Journal of Statistical Mechanics: Theory and Experiment
An IOP and SISSA journal

Clustering with shallow trees

M Bailly-Bechet[1], S Bradde[2,3], A Braunstein[4], A Flaxman[5], L Foini[2,3] and R Zecchina[4]

[1] Université Lyon 1, CNRS UMR 5558, Laboratoire de Biométrie et Biologie Évolutive, Villeurbanne, France
[2] SISSA, via Beirut 2/4, Trieste, Italy
[3] INFN Sezione di Trieste, Italy
[4] Politecnico di Torino, Corso Duca degli Abbruzzi 24, Torino, Italy
[5] IHME, University of Washington, Seattle, WA, USA
E-mail: mbailly@biomserv.univ-lyon1.fr, bradde@sissa.it, alfredo.braunstein@polito.it, abie@u.washington.edu, laura.foini@sissa.it and riccardo.zecchina@polito.it

Received 6 October 2009
Accepted 25 November 2009
Published 21 December 2009

Online at stacks.iop.org/JSTAT/2009/P12010
doi:10.1088/1742-5468/2009/12/P12010

Abstract. We propose a new method for obtaining hierarchical clustering based on the optimization of a cost function over trees of limited depth, and we derive a message-passing method that allows one to use it efficiently. The method and the associated algorithm can be interpreted as a natural interpolation between two well-known approaches, namely that of single linkage and the recently presented affinity propagation. We analyse using this general scheme three biological/medical structured data sets (human population based on genetic information, proteins based on sequences and verbal autopsies) and show that the interpolation technique provides new insight.

Keywords: cavity and replica method, message-passing algorithms

ArXiv ePrint: 0910.0767

1742-5468/09/P12010+17$30.00

Contents

1. Introduction

A standard approach to data clustering, that we will also follow here, involves defining a measure of distance between objects, called dissimilarity. In this context, generally speaking, data clustering deals with the problem of classifying objects so that objects within the same class or cluster are more similar than objects belonging to different classes. The choices of the measure of similarity and the clustering algorithms are crucial in the sense that they define an underlying model for the cluster structure. In this work we discuss two somewhat opposite clustering strategies, and show how they nicely fit as limit cases of a more general scheme that we propose.

Two well-known general approaches that are extensively employed are partitioning methods and hierarchical clustering methods [1]. Partitioning methods are based on the choice of a given number of *centroids*—i.e. reference elements—to which the other elements have to be compared. In this sense the problem reduces to finding a set of centroids that minimizes the cumulative distance to points of the data set. Two of the most commonly used partitioning algorithms are the K-means (KM) and affinity propagation (AP) ones [2, 3]. Behind these methods, there is the assumption of spherical distribution of data: clusters are forced to be loosely of spherical shape, with respect to the dissimilarity metric. These techniques give good results normally only when the structure underlying the data fits this hypothesis. Nevertheless, with soft affinity propagation [2] the hard spherical constraint is relaxed, allowing for cluster structures including deviation from the regular shape. This method however recovers only partially information on hierarchical organization. On the other hand, hierarchical clustering methods, such as that based on single linkage (SL) [4], start by defining a cluster for each element of the system and then proceed by repeatedly merging the two closest clusters into one. This procedure provides a hierarchical sequence of clusters.

Recently an algorithm for efficiently approximating optimum spanning trees with a maximum depth D was presented in [5]. We show here how this algorithm may be

doi:10.1088/1742-5468/2009/12/P12010

J. Stat. Mech. (2009) P12010

used to cluster data, in a method that can be understood as a generalization of both (or rather an interpolation between) the AP and SL algorithms. Indeed in the $D = 2$ and n limits—where n is the number of objects to cluster—one recovers respectively the AP and SL methods. As a proof of concept, we apply the new approach to a collection of biological and medical clustering problems for which intermediate values of D provide new interesting results. In section 2, we define an objective function for clustering based on the cost of certain trees over the similarity matrix, and we devise a message-passing strategy for optimizing the objective function. The following section is devoted to recovering two known algorithms, AP and SL, which are shown to be special cases for appropriately selected values of the external parameters D. Finally, in the last section we apply the algorithm to three biological/medical data clustering problems for which external information can be used to validate the algorithmic performance. First, we cluster human individuals from several geographical origins using their genetic differences, then we tackle the problem of clustering homologous proteins using only their amino acid sequences. Finally we consider a clustering problem arising in the analysis of causes of death in regions where vital registration systems are not available.

2. A common framework

Let us start with some definitions. Given n data points, we introduce the similarity matrix connecting pairs $s_{i,j}$, where $i, j \in [1, \ldots, n]$. This interaction could be represented as a fully connected weighted graph $G(n, s)$ where s is the weight associated with each edge. This matrix constitutes the only data input for the clustering methods discussed in this work. We refer in the following to the neighbourhood of node i with the symbol ∂i, denoting the ensemble of all nearest neighbours of i. By adding to the graph G one artificial node v^*, called the *root*, whose similarity to all other nodes $i \in G$ is a constant parameter λ, we obtain a new graph $G^*(n+1, s^*)$ where s^* is an $(n+1) \times (n+1)$ matrix with one added row and column of constant value to the matrix s (see figure 1).

We will employ the following general scheme for clustering based on trees. Given any tree T that spans all the nodes in the graph $G^*(n+1, s^*)$, consider the (possibly disconnected) subgraph resulting of removing the root v^* and all its links. We will define the output of the clustering scheme as the family of vertex sets of the connected components of this subgraph. That is, each cluster will be formed by a connected component of the pruned $T \setminus v^*$. In the following, we will concentrate on how to produce trees associated with G^*.

The algorithm described in [5] was devised to find a tree of minimum weight with a depth bounded by D from a selected root to a set of terminal nodes. In the clustering framework, all nodes are terminals and must be reached by the tree. As a tree has exactly $n-1$ links, for values of D greater than or equal to n the problem becomes the familiar (unconstrained) minimum spanning tree problem. In the rest of this section we will describe the D-MST message-passing algorithm of [5] for Steiner trees in the simplified context of (bounded depth) spanning trees.

With each node of the graph we associate two variables π_i and d_i where $\pi_i \in \partial i$ could be interpreted as a pointer from i to one of the neighbouring nodes $j \in \partial i$. Meanwhile $d_i \in [0, \ldots, D]$ is thought of as a discrete distance between the node i and the root v^* along the tree. Necessarily, only the root has zero distance $d_{v^*} = 0$, while for all other

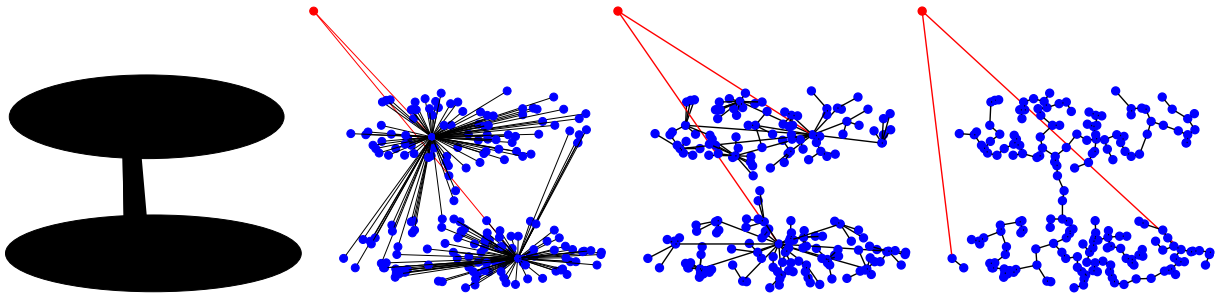

Figure 1. Clustering an artificial 2D image. The black image on the left was randomly sampled and the Euclidean distance was used as a measure of dissimilarity between nodes. Clustering by D-MST was then attempted on the resulting graph. One external root vertex v^* (the red point) was added, with distance λ to every other points. The output of the algorithm consists of a minimum weight rooted spanning tree of depth D indicated by bold links. The last three figures concern the resulting clustering for different choices of the depth limit $D = 2, 4, > n$ respectively. Different clusters with a complex internal structure can be recovered after removing the red node v^*. In the case of AP, $D = 2$ (the second figure), the spherical clusters do not fit the ellipsoidal shape of the original figure while for 4-MST (the third figure) the structure of two ellipses can be recovered. The fourth and last figure corresponds to SL ($D > n$): in this case nodes are split into two arbitrary components, disregarding the original shape.

nodes $d_i \in [1, \ldots, D]$. In order to ensure global connectivity of the D-MST, these two variables must satisfy the following condition: $\pi_i = j \Rightarrow d_i = d_j + 1$. This means that if node j is the parent of node i, then the depth of node i must exceed the depth of the node j by precisely 1. This condition avoids the presence of loops and forces the graph to be connected, assigning non-null weight only to configurations corresponding to trees. The energy function thus reads

$$E(\{\pi_i, d_i\}_{i=1}^N) = \sum_i s_{i,\pi_i} - \sum_{i,j\in\partial i} (h_{ij}(\pi_i, \pi_j, d_i, d_j) + h_{ji}(\pi_j, \pi_i, d_j, d_i)), \qquad (1)$$

where h_{ij} is defined as

$$h_{ij} = \begin{cases} 0 & \{\pi_i = j \Rightarrow d_i = d_j + 1\} \\ -\infty & \text{else.} \end{cases} \qquad (2)$$

In this way only configurations corresponding to a tree are taken into account with the usual Boltzmann weight factor $\mathrm{e}^{-\beta s_{i,\pi_i}}$ where the external parameter β fixes the value of energy level. Thus the partition function is

$$Z(\beta) = \sum_{\{\pi_i, d_i\}} \mathrm{e}^{-\beta E(\{\pi_i, d_i\})} = \sum_{\{\pi_i, d_i\}} \prod_i \mathrm{e}^{-\beta s_{i,\pi_i}} \times \prod_{ij\in\partial i} f_{ij}, \qquad (3)$$

where we have introduce an indicator function $f_{ij} = g_{ij} g_{ji}$. Each term $g_{ij} = 1 - \delta_{\pi_i,j}(1 - \delta_{d_j,d_i-1})$ is equivalent to $e^{h_{ij}}$ while δ_{ij} is the delta function. In terms of these quantities f_{ij} it is possible to derive the cavity equations, i.e. the following set of coupled equations

J. Stat. Mech. (2009) P12010

for the cavity marginal probability $P_{j\to i}(d_j, \pi_j)$ of each site $j \in [1, \ldots, n]$ after removing one of the nearest neighbours $i \in \partial j$:

$$P_{j\to i}(d_j, \pi_j) \propto \mathrm{e}^{-\beta s_{i,\pi_i}} \prod_{k\in\partial j/i} Q_{k\to j}(d_j, \pi_j) \tag{4}$$

$$Q_{k\to j}(d_j, \pi_j) \propto \sum_{d_k \pi_k} P_{k\to j}(d_k, \pi_k) f_{jk}(d_j, \pi_j, d_k, \pi_k). \tag{5}$$

These equations are solved iteratively and in graphs with no cycles they are guaranteed to converge to a fixed point that is the optimal solution. In terms of cavity probability we are able to compute marginal and joint probability distributions using the following relations:

$$P_j(d_j, \pi_j) \propto \prod_{k\in\partial j} Q_{k\to j}(d_j, \pi_j) \tag{6}$$

$$P_{ij}(d_i, \pi_i, d_j, \pi_j) \propto P_{i\to j}(d_i, \pi_i) P_{j\to i}(d_j, \pi_j) f_{ij}(d_i, \pi_i, d_j, \pi_j). \tag{7}$$

For general graphs, convergence can be forced by introducing a 'reinforcement' perturbation term as in [5, 6]. This leads to a new set of perturbed coupled equations that show good convergence properties. The $\beta \to \infty$ limit is taken by considering the changes of variable $\psi_{j\to i}(d_j, \pi_j) = \beta^{-1} \log P_{j\to i}(d_j, \pi_j)$ and $\phi_{j\to i}(d_j, \pi_j) = \beta^{-1} \log Q_{j\to i}(d_j, \pi_j)$; then the relations (4) and (5) reduce to

$$\psi_{j\to i}(d_j, \pi_j) = -s_{i,\pi_i} + \sum_{k\in\partial j/i} \phi_{k\to j}(d_j, \pi_j) \tag{8}$$

$$\phi_{k\to j}(d_j, \pi_j) = \max_{d_k \pi_k : f_{kj}\neq 0} \psi_{k\to j}(d_k, \pi_k). \tag{9}$$

These equations are in the 'max-sum' form and equalities hold up to some additive constant. In terms of these quantities, marginals are given by $\psi_j(d_i, \pi_j) = -c_{j\pi_j} + \sum_k \phi_{k\to j}(d_j, \pi_j)$ and the optimum tree is the one obtained using argmax ψ_j. If we introduce the variables $A^d_{k\to j} = \max_{\pi_k \neq j} \psi_{k\to j}(d, \pi_k)$, $C^d_{k\to j} = \psi_{k\to j}(d, j)$ and $E^d_{k\to j} = \max(C^d_{k\to j}, A^d_{k\to j})$ it is enough to compute all the messages $\phi_{k\to j}(d_j, \pi_j) = A^{d_j - 1}_{k\to j}, E^{d_j}_{k\to j}$ for $\pi_j = k$ and $\pi_j \neq k$ respectively. Using equations (8) and (9) we obtain the following set of equations:

$$A^d_{j\to i}(t+1) = \sum_{k\in N(j)/i} E^d_{k\to j}(t) + \max_{k\in N(j)/i} (A^{d-1}_{k\to j}(t) - E^d_{k\to j}(t) - s_{j,k}) \tag{10}$$

$$C^d_{j\to i}(t+1) = -s_{j,i} + \sum_{k\in N(j)/i} E^d_{k\to j}(t) \tag{11}$$

$$E^d_{j\to i}(t+1) = \max(C^d_{j\to i}(t+1), A^d_{j\to i}(t+1)). \tag{12}$$

It has been demonstrated [7] that a fixed point of these equations with depth $D > n$ is an optimal spanning tree. In the following two subsections, we show how to recover the

J. Stat. Mech. (2009) P12010

SL and AP algorithms. On one hand, by computing the (unbounded depth) spanning tree on the enlarged matrix and then considering the connected components of its restriction to the set of nodes removing v^*, we recover the results obtained by SL. On the other hand we obtain AP by computing the $D = 2$ spanning tree rooted at v^*, defining the self-affinity parameter as the weight for reaching this root node.

2.1. The single-linkage limit

The single-linkage approach is one of the oldest and simplest clustering methods, and there are many possible descriptions of it. One of them is the following: order all pairs according to distances, and erase as many of the pairs with the largest distance as possible such that the number of resulting connected components is exactly k. Define clusters as the resulting connected components.

An alternative method consists in removing initially all *useless* pairs (i.e. pairs that would not change the set of components when removed in the above procedure). This reduces to the following algorithm: given the distance matrix s, compute the minimum spanning tree on the complete graph with weights given by s. From the spanning tree, remove the $k-1$ links with largest weight. Clusters are given by the resulting connected components. In many cases there is no *a priori* desired number of clusters k and an alternative way of choosing k is to use a continuous parameter λ to erase all weights larger than λ.

The D-MST problem for $D > n$ identifies the minimum spanning tree connecting all $n+1$ nodes (including the root v^*). This means that each node i will point to one other node $\pi_i = j \neq v^*$ if its weight satisfies the condition $\min_j s_{i,j} < s_{i,v^*}$; otherwise it would be cheaper to connect it to the root (introducing one more cluster). We will make this description more precise. For simplicity, let us assume that no edge in $G(n, s)$ has weight exactly equal to λ.

The Kruskal algorithm [8] is a classical algorithm for computing a minimum spanning tree. It works by iteratively creating a forest as follows: start with a subgraph that is all nodes and has no edges. Then scan the list of edges ordered with increasing weight, and add the edge to the forest if it connects two different components (i.e. if it does not close a loop). At the end of the procedure, it is easy to prove that the forest has only one connected component that forms a minimum spanning tree. It is also easy to see that the edges added when applying the Kruskal algorithm to $G(n, s)$ up to the point when the weight reaches λ are also admitted on the Kruskal algorithm for $G(n+1, s^*)$. After that point, the two procedures diverge because on $G(n, s)$ the remaining added edges have weight larger than λ while on $G(n+1, s^*)$ all remaining added edges have weight exactly λ. Summarizing, the MST on $G(n+1, s^*)$ is a MST on $G(n, s)$ on which all edges with weight greater than λ have been replaced by edges connecting with v^*.

2.2. The affinity propagation limit

Affinity propagation is a method that was recently proposed in [3], based on the choice of a number of 'exemplar' data points. Starting with a similarity matrix s, choose a set of exemplar data points $X \subset V$ and an assignment $\phi: V \mapsto X$ such that: $\phi(x) = x$ if $x \in X$ and the sum of the distances between data points and the exemplars that they map to is minimized. It is essentially based on iteratively passing messages of

two types between elements, representing *responsibility* and *availability*. The first, $r_{i\to j}$, measures how much an element i would prefer to choose the target j as its exemplar. The second $a_{i\to j}$ gives the preference for i to be chosen as an exemplar by data point j. This procedure is an efficient implementation of the max-sum algorithm that improves the naive exponential time complexity to $O(n^2)$. The self-affinity parameter, namely $s_{i,i}$, is chosen as the dissimilarity of an exemplar with itself, and *in fine* regulates the number of groups in the clustering procedure, by allowing more or less points to link with 'dissimilar' exemplars.

Given a similarity matrix s for n nodes, we want to identify the *exemplars*, that is, to find a valid configuration $\bar{\pi} = \{\pi_1, \ldots, \pi_n\}$ such that $\pi\colon [1, \ldots, n] \mapsto [1, \ldots, n]$ so as to minimize the function

$$E(\bar{\pi}) = -\sum_{i=1}^{n} s_{i,\pi_i} - \sum_i \delta_i(\bar{\pi}), \tag{13}$$

where the constraint reads

$$\delta_i(\bar{\pi}) = \begin{cases} -\infty & \pi_i \neq i \cap \exists\, j\colon \pi_j = i \\ 0 & \text{else.} \end{cases} \tag{14}$$

These equations take into account the only possible configurations, where node i either is an exemplar, meaning $\pi_i = i$, or it is not chosen as an exemplar by any other node j. The energy function thus reads

$$E(\bar{\pi}) = \begin{cases} -\sum_i s_{i,\pi_i} & \forall\, i\{\pi_i = i \cup\ \forall j \pi_j \neq i\} \\ \infty & \text{else.} \end{cases} \tag{15}$$

The cavity equations are computed starting from this definition, and after some algebra they reduce to the following update conditions for responsibility and availability [3]:

$$r_{i\to k}^{t+1} = s_{i,k} - \max_{k'\neq k}(a_{k'\to i}^{t} + s_{k',i}) \tag{16}$$

$$a_{k\to i}^{t+1} = \min\left(0, r_{k\to k} + \sum_{i'\neq k\ i} \max(0, r_{i'\to k}^{t})\right). \tag{17}$$

In order to prove the equivalence between the two algorithms, i.e. D-MST for $D = 2$ and AP, we show in the following how the two employ an identical decomposition of the same energy function, thus resulting necessarily in the same max-sum equations. In the 2-MST equations, we are partitioning all nodes into three groups: the first one is just the root whose distance $d = 0$, the second one is composed of nodes pointing at the root $d = 1$ and the last one is made up of nodes pointing to other nodes that have distance $d = 2$ from the root. The following relations between d_i and π_i make this condition explicit:

$$d_i = \begin{cases} 1 & \Leftrightarrow \pi_i = v^* \\ 2 & \Leftrightarrow \pi_i \neq v^*. \end{cases} \tag{18}$$

It is clear that the distance variable d_i is redundant because the two kinds of nodes are perfectly distinguished with just the variable π_i. Going a step further we could

J. Stat. Mech. (2009) P12010

remove the external root v^* upon imposing the following condition for the pointers $\pi_i = i \Leftrightarrow \pi_i = v^*\ \pi_i = j \neq i \Leftrightarrow \pi_i \neq v^*$. This can be understood by thinking of the AP procedure: since nodes at distance 1 from the root are the exemplars, they might point to themselves, as defined in AP, and all the non-exemplars are at distance $d = 2$, so they might point to nodes at distance $d = 1$. Using this translation, from equation (2) it follows that

$$\sum_{ij\in\partial i} h_{ij} + h_{ji} = \begin{cases} 0 & \forall\, i\{\pi_i = i \cup\ \forall\, j \neq i \pi_j \neq i\} \\ -\infty & \text{else} \end{cases} \tag{19}$$

meaning that the constraints are equivalent: $\sum_{ij\in\partial i} h_{ij} + h_{ji} = \sum_i \delta_i(\bar{\pi})$. Substituting (19) into equation (1) we obtain that

$$E(\{\pi_i, d_i\}_{i=1}^n) = \begin{cases} -\sum_i s_{i,\pi_i} & \forall\, i\{\pi_i = i \cup\ \forall\, j \neq i \pi_j \neq i\} \\ \infty & \text{else.} \end{cases} \tag{20}$$

The identification of the self-affinity parameter and the self-similarity $s_{i,v^*} = \lambda = s_{i,i}$ allows us to prove the equivalence between this formula and the AP energy given in equation (15), as desired.

3. Applications to biological data

In the following sections, we shall apply the new technique to different clustering problems and give a preliminary comparison to the two extreme limits of the interpolation, namely $D = 2$ (AP) and $D = n$ (SL).

Clustering is a widely used method of analysis in biology, most notably in the recently developed fields of transcriptomics [9], proteomics and genomics [10], where huge quantities of noisy data are generated routinely. A clustering approach presents many advantages for such data: it can use all pre-existing knowledge available to choose group numbers and to assign elements to groups, it has good properties of noise robustness [11], and it is computationally more tractable than other statistical techniques. In this section we apply our algorithm to structured biological data, in order to show that by interpolating between two well-known clustering methods (SL and AP) it is possible to obtain new insight.

3.1. Multilocus genotype clustering

In this application we used the algorithm to classify individuals according to their original population using only information from their sequence SNPs as a distance measure [12]. A single-nucleotide polymorphism (SNP) is a DNA sequence variation occurring when a single nucleotide (A, T, C, or G) in the genome (or other shared sequence) differs between members of a species (or between paired chromosomes in a diploid individual). The data set that we used is from the HapMap Project, an international project launched in 2002 with the aim of providing a public resource for accelerating medical genetic research [13]. It consists of SNP data from 269 individuals from four geographically diverse origins: 90 Utah residents from North and West Europe (CEU), 90 Yoruba of Ibadan, Nigeria (YRI), 45 Han Chinese of Beijing (CHB) and 44 Tokyo Japanese (JPT). CEU and YRI samples

J. Stat. Mech. (2009) P12010

are articulated in thirty families of three people each, while CHB and JPT have no such structure. For each individual about four million SNPs are given, allocated on different chromosomes. In the original data set some SNPs were defined only in subpopulations; thus we extracted those which were well defined for every sample in all populations and after this selection the number of SNPs for each individual dropped to 1.8 million. We defined the distance between samples as the number of different alleles on the same locus between individuals normalized by the total number of counts. The 269×269 matrix of distance S was defined as follows:

$$s_{i,j} = \frac{1}{2N} \sum_{n=1}^{N} d_{ij}(n), \tag{21}$$

where N is the number of valid SNP loci and $d_{ij}(n)$ is the distance between the nth genetic loci of individuals i and j:

$$d_{ij}(n) = \begin{cases} 0 & \text{if } i \text{ and } j \text{ have two alleles in common at the } n\text{th locus,} \\ 1 & \text{if } i \text{ and } j \text{ share only one single allele in common,} \\ 2 & \text{if } i \text{ and } j \text{ have no alleles in common.} \end{cases} \tag{22}$$

The resulting distance matrix was given as input to the D-MST algorithm. In figure 3 we show the clusters found by the algorithm using a maximum depth $D = 5$. Each individual is represented by a number and coloured according to the population that it belongs to: green for YRI, yellow for CEU, blue for JPT and red for CHB. One can see that the algorithm recognizes the populations, grouping the individuals in four clusters. There is only one misclassified case, a JPT individual placed in the CHB cluster.

Moreover, noticing that yellow and green clusters have a more regular internal structure than the other two, it is possible to consider them separately. Therefore, if one applies the D-MST algorithm to this restricted subset of data, all families consisting of three people can be immediately recovered, and the tree subdivides into 60 families of 3 elements, without any error (details not reported).

This data set is particularly hard to classify, due to the complexity of the distance distribution. In fact the presence of families creates a sub-clustered structure inside the groups of YRI and CEU individuals. Secondly CHB and JPT people, even if they belong to different populations, share in general smaller distances with respect to those subsisting among different families inside one of the other two clusters. The D-MST algorithm overcomes this subtlety with the possibility of developing a complex structure and allows the correct detection of the four populations while other algorithms, such as AP, cannot adapt to this variability of the typical distance scale between groups in the data set. Indeed, the hard constraint in AP relies strongly on cluster shape regularity and forces clusters to appear as stars of radius 1: there is only one central node, and all other nodes are directly connected to it. Elongated or irregular multi-dimensional data might have more than one simple cluster centre. In this case AP may force division of single clusters into separate ones or may group together different clusters, according to the input self-similarities. Moreover, since all data points in a cluster must point to the same exemplar, all information about the internal structure, such as family grouping, is lost. Thus, after running AP, CHB and JPT are grouped in a unique cluster, and CHB and JPT have the

J. Stat. Mech. (2009) P12010

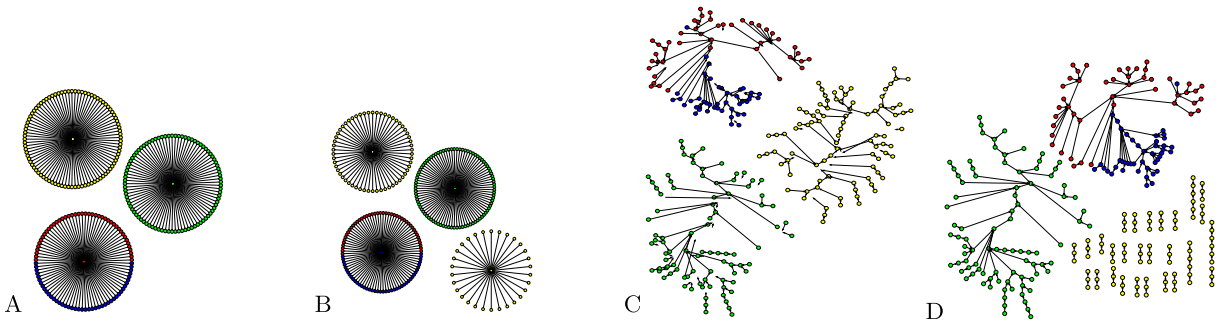

Figure 2. In this figure we compare the results of single-linkage and affinity propagation techniques on a SNP distance data set. The data set is composed of 269 individuals divided into four populations: CHB (red), CEU (green), YRI (yellow) and JPT (blue). The panels (A) and (B) are AP results while panels (C) and (D) show clusters obtained with SL. As λ increases, both algorithms fail to divide Chinese from Japanese populations (panels (A), (C)) before splitting the Nigerian population (yellow).

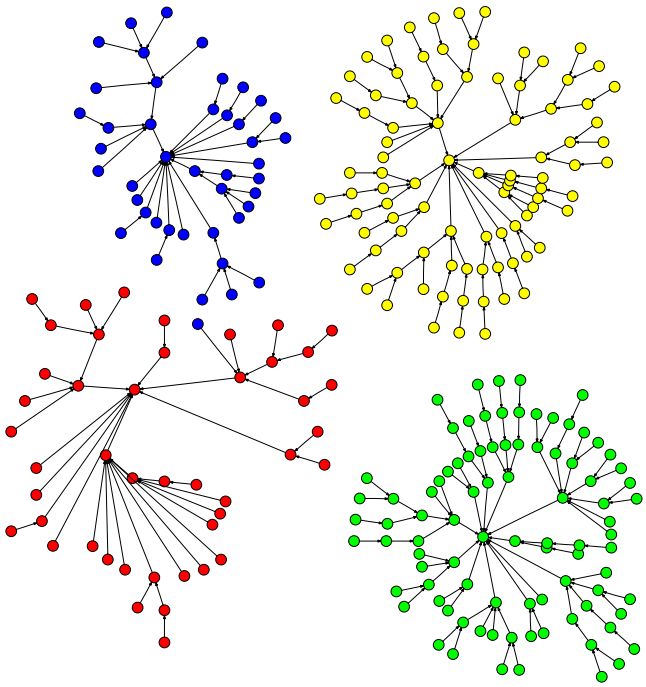

Figure 3. In this figure we report the clustering by D-MST with a fixed maximum depth $D = 5$. The algorithm gives only one misclassification.

same exemplar, as shown in figure 2(A). Going a step further and forcing the algorithm to divide Chinese from Japanese we start to split the YRI population (figure 2(B)).

Hierarchical clustering also fails on this data set, recognizing the same three clusters found by affinity propagation at the three-cluster level (figure 2(C)) and splitting the yellow population into families before dividing blue from red (see figure 2(D)). This makes sense relative to the typical dissimilarities between individuals, but prevents grasping the entire population structure.

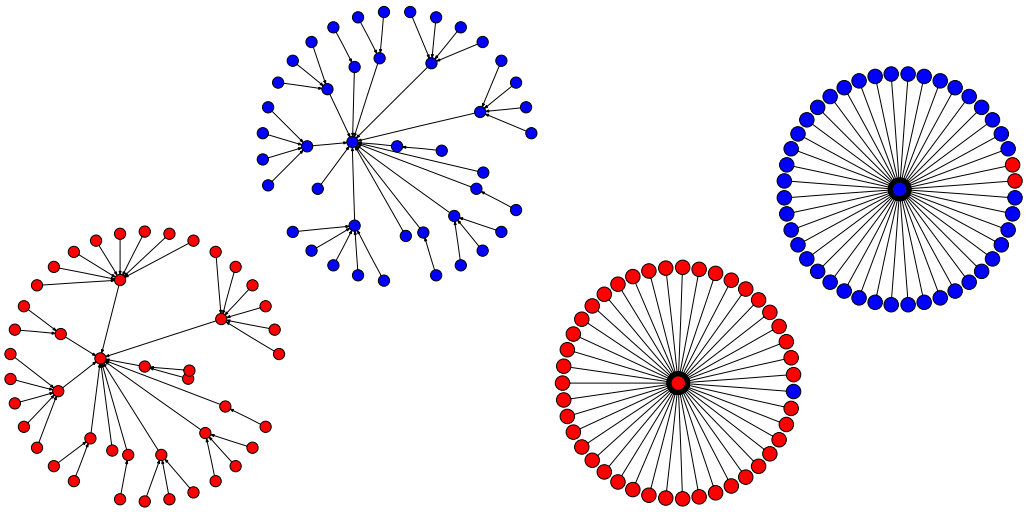

Figure 4. We report clusters results obtained using the D-MST algorithm with $D = 2$ (right) and $D = 3$ (left), considering only CHB and JPT. While both algorithms perform well for this subset, the 3-MST algorithm correctly classifies all individuals.

After considering all four populations together, we applied the D-MST algorithm only to the subset consisting of CHB and JPT individuals, because these data appeared the hardest to cluster correctly. The result is that the D-MST algorithm with depth $D = 3$ succeeds in correctly detecting the two clusters without any misclassification, as shown in the left part of figure 4. Limiting the analysis to this selected data set, affinity propagation identifies two different clusters, and is unable to divide CHB from JPT, still yielding three misclassifications, as viewed in the right panel of figure 4.

The cluster structure found is controlled both by the maximum depth D and by λ, the two input parameters. In fact, in the extreme case $D = 1$ all samples in the data set would be forced to point to the root, representing a single cluster. The next step is $D = 2$ and corresponds to affinity propagation. This allows us to identify more than trivial clusters, as in the previous case, but, as we said, still imposes a strong constraint, which, in general, may be not representative of the effective data distribution. Increasing D, one can detect more structured clusters, where different elements in the same group do not necessarily share a strong similarity with the same reference exemplar, as has to be the case with affinity propagation and K-means approaches. On the other hand, the possibility of detecting an articulated shape gives some information about the presence of eventual internal notable sub-structures, to be analysed separately, as in our case of the partitioning of two groups into families.

The parameter λ also affects the result of the clustering, in particular as regards the number of groups found. Assigning a large value to this parameter amounts to paying a high cost for every node connected to the root, and so to reducing the number of clusters; on the other hand, decreasing λ will create more clusters. In this first application we used a value of λ comparable with the typical distance between elements, allowing us to detect the four clusters. In this regime, one can expect a competition between the tendencies of the elements to connect to other nodes and to form new clusters with links to the root, allowing the emergence of the underlying structure in the data.

J. Stat. Mech. (2009) P12010

3.2. Clustering of protein data sets

An important computational problem is grouping proteins into families according to their sequence only. Biological evolution lets proteins fall into so-called families of similar proteins—in terms of molecular function—thus imposing a natural classification. Similar proteins often share the same three-dimensional folding structure, active sites and binding domains, and therefore have very close functions. They often—but not necessarily—have a common ancestor, in evolutionary terms. To predict the biological properties of a protein on the basis of the sequence information alone, one needs to be able either to predict precisely its folded structure from its sequence properties or to assign it to a group of proteins sharing a known common function. This second possibility stems almost exclusively from properties conserved through evolutionary time, and is computationally much more tractable than the first one. We want here to underline how our clustering method could be useful for handling this task, in a similar way to the approach that we used in the first application, by introducing a notion of distance between proteins based only on their sequences. The advantage of our algorithm is its global approach: we do not take into account only distances between a couple of proteins at a time, but solve the clustering problem of finding all families in a set of proteins in a *global* sense. This allows the algorithm to detect cases where related proteins have low sequence identity.

To define similarities between proteins, we use the BLAST E-value as a distance measure to assess whether a given alignment between two different protein sequences constitutes evidence for homology. This classical score is computed by comparing how strong an alignment is with respect to what is expected by chance alone. This measure accounts for the length of the proteins, as long proteins have more chance of randomly sharing some subsequence. In essence, if the E-value is 0 the match is perfect, while as the E-value becomes higher the average similarity of the two sequences becomes lower and can eventually be considered as being of no evolutionary relevance. We perform the calculation in a all-by-all approach using the BLAST program, a sequence comparison algorithm introduced by Altshul *et al* [14].

Using this notion of distance between proteins we are able to define a matrix of similarity s in which each entry $s_{i,j}$ is associated with the E-value between protein i and j. The D-MST algorithm is then able to find the directed tree between all the sets of nodes minimizing the same cost function as previously. The clusters that we found are compared to those computed by other clustering methods in the literature, and to the 'real' families of functions that have been identified experimentally.

As in the work by [15], we use the Astral 95 compendium of the SCOP database [16] where no two proteins share more than 95% similarity, so as not to overload the clustering procedure with huge numbers of very similar proteins that could easily be attributed to a cluster by direct comparison if necessary. As this data set is hierarchically organized, we choose to work at the level of superfamilies, in the sense that we want to identify, on the basis of sequence content, which proteins belong to the same superfamily. Proteins belonging to the same superfamily are evolutionarily related and share functional properties. Before going into the detail of the results we want to underline the fact that we do not modify our algorithm to adapt to this data set structure, and without any prior assumption on the data, we are able to extract interesting information on the relative size and number of clusters selected (figure 6). Notably we do not use a training set to

J. Stat. Mech. (2009) P12010

optimize a model of the underlying cluster structure, but focus only on raw sequences and alignments.

One issue that was recently highlighted is the alignment variability [17] depending on the algorithms employed. Indeed some of our results could be biased by errors or dependence of the dissimilarity matrix upon the particular details of the alignments that are used to compute distances, but in the framework of a clustering procedure these small scale differences should stay unseen due to the large scale of the data set. On the other hand, the great advantage of working only with sequences is the opportunity to use our method on data sets where no structure is known *a priori*, such as fast developing metagenomics data sets [18]. We choose as a training set five different superfamilies belonging to the ASTRAL 95 compendium for a total number of 661 proteins: (a) globin-like, (b) EF-hand, (c) cupredoxin, (d) *trans*-glycosidases and (e) thioredoxin-like. Our algorithm is able to identify a good approximation to the real number of clusters. Here we choose the parameter λ well above the typical weight between different nodes, so as to minimize the number of groups found. As a function of this weight you can see the number of clusters found by the D-MST algorithm reported in figure 5, for the depths $D = 2, 3, 4$. In these three plots we see that the real value of the number of clusters is reached for different values of the weight $\lambda \sim 12, 2, 1.4$ respectively. The performance of the algorithm can be analysed in terms of precision and recall. These quantities are combined in the F-value [15] defined as

$$F = \frac{1}{N} \sum_h n_h \max_i \frac{2n_i^h}{n^h + n_i}, \tag{23}$$

where n_i is the number of nodes in cluster i according to the classification λ that we find with the D-MST algorithm, n^h is the number of nodes in the cluster h according to the real cluster classification K and n_i^h is the number of predicted proteins in the cluster i and at the same time in the cluster h. In both cases the algorithm performs better as regards the results for lower values of λ. This could be related to the definition of the F-value because starting to reduce the number of expected clusters may be misleading as regards the accuracy of the predicted data clustering.

Since distances between data points have been normalized to be real numbers between 0 to 1, when $\lambda \to \infty$ we expect to find the number of connected components of the given graph $G(n, s)$. On lowering this value, we start to find some configurations which minimize the weight with respect to the single-cluster solution. The role played by the external parameter λ could be seen as the one played by a chemical potential tuning from outside the average number of clusters.

We compare our results to the ones in [15] for different algorithms and it is clear that intermediate values of D give the best results for the number of clusters detected and the F-value reached without any *a priori* treatment of data. It is also clear that D-MST algorithm with $D = 3, 4, 5$ gives better results than AP (case $D = 2$), as can be seen in figure 7.

We believe that the reason is that clusters do not have an intrinsic spherical regularity. This may be due to the fact that two proteins having a high number of differences between their sequences at irrelevant sites can be in the same family. Such phenomena can create clusters with complex topologies in the sequence space, hard to recover with methods based on a spherical shape hypothesis. We compute the F-value also in the single-linkage

J. Stat. Mech. (2009) P12010

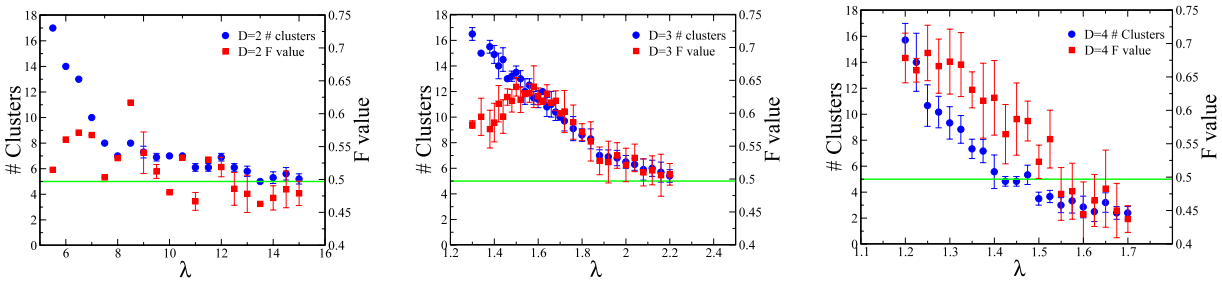

Figure 5. In the three panels we show the average number of clusters over the random noise as a function of the weight of the root for $D = 2, 3, 4$. For each graph we show the number of clusters (circle) and the associated F-value (square), computed as a function of precision and recall. We want to emphasize the fact the highest F-values are reached for depth $D = 4$ and weight $\lambda \sim 1.3$. With this choice of the parameters we found the number of clusters is of order 10, a good approximation of the number of superfamilies shown in the figure as a straight line.

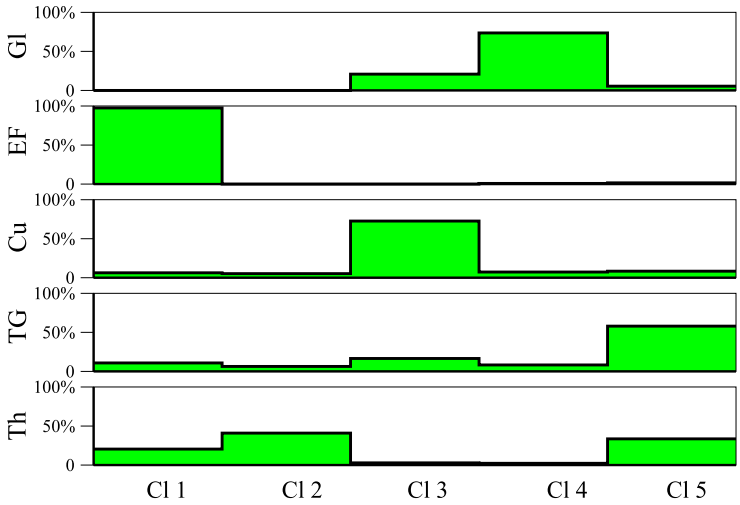

Figure 6. We show the results of clustering proteins for the five subfamilies globin-like (Gl), EF-hand (EF), cupredoxin (Cu), *trans*-glycosidases (TG), thioredoxin-like (Th) using 4-MST with parameter $\lambda = 1.45$. We see that most of the proteins of the first three families (Gl, EF and Cu) are correctly grouped together respectively in clusters 4, 1 and 3 while the last two families are identified with clusters 2 and 5 with some difficulties.

limit ($D > n$) and its value is almost $\sim$0.38 throughout the range of clusters detected. This shows that the quality of the predicted clusters improves, reaching the highest value when $D = 4$, and then decreases when the maximum depth increases.

3.3. Clustering of verbal autopsy data

The verbal autopsy is an important survey-based approach to measuring cause-specific mortality rates in populations for which there is no vital registration system [19, 20].

J. Stat. Mech. (2009) P12010

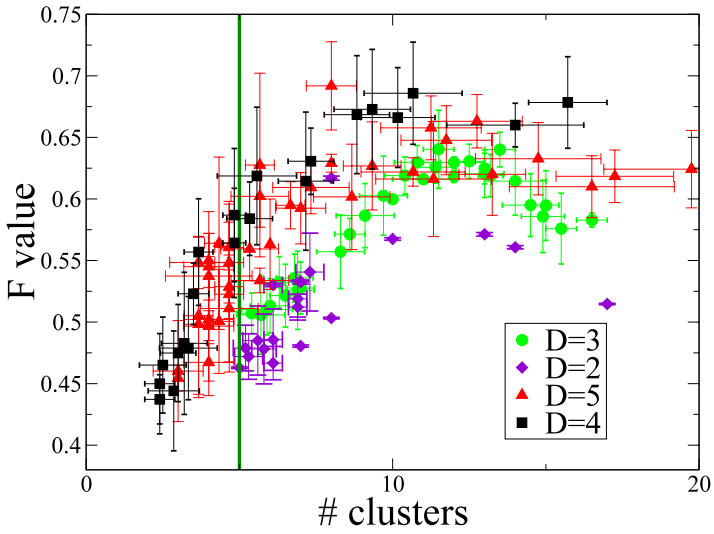

Figure 7. We plot the F-value for depths $D = 2, 3, 4, 5$ as a function of the number of clusters found by the D-MST algorithm. The case $D = 2$ provides the AP results while $D > n$ is associated with SL and gives a value well below 0.4. The highest performance in terms of the F-value is reached for depth $D = 4$ and number of clusters ∼10. We draw a line in correspondence to the presumed number of clusters, which is 5, where again the algorithm with parameter $D = 4$ obtains the highest performance score.

We applied our clustering method to the results of 2039 questionnaires in a benchmark verbal autopsy data set, where gold-standard cause-of-death diagnosis is known for each individual. Each entry in the data set is composed of responses to 47 *yes/no/do not know* questions.

To reduce the effect of incomplete information, we restricted our analysis to the responses for which at least 91% of questions answered yes or no (in other words, at most 9% of the responses were 'do not know'). This leaves 743 responses to cluster (see [19] for a detailed descriptive analysis of the response patterns in this data set.)

The goal of clustering verbal autopsy responses is to infer the common causes of death on the basis of the answers. This could be used in the framework of 'active learning', for example, to identify which verbal autopsies require further investigation by medical professionals.

As in the previous applications, we define a distance matrix on the verbal autopsy data and apply D-MST with different depths D. The questionnaires are turned into vectors by associating with the answers yes/no/do not know the values 0/1/0.5 respectively. The similarity matrix is then computed as the root mean square difference between vectors, $d_{ij} = (1/N)\sqrt{\sum_k (s_i(k) - s_j(k))^2}$, where $s_i(k) \in \{0, 1, 0.5\}$ refers to the symptom $k \in [0, 47]$ in the i th questionnaire.

We first run 2-MST (AP) and 4-MST on the data set and find how the number of clusters depends on λ. We identify a stable region which corresponds to three main clusters for both $D = 2, 4$. As shown in figure 8, with each cluster we can associate a

J. Stat. Mech. (2009) P12010

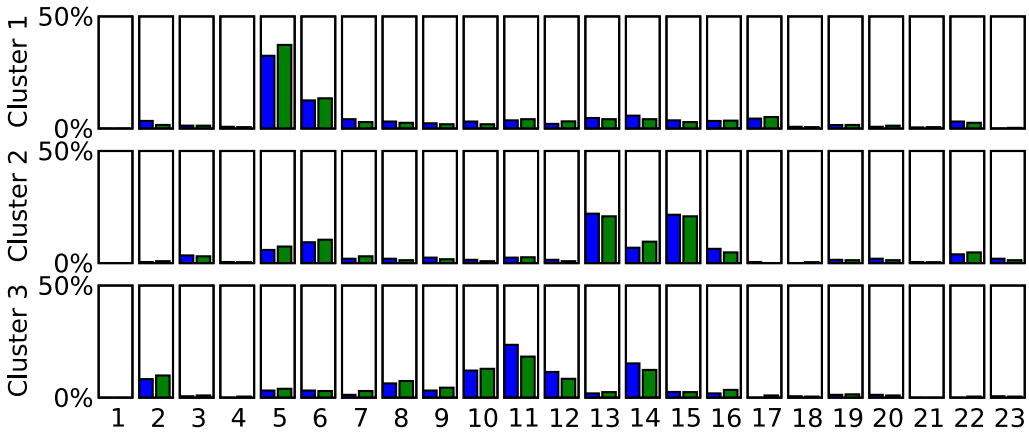

Figure 8. Cluster decomposition broken down by cause of death (from 1 to 23) produced by AP (blue) and D-MST (green). The parameter λ is chosen from the stable region, where the number of clusters is constant.

different cause of death. Cluster 1 contains nearly all of the ischaemic heart disease deaths (cause 5) and about half of the diabetes mellitus deaths (cause 6). Cluster 2 contains most of the lung cancer deaths (cause 13) and chronic obstructive pulmonary disease deaths (cause 15). Cluster 2 also contains most of the additional IHD and DM deaths (30% of all deaths in the data set are due to IHD and DM). Cluster 3 contains most of the liver cancer deaths (cause 11) as well as most of the tuberculosis deaths (cause 2) and some of the other prevalent causes. For $D = 2$ we find no distinguishable hierarchical structure in the three clusters, while for higher value we find a second-level structure. In particular for $D = 4$ we obtain 57–60 subfamilies for values of λ in the region of 0.15–0.20. Although the first-level analysis (figure 8(B)) underlines the similarity of the D-MST algorithm with AP, increasing the depth leads to a finer sub-cluster decomposition [21].

3.4. Conclusion

We introduced a new clustering algorithm which naturally interpolates between partitioning methods and hierarchical clustering. The algorithm is based on the cavity method and finds a bounded depth D spanning tree on a graph $G(V, E)$ where V is the set of n vertices identified with the data points plus one additional root node and E is the set of edges with weights given by the dissimilarity matrix and by a unique distance λ from the root node. The limits with $D = 2$ and n reduce to the well-known AP and SL algorithms. The choice of λ determines the number of clusters. Here we have adopted the same criterion as in [3]: the first non-trivial clustering occurs when the cluster number is constant for a stable region of λ-values.

Preliminary applications to three different biological data sets have shown that it is indeed possible to exploit the deviation from the purely $D = 2$ spherical limit to gain some insight into the data structures. Our method has properties which are of generic relevance for large scale data sets, namely scalability, simplicity and parallelizability. Work is in progress to systematically apply this technique to real world data.

Acknowledgments

The work was supported by a Microsoft External Research Initiative grant. SB acknowledges MIUR grant 2007JHLPEZ.

References

[1] Jain A K, Murty M N and Flynn P J, *Data clustering: a review*, 1999 *ACM Comput. Surv.* **31** 264
[2] Leone M, Sumedha S and Weigt M, *Clustering by soft-constraint affinity propagation: applications to gene-expression data*, 2007 *Bioinformatics* **23** 2708
[3] Frey B J J and Dueck D, *Clustering by passing messages between data points*, 2007 *Science* **315** 972
[4] Eisen M B, Spellman P T, Brown P O and Botstein D, *Cluster analysis and display of genome-wide expression patterns*, 1998 *Proc. Nat. Acad. Sci.* **95** 14863
[5] Bayati M, Borgs C, Braunstein A, Chayes J, Ramezanpour A and Zecchina R, *Statistical mechanics of Steiner trees*, 2008 *Phys. Rev. Lett.* **101** 37208
[6] Braunstein A and Zecchina R, *Learning by message passing in networks of discrete synapses*, 2006 *Phys. Rev. Lett.* **96** 30201
[7] Bayati M, Braunstein A and Zecchina R, *A rigorous analysis of the cavity equations for the minimum spanning tree*, 2008 *J. Math. Phys.* **49** 125206
[8] Kruskal J B, *On the shortest spanning subtree of a graph and the traveling salesman problem*, 1956 *Proc. Am. Math. Soc.* **7** 48
[9] Eisen M B, Spellman P T, Brown P O and Botstein D, *Cluster analysis and display of genome-wide expression patterns*, 1998 *Proc. Nat. Acad. Sci.* **95** 14863
[10] Barla A, Jurman G, Riccadonna S, Merler S, Chierici M and Furlanello C, *Machine learning methods for predictive proteomics*, 2008 *Brief Bioinform.* **9** 119
[11] Dougherty E R, Barrera J, Brun M, Kim S, Cesar R M, Chen Y, Bittner M and Trent J M, *Inference from clustering with application to gene-expression microarrays*, 2002 *J. Comput. Biol.* **9** 105
[12] Gao X and Starmer J, *Human population structure detection via multilocus genotype clustering*, 2007 *BMC Genet.* **8** 34
[13] The International HapMap Consortium, *A second generation human haplotype map of over 3.1 million snps*, 2007 *Nature* **449** 851
[14] Altschul S F, Gish W, Miller W, Myers E W and Lipman D J, *Basic local alignment search tool*, 1990 *J. Mol. Biol.* **215** 403
[15] Paccanaro A, Casbon J A and Saqi M A S, *Spectral clustering of protein sequences*, 2006 *Nucleic Acid Res.* **34** 1571
[16] Murzin A G, Brenner S E, Hubbard T and Chothia C, *Scop: a structural classification of proteins database for the investigation of sequences and structures*, 1995 *J. Mol. Biol.* **247** 536
[17] Wong K M, Suchard M A and Huelsenbeck J P, *Alignment Uncertainty and Genomic Analysis*, 2008 *Science* **319** 473
[18] Venter J C, Remington K, Heidelberg J F, Halpern A L and Rusch D, *Environmental genome shotgun sequencing of the Sargasso sea*, 2004 *Science* **304** 66
[19] Murray C J L, Lopez A D, Feehan D M, Peter S T and Yang G, *Validation of the symptom pattern method for analyzing verbal autopsy data*, 2007 *PLoS Med.* **4** e327
[20] King G and Lu Y, *Verbal autopsy methods with multiple causes of death*, 2008 *Stat. Sci.* **23** 78
[21] Bradde S, Braunstein A, Flaxman A and Zecchina R, in progress