Advanced Clustering Algorithm for Gene Expression Analysis using Statistical Physics Methods

Omer Barad

M.Sc Thesis submitted to the Scientific Council of the
Weizmann Institute of Science

Research conducted under the supervision of
Prof. Eytan Domany

December 2003

Acknowledgments

I wish to thank my supervisor, Professor Eytan Domany, for his devoted guidance, advices, support and endless patience throughout the course of this research.

I am grateful to the member of 'Domany's group' and especially to Uri Einav, Gaddy Getz, Shiri Margel, Noam Shental and Ilan Tsafrir for fruitful discussions.

My deep gratitude goes to my wife, Efrat, and the boys for their support, love and understanding.

Contents

Abstract

We present a mean field solution to the inhomogeneous ferromagnetic Potts model. We study the model's phase transition and show that within mean field, as a result of the model's inhomogeneity, each region in space has its unique critical temperature. We estimate the local density function of spins in space using the local critical temperatures. We present a novel clustering algorithm - DEC, which generates a hierarchial clustering solution from the local density functions of the data points on varying length scales. DEC makes no assumptions on the clusters' structure and is more robust than other non-parametric clustering methods. We demonstrate how the use of DEC can improve gene expression clustering analysis.

Chapter 1

Outline

The outline of this work is as follows:

1. Introduction - the clustering problem and gene expression.

   We start with an overview of gene expression and the central role of clustering in their analysis, then introduce the clustering problem and different approaches to solving it. A general definition of a cluster is suggested, as a region of points with local density higher then the surrounding points. To identify such a region, we need to calculate the local density of points in various regions of space. The problems of calculating the density function and in particular, the effect of noise in the data, are discussed. The main idea is to use statistical physics in order to estimate the density function in a very robust way.

2. Estimation of the density using statistical physics.

   Brief overview of the ferromagnetic Potts model and the relation between order-disorder phase transitions and estimation of the density. We define local disorder-order phase transitions in Inhomogeneous Potts models which are related to the local density. The local phase transitions of Inhomogeneous Potts models are calculated using the following approximations:

   - Monte Carlo (the method used by the existing version of SuperParamagnetic Clustering, SPC).

- Mean Field
- Approximate Mean Field

3. Description of the new version of the SPC algorithm, using the mean field approximation. We compare the results of the different approximations to synthetic and real data examples. Then we discuss the main problems of SPC algorithm, mainly the major influence of the parameter K (the number of neighbors with which a spin interacts) on the SPC results, which rise the idea of suggesting new clustering algorithm.

4. Description of the new algorithm - DEC, we replace K by a different parameter, a length scale - a. The main idea is that each cluster has a specific a which best distinguishes between the typical distances in the cluster and the typical distances between the cluster's surface and the surrounding points; for the optimal a the cluster has it's maximal stability. By running the clustering algorithm with different values of a we obtain a dendrogram for each a; scanning all the dendrograms we find for each cluster its optimal a and define each cluster as the one obtained for it's optimal a. Using this approach we get a general clustering algorithm whose results are not sensitive to any arbitrarily tuned parameters, but which remains sensitive to probing the data at different length scales.

5. We test DEC on representative gene expression data sets.

6. Appendix: Gene expression analysis tools - CTWC. Description of the algorithm, focusing on the improvements in CTWC due to this work.

Chapter 2

Introduction

Cluster analysis is an important technique in exploratory data analysis, where a priori knowledge of the distribution of the observed data is not available. Clustering methods, that divide the data according to the natural classes present in it, have been used in a variety of scientific disciplines and engineering applications. Recently, clustering methods have extensively been used in analyzing biological data, especially from DNA microarraies [12] measurements.

DNA microarray technologies enable monitoring simultaneously the expression level of thousands of genes. This allows a global view on the transcription levels of many genes when the cell undergoes specific conditions or processes. The potential of this technologies for functional genomics is tremendous: Measuring gene expression levels in different developmental stages, different body tissues, different clinical conditions etc. is instrumental in understanding genes function, gene networks, biological processes, effects of medical treatment, in the discovering of new diseases and it is also used as a diagnostic tool.

A key step in the analysis of gene expression data is the identification of group of genes or conditions. Grouping together genes that manifest similar expression patterns over several conditions into clusters helps in revealing relations between genes and their function. Grouping together conditions that manifest similar expression patterns over several genes helps in discovering common biological processes.

The problem of clustering can be formally stated as follows. Determine the partition of N given patterns $\{v_i\}_{i=1}^{N}$ into groups, called *clusters*, such that the patterns of a cluster are more similar to each other than to patterns in different clusters. It is assumed that each pattern v_i is represented by a point $\overrightarrow{x}_i$ in a D-dimensional metric space; the distance between points $d_{ij} = |\overrightarrow{x}_i - \overrightarrow{x}_j|$ is the measure of the dissimilarity between patterns v_i and v_j.

The two main approaches to clustering are called *parametric* and *non-parametric*. In parametric approaches some knowledge of the clusters' structure is assumed, mostly that the clusters are compact; for instance, each cluster can be parameterized by a center around which the points that belong to it are spread with a locally Gaussian distribution. The classical representatives for this approach are fitting Gaussian mixtures and the K-means algorithm.

In many cases of interest, however, there is no a priori knowledge about the data structure. Therefore we prefer to adopt a non-parametric approach and suggest a general definition of clusters. A cluster will be defined as a region of points in space, in which the local density is higher than in it's surrounding region. To identify such a region, we need to calculate the local densities of points in space, assuming that the N given points constitute a sample of reasonable size, to use this local density function as an estimate of the distribution function of the observed data. The density of points is governed by two factors: (a) the typical distance between nearest neighbors, and (b) the number of nearest neighbors of a point, indicative of the dimension in which the points are embedded. A typical example for a popular non-parametric algorithm is the Single Linkage version of Agglomerative Hierarchical Clustering [20]. This algorithm ignores part (b) of the density definition, and gives a hierarchical clustering affected only by the distances between nearest neighbors points. Because of this the algorithm is very sensitive to noise. In high dimensions there is considerable probability that points be placed by chance on a quasi one dimensional string with relatively short distances between the points. Single Linkage

will not distinguish between this string and points of a dense, high dimensional region.

Calculating the local density function is a complex task. In a typical clustering problem we have a dilute background of points with varying densities in a high dimensional space. It is clear that in each region we need to work at a specific resolution, *i.e.* at a length scale that reflects the typical distances between points, in order to obtain the correct estimation for the distribution function. The naive approach for calculating local densities at a specific resolution, a, is to count for each region the number of points in a sphere with radius a and divide it to the sphere volume. This method, however, suffers from large fluctuations especially in high dimensions; for example, it produces variance in the local densities measured for uniformly distributed data points.

The main idea of this work is to use statistical physics, more precisely the physical properties of a ferromagnetic system, in order to estimate the local density function in a very robust way, to get a better signal to noise ratio. We will present two algorithms: (a) a new version of SPC, Super-Paramagnetic Clustering, [3] based on mean field approximation, which is much more efficient than the presently used, Monte Carlo based implementation. This algorithm yields a hierarchical clustering solution (dendrogram), with resolution controlled by a single parameter that combines the effects of length and dimensionality. (b) a novel clustering algorithm - DEC, which gives a hierarchial clustering solution along two separate axes: length and dimensionality. Although we present a general clustering algorithm which can be applied to any applications, in this work we are focusing on implement it on gene expression data.

The outline of this work is as follows. The estimation of the local density function using statistical physics is described in chapter 3. The new version of SPC is presented in chapter 4. The new clustering algorithm, DEC, is described in chapter 5. In chapter 6 we analyze gene expression data examples to demonstrate the main features of the method, and compare its performance with other techniques.

Chapter 3

Density estimation using statistical physics of ferromagnetic system

To estimate the local density of points $\vec{x}_i, i = 1, ...N$, we place at every $\vec{x}_i$ a Potts spin s_i and introduce ferromagnetic couplings between neighboring spins. We now present a brief review of the physical properties of homogenous Potts ferromagnets and our novel Mean Field approximation for inhomogenous Potts models.

3.1 The Potts model

We now briefly describe the physics of the Potts model, focus on its ferromagnetic-paramagnetic phase transition and explain its relation to density.

Ferromagnetic Potts models have been extensively studied for many years [26]. The basic spin variable s can take one of q integer values: $s = 1, 2, ...q$. In a magnetic model the Potts spins are located at points v_i that reside on (or off) the sites of some lattice. Pairs of spins associated with points i and j are coupled by an interaction of strength $J_{ij} > 0$. Denote by $\mathcal{S}$ a configuration of the system, $\mathcal{S} = \{s_i\}_{i=1}^N$. The energy of such configuration is given by the Hamiltonian

$$\mathcal{H}(\mathcal{S}) = -\sum_{<i,j>} J_{ij}\delta_{s_i,s_j} \tag{3.1}$$

Where the notation $< i, j >$ stands for neighboring sites v_i and v_j. In order to calculate the thermodynamic average of a physical quantity A at a temperature T (in k_B units),

one has to calculate the sum

$$\langle A\rangle = \sum_{\mathcal{S}} A(\mathcal{S})\rho(\mathcal{S}) \tag{3.2}$$

where $\rho(\mathcal{S})$ plays the role of the probability density which gives the statistical weight of each spin configuration $\mathcal{S}$. Thermal equilibrium is characterized by $\rho(\mathcal{S})$ that minimizes the free energy, which is the sum of two parts, energy and entropy:

$$F = \langle\mathcal{H}\rangle - T\langle S\rangle = \sum_{\mathcal{S}} \mathcal{H}\rho + T\sum_{\mathcal{S}} \rho\log\rho \tag{3.3}$$

F is minimized by the Boltzmann factor,

$$\rho(\mathcal{S}) = \frac{1}{Z}e^{-\frac{\mathcal{H}(\mathcal{S})}{T}}, \tag{3.4}$$

where Z ,the partition function, is a normalization constant, $Z = \sum_{\mathcal{S}} e^{-\frac{\mathcal{H}(\mathcal{S})}{T}}$.

3.2 Homogenous Potts Model

First we look at the simple homogeneous Potts model: the spins reside on the site of a D dimensional hypercubic lattice with distance a between nearest neighbor sites, with periodic boundary conditions. The interaction between two spins is $J > 0$ if they are nearest neighbors and 0 otherwise. A simple generalization is to allow further neighbor interactions that are a decreasing function of the distance $d_{ij} \equiv d(v_i, v_j)$.

We are interested in the properties of the thermal average of the magnetization, the order parameter of this ferromagnetic system. The magnetization of specific configuration is defined as

$$m(\mathcal{S}) = \frac{qN_{max}(\mathcal{S}) - N}{(q-1)N} \tag{3.5}$$

with

$$N_{max}(\mathcal{S}) = \max\{N_1(\mathcal{S}), N_2(\mathcal{S}), ...N_q(\mathcal{S})\}, \tag{3.6}$$

where $N_\mu(\mathcal{S})$ is the number of spins with the value μ; $N_\mu(\mathcal{S}) = \sum_i \delta_{s_i,\mu}$.

Another quantity of interest is the thermal average of δ_{s_i,s_j}, the spin-spin correlation function,

$$G_{ij} = \langle \delta_{s_i,s_j} \rangle \tag{3.7}$$

which is the probability of the two spin s_i and s_j to be aligned. Assuming $N_\mu(\mathcal{S})$ of the non most occupied states is equal we find that the relation between the correlation and the magnetization is

$$G_{ij} \geq \frac{(q-1)m^2 + 1}{q} \tag{3.8}$$

where the equality exist when the distance between spin i and j is much larger then the correlation length.

The competition between energy and entropy is clear from (eq. 3.3 and 3.4). At high temperatures the entropy is the dominant term; at $T = \infty$ each configuration has the same probability and the system is paramagnetic or disordered; $\langle m \rangle = 0$.

As the temperature is lowered the energy becomes the dominant term, and configurations with low energy, with most spins aligned, have high probability; the system undergoes transition to an ordered, ferromagnetic phase. In first order transitions the magnetization jumps from zero to $\langle m \rangle \neq 0$.

Direct evaluation of sums like (eq. 3.2) is impractical, since the number of configurations $\mathcal{S}$ increases exponentially with the system size N. We will use a mean field approximation to demonstrate the main features of the ferromagnetic-paramagnetic phase transition, The main idea of mean field is to perform the partition sum for each spin individually, with only one spin "active" at a time, while all the others are "held fixed": their interaction with the active spin produces an external field on it. Using the variational method, see for example [7], we substitute in eq. (3.3) a trial probability density

$$\rho_t(\mathcal{S}) = \frac{1}{Z_t} e^{-\frac{\mathcal{H}_0(\mathcal{S})}{T}}, \tag{3.9}$$

$$\mathcal{H}_0 = -\sum_i \sum_j J\langle m\rangle \delta_{s_i,q_0} \tag{3.10}$$

where q_0 is the most occupied state. For a homogenous model all sites are equivalent and hence $\frac{q\langle\delta_{s_i,q_0}\rangle-1}{q-1} = \langle m\rangle$ for every i. Denoting by $\sum_j \equiv \zeta$ the number of nearest neighbors and by $\alpha \equiv \frac{J\zeta\langle m\rangle}{T}$, the trial density function takes the form

$$\rho_t = \prod_i \frac{e^{\alpha\delta_{s_i,q_0}}}{(q-1)+e^{\alpha}} \tag{3.11}$$

Substituting ρ_t in (eq. 3.3) we get

$$\frac{F_t}{N} = -\frac{J\zeta}{2}\left(\frac{(q-1)+e^{2\alpha}}{((q-1)+e^{\alpha})^2}\right) + T\left(\frac{\alpha e^{\alpha}}{(q-1)+e^{\alpha}} - \log((q-1)+e^{\alpha})\right). \tag{3.12}$$

Minimizing F_t with respect to the model parameter α, we get the self-consistent mean field equation

$$\alpha = \frac{1}{T}\frac{J\zeta(e^{\alpha}-1)}{e^{\alpha}+(q-1)} \tag{3.13}$$

At $T >> J$ (near infinite temperature) there is only one, paramagnetic solution, $\alpha = 0$. For $T \to 0$ the solution is $\alpha \approx \frac{J\zeta}{T}$ or $\langle m\rangle \approx 1$, *i.e.* we have a fully aligned ferromagnet. Hence there is a transition at some T, which, for Potts models with $q > 2$ the ferromagnetic-paramagnetic phase transition is of first order. Within our mean field approximation the free energy has, near the transition, *two* minima, one at $\langle m\rangle = 0$ and the other at some $\langle m\rangle \neq 0$. At the transition temperature T_c the values of the free energy at the two minima become equal. Thus, in order to find the critical temperature and critical magnetization we add a second equation:

$$F(\alpha_c, T_c) = F(0, T_c) \tag{3.14}$$

solving these equations and changing variables from α to $\langle m\rangle$ we get

$$m_c = \frac{q-2}{q-1} \tag{3.15}$$

$$T_c = \frac{1}{2\log(q-1)} m_c J\zeta \tag{3.16}$$

We can see that the discontinuity of m increases with q and the transition (from $m_c \to 1$ to zero) becomes sharper and more strongly first order. The relation between T_c and the density of the points at which the spins reside is now clear: the critical temperature is proportional to $J\zeta$, where the strength of J is determined by the nearest neighbors' distances a, and the number of interacting neighbors $\zeta = 2D$. Increasing the local density of lattice points, by either reducing a, increasing the dimension of the lattice or the range of the interactions, induces increase of the critical temperature. This is the basis of our method, discussed below, of using hence T_c to estimate density.

The approximations inherent in mean field neglect fluctuations of the spins that provide the "mean field", and the effect of the state of the active spin on the field it feels. The approximation's accuracy depends on the model dimension and the number of interacting neighbor spins [8]. Since the higher the number of interacting neighbors spins, lower is the influence of a few individual spins' fluctuations, we expect to get better accuracy from the mean field approximation when the density increases. In particular, the mean field estimation to the critical temperature, which is in general higher than the true value, is expected to approach the true transition temperature as the model's dimension increases.

3.3 Inhomogeneous Potts model

We are interested, however, in a finite system where the local density varies, *i.e.* the typical distances between points and also the local dimensionality change. We model such a system by an inhomogeneous Potts model. In inhomogeneous Potts models there is no equivalence of all sites, since the symmetry of the homogeneous model is broken; each spin i has different interactions J_{ij} and the assumption of one order parameter, the global magnetization, is no longer valid. In the case typical for the clustering problem there is no symmetry in the model, and we define as an *order parameter for each spin* it's

magnetization

$$\langle m_i \rangle = \frac{q\langle \delta_{s_i,q_{i0}} \rangle - 1}{q-1} \tag{3.17}$$

where q_{i0} is the most occupied state in the region of spin i. We look for local disorder-order phase transitions in the system, which makes the picture much more complicated. Think, for example, of a system composed of two similar, very dense region of spins, separated by a dilute region. The spins of the two dense regions will have a phase transition at T_c^1, causing their magnetizations to jump from $\langle m_i \rangle = 0$ to $\langle m_i \rangle \approx 1$ (for large q). These two ordered regions will act like a field on the interface with the dilute region, causing a (usually small) jump in the magnetization of its spins. At a lower temperature, T_c^2, the dilute region's spins will undergo a phase transition "of their own", with their magnetization jumping to ≈ 1. This picture may of course be richer, as the spins' distribution in space may be more complicated. The outcome is that the spins' magnetization, instead of having a clear jump only at a single phase transition, may undergo many jumps due to phase transitions in the neighborhood and to their own phase transition. We will identify the temperature at which the largest jump in the local magnetization occurs as *the local critical temperature*. This local critical temperature will be our estimate of the local density.

First we review how this model is solved using Monte Carlo simulations, which yield the best approximation to the exact solution. Then we introduce our solution that uses a mean field approximation, which is much more efficient computationally than Monte Carlo, and compare it's accuracy with the Monte Carlo results.

3.3.1 Monte Carlo simulation of Potts models

As mentioned, direct evaluation of sums such as eq. (3.2) is impractical, since the number of configurations $\mathcal{S}$ increases exponentially with the system size N. Monte Carlo simulation methods overcome this problem by generating a characteristic subset of configurations

which are used as a statistical sample. They are based on the notion of *importance sampling*, in which a set of spin configurations $\{\mathcal{S}_1, \mathcal{S}_2, ...\mathcal{S}_M\}$ is generated according to the Boltzmann probability distribution. Then, the thermodynamic average is reduced to a simple arithmetic average

$$\langle A \rangle \approx \frac{1}{M} \sum_{i}^{M} A(S_i) \tag{3.18}$$

We will use the Swendsen-Wang [27] Monte Carlo algorithm (SW) to generate those M configurations we will use the two-points connectedness c_{ij}

$$c_{ij} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ belong to the same SW-cluster} \\ 0 & \text{otherwise} \end{cases} \tag{3.19}$$

and calculate the spin-spin correlation G_{ij} from the relation

$$G_{ij} = \frac{(q-1)\langle c_{ij} \rangle + 1}{q} \tag{3.20}$$

Further details regarding the exact Monte Carlo algorithm appear in paper by Blatt *et al.* [5].

In order to use Monte Carlo to estimate local T_c we identify for each interaction its critical temperature, $T_{c_{ij}}$, as the temperature in which the larger jump in its correlation function occur and then identify for each spin i its critical temperature as $\max(T_{c_{ij}})$.

3.3.2 Mean field of Potts models

The inhomogeneous Potts model Hamiltonian is defined in (eq. 3.1). Using the variational method we write a model Hamiltonian

$$\mathcal{H}_0 = -\sum_i \sum_j J_{ij} \langle m_j \rangle \delta_{s_i, q_0} \tag{3.21}$$

and use

$$\rho_t(\mathcal{S}) = \frac{1}{Z_t} e^{-\frac{\mathcal{H}_0(\mathcal{S})}{T}}, \tag{3.22}$$

as the trial probability density. We assume that the most occupied state in all the regions in space have the same direction q_0, this is a reasonable assumption since the entropy the system can gain from different most occupied states is at maximum $q \log q$, which is not extensive, moreover this quantity is decreasing with the temperature, thus has low influences on the free energy. Setting $\alpha_i \equiv \frac{\sum_j J_{ij}\langle m_j \rangle}{T}$ we obtain the trial density function

$$\rho_t = \prod_i \frac{e^{\alpha_i \delta_{s_i,q_0}}}{(q-1)+e^{\alpha_i}} \tag{3.23}$$

The trial free energy is given by

$$F_t = \sum_{\mathcal{S}} \mathcal{H}\rho_t + T \sum_{\mathcal{S}} \rho_t \log \rho_t \tag{3.24}$$

where

$$\sum_{\mathcal{S}} \mathcal{H}\rho_t = -\sum_{<i,j>} J_{ij} \frac{(q-1)+e^{\alpha_i+\alpha_j}}{((q-1)+e^{\alpha_i})((q-1)+e^{\alpha_j})} \tag{3.25}$$

$$\sum_{\mathcal{S}} \rho_t \log \rho_t = \sum_i [\frac{\alpha_i e^{\alpha_i}}{(q-1)+e^{\alpha_i}} - \log((q-1)+e^{\alpha_i})] \tag{3.26}$$

Minimizing F_t with respect to α_i yields the mean field equations for each site $i = 1, 2, ...N$:

$$\alpha_i = \frac{1}{T} \sum_j J_{ij} \frac{e^{\alpha_j}-1}{(q-1)+e^{\alpha_j}} \tag{3.27}$$

which become upon changing variables to m_i

$$m_i = \frac{1-e^{-\frac{\sum_j J_{ij}m_j}{T}}}{1+(q-1)e^{-\frac{\sum_j J_{ij}m_j}{T}}} \tag{3.28}$$

We solve these N non linear coupled equations iteratively for each temperature to obtain the magnetization of each spin as a function of the temperature. We are interested in the global minimum of the free energy. We start from $T = 0$, where the solution is known: $m_i = 1$ for every i. Then we heat the system slowly, moving to the next (slightly higher) temperature; the solution at the previous temperature is the starting point for the iterative search for the new minimum. Note that the free energy increases with increasing

T. For small changes of the temperature one normally expects small changes of the local magnetizations that minimize the free energy at the new T. However, as a transition is approached, a new (local) minimum $\{m_i'\}$ appears; initially it's free energy is higher, but as we pass the transition temperature, it's free energy becomes the global minimum. Since our search follows the "old" minimum $\{m_i\}$, from this point on we are stuck in a local minimum of the free energy function, fig. 3.1. To avoid this, we use the fact that as the temperature increases such a local minimum disappears at some "spinodal" T_s. So if we had $T_1 < T_s$ and $T_2 > T_s$ up to T_1 we followed adiabatically one minimum, $\{m_i\}$, and at T_2 suddenly our solution "spills over" to the true global minimum $\{m_i'\}$. In this case $F[m'(T_2)] < F[m(T_1)]$ and hence the easiest way to observe that this has happened is to lower the temperature to its previous value, T_1', redo the minimization, but starting from $\{m_i'\}$, and compare the new value of the free energy, $F[m'(T_1)]$, to the previous value $F[m(T_1)]$. If we were following the same solution, $m_i' = m_i$ and the two free energies must be equal. If we find that $F[m'(T_1)] < F[m(T_1)]$, this means that while raising the temperature from T_1 to T_2 we left one branch of solutions and spilled over to a different one of lower free energy, *i.e.* we have been following for a while a local minimum. In this case we use the solution of lower free energy and slowly lower the temperature, retracing our path, and identifying the temperature at which the free energies of the two solutions ($\{m_i(T)\}$ obtained by heating and $\{m_i'(T)\}$, obtained by cooling) cross. After every heating step we perform such a check.

The procedure outline above is summarized in the following scheme:

1. General Initialize:

   $m_i(0) = 1$

   $F(0) = F(m_i(0), 0)$

   $F(T) = \infty, \qquad \forall T > 0$

   $T = 0$

   $Tnext = Tstep$

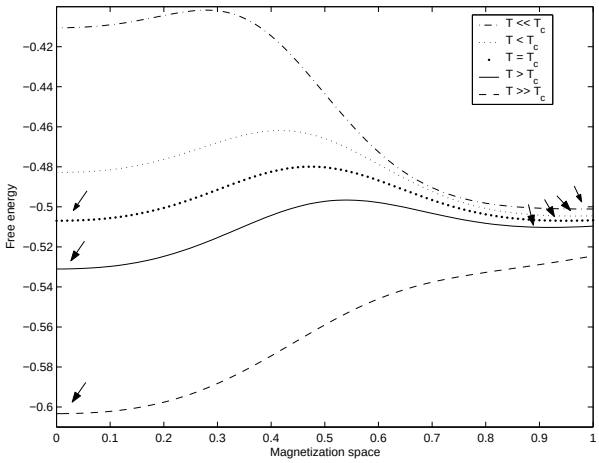

Figure 3.1: Sketch of the free energies as a function of the magnetizations, $m_1 ... m_N$, for different temperature, we start from the global minimum at $T \ll T_c$, marked by arrow, and follow this minimum while increasing the temperature, from $T > T_c$ we stack in a local minimum until it disappears at $T \gg T_c$.

2. Initialize:

   $m_i^0 = m_i(T)$

   $T = Tnext$

3. Iterative step:

   $$m_i^{n+1} = \frac{1-e^{-\frac{\sum_j J_{ij} m_j^n}{T}}}{1+(q-1)e^{-\frac{\sum_j J_{ij} m_j^n}{T}}}$$

4. Stopping condition:

   if $\max\{m_i^n - m_i^{n-1}\} > Threshold$ move to step 3

5. Setting:

   if $F(m_i^n, T) < F(T)$

   (a) $m_i(T) := m_i^n$

   (b) $F(T) := F(m_i(T), T)$

   (c) $Tnext := T - Tstep$

   else $Tnext := T + Tstep$ move to step 2.

3.3.3 Approximation of Potts models Mean field

Finally we present a very simple way to approximate the local critical temperature, which gives us an intuition of the statistical physics effect on the density estimation. This approximation uses a simplified local magnetization function and assumes that it behaves like the magnetization of the homogeneous Potts model's mean field solution in the large q limit, *i.e.* as a step function

$$\langle m_i(T) \rangle = \begin{cases} 1 & T \leq T_{c_i} \\ \\ 0 & T > T_{c_i} \end{cases} \tag{3.29}$$

those magnetizations, of course, does not obey the mean field equations (3.28) but we can use them with a little manipulation to estimate the local critical temperature. At the critical temperature of spin i, T_{c_i}, the local magnetization m_i jumps from 0 to 1, i.e. these two solutions coexist. We approximate this fact by assuming that $m_i[T_{c_i}] = \frac{1}{2}$ on the left side of eq. 3.28, which then become

$$\frac{1}{2} = \frac{1 - e^{-\frac{\sum_j J_{ij} m_j(T_{c_i})}{T_{c_i}}}}{1 + (q-1)e^{-\frac{\sum_j J_{ij} m_j(T_{c_i})}{T_{c_i}}}} \tag{3.30}$$

Remembering that all the neighbor spins have $m_j(T_{c_i}) = 0$ or 1, depending on whether $T_{c_i} > T_{c_j}$, after a little algebra this becomes

$$T_{c_i} = \frac{1}{\ln(q+1)} \sum_j J_{ij} \begin{cases} 1 & T_{c_i} \leq T_{c_j} \\ \\ 0 & T_{c_i} > T_{c_j} \end{cases} \tag{3.31}$$

We solve those N self consistent coupled equation for T_{c_i} using the following iterative process:

1. Initialize:

   $T_{c_i}^0 = \infty$

2. Iterative step:

   $$T_{c_i}^{n+1} = \frac{1}{\ln(q+1)} \sum_j J_{ij} \begin{cases} 1 & T_{c_i}^{n+1} \leq T_{c_j}^n \\ 0 & T_{c_i}^{n+1} > T_{c_j}^n \end{cases}$$

3. Stoping condition:

   if $\max\{T_{c_i}^{n+1} - T_{c_i}^n\} > Threshold$ move to step 2

To gain insight into the solution obtained this way, take the following simple form of the couplings J_{ij}:

$$J_{ij} = \begin{cases} 1 & d_{ij} \leq a \\ 0 & d_{ij} > a \end{cases} \tag{3.32}$$

where a is some characteristic length scale. The naive approach of local density estimation near point i at this length scale is to count $N_i(a)$, the number of point that reside in a sphere of radius a centered at i: *i.e.* $N_i(a) = \sum_j J_{ij}$. This is proportional to the value we get for T_{c_i} *in the first iteration*. Using statistical physics, we improve on this estimate by sharing information between points, giving less weight to points in a low density region. In the above example, our solution will count the number of points in the sphere that have the same local density as point i or higher. This makes our method much more robust against noise; when our method is tested on points selected randomly from a uniform distribution, the measured variation of local densities is reduced by a factor of 10.

Chapter 4

Super-Paramagnetic Clustering - SPC

4.1 Description of the algorithm

In the new version of the Super-Paramagnetic Clustering (SPC) algorithm various equilibrium averages are calculated using the mean field approximation instead of Monte Carlo simulations. This replacement has two advantages: running time is much shorter and the algorithm is deterministic. Here we describe briefly the original SPC algorithm (for a detailed description see [5]) and emphasize the differences with the approach introduced here, as they arise. The algorithm has three main stages which can be summarized as follows:

1. **Preprocess - Construct the physical analog Potts model:**

   (a) Associate a Potts spin variable $s_i = 1, 2, ...q$ to each data point; usually we work with $q = 20$.

   (b) Identifying neighbors:

   Whenever the data do not form a regular lattice, we need to give a reasonable definition of (interacting) neighbor spins. We identify the neighbors of each point v_i according to the K mutual nearest neighbors criterion: v_i and v_j are regarded as a neighbors if and only if v_i is one of the K nearest neighbors of v_j and v_j is one of the K nearest neighbors of v_i. Connecting each neighbors

pair with an edge we construct the initial *SPC graph*. Since we want to work with a connected graph [5], we superimpose on the initial SPC graph the edges corresponding to the minimal spanning tree associated with the data. The result is the SPC graph we work with.

(c) Calculate the model coupling constants J_{ij}:

$$J_{ij} = \begin{cases} \frac{1}{\hat{K}} exp(-\frac{d_{ij}^2}{2a^2}) & \text{if } v_i \text{ and } v_j \text{ are neighbors} \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

The "local length scale" a is the average of all distances d_{ij} between neighboring pairs v_i and v_j. $\hat{K}$ is the average number of neighbors. Choosing J_{ij} this way creates strong interactions between spins associated with data from high density regions, and weak interactions between neighbors that are in low density regions.

After these steps we finally have the Potts model Hamiltonian associated with the data:

$$H(S) = -\sum_{<i,j>} J_{ij}\delta_{s_i,s_j} \tag{4.2}$$

2. **Calculating the correlations of neighbor spin pairs:** For each edge $\langle ij \rangle$ of the SPC graph we calculate G_{ij}, the correlation of the two spins, s_i and s_j connected by the edge. We calculate G_{ij} for a range of temperatures $0 \leq T < T_{max}$ with a temperature step $Tstep$. While the original SPC calculates this correlation function using Monte Carlo, we will calculate it by mean field. The results of the two methods are compared in the next section.

   Once the correlations between neighbor spins have been calculated, we estimate the local density function, by calculating the local ferromagnetic - paramagnetic critical temperatures. For each edge $\langle ij \rangle$ of the SPC graph we define a local critical temperature $T_{c_{ij}}$ as that temperature, below which both s_i and s_j are in their

respective local ferromagnetic phases. A simple estimate of this critical temperature is obtained by setting

$$G_{ij}(T_{c_{ij}}) = 0.5. \tag{4.3}$$

An alternative definition of the $T_{c_{ij}}$ is the temperature at which the temperature derivative of the spin-spin correlation function has its maximum. The values of $T_{c_{ij}}$ obtained by the two definitions are very close.

3. **Constructing clusters**; we have two options:

   (a) Building the cores of clusters: Vary the temperature (say, increase) in steps $Tstep$. At each temperature define as "valid" those edges of the SPC graph, whose $T_{c_{ij}} > T$. The connected components of the graph constructed from the valid edges constitute the clusters at the specific temperature at which we work. As the temperature increases the set of clusters changes, from one cluster that contains all the data points at $T = 0$, to each single point being a cluster at high enough temperatures. Since as T increases edges can turn from valid to non-valid (and not vice versus), this procedure yield a tree or dendrogram, *i.e.* a hierarchical clustering solution. As opposed to most agglomerative algorithms, SPC has a natural measure for the relative stability of any particular cluster: the range of temperatures $\Delta T = T_2 - T_1$, that define the "lifetime" of each cluster, from separation from its parent cluster at T_1 to breaking up at $T_2 > T_1$. The more stable the cluster, the larger the range ΔT.

   (b) Directed Growth method: expanding the cluster core and building a larger, less stable cluster. At each temperature step we extend the set of valid edges obtained as described above; for each spin i we identify the edge $\langle ij \rangle$ with maximal $G_{ij}(T)$, and call it also valid. The clusters are defined as before, but using this extended set of valid edges.

4.2 Evaluating the correlation function: comparison of different methods

The main advantage of using the mean field approximations is their shorter running time. We first discuss the computational complexity and efficiency of the different methods.

The first stage of the SPC algorithm, identifying neighbors and calculating the interactions, is of $O(N^2 \cdot \max\{D, \log(N)\})$. This part is common to all the methods we compare and in terms of the N-dependence it dominates the complexity. The typical sizes of gene expression data have N in the range of thousands of genes or data points. The corresponding Potts model is equilibrated, typically, at order 10 temperatures, with a few thousand Monte carlo step at each temperature. The computational complexity of each Monte carlo step is $O(N)$. Under these circumstances, equilibration, using Monte Carlo, takes at least 10 times more time then the first, preparatory stage. The gain of mean field is in its efficiency of the second stage, which we now discuss.

A Monte Carlo step takes $O(N)$ operations; the same is needed for one iteration of the mean field solution. Hence the computational complexity of all the methods is of $O(N\hat{K})$, where $\hat{K}$ is the average number of neighbors for each point. However, the pre factors are very different:

1. Monte Carlo: (Number of cycles) $\cdot$ (number of temperature steps) $\sim$ (thousands) $\cdot$ (tens)

2. Mean field: (number of iterations needed to convergence) $\cdot$ (number of temperature steps) $\sim$ (tens) $\cdot$ (tens)

3. Approximate solution of the mean field equations: (number of iteration to convergence) $\sim$ tens

so practically, for a typical clustering problem the ratio between running time of a Monte Carlo simulation and that of a mean field solution is governed by the ratio of N_{cyc}, the number of MC cycles at each T, to N_{it}, the number of mean field iterations

needed to converge to a solution. This ratio is between 100 to 1000. Using the mean field approximation reduced the running time of SPC by factor of 10, using approximate solution of the mean field equations will not help has more at this stage, thus in this section we compare only the results obtained using Monte Carlo versus mean field.

Using Monte Carlo we calculate the correlation between interacting spins, G_{ij}, in a direct fashion, while using the mean field approximation we can calculate only the magnetization of each spin, m_i. We need a way to relate the spin magnetizations and spin-spin correlation.

The edges of the SPC graph consist of two types: (a) edges due to the K mutual nearest neighbors criterion, and (b) edges from the minimal spanning tree, *mst*. For edges of type (a) we assume that the two neighbors i, j also share other spins as neighbors. Hence the mean field on i is in the same state as on j, and in their magnetized phases we expect s_i and s_j to be in the same most occupied state, in which case eq. 3.8 holds as an equality within the mean field approximation. Using this we get

$$G_{ij_{mf}} = \frac{(q-1)m_{mf}^2 + 1}{q} \tag{4.4}$$

For edges of type (b) this assumption is not always valid. We estimate correlation of two spins connected by an mst edge on the basis of their direct interaction alone. Since the mst edges belong to a tree, they do not belong to a loop, and the correlation of the two spins depends only on the direct interaction and can be calculated analytically, yielding

$$G_{i,j_{mf}} = \frac{1}{1 + (q-1)\exp(-\frac{J_{i,j}}{T})} \tag{4.5}$$

In regions with only mst edges this definition is accurate, in regions with both edges types the meaning of this definition is neglecting of those mst edges.

In the mean field approximation we neglect fluctuations of the spins that produce the mean field. Thus the approximation's quality depends on the model's dimension and connectivity, which determine the number of interacting neighbor spins. The higher the

dimension and the number of interacting neighbors, the lower is the influence of a few spins' fluctuations. In general, since fluctuations reduce the field felt by a spin, we expect the mean field estimation for the local magnetization to be higher then the real value, but this differences is expected to decrease as the dimension and the connectivity increase.

We expect two contradicting effects:

1. For some edges $G_{ij} > G_{ij_{mf}}$, due to direct spin-spin interaction dependencies which are neglected by mean field, causing $T_{c_{ij}} > T_{c_{ij_{mf}}}$. These differences may happen at a cluster's core and have no important influence on the clustering results, which depend on the ability to distinguish between the cluster and the surrounding region.

2. Edges where $T_{c_{ij_{mf}}} > T_{c_{ij}}$, due mainly to regions of low connectivity in the SPC graph; these differences are much more significant. The K mutual nearest neighbors criterion can induce low connectivity even in high dimensional data. For example, if two highly connected regions in the SPC graph are connected via a one dimensional string, for the string's edges we get $T_{c_{ij_{mf}}} > T_{c_{ij}}$, decreasing the algorithm's sensitivity. This problem is noticeable when one of the highly connected regions is more dilute and smaller than the other, in such a case the estimated critical temperature of the string's spins may be raised to the critical temperature of the dilute cluster and our procedure will not distinguish this cluster. We will test and discuss those problem on examples from real data.

After defining the spin-spin correlation in the mean field approximation we compare its result to Monte Carlo. First, we test a very simple inhomogeneous model: a 30×30 square lattice without periodic boundary conditions, with $J_{ij} = 1$ for nearest neighbors and 0 otherwise. We compare the spin-spin correlation, G_{ij} as a function of temperature, for each pair of interacting spins, obtained using Monte Carlo and mean field.

In fig. 4.1 we can see that even in this low connectivity model the mean field approximation gives accurate results for most of the temperature steps and capture qualitatively the overall behavior of the correlation function.

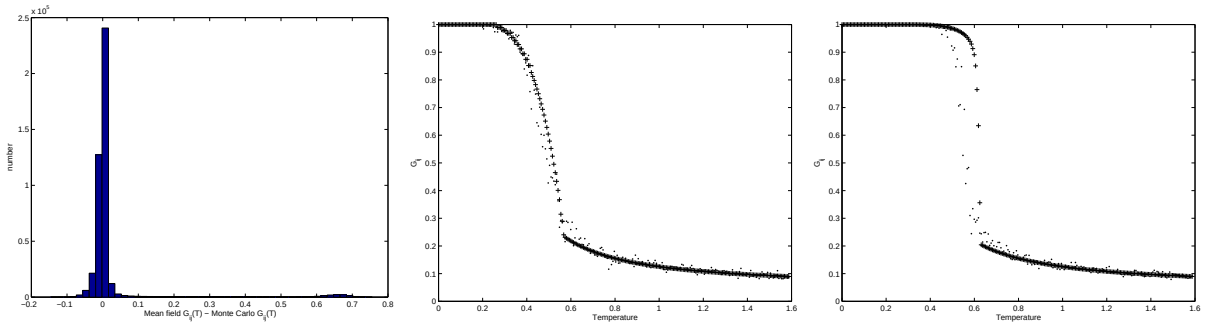

Figure 4.1: Mean field vs. Monte Carlo Comparison on a $2D$ lattice model. On the left, histogram of the differences of spin-spin correlation functions for all interacting pairs, at all temperatures. In the middle and right typical examples of the correlation function between spins on the edge of the lattice and near the edge, respectively; dots represent MC and + mean field.

Now we move to examples with real gene expression data, each experiment monitoring the mRNA expression of many thousands of genes simultaneously over n_s (a few tens) of different samples. First one we filters the genes down to n_g of a few thousands, keeping those genes whose expression levels varied significantly over the different samples, obtaining an $n_g \times n_s$ raw data matrix $\Re$. Then we adopt a standard preprocessing procedure and apply logarithmic transformation to the raw data and normalize each gene, such that its mean vanishes and its norm is one. We compare results obtained for the following representative experiments:

1. Leukemia data [2], with $n_s = 72$ and $n_g = 906$ (see Fig. 4.2).

2. Breast cancer primary tumor tissues [22], with $n_s = 49$ and $n_g = 1572$ (see Fig. 4.4).

We can see from fig. 4.2 that except of the two edges marked with an arrow, the mean field approximation gives very similar correlations to those of Monte Carlo, hence yielding similar hierarchical clustering results. Moreover, this case is an example to two highly connected subgraphs connected via a one dimensional string (fig. 4.3). The effect of mean field's over estimation of the local critical temperatures of the string is that the two

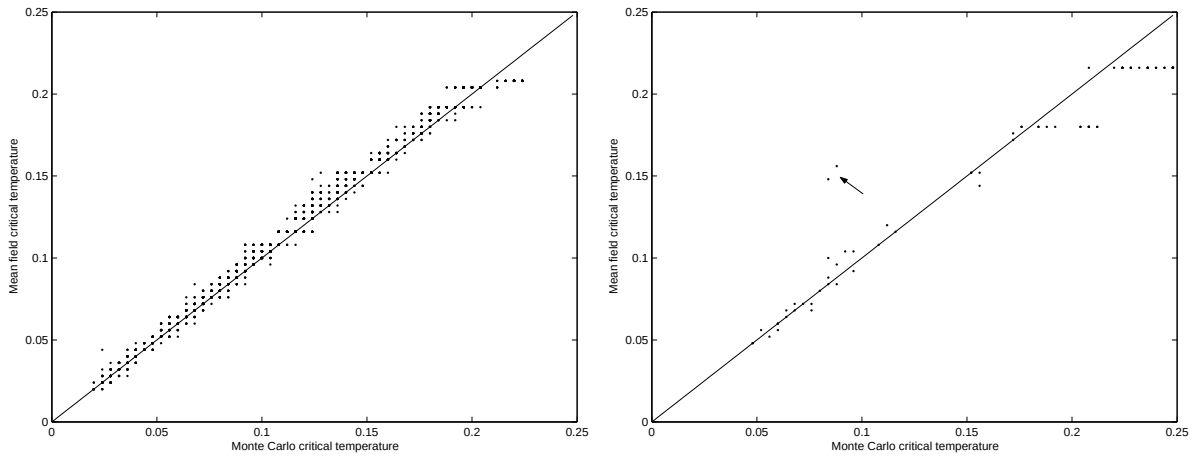

Figure 4.2: Monte Carlo $T_{c_{ij}}$ vs. Mean Field estimated values for the leukemia data [2]. On the left - 906 genes in the space of 72 samples, with ~ 5.5 neighbors, on average, for each gene. On the right - 72 samples in the 906 dimensional space of gene expression, with average ~ 2.33 neighbors for each sample. The major differences, corresponding to edges $31-39$ and $47-39$ (see Fig 4.3), are directed by an arrow.

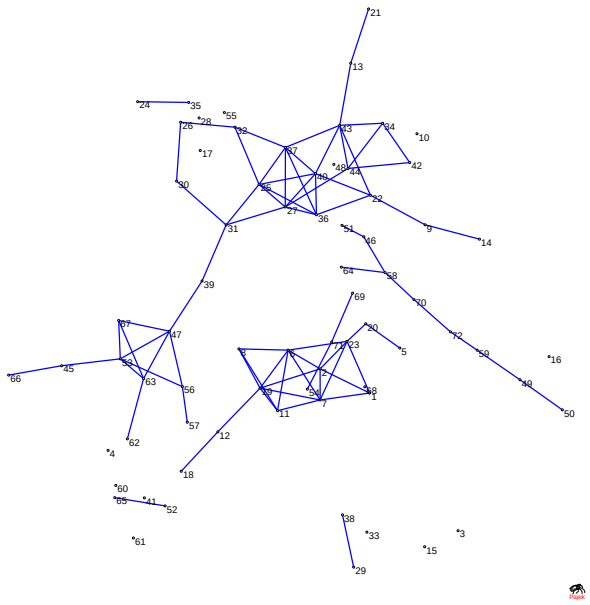

Figure 4.3: The initial SPC graph obtained from the leukemia data [2], for 72 samples, obtained using $K = 4$ (without minimal spanning tree edges). A string of two edges, $(31-39$ and $47-39)$, bridges between two highly connected subgraphs. As expected, for these two edges mean field significantly overestimates the local critical temperatures estimation (points directed by an arrow in Fig. 4.2).

clusters connected by it disassociate from each other at a higher T, thereby reducing of the the two clusters' stability estimates, ΔT, but without losing the ability to distinguish between them.

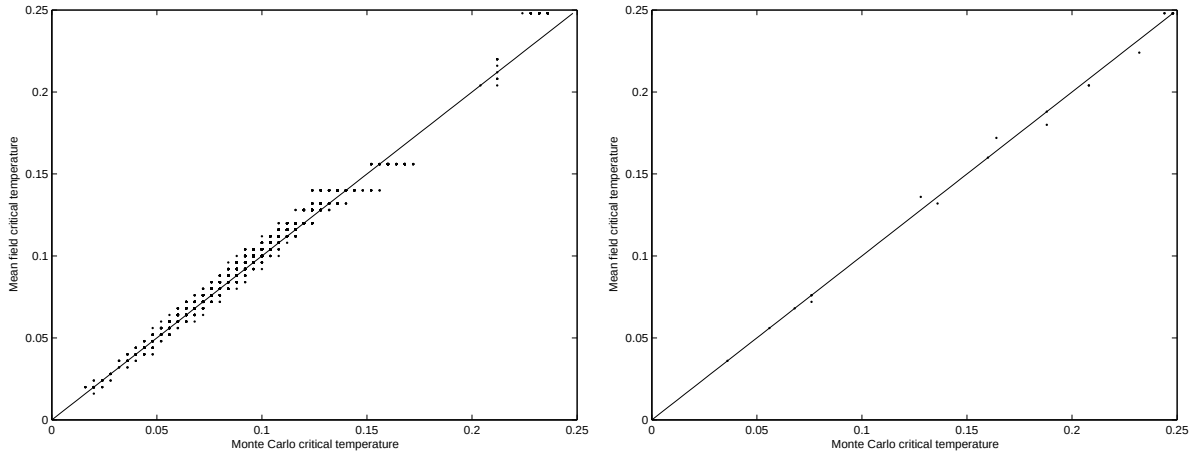

Figure 4.4: Monte Carlo $T_{c_{ij}}$ vs. Mean Field estimated values for the breast cancer data [22]. On the left 1572 genes in 49 dimensional space with average of ~ 7.9 neighbors for each gene. On the right 49 samples in 1572 dimensional gene-space, with average ~ 2.3 neighbors for each sample.

On the other hand, for breast cancer data we obtained very accurate fit between mean field and Monte Carlo (see Fig. 4.3). However, in this case there were a few small clusters connected via single edges to bigger clusters. The mean field approximation overestimate the correlation of these edges by a small amount, rising their critical temperature to that of the small clusters, and one loses the sensitivity to identify the small clusters. We can reconstructed these small clusters using the directed growth cluster building method (which was also essential for increasing the sensitivity of the Monte Carlo-based version of SPC).

SPC with mean field version was used for clustering analysis of stem cells gene expression dataset [18], This experiment contains total of 17 Affymetrix chips (samples) monitoring the gene expression from different development stages: (a) embryonic stem cells (ESC), (b) adult stem cells (ASC) from a variety of tissues: hematopoietic (HSC)

and keratinocytic (KSC), and (c) their terminally differentiated counterparts (HDC and KDC). The new method enable us cluster more then 8000 genes with many different parameters assignments with a reasonable running time.

4.3 SPC - Discussion

As described above, SPC yields a hierarchical clustering solution; the parameter that controls resolution is the local critical temperature of the spins. This critical temperature is a single parameter that represents both of the two factors that determine the local density of points: (a) the typical distance between nearest neighbors, and (b) the number of nearest neighbors of a point. Both of these factors affect the local critical temperature: the closer the neighboring points and or higher the dimensionality, a higher critical temperature is obtained in that region. This behavior is illustrated in the following example. Our dataset consists of three well separated groups of 216 points in each, taken from three uniform distributions : (1) embedded in $2D$ with typical nearest neighbor distance of $\frac{1}{6}$ between points, (2) embedded in $3D$ with typical distence of $\frac{1}{6}$ between nearest neighbor points, and (3) $3D$ with typical distance of $\frac{2}{15}$ between nearest neighbor points. We show in Fig. 4.5 the dendrogram obtained from this data; the critical temperatures of the three clusters are in the expected ascending order.

Returning to the manner in which the local critical temperature reflects the density which, in turn, is governed by the dimensionality and the distance to neighbors, one should note the significant influence the choice of K (the number of mutual neighbors) has on these factors. K determines the length scale a, whose value is the global average of the distances between neighbors. K also determines the maximum distinguishable dimensionality by limiting the number of nearest neighbors, thus restricts SPC distinguishing ability to a limited range of length scale and dimensionality. The result's of applying SPC depend significantly on the parameter K. There are two procedures for choosing the optimal K for a specific problem. The first is heuristic [1] and the second is more rigor-

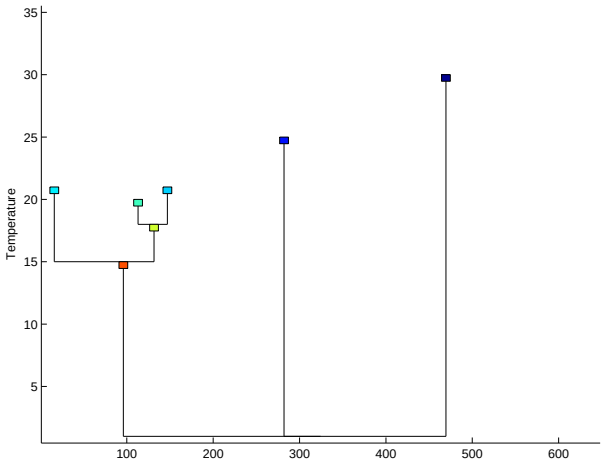

Figure 4.5: Dendrogram obtained from synthetic data consisting of three groups of 216 points in each. On the left, points embedded in $2D$ with typical length of $\frac{1}{6}$ between nearest neighbor points, in the middle, cluster of points embedded in $3D$ with typical length of $\frac{1}{6}$ between nearest neighbors, and on the right, points embedded in $3D$ with typical length of $\frac{2}{15}$ between nearest neighbor points.

ous (based on comparison of resampled partial data) but very time consuming [21]. The heuristic method may choose the "right" value for K, that captures the most information of the specific problem's features, but there are distributions of data with locally varying intrinsic length scales and local dimensionality, for which no single optimal K exists.

As shown in [13], the main influence of the choice of K on the SPC graph and on the clustering results is in regions where the density gradient is high. In these regions there are less mutual nearest neighbors then in a region of uniform density. Since near the surface of a cluster the density gradient likely to be high, the mutual neighborhood criterion helps to separate the cluster from its surrounding points. It was found that in some cases we can neglect the distance dependence of the interaction strength, and use a constant $J \neq 0$ only for all the edges of the SPC graph, without a significant effect on the clustering results. However, the mutual K criterion has no physical sense; it may cause that pairs of spins that are at a large distance (in dilute regions) have a non-vanishing interaction, whereas much less separated pairs, that belong to a dense region, do not interact. This introduces and error in our estimation of the local density by the local

transition temperature; the estimated local density in regions with high gradients are underestimated.

We will use the advantage of the high efficiency of our mean field approximation; instead of obtaining an exact (up to sampling errors) estimate of the correlations, using Monte Carlo, which then is used to estimate the local density in an approximate way, we suggest a method to solve approximately the "right question". Instead of fixing K and scanning all temperatures, we will scan a and for each a obtain a dendrogram as a function of temperature. This will yield a dendrogram controlled by in two parameters: the length scale - a, and temperature T. We extract from the dendrogram the stable clusters, each at it's maximal stability. This novel method of constructing the clusters from the critical temperatures information is presented in the next chapter.

Chapter 5

Density estimation clustering - DEC

We define a cluster as a region in space in which the density of data points is higher then in the surrounding region. This definition implicitly assumes that each cluster c has an optimal length scale a_c which is larger than the typical distances between neighbor points within the cluster, and less than typical distances from the cluster to points around it (that do not belong to c). Say we measure the local density at a data point by centering a sphere S_a of radius a on it and counting the number of points within S_a. This way of estimating the local density yields maximal difference between the local densities of the cluster c and its surrounding region if we choose $a = a_c$, i.e. measure local densities at the optimal length scale of cluster c. Our approach, outlined here and presented in detail below, relies on using statistical mechanics methods to estimate the local density, with the length scale a playing a role that resembles that of the simple method mentioned above.

The outline of our novel clustering algorithm, DEC, can be summarized as follows:

1. select a set of length scales a_μ

2. Use approximate solution for the mean field equations to calculate the local critical temperatures T_{c_i} at all points i. The length scale a_μ affects of estimates $T_{c_i}(a)$ which, in turn, serve as our estimates of the local density function.

3. For each length scale a_μ generate a dendrogram of clusters by varying the temperature T. This yields a dendrogram of clusters at each point in the two-dimensional

parameter space (a, T).

4. Mapping between dendrograms of ascending $a'_\mu s$, combing homolog clusters and generate the final list of clusters in the data.

5. Overcome possible ambiguities between clusters obtained at different length scales and generate a dendrogram with a single resolution axis, which contains the most significant clusters present in the data.

We turn now to a detailed description of the method, illustrating and explaining the procedure by treating a synthetic example, that consists of the following 400 points in $2D$ (shown in Fig. 5.1:

1. 100 points from a Gaussian distribution with mean $[0, 0]$ and variance $[1, 1]$.

2. 100 points from a Gaussian distribution with mean $[3, 3]$ and variance $[1, 1]$.

3. 100 points from a Gaussian distribution with mean $[8, 8]$ and variance $[2, 2]$.

4. 100 points from a uniform distribution with coordinates $[-2, -2] < x, y < [12, 12]$.

5.1 Critical temperatures at different resolutions

First we need to find the relevant length scales on which we will probe a specific problem. We use as a lower bound the minimal distance between points, and as an upper bound the average distance between all the points in the data. Next we choose *a step* to generate a sequence of length scales a_μ, equally spaced between the lower and the upper bounds. If the user is interested in a specific range of length scales he can insert the desired set of a_μ as an input.

At each length scale a_μ we estimate the local density function by calculating the local critical temperature T_{c_i} at each point i, using the simplest and most efficient version of calculating the critical temperatures, the approximate solution of the mean field equations,

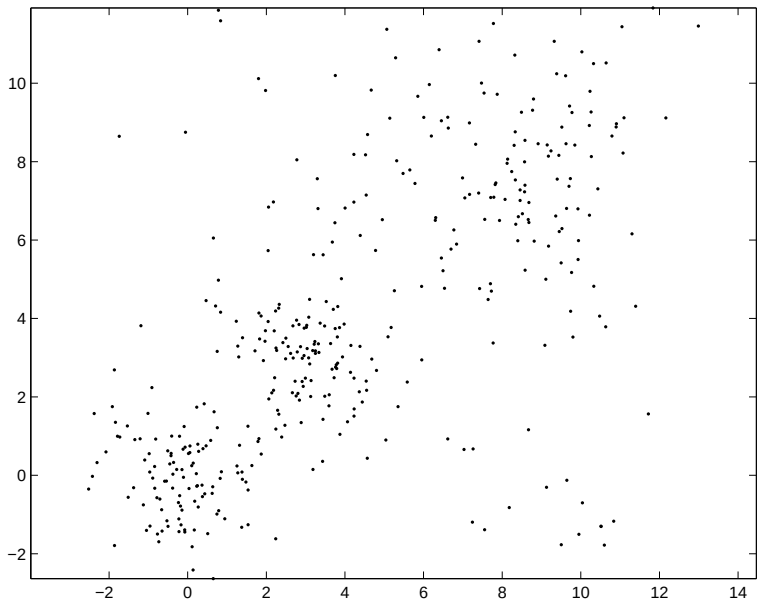

Figure 5.1: The synthetic data points example, 400 points in $2D$.

Sec. 3.3.3. We first set the interaction between spins J_{ij} to be the following step function of the distance d_{ij} between points i, j:

$$J_{ij} = \begin{cases} 1 & d_{ij} \leq a_\mu \\ \\ 0 & d_{ij} > a_\mu \end{cases} \tag{5.1}$$

neglecting the pre-factor, equations (3.31) become

$$T_{c_i} = \sum_j J_{ij}\Delta_{ij} \qquad\qquad \Delta_{ij} = \begin{cases} 1 & T_{c_i} \leq T_{c_j} \\ \\ 0 & T_{c_i} > T_{c_j} \end{cases} \tag{5.2}$$

This generates a very simple interpretation of the local critical temperature T_{c_i}, as the number of neighbors of point i, whose local critical temperature is equal to or higher than T_{c_i}.

The advantage of using statistical physics for estimating the local density function is it's ability to overcome fluctuations in the data and generate a much smoother density function. These properties make DEC more robust and are crucial for its next steps.

5.2 Building dendrograms along two axes

First we describe how at a fixed length scale a we construct a dendrogram for varying T.

Within our approximation all T_{c_i} take integer values; therefore at each length scale a_μ we vary the temperature from $\max\{T_{c_\mu}\}$ to 1 with steps of 1. At each temperature T we identify cluster cores as follows: set as "valid" the points i with $T_{c_i} > T$, and then set as valid all the edges for which $J_{ij} = 1$ and both i, j are valid nodes. Each disconnected subgraph of points connected by valid edges constitutes the core of a cluster at the temperature T.

The effect of lowering T: Say we have our clusters C_α at a temperature T and now lower the temperature to $T' < T$, and perform the procedure described above to create a new clustering. The simplest change is the emergence of C', composed of points that did not belong to any cluster at T and now, at T' they do form the core of a new cluster; with the membership of the old clusters, C_α, unchanged. In this case C' is added as a new cluster. Another possible change of the clustering assignment of the data is the following: N_1 points, that did not belong to a cluster at T, become connected to each other at T', *and* become also connected to an old cluster, say C_1. In order to increase our algorithm's sensitivity to small clusters, we optionally check whether N_1 exceeds by more than twice the number of those points of the old cluster, to which the new cluster is connected. If it does, we first identify these N_1 points as a new independent cluster C_2, and then merge it with C_1 to form their "parent" cluster C. Otherwise - we simply expand C_1 to include the new group.

The third possible change is that the N_1 points that became valid at T' connect two or more old clusters, C_α. Again we compare N_1 to the number of members of old clusters to which the new points became connected, and either generate a new cluster of size N_1 which is then joined to the C_α to form a single cluster C, or we form C directly.

New directed growth. Treating the local density function as a topographic map, we think of a cluster's core as the peak of a mountain, above any saddle leading to neighbor

mountains. We suggest a new version of the directed growth procedure, see chapter 4.1, which expands a cluster's core to contain also the hillside points. Say at some temperature T we have created the core of a particular clusters, C_1. Now we lower the temperature by one step, to $T' < T$, setting additional points as valid. If a group of such additional valid points is connected only to C_1 (and not to another existing cluster), we add them to C_1 at temperature T. Now we lower the temperature by another step, expanding the current C_1 the same way. The process of expanding C_1 terminates when no new points join it. Note that for the growth process we retain the identity of C_1 even below temperatures at which it merged with another cluster.

The new directed growth version has three advantages compared to the previously introduced directed growth method:

1. The old directed growth is based on correlations. When two clusters C_1 and C_2 are connected via a relatively high saddle, their separation temperature will be high, and it is likely that at this temperature the correlation between the core of C_1 and a set of points H_1, which are at the hillside of C_1 (and "belong" to it), are already lower than the old directed growth threshold. In such a case the old directed growth method will not expand C_1 to its maximal natural size; H_1 will be added only to the cluster obtained by merging C_1 and C_2. In the new method C_2 does not influence the expansion of C_1.

2. When the correlation is low, its dominant part is due to the direct interaction between neighbor spins, *i.e.* is a function of the distance between them. Thus, like the Single Linkage method, it may suffer from high sensitivity to fluctuations in the data. The new method uses only information due to collective behavior of the points.

3. The new method expands only an existing cluster's core and does not generate new, less significant clusters, like the old directed growth.

Thus, the new expansion method yields larger clusters in a more reliable fashion.

Creating dendrograms: The procedures described above assign data points to clusters, at each length scale a_μ and temperature T. In order to construct a dendrogram at each fixed a_μ, we move from low to high T and connect each parent cluster to it's "children", that split from it as T is raised. The results for our synthetic example are shown in fig. 5.2. As expected, for the very small value of $a_1 = 0.3$ we get a trivial "dendrogram", in which only the core of the first dense Gaussian shows up, for a short range of very low temperatures, as a cluster. For a wide range of subsequent a_μ the "true" structure becomes apparent - each Gaussian cloud is identified and is long-lived (exists over a significant range of T values), until the true structure starts to get washed out, for $a \geq 1.9$.

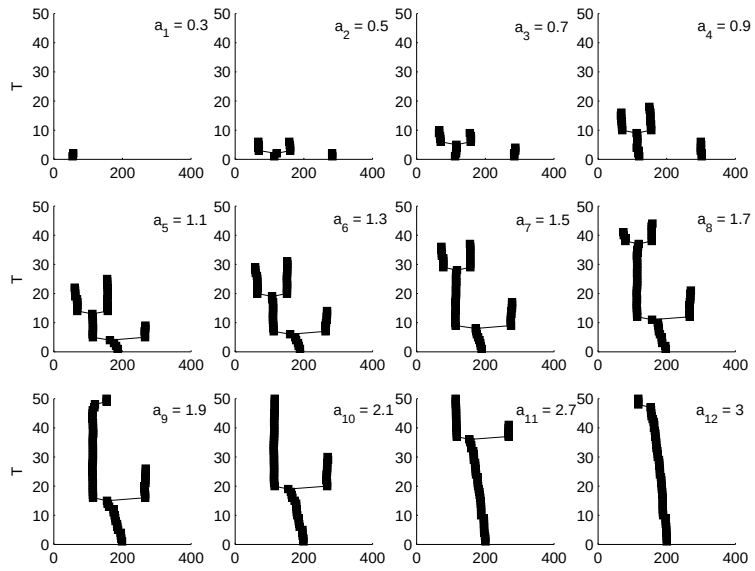

Figure 5.2: Dendrograms obtained from 3 gaussians in $2D$ data points for several a_i. The vertical coordinate is T and the horizontal coordinate is the indices of the data points.

5.3 Extracting the relevant clusters from the dendrograms

Since the natural clusters in the data are represented many times in our dendrograms, we merge duplicates of the same cluster in two steps: first at fixed a_μ, identify homolog

clusters that appear at various T values, and then do the same for different values of a_μ. The first step is done in the following way: say C' is a cluster at $T' < T$ and C_1 is a cluster at T. We call them *homolog clusters* if C' contains the members of C_1 but does not contain points that belong (at T) to another cluster C_2, which is a "brother" of C_1 (i.e. its merge with C_1 created C in the process described above).

We scan this way at each a_i the clusters that were generated by lowering T and find chains of homolog clusters and gather every chain to one united cluster. Each such chain extends over a range of temperatures; in the resulting *reduced dendrogram* we display each of these united clusters at the highest temperature at which the chain of homolog clusters exists. The reduced dendrograms obtained for our example at each a_μ are shown in Fig. 5.3. Finally, the stability of each united cluster is represented by ΔT, the range of temperatures over which it exists.

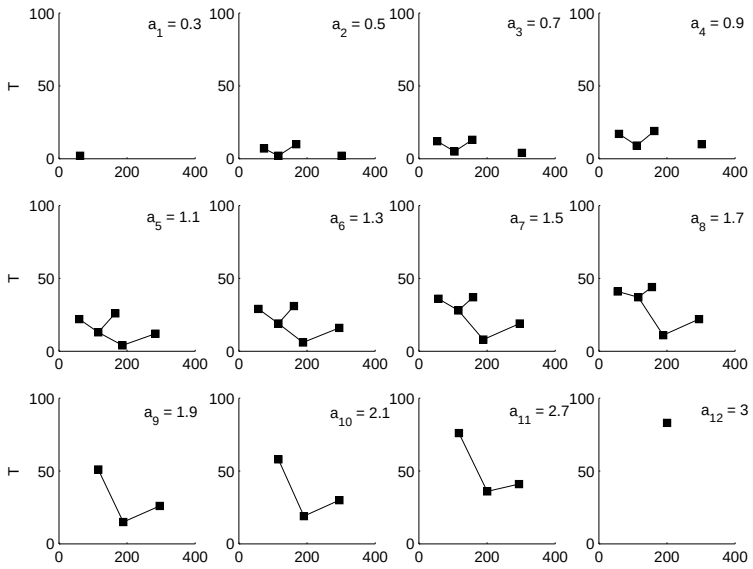

Figure 5.3: The reduced dendrograms obtained from 3 Gaussians in $2D$ data points for several a_i.

Next we turn to merge duplicates of the same united cluster that were obtained at different a_μ. We scan the a axis and construct a mapping between clusters obtained at two consecutive ascending a_μ in the following way. First, calculate the Jaccard similarity

measure between each pair of clusters from different a's, defined as the ratio of the size of their intersection to the size of their union. For each cluster at a_μ we identify the one most similar to it out of the clusters at $a_{\mu'}$, and do the same for each cluster at $a_{\mu'}$. If two clusters are mutually at the top of the other's list, we tentatively call them homologous. For each tentatively homologous pair we perform a test to check that it generates less than the *maximum allowed ambiguity* which is defined in the next section, 5.4. If the pair passes the test, it's two clusters are identified as homologous.

This procedure yields chains of homolog clusters; a chain typically starts from a core of a cluster obtained at some small a. As we increase a, the stability of the cluster, ΔT, increases, reaching a maximal value at the cluster's optimal a, and then ΔT starts to decrease. The chain ends when the cluster stops existing as an independent entity. From each chain of homolog clusters we choose as representative the cluster with maximal ΔT. This procedure yields, finally, a single representative for each relevant family of homolog clusters in the data. We can also define an additional stability measure for each such cluster: the range of a values through which the corresponding chain of homolog clusters exists, Δa.

5.4 Creating the final dendrogram, in a and T

A legal single-rooted dendrogram must have a tree structure; the constituent clusters must obey the following conditions:

1. The root is a cluster that contains all points

2. When a cluster C splits into two (or more) parts (i.e. clusters or singletons), a point of C must belong to one (and only one) of these parts.

3. C must contain all the points that constitute it's descendant clusters.

The clusters created by the procedure outlined in the previous Section may violate these conditions, giving rise to ambiguities that prevent us from composing from them a legal

dendrogram. Some of these ambiguities are minor, related to only a few points at the borders of the clusters. Other ambiguities can be more acute, related to a discontinuity in the data structure in the length scale space, as shown in Fig. 5.4. In this example we generate two elongated clouds of points in 2D, each cloud contains 400 points from exponential distributions centered at the top and bottom edges of the clouds. When the clustering procedure outlined above is applied to this data, we get the following assignments. For length scales $a < 1$ at high T we identify first the four edges of the clouds as independent clusters. At some lower temperature the two clusters that grew out of the top and bottom edges of a cloud are merged and we have two clusters, each containing the pints of one cloud. If the temperature is lowered at length scales $a > 1$, there are also interactions with $J_{ij} = 1$ between points that belong to different clouds. Then the four edges are merged differently: first the two upper edges merge and the lower edges merge, and at some lower temperature we obtain one large cluster which consists of all the data points. It is clear that we can not build a legal dendrogram by combining the clusters of the two dendrograms obtained in these two regimes of a.

We leave for future work the optimal solution of such problems, which probably will be based on some a fuzzy clustering procedure. Here we present one particular protocol that ensures that we do obtain a legal dendrogram that contains the most significant clusters. The idea is to solve the minor ambiguities by making the minimal number of changes in a cluster, and identify the acute ambiguities that cannot be eliminated by moving or adding a few points. These more serious conflicts are resolved by choosing the more stable of two competing clustering assignments of the same points and ignoring the other. In the context of our example the upper part of the left cloud can form C_1 by merging with the lower part of the left cloud, or C_1' by merging with the upper part of the right cloud. These two clustering assignments conflict and we select the first if C_1 is more stable than C_1'. To implement this we sort the clusters according to their ΔT and then create the final list of clusters incrementally; we add to the current list of clusters (which do not violate

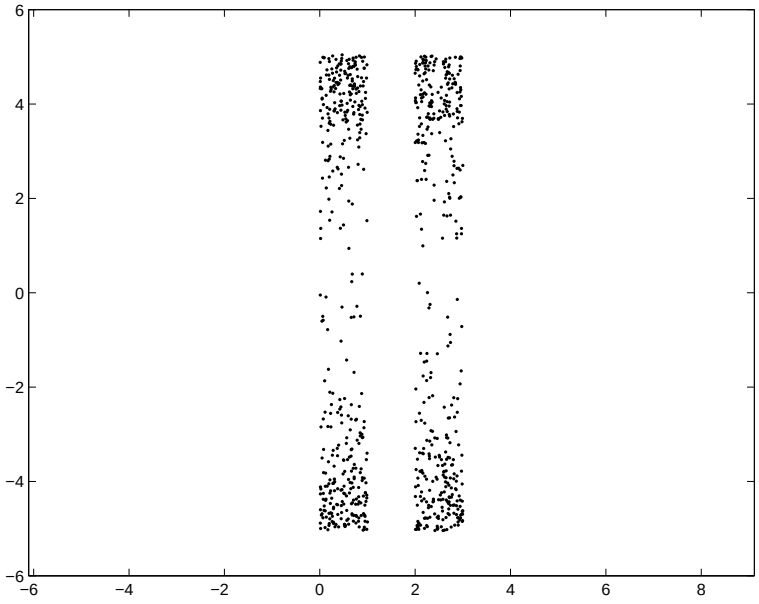

Figure 5.4: two elongated clouds of points in 2D, both with exponential distributions of points centered at the top and bottom edges of the clouds.

the dendrogram rules), the most stable new cluster if by adding or removing less than N_m points we can ensure that it's addition generates a legal dendrogram. The parameter N_m is called the *maximum allowed ambiguity*. If we have to add or remove more than N_m points, this cluster is ignored, and so on. The *maximum allowed ambiguity* is usually set to be close to the minimal size of clusters that are of interest to the user.

When this procedure is applied to the clusters generated for the data of our example of three Gaussians, we obtain the final dendrogram shown in Fig. 5.5. Each cluster is placed at coordinates that represent it's optimal length scale a and critical temperature at that a.

5.5 Future improvements

The proposed way of scanning resolution space is not the most efficient one. Developing a better way for probing the data only at it's relevant length scales will decrease the algorithm's running time and improve it's performance.

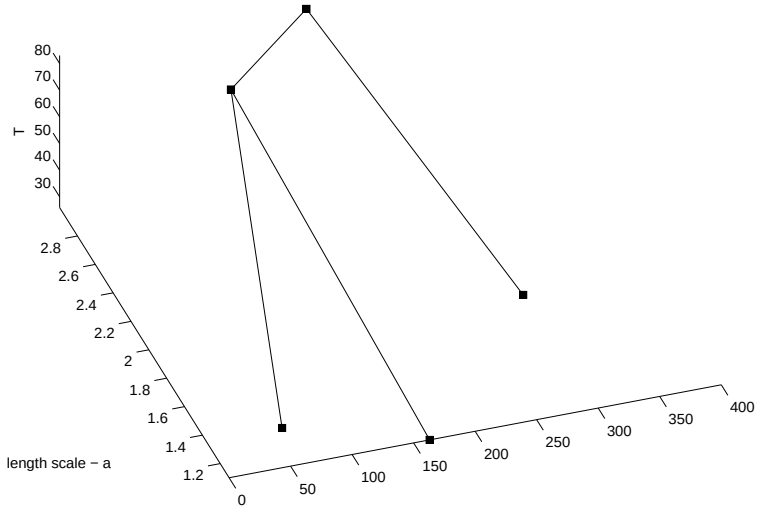

Figure 5.5: The final dendrogram obtained from 3 gaussians in $2D$ data points.

However, the scheme outlined above does have several advantages. One is the observation regarding a cluster's ΔT, which is maximal at the cluster's optimal a. This part of the procedure is general, allowing deduction of the optimal a of a cluster from calculating the stability along the chain of homolog clusters. By searching a using the "lion in a desert" method, a cluster's optimal a can be found efficiently with any desired accuracy.

Development of a fuzzy clustering solution, where a point can belong to many clusters with some probability may help to overcome the violations of the dendrogram rules, in a less ad hoc way than that of our present procedure of enforcing generation of a legal final dendrogram.

These suggestions are left for future development.

Chapter 6

Applications - Clustering Gene Expression data

In this chapter we test our algorithm's performance on real data sets. Although the algorithm is general and can be applied to any kind of data, in this work we are focusing on gene expression.

A typical gene expression experiment monitors the concentrations of many thousands of mRNA transcripts simultaneously, over n_s (of a few tens) of different samples. After filtering the probe sets down to n_g of a few thousands, keeping those genes whose expression levels varied significantly over the different samples, we obtain an $n_g \times n_s$ raw data matrix $\Re$. Then a logarithmic transformation is applied to the raw data in order to bring the gene expression values to the same relative scale and to turn the noise into (approximately) additive.

When clustering gene expression data one has to keep in mind some of it's unique properties; (a) The points, both representing genes in sample space or the samples in gene space, are embedded in a high dimensional space. (b) Usually the shapes of the clusters are not known; a cluster may be compact as well as elongating. If we apply a typical preprocessing, of centering and normalizing each gene, such that its mean vanishes and its norm is one, *i.e.* the distance between genes is simply related to their correlations, clusters tend to be compact, but exceptions are fairly common. (c) The sizes of the relevant clusters have large variations. When clustering genes, we are interested in clusters

whose size varies from several hundreds to a few. The ability to identify small clusters is an important test of the sensitivity of the clustering algorithm.

Assessing clustering results is a hard question. The clustering problem is "ill defined" and the answer is largely subjective. Moreover, when clustering gene expression data we do not have the true solution. There are a few scores which try to measure the cluster homogeneity and separation , see for example [24]; these scores must assume something about the cluster's desired structure and may miss relevant clusters. For biological data the structure of the distribution function from which the data points were sampled is unknown, which prevents us from giving a biology-based p-value to a cluster. Only by combining the clustering results with known biological information, such as genes' annotation and the clinical labels of the samples can reveal which are the meaningful clusters in the data. Following this point of view, we adopt a practical test of the clustering results; we are looking for a clustering algorithm that minimizes both (a) the false positive rate: (number of found clusters - number of found relevant clusters) / number of found clusters, and (b) the false negative rate: (number of relevant clusters - number of found relevant clusters) / number of relevant clusters.

We will compare DEC's results, for a few representative examples, with other clustering algorithms that do not make any assumption on the clusters' structure: SPC and Single Linkage.

Leukemia data: We ran DEC to cluster $n_g = 906$ normalized genes in the space of $n_s = 72$ samples [2], (see Fig. 6.1).

The comparison to the other methods is summarized in table 6.1

Most of the clusters which DEC identifies are relevant and we would recommend that the user checks their possible biological meaning. It is clear that SPC without directed growth and Single Linkage are missing many small but interesting clusters, *i.e.* have higher false negative rate, while SPC with directed growth has a very poor false positive rate, which makes it hard to extract the relevant clusters. Moreover, most of the clusters

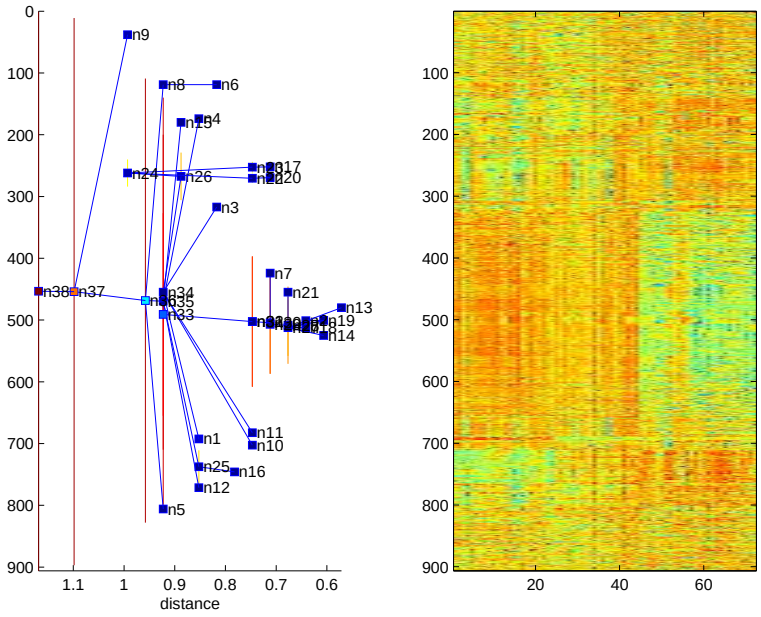

Figure 6.1: Projection of the final dendrogram on the distance axis and the reordered Leukemia data [2], obtained from clustering 906 normalize genes in the space of 72 samples, using the DEC algorithm. The dendrogram contains clusters with minimal size of four genes. The vertical lines indicate the group of datapoints that belong to the cluster through which the line passes.

Method	♯ Clusters with size > 3	Size of largest relevant clusters
DEC	38	329, 78, 54
SPC with directed growth	92	293, 100, 82
SPC without directed growth	13	152, 51, 48
Single Linkage	17	241, 49, 25

Table 6.1: A comparison between DEC, SPC with and without directed growth and Single Linkage on the Leukemia data [2].

were created by the directed growth part of SPC and thus they are less stable and reliable.

Breast cancer: Similar behavior was observed from another data set. We clustered $n_g = 1572$ normalized genes in the space of $n_s = 49$ breast cancer samples, taken from primary tumor tissues [22] (see Fig. 6.2). This data contains many clusters, all of which have with relatively small size. The comparison is summarized in table 6.2

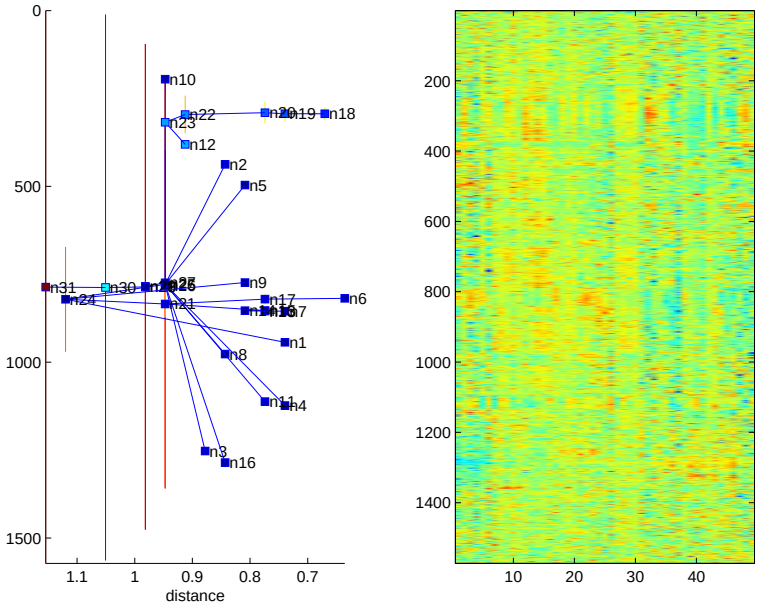

Figure 6.2: Projection of the final dendrogram on the distance axis and the reordered breast cancer data [22], obtained from clustering 1572 normalized genes in the space of 49 samples, using the DEC algorithm. Clusters with minimal size of eight genes are shown in the dendrogram. The vertical lines indicate the group of datapoints that belong to the cluster through which the line passes.

Method	♯ Clusters with size > 7	Size of largest relevant clusters
DEC	31	62, 32, 30
SPC with directed growth	51	56, 74, 54
SPC without directed growth	11	21, 23, 14
Single Linkage	19	70, 39, 19

Table 6.2: A comparison between DEC, SPC with and without directed growth and Single Linkage on the breast data [22].

The advantage of DEC over Single Linkage can be understood in the following way. In DEC we add a new axis, of temperature, to the distance axis of the hierarchial dendrogram solution of the Single Linkage algorithm. The new axis measures the local density of the points at a given length scale, a, since the critical temperature of a point i is determined by the number of neighbors it has at a distance less than a.

In high dimensions there is considerable probability that points be placed by chance on a quasi one dimensional string with relatively short distances between the points. While Single Linkage will not identify the separation of two well separated highly dimensional clouds connected via a quasi one dimensional string, along the new axis the two clouds will be recognized as independent at high temperatures and will be connected only at low temperatures; thus DEC is more robust against noise in the datasets.

DEC is implemented in Matlab, its running time is $O(n^2)$, taking a few minutes to cluster thousands of points.

Chapter 7

Summary

This work has two main parts. In the first, which deals with statistical physics, we presented a mean field solution to the inhomogeneous Potts model, we studied the model's Paramagnetic-Ferromagnetic phase transition and showed that within mean field, as a result of the model's inhomogeneity, each region in space has its unique critical temperature. Then we described the relation between this local critical temperature and the local density of spins in space. We used this mean field solution in order to improve dramatically the running time efficiency of the previously introduced clustering algorithm - SPC.

The second part deals with the clustering problem. We introduced the problem and suggested a general definition of a cluster as a region in space with higher density than in it's surrounding regions. We presented a novel clustering algorithm - DEC, which generates a hierarchial clustering solution from the local density function of the data points in space. The density of points was measured at fixed values of a length scale parameter a. The value of a was varied; at each length scale we calculated, using an approximate mean field solution, the local density function of the data points. At each length scale we produced a hierarchial clustering solution as a function of temperature. The critical temperature of a point i is determined by the number of neighbors it has at a distance less than a, *i.e.* $T_c(i)$ is indicative of the local dimensionality near point i at length scale a. This way we get a clustering assignment of points with two axes: length

scale and temperature. We describe a procedure to extract the relevant clusters and give a final hierarchial assignment, where each cluster has two coordinate: its optimal length scale and its corresponding critical temperature at this length scale.

We demonstrated how to add a new axis within DEC to the hierarchial dendrogram solution of Single Linkage algorithm, which measures the dimensionality of the points. Adding the new axis enables us to overcome Single Linkage's limitations and to suggest a general and robust clustering algorithm.

Although there are still open issues for further development in order to exhaust the algorithm's potential, even the current version demonstrates its power and significant advantages. Gene expression analysis provides a great challenge to a clustering algorithm; we demonstrated how the use of DEC can improve clustering analysis and give a better data mining from gene expression data sets.

Appendix A

Coupled Two Way Clustering (CTWC)

Coupled Two-Way Clustering (CTWC) [14] is a method designed to mine data from gene expression microarray experiments. The basic idea is to perform two kinds of cluster analysis; (a) viewing the samples as the data points for clustering (b) clustering the genes. The standard approach is to use for aim (a) all genes (that passed some threshold) to represent a sample, and cluster all samples, and for aim (b) to use all samples to represent a gene, and then cluster all genes.

The novelty of the CTWC method is that the clustering operations are preformed in an iterative and coupled fashion. The idea is to use a subset of the genes to cluster a subset of the samples, and vice versa. The underlying reason is that only a small subset (a few tens) of the genes is expected to belong to a particular process of interest, while the vast majority (thousands) of the genes are bystanders and play no role in the process. Hence when clustering samples on the basis of similarity of their expression profiles, the "message" of the small but relevant gene subset might be masked by transcriptional 'noise' from the many irrelevant genes. Similarly, the genes that belong to the relevant subset may have highly correlated expression over those samples in which the process of interest actually takes place; including samples in which it does not occur may mask these correlations.

Furthermore, one may have a situation in which the expression levels of the relevant

subset of genes clearly separates a group A of the samples into two subgroups, A1 and A2, but this partition is hidden by the presence of an unrelated group of samples B. Hence in order to see the partition one should use only a submatrix of the expression matrix; one that contains only the rows corresponding to the relevant genes and the columns that contain the samples of group A; such a submatrix is called *bicluster*. Checking every possible partition in the data guarantees that one finds every possible bicluster in the data. This approach is, of course, impossible to implement, since the number of possible biclusters in the data grows exponentially with the size of the problem. Several algorithms were developed during the recent years to attack this problem ([9], [6], [19], [25]).

CTWC provides an efficient heuristic method to identify such biclusters; it breaks down the total data set into subsets of genes and samples that can reveal significant partitions into clusters. This is done in an iterative way. The iterative step is the following: First we cluster all samples using all genes, and vice versa (these operations are referred to as DEPTH 1), and identify stable clusters (of genes and of samples). Then (at DEPTH 2 and 3) each stable gene cluster, GI, is used to cluster all stable sample clusters, SJ; this clustering operation is denoted by SJ(GI). The operations GI(SJ), using each SJ to cluster the members of every GI, are also performed. Note that G1 is the group of all genes and S1 - of all samples. Whenever a clustering operation generates a new stable subcluster, it is recorded and its members are used in the next iterative step (DEPTH 4 and higher). The process stops when no new stable clusters, that exceed a minimal size, are generated.

The output of the procedure is a set of statements of the following kind: "when the genes of GI are used to cluster the samples of SJ, one obtains the following dendrogram, which contains stable sample clusters SK".

CTWC is implemented by a publicly available web-server [15] and has been used extensively for gene expression analysis in the last few years and for applications to colon, breast [16], skin cancer [10], leukemia [23, 11] and glioblastoma [17] data analysis.

The core of the CTWC method is a clustering algorithm. The most important requirement from the clustering algorithm used is its ability to identify stable clusters. Identifying all the stable clusters is crucial for mining all the information from the data. It is also essential *not to identify* clusters due to noise and fluctuations in the data as stable (avoid false positives). This ability not only reduces dramatically the running time of the algorithm preventing, CTWC wasting time on irrelevant clusters, but also, and maybe more importantly, reduces the output size of the method, enabling the user to focus his attention on the relevant and meaningful results.

Another requirement is to obtain the clusters at their maximal size,; it is important that the samples' clusters be as large as possible to make the next iterative step meaningful.

The current version of CTWC uses the mean field SPC version as an option, which reduces significantly the CTWC running time.

Its seems that DEC meets better the CTWC requirements, and using it as the underlying clustering algorithm will make CTWC much more efficient for users, to

1. Generate the significant clusters at their maximal size,

2. with only minor dependence on specific parameters.

Bibliography

[1] Agrawal, H., and Domany, E. 2003. "Potts ferromagnets on co-expressed gene networks: Identifying maximally stable partitions", *Physical Review Letters* **90**, 158102.

[2] Armstrong SA, Staunton JE, Silverman LB, Pieters R, den Boer ML, Minden MD, Sallan SE, Lander ES, Golub TR, Korsmeyer SJ. 2002. "MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia", *Nature Genetics* **30**,(1)41-47.

[3] Blatt, M., Wiseman, S., and Domany, E. 1996a. "Super–paramagnetic clustering of data", *Physical Review Letters* **76**, 3251-3255.

[4] Blatt, M., Wiseman, S., and Domany, E. 1996b. "Clustering data through an analogy to the Potts model", *Advances in Neural Information Processing Systems* **8**, 416-422 (1996), Touretzky, Mozer, Hasselmo, eds., MIT Press.

[5] Blatt, M., Wiseman, S., and Domany, E. 1997. "Data Clustering Using a Model Granular Magnet", *Neural Computation* **9**, 1805-1842.

[6] Califano, A., Stolovitzky, G., and Tu, Y. 2000, "Analysis of gene expression microarrays for phenotype classification", *Proc. ISMB'00* **8**, 75-85.

[7] Chaikin, P. M., and Lubensky, T. C. 2000. *Principles of condensed matter physics*, 198-201, Cambridge : Cambridge University Press.

[8] Chaikin, P. M., and Lubensky, T. C. 2000. *Principles of condensed matter physics*, 213-214, Cambridge : Cambridge University Press.

[9] Cheng, Y., and Church, G. 2000. "Biclustering of expression data", *Proc. ISMB'00* **8**, 93-103.

[10] Dazard, J.-E., Gal, H., Amariglio, N., Rechavi, G., Domany, E., and Givol, D. 2003. "Genome-wide comparison of human keratinocyte and squamous cell carcinoma responses to UVB irradiation: implications for skin and epithelial cancer", *Oncogene* **22**, 19:2993-3006.

[11] Einav, U. 2003. "Class discovery in Acute Lymphoblastic Leukemia using gene expression analysis". M. Sc. Thesis, Weizmann Institute of Science.

[12] Eisen, M. B., and Brown, P. O. 1999. "DNA arrays for analysis of gene expression", *Methods in Enzymology* **303**, 179-205.

[13] Getz, G. 1998. "Classification and Clustering of Protein Structures", M. Sc. Thesis, Tel Aviv University.

[14] Getz, G., Levine, E., and Domany, E. 2000. "Coupled Two Way Clustering analysis of gene microarray data", *PNAS* **97**, 12079-12084.

[15] Getz, G. and Domany, E. 2003a. "Coupled Two-Way Clustering Server", *Bioinformatics* **19**, 9:1153-1154.

[16] Getz, G., Gal, H., Kela, I., Notterman, D. A., and Domany, E. 2003b. "Coupled two-way clustering analysis of breast cancer and colon cancer gene expression data", *Bioinformatics* **19**, 9:1079-1089.

[17] Godard, S., Getz, G., Delorenzi, M., Farmer, P., Kobayashi, H., Desbaillets, I., Nozaki, M., Diserens, A.-C., Hamou, M.-F., Dietrich, P.-Y., Regli, L., Janzer, R. C., Bucher, P., Stupp, R., de Tribolet, N., Domany, E. and Hegi, M. E. 2003. "Classification of Human Astrocytic Gliomas on the Basis of Gene Expression: A Correlated Group of Genes with Angiogenic Activity Emerges As a Strong Predictor of Subtypes" *Cancer Research*,**63** 20:6613 - 6625.

[18] Golan-Mashiach, M. 2004. "The meaning of stemness. Analysis by transcriptomes", M. Sc. Thesis, Weizmann Institute of Science.

[19] J. Ihmels, G. Friedlander, S. Bergmann, O. Sarig and N. Barkai. 2002. "Revealing Modular Organization in the Yeast Transcriptional Network", *Nature Genetics* **31**, 370-377.

[20] Jain, A.K., and Dubes, R.C. 1988. *Algorithms for clustering Data*, Prentice-Hall, Englewood Cliffs.

[21] Levine, E., and Domany, E. 2001. "Resampling method for unsupervised estimation of cluster validity", *Neural Computation* **13**, 2573-2593.

[22] Perou, C. M., Sorlie, T., Eisen, M. B., van de Rijn, M., Jeffrey, S. S., Pollack, J. R., Rees, C. A., Ross, D. T., Johnsen, H., Akslen, L. A., Pergamenschikov, C. W., Zhu, S. X., Lonning, P.E., Borresen-Dale, A.-L., Brown, P. O. and Botstein, D. 2000. "Molecular portraits of human breast tumours". *Nature* **406**, 747-752.

[23] Rozovskaia, T., Ravid-Amir, O., Tillib, S., Getz, G., Feinstein, E., Agrawal, H., Nagler, A., Rappaport, E. F., Issaeva, I., Matsuo, Y., Kees, U. R., Lapidot, T., Lo Coco, F., Foa, R., Mazo, A., Nakamura, T., Croce, C. M., Cimino, G., Domany, E. and Canaani, E. 2003. "Expression profiles of acute lymphoblastic and myeloblastic leukemias with ALL-1 rearrangements" *PNAS*,**100**, 13:7853-7858.

[24] Sharan, R. and Shamir, R. 2000. "CLICK: A Clustering Algorithm with Applications to Gene Expression Analysis" *Proc. 8th International Conference on Intelligent Systems for Molecular Biology (ISMB '00)*, 307-316, AAAI Press, Menlo Park, CA.

[25] Tanay, A., Sharan, R., and Shamir, R. 2002. "Discovering statistically significant biclusters in gene expression data", *Bioinformatics* **18** S136-S144.

[26] Wu, F.Y., 1982. "The Potts model",*Reviews of modern physics* **54**,(1)235-286.

[27] Wang, S., and Swendsen, R.H. 1990. "Cluster Monte Carlo algorithms", Physica A **167**,565-579.