Language engineering and information theoretic methods in protein sequence similarity studies

A. Bogan-Marta[1,3], A. Hategan[2], I. Pitas[3], and I. Tabus[2]

[1] University of Oradea, Department of Computers, Universtatii No1, 410087, Oradea, Romania, alinab@uoradea.ro
[2] Tampere University of Technology, Institute of Signal Processing, Korkeakoulunkatu 1, P.O. Box 553, FIN-33101, {andrea.hategan,ioan.tabus}@tut.fi
[3] Aristotle University of Thessaloniki, Department of Informatics, Artificial Intelligence and Information Analysis Laboratory, Box 451,Thessaloniki, Greece, pitas@aiia.csd.auth.gr

Prejudice is encountered almost everywhere in everyday life; it's in human nature. When someone new appears in a community some of the first questions the members of that community ask are: "Where does she/he come from? To what family does she/he belong?", hopping to get a clue about the character of the new person. Even if families are never uniform, in absence of any other information, some guesses can be made about the new person, but any inference should be taken with care. This way of thinking is often extrapolated to objects, substances, etc.

Proteins are essential molecules to sustain life in all organisms. A normal development and function of an organism depends on the normal function of all proteins in the big chemical machinery of reactions. Many diseases appear due to abnormal function of some proteins.

The first question to be ask about a newly discovered protein is "What is the function of this new protein?". The most reliable way to infer the function of a newly discovered protein is by wet-lab techniques, but these methods are money and time consuming and are also subject to human errors. Because it's more easier to get the primary structure (amino acids sequence) of a new protein it is desirable to have a method to infer the function based on its amino acid sequence.

One way is to find the family to which the new protein belongs and based on the already annotated proteins in that family, predict the function. Gen-

[0] The content of this chapter is mainly based on experiments published in conference proceedings freely available. For reproduction of some fragments relevant for methods description we kindly demand authors and publishers understanding.

erally, a protein family contains homologous proteins , i.e. proteins that have a common ancestor. To infer that two proteins are homologous we need a method to quantify the similarity of the two proteins based on their amino acid sequences . If the two sequences share a certain degree of similarity and it's proven that the similarity is significant (i.e. it's hard to obtain by chance such a degree of similarity), it might happen that the two sequences are homologous and then their alignment become a powerful tool for evolutionary and functional inference. Protein structure is much better conserved during evolution than protein sequence . Even if several proteins have low sequence similarity, but still adopt similar structures, contain identical or related amino acid residues in their active sites and have similar catalytic mechanism, they have sufficient evidence for homology.

Classical methods for measuring the similarity of two proteins use a scoring matrix and an alignment algorithm to align the sequences. The proteins similarity is quantified by the alignment score . Despite the maturity of the developed methodologies for genomic sequence similarity detection and alignment, the derivation of new similarity measures is still an active research area. The interest is actually renewed, due to the continuous growth in size of the widely available proteomic databases that calls for alternative cost-efficient algorithmic procedures, which can reliably quantify protein similarity without resorting to any kind of alignment. Apart from efficiency, a second specification of equal importance for the establishment of similarity measures is the avoidance of parameters that need to be set by the user (a characteristic inherent in the majority of the well known methodologies). It is often the case with the classical similarity approaches that the user faces a lot of difficulties in the choice of a suitable search algorithm, scoring matrix or function as well as a set of optional parameters, whose optimum values correspond to the most reliable similarity.

Because DNA, RNA and proteomic sequences can be represented in text format, language engineering and information theoretic methods have recently started to be used in protein similarity studies. Within this book chapter, we underline the potential of language engineering techniques and those involving information theoretic principles in analyzing protein sequences from similarity perspective. First, we formulate a framework of the different approaches identified, presenting a survey of the state of the art in the subject. Secondly, the attention is focused on other two methods we experimented that prove to be promising strategies in comparing sequences of proteins. In the sequel, we describe the main aspects involved in these two new methods proposed for protein similarity analysis underlining their advantages over the other classical well known methodologies. Even if the two new methods are using some common concepts from information theory field they are independent, proving how interesting and efficient the fundamental theory merges in revealing solutions for biological needs.

1 A survey of language engineering and information theoretic methods for protein sequences similarity

The fact that protein sequences from all different organisms can be treated as texts written in a universal language where the alphabet consists of 20 distinct symbols, the amino acids , opens the perspective of many techniques from language and text processing domains to be accessed. The mapping of a protein sequence to its structure, functional dynamics and biological role then becomes analogous to the mapping of words to their semantic meaning in natural languages. This analogy is exploited in many applications that use statistical language modeling and text processing techniques for the advancement of biological sequences understanding. Also, information content stored in biological sequences open the door for information theoretic methods to bring their contribution to efficient exploration of biological field. They provide measures and methods for evaluation and quantification of biological sequence information useful to a large diversity of investigations. In this section we are presenting some interesting techniques identified in research works that are proving benefits for protein sequence similarity detection based on linguistic approaches and information theoretical measures.

1.1 NLP for the extraction of protein relationships

Extracting protein interaction relationships from textual repositories, prove to be useful in generating novel biological hypotheses. Scientists often use textual databases to ascertain further information about specific biological entities such as proteins.

Using a natural-language processing tool, in [1] is realized a rule-based analysis to retrieve textual data in order to find similar proteins , with the similarity expressing the notion of common functional attributes . With the relevant abstracts to two known functionally related proteins, and a modified existing natural language processing tool able to extract protein interaction terms, were obtained functional information about Amyloid Precursor Protein(APP) and Prion Protein(PrP). Both of them have been implicated in the etiology of Alzheimer's disease and Creutzfeldt-Jakob disease, respectively. The program (called arbiter_pi) was developed in order to recover functional! relationships from a selected set of biomedical titles and abstracts. In this work, the attention is focused on three specific relation types: INDUCE, INHIBIT, and REGULATE. An important step in the application is the recognition of the verbs and nominalizations that cue relationships. Such syntactic predicates are determined from the development dataset based on the semantic characteristics of the target predications within the individual sentences. Example of verbs involved in relation types identification observed in training set are:

INDUCES - induce, activate, stimulate, cause, increase;

INHIBIT - inhibit, attenuate, block, damage, disrupt, impair;
REGULATE - regulate, participate, modulate, mediate.

The practical application relies on input noun phrases being mapped [2] to concepts in Unified Medical Language System Metathesaurus [3]. Concepts in the Metathesaurus are assigned one or more semantic types, which provide allowable semantic categories for the arguments of protein interaction predictions [1] like:

- Amino Acid, Peptide, or Protein;
- Biologically Active Substance;
- Biologic Function;
- Cell Function;
- Cell or Molecular Dysfunction;
- Molecular Function;
- Organic Chemical;
- Organism Function;

The experiments furnished some expected relations such as the inducement of neuronal cells (see Figure 1) or the fact that both APP and PrP are involved in inducing (see Figure 2).

Original sentence
Furthermore, treatement of cultures with 4-methylumbelliferly-beta-D-xyloside, a competitive inhibitor of proteoglycan glycanation, inhibited APP-induced neurite outgrowth but did not inhibit laminin-induced neurite outgrowth.

Arbiter-pi processed output
precursor amyloid protein-INDUCES-neurite outgrowth

Original sentence
PrP106-126 a peptide fragment of the prion protein induces proliferation of astrocytes.

Arbiter-pi processed output
peptide fragment, prion protein-INDUCES-proliferation, astrocyte

Fig. 1. Results of the experiments that obtain inducement of neuronal cells reproduced from [1]

The main result of this study was that by running over 70 sentences containing 40 marked predications, 27 protein interaction relationships were identified with 18 correctly. Therefore, recall was 45% and precision 67%.

Discovering functional similarity from textual information is parallel to what many researchers do in order to generate new hypotheses. Assistance

Original sentence
Then, we examined the effect of the amino-terminal fragment of sAPP and the epitope peptide of 22C11 antibody,and found that both of them also promoted DNA synthesis,suggesting that the amino terminal region of sAPP is responsible for the biological activity.

Arbiter-pi processed output
fragment of sAPP-INDUCES-dna synthesis

Original sentence
PrP106-126induces increased progression through the cell cycle to late G1 and enhances the level of both p53 and phosphorylated ERKs in astrocytes.

Arbiter-pi processed output
prp106-126-INDUCES-cell cycle, late g1

Fig. 2. Results of the experiments indicating that APP and PrP are involved in inducing DNA synthesis reproduced from [1]

from NLP has the potential to not only increase the number of articles that can be automatically reviewed but also the extraction of potential functional properties about certain proteins that had not been previously noticed.

It is important to state that linguistic approaches will never eliminate the need for experimental validation. Linguistic tools will also never replace biomedical researchers but using NLP tools helps generate testable hypothesis proving to be a boon for biomedical scientists.

1.2 Similarity due to conversed regions as index terms in information retrieval

A new technique for comparing pairs of biological sequences based on small patterns associated with highly conversed regions on some protein sequences is proposed in [4]. Highly conversed regions implies that the corresponding subsequences are not exactly the same, but only similar. Using a technique of transforming protein sequences by grouping amino acids into different sets according to their similarity and assign each set a unique code is a first step for this new strategy. The algorithmic procedure implies the determination of groups of similar amino acids based on a score matrix, like BLOSUM62. Two amino acids are considered similar if they have a positive score in BLOSUM62. There are used codes to represent similarity among amino acids and coding sequences to represent similarity among peptides. In order to generate codes, is constructed a set $A = \{b|a \text{ is similar to } b\}$ for each amino acid a, where a itself must be in A. Twenty such sets are generated (see Figure 3) but not all elements in such a set are similar; for example in the set $\{S, A, T, N\}$, S is similar to A, T and N but A is not similar to T, and T is not similar to

N. To each set is assigned a code. The authors of [4] remove redundant sets and kept only 15 sets. The codes and their corresponding sets are listed in Table 1.

$$\begin{array}{cccc} \{C\} & \{S,A,T,N\} & \{T,S\} & \{P\} \\ \{A,S\} & \{G\} & \{N,S,D,H\} & \{D,N,E\} \\ \{E,D,Q,K\} & \{Q,E,R,K\} & \{H,N,Y\} & \{R,Q,K\} \\ \{K,E,Q,R\} & \{M,I,L,V\} & \{I,M,L,V\} & \{L,M,I,V\} \\ \{V,M,I,L\} & \{F,Y,W\} & \{Y,H,F,W\} & \{W,F,Y\} \end{array}$$

Fig. 3. 20 sets of amino acids

Table 1. Codes and their corresponding code sets

Code	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Amino acids	C	S	T	P	A	G	N	D	E	Q	R	H	M	Y	F
		A	S		S		S	N	D	E	Q	N	I	H	Y
		T					D	E	Q	R	K	Y	L	F	W
		N					H		K	K			V	W	

A set of amino acids are *code similar* (or *c-similar*) if they are contained in the same code and a pattern is defined to be a sequence of codes having the length 4. A peptide of length 4 is said to be an instance of a pattern P if the amino acid at each position is included in the code at that position of P. A collection of peptides of length 4 are said to be *c-similar* if there exists a pattern P such that each peptide is an instance of P. In this way, for each protein sequence may be generated a code sequence consisting of its corresponding sets of codes. For example, the sequence V-L-S-T-D-N has the corresponding coding sequence {12}-{12}-{1,2,4,6}-{1,2}-{6,7,8}-{1,6,7}. Using a sliding window of length 4 through this coding sequence, for each subsequence of length 4 are generated all possible patterns. Considering the sequence V-L-S-T, the coding sequence created is {12}-{12}-{1,2,4,6}-{1,2} and the patterns generated are:

12-12-1-1; 12-12-1-2; 12-12-2-1; 12-12-2-2;
12-12-4-1; 2-12-4-2; 12-12-6-1; 12-12-6-2.

The -based similarity measure *Pattern(p)* is defined as the set of all patterns contained in the protein sequence p and is measured using one of the two following scores:

a)$S1(p1, p2) = c \times |Match(p1, p2)|/(|Pattern(p_1)| + |Pattern(p_2)|)$, where $Match$(p1, p2) is the set of patterns shared by sequences p_1 and p_2; c is a constant, normalizing factor used when the length of the proteins is ignored;

b)$S2(p_1, p_2) = c \times Match(p_1, p_2)|/(|Pattern(p_1)|+|Pattern(p_2)|)-|Length(p_1)+Length(p_2)|$, when the two protein-sequences are required to have the same length.

This method was applied to build phylogenetic trees , for proteins clustering and secondary structure prediction. The conclusion was that protein secondary structure prediction using patterns seems to outperform other existing methods; it is easy to implement and have relatively high sensitivity and specificity. In addition, the coding mechanism reduces the number of candidate fragments to be checked.

1.3 Semantic similarity measures

Many repositories of biological sequences may have a large amount of annotation associated with their entries. This ranges from semi-structured data, such as species information, to unstructured free text descriptions [5]. This additional information was important for human to read but it caused difficulties when trying to analyze it computationally. The growing interest in ontologies within bioinformatics provide a mechanism for capturing a community's view of a domain in a sharable form accessible by humans and also computationally amenable [5]. An ontology provides a set of vocabulary terms that label domain concepts. These terms should have definitions and be placed within a structure of relationships, the most important being the "is-a" relationship between *parent* and *child* and the "part-of" relationship between *part* and *whole*. The Gene Ontology (GO) is one of the most important ontologies within the bioinformatics community [6]. It comprises three orthogonal taxonomies or "aspects" that hold terms describing the *Molecular function*, *Biological process*, and *Cellular component* for a gene product. Gene Ontology represents terms within a Directed Acyclic Graph (DAG) consisting of a number of terms, represented as nodes within the graph, connected by relationships, represented as edges. Terms can have multiple parents as well as multiple children along the "is-a" relationships, together with "part-of" relations. The terms of this structure are used to annotate database entries (http://www.geneontology.org/goa). By providing a standard vocabulary across many biological resources such as SWISS-PROT and InterPro annotated protein databases, this kind of shared understanding enable query across the databases. One way to interrogate these databases would be to ask for proteins which are *semantically similar* to a query protein. Three of such semantic similarity measures are tested in [5].

The measurements are based on the information content of each term expressed as a probability . This is defined as the number of times of each term, or any child term occurs in the corpus.

Three measures of semantic similarity are tested on the same data using the information content of the shared parents of the two terms, as defined in Equation 1, where $S(c1, c2)$ is the set of parental concepts shared by both $c1$

and $c2$. As GO allows multiple parents for each concept, two terms can share parents by multiple paths. It is taken the minimum $p(c)$, where there is more than one shared parent, called p_{ms}

$$p_{ms}(c1, c2) = min_{s \in S(c1,c2)}\{p(c)\}. \tag{1}$$

A first measure is after Resnik [7] and uses only the information content of the shared parents. While p_{ms} can vary in general between 0 and 1, this measure vary between infinity (for very similar concepts) to 0. In practise, for terms actually present in the corpus, the maximum value of this measure is defined by $-ln(1/t) = ln(t)$, where t is the number of occurrences of any term in the corpus

$$sim(c1, c2) = -ln\ p_{ms}(c1, c2). \tag{2}$$

The second measure is after Lin [8] and uses both, the information content of the shared parents and that of the query terms. In this case, as $p_{ms} \geq p(c1)$ and $p_{ms} \geq p(c2)$, this value varies between 1(for similar concepts) and 0

$$sim(c1, c2) = \frac{2 \times [ln\ p_{ms}(c1, c2)]}{ln\ p(c1) + ln\ p(c2)}. \tag{3}$$

The last measure expressed in Equation 4 is after Jiang [9] and involve the semantic distance , which is the inverse similarity. It uses the same terms as (3) but not in the same order. According to [5], this can give arbitrarily large values although in practice has a maximum value of $2ln(t)$

$$dist(c1, c2) = -2ln\ p_{ms}(c1, c2) - (ln\ p(c1) + ln\ p(c2)). \tag{4}$$

While the interest is in semantic similarity between proteins, these measures are applied on protein sequences . In order to test the semantic similarity versus a classical method like that performed by BLAST tool, the correlation is calculated with Equation 5, where x_i and y_i are the semantic similarity between two proteins, over different aspects of GO, for all possible pairs of proteins identified in the SWISS-PROT-Human dataset

$$corr(x, y) = \frac{\sum(x_1 - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}. \tag{5}$$

In [5] the work was limited to those associations between GO terms, and SWISS-PROT-Human proteins available. For all three measures, the correlation coefficients show that sequence similarity is most tightly correlated (or inversely correlated in the case of distance measure) with the *Molecular function* aspect of GO, followed by the *Cellular component aspect* and finally the *Biological process aspect*. From the three the measures, the one proposed by Resnik prove the strongest correlation with sequence similarity.

The evidence in similarity performance may be observed in the Table 2 reproduced from [5].

Table 2. Correlation co-eficients between BLAST bit scores, and semantic similarity for three different measures and three biological aspects

Aspect	Resnik measure	Lin measure	Jiang measure
Molecular Function	0.577	0.541	-0.483
Biological process	0.280	0.303	-0.312
Cellular Component	0.368	0.452	-0.414

1.4 Language techniques for sequence relations

Advances in genome sequencing have made available amino acid compositions of thousands of proteins. Knowing the three dimensional shape of the protein, that is knowing the relative positions of each of the atoms in space, would give information on potential interaction sites in the protein. This aspect make possible the analysis or inference of the protein function. Thus, the study of determining or predicting protein structure from the amino acid sequences has secured an important place both in experimental and computational areas of research. Even if our interest was mainly on protein similarity methods using primary structure of proteins, we observed some language techniques that use the implication of secondary structure elements in establishing structural similarity relations between proteins. The main ideas of two of these approaches are shortly described in the following paragraphs.

Similarity search improved by a feature vectors extraction method

Feature vectors are extracted in [10] on triplets of Secondary Structure Elements (SSEs) that are later indexed using a multidimensional index structure. For the problem of finding similarities in protein structure datasets, this technique quickly prune away unpromising proteins in the database. The remaining sequences are then aligned using one of the available tools.

In order to construct the index structure is approximated each SSE using a line segment in 3-D. For each SSE is made a set of SSE triplets by considering the SSEs in the local neighborhood around that SSE. For each triplet is stored information about pairwise distances and pairwise angles for all pairs of SSEs in that triplet. The pairwise distance information is a range of values obtained by considering a set of points around the center of the line segment approximation of each SSE. This range is defined by using the minimum and maximum of these distances between the set of points chosen from the two SSEs under consideration. The pairwise angle information is a single value that measures the angle between the line segment approximations of the two SSEs. In this way they have a set of three range values and three angle values for each SSE triplet as the feature vector. These feature vectors are indexed using an R*-tree [11]. For a given query protein, the search technique runs in two phases:

- *Phase 1*: A set of feature vectors is obtained from the query protein and the R*-tree is searched using an appropriate range with each of these vectors. Using the results of these range searches, a candidate set of database proteins is determined at the end of this phase.
- *Phase 2*: A pairwise structure alignment program, is run on the candidate proteins to find the C_α alignments.

Experimental results show that this technique called PSI (*Protein Structure Index*) improve the pruning of VAST alignment tool while maintaining similar sensitivity.

Conceptual relations captured by LSA

Latent semantic analysis (LSA) is a method based on singular value decomposition (SVD) technique. It is an extremely useful approach in natural language processing to generate summaries, compare documents, generate thesauri and further for information retrieval [12],[13]. In the way that LSA captures conceptual relations in text, based on the word distribution in documents, the authors of [14] use it to capture secondary structure propensities (tendencies) in protein sequences using different vocabularies. Considering the documents $d_1, d_2, \ldots d_{N1}$ to be the non overlapping protein segments for which structural categories $C_1, C_2, \ldots, C_{N1}$ are known and $t_1, t_2, \ldots, t_{N2}$ the non overlapping test segments with known length for which secondary structure is to be predicted, the secondary structure of test data is predicted using a k nearest neighbor (kNN) classification method. For each test segment t_i the cosine similarity of t_i to all the training segments $d_1, d_2, \ldots, d_{N1}$ is computed and the k segments having maximum similarity with t_i are identified. These k segments are the kNN of t_i. The predicted category of t_i is the structural category to which belong the most of the kNNs. This process is repeated for each of the test segments. As vocabularies are considered: the 20 *amino acids*, *chemical groups* and *amino acid types*. For documents are considered: *helix*, *strand* and *coil* structures identified in each protein secondary structure. Next to the LSA method, vector space model (VSM) method was applied for the same models and data. Subsequences of proteins are treated as documents in vocabulary/documents matrix. Results are compared using and measures from information retrieval theory. They prove that VSM and LSA capture sequence preferences in structural types. Protein sequences represented in terms of chemical groups and amino acid types provide more clues on structure then the classically used amino acids as functional building blocks. Another aspect is that different alphabets differ in the amount of information they carry for a specific prediction task within a given prediction method.

1.5 Sequence similarity based on information theoretic methods

In many biological applications we are interested to quantify the similarity of a pair of sequences, and to further state to what extent one is redundant with

respect to the other, i.e., the information content in one is repeated in the other. Kolmogorov complexity theory, known also as algorithmic information theory, deals with quantifying the information in individual sequences. Algorithmic information theory has been introduced independently, with different motivations, by R.J. Solomonoff, A.N. Kolmogorov and G. Chaitin, between 1960-1966. Kolmogorov complexity of a sequence can be defined in an elegant way as the length of the program needed to be run for recovering the sequence on an universal computer. However, one of the important results of the theory tells that Kolmogorov complexity is non-computable, which makes it necessary to resort to approximations when the concept is used in practice. In many instances the evaluation of Kolmogorov complexity needed in various definitions of similarity is based on the simpler notion of codelength of a compressed sequence as provided by one of the general use compressors (like Lempel-Ziv or Burrows-Wheeler compressors). Several similarity distances based on approximates of the Kolmogorov complexity have been shown to perform well in different application such as: language and authorship recognition[15], plagiarism detection[16],[17], language tree and genome phylogenetic trees reconstruction[15],[18],[19],[20],[21],[22],[23], phylogeny of chain letters [24] or protein sequence classification[25].

The Kolmogorov complexity $K(x)$ of a sequence x is the length of the shortest binary program that outputs x on an universal computer and can be thought as the minimal amount of information necessary to produce x. The conditional Kolmogorov complexity, $K(x|y)$, is defined as the length of the shortest binary program that computes x when y is given as input, using an universal computer [26].

The problem of an absolute information distance metric between two individual objects was studied in [27] in terms of Kolmogorov complexity, which resulted in defining the information distance as the length of the shortest binary program that can transform either object into the other. Because the program is the shortest it has to take into account any redundancy between the information required to obtain x starting from y, or, to obtain y starting from x. It has been shown in [27] that the information distance equals

$$E(x, y) = \max\{K(x|y), K(y|x)\} \tag{6}$$

up to an additive $O(\log \max\{K(y|x), K(x|y)\})$ term. $E(x, y)$ satisfies the metric properties up to an additive finite constant.

The problem with $E(x, y)$ is that it measures the absolute information between the two objects without taking into account their lengths. For example, if we have two very long strings that differ only in 100 bit positions, that would represent about 0.001% of their length, we want to call them very similar; while if we have two other much shorter strings that differ also in 100 bit positions, but this represents about 90% of their length we want to call the strings very dissimilar. However, the quantities $E(x, y)$ may be close to 100 in both cases, making it difficult to evaluate the similarity/dissimilarity in these cases of notably different length of sequences.

To overcome this problem, the first attempt to a normalized distance was introduced in [18]. The normalized similarity distance aims to provide a relative similarity between the objects: when the distance is 0 the objects under investigation are maximally similar and when the distance is 1 the objects are maximally dissimilar.

The normalized distance introduced in [18] was based on the property that the algorithmic mutual information between two objects $I(x,y) = K(x) - K(x|y) = K(y) - K(y|x)$ is symmetric within an additive logarithmic factor [28], yielding the following formula for the of two sequences:

$$d(x,y) = 1 - \frac{K(x) - K(x|y)}{K(xy)} = 1 - \frac{K(y) - K(y|x)}{K(xy)} \tag{7}$$

where $K(xy)$ represents the Kolmogorov complexity of the concatenated strings. It has been shown in [19], that this distance is a metric , i.e. it has the following properties: (a)-positivity $d(x,y) > 0$ for $x \neq y$; (b)-identity $d(x,x) = 0$; (c)-symmetry $d(x,y) = d(y,x)$ and (d)-triangle inequality $d(x,y) \leq d(x,z) + d(z,y)$.

In the paper [21], the authors introduced the normalized information distance "mathematically, more precise and satisfying" than the previous one:

$$d_i(x,y) = \frac{max\{K(x|y^*), K(y|x^*)\}}{max\{K(x), K(y)\}} \tag{8}$$

where for any string v the notation v^* specifies the shortest binary program to compute v. There is an intuitive interpretation of this distance: if $K(y) > K(x)$ then $d_i(x,y) = \frac{K(y)-I(x,y)}{K(y)} = 1 - \frac{I(x,y)}{K(y)}$, where $I(x,y) = K(y) - K(y|x)$ is the algorithmic mutual information. It follows that $1 - d_i(x,y)$ is the number of bits of information that is shared between the two strings per bit of information of the string with most information. It is shown that the normalized information distance is a metric and it is universal up to a certain precision.

The problem with the normalized information distance is that its generality comes with the price of noncomputability, because it is expressed in terms of $K(x), K(y), K(x|y)$, and $K(y|x)$, which are noncomputable quantities. To overcome this problem, the normalized compression distance was introduced in [21], that uses a real-world reference compressor C to approximate Kolmogorov complexity:

$$NCD(x,y) = \frac{C(xy) - min\{C(x), C(y)\}}{max\{C(x), C(y)\}} \tag{9}$$

where $C(xy)$ denotes the compressed size of the concatenated sequences, $C(x)$ denotes the compressed size of x and $C(y)$ denotes the compressed size of y. It has been shown in [23] that if the compressor satisfies simple regularity conditions, then $NCD(x,y)$ is a similarity metric. The performance of the normalized compression distance was tested in [23] for applications in different

areas such as: genomic, music, language, handwriting, combination of objects from different domains, etc.

One of the most useful applications of the normalized compression distance is the comparison of two different genomes/proteomes and the study of the evolution of a group of species. Traditional phylogeny studies based on individual genes depend on the multiple alignment of the given gene shared by all the organism in the study. The problem with constructing phylogenetic trees based on individual genes is that different genes yield different trees. As the complete genomes for more and more organisms become available, the study of evolution based on the genome is more attractive. Any method based on multiple alignments will provide too many results to be combined when full genomes are involved, and the aggregation of partial results to infer a similarity measure has no straightforward solution.

To take full advantages of the normalized compression distance, it turns out that we need a very efficient compressor in order to have a powerful tool in genomic and proteomic sequence similarity studies. Several algorithms for compression of biological sequences have been proposed with varying degree of success. While the compression of DNA sequences was carefully studied in the last decade, less is known about the compressibility of protein sequences. The main reason is that compression of DNA sequences was shown successful from its first attempts in the early 90's, and that attracted many research groups to compete in capturing in the best way the regularities present in DNA sequences [29],[30],[31],[32],[33],[34]. Unexpectedly, the compression of protein sequences attracted less research, a possible reason being the fact that the first elaborate report on this topic was negative: in a 1999 paper [35] at Data Compression Conference an authoritative opinion was expressed in the negative, the statement making the title being: "Protein is incompressible". Although several plausible schemes have been tried in [35], including making use of the statistical description by substitution probabilities, the compression results by using these mutational processes were not better than the results by using only memory models of small order, leading to the conclusion that proteins are incompressible. In 2004, a report [22] was published, in which the compressibility of protein sequences is revealed in a scheme making use of substitution probabilities. The main feature of the scheme was the adaptivity in estimating the substitution probabilities, in that only the substitution statistics collected over regions where these statistics will improve the description length, when compared to the raw model or to a simple memory model, were used. The positive conclusion regarding the compressibility of protein sequences was drawn in [22] from the significant improvement of the compression results over the results reported in [35].

2 Statistical language modeling method for sequence similarity

The mapping of a protein sequence to its structure, functional dynamics and biological role becomes analogous to the mapping of words to their semantic meaning in natural languages. This analogy can be exploited by applying statistical language modeling and text classification techniques for the advancement of biological sequences understanding. The identification of Gramar/Syntax rules could reveal systematics of high importance for biological and medical sciences.

Some theoretical concepts

There are various kinds of language models that can be used to capture different aspects of regularities in natural language [36]. Markov chains are generally considered among the more fundamental concepts for building language models. In this approach, the dependency of the conditional probability of observing a word w_k at a position k in a given text is assumed to depend only upon its immediate n predecessor words $w_{k-n} \dots w_{k-1}$. The resulting stochastic models , usually referred as n-grams, constitute heuristic approaches for building language grammars.

Nowadays n-gram language modeling stands out as superior to any formal linguistic approach [37] and has gained high popularity due to its simplicity. Closely related with the design of models for textual data are algorithmic procedures for validating them. Apart from the justification of a single model, they can facilitate the selection of the specific one (among competing alternatives), most faithfully representing the available data. Entropy is a key concept for this kind of procedures. In general, its estimation is considered to provide a quantification of the information in a text and has strong connections to probabilistic language modeling [38]. As described in [39] and [40], the entropy of a random variable X that ranges over a domain $\aleph$, and has a probability density function , P(X) is defined as:

$$H(X) = -\sum_{X \in \aleph} P(X) log P(X). \tag{10}$$

In the specific case where a written sequence
$W = w_1, w_2, \dots, w_{k-1}, w_k, w_{k+1}, \dots$ is treated as a language model L based composition, the entropy may result in the following estimating formula:

$$\hat{H}(X) = -\frac{1}{N} \sum_{W^*} Count(w_1^n) log_2 p_L(w_n \mid w_1^{n-1}), \tag{11}$$

where the variable X has the form of an n-gram, $X = w_1^n \Leftrightarrow \{w_1, w_2, \dots, w_n\}$ and $Count(w_1^n)$ is the number of occurrences of w_1^n . The summation runs over all the possible n-length combinations of consecutive w ($i.e. W^* =$

$\{\{w_1, w_2, \ldots, w_n\}, \{w_2, w_3, \ldots, w_{n+1}\}, \ldots\})$ and N is the total number of n-grams in the investigated sequence. The second term in the summation (11) is the conditional probability that relates the n-th element of an n-gram with the preceding n-1 elements. Following the principles of maximum likelihood estimation (MLE)[39], it can be expressed by using the corresponding relative frequencies :

$$\hat{p}\left(w_n \mid w_1^{n-1}\right) = \frac{Count(w_n)}{Count(w_1^{n-1})}. \tag{12}$$

According to a general definition, the cross-entropy between the actual probability distribution of a data P(X), and the probability distribution Q(X) estimated from a model, is defined as:

$$H(X, Q) = -\sum_{X \in \aleph} P(X) log Q(X). \tag{13}$$

Two important aspects involved in this approach are: first, the cross-entropy of a stochastic process, measured by using a model, is an upper bound on the entropy of the process (i.e. $H(X) = H(X, Q)$) [39], [40]); second, as mentioned in [41], between two given models, the more accurate is the one with the lower cross-entropy. The above entropic estimations are the basis for building the protein similarity measure, described in the sequel.

Method description

Choosing a hypothetical protein sequence WASQVSENR, in the 2-gram modeling the available "words" are WA AS SQ QV VS SE EN NR, while in the 3-gram representation the words are WAS ASQ SQV QVS VSE SEN ENR. Based on the frequencies of these words (estimated by counting) and by forming the appropriate ratios of frequencies, the entropy of a n-gram model can be readily estimated (see 11). This measure is indicative about how well a specific protein-sequence is modeled by the corresponding n-gram model. While this measure could be applied to two distinct proteins (and help us to decide about which protein is better represented by the given model), the outcomes cannot be used for a direct comparison of the two proteins. This shortcoming is leading to a corresponding cross-entropy measure, in which the n-gram model is first built based on the word-counts of one protein sequence Y and then used in sequence X, contrasting the two proteins. Thus, the common information content between two proteins X, Y is expressed via the formula:

$$E(X, Y) = -\sum_{all w_1^n} P_X(w_1^n) log P_Y(w_n \mid w_1^{n-1}). \tag{14}$$

The first term $P_X(w_1^n)$ in the above summation refers to the reference protein sequence X (i.e. it results from counting the words of that specific protein). The second term, $log P_Y(w_n \mid w_1^{n-1})$, refers to the sequence Y based

on which the model has to be estimated (i.e. it results from counting the words of that protein). Variable w_1^n ranges over all the words (that are represented by n-grams) of the reference protein sequence.

Searching with this similarity measure

The essential point of this approach is that the unknown query-protein (e.g. a newly discovered protein) as well as each protein in a given database (containing annotated proteins with known functionality, structure etc.) are represented via n-gram encoding and the above introduced similarity is utilized to compare their representations. Here are devised two different ways in which the n-gram based similarity is engaged in efficient database searches. The most straightforward implementation is called hereafter as *direct method*. A second algorithm, the *alternating method*, was devised in order to cope with the fact that the proteins to be compared might be of very different length. The implication of this aspect is observable in the ratio value between the number of words from the reference sequence involved in computing the similarity score and the total number of words in the particular sequence (involved in first probability of equation (14)). Experimenting with both methods and contrasting their performances give the opportunity to check the sensitivity of the proposed measure to the length of the sequences.

Direct method:
Let Sq be the sequence of a query-protein and $\{S\} = \{S_1, S_2, \ldots S_N\}$ the given protein database. The first step is the computation of the '*perfect*' score (PS) or '*reference*' score for the query-protein. This is done by computing $E(S_q, S_q)$ using the query-protein both as reference and model sequence in equation (14). In the second step, each protein S_i, i=1,...,N, from the database serves as the model sequence in the computation of a similarity score $E(S_q, S_i)$, using the same equation (14) with the query-protein serving as the reference sequence. In this way, N similarities are computed $E(S_q, S_i)$, i=1,...,N. Finally, these similarities are compared against the perfect score PS. By computing the absolute differences $D(S_q, S_i) = |E(S_q, S_i) - PS|$, the 'discrepancies' in term of information content between the query-protein and the database-proteins are expressed. By these N measurements, we can easily identify the most similar proteins to the query-protein as those which have been assigned the lowest $D(S_q, S_i)$.

Alternating method:
The only difference with respect to the direct method is that when comparing the query-protein with each database-sequence, the role of reference and model protein can be interchanged based on which of the two sequences is the shortest (the shortest sequence plays the role of reference sequence in (14)). The rest of the steps (i.e. perfect-score estimation, ranking and selection) follow as previously.

Discussions

This strategy proposed for measuring protein similarity was demonstrated and validated in some experiments that gave a real motivation to keep the attention on it. For different n-gram models are performed the same validation procedures so that the best models to evaluate biological sequences proved to be 4-gram models. Performing an $N \times N$ comparisons in order to identify the similarity between the sequences involved in the experimental sets are obtained very good results. In works like [42], [43], for a 100 sequences and 4-gram model is obtained a score of about 98% true positive rates against 2% of false positives in a ROC (Receiver Operating Characteristic) evaluation. Also, using an exploratory data analysis method, the visual separation between similar and non similar groups of proteins is showing a high grade of accuracy. More experiments are in [44] proving that the biological information captured by this statistical modeling approach may reach the high performance of Clustal W in similarity scores accuracy. Considering the general dichotomy between "global" and "local" protein similarity measures, this new approach belongs to the former category. In this stage of its development should be mentioned one of the aspects that restrict a very good performance and that waits to be solved. It is a suitable normalization factor to cover the lack of frequencies for some events in a sequence. Conceptual simplicity and facility in implementation of this method deserve attention for future developments. The sequence content information are evaluated using concepts from theoretical information field, offering a large perspective in handling, understanding and exploration of biological sequences as text data.

3 Protein similarity detection based on lossless compression of protein sequences

The primary goal of a compression algorithm is to reduce as much as possible the size of a data set. For biological sequences, the study of their compressibility has a double value. The first one is a concrete, practical value, since storing or transmitting a sequence which was compressed leads to savings of computer resources and transmission costs. However, at the current compressibility rates obtained for DNA and proteins these savings seem to be only marginally important. The second value, which is probably the most important one, is the value of the statistical models uncovered by an efficient coding technique from the sequence to be compressed, models that can be subsequently used to measure the similarity of different sequences or for further statistical inference in various biological problems. The most useful application of protein similarity detection based on the compressibility of their sequences is the building of phylogenetic trees using the whole proteome of the given organisms. The proteome is the collection of all protein sequences in one organism.

3.1 Theoretical concepts

To define the relatedness or similarity of two proteome sequences, based on the average description length obtained by a given compressor, we introduce first the mutual information of two random variables X and Y that range over an alphabet $\mathcal{A}$. According to [45], the mutual information of two random variables is a measure of the amount of information one random variable contains about the other and is defined as:

$$I(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \tag{15}$$

where $H(X)$ and $H(Y)$ are the entropies of X and Y defined by (10). $H(Y|X)$ is the conditional entropy of Y given X and is defined as the expected value of the entropies of the conditional distributions over the conditioning random variable [45]:

$$H(Y|X) = \sum_{x \in \mathcal{A}} p(x) H(Y|X = x) = -E_{p(x,y)} \log p(Y|X) \tag{16}$$

Since the entropy is an idealistic measure of the average codelength for encoding a symbol generated by the source, it might be replaced by the implementable average codelength obtained by a compression algorithm, to obtain a realistic evaluation of average codelengths, or information content. Because the protein sequences are kept in text files, one might think that the classic algorithms for text compression could be used to compress such a file. Practically, these algorithms do not perform better than $\log_2 20$ bits per symbol which is the cost for encoding a symbol without any modelling. The reason for this poor performance is that protein sequences obey other rules than human created text. It turns out that we need a specialized compressor for proteome sequences. Such a compressor, named *ProtComp* was introduced in [22], and the main idea of the algorithm and its use in computing the similarity of two sequences, is described in the next section.

3.2 ProtComp algorithm and its use in sequence comparison

One of the first attempt at compressing proteins [35], presented a result in the negative, claiming from the title that protein is incompressible. The authors proposed a scheme base on a sophisticated context algorithm, where not only the current context is used, but also similar contexts are used for prediction, the weighting of contexts being dependent on the mutation probabilities of amino-acids. However, their results were not better than the results obtained with simple, low order Markov models, which led them to the conclusion that the approximate repeats in protein can not provide statistically significant information to be exploited for compression. The main problem of this scheme is that the model used is based on the patterns found in the immediate vicinity of the symbol that is to be encoded at a given moment and it ignores the

redundancy of different regions of the proteome, that are the result of different biological phenomenon such as gene duplication.

The *ProtComp* algorithm was developed with the purpose of extracting the regularities at the scale of full proteome, by searching for approximate repeats and adaptively estimate the amino acids substitution probabilities over the regions where the statistics will improve the description length of that region, when compared to the description length obtained by the raw model. The *ProtComp* algorithm can operate in *single* mode, when the input is a single proteome and the regularities are extracted based on this sequence; and it can operate also in *conditional* mode, when the input is composed of two proteomes, one that is encoded and one that is considered to be known and the regularities are extracted from both sequences.

ProtComp is a two pass algorithm. The goal of the first pass is to collect the substitution probabilities and the goal of the second pass is to encode the symbols. The proteome sequence that is to be encoded is split into non-overlapping blocks of same length and for each *current block* we look for that block in the sequence from the beginning until the current block, obtaining the greatest number of matches (the same amino acid in the same position) and we refer to it as a *regressor block*. When the algorithm operates in the conditional mode, the regressor block can be found in the conditioning proteome. In the first pass, the probability substitution matrix is collected for the symbols having the same rank in the current block and in its regressor block, but only for those pairs (current block, regressor block) for which the number of matches is greater than a given threshold. For each conditional distribution, obtained from each row of the substitution matrix, we design an optimal Huffman code [46], and all these optimal codes have to be transmitted to the decoder as a prefix of the encoded stream. In the second pass, we go through the proteome sequence and encode a block that have the number of matches greater than the threshold using the optimal codes built in the first pass. The rest of the blocks, for which the number of matches does not exceed the threshold, are encoded by arithmetic coding [47], using the statistics of an adaptive first order Markov model [48].

Inspired by the mutual information of two random variables $I(X, Y)$ (15), and using the average codelengths obtained by *ProtComp* algorithm, in [22] the relatedness or similarity of two proteomes X and Y was defined as:

$$R(X, Y) = ProtComp(X) - ProtComp(X|Y) \tag{17}$$

where $ProtComp(X)$ is the average codelength obtained when the proteome X is encoded by *ProtComp* in the single mode and $ProtComp(X|Y)$ is the average codelength obtained when the proteome X is encoded by *ProtComp* in the conditional mode, i.e. when the proteome Y is given as input argument and is considered to be known. The value of $R(X, Y)$ will be close to $ProtComp(X)$ when the two proteomes are very similar, because $ProtComp(X|Y)$ will tend to zero (if Y is already known and X is very similar to Y then the average codelength of encoding X knowing Y will tend to zero) and the value of

$R(X,Y)$ will tend to zero when X and Y are maximum dissimilar (if X and Y are completely dissimilar or independent, than knowing Y does not help at all and the average codelength of encoding X knowing Y will tend to $ProtComp(X)$).

Using this measure of similarity of two proteomes a phylogenetic tree can be built for a given set of organisms. The building process of the phylogenetic tree consists at each step in computing the relatedness of all pairs of proteomes and the two proteomes that yield the maximum R are grouped, i.e. the two proteome sequences are concatenated and in the next step they form a new single proteome sequence. The process is repeated until only two proteome sequences are left.

The fact that plausible phylogenetic trees can be built using this measure of similarity means that the *ProtComp* algorithm manages to capture the biological meaningful features of the proteome sequences. Thus, a natural question arise, i.e. if the similarity measure defined based on the average codelength produced by *ProtComp* can be used to measure the similarity of two protein sequences, which are of much shorter length than a whole proteome. Using a slightly different version of *ProtComp* algorithm, the relatedness of two protein sequences can be defined similar to (17) such that the resulted codelength can be seen as the sum of substitution scores over the similar parts of the proteins and can be compared to alignment scores obtained by classical algorithms for sequence comparison.

A modified version of the *ProtComp* algorithm and its use in defining a measure of similarity for pairs of proteins was presented in [49]. Because the goal now is to compare pairs of proteins in different organisms, the substitution matrix is not collected for each pair of proteins because there is not enough statistics at this level and only the cost of transmitting the matrix may be greater than the cost of encoding the whole sequence of amino acids. Then, for each pair of organisms, a substitution frequency matrix is collected at the proteome level and the associated Huffman codes are used to compute the similarity of proteins from that organisms.

Following the same idea as in the original *ProtComp* algorithm, for a given pair of proteins, the protein to be encoded is first compressed using the statistics of its own sequence and then is conditionally encoded using the statistics of the other sequence. The similarity of the two proteins is given by the difference in the encoding costs. The protein to be encoded is also split in non-overlapping blocks of a certain length. In the first case, the regressor block is searched only in the already seen sequence and in the second case, the regressor is searched also in the other protein. The pair of blocks having the number of matches greater than a fixed threshold are encoded conditional on their regressor using the Huffman codes designed at the proteome level, while the blocks with less number of matches than the threshold are encoded in clear using $\log_2(20)$ bits/amino acid. Using this method to encode the amino acids in the blocks with number of matches less than the threshold has an interesting interpretation when computing the relatedness of the two proteins.

Let $X = x_1, \ldots, x_{N_x}$ be the protein which is to be encoded and $Y = y_1, \ldots, y_{N_y}$ the conditioning protein. For each block $x^k = x_{(i-1)k+1}, \ldots, x_{(i-1)k+L}$, where L is the length of the block, a regressor block is found $r_p^k = r_1, \ldots, r_L$ where depending on the value of p we have two cases: $r_1, \ldots, r_L = x_t, \ldots, x_{t+L-1}$ if $p = 1$, which means that the regressor is found before the current block in the protein sequence which is to be encoded, or $r_1, \ldots, r_L = y_t, \ldots, y_{t+L-1}$ if $p = 2$, which means that the regressor is found in the other protein. If $p = 1$, then $t \leq (i-2)k + L$ and if $p = 2$, then $t \leq N_y$. The similarity of the two proteins X and Y is given by $R(X, Y)$ (17), where *ProtComp* is modified to work with short sequences. Let $N_b = \lfloor \frac{N_x}{L} \rfloor$ be the number of non-overlapping blocks in the protein X and let $\mathcal{L}(a|b)$ be the encoding cost of a given block a when the regressor block b is given. Then (17) becomes:

$$\begin{aligned} R(X,Y) &= \sum_{k=1}^{N_b} [\mathcal{L}(x^k|r_1^k) - \mathcal{L}(x^k|r_p^k)] = \\ &= \sum_{i=1}^{N_{b1}} [\mathcal{L}(x^i|r_1^i) - \mathcal{L}(x^i|r_p^i)] + \\ &+ \sum_{j=1}^{N_{b2}} [\mathcal{L}(x^j|r_1^j) - \mathcal{L}(x^j|r_2^j)] = \\ &= \sum_{j=1}^{N_{b2}} [\mathcal{L}(x^j|r_1^j) - \mathcal{L}(x^j|r_2^j)] \end{aligned} \tag{18}$$

where N_{b1} is the number of blocks for which the number of matches is less then the fixed threshold and N_{b2} is the number of blocks for which at least in the conditional case the number of matches is greater than the fixed threshold. Because the sum of the encoding costs over the blocks that don't have the number of matches greater than the threshold is the same even the protein is conditionally encoded or not, is zero, we can further write (18) as:

$$\begin{aligned} R(X,Y) = &\sum_{i=1}^{N_{b21}} [L \log_2(20) - \mathcal{L}(x^i|r_2^i)] + \\ &+ \sum_{j=1}^{N_{b22}} [\mathcal{L}(x^i|r_1^i) - \mathcal{L}(x^i|r_2^i)] \end{aligned} \tag{19}$$

where N_{b21} is the number of blocks for which only in the conditional case the number of matches exceeds the fixed threshold and N_{b22} is the number of blocks for which in both cases the number of matches is greater than the fixed threshold. It turns out that our method to define the relatedness of two proteins is in fact a measure of the local similarities of the two proteins because the regions where the proteins are not similar are discarded.

3.3 Experimental results

To assess the ability of the *ProtComp* algorithm to help on finding similarities at a macro scale, i.e. at proteome level, as well at a micro scale, i.e. at the protein level, two experiments were done in [22] and [49].

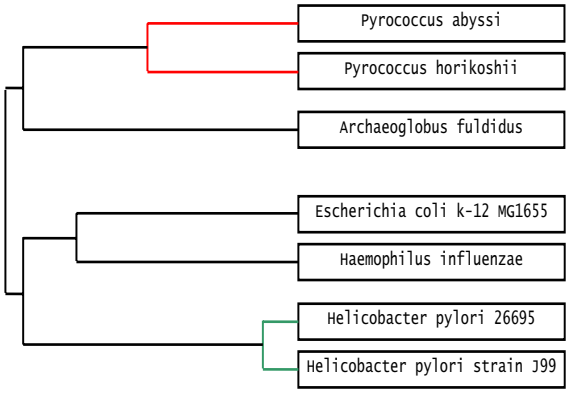
In the first experiment, the goal was to build a phylogenetic tree, to see if the similarity measure (17) can manage to capture regularities at the full proteome level. To do this, seven organisms for which the phylogentic tree was built previously at the genome level [18], were used. The proteome sequences for the following organisms have been used: *Archaeoglobus fulgidus* (AF), *Escherichia coli K-12 MG1655* (EC), *Pyrococcus abyssi* (PA), *Pyrococcus horikoshii* (PH), *Haemophilus influenzae Rd* (HI), *Helicobacter pylori 26695* (HP1) and *Helicobacter pylori, strain J99* (HP2). The first step in building the tree is to compute the similarity between all the proteomes and the results are presented in Table 3. Because the similarity measure is not symmetric in practice, we pick the proteomes having the highest sum $R(X,Y) + R(Y,X)$ as being the most related, so that for the first step $HP1$ and $HP2$ are the most related. For the next step, $HP1HP2$ is treated as a single proteome and we have to compute the relatedness between this new proteome and all the others. In the second step, the most related proteomes will be PA and PH, in the third step EC and HI, in the fourth step $PAPH$ and AF, in the fifth step $HP1HP2$ and $ECHI$ and finally, $HP1HP2ECHI$ and $PAPHAF$. The final phylogenetic tree, that is the same as in [16], is presented in Figure 4.

Table 3. The similarity between all the proteomes in the first step of the phylogenetic tree building.

	AF	EC	PA	PH	HI	HP1	HP2
AF		0.01	0.11	0.10	0.01	0	0
EC	0		0	0	0.29	0.03	0.03
PA	0.14	0.01		1.51	0.01	0	0
PH	0.13	0.01	1.52		0	-0.01	0
HI	0.01	0.78	0.01	0		0.07	0.07
HP1	0.01	0.09	0	0	0.08		**2.22**
HP2	0.01	0.09	0	0	0.08	**2.26**	

In the second experiment, the goal was to test if the similarity measure (17) can be used to capture regularities at the protein level, i.e. to detect related proteins. To do this, for different pairs of organisms two data sets were chosen: the positive control data set, i.e. all the orthologous proteins in the two organisms and the negative control data set, i.e. all pairs of proteins in the positive set, except the orthologous pairs. For the positive data set the similarity measure should yield values as big as possible, while for the negative control data set, the similarity measure should yield values close to zero.

Fig. 4. The resulted phylogenetic tree



In [50] the authors studied if standard substitution matrices, like BLOSUM [51] are appropriate for comparison of sequences with non-standard amino acid composition. They argue that in most commonly used substitution! matrices, the substitution score is in the form of log-odds ratio of the target frequencies and of the background frequencies, derived from accurate alignments of closely related proteins. These matrices are then appropriate for comparison of protein sequences for which the amino acid composition is close to the background frequencies used to construct them. Unfortunately, the standard substitution matrices are also used when comparing protein sequences with very different background frequencies. To overcome this problem, a method for adjusting the implicit target frequencies of the substitution matrix used for comparison was presented. They show that composition specific substitution matrix adjustment is useful for comparing compositional biased proteins, including those of organism with nucleotide bias and therefore with codon bias composition.

To test the ability of the similarity measure (17) we used the same data set as in [50] where three pairs of organisms with very biased AT or GC genomes were used. The three pairs of organisms considered are: (i) *Clostridium tetani* (AT-rich) and *Mycobacterium tuberculosis* (GC-rich) with contrasting strong biases; (ii) *Bacillus subtilis* and *Lactococcus lactis* both with relatively unbiased genomes; and (iii) *Mycobacterium tuberculosis* and *Streptomyces coelicolor* with strong biases in the same GC direction. For each pair of organisms there is one positive control and one negative control data set. The optimal Huffman codes used by *ProtComp* algorithm are built at the proteome level of the organism for which the proteins are compared.

The results for the positive control data set, for each pair of organisms are listed in Table 4. In this table, the values in the columns denoted by

"Relatedness" are computed with (19), while the others are taken from [50]. In the fourth column, for each pair of organisms, is listed the mean of the local alignment bit scores obtained when using a scaled version of the BLOSUM 62 substitution matrix for comparing the orthologous proteins. In the fifth and sixth columns are listed the median change in bit scores with respect to BLOSUM 62, when using composition-adjusted BLOSUM 62 matrices. For the composition-adjusted BLOSUM 62 matrices, the background frequencies were adjusted for proteome frequencies (the column denoted by "Organism") and for the frequencies of the two sequences considered (the column denoted by "Sequence"). In the last two columns are presented: the median changes in bit score when using (19) to compute the local similarity score with respect to the values obtained when using the scaled version of the BLOSUM 62 substitution matrix and the median changes in bit score when using (19) to compute the local similarity score with respect to the values obtained when composition adjusted BLOSUM 62 matrices with background frequencies adjusted for the frequencies of the two sequences compared.

For the negative control data set, the only values that can be reported are the mean of the local alignment bit score for the three pair of organisms, because the number of sequences compared for each pair of organisms is quite big and the original paper [50] did not reported the local alignment bit score for all protein comparisons, when using the scaled version of BLOSUM 62 matrix or compositional adjusted BLOSUM 62 matrices. The third column contains the mean bit score obtained when comparing the unrelated protein pairs using the scaled version of the BLOSUM 62 substitution matrix and the last column presents the mean bit score when comparing the unrelated pairs of proteins using (19).

From the results presented for the positive control data set in Table 4 and the results presented for the negative control data set in Table 5, it can be concluded that the similarity measure computed based on the codelength obtained by the modified version of the *ProtComp* algorithm, does not artificially increase the local similarity score, because for the two pairs of organisms for which it yields a bigger median change in the bit score for the positive data set, it also yields a smaller mean bit score for the negative control data set.

Table 4. The relatedness computed for the orthologous pairs of proteins in the three pairs of organisms. [1] Values taken from [50]. [2] Values computed with (19).

Sequence pairs	Organisms compared	No. of sequences	Mean BLOSUM 62 bit score	Median change in bit score with respect to BLOSUM 62			Median change in bit score with respect to Sequence
				Organism [1]	Sequence [1]	Relatedness [2]	Relatedness [2]
Related	*C.tetani* and *M.tuberculosis*	40	68.3	+1.6	**+2.3**	+0.6	-3.5
	B.subtilis and *L.lactis*	37	59.8	+1.1	+2.1	**+10.9**	+7.5
	M.tuberculosis and *S. coelicolor*	34	58.6	+1.4	+2.7	**+4.6**	+1.79

Table 5. The mean bit score for the unrelated pairs of proteins in the three pairs of organisms. [1] Values taken from [50]. [2] Values computed with (19).

Sequence pair	Organism compared	No. of sequences	Mean bit score	
			BLOSUM 62 [1]	Relatedness [2]
Unrelated	*C. tetani* and *M. tuberculosis*	1,560	**16.7**	17.05
	B.subtilis and *L. lactis*	1,332	15.7	**12.26**
	M. tuberculosis and *S. coelicolor*	1,122	16.4	**14.33**

3.4 Discussions

The similarity measure of two sequences $R(X, Y)$ computed based on the estimated information shared by the sequences using the *ProtComp* algorithm, is able to operate at a macro scale, by comparing proteome sequences and at a micro scale, by comparing protein sequences. The results show that the *ProtComp* algorithm manages to capture the biological meaningful patterns of protein sequences and it proves that certainly there are regularities in the protein sequences that can be exploited in order to compress protein sequences. This conclusions are in deep contrast with the conclusions in [35] were the authors stated from the title that "Protein is incompressible".

4 Conclusions

This chapter is conceived as a collection of linguistic and theoretical techniques tested in practice, which claim attention on textual analysis of biological sequence descriptions. Even if they do not perform at this stage with excellent results, to a further stage in development is possible to become alternative and efficient similarity methods to those classical, based on sequence alignment. As textual description of proteins offers the opportunity to encode biological information, theoretical information measures bring their contribution in quantification of this data in relevant forms for comparisons or relational attributes detection. Each of the methods introduced here serves to a well defined scope and are tight connected to the input data format. If natural language processing methods uses biological description of sequences they may extract sequence relationships. Index terms, largely used in information! retrieval field, found their application in sequence similarity detection using conversed regions or secondary structure attributes. In this way, relations between sequences involve new developed techniques or some largely known like LSA, SVM or kNN. Gene Ontology is another interesting way of data description allowing semantical similarity search for a query protein. Kolmogorov

complexity is many times applied in methods that try to quantify the similarity between pairs of sequences. In addition to these linguistic and theoretical methods, the two methods new introduced are bringing their contribution to the investigated direction. Linguistic models seems to be able to capture sequence information content while a lossless compression algorithm open the perspective of sequence similarity measurement as coded information. It deserve to be underlined the idea that many strategies originally developed for linguistic and information theoretical field found their application in biological knowledge discovery. We let the reader to evaluate and appreciate each of the methods while in this context they comes to add more color to the panoramic view of proteins sequence similarity detection.

References

1. Sarkar I, Rindflesch T (2002) Discovering Protein Similarity using Natural Language Processing, Cognitive Science Branch of Lister Hill National Center for Biomedical Communications, National Library of medicine.
2. Aronson AR (2001) Effective mapping of biomedical text to the UMLS Metathesaurus: The MetaMap program, Proc.AMIA Symp :17-21.
3. Humpreys BL, Lindberg DAB, Schoolman HM, Barnet GO (1998) The Unified Medical Language System:An informatics research collaboration. JAMIA, 5(1):1-13.
4. Wu K-P, Lin H-N, Sung T-Y and Su W-L (2003) A new Similarity Measure among Protein Sequences. IEEE Computer Society Bioinformatics Conference (CSB'03)Proceedings :347–352.
5. Lord PD, Stevens RD, Brass A nd Goble CA (2003) Semantic similarity measures as tools for exploring the gene ontology, PubMed, Pac. Symp. Biocomput.: 601-612.
6. The Gene Ontology Consortium (2001) Creating the gene ontology resource: design and implementation. Genome Res 11(8):1425–33.
7. Resnik P (1995) Using information content to evaluate semantic similarity in a taxonomy. IJCAI :448–453.
8. Lin D (1998) An information-theoretic definition of similarity. In Morgan Kaufman(EDS) Proc 15th International Conf. on Machine Learning. San Francisco, CA :296–304.
9. Jiang JJ and Conrath DW (1998) Semantic similarity based on corpus statistics and lexical taxonomy. In Proc.of International Conference on Research in Computational Linguistics, Taiwan.
10. Camoglu O, Kahveci T and Sigh AK (2003) PSI:indexing protein structures for fast similarity search. Bioinformtics 19(1):i81–i83.
11. Beckmann N, Kriegel HP, Schneider R and Seeger B (1990) The R*-tree: an efficient and robust access method for points and rectangles. SIGMOD, :322–331.
12. Bellegarda J (2000) Exploiting latent semantic information in statistical language modeling. IEEE Proc. 88(8):1279–1296.
13. Landauer T, Foltx P and Laham D (1998) Introduction to latent semantic analysis. Discourse Processes 25:259–284.

14. Ganapathiraju MK, Klein-Seetharaman, Balakrishnan N and Reddy R (2004) Characterization of Protein Secondary Structure.Application of latent semantic analysis using different vocabularies.IEEE Signal Processing Magazine, May 2004 :78–86.
15. Benedetto D, Caglioti E, and Loreto V (2002) Language trees and zipping. Physical Review Letters 88(4):048702.
16. Chen X, Francia B, Ming L, McKinnon B and Seker A (2004) Shared information and program plagiarism detection. IEEE Transactions on Information Theory 50(7):1545–1551.
17. Grozea C (2004) Plagiarism detection with state of the art compression programs. In: CDMTCS Research Report Series.
18. Chen X, Kwong S, and Li M (1999) A compression algorithm for DNA sequences and its applications in genome comparison. In: Genome Informatics. Universal Academy Press. Tokyo.
19. Li M., Badger J.H., Chen X., Kwong S., Kearney P., and Zhang H. (2001) An information-based sequence distance and its application to whole mitochondrial genome phylogeny. Bioinformatics, 17(2):149154.
20. Otu H.H. and Sayood K. (2003) A new sequence distance measure for phylogenetic tree construction. Bioinformatics, 19(16):2122–2130.
21. Li M., Chen X., Li X., Ma B., and Vitnyi P.M.P. (2004) The similarity metric. IEEE Transactions on Information Theory, 50(12):3250–3264.
22. Hategan A and Tabus I (2004) Protein is compressible. In: Proceedings of the 6th Nordic Signal Processing Symposium - NORSIG2004.
23. Cilibrasi R and Vitnyi P M P (2005) Clustering by compression. IEEE Transactions on Information Theory, 51(4):1523–1545.
24. Bennett C H, Li M, and Ma B (2003) Chain letters and evolutionary histories, Scientific American 288(6):76–81.
25. Kocsor A, Kertsz-Farkas A, Kajn L, and Pongor S. (2006) Application of compression-based distance measures to protein sequence classification: a methodological study. Bioinformatics, 22(4):407–412.
26. Kolmogorov A N (1965) Three approaches to the definition of the concept "quantity of information". Problemy Peredachi Informatsii, 1:3–11.
27. Bennett C H, Gacs P, Li M, Vitanyi P M B, Zurek W H (1998) Information Distance, IEEE Transacations on Information Theory, 44(4):1407–1423.
28. Li M. and Vitnyi P.M.P. (1997) An Introduction to Kolmogorov Complexity and its Applications. Springer-Verlag, 2nd ed.
29. Apostolico A. and Lonardi S. (2000) Compression of biological sequences by greedy off-line textual substitution. In: Data Compression Conference. IEEE Computer Society Press.
30. Chen X., Kwong S. and Li M.(2001) A compression algorithm for DNA sequences. IEEE-EMB Special Issue on Bioinformatics, 20(4):61–66.
31. Chen X., Li M., Ma B., and Tromp J. (2002) DNACompress: Fast and effective DNA sequence compression. Bioinformatics 18:1696-1698.
32. Grumbach S. and Tahi F. (1993) Compression of DNA sequences. In: Data Compression Conference. IEEE Computer Society Press.
33. Korodi G. and Tabus I. (2005) An efficient normalized maximum likelihood algorithm for DNA sequence compression. ACM Transactions on Information Systems 23(1):3–34.

34. Tabus I., Korodi G. and Rissanen J. (2003) DNA Sequence Compression Using the Normalized Maximum Likelihood Model for Discrete Regression. In: Data Compression Conference. IEEE Computer Society Press.
35. Nevill-Manning C. G. and Witten I. H. (1999) Protein is incompressible. In: Data Compression Conference. IEEE Computer Society Press.
36. The ASTRAL Compendium for Sequence and Structure Analysis, http://astral.berkeley.edu.
37. Wang S, Schuurmans D, Pengun F and Zhao Y (2003) Semantic N-gram Language Modeling With The Latent Maximum Entropy Principle. In IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-03) available at : http://citeseer.nj.nec.com/575237.html
38. Van Compernolle D (2003) Spoken Language Science and Technology, available at : *http* : *//www.esat.kuleuven.ac.be/ compi/pub/spoken*$_l$*anguage/TOC.htm*
39. Manning CD and Schtze H (2000) Foundations of statistical natural language processing. Massachusetts Institute of Technology Press, Cambridge, Massachusetts London, England,:554 - 556;557 - 588.
40. Brown PF, Della Pietra AS, Della Pietra VJ, Mercer Robert LR and Jennifer CL (1992) An estimation of an upper bound for the entropy of English. In Association for Computational Linguistics, Yorktown Heights, NY 10598, P.O. Box 704.
41. Jurafsky D and Martin J (2000) Speech and Language Processing. Prentice Hall(EDS) :223-231.
42. Bogan-Marta A, Gavrielides M, Pitas I and Lyroudia K (2005) A New Statistical Measure of Protein Similarity based on Language Modeling. GENSIPS 05, IEEE International Workshop on Genomic Signal Processing and Statistics,Newport, Rhode Island, USA
43. Bogan-Marta A, Laskaris N, Gavrielides M, Pitas I, Lyroudia K (2005) A novel efficient protein similarity measure based on n-gram modeling. IEEE, IEE, CIMED 2005, Second International Conference on Intelligence in Medicine and Healthcare, Costa da Caparica, Lisbon, Portugal :122-127;
44. Bogan-Marta A, Pitas I, Lyroudia K (2006) Statistical Method of Context Evaluation for Biological Sequence Similarity. IEEE, IFIP World Computer Congress, Santiago de Chile, 21-24 August, published in 'Artificial Intelligence in Theory and Practice', Springer New York, 2006, pos. 11:1-10;
45. Cover T. M. and Thomas A. J. (1991) Elements of information theory, New York.
46. Huffman D. A. (1952) A method for the construction of minimum redundancy codes. Proceedings of the IRE 40:1098–1101.
47. Rissanen J.J. (1976) Generalized Kraft inequality and arithmetic coding. IBM Journal of Research and Development 20:198–203.
48. Ross S. M. (1996) Stochastic processes, 2nd Edition, New York.
49. Hategan A. and Tabus I. (2005) Detecting local similarity based on lossless compression of protein sequences. In International workshop on Genomic Signal Processing 95–99.
50. Yu Y-K, Wootton J. C and Altschul S. F. (2003) The compositional adjustment of amino acid subtitution matrices. PNAS 100(26):15688–15693.
51. Henikoff S. and Henikoff J. G. (1992) Amino acid substitution matrices from protein block. Proceedings of the National Academy of Science USA 89(22):10915–10919.

Index