Granular Computing Techniques for Bioinformatics Pattern Recognition Problems in Non-metric Spaces

Alessio Martino, Alessandro Giuliani and Antonello Rizzi

Abstract Computational intelligence and pattern recognition techniques are gaining more and more attention as the main computing tools in bioinformatics applications. This is due to the fact that biology by definition, deals with complex systems and that computational intelligence can be considered as an effective approach when facing the general problem of complex systems modelling. Moreover, most data available on shared databases are represented by sequences and graphs, thus demanding the definition of meaningful dissimilarity measures between patterns, which are often non-metric in nature. Especially in such cases, evolutive and fully automatic machine learning systems are mandatory for dealing with parametric dissimilarity measures and/or for performing suitable feature selection. Besides other approaches, such as kernel methods and embedding in dissimilarity spaces, granular computing is a very promising framework not only for designing effective data-driven modelling systems able to determine automatically the correct representation (abstraction) level, but also for giving to field-experts (biologists) the possibility to investigate information granules (frequent substructures) that have been discovered by the machine learning system as the most relevant for the problem at hand. We expect that many important discoveries in biology and medicine in the next future will be determined by an increasingly stronger integration between the ongoing research efforts of natural sciences and modern inductive modelling tools based on computational intelligence, pattern recognition and granular computing techniques.

A. Martino (✉) · A. Rizzi
Department of Information Engineering, Electronics and Telecommunications,
University of Rome "La Sapienza", Via Eudossiana 18, 00184 Rome, Italy
e-mail: alessio.martino@uniroma1.it

A. Rizzi
e-mail: antonello.rizzi@uniroma1.it

A. Giuliani
Department of Environment and Health, Istituto Superiore di Sanità,
Via Regina Elena 299, 00161 Rome, Italy
e-mail: alessandro.giuliani@iss.it

W. Pedrycz and S.-M. Chen (eds.), *Computational Intelligence for Pattern Recognition*, Studies in Computational Intelligence 777,
https://doi.org/10.1007/978-3-319-89629-8_3

Keywords Computational intelligence · Pattern recognition · Machine learning
Granular computing · Bioinformatics · Computational biology · Systems biology
Non-metric spaces analysis

1 Introduction

1.1 Bioinformatics, Computational Intelligence and Pattern Recognition

The word 'bioinformatics' took different meanings since its introduction around forty years ago [17]. The definition of an autonomous 'bioinformatics' field started with the need to efficiently analyse and store increasing amounts of sequence data. Consequently, in the first years of the application of computational science in biology, bioinformatics was mainly devoted to technical and instrumental problems with no relation at all with the core of biological sciences. Computational scientists were hired to give a service to biologists because 'they were able to play with computers' in a way not too dissimilar of any laboratory technician taking care of a spectrophotometer properly working.

It is worth noting that the relation between biology and statistical methodology (the first root of pattern recognition approaches in life sciences) started with completely different premises. From the beginning of their relation, in the first years of the last century, biology and statistics interacted on a peer-to-peer basis and many statistical tools were developed in the core of biological community (e.g. Ronald Fisher, one of the fathers of modern statistics, was a geneticist and he developed linear regression in the frame of human genetics and evolution studies [65, 84, 95]).

During the years, the relation of biology with bioinformatics became something more than a purely occasional affair and approachedv the 'true-love wedding' level of the one-hundred years lasting relation between biology and statistics. Notwithstanding that, the term 'bioinformatics' is still largely prevalent with respect to other terms lexically more suited for describing the growing maturity of Biology and Computational Intelligence relation, such as 'computational biology' and 'systems biology'.

Besides the terminology, pattern recognition and computational intelligence techniques are nowadays gaining attention from the bioinformatics community [43, 71]. Many machine learning problems that can be instantiated in both biology and medicine are defined on domains in which each entry of the database at hand is a data structure far more complex than a plain real-valued feature vector, such as sequences, graphs, images or often even more complex structures arising from the concatenation of different data types (unconventional, structured data). Dealing with such structured domains usually demands to be able to define custom and meaningful (dis)similarity measures between elements in such unconventional domains, relying on sequence

and graph matching techniques. Specifically, networks (graphs) are nowadays the most powerful approaches to describe the complexity behind biological systems.

In fact, the application of computationally intensive methods to biological problems became strictly intermingled with the actual frontiers of biomedicine and went well beyond the biological polymers sequence analysis, directly tackling the archetypal form of biological objects from protein science to ecology: complex networks, interpreted as simplified, yet powerful, representations of complex systems.

Complex systems are everywhere in nature, as well as in most artificial systems designed and built by mankind (telecommunications and energy distribution systems, as instances). Complex systems are by far more frequent than 'simple' ones, which are the true outliers in our world. However, a precise definition of what should be a 'complex system' is still a disputable issue. This challenge is due to the fact that complex systems are nowadays a research topic faced by many different scientific areas, such as mathematics, biology, physics, chemistry and engineering, each one bringing its own point of view, concepts and terms into the discussion. Since 1995, when John Horgan published his famous paper entitled "From Complexity to Perplexity" [36] evidencing the lack of a shared and precise definition about complex systems, the debate is still well alive. However, most authors agree in considering the following characteristics as necessary conditions to consider a given system as 'complex':

- The system is composed by many mutually interactive elements
- Elements behaviour is characterised by nonlinear dynamics
- The graph representing the causal relationships between elements contains loops

Elements are usually defined as atomic entities at the semantic level chosen for system description. For example, proteins can be considered as atomic entities in the network of chemical reactions in a biological cell; neurons are the basic constituents of the brain, when focusing on purely computational issues; each individual can be considered as an atomic entity in an ecosystem or in a social network. These examples of complex systems underline a property frequently found in such systems, concerning the fact the usually complexity arises in the form of a hierarchical organisation, as nested Systems of Systems. From this last point of view, it is possible to consider causal relations between elements belonging to different levels in the hierarchical organisation. When the network of these relations contains a loop, sometimes it is referred to as 'strange loop', i.e. a causal loop between different levels of the hierarchy [35]. This property is strongly related with the emergence of the most interesting behaviours of a given Systems of Systems, when considered as a whole.

In a fundamental paper appeared in 1948 entitled "Science and Complexity" [92], Warren Weaver, one of the fathers of modern information science together with Claude Shannon, proposed a tri-partition of science styles.

Scientific themes can be sub-divided into:

1. Problems of simplicity
2. Problems of disorganised complexity
3. Problems of organised complexity

The first class (simplicity) roughly corresponds to problems that can be solved in terms of differential equations. These 'simple problems' are the ones allowing for a high degree of abstraction (e.g. a planet could be considered an abstract dimensionless 'material point' for sketching general gravitational laws on the pure basis of its mass and distance from the sun).

Problems belonging to the second class (disorganised complexity) allow for a higher degree of generalisation than first class problems without losing in precision. These problems imply a somewhat opposite style of reasoning: the efficiency does not stem from the possibility to get an efficient abstract description of the involved players, but from totally discarding such 'atomic' knowledge in favour of very coarse grain macroscopic descriptors corresponding to gross averages on a transfinite number of atomic elements. This is the case of thermodynamic parameters (e.g. pressure, volume, temperature, etc.). The two above mentioned approaches have drastic limitations of their applicability range: class 1 needs the presence of very few involved players interacting in a stable way with a practically null boundary conditions effect, whereas class 2 needs very large numbers of particles with only negligible interactions among them.

Problems of organised complexity (class 3) arise in all those situations in which many (even if not-so-many as in class 2) elements are involved with non-negligible interactions among them. This is the 'middle kingdom' of complexity, where biological systems live and where computational intelligence and pattern recognition can 'make the difference'.

Network (or graph) is the archetype of organised complexity: a set of nodes (e.g. genes, brain areas, animal species) are each other connected by mutual correlations (edges). The wiring architecture of these graphs can vary in both space and time and it is of utmost importance to get quantitative similarities and differences among them. When graphs are adopted to represent only topological information concerning a set of objects and their relations, the network approach can roughly be described as the answer to the question "what can we derive from the sole knowledge of the wiring diagram of a system?" [28, 58].

The most crucial questions at the frontiers of biomedical sciences demands a reliable answer to the above question. Fields (just to name a few) that are increasing their formalisation in terms of network representations are: neuroscience at both clinical and basic research level [11, 68], biochemistry [5], cancer research [94], structural biology [46], ecology [27].

Moreover, when dealing with fully labelled graphs (where both nodes and edges are associated with possibly structured data), a fundamental topic is how to define proper dissimilarity measure between pairs of such patterns (the graph matching problem [47]).

Modelling a complex system is a matter of identifying the correct level of abstraction, which usually means to extract a hierarchy of information granules, searching for the level of the hierarchy better related to the semantic of the problem at hand. At any level, information granules are nodes of a network, so that the granulation process must deal with the problem of searching for frequent substructures in labelled graphs which, in turn, means to define algorithms able to automatically identify suitable

dissimilarity measures in graph spaces. To this aim granular computing techniques are nowadays a promising tool.

Keeping this general frame in mind, in order to fix clear boundaries to this Chapter, a general definition of computational intelligence and pattern recognition is sketched in the following.

1.2 Theoretical Background and Definitions

Computational Intelligence, formerly known as *Soft Computing* thanks to the seminal work [96], is a set of data processing techniques tolerant to imprecisions, uncertainty, partial truth and approximation (in the data and/or models), aimed to provide robust and low-cost solutions and to achieve tractability when dealing with complexity. Such toolbox includes mostly biologically-inspired algorithms, usually exploiting inductive reasoning (i.e. based on generative logic inferences, such as analogy and induction) [13]. Basically, in this toolbox it is possible to find:

- Artificial Neural Networks
- Fuzzy Logic and Neuro-Fuzzy Systems
- Evolutionary Computation and derivative-free optimisation metaheuristics, such as genetic algorithms and swarm intelligence

Such a (heterogeneous) set of computational tools are usually combined to design powerful data-driven modelling systems. Being able to synthesise a (predictive) model of a given (physical or even abstract) process P is a fundamental topic in all natural sciences, as well as in engineering.

Before the pervasive widespread of digital computing devices, modelling was performed 'by hand', mostly relying on field-experts (*analytical modelling*), consisting in identifying meaningful quantities and relations among them and finally writing a system of integro-differential equations as the final output. This implies a clear understanding of the process at hand to be modelled.

However, when a meaningful sampling S of the process P to be modelled is available, a second approach (*data-driven modelling*) consists in writing an algorithm (often suitable to be run on a Von Neumann computing architecture) able to automatically synthesise a model M of P according to some predefined optimality criteria. This modelling approach is nowadays usually referred to as *Machine Learning*. The design and development of such learning systems is basically an engineering problem.

A formal machine learning definition has been given in [59], where the author considers machine learning as the following, well-posed problem:

> A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

More broadly, machine learning can be defined as a (set of) complex intelligent processing system(s), usually defined by means of adaptive learning algorithms,

able to act without being explicitly programmed or, in other words, able to learn from data and experience.

Pattern recognition techniques fall under the machine learning umbrella, focusing on classification of objects in a given number of categories (classes). Indeed, pattern recognition includes a wide range of techniques employed to solve (properly said) classification problems and clustering problems. Broadly, pattern recognition techniques can generally be divided into two main families: *supervised* and *unsupervised learning*, both of which fall under the aforementioned data-driven modelling paradigm.

For a more formal definition, let us consider an orientated process $P : \mathcal{X} \to \mathcal{Y}$ where $\mathcal{X}$ is the input space (domain) and $\mathcal{Y}$ is the output space (codomain). Moreover, let $\langle x; y \rangle$ be a generic input-output sample drawn from P, i.e. $y = P(x)$. In supervised learning, a finite set S of input-output observations drawn from P are supposed to be known. Common supervised learning tasks can be divided into two families, depending on the output space nature: *classification* and *function approximation*. In classification, outputs take values from a set of categorical labels, each of which correspond to a given problem-related class (e.g. "sick" or "healthy" in a predictive diagnosis/medicine problem). Conversely, in function approximation (such as regression, interpolation, extrapolation, fitting) outputs take values usually in the real field. Formally, in the former case, $\mathcal{Y}$ is a discrete label set where it is not possible to establish any total ordering between its elements, whereas, in the latter case, $\mathcal{Y}$ can be considered as a normed space.

In unsupervised learning there are no output classes or labels and regularities have to be discovered by considering mutual relations between elements drawn from the input space only. One of the mostly acclaimed unsupervised learning approaches relies on data *clustering* [37]. Aim of a clustering algorithm is to discover groups (clusters) of patterns in such a way that similar pattern will fall into the same cluster, whereas dissimilar pattern will fall into different clusters. Formally, let S be a sampling of a non-orientated process P and let c be the number of clusters, constrained to $2 \leq c \leq |S|$. Aim of a clustering algorithm is to assign to every $x \in \mathcal{X}$ an integer $h \in [1, c]$ starting from the set of c clusters induced over S.

In both of these cases, the goal of a learning machine is to build a predictive model from observations, aiming to discover the underlying model structure. Moreover, learning machines must be able to generalise their discrimination capabilities to previously unseen patterns or, in plain terms, they must be able to assign a label (either a class label or a cluster label) to patterns not belonging to S.

For the sake of completeness, it is worth stressing that clustering and classification algorithms might as well co-operate and shall not be considered as two diametrically opposed techniques. For classification purposes, a rather common approach relies on clustering labelled data without considering their respective labels, then assigning a label to each cluster by considering, for example, the most frequent label amongst the patterns belonging to the cluster itself. Finally, each new pattern is classified according to the nearest cluster's label. An example of such workflow can be found in [21, 22].

1.3 Chapter Scope

Aim of this Chapter is to review and discuss major issues when dealing with pattern recognition problems in non-metric spaces, namely input spaces for which a (dis)similarity measure might not be metric. As a case study, bioinformatics and computational biology-related problems will be investigated, since in these fields not only pattern recognition has emerged as a breakthrough discipline, but it is also very common to find structured data such as graphs or sequences which lie in non-metric spaces (see Sect. 1). Moreover, biological processes are excellent examples of complex systems, strongly suggesting the use of granular computing techniques for facing the challenging problem of (data-driven) model synthesis.

In Sect. 2 the data-driven modelling steps at the basis of pattern recognition problems will be described in detail, with particular emphasis on classification and clustering, underlying the role of computational intelligence techniques in designing pattern recognition systems.

Section 3 will regard non-metric spaces, remarking some examples of bioinformatics and computational biology-related problems in which structured data are commonly used. Moreover, some important issues when dealing with pattern recognition in non-metric spaces and possible solutions, including information granulation-based techniques, will be discussed.

In Sect. 4 some real case studies of bioinformatics/computational biology problems faced by means of pattern recognition techniques design to work in structured and non-metric domains will be summarised.

Finally, Sect. 5 will draw some conclusions, stressing major advantages of granular computing-based techniques over more 'traditional' approaches.

2 Machine Learning Systems Design

In conventional machine learning, a *pattern* is defined by a set of measures related to the original object to be represented, arranged in an array. Each entry (*feature*) is usually a real-valued variable. When a metric dissimilarity measure is implicitly or explicitly fixed in order to compare a pair of such simple data structures, usually it is referred to as a *feature vector*. The multi-dimensional space spanned by feature vectors forms the *feature space*. A well-defined feature space is able to facilitate the modelling process. For example, in the classification (supervised) case a well-designed feature space yields simpler decision surfaces in terms of structural complexity (smooth and regular).

Let us consider a plain supervised pattern recognition (classification) problem, as an instance of the more general machine data-driven modelling paradigm. Recalling Sect. 1.2, aim of a classification system is to assign an input pattern (represented by its feature vector) to one amongst the class labels defining the problem at hand.

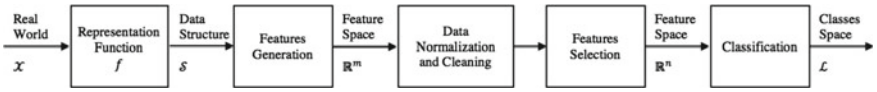

Fig. 1 A simplified pattern recognition system workflow

In Fig. 1, the main steps in order to build a classification system are summarised. First, real-world data, belonging to a generic (and possibly abstract) space $\mathcal{X}$ are casted into a proper data structure $\mathcal{S}$, processable by a computational device, by means of a representation function f which must ad-hoc be chosen for the problem at hand.

From structured data $\mathcal{S}$, a given number m of (usually numerical) features is extracted, thus casting data in $\mathcal{S}$ towards $\mathbb{R}^m$ (the aforementioned feature space).

The two following blocks are not mandatory, but they have been added for the sake of completeness and in order to take into account inevitable uncertainties in data collection and processing. The first block is in charge of data normalisation and cleaning: the former task is sometimes crucial in order to facilitate the classification algorithm under particular circumstances[1]; the latter deals with missing and noisy data. An intuitive data cleaning task is, for example, outliers' removal.[2] Conversely, the Feature Selection block allows to select a significant subset of the previously generated features; indeed, as a general rule, the feature vector should be small, yet informative,[3] in order to avoid undesired phenomena such as overfitting and/or the so-called *curse of dimensionality*. Further, it is recommended to get rid of unreliable features and correlations with existing features. At the end of this selection stage, feature vectors will lie in a (possibly) reduced features space $\mathbb{R}^n$, where $n \leq m$. Finally, the set of feature vectors will be used in order to train the classification system, with the final goal of estimating the correct label (identified, for the sake of ease, as an instance of a nominal value set $\mathcal{L}$ in Fig. 1) for any input vector.

For a better understanding of Fig. 1 and all of its steps, let us consider a real-world, Bioinformatics-related scenario, where $\mathcal{X}$ corresponds to the protein space (i.e. the set of real macromolecules). Let us suppose to represent proteins as graphs (cf. Sect. 3.1), then f is an (hypothetical) function which must convert macromolecules into graphs ($\mathcal{S}$). Fortunately, at least from a machine learning point of view, molecular

[1] For example, let us consider a classification/clustering algorithm driven by the Euclidean distance. A common problem with the Euclidean distance is that features spanning a wider range of values have more influence in the resulting distance measure, therefore normalising all attributes in the same range (usually [0, 1] or $[-1, +1]$) ensures fair contribution from all attributes, regardless of their original range.

[2] In Statistics, *outliers* are "anomalous data" that for a given dissimilarity measure lie far away from most observations.

[3] *Non sunt multiplicanda entia sine necessitate (Entities are not to be multiplied without necessity)*, commonly known as "The Ockham's Razor" Criterion (William of Ockham, circa 1287–1347). This criterion states that among a set of predicting models sharing the same performances, the simplest one (i.e. the one with the simplest decision surfaces) should be preferred. It is for sure one of the fundamental axioms for thoughtful and practical data-driven modelling.

biology helps: 3-dimensional protein structures, mainly gathered by crystallography, are available in online databases (e.g. Protein Data Bank [7]), therefore it is rather easy to build graph-based protein representations, either labelled or unlabelled on nodes and/or edges. The Features Generation block is in charge of extracting numerical features from graphs in $\mathcal{S}$ (cf. Sect. 3.2.1) which, after possible further processing, will be directly fed into the classification/clustering system.

The training phase for a classification system is a rather delicate task and it needs a separate discussion. Indeed, thanks to the training phase, the classification system learns how to map and discriminate input patterns according to their class labels. In other words, it learns the decision surfaces (decision regions boundaries) which separates patterns corresponding to different classes.

A usual procedure for measuring in a fair way the generalisation capability of a classification model consists in splitting the entire available dataset into two non-overlapping subsets, namely the Training Set and the Test Set. Specifically, as far as classification tasks are concerned, one shall figure both Training and Test Sets as composed by $\langle x; y \rangle$ pairs (see Sect. 1.2). The classification system, driven by a training algorithm which strictly depends on the chosen model (e.g. Support Vector Machine, Artificial Neural Network, K-Nearest Neighbours), will use the Training Set in order to learn the input-output mapping. The Test Set will then be used on such trained model, without further adaptive changes, in order to compute its performances (e.g. percentage of correctly classified patterns). For a thoughtful modelling, the two sets (albeit distinct) should satisfactorily represent the same statistical properties of the process to be modelled.

This double-split procedure, however, is not effective since every training algorithm depends on a set of parameters,[4] which must be tuned with the ultimate goal of maximising the generalisation capability of the synthesised model. In order to find the optimal set of hyperparameters (i.e. model selection) a three-split procedure is usually employed: the whole dataset is split into three non-overlapping parts, namely Training Set, Validation Set and Test Set. The training algorithm, driven by the set of hyperparameters Γ, will again exploit the Training Set and its performances will be evaluated on the Validation Set. The parameters Γ will be tuned in order to maximise the performances on the Validation Set and once the optimal $\Gamma^\star$ has been found, the final performances will be evaluated on the Test Set.

In literature, several ways to perform the aforementioned search for $\Gamma^\star$ have been proposed, amongst which grid search, random search [6] and evolutionary optimisation-based techniques emerge (see Sect. 2.1).

When dealing with unsupervised learning, the scheme reported in Fig. 1 does not change significantly, apart from the rightmost block. Indeed, rather than feature a Classification algorithm, a Clustering algorithm must be placed instead. A clustering algorithm is in charge of returning groups of data (clusters) according to a given (dis)similarity measure and to a predefined objective function.

In literature, three main families of clustering algorithms can be found, which mainly differ for their objective function (i.e. according to which criterion clusters

[4] Also known as *hyperparameters* in the Machine Learning terminology.

should be discovered): *partitional clustering* (e.g. k-means [51, 52], k-medians [10], k-medoids [39]), which split the dataset into k non-overlapping partitions; *hierarchical clustering* (e.g. BIRCH [97], CURE [32]), where clusters are found by building a dendrogram in either top-down or bottom-up approach; *density-based clustering* (e.g. DBSCAN [24], OPTICS [3]), which detect clusters as the most dense regions of the dataset.

Clustering algorithms do need some parameters tuning as well. Selecting their respective optimal value(s) can be done according to some internal validation measures, such as the Silhouette [74] or the Davies-Bouldin Index [19]. Both manual or fully automatic tuning by means of evolutionary optimisation techniques can be employed in unsupervised learning as well.

2.1 Evolutive and Fully Automatic Approaches

Evolutionary optimisation metaheuristics such as genetic algorithms [30], particle swarm optimisation [40], ant colony optimisation [16] and simulated annealing [41], are one of the main topics under the Computational Intelligence umbrella (Sect. 1.2). Such metaheuristics are well suited when the objective function to be optimised is not known in closed-form and gradient-based methods turn to be unfeasible.[5] Indeed, the decision boundary which separates two or more classes in a classification problem is determined thanks to a sampling of the boundary itself, namely the set of patterns which compose the dataset at hand, along with their respective class labels. As introduced in Sect. 2, they are often used in order to automatise the hyperparameters' tuning for classification and/or clustering algorithms. Further, they can help in conducting the feature selection phase (see Fig. 1). Indeed, one might ask which is the most relevant set of features in order to maximise the classification and/or clustering performances. To this end, evolutionary optimisation metaheuristics play a huge role.

Let us consider a genetic algorithm as an example. One can consider the genetic code to have the form $[\Gamma, \mathbf{w}]$ where Γ, as in Sect. 2, is a set of hyperparameters for the clustering/classification algorithm at hand, whereas $\mathbf{w}$ is an m-length real valued vector which tunes the (dis)similarity measure, core of the algorithm itself.

As far as classification tasks are concerned, a typical workflow might consist in letting each individual in the evolving population to be considered for training the classification model on the Training Set using both the hyperparameters and the (dis)similarity weights specified by its genetic code. The classification model's performance will later be evaluated on the Validation Set and such performance will serve as (part of) the fitness function.[6] Trivially, at the end of the evolutionary

[5] That is why evolutionary optimisation metaheuristics fall within the *derivative-free* methods.

[6] A common choice for a genetic algorithm fitness function takes into account both the model performance and its structural complexity. Specifically, whilst the former should be maximised, the latter should be minimised in order to avoid overfitting (cf. the Ockham's Razor Criterion).

stage, the best individual will be the one which maximises the performances on the Validation Set and its final performances will be evaluated on the Test Set. An example of such workflow can be found in [55] for classification algorithms and in [21, 22] for re-adaptation of clustering algorithms for classification purposes.

When dealing with clustering algorithms, the overall workflow does not change significantly. However, each individual will process the entire dataset according to the parameters stored in its genetic code and, similarly, since the performances cannot rely on any ground-truth labels, other internal validation measures should be used as the fitness function. An overview of clustering with evolutionary-driven feature selection can be found in [1].

It is worth stressing that, in both cases, the resulting best individual's genetic code contains the set of hyperparameters $\Gamma^\star$ which, along with the weights vector, maximise the algorithm's performances. Specifically, the latter deserves some further notes: if one considers $\mathbf{w} \in [0, 1]^m$, such vector acts as a feature selector, where 0's correspond to features which will not be considered in the (dis)similarity measure, and 1's correspond to features which, conversely, will be considered. The subset of n elements for which $\mathbf{w}$ is not-null can be seen as the reduced features space.

3 Dealing with Non-metric Spaces

So far, the design of a pattern recognition system has been described in its standard and most common form, where patterns are represented by means of real-valued vectors. In these cases, any Minkowski-based (e.g. Euclidean) distances can be good and straightforward candidates. Moreover, such (dis)similarity measures are metric.

Formally, a dissimilarity measure d defined on a generic space $\mathcal{S}$ is a function $d : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ satisfying the following properties:

1. $$\exists d_0 \in \mathbb{R} \text{ such that } -\infty < d_0 \leq d(x, y) < \infty \tag{1}$$
2. $$d(x, x) = d_0 \tag{2}$$
3. $$d(x, y) = d(y, x) \tag{3}$$

for any two objects $x, y \in \mathcal{S}$. If, alongside Eqs. (1)–(3), d satisfies the following two properties

1. $$d(x, y) = d_0 \text{ if and only if } x = y \tag{4}$$
2. $$d(x, z) \leq d(x, y) + d(y, z) \tag{5}$$

for any three objects $x, y, z \in \mathcal{S}$, then d is said to be *metric*.

Similarly, in $\mathcal{S}$ it is possible to define a similarity measure $s : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ whether it satisfies the following properties:

1.
$$\exists s_0 \in \mathbb{R} \text{ such that } -\infty < s(x, y) \leq s_0 < \infty \tag{6}$$
2.
$$s(x, x) = s_0 \tag{7}$$
3.
$$s(x, y) = s(y, x) \tag{8}$$

for any two objects $x, y \in \mathcal{S}$. If, alongside Eqs. (6)–(8), s satisfies the following two properties

1.
$$s(x, y) = s_0 \text{ if and only if } x = y \tag{9}$$
2.
$$s(x, y) \cdot s(y, z) \leq (s(x, y) + s(y, z)) \cdot s(x, z) \tag{10}$$

for any three objects $x, y, z \in \mathcal{S}$, then s is said to be *metric*.

Moreover, it is possible to prove that:

Theorem 1 *If d is a metric dissimilarity measure with $d(x, y) > 0, \forall x, y \in \mathcal{S}$, then $s = a/d$ is a metric similarity measure for $a > 0$.*

Theorem 2 *If d is a metric dissimilarity measure, let d_{max} be the maximum pairwise distance between elements in $\mathcal{S}$, then $s = d_{max} - d$ is a metric similarity measure.*

The above two theorems demonstrate that, under particular circumstances, one can easily 'switch' between (metric) similarity and dissimilarity measures in a given input space. Indeed, dissimilarity measures quantify the degree of separation, whereas similarity measures estimate the complementary notion of closeness.[7]

3.1 Examples of Structured Data in Bioinformatics and Computational Biology

Dealing with non-metric spaces is a common issue when unconventional (structured) data, such as graphs or sequences, are considered as the input domain.

As introduced in Sect. 1, especially in bioinformatics and computational biology, patterns are usually described by means of data structures more complex than plain real-valued feature vectors: some common examples include proteins, DNA and RNA, metabolic pathways and brain connectivity networks.

[7] That is why in most of the Chapter, unless explicitly specified, the generic term *(dis)similarity* will be used.

Indeed, DNA and RNA transcripts are usually described as sequences of 4 possible nucleotides: adenine (A), cytosine (C), thymine (T), guanine (G) for DNA and adenine (A), cytosine (C), uracil (U), guanine (G) for RNA.

Proteins can be described by both sequences and graphs. The former representation is more straightforward: a protein is encoded in genes (DNA sequence) which is transcribed into pre-messenger RNA (RNA sequence). The RNA transcript is loaded into the ribosome which reads three nucleotides at the time (codons) and converts each triplet into one of the 20 amino-acids. It is clear that there exist up to three sequence-based protein representations, which mainly differ from their alphabet (4 nucleotides vs. 20 amino-acids) and their length (nucleotide-based sequences are three times longer than amino-acid-based ones). The protein representation as a sequence of amino-acids is also known as *primary structure*.

Graph representations result from a biological step forward in protein biosynthesis. Indeed, when the protein leaves the ribosome, a process called *protein folding* starts, during which the protein folds on itself, leading to a unique three-dimensional structure (also known as the *tertiary structure*). Protein Contact Networks [23] are an example of graph-based protein representation [29], where nodes correspond to amino-acids and edges between any two nodes exist whether their Euclidean distance falls within a given range, typically [4, 8]Å (e.g. [44–46, 54, 55]). The lower bound is usually considered in order to discard trivial backbone first-order neighbour contacts (i.e. sequence proximity), whereas the upper bound is usually defined by taking into account the peptide bonds geometry; indeed, 8Å roughly correspond to two peptide bond lengths or, equivalently, to two Van der Waals radii between residues' alpha-carbon atoms. In their original formulation, Protein Contact Networks are undirected graphs with no labels on nodes and edges: information regarding the type of amino-acid and their respective proximities are deliberately discarded in order to focus on proteins' topological structure and their complex nature.

Metabolic pathways are mainly described by graphs as they can be seen as protein networks and chemical networks. In the former, nodes correspond to proteins, whereas links correspond to physical (protein-protein interaction) and/or functional relations between them. In the latter, links correspond to chemical reactions (catalysed by specific enzymes) transforming the nodes (organic molecules produced – or used – in the metabolic processes) at their extremities into one another.

To our knowledge, the brain is probably the most complex circuit in the Universe, a complex system of nested subsystems, usually modelled as a network, since its functions strictly depend on the anatomical and functional wiring of billions of neurons [11, 31, 75, 88].

While in the case of brain networks based on the anatomical links between parts of the brains (macroscopic scale) or between single neurons in a small brain portion (microscopic scale) it is possible to rely on the assumption of a certain degree of invariance in time,[8] this is not the case as for functional brain networks (e.g. related to areas metabolic activity correlations observed by Nuclear Magnetic Resonance

[8]Indeed, the anatomical structure changes in the order of months/years depending on the age of subjects.

(NMR) or Positron Emission Tomography (PET)) that modify their wiring patterns on very short time scales [25, 81, 83].

Spontaneous neuronal activity in resting state depends on dynamic communication between brain regions allowing both local segregation and long-distance integration of neuronal processes. Several functional networks in which temporally or spatially coherent connections exist [18]. These networks have been identified in healthy subjects by functional Magnetic Resonance Imaging (fMRI) and by PET, respectively. Both these techniques deal with the quantification of metabolic rate correlation across different brain areas. Specifically, fMRI measures as 'marker' the variation of the amount of blood flowing across brain areas (coupled with metabolism by the dynamics between oxidised and reduced haemoglobin) [26], whereas PET focuses on the different metabolic rate of glucose (the most important energy source for brain cells) across different brain areas [79].

Both fMRI and PET techniques define a brain connectivity network in correlative terms: two nodes i, j of the network are linked by an edge if the metabolic rates of nodes i and j are each other correlated (given the quantitative character of the measures used by Pearson correlation coefficient metrics).

3.2 Pattern Recognition in Non-metric Spaces

When dealing with complex data structures such as graphs or sequences, the scheme from Fig. 1 should be revisited since patterns cannot be directly described by means of real-valued vectors.

In literature, three major approaches can be found [49, 50]:

1. directly working in the input data structure space, by defining ad-hoc (dis) similarity measures
2. by means of kernel transformations and kernel machines
3. by defining an embedding function from the input space to real-valued vectors

These approaches are summarised in Fig. 2 and, along with the 'classical' Feature Generation procedure, will be discussed separately.

3.2.1 Classical Processing Chain by Feature Generation

Recalling Sect. 2, given a generic and possibly non-metric input space $\mathcal{S}$, the most straightforward approach consists in defining a mapping function $\phi : \mathcal{S} \to \mathbb{R}^n$ specifically designed for the input space at hand. In this section, three examples of mapping function suitable for dealing with graphs will be described. Moreover, the additional challenge of dealing with patterns of different size in $\mathcal{S}$ will be discussed. For the sake of argument, let us consider graphs representing proteins since, notably, proteins have different sizes both in terms of primary and tertiary structures, meaning that

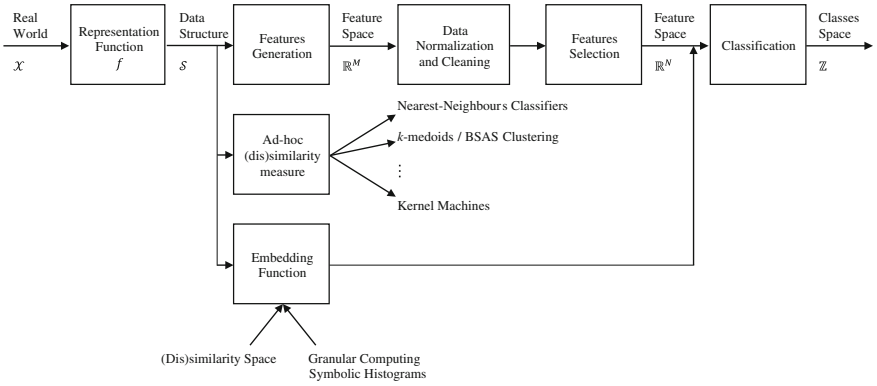

Fig. 2 Overview of possible approaches for pattern recognition in non-metric spaces

their amino-acids sequences and, by extension, their folded 3-dimensional structures have different size.

In [46, 54, 55] a mapping function based on graphs spectra has been proposed. Specifically, each graph has been described by means of its normalised spectrum, i.e. the set of eigenvalues evaluated from its corresponding normalised Laplacian matrix [38]. Such eigenvalues lie in range [0, 2], making such approach suitable for comparing graphs with different sizes. However, the number of eigenvalues composing the spectrum equals the number of nodes in the graph and, in order to overcome this problem, it is possible to estimate the spectral density by means of a kernel density estimator [63]. In this way, the distance between any two graphs can be evaluated by integrating the squared difference between their respective spectral densities all over the [0, 2] range. This evaluation can be performed also in the discrete domain by sampling a finite number (n) of points from such spectral densities (being the support domain equal for all graphs, regardless of their respective sizes). In such finite domain, the distance between two graphs can be evaluated as the considered distance (e.g. Euclidean) between their respective sets of samples.

In [45] a feature-engineering based approach has been employed in order to predict proteins' solubility starting from their topological structures. Several features have been manually selected such as the number of nodes and edges, the number of protein chains, some centrality measures (e.g. closeness and degree) and some physical characteristics (e.g. heat trace, energy). The union of these features forms the feature vector for a given graph.

Other feature extraction procedure(s) can rely on a rather novel field known as Topological Data Analysis [12, 91]. Topological Data Analysis consists in a set of techniques in order to extract information from data starting from topological information by means of dimensionality reduction, manifold estimation and persistent homology in order to study how components lying in a given multidimensional space are connected (e.g. in terms of loops and multidimensional surfaces). This

can be done by starting either by so-called *point clouds*[9] or by explicitly providing a similarity matrix (cf. the *Kernel Methods* paragraph). Albeit this field has very solid and rigorous foundations (from algebraic topology to pure mathematics), there are very few 'numerical' features which can be extracted, mainly the sequence of Betti numbers. Formally, the ith Betti number corresponds to the rank of the ith homology group. In plain terms, the ith Betti number corresponds to the number of i-dimensional 'holes' in a topological surface. For example, let us consider a three-dimensional graph, its first three Betti numbers have the following interpretations: the 0-th Betti number corresponds to the number of connected components in the graph; the 1-st Betti number corresponds to the number of 1-dimensional holes (e.g. circular holes); the 2-nd Betti number corresponds to the number of 2-dimensional holes (e.g. cavities). If the multidimensional space under analysis has a finite dimension, the Betti numbers vanish after the spatial dimension (e.g. the number of 4-dimensional holes in a 3-dimensional space is always equal to zero). Whether the Betti numbers can be an effective mapping function for pattern recognition purposes it still an open question.

3.2.2 Ad-Hoc Dissimilarities in the Input Space

One of the mostly acclaimed ad-hoc (dis)similarity measures for structured data are the so-called *edit distances*, according to which the distance between two objects is given by the minimum number of atomic edit operations (usually insertions, deletions and substitutions of elements in the sequence) needed to transform the first object into the second object. As regards strings, the Levenshtein distance [42] is the seminal example of an edit distance, which can be seen as a generalised Hamming distance[10] [33].

The same approach can be used to define dissimilarity measures between graphs as well, leading to the Graph Edit Distances [47, 60], which inherit the idea at the basis of the Levenshtein distance, defining atomic edit operations in both the sets of nodes and edges. In many pattern recognition applications defined in sequences domains the Dynamic Time Warping [76] can be adopted, where the sequence support is explicitly related with time. Specifically, by applying a non-linear distortion on the support independent variable (time), it returns the optimal correspondence (i.e. similarity) between two sequences.

Amongst these methods, the Levenshtein/Hamming distances are well-known to be metric; the same might not be true for Graph Edit Distances (as they might violate the symmetry property – cf. Eqs. (3) and (8)) and Dynamic Time Warping (as it

[9] A finite set of points equipped with a notion of distance in a finite multidimensional space.

[10] According to which the distance between two strings of equal length is given by the number of mismatches.

might violate the triangle inequality – cf. Eqs. (5) and (10)). Also, edit distances are not recommended if patterns have a high dimension variability as deletion/insertion costs can easily prevail over substitution costs.

On the plus side, however, methods based on ad-hoc (dis)similarity measures notably work in cases where the pattern recognition system does not need to define an algebraic structure on the input space. For example, let us consider a clustering task to be performed directly into a non-metric space with an a-priori chosen (dis)similarity measure. Algorithms such as k-means or k-medians cannot be considered as good candidates since the former needs to evaluate the component-wise mean amongst the pattern in a given cluster in order to evaluate its representative, whereas the latter needs to evaluate the component-wise median. Therefore, the need to define a meaningful algebraic structure emerges which, however, turns into a non-sense as concerns non-metric input spaces. Suitable clustering algorithm candidates for dealing with non-metric spaces are k-medoids, as discussed in [56], and BSAS [85], since they do rely on (dis)similarity measures only in order to form clusters and to update their representatives. Similarly, as far as classification algorithms are concerned, a good candidate is the K-Nearest Neighbour, since it classifies patterns according to their respective distances rather than defining operators such as the inner product, mandatory in Artificial Neural Networks, or Support Vector Machines, whether equipped with an ad-hoc kernel transformation (see the *Kernel Methods* paragraph).

Kernel Methods

Typically, kernel methods can safely be employed whether the input space has an underlying Euclidean geometry, since they are based on inner products. Given a pair of patterns $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, their inner product is given by:

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{n} x_i \cdot y_i \tag{11}$$

Further, let us consider the instances matrix for the dataset at hand, $\mathbf{X} \in \mathbb{R}^{N_P \times n}$, namely a matrix where each row corresponds to a given pattern. Let N_P indicate the number of patterns. It is possible to define the *kernel matrix*[11] as

$$\mathbf{K}_{i,j} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle \tag{12}$$

or, in batch fashion

$$\mathbf{K} = \mathbf{X} \cdot \mathbf{X}^T \tag{13}$$

[11] Also known as the *Gram Matrix*, after Danish mathematician Jørgen Pedersen Gram.

More in general, let k be a symmetric and positive semi-definite kernel function from the input space at hand $\mathcal{S}$ towards $\mathbb{R}$, i.e. $k : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}$ such that:

$$k(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_j, \mathbf{x}_i) \qquad \forall \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X} \tag{14}$$

$$\sum_{i=1}^{N_P} \sum_{j=1}^{N_P} c_i c_j k(\mathbf{x}_i, \mathbf{x}_j) \qquad \forall c_i, c_j \in \mathbb{R}, \forall \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X} \tag{15}$$

As in the inner product case, starting from $k(\mathbf{x}_i, \mathbf{x}_j)$ one can easily evaluate the Kernel Matrix as

$$\mathbf{K}_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j) \tag{16}$$

and if $\mathbf{K}$ is a positive semi-definite kernel matrix, then k is a positive semi-definite kernel function. One of the most intriguing property of kernel methods relies in the so-called *kernel trick* [77]: kernel of the form (14)–(15) are also known as *Mercer's kernels* since they satisfy the Mercer's theorem [57]; they can be seen as the inner product evaluation on a (possibly) infinite-dimensional and usually unknown Hilbert space $\mathcal{H}$. The kernel trick is usually defined by means of the following, seminal equation:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle \psi(\mathbf{x}_i), \psi(\mathbf{x}_j) \rangle_{\mathcal{H}} \tag{17}$$

where $\psi : \mathcal{S} \rightarrow \mathcal{H}$ is the implicit and usually unknown mapping function.

Several positive semi-definite functions commonly used as kernels include the linear, exponential, radial basis function and polynomial [77, 78], which are usually employed in kernel machines, such as (non-linear) Support Vector Machines.

However, in many cases, defining the kernel function might not be easy, especially when dealing with non-metric spaces. Regardless of the nature of the input space, it is possible to evaluate the similarity matrix (cf. Sect. 3) $\mathbf{S} \in \mathbb{R}^{N_P \times N_P}$ where

$$\mathbf{S}_{i,j} = s(\mathbf{x}_i, \mathbf{x}_j) \tag{18}$$

If s is a metric similarity measure, it is possible to directly use $\mathbf{S}$ as the kernel matrix, as suggested in [15], or include similarities in widely-known kernel functions (e.g. radial basis function), as suggested in [77].

Conversely, if the (dis)similarity measure is not metric, two mainstream approaches can be followed. The former relies on moving the pattern recognition problem towards a dissimilarity space (as explained in the next paragraph), the latter relies on 'modifying' the similarity matrix in order to be a valid kernel matrix (i.e. satisfying Mercer's theorem) [14, 15, 89].

Embedding Functions and Information Granulation

Embedding functions can be seen as particular cases of mapping functions as defined in the *Classical processing chain by Feature Generation* paragraph. Indeed, while both of them aim at moving the problem from a generic input space $\mathcal{S}$ towards

$\mathbb{R}^n$, embedding functions, at least in this context, do rely on other patterns or on substructures extracted from the dataset at hand in order to build such mapping.

A first example of embedding consists in moving the pattern recognition problem into a *dissimilarity space* [66].

In turn, dissimilarity representations can follow two further approaches:

1. Each pattern is described by its own row[12] from the similarity matrix **S** (cf. the *Kernel Methods* paragraph); that is, each pattern is described by the distance(s) vector with respect to other patterns (including self-distance)
2. Each pattern is described by the distance(s) vector with respect to a given number of representatives drawn from the input space at hand. Certainly, the selection of such representatives is a crucial task since a) they must well-characterise the decision boundary between patterns in the input space and b) there should be few of those since the number of representatives has a major impact on the model complexity. Representatives selection heuristics range from class-aware random selections to clustering procedures directly in the input space [48] (cf. the *Ad-hoc Dissimilarities in the Input Space* paragraph).

Regardless of which of the two methods is employed, a dissimilarity space can be equipped with algebraic structures and operators, such as the inner product, in order to be suitable with traditional kernel methods [48]. But, more in general, since patterns are now casted in $\mathbb{R}^{N_P}$ (former case) or $\mathbb{R}^R$ (latter case – where R indicates the number of representatives), any "standard" pattern recognition algorithm can be used.

In order to introduce the embedding by means of substructures, let us introduce widely known embedding functions for sequences. Since sequences are finite collections of objects drawn from a finite alphabet (cf. RNA/DNA sequences or proteins' primary structure, Sect. 3.1) one of the most intuitive approaches relies on histograms. Indeed, a sequence can effectively be described as the number of occurrences of any alphabet symbol within the sequence itself. For 'simple' sequences such as nucleotides or amino-acids sequences, histograms defined as above suffice. For example, in [90] a double experiment has been proposed in order to classify proteins starting from their primary structure according to their physiological role; in a first experiment, each protein is described by the number of occurrences of each amino-acid within the primary structure and, in a second experiment, such histogram-based representation has been extended to triplets of amino-acids in order to take into account also information about proximity and ordering. Further, in [53], the histogram-based representation considers pairs of amino-acids whose distance along the protein backbone is within a minimum and maximum value, a-priori defined.

For more complex sequences such as sentences or entire text documents, bag-of-words and word-count models have been proposed, where the alphabet is composed by the set of unique words in the sentence or document. 'Complex sequences' such

[12] If the similarity measure at hand is not symmetric, patterns' distance vectors as taken by rows or columns will be different. In order to overcome this problem, one can 'force' a similarity measure to be symmetric by considering $\mathbf{S} := (\mathbf{S} + \mathbf{S}^T)/2$ (e.g. [14]).

as sentences or entire text documents are rather rare (if non-existent altogether) in bioinformatics as such, but bag-of-words models, along with statistical and/or machine learning techniques, have been successfully employed for health analysis and forecasting (e.g. [82] for anastomosis leakage detection, [93] for diabetes-related notes in electronic health records).

In the last years, granular computing [4] emerged as a novel and promising information processing paradigm. In granular computing, atomic quantities known as *information granules* have to be extracted in order to be further studied and analysed, for gathering useful knowledge and insights from data, but finding the adequate level of abstraction for the problem at hand might be a challenging task. Along with symbolic histograms, granular computing can play the role of a promising data-driven framework which can simultaneously deal with embedding functions in non-metric spaces and knowledge discovery. In [8, 9, 69, 72] have been proposed fully automated data-driven and granular computing-based classification systems both for graphs and sequences. These systems are composed by four main macroblocks: motifs extractor, granulator, embedder and classifier.

The motifs extractor is in charge of extracting, according to some heuristics (possibly exhaustively), substructures (i.e. subgraphs/subsequences) from the dataset at hand.

The set of motifs is then forwarded to the granulator which runs a clustering algorithm on it, relying on a suitable inexact matching procedure (i.e. on a given dissimilarity measure in the substructures space), yielding a set of frequent sub-structures (clusters), whose representatives can be considered as candidate information granules (symbols). It is worth stressing that the clustering algorithm works in the input space since motifs are frequent substructures, and that free-clustering algorithms such as BSAS should be preferred, in order to automatically return a suitable number of clusters, avoiding to set it is advance. Further, since the input space might not be metric, a suitable cluster's representative is the medoid (or MinSoD) [20, 56].

The set of information granules are the main input for the embedder block which, according to the *symbolic histograms* approach, maps each pattern into an integer-valued vector. Specifically, each pattern is represented as the number of occurrences of each information granule within the pattern itself. The embedder, therefore, returns a set of vectors which can feed any standard pattern recognition algorithm for classification or clustering purposes.

The whole cascade is driven by a genetic algorithm, following the workflow as described in Sect. 2.1, in order to maximise the classifier's performances. The genetic algorithm acts as an orchestrator, and is in charge of optimising the final classifier synthesis, accomplishing two tasks: under an algorithmic point of view, it automatically tunes the clustering algorithm and possible (dis)similarity measure parameters, maximising the classifier's performances and selecting the subset of information granules better related with the classification task at hand (cf. Feature Selection block in Fig. 1); under a knowledge discovery point of view, since it returns the (sub)optimal set of information granules for the problem at hand.

The latter deserves some further observations. It is clear that embracing a granular computing/symbolic histograms approach is more computationally expensive

than any other technique discussed so far. Indeed, the embedding procedure requires a clustering phase, searching for candidate granules. Even for small datasets, an exhaustive approach for the extractor might be unfeasible (since its complexity is combinatorial with respect to the pattern size and the substructure order), and it must be replaced by a stochastic approach (random subsampling). Moreover, when dealing with sequences or graphs, the (dis)similarity measure adopted by the core clustering procedure is by far more computationally demanding with respect to Euclidian distance performed on plain real-valued vectors. Furthermore, the selection of the most informative information granules, as well as of the best (dis)similarity measure parameters, demands additional computational burden by the evolutionary optimisation, since for every candidate solution it is needed to launch a full classification model synthesis procedure (for example a Support Vector Machine) in order to evaluate its fitness, computed as the performance of the classifier on the Validation Set (cf. Sect. 2.1). For these reasons, the symbolic histograms approach is practically feasible only when relying on parallel/distributed computing software/hardware environments.

But, on the plus side, granular computing-based techniques unleash an invaluable potential thanks to information granules. Indeed, if the training procedure yields a classification model with satisfying performances, able to correctly discriminate the input patterns for the problem at hand, the resulting information granules subset brings useful knowledge on the problem at hand, since information granules are at the basis of the embedding feature space. Information granules selected by the evolutionary optimisation are therefore responsible for the final definition of decision surfaces in that space and, consequently, they can show useful information that can be exploited by field-experts. This is the main advantage of granular computing techniques with respect to competitive approaches: extracting automatically meaningful information granules is useful both under an algorithmic point of view and under the application field point of view (biology, in this case).

As a more concrete example, let us consider a metabolic pathways problem, where metabolic pathways are described by graphs as in Sect. 3.1. One of the information granules might be the citric acid cycle.[13] The Krebs cycle (in network terms, a motif with a set of nodes lined to form a closed loop) is driven by oxygen and therefore it might be a key granule in order to discriminate between aerobic and anaerobic organisms. For this example a well-known chemical reaction has been considered, but the opposite might also happen: indeed, information granules can *pose* questions other than *confirm* statements: why these information granules are considered as significant for the discrimination/classification problem at hand?

[13] Also known as the *Krebs cycle*.

4 Case Studies and Applications

In most of the cases introduced in Sect. 3.1, it is almost impossible to project the analysed objects into a proper metric space spanned by a shared set of descriptors without considering some global features (e.g. classical network invariants like degree, characteristic length, closeness centrality, etc.) and thus losing a considerable part of information linked to 'who-is-connected-with-whom'. On the contrary, such information can be easily recovered projecting the objects into a non-metric space defined by motifs and/or frequent substructures (Sect. 3.2.2).

The need of a non-metric approach is evident in many biologically relevant cases. This need not necessarily derive by the lack of a common feature space, but it is motivated by the importance to individuate particular motifs endowed with a meaningful semantics.

In the field of protein sequence analysis this is the case of the identification of 'natively unfolded' tracts. This is a particularly intriguing problem in structural biology [87]. Until the end of last century, the general view of structure/function relation in protein molecules was apparently straightforward (cf. Sect. 3.1): protein primary structures correspond to the amino-acid residues linear ordering along the sequence. The primary structure determines both the mutual position of nearby (secondary structure) and distant along the sequence amino-acid residues (tertiary structure). The specific 3-dimensional arrangement of the protein molecule in turn determines its physiological role [70]. This view was questioned some years ago [87] by the discovery of 'natively unfolded' proteins that are molecules that do not have a definite 3-dimensional structure but that, on the contrary, remain in a random coil state until they interact with some partners (e.g. other proteins) and, after the binding, assume a specific 3-dimensional configuration. The same natively unfolded protein (and thus with only one specific sequence) can assume completely different 3-dimensional structures (and functions) depending on the different partners it interacts with. All the vital functions of a cell are managed by the creation of aggregates of different proteins generating a sort of nano-machine performing a specialised task (e.g. energy production, biosynthesis, immune response, DNA repair and duplication, etc.), where natively unfolded proteins are the 'hubs' of such protein-protein interaction networks, given their ability to change structure 'on demand' and thus to participate to different nano-machines (protein aggregations) [80]. Besides proteins that are natively unfolded in their entirety, all the proteins do have (smaller or longer) tracts that are natively unfolded corresponding to their interaction sites. If the goal is to modify the behaviour of a protein aggregate for a therapeutic intervention (e.g. by a drug binding to the protein molecule) it is of utmost importance to recognise such natively unfolded parts of the molecule from their sequence.

This is a very challenging task for classical machine learning approaches, due to the following reasons:

1. The context dependence of the problem: the same subsequence can be natively unfolded in protein *A* and perfectly folded in protein *B* due the general properties of the entire protein molecule [67]

2. The ambiguous character of the definition of 'unfolding': many of the so-called unfolded proteins (or tracts) could be only highly flexible systems that have only one preferred fold without structuring on-demand [34]
3. The dependence on the chemico-physical micro-environment the protein experiences (i.e. pH, molecular crowding, etc.) deciding the disordered/ordered condition [34]
4. The highly variable length of the disordered patterns [34]

This is why (even if never defining explicitly in these terms) all the tentative solutions of the problem used non-metrics approaches that in turn allowed to both select some 'relatively context independent unfolded motifs' and individuating some regularities in these motifs [73].

A somewhat related problem is to predict the relative solubility in water of protein molecules. Again, there exist a similar context dependences of the disordered/ordered case and, in [44], the problem was approached by considering several different representations. The protein folding problem has interested biologists for many years: if the native protein structure is 'encoded' in its primary structure, is it possible to predict its folded state? Relative solubility in water is the major feature for proteins' folding propensity. However, some proteins spontaneously fold, whereas other proteins need so-called *chaperones*[14] in order to fold correctly.

Recall from Sect. 3.1 that a protein can be described in different ways by either taking into account its primary or tertiary structure; therefore in [44] a subset of the *Escherichia Coli* proteome has been considered in three different representations: the plain primary structure; an 'extended' Protein Contact Network representation (cf. Sect. 3.1) where labels exist on both nodes and edges (nodes labels correspond to one of the 20 amino-acids, edges labels correspond to the Euclidean distance between the two vertices at their extremities); a serialised version of the graph-representation, where each vertex is associate with a 3-dimensional real-valued vector derived from the graph transition matrix. The goal was to predict the relative water solubility of each protein *in vitro* (i.e. without the help of chaperones). Given that water solubility encompasses the ability to reach of a correctly folded structure, this prediction task can be considered as an explorative study in the chemico-physical drivers of folding process.

The different representations allowed us to grasp different aspects of 'relative folding propensity' of proteins, being the extended Protein Contact Network the most promising representation.

The impossibility to design a data set on a shared feature space (and consequently the need of non-metric approaches) is evident in neuroscience in the case of comparing different brain connectivity networks [88]. Recall from Sect. 3.1, both fMRI and PET outputs are images of the brain: a quantitative value is attached to each voxel corresponding to the entity of metabolism activity in that location. The voxels are in the order of tens of thousands and their actual quantitative value is not rele-

[14]Protein molecules driving the folding of other protein systems.

vant per-se[15]: what differentiates healthy and pathological subjects is the degree of organisation (correlation among areas) of the system. The selective breakdown of intrinsic brain networks during the progression from the healthy state to mild cognitive impairment to Alzheimer's disease has been observed using both fMRI and PET. Using the single voxels as nodes can be highly misleading in the comparison of images across patients: not only their high number produces networks very difficult to analyse but the pairing of the voxels across different subjects (i.e. to recognise that the j-th voxel of patient A corresponds to the j-th voxel of patient B) is virtually impossible. To solve the problem anatomical knowledge is considered: the physician segments the brain image into ROIs (Region of Interest) correspondent to the well-known anatomical areas of the brain (e.g. hippocampus, amygdala, cerebellum, etc.) that all patients do have, so ROIs become the nodes and edges correspond to the scoring of a strong correlation between pairs of ROIs.

Alzheimer's disease risk scales with the progressive disruption of 'long range' correlations in favour of 'small scale' correlation between nearby areas [62]. This implies that for discriminating different risk levels it is not possible to rely on shared 'global correlation measures' on the brain, nor on the focusing on 'specific relations' between key areas because they can be very different across different patients, while maintaining the above described pattern of 'decrease in long range and increase in small range correlations'. This situation is solved by non-metric approaches, in which different brain connectivity networks are compared on the basis of the dynamics of 'attachment'-'detachment' from the giant component of the network (the bulk of connected ROIs) on a subject by subject basis [61].

In the case of brain connectivity studies, computational intelligence is having a great expansion and the search for suitable context-dependent metrics for comparing different conditions is highly debated in both clinical and basic research communities.

5 Conclusions and Future Directions

In synthesis, we can surely affirm that non-metric approaches rely on a sufficiently stable and reliable theoretical basis implemented on very efficient algorithms. On the other hand, the 'biological side' generates an ever-increasing amount of data amenable to be faced by computational intelligence approaches. The crucial point (deciding for the success/failure of the particular application) is the choice of a representation located at the most 'fruitful' level of biological organisation. The search for the scale maximising 'non-trivial determinism' is a crucial issue in applied statistics [64] and roughly corresponds to the search of the level where the number (and strength) of correlations between the different pieces of information (e.g. different descriptors) reaches a maximum.

[15]Indeed, the absolute entity of metabolic rate can vary for a lot of reasons going from anatomical differences among patients to their actual nutrition state.

Convesely to the classical reductionist tenet, this level (in the case of organised complexity [92]) is seldom located at the most detailed scale of analysis (e.g. single patients in epidemiological studies, single genes, primary structures of proteins, single pixels of an NMR or PET image of the brain, etc.) that in the great majority of cases are dominated by noise [86].

The search for the optimal representation (see the protein solubility case described in Sect. 4) asks for a conscious (and knowledge oriented) decision about the representation level to adopt (e.g. sequence-graph-labelled graph). This choice can only be a mixture of theory and data-driven choices, and thus asks for a real interaction of data scientists and biologists. For this interaction to be fruitful, both the communities must develop a similar language and share at least the basic principles of both the fields.

We think that, beside some 'bombastic exaggerations' on the 'death of science' to be substituted by a purely data-driven theoretically blind approach [2], the future will be characterised by an increasingly stronger integration between computational intelligence and pattern recognition techniques, and the different application fields. Indeed, computational intelligence techniques rely on data-driven modelling (see Sect. 2), which particularly suits problems where the process to be modelled – or at the heart of the problem itself – is unknown or hard to determine in closed-form (e.g. by analytical modelling).

As far as biology (and related fields) are concerned, computational intelligence and pattern recognition can be seen as useful methodological tools in order to perform "in-vitro experiments" and formulate hypotheses to be, if needed, further investigated by means of proper laboratory equipment by field-experts.

In this Chapter, we reviewed and discussed the major challenges and related modus-operandi when dealing with non-metric input spaces in computational intelligence and pattern recognition. By considering bioinformatics and computational biology as application fields, we explored several case studies in which data are conveniently represented by means of complex structures.

We stress that, amongst the three main macro-techniques for solving pattern recognition in non-metric spaces (Sect. 3.2), granular computing seems to be the most appealing in terms of results interpretability and knowledge discovery. Indeed, the automatically extracted information granules are the ones which maximize the classification performances, therefore the most informative and significant for the problem at hand. The set of information granules which, recall, is a set of motifs (i.e. recurrent substructures) can be analysed by field-experts in order to check whether they have some biological soundness and, possibly, boost further research, not only in granular computing and computational intelligence as such, but also in the proper application field in which such techniques have been employed.

References

1. S. Alelyani, J. Tang, H. Liu, Feature selection for clustering: a review. Data Clust. Algorithms Appl. **29**, 110–121 (2013)
2. C. Anderson, The end of theory: the data deluge makes the scientific method obsolete. Wired mag. **16**(7), 16–07 (2008)
3. M. Ankerst, M.M. Breunig, H.P. Kriegel, J. Sander, Optics: ordering points to identify the clustering structure. ACM Sigmod Rec. **28**, 49–60 (1999)
4. A. Bargiela, W. Pedrycz, *Granular Computing: An Introduction* (Kluwer Academic Publishers, Boston, 2003)
5. V. Beckers, L.M. Dersch, K. Lotz, G. Melzer, O.E. Bläsing, R. Fuchs, T. Ehrhardt, C. Wittmann, In silico metabolic network analysis of arabidopsis leaves. BMC Syst. Biol. **10**(1), 102 (2016)
6. J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization. J. Mach. Learn. Res. **13**, 281–305 (2012)
7. H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne, The protein data bank. Nucleic Acids Res. **28**(1), 235–242 (2000)
8. F.M. Bianchi, L. Livi, A. Rizzi, A. Sadeghian, A granular computing approach to the design of optimized graph classification systems. Soft Comput. **18**(2), 393–412 (2014)
9. F.M. Bianchi, S. Scardapane, A. Rizzi, A. Uncini, A. Sadeghian, Granular computing techniques for classification and semantic characterization of structured data. Cogn. Comput. **8**(3), 442–461 (2016)
10. P.S. Bradley, O.L. Mangasarian, W.N. Street, Clustering via concave minimization, in *Advances in Neural Information Processing Systems* (1997), pp. 368–374
11. E. Bullmore, O. Sporns, Complex brain networks: graph theoretical analysis of structural and functional systems. Nat. Rev. Neurosci. **10**(3), 186–198 (2009)
12. G. Carlsson, Topology and data. Bull. Am. Math. Soc. **46**(2), 255–308 (2009)
13. C. Cellucci, *Rethinking Logic: Logic in Relation to Mathematics, Evolution, and Method* (Springer Science & Business Media, 2013)
14. Y. Chen, E.K. Garcia, M.R. Gupta, A. Rahimi, L. Cazzanti, Similarity-based classification: concepts and algorithms. J. Mach. Learn. Res. **10**, 747–776 (2009)
15. Y. Chen, M.R. Gupta, B. Recht, Learning kernels from indefinite similarities, in *Proceedings of the 26th Annual International Conference on Machine Learning* (ACM, 2009), pp. 145–152
16. A. Colorni, M. Dorigo, V. Maniezzo, Distributed optimization by ant colonies, in *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life* (Mit Press, 1992), p. 134
17. D. Counsell, A review of bioinformatics education in the uk. Brief. Bioinform. **4**(1), 7–21 (2003)
18. J. Damoiseaux, S. Rombouts, F. Barkhof, P. Scheltens, C. Stam, S.M. Smith, C. Beckmann, Consistent resting-state networks across healthy subjects. Proc. Natl. Acad. Sci. **103**(37), 13848–13853 (2006)
19. D.L. Davies, D.W. Bouldin, A cluster separation measure. IEEE Trans. Pattern Anal. Mach. Intell. **2**, 224–227 (1979)
20. G. Del Vescovo, L. Livi, F.M. Frattale Mascioli, A. Rizzi, On the problem of modeling structured data with the minsod representative. Int. J. Comput. Theory Eng. **6**(1), 9 (2014)
21. A. Di Noia, P. Montanari, A. Rizzi, Occupational diseases risk prediction by cluster analysis and genetic optimization, in *Proceedings of the International Joint Conference on Computational Intelligence* (SCITEPRESS-Science and Technology Publications, Lda, 2014), pp. 68–75
22. A. Di Noia, P. Montanari, A. Rizzi, Occupational diseases risk prediction by genetic optimization: Towards a non-exclusive classification approach, in *Computational Intelligence* (Springer, Berlin, 2016), pp. 63–77
23. L. Di Paola, M. De Ruvo, P. Paci, D. Santoni, A. Giuliani, Protein contact networks: an emerging paradigm in chemistry. Chem. Rev. **113**(3), 1598–1613 (2012)
24. M. Ester, H.P. Kriegel, J. Sander, X. Xu et al., A density-based algorithm for discovering clusters in large spatial databases with noise. Kdd **96**, 226–231 (1996)

25. M.D. Fox, M.E. Raichle, Spontaneous fluctuations in brain activity observed with functional magnetic resonance imaging. Nat. Rev. Neurosci. **8**(9), 700–711 (2007)
26. K.J. Friston, C.D. Frith, R.S. Frackowiak, R. Turner, Characterizing dynamic brain responses with fmri: a multivariate approach. Neuroimage **2**(2), 166–172 (1995)
27. J. Gao, B. Barzel, A.L. Barabási, Universal resilience patterns in complex networks. Nature **530**(7590), 307–312 (2016)
28. A. Giuliani, S. Filippi, M. Bertolaso, Why network approach can promote a new way of thinking in biology. Front. Genet. **5** (2014)
29. A. Giuliani, A. Krishnan, J.P. Zbilut, M. Tomita, Proteins as networks: usefulness of graph theory in protein science. Curr. Protein Peptide Sci. **9**(1), 28–38 (2008)
30. D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning* (Addison-Wesley, USA, 1989)
31. M.D. Greicius, B. Krasnow, A.L. Reiss, V. Menon, Functional connectivity in the resting brain: a network analysis of the default mode hypothesis. Proc. Natl. Acad. Sci. **100**(1), 253–258 (2003)
32. S. Guha, R. Rastogi, K. Shim, Cure: an efficient clustering algorithm for large databases. ACM Sigmod Rec. **27**, 73–84 (1998)
33. R.W. Hamming, Error detecting and error correcting codes. Bell Labs Tech. J. **29**(2), 147–160 (1950)
34. B. He, K. Wang, Y. Liu, B. Xue, V.N. Uversky, A.K. Dunker, Predicting intrinsic disorder in proteins: an overview. Cell Res. **19**(8), 929–949 (2009)
35. D.R. Hofstadter, *I Am a Strange Loop*, Basic Books (2007)
36. J. Horgan, From complexity to perplexity. Sci. Am. **272**(6), 104–109 (1995)
37. A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review. ACM Comput. Surv. (CSUR) **31**(3), 264–323 (1999)
38. G. Jurman, R. Visintainer, C. Furlanello, An introduction to spectral distances in networks. Front. Artif. Intell. Appl. **226**, 227–234 (2011)
39. L. Kaufman, P. Rousseeuw, Clustering by means of medoids. Stat. Data Anal. Based L1-Norm Relat. Methods, 405–416 (1987)
40. J. Kennedy, R. Eberhart, Particle swarm optimization, in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4 (IEEE, 1995), pp. 1942–1948
41. S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing. Science **220**(4598), 671–680 (1983)
42. V.I. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals. Soviet physics doklady. **10**, 707–710 (1966)
43. A.W.C. Liew, H. Yan, M. Yang, Pattern recognition techniques for the emerging field of bioinformatics: A review. Pattern Recognition **38**(11), 2055–2073 (2005)
44. L. Livi, A. Giuliani, A. Rizzi, Toward a multilevel representation of protein molecules: comparative approaches to the aggregation/folding propensity problem. Inf. Sci. **326**, 134–145 (2016)
45. L. Livi, A. Giuliani, A. Sadeghian, Characterization of graphs for protein structure modeling and recognition of solubility. Curr. Bioinform. **11**(1), 106–114 (2016)
46. L. Livi, E. Maiorino, A. Giuliani, A. Rizzi, A. Sadeghian, A generative model for protein contact networks. J. Biomol. Struct. Dyn. **34**(7), 1441–1454 (2016)
47. L. Livi, A. Rizzi, The graph matching problem. Pattern Anal. Appl. **16**(3), 253–283 (2013)
48. L. Livi, A. Rizzi, A. Sadeghian, Optimized dissimilarity space embedding for labeled graphs. Inf. Sci. **266**, 47–64 (2014)
49. L. Livi, A. Rizzi, A. Sadeghian, Granular modeling and computing approaches for intelligent analysis of non-geometric data. Appl. Soft Comput. **27**, 567–574 (2015)
50. L. Livi, A. Sadeghian, Granular computing, computational intelligence, and the analysis of non-geometric input spaces. Granul. Comput. **1**(1), 13–20 (2016)
51. S. Lloyd, Least squares quantization in pcm. IEEE Trans. Inf. Theory **28**(2), 129–137 (1982)
52. L. MacQueen, Some methods for classification and analysis of multivariate observations, in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1 (Oakland, USA, 1967), pp. 281–297

53. H.A. Maghawry, M.C. Mostafa, M.H. Abdul-Aziz, T.E. Gharib, A modified cutoff scanning matrix protein representation for enhancing protein function prediction, in *9th International Conference on Informatics and Systems (INFOS)* (IEEE, 2014), pp. DEKM–40
54. E. Maiorino, A. Rizzi, A. Sadeghian, A. Giuliani, Spectral reconstruction of protein contact networks. Phys. A: Stat. Mech. Appl. **471**, 804–817 (2017)
55. A. Martino, E. Maiorino, A. Giuliani, M. Giampieri, A. Rizzi, Supervised approaches for function prediction of proteins contact networks from topological structure information, in *Scandinavian Conference on Image Analysis* (Springer, Berlin, 2017), pp. 285–296
56. A. Martino, A. Rizzi, F.M. Frattale Mascioli, Efficient approaches for solving the large-scale k-medoids problem, in *Proceedings of the 9th International Joint Conference on Computational Intelligence. IJCCI*, vol. 1 (INSTICC, 2017), pp. 338–347
57. J. Mercer, Functions of positive and negative type, and their connection with the theory of integral equations, in *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 209 (1909), pp. 415–446
58. D.C. Mikulecky, Network thermodynamics and complexity: a transition to relational systems theory. Comput. Chem. **25**(4), 369–391 (2001)
59. T.M. Mitchell, *Machine Learning* (McGraw-Hill Boston, MA, 1997)
60. M. Neuhaus, H. Bunke, *Bridging the Gap Between Graph Edit Distance and Kernel Machines*, vol. 68 (World Scientific, 2007)
61. M. Pagani, A. Giuliani, J. Öberg, A. Chincarini, S. Morbelli, A. Brugnolo, D. Arnaldi, A. Picco, M. Bauckneht, A. Buschiazzo et al., Predicting the transition from normal aging to alzheimer's disease: a statistical mechanistic evaluation of fdg-pet data. NeuroImage **141**, 282–290 (2016)
62. M. Pagani, A. Giuliani, J. Öberg, F. De Carli, S. Morbelli, N. Girtler, F. Bongioanni, D. Arnaldi, J. Accardo, M. Bauckneht et al., Progressive disgregation of brain networking from normal aging to alzheimer's disease. independent component analysis on fdg-pet data. J. Nucl. Med. jnumed–116 (2017)
63. E. Parzen, On estimation of a probability density function and mode. Ann. Math. Stat. **33**(3), 1065–1076 (1962)
64. M. Pascual, S.A. Levin, From individuals to population densities: searching for the intermediate scale of nontrivial determinism. Ecology **80**(7), 2225–2236 (1999)
65. K. Pearson, Mathematical contributions to the theory of evolution. iii. regression, heredity, and panmixia, in *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 187 (1896), pp. 253–318
66. E. Pękalska, R.P. Duin, *The Dissimilarity Representation for Pattern Recognition: Foundations and Applications* (World Scientific, 2005)
67. K. Peng, P. Radivojac, S. Vucetic, A.K. Dunker, Z. Obradovic, Length-dependent prediction of protein intrinsic disorder. BMC Bioinform. **7**(1), 208 (2006)
68. J.B. Pereira, M. Mijalkov, E. Kakaei, P. Mecocci, B. Vellas, M. Tsolaki, I. Kłoszewska, H. Soininen, C. Spenger, S. Lovestone et al., Disrupted network topology in patients with stable and progressive mild cognitive impairment and alzheimer's disease. Cereb. Cortex **26**(8), 3476–3493 (2016)
69. F. Possemato, A. Rizzi, Automatic text categorization by a granular computing approach: facing unbalanced data sets, in *The International Joint Conference on Neural Networks (IJCNN)* (IEEE, 2013), pp. 1–8
70. J.S. Richardson, The anatomy and taxonomy of protein structure. Adv. Protein Chem. **34**, 167–339 (1981)
71. D. de Ridder, J. de Ridder, M.J. Reinders, Pattern recognition in bioinformatics. Brief. Bioinform. **14**(5), 633–647 (2013)
72. A. Rizzi, F. Possemato, L. Livi, A. Sebastiani, A. Giuliani, F.M. Frattale Mascioli, A dissimilarity-based classifier for generalized sequences by a granular computing approach, in *The International Joint Conference on Neural Networks (IJCNN)* (IEEE, 2013), pp. 1–8
73. P. Romero, Z. Obradovic, X. Li, E.C. Garner, C.J. Brown, A.K. Dunker, Sequence complexity of disordered protein. Proteins Struct. Funct. Bioinform. **42**(1), 38–48 (2001)

74. P.J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. **20**, 53–65 (1987)
75. M. Rubinov, O. Sporns, Complex network measures of brain connectivity: uses and interpretations. Neuroimage **52**(3), 1059–1069 (2010)
76. H. Sakoe, S. Chiba, Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans. Acoust. Speech Signal Process. **26**(1), 43–49 (1978)
77. B. Schölkopf, A.J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond* (MIT press, 2002)
78. J. Shawe-Taylor, N. Cristianini, *Kernel Methods for Pattern Analysis* (Cambridge university press, Cambridge, 2004)
79. D.H. Silverman, G.W. Small, C.Y. Chang, C.S. Lu, M.A.K. de Aburto, W. Chen, J. Czernin, S.I. Rapoport, P. Pietrini, G.E. Alexander et al., Positron emission tomography in evaluation of dementia: regional brain metabolism and long-term outcome. Jama **286**(17), 2120–2127 (2001)
80. G.P. Singh, M. Ganapathi, D. Dash, Role of intrinsic disorder in transient interactions of hub proteins. Proteins Struct. Funct. Bioinform. **66**(4), 761–765 (2007)
81. J. Smucny, K.P. Wylie, J.R. Tregellas, Functional magnetic resonance imaging of intrinsic brain networks for translational drug discovery. Trends Pharmacol. Sci. **35**(8), 397–403 (2014)
82. C. Soguero-Ruiz, K. Hindberg, J.L. Rojo-Álvarez, S.O. Skrøvseth, F. Godtliebsen, K. Mortensen, A. Revhaug, R.O. Lindsetmo, K.M. Augestad, R. Jenssen, Support vector feature selection for early detection of anastomosis leakage from bag-of-words in electronic health records. IEEE J. Biomed. Health Inf. **20**(5), 1404–1415 (2016)
83. P.G. Spetsieris, J.H. Ko, C.C. Tang, A. Nazem, W. Sako, S. Peng, Y. Ma, V. Dhawan, D. Eidelberg, Metabolic resting-state brain networks in health and disease. Proc. Natl. Acad. Sci. **112**(8), 2563–2568 (2015)
84. J.M. Stanton, Galton, pearson, and the peas: A brief history of linear regression for statistics instructors. J. Stat. Education **9**(3), 1–16 (2001)
85. S. Theodoridis, K. Koutroumbas, *Pattern Recognition*, 4th edn. (Academic Press, 2008)
86. M.K. Transtrum, B.B. Machta, K.S. Brown, B.C. Daniels, C.R. Myers, J.P. Sethna, Perspective: sloppiness and emergent theories in physics, biology, and beyond. J. Chem. Phys. **143**(1), 07B201_1 (2015)
87. V.N. Uversky, Natively unfolded proteins: a point where biology waits for physics. Protein Sci. **11**(4), 739–756 (2002)
88. B.C. Van Wijk, C.J. Stam, A. Daffertshofer, Comparing brain networks of different size and connectivity density using graph theory. PloS one **5**(10), e13701 (2010)
89. J.P. Vert, K. Tsuda, B. Schölkopf, *Kernel Methods in Computational Biology*, A primer on kernel methods (2004), pp. 35–70
90. Y.C. Wang, Y. Wang, Z.X. Yang, N.Y. Deng, Support vector machine prediction of enzyme function with conjoint triad feature and hierarchical context. BMC Syst. Biol. **5**(1), S6 (2011)
91. L. Wasserman, Topological data analysis. Ann. Rev. Stat. Appl. **5**(1) (2018)
92. W. Weaver, Science and complexity. Am. Sci. **36**(4), 536 (1948)
93. A. Wright, A.B. McCoy, S. Henkin, A. Kale, D.F. Sittig, Use of a support vector machine for categorizing free-text notes: assessment of accuracy across two institutions. J. Am. Med. Inf. Assoc. **20**(5), 887–890 (2013)
94. Y. Yang, L. Han, Y. Yuan, J. Li, N. Hei, H. Liang, Gene co-expression network analysis reveals common system-level properties of prognostic genes across cancer types. Nat. Commun. **5**, 3231 (2014)
95. F. Yates, K. Mather, Ronald aylmer fisher, 1890–1962. Biogr. Mem. Fellows R. Soc. **9**, 91–129 (1963)
96. L.A. Zadeh, Soft computing and fuzzy logic. IEEE Softw. **11**(6), 48–56 (1994)
97. T. Zhang, R. Ramakrishnan, M. Livny, Birch: an efficient data clustering method for very large databases. ACM Sigmod Rec. **25**, 103–114 (1996)