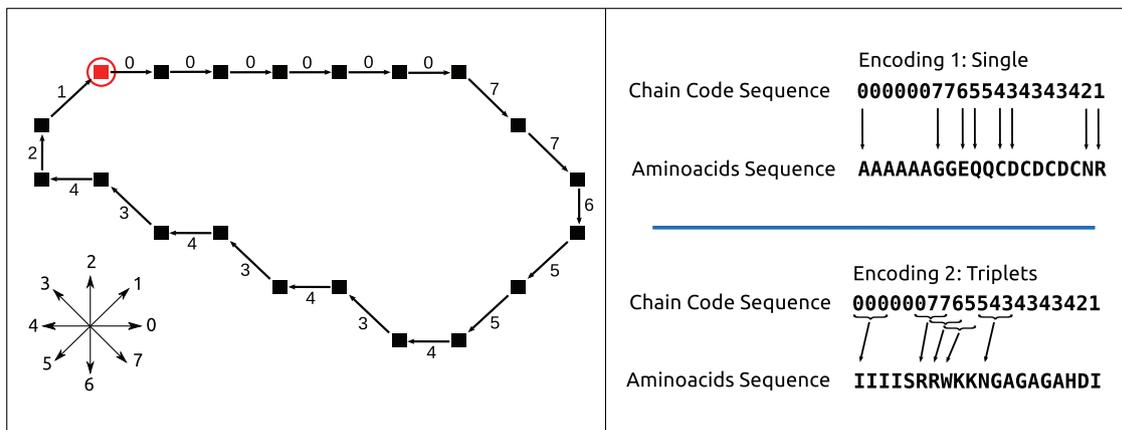


Fig. 2. Biological shape mappings: On the left, the general idea of the chain code. On the right, a graphical visualization for the mappings described in the text.

changes⁶. Even if we are aware that many more complex descriptors are available (such as Multi-scale contour fragments [11,30]), this simple approach allows an easy encoding of a shape into a biological sequence; it also facilitates the exploitation of visualization tools. Remarkably, even with this simple scheme, we already achieve satisfactory results. Since Chain Codes are not invariant to rotation, the initial point should be carefully selected. In particular, in all our experiments, we determine such initial point by first pre-aligning the shapes using a Procrustes analysis, and then selecting the most upper left point of the contour.

The first strategy to encode 2D shapes into biological sequences is rather simple, in order to analyse the basic potential of our approach. In particular, each chain code value is directly mapped onto an aminoacid. More in detail, the chain codes are mapped onto the first 8 aminoacids, as given in the IUPAC coding⁷: A, R, N, D, C, Q, E, and G. Even if this scheme is very simple, some good results have already been obtained – see our preliminary works [31,32].

Encoding scheme 2: Triplets. The second encoding scheme is based on two ideas, both aimed at exploiting the whole set of 20 aminoacids available in nature. Given the chain code representation, we consider the shape as composed by many short subsequences of length 3 – also called *k*-mers or *N*-grams in some papers. This idea of analysing short fragments of contiguous symbols – which is not new in the 2D shape classification scenario [29,30] – seems to be a reasonable choice also in biology: it has been recently shown that different information can be extracted, at the genomic level, from the analysis of the distribution of such words [33,34].

If we consider subsequences/fragments of length 3 (i.e. triplets), then we have 8^3 possible values (when using 8 directional chain-code), i.e. every sequence can be described using words coming from a dictionary of 8^3 entries; the main idea of the triplet encoding is to find a mapping between this dictionary and the aminoacidic alphabet; once given this mapping we can substitute every occurrence of a given fragment in a sequence with the corresponding aminoacid. To find this mapping, we adopt a way of reasoning similar to the one used by nature, where specialized molecules read a triplet of DNA (also called codon) and translate it into an aminoacid. This code, which starts from a low cardinality alphabet (A,T,C,G), is redundant: there are $4^3 = 64$ possible codons, which are mapped on 20 aminoacids by means of a lookup table. In our mapping we exploited directly this scheme of nature: we considered triplets of chain codes,

and we found a mapping between each of them and a codon (i.e. a triplet of DNA characters). Since we used a 8-directional chain code, there will be more triplets mapped into a single codon.

More in detail, the mapping is performed in two steps:

- Chain code triplets clustering:** in this phase all the chain code triplets⁸ are clustered into 61 groups, in order to allow, in the second step, to compute a one to one mapping with codons (we used 61 codons, not considering stop codons, which are particular triplets which are used in nature to depict the end of the sequence). The clustering is performed using an agglomerative hierarchical scheme, with the average link rule (as in the UPGMA algorithm, a widely used technique for phylogeny). The distance between two triplets of chain code $x = x_1x_2x_3$ and $y = y_1y_2y_3$ is defined as

$$d(x, y) = \sum_i \min [\text{mod}(x_i - y_i, \sim 8), \text{mod}(y_i - x_i, \sim 8)] \quad (1)$$
- Mapping codons to clusters of chain code triplets:** this second phase aims at finding a direct correspondence between every cluster (group of chain code triplets) and every codon. In this case we adopt two approaches: the former starts from the observation that aminoacids (and therefore codons) do not appear in nature with the same frequency; since the same holds for chain code triplets in a given dataset, a reasonable choice is to map triplets to codons by aligning the frequencies (we employed the codon frequencies of the Homo sapiens¹⁰), so that a triplet which appears often in a dataset is mapped to a DNA triplet widely present in nature – we call this scheme Triplets-Freq. The second method tries to align pairwise distances between codons and chaincode triplets. Specifically, we try to preserve the relation every codon/triplet has with all the other codons/triplets. To do that, we compute for every codon its averaged distance to all the other codons: this represents a measure of how “central” is the codon in the population. We repeated this operation for groups of chaincode triplets. Finally, the mapping is carried out by aligning these two measures of centrality: as before, the ordering of codon and triplet centralities returns the direct mapping – we call this

⁸ We discard from the clustering the triplets which are meaningless (such as those containing “40”, which indicates a “move to right” followed by a “move to left”) and those not present in the considered dataset – in average about 300 triplets were considered.

⁹ For example, a cluster is composed by 000, 001, 010, and 011.

¹⁰ Available at <http://www.kazusa.or.jp/codon/cgi-bin/showcodon.cgi?species=9606>.

⁶ A well known extension of the chain code, called differential chain code, is invariant to rotation; however, a preliminary set of experiments revealed a degradation in performances. This may be related to the loss of information due to the use of derivative operation.

⁷ <http://www.iupac.org/publications/pac/1984/pdf/5605x0595.pdf>

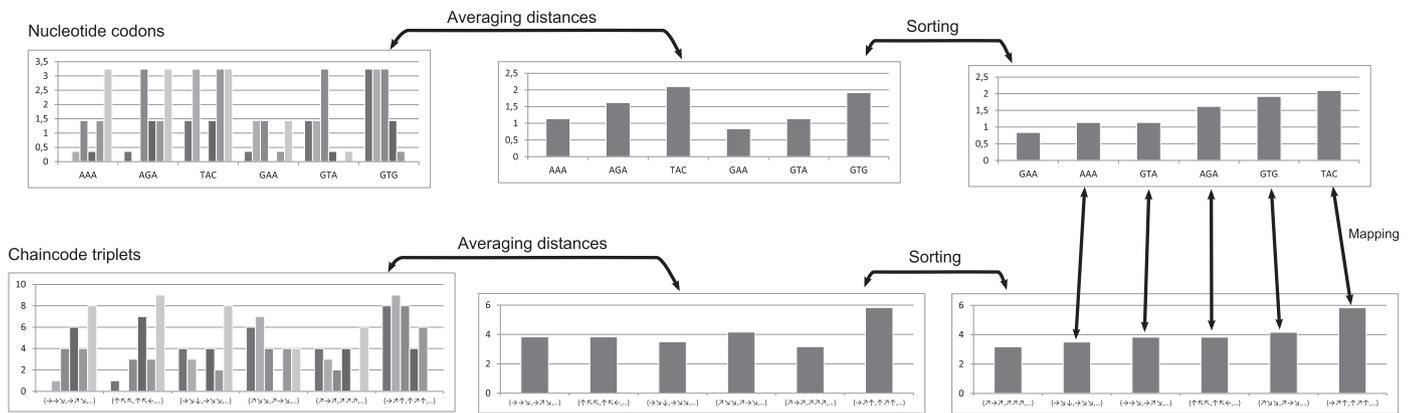


Fig. 3. From codons to chaincode triplets using the Triplets-Dist scheme. The left part of the figure shows, for 6 biological codons and 6 chaincode triplets, their distance (the height of each bin in the histogram) relative to each other. The average distance is then sorted, both for codons and for triplets of chain code, resulting in two rankings. Pairing the two rankings will give the final mapping.

scheme Triplets-Dist, summarized in Fig. 3. Distances between triplets of chain codes have been obtained as described before (i.e. as a result of the the average link scheme), whereas distances between codons have been computed with the Tajima–Nei function. This function, introduced in [35], computes the distance between two nucleotidic sequences by counting the mismatches which occur between them, each one weighted by the frequency of each symbol in the sequence. Even if this method was proposed several years ago, this distance is still largely applied when comparing DNA sequences.

Once we established the mapping between chain code triplets and codons, we use the standard biological lookup table for translating every codon into an aminoacid, hence obtaining the final encoding.

Summarizing, a given 2D shape is encoded into a biological sequence by scanning its chain code sequence and extracting triplets: we considered overlapped triplets, in order to remove the dependence from the reading frame (right part of Fig. 2). After extraction, every triplet is mapped to the aminoacid via one of the two mappings described above (Triplets-Freq or Triplets-Dist); as a result the sequence of aminoacids is obtained.

3.3. From sequence alignment to shape classification

Given the encoding, it is now straightforward to define a classification strategy based on standard K-nearest neighbor classifier [36]¹¹. Given an unknown object X and a distance, this classifier finds the K points in the training set which are nearest to X , assigning X to the most frequent class in that set. This is a natural choice because, given our framework, it is straightforward to define a distance between 2D shapes: after encoding the two shapes into aminoacid sequences, we can align them and use the alignment error as a measure of distance.

In our experiments we used both local and global alignment tools: in particular, we employed the two historical approaches described in previous section, namely the Needleman-Wunsch [23] and the Smith-Waterman [24] algorithms. We used, as substitution matrix, the classical family of BLOSUM matrices [38]¹².

¹¹ We are aware that, given a similarity matrix, interesting alternatives to K-NN exist (e.g. the dissimilarity-based representation paradigm [37]). However, KNN is accurate and simple enough to demonstrate the suitability of our proposed approach – in this paper we are more interested in showing the feasibility of our perspective, rather than reaching state of the art results. Moreover, this technique is very easy to interpret, since it gives an intuitive motivation of the assigned class label by showing the nearest neighbor to the user.

¹² Recently we investigated the possibility of employing substitution matrices which are learnt from shape datasets, see [39].

4. Classification results and discussion

In this section we evaluate the proposed framework in the context of shape classification. In particular, we first describe the datasets we used and the corresponding evaluation protocols; then we provide some details on the parameters of the proposed framework; finally we present and discuss our classification results, putting them in perspective with respect to the state of the art.

4.1. Datasets and classification protocol

The proposed framework has been analysed using five different datasets:

- The Vehicle Shape dataset (Vehicle)¹³ [40], which contains 120 vehicle shapes classified in 4 classes; following [40], classification accuracies have been determined using 10-fold cross validation.
- The Chicken Pieces dataset (Chicken)¹⁴ [41], containing 446 shapes of chicken pieces, divided in 5 classes; in this case we employed the Leave-One-Out (LOO) protocol, since most of the literature methods tested on this dataset in the past have been evaluated using this protocol.
- The MPEG-7 CE-Shape-1 dataset (MPEG-7)¹⁵ [42], representing a widely employed dataset used as reference in many 2D shape classification analysis; it contains 1400 shapes, divided in 70 classes. In this case we used both the Leave One Out and the Averaged Holdout Cross Validation (with 10 repetitions) – these being the two most common choices for classification¹⁶ [30].
- The ETH80 dataset (ETH80)¹⁷ [43], which contains 80 high-resolution color images of 3D objects from 8 categories: each object is represented by 41 images taken from different points of view, leading to a total of 3280 images. For this dataset we used the leave-one-object-out protocol, as detailed in the original paper [43].
- The Animal dataset (Animal)¹⁸ [44], which contains 2000 shapes describing 20 kinds of animals – each category has 100 animal images. The dataset is very challenging, containing articulation changes, part missing, large-scale intra-class variation, and noisy

¹³ http://visionlab.uta.edu/shape_data.htm.

¹⁴ <http://algoal.essex.ac.uk:8080/data/sequence/chicken/>.

¹⁵ <http://www.dabi.temple.edu/~shape/MPEG7/dataset.html>

¹⁶ On the contrary, the Bull's eye test [9] is typically employed to measure the retrieval performances.

¹⁷ <http://www.d2.mpi-inf.mpg.de/Datasets/ETH80>

¹⁸ <https://sites.google.com/site/xiangbai/animaldataset>

shapes. Following [30], we compute the classification accuracy with the averaged holdout protocol (with 10 repetitions).

Some examples of the various datasets are highlighted in Fig. 4. For all datasets, as in other approaches (e.g. [12]), 2D shapes have been pre-aligned using a classic Procrustes analysis. This allows to select a good starting point for the contour. Moreover, following [45], in the ETH80 and MPEG-7 case we sub-sampled the contour retaining only 200 points.

4.2. Experimental details

For all datasets we apply the proposed framework: we extract the shape chain codes and we encoded them using the 3 coding strategies described in the previous section (Single, Triplets-Freq and Triplets-Dist). We compute similarities between sequences using two alignment schemes: the Needleman–Wunsch (NW) and the Smith–Waterman (SW), as implemented in the MATLAB Bioinformatics toolbox (the alignment score has been normalized by the length of the two compared sequences). Finally, we perform the classification with the K-NN rule (with K automatically selected with Leave One Out cross-validation on the training set).

As explained in the Section 2, different parameters can be set within the bioinformatics tools; as a general rule, for most of them, we used the default values which are typically offered to the user – this represents the typical choice in biology when no extra information is available. However, for what concerns the two most important parameters of sequence alignment (the substitution matrix and the gap opening/extending penalty), we tested two possible configurations: in the first one, we used the *default configuration*: BLOSUM62 as substitution matrix and the pair (11,1) for gap penalties – where the former number represents the gap opening penalty, and the latter the gap extending penalty. In the second configuration, we adjusted the biological parameters in order to take into account the specific scenario (the shape classification context) – also in biology the tuning of parameters represents the best way to insert a priori knowledge into the process. In particular, in biology, the default gap penalty is rather high, since in this context it is not usually desirable to break a biological sequence. In the shape context, however, such a strong constraint does not hold: actually, gaps can significantly help in dealing with occlusions and scale changes. Therefore we changed the gap penalty from (11,1) to (6,2) – we restricted our choices to standard gap costs implemented within the BLAST algorithm. Furthermore, we used a substitution matrix which highly penalizes changes in the sequences (namely, the algorithm is forced to try to align the sequences in the best possible way). The idea here is that whereas there are cases in biology where it is preferable to allow a high degree of mutation (i.e. tolerating mismatches), in the 2D shapes context an exact matching may be preferable. In particular we used a BLOSUM90 matrix (the higher the number after the word “BLOSUM” the more “conservative” the substitution matrix) – we called this *Advanced configuration*.

4.3. Results and discussion

All results, for all datasets, all coding strategies, all alignment algorithms and parameters are presented in Table 1.

In order to get an idea of the statistical significance of the results, we computed for all experiments the standard errors of the mean, a common way of estimating the variance of the accuracy [46], computed by dividing the averaged accuracy by the squared root of the number of objects. In all experiments concerning the five datasets such errors were all less than 0.0077, 0.0008, 0.0001, 0.0023, 0.0004 for the Vehicle, the Chicken, the MPEG, the ETH80, and the Animal, respectively.

From the table it is evident that the proposed framework performs reasonably well, being also very competitive with respect to the state of the art – reported in Tables 2, 3 and 4.

Table 1

Results for different datasets, coding strategies, alignment algorithms and parameter configurations.

Coding	Algo	Vehicle	Chicken	MPEG (LOO)	MPEG (HO)	ETH80	Animal
<i>(Default configuration)</i>							
Single	NW	0.883	0.915	0.976	0.954	0.907	0.817
Single	SW	0.904	0.904	0.974	0.949	0.903	0.802
Triplets-Freq	NW	0.908	0.872	0.969	0.933	0.861	0.728
Triplets-Freq	SW	0.906	0.845	0.966	0.929	0.858	0.697
Triplets-Dist	NW	0.935	0.881	0.966	0.932	0.870	0.687
Triplets-Dist	SW	0.922	0.886	0.959	0.925	0.859	0.654
<i>(Advanced configuration)</i>							
Single	NW	0.926	0.917	0.981	0.961	0.915	0.837
Single	SW	0.924	0.910	0.979	0.957	0.914	0.823
Triplets-Freq	NW	0.929	0.890	0.975	0.947	0.876	0.772
Triplets-Freq	SW	0.932	0.863	0.971	0.940	0.867	0.731
Triplets-Dist	NW	0.929	0.881	0.976	0.947	0.886	0.736
Triplets-Dist	SW	0.930	0.888	0.969	0.940	0.880	0.702

Table 2

Comparison with the state of the art: chicken and vehicle.

Methodology	Accuracy
<i>(Chicken pieces)</i>	
1-NN + Levenshtein edit dist [47]	≈0.67
1-NN + HMM-based distance [48]	0.738
1-NN + mBm-based features [48]	0.765
SVM + HMM + Marg. kernel [49]	0.775
1-NN + approx. cyclic dist [47]	≈0.78
SVM + HMM + Top Kernel [50]	0.808
SVM + cyclic string edit [51]	0.811
SVM + HMM + Trans embedding [52]	0.811
SVM + HMM-based entropies [53]	0.812
SVM + HMM + Fisher Kernel [50]	0.817
SVM + HMM + FESS-embedding [50]	0.830
Contour Fragments [54]	0.845
SVM + HMM + CB Fisher kernel [55]	0.858
SVM + HMM + NL Marg. kernel [49]	0.855
Lattice-Computing (LC) [56]	0.865
SVM + Kernel Edit Distance [57]	0.871
SVM + IT kernels on n-grams [29]	0.895
Linearity Closed Curves [58]	0.933
Our best	0.917
<i>(Vehicle – all from [40])</i>	
SVM + curvature	0.625
Ergodic HMM + Max Lik.	0.625
Left Right HMM + Max Lik.	0.708
Circular HMM + Max Lik.	0.733
SVM + Zernike moments	0.792
SVM + Fourier descriptors	0.825
HMM + Weighted likelihood	0.842
Our best	0.935

Going more into the details, it is interesting to note that for the chicken, the MPEG, the ETH80 and the Animal datasets the Simple encoding represents the best choice, whereas for vehicle a substantial improvement is obtained when using the two triplets codings (especially with the default configuration of the alignment parameters). At the same time, in the vehicle dataset the Smith–Waterman algorithm provides better results, whereas for the remaining four the Needleman–Wunsch gives superior performances. This may be due to the low between class variability in the vehicle dataset, hence it is more convenient to employ a local algorithm such as SW, which can better align local parts, namely details – which are also better captured by the richer Triplets coding. On the contrary, in the other datasets, the differences between the classes is more pronounced (in such datasets the challenge is represented by the intra-class variability), therefore a global alignment method can be more appropriate. In any case, in all configurations a proper tailoring of the alignment parameters led to an improvement of the performances.

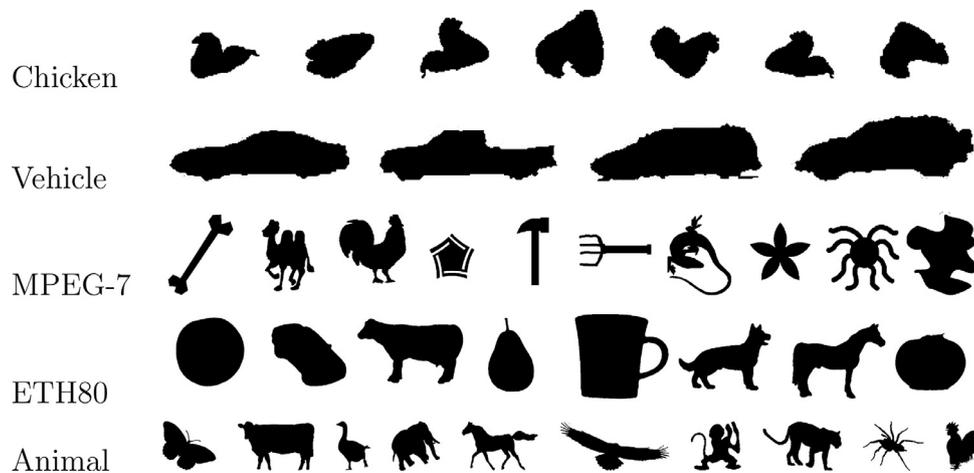


Fig. 4. Examples from the datasets used.

Table 3

Comparison with the state of the art: MPEG and ETH80.

Methodology	HO	LOO
	<i>(MPEG-7)</i>	
Procrustes 1-NN [59]	–	0.916
Tangent kernel SVM [59]	–	0.936
Graph inner distance [60]	–	0.947
Graph transduction distance [60]	–	0.957
Manifold kernel SVM [59]	–	0.966
Polygonal multi-resolution [11]	–	0.976
String of symbols [61]	–	0.974
Chance probabilities [62]	–	0.974
String kernels [63]	–	0.978
Robust symbolic representation [12]	–	0.986
Kernel edit distance [57]	–	0.989
Skeleton paths [44]	0.867	–
Contour segments [44]	0.911	–
ICS [44]	0.966	–
Class segment set [64]	0.909	0.979
BCF [30]	0.972	0.989
Our best	0.961	0.981

Table 4

Comparison with the state of the art: Animal. Table is taken from [30] and [69].

Methodology	Accuracy	
	<i>(ETH80)</i>	
Color histogram [43]	0.649	
PCA gray [43]	0.830	
PCA masks [43]	0.834	
SC + DP [43]	0.864	
IDSC + DP [9]	0.880	
IDSC + Morphological strategy [65]	0.880	
Height functions [10]	0.887	
Robust symbolic representation [12]	0.903	
Kernel edit distance [57]	0.913	
BCF [30]	0.915	
Our best	0.915	
	<i>(Animal)</i>	
Class segment set [64]	0.697	
IDSC [9]	0.736	
Bag of SIFT [66]	0.749	
Contour segments [44]	0.717	
Skeleton paths [44]	0.679	
ICS [44]	0.784	
CS&SP&IDSC-F [44]	0.787	
CS&SP-DP [67]	0.807	
Shape tree [68]	0.800	
HOG-SIFT BoW [66]	0.804	
BCF [30]	0.834	
Contextual BOW [69]	0.860	
Our best	0.837	

5. Deeper analysis

In this part we provide an example of how it is possible to exploit the huge amount of bioinformatics tools to have a deeper understanding of the results. To do that, we evaluated our framework in a slightly different task (the retrieval task), trying to exploit bioinformatics tools and concepts to better understand results that were not satisfactory. Even if related, the retrieval task is slightly different from classification: given a testing object, the goal is to retrieve as many shapes as possible from the same class. Clearly, given our framework, a retrieval system can be easily defined: given an unknown shape encoded in an aminoacid sequence, its alignment with respect to all sequences of the dataset can induce a ranking which allows to extract the most similar shapes.

The MPEG-7 dataset has been widely applied in the retrieval task, where the performances are compared using the well established protocol called bull's eye test [9]: for every query shape, we count the number of objects that belong to the same class in the top 40 matches; the retrieval rate is then measured as the ratio of the total number of shapes from the same class (which at most is 20) to the highest possible number (which is $20 \cdot 1400 = 28,000$). A large amount of results appeared in the literature on this task, with some very advanced techniques reaching impressive retrieval rates – for example, an almost perfect retrieval accuracy was given in [70], where many different distance measures are fused using co-transduction.

Using our proposed framework, the best result we achieved was a bulls eye score of 77.24%: in this case, our algorithm does not represent a valuable competitor of other specific algorithms, probably because it is too general to face the larger complexity of the retrieval task with respect to the classification task. Let us go deeper on this aspect: within our framework, which exploits KNN for classification, the starting point is the same for both classification and retrieval, namely the similarity of the query (testing shape) with all objects in the dataset. Roughly speaking, it can be said that here good classification performances can be obtained if the query shape has a very good match with *some* of the correct training shapes (i.e. that belong to the same class) – in the extreme case of Nearest Neighbor, it is enough that the query has the highest match with only one correct shape, no matter how low are the matching scores with all the remaining correct shapes. On the contrary, in order to have a good retrieval rate, the query shape should have a rather good match with *all* the shapes of the same class – in MPEG a perfect retrieval is obtained when all the correct shapes are retrieved among the first 40 hits (over 1400). Therefore, in our case, the retrieval task is definitely more difficult than the classification task: within the proposed framework, it is

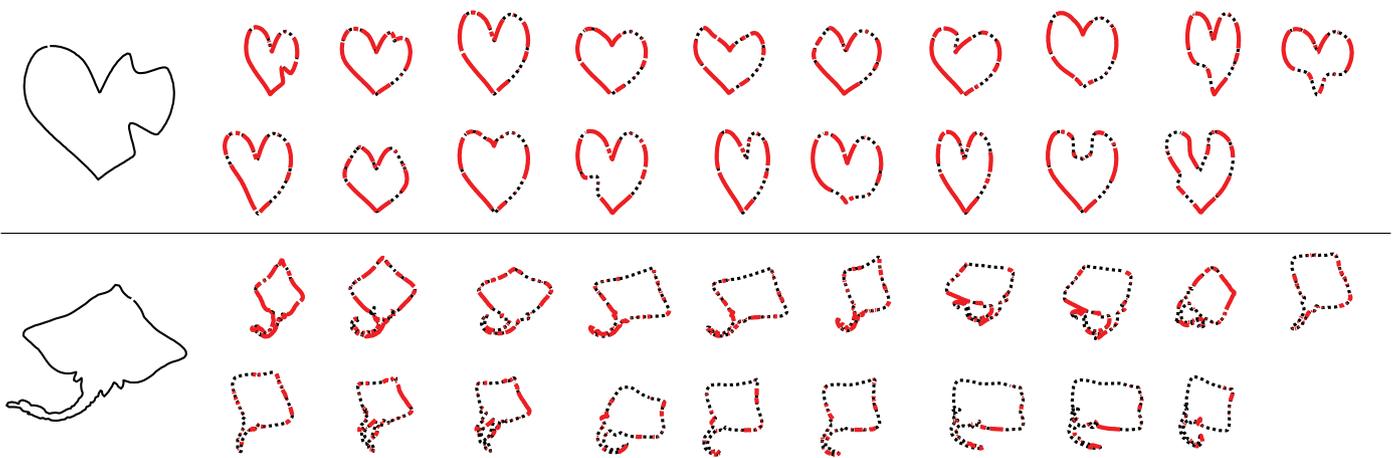


Fig. 5. Visualization of high similarity segments for two query shapes: in each part, for every shape of the training set (right), we show with a continuous bold line the high similarity segments after the pairwise alignment with the query shape (left).

Class 60

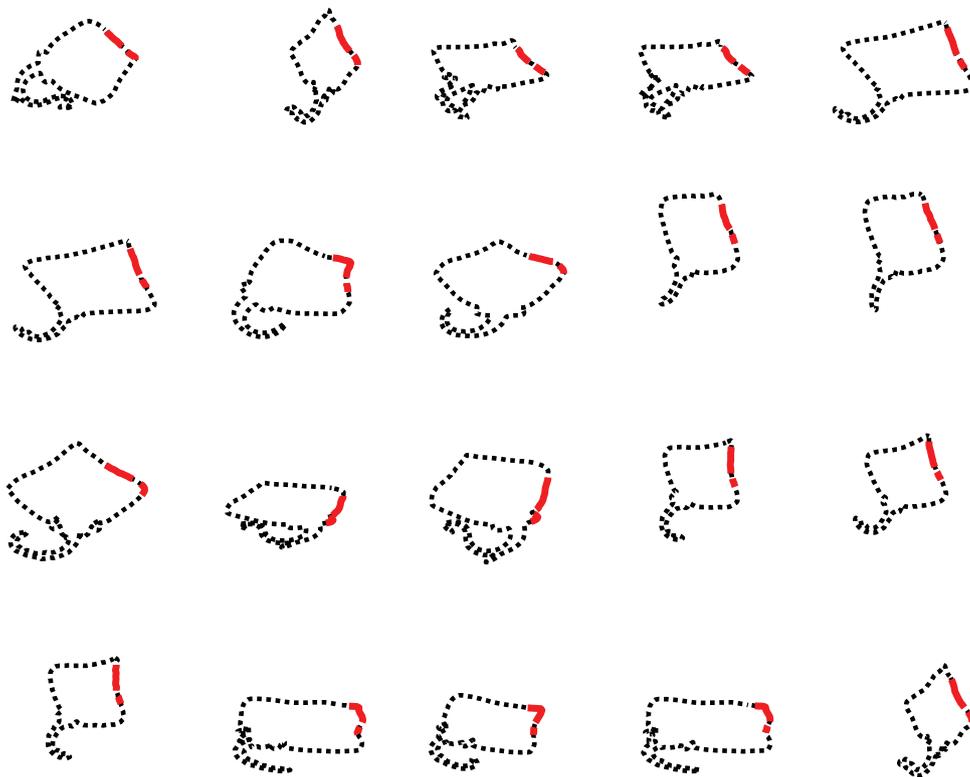


Fig. 6. The class of rays from the MPEG-7 dataset with the conserved domains highlighted with a continuous bold line.

evident that it is possible to derive good matches of the query shape with *many* correct shapes, but not with *all*.

In the last part of this section we will provide some more complex versions of the framework which are able to partially alleviate this problem. Before that, however, we would like to show an example of how it is possible to exploit some other standard bioinformatics tools to qualitatively investigate the intuitions sketched before. In particular, we started by visually analysing the alignment results, in particular inspecting and visualizing the regions of high similarity derived from the application of the local Smith Waterman algorithm. Visual analysis of alignments represents a standard tool employed by bioinformaticians, which allows to recover from align-

ment errors as well as to detect interesting parts of the paired sequences. In our case, we visualized classes where classification and retrieval were both satisfactory, as well as classes where classification was appropriate but retrieval failed. As an example, we report in Fig. 5 two query shapes, together with the corresponding high similarity regions obtained after aligning them with all the other shapes of the same class. The first example refers to a class where both classification and retrieval are good. It shows that the regions of high similarities are (i) long and, more importantly, (ii) mainly involved with the same part of the shape. In contrast, in the second example, only the first matches are with long segments; moreover, the matched parts are different – compare for example the first match with the fourth.

to deeper inspect the unsatisfactory results obtained in the retrieval case.

Acknowledgments

Authors would like to thank Nebojsa Jojic and Alessandro Farinelli for helpful discussions and suggestions. Authors are also grateful to the anonymous reviewers for their precious comments.

References

- [1] P. Baldi, S. Brunak, *Bioinformatics: The Mmachine Learning Approach*, MIT Press, 2001.
- [2] S. Eddy, Profile hidden markov models, *Bioinformatics* 14 (9) (1998) 755–763.
- [3] S. Madeira, A. Oliveira, Biclustering algorithms for biological data analysis: a survey, *IEEE/ACM Trans. Comput. Biol. Bioinf.* 1 (1) (2004) 24–45.
- [4] A. Bronstein, M. Bronstein, R. Kimmel, The video genome, *CoRR abs/1003.5320*.
- [5] S. Altschul, W. Gish, W. Miller, E. Myers, D. Lipman, Basic local alignment search tool, *J. Mol. Biol.* 215 (3) (1990) 403–410.
- [6] J. Chang, J. Boyd-Graber, S. Gerrish, C. Wang, D. Blei, Reading tea leaves: How humans interpret topic models, in: *NIPS*, 2009, pp. 288–296.
- [7] S. Loncaric, A survey of shape analysis techniques, *Pattern Recogn.* 31 (1998) 983–1001.
- [8] D. Zhang, G. Lu, Review of shape representation and description techniques, *Pattern Recogn.* 37 (1) (2004) 1–19.
- [9] H. Ling, D. Jacobs, Shape classification using the inner-distance, *IEEE Trans. Pattern Anal. Mach. Intell.* 29 (2007) 286–299.
- [10] J. Wang, X. Bai, X. You, W. Liu, L. Latecki, Shape matching and classification using height functions, *Pattern Recogn. Lett.* 33 (2011) 134–143.
- [11] E. Attala, P. Siy, Robust shape similarity retrieval based on contour segmentation polygonal multiresolution and elastic matching, *Pattern Recogn.* 38 (2005) 2229–2241.
- [12] M. Daliri, V. Torre, Robust symbolic representation for shape recognition and retrieval, *Pattern Recogn.* 41 (5) (2008) 1799–1815.
- [13] P. Felzenszwalb, J. Schwartz, Hierarchical matching of deformable shapes, in: *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.
- [14] A. Torsello, A. Robles-Kelly, E. Hancock, Discovering shape classes using tree edit-distance and pairwise clustering, *Int. J. Comput. Vis.* 72 (3) (2007) 259–285.
- [15] A. Torsello, E. Hancock, A skeletal measure of 2d shape similarity, *Comput. Vis. Image Underst.* 95 (1) (2004) 1–29.
- [16] X. Bai, L. Latecki, Path similarity skeleton graph matching, *IEEE Trans. Pattern Anal. Mach. Intell.* 30 (7) (2008) 1282–1292.
- [17] L. Chen, R. Feris, M. Turk, Efficient partial shape matching using smith-waterman algorithm, in: *CVPR workshop on Non-Rigid Shape Analysis and Deformable Image Alignment*, 2008, pp. 1–6.
- [18] R. Huang, V. Pavlovic, D. Metaxas, A profile hidden markov model framework for modeling and analysis of shape, in: *Proceedings of the International Conference on Image Processing*, 2006, pp. 2121–2124.
- [19] R. Gonzalez, R. Woods, *Digital Image Processing*, 2nd edition, Prentice Hall, 2002.
- [20] R. Durbin, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, Cambridge University Press, 1998.
- [21] M. Dayhoff, R. Schwartz, B. Orcutt, A model of evolutionary change in proteins, in: *Atlas of Protein Sequence and Structure*, National Biomedical Research Foundation, 1978, pp. 345–352.
- [22] S. Henikoff, J. Henikoff, Amino acid substitution matrices from protein blocks, *Proc. Natl. Acad. Sci.* 89 (1992) 10915–10919.
- [23] S. Needleman, C. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins, *J. Mol. Biol.* 48 (1970) 443–453.
- [24] T. Smith, M.M.S. Waterman, Identification of common molecular subsequences, *J. Mol. Biol.* 147 (1) (1981) 195–197.
- [25] J. Pevsner, *Bioinformatics and Functional Genomics*, Wiley, 2003.
- [26] M. Larkin, Clustal w and clustal x version 2.0, *Bioinformatics* 23 (21) (2007) 2947–2948.
- [27] O. Poirat, K. Suhre, C. Abergel, E. O’Toole, C. Notredame, 3dcoffee@igs: a web server for combining sequences and structures into a multiple sequence alignment, *Nucl. Acids Res.* 32 (2004) 37–40.
- [28] J. Smagala, E. Dawson, M. Mehlmann, M. Townsend, R. Kuchta, K. Rowlen, Confind: a robust tool for conserved sequence identification, *Bioinformatics* 21 (24) (2005) 4420–4422.
- [29] M. Bicego, A. Martins, V. Murino, P. Aguiar, M. Figueiredo, 2d shape recognition using information theoretic kernels, in: *Proceedings of the International Conference on Pattern Recognition*, 2010, pp. 25–28.
- [30] X. Wang, B. Feng, X. Bai, W. Liu, L. Latecki, Bag of contour fragments for robust shape classification, *Pattern Recogn.* 47 (6) (2014) 2116–2125.
- [31] P. Lovato, M. Bicego, 2d shapes classification using blast, in: *Proceedings of the International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, 2012, pp. 273–281.
- [32] M. Bicego, P. Lovato, 2d shape recognition using biological sequence alignment tools, in: *Proceedings of the International Conference on Pattern Recognition*, 2012, pp. 1359–1362.
- [33] B. Chor, D. Horn, N. Goldman, Y. Levy, T. Massingham, Genomic dna k-mer spectra: models and modalities, in: *Proceedings of the International Conference on Research in Computational Molecular Biology*, 2010 571–571.
- [34] A. Castellini, G. Franco, V. Manca, A dictionary based informational genome analysis, *BMC Genom.* 13 (1) (2012) 485.
- [35] F. Tajima, M. Nei, Estimation of evolutionary distance between nucleotide sequences, *Mol. Biol. Evol.* 1 (3) (1984) 269–285.
- [36] R. Duda, P. Hart, D. Stork, *Pattern Classification*, 2nd edition, Wiley Interscience, 2001.
- [37] E. Pekalska, R.P. Duin, The dissimilarity representation for Pattern Recognition - Foundations and Applications, World Scientific, 2005.
- [38] S. Henikoff, J. Henikoff, Amino acid substitution matrices from protein blocks, *Proc. Natl. Acad. Sci.* 89 (22) (1992) 10915–10919.
- [39] P. Lovato, A. Milanese, C. Centomo, A. Giorgetti, M. Bicego, S-blosum: classification of 2d shapes with biological sequence alignment, in: *Proceedings of the International Conference on Pattern Recognition (ICPR2014)*, 2014, pp. 2335–2340.
- [40] N. Thakoor, J. Gao, S. Jung, Hidden markov model-based weighted likelihood discriminant for 2-d shape classification, *IEEE Trans. Image Process.* 16 (11) (2007) 2707–2719.
- [41] G. Andreu, A. Crespo, J.M. Valiente, Selecting the toroidal self-organizing feature maps (tsomfm) best organized to object recognition, in: *Proceedings of the International Symposium on Neural Networks*, Vol. 2, 1997, pp. 1341–1346.
- [42] L. Latecki, R. Lakamper, U. Eckhardt, Shape descriptors for non-rigid shapes with a single closed contour, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2000, pp. 424–429.
- [43] B. Leibe, B. Schiele, Analyzing appearance and contour based methods for object categorization, in: *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2003, p. 409.
- [44] X. Bai, W. Liu, Z. Tu, Integrating contour and skeleton for shape classification, in: *Proceedings of the International Conference Computer Vision Workshops*, 2009, pp. 360–367.
- [45] X. Shu, X.J. Wu, A novel contour descriptor for 2d shape matching and its application to image retrieval, *Image Vis. Comput.* 29 (4) (2011) 286–294.
- [46] K. Brodersen, C. Ong, K. Stephan, J. Buhmann, The balanced accuracy and its posterior distribution, in: *Proceedings of the International Conference on Pattern Recognition*, 2010, pp. 3121–3124.
- [47] R. Mollineda, E. Vidal, F. Casacuberta, Cyclic sequence alignments: approximate versus optimal techniques, in: *IX Spanish Symp. Pattern Recogn. Image Anal.*, Vol. 6, 2001, pp. 97–102.
- [48] M. Bicego, A. Trudda, 2d shape classification using multifractional brownian motion, in: *Proceedings of the SSPR & SPR ’08*, 2008, pp. 906–916.
- [49] A. Carli, M. Bicego, S. Baldo, V. Murino, Non-linear generative embeddings for kernels on latent variable models, in: *Proceedings of the ICCV09 Workshop on Subspace Methods*, 2009, pp. 154–161.
- [50] A. Perina, M. Cristani, U. Castellani, V. Murino, N. Jojic, A hybrid generative/discriminative classification framework based on free-energy terms, in: *Proceedings of the International Conference on Computer Vision*, 2009, pp. 2058–2065.
- [51] M. Neuhaus, H. Bunke, Edit distance-based kernel functions for structural pattern classification, *Pattern Recogn.* 39 (10) (2006) 1852–1863.
- [52] M. Bicego, E. Pekalska, D. Tax, R. Duin, Component-based discriminative classification for hidden markov models, *Pattern Recogn.* 42 (11) (2009) 2637–2648.
- [53] A. Perina, M. Cristani, U. Castellani, V. Murino, A new generative feature set based on entropy distance for discriminative classification, in: *Proceedings of the International Conference on Image Analysis and Processing*, 2009, pp. 199–208.
- [54] M. Daliri, V. Torre, Classification of silhouettes using contour fragments, *Comput. Vis. Image Underst.* 113 (2009) 1017–1025.
- [55] M. Bicego, M. Cristani, V. Murino, E. Pekalska, R. Duin, Clustering-based construction of hidden Markov models for generative kernels, in: *Proceedings of the International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2009, pp. 466–479.
- [56] V. Kaburlasos, S. Papadakis, A. Amanatiadis, Binary image 2d shape learning and recognition based on lattice-computing (lc) techniques, *J. Math. Imaging Vis.* 42 (2–3) (2012) 118–133.
- [57] M. Daliri, V. Torre, Shape recognition based on kernel-edit distance, *Comput. Vis. Image Underst.* 114 (10) (2010) 1097–1103.
- [58] P. Rosin, J. Pantovic, J. Zunic, Measuring linearity of closed curves and connected compound curves, in: *Proceedings of the Asian Conference on Computer Vision*, 2012, pp. 310–321.
- [59] S. Jayasumana, M. Salzmann, H. Li, M. Harandi, A framework for shape analysis via hilbert space embedding, in: *Proceedings of the International Conference on Computer Vision (ICCV)*, 2013, pp. 1249–1256.
- [60] X. Bai, X.Y.L. Latecki, W. Liu, Z. Tu, Learning context-sensitive shape similarity by graph transduction, *IEEE Trans. Pattern Anal. Mach. Intell.* 32 (5) (2010) 861–874.
- [61] M. Daliri, V. Torre, Shape recognition and retrieval using string of symbols, in: *Proc. Int. Conf. Mach. Learn. Appl.*, 2006, pp. 101–108.
- [62] B. Super, Retrieval from shape databases using chance probability functions and fixed correspondence, *Int. J. Pattern Recogn. Artif. Intell.* 20 (8) (2006) 1117–1138.
- [63] M. Daliri, E. Delponte, A. Verri, V. Torre, Shape categorization using string kernels, in: *Proceedings of the SSPR/SPR*, 2006, pp. 297–305.
- [64] K. Sun, B. Super, Classification of contour shapes using class segment sets, in: *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2005, pp. 727–733.

- [65] R. Hu, W. Jia, Y. Zhao, J. Gui, Perceptually motivated morphological strategies for shape retrieval, *Pattern Recogn.* (2012) 105–111.
- [66] K.L. Lim, H. Galoogahi, Shape classification using local and global features, in: *Proceedings of Pacific-Rim Symposium on Image and Video Technology*, 2010, pp. 115–119.
- [67] C. Li, A.B.H.X. You, W. Zeng, L. Zhou, Distinctive parts for shape classification, in: *Proceedings of the International Conference on Wavelet Analysis and Pattern Recognition*, 2011, pp. 97–102.
- [68] Y. Li, J. Zhu, F. Li, A hierarchical shape tree for shape classification, in: *Proceedings of the International Conference of Image and Vision Computing New Zealand*, 2010, pp. 1–6.
- [69] R. Bharath, C. Xiang, T. Lee, Shape classification using invariant features and contextual information in the bag-of-words model, *Pattern Recogn.* 48 (3) (2015) 894–906.
- [70] X. Bai, B. Wang, C. Yao, W. Liu, Z. Tu, Co-transduction for shape retrieval, *IEEE Trans. Image Process.* 21 (5) (2012) 2747–2757.