Segmentation of cDNA Microarray Images using Parallel Spectral Clustering

Sandrine Mouysset[a], Ronan Guivarch[b], Joseph Noailles[b], Daniel Ruiz[b]

[a] University of Toulouse - UPS - IRIT,
[b] University of Toulouse - INPT(ENSEEIHT) - IRIT.

KEYWORD

Spectral Clustering

Domain Decomposition

Image Segmentation

Microarray Image

ABSTRACT

Microarray technology generates large amounts of expression level of genes to be analyzed simultaneously. This analysis implies microarray image segmentation to extract the quantitative information from spots. Spectral clustering is one of the most relevant unsupervised methods able to gather data without a priori information on shapes or locality. We propose and test on microarray images a parallel strategy for the Spectral Clustering method based on domain decomposition with a criterion to determine the number of clusters.

1 Introduction

Image segmentation in microarray analysis is a crucial step to extract quantitative information from the spots [RUEDA, 2009], [USLAN, 2010], [CHEN, 2011]. Clustering methods are used to separate the pixels that belong to the spot from the pixels of the background and noise. Among these, some methods imply some restrictive assumptions on the shapes of the spots [YANG, 2001], [RUEDA, 2005]. Due to the fact that the most of spots in a microarray image have irregular-shapes, the clustering based-method should be adaptive to arbitrary shape of spots such as fuzzy clustering [GLEZ-PENA, 2009], but it should also not depend on many input parameters. To address these requirements, the spectral methods, and in particular the spectral clustering algorithm introduced by Ng-Jordan-Weiss [NG, 2002], are useful to partition subsets of data with no a priori on the shapes. Spectral clustering exploits eigenvectors of a Gaussian affinity matrix in order to define a low dimensional space in which data points can be easily clustered. But when very large data sets are considered, the extraction of the dominant eigenvectors becomes the most computational task in the algorithm. To address this bottleneck, several approaches about parallel Spectral Clustering [SONG, 2008], [FOWLKES, 2004], were recently suggested, mainly focused on linear algebra techniques to reduce computational costs. In this paper, by exploiting the geometrical structure of microarray images, a parallel strategy based on domain decomposition is investigated. Moreover, we propose solutions to overcome the two main problems from the divide and conquer strategy: the difficulty to choose a Gaussian affinity parameter and the number of clusters *k* which remains unknown and may drastically vary from one subdomain to the other.

2 Spectral Clustering for cDNA microarray images

Let first introduce some notations and recall the Ng-Jordan-Weiss algorithm [NG, 2002] and then adapt the spectral clustering for image segmentation.

2.1 Spectral clustering

Let consider a microarray image *I* of size *l* x *m*. Assume that the number of targeted clusters *k* is known. The algorithm contains few steps which are described in Algorithm 1:

Algorithm 1. Spectral clustering algorithm

Input : Microarray image *I*, number of clusters *k*.

1. Form the affinity matrix with $n = l \times m$ defined by equation (1).

2. Construct the normalized matrix:

$$L = D^{-1/2}AD^{-1/2} \text{ with } D_{ii} = \sum_{r=1}^{n} A_{ir},$$

3. Assemble the matrix $X = [X_1 X_2 \ldots X_k] \in R^{n \times k}$ by stacking the eigenvectors associated with the *k* largest eigenvalues of *L*,

4. Form the matrix *Y* by normalizing each row in the *n* x *k* matrix *X*,

5. Treat each row of *Y* as a point in R^k , and group them in *k* clusters via the *K-means* method,

6. Assign the original point I_{ij} to cluster *t* when row *i* of matrix *Y* belongs to cluster *t*.

First, the method consists in constructing the affinity matrix based on the Gaussian affinity measure between I_{ij} and I_{rs} the intensities of the pixels of coordinates (i, j) and (r, s) for $i, r \in \{1,\ldots,l\}$ and $j, s \in \{1,\ldots,m\}$. After a normalization step, the k largest eigenvectors are extracted. So every data point I_{ij} is plotted in a spectral embedding space of R^k and the clustering is made in this space by applying *K-means* method. Finally, thanks to an equivalence relation, the final partition of the data set is directly defined from the clustering in the embedded space.

2.2 Affinity measure

For image segmentation, the microarray image data can be considered as isotropic enough in the sense that there does not exist privileged directions with very different magnitudes in the distances between points along theses directions. The step between pixels and brightness are about the same magnitude. So, we can include both 2D geometrical information and 1D brightness information in the spectral clustering method. We identify the microarray image as a 3-dimensional rectangular set in which both geometrical coordinates and brightness information are normalized. It is equivalent to setting a new distance, noted d , between pixels by equation (2). So by considering the size of the microarray image, the Gaussian affinity A_{ir} is defined as follows:

$$A_{ir} = \begin{cases} \exp\left(-\frac{d(I_{ij}, I_{rs})^2}{(\sigma/2)^2}\right) if\,(i,j) \neq (r,s) \\ 0 \quad otherwise, \end{cases} \quad (1)$$

where σ is the affinity parameter and the distance d between the pixel (i, j) and (r, s) is defined by:

$$d(I_{ij}, I_{rs}) = \sqrt{\left(\frac{i-r}{l}\right)^2 + \left(\frac{j-s}{m}\right)^2 + \left(\frac{I_{ij} - I_{rs}}{256}\right)^2} \quad (2)$$

This definition (2) permits a segmentation which takes into account the geometrical shapes of the spots and the brightness information among them. In the same way, for colored microarray images with Cy3 and Cy5 hybridizations, we can consider 5D data with 2D geometrical coordinates and 3D color levels.

3 Parallel Spectral Clustering: method

The Gaussian affinity matrix A whose components are defined by (1) could be interpreted as a discretization of the Heat kernel [BELKIN, 2002]. And in particular, it is shown in [MOUYSSET, 2010] that this matrix is a discrete representation of the L^2 Heat operator onto appropriate connected domains in R^k. By combining tools from Heat equations and Finite Elements theory, the main result of [MOUYSSET, 2010] is that for a fixed data set of points, the eigenvectors of A are the representation of functions whose support is included in only one connected component at once. The accuracy of this representation is shown, for a fixed density of points, to depend on the affinity parameter. From this theoretical material, the Spectral Clustering could be formulated as a ”connected components” method in the sense that clustering in subdomains is equivalent in restricting the support of these L^2 particular eigenfunctions. So a ”divide and conquer” strategy could be formulated to adapt spectral clustering for parallel implementation. As the main drawback of domain decomposition is how to ensure uniform distribution of data per processor, the intrinsic property of microarray image can be exploited in that respect.

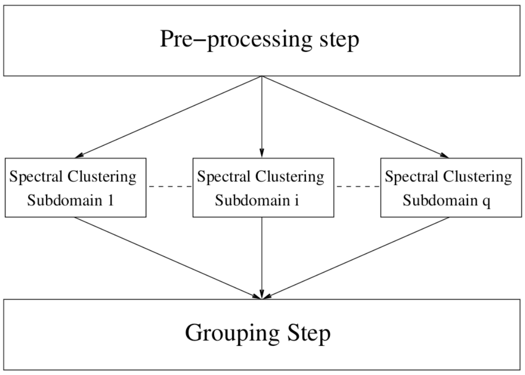
Pre−processing step
Spectral Clustering Subdomain 1
Spectral Clustering Subdomain i
Spectral Clustering Subdomain q
Grouping Step

Fig.1 Principle of parallel spectral clustering.

Due to the fact that microarray presents a block structure of cDNA spots, dividing the image in q sub-images is appropriate for a domain decomposition strategy because it ensures a uniform distribution of data per processor. An overlapping interface is investigated to gather the local partitions from the different subdomains.

This interface is characterized by an overlapping subset of points whose geometrical coordinates are close to the boundaries of neighboring subdomains. This partitioning will connect together clusters which belong to different subdomains thanks to the following transitive relation:

$$\begin{aligned}&\forall I_{i_1j_1}, I_{i_2j_2}, I_{i_3j_3} \in I,\\&if\ I_{i_1j_1}, I_{i_2j_2} \in C^1\ and\ I_{i_2j_2}, I_{i_3j_3} \in C^2\\&then\ C^1 \cup C^2 = P\ and\ I_{i_1j_1}, I_{i_2j_2}, I_{i_3j_3} \in P\end{aligned} \tag{3}$$

where I is the microarray image, C^1 and C^2 two distinct clusters and P a larger cluster which includes both C^1 and C^2. We experiment this strategy whose principle is represented in Fig.1 on several microarray images of the Saccharomyces cerevisiae database from the Stanford Microarray database (http://smd.stanford.edu/index.shtml) like the one in Fig.2.

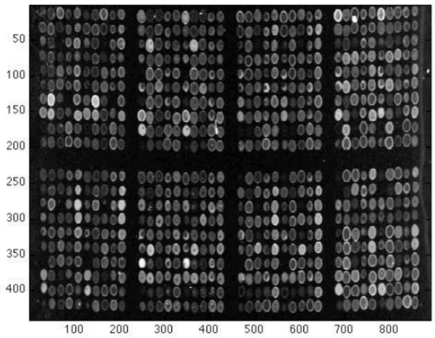

Fig.2 Block structure of microarray image.

It is important to see how the parallel approach can take advantage of the specificities of this particular application. Indeed, when splitting the original image into overlapping sub-pieces of images, the local spectral clustering analysis of each sub-piece involves the creation of many affinity matrices of smaller size. The total amount of memory needs for all these local matrices is much less than the memory needed for the affinity matrix covering the global image.

3.1 Choice of the affinity parameter

The Gaussian affinity matrix is widely used and depends on a free parameter which is the affinity parameter, noted σ, in equation (1). It is known [NG, 2002] that this parameter conditions the separability between clusters in spectral embedding space and affects the results. A global heuristics for this parameter was proposed in [MOUYSSET, 2008] in which both the dimension of the problem as well as the density of points in the given *p*-th dimensional data set are integrated. With an assumption that the data set is isotropic enough, the image data set I is included in a *p*-dimensional box bounded by $D_{\max}$ the largest distance d (defined by (2)) between pairs of points in I :

$$D_{\max} = \max_{\{1\le i,r\le l;1\le j,s\le m\}} d(I_{ij}, I_{rs}) \quad (4)$$

A reference distance which represents the distance in the case of an uniform distribution is defined as follows:

$$\sigma = \frac{D_{\max}}{n^{1/p}} \quad (5)$$

in which *n= l* x *m* is the size of the microarray image and *p=3* (resp. *p=5*) with 2D geometrical coordinates and 1D brightness (resp. 3D color). From this definition, clusters may exist if there are points that are at a distance no more than a fraction of this reference distance σ . This global parameter is defined with the whole image data set I and gives a threshold for all spectral clustering applied independently on the several subdomains.

3.2 Choice of the number of clusters

The problem of the right choice of the number of clusters k is crucial. We therefore consider in each subdomain a quality measure based on ratios of Frobenius norms, see for instance [MOUYSSET, 2008]. After indexing data points per cluster for a value of k , we define the indexed affinity matrix whose diagonal affinity blocks represent the affinity within a cluster and the off-diagonal ones the affinity between clusters (Fig.3).

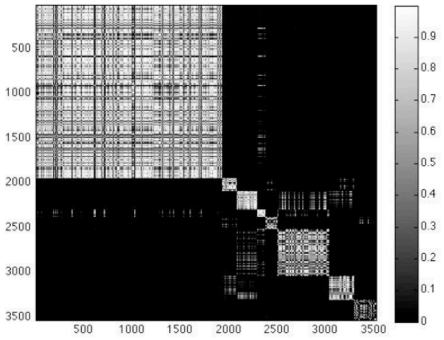

Fig.3 Block structure of the indexed affinity matrix for k=8 clusters.

The ratios, noted r_{ij} , between the Frobenius norm of the off-diagonal blocks (ij) and the norm of the diagonal ones (ii) could be evaluated. Among various values for k , the final number of cluster is defined so that the affinity between clusters is the lowest and the affinity within cluster is the highest:

$$k^* = \arg\min \sum_{i\neq j} r_{ij} \quad (6)$$

Numerically, the corresponding loop to test several values of k until satisfying (6) is not extremely costly but only requires to concatenate eigenvectors, apply *K-means*, and a reordering step on the affinity matrix to compute the ratios. Furthermore, this loop becomes less and less costly when the number of processors increases. This is due to the fact that eigenvectors become much smaller with affinity matrices of smaller size. Also, subdividing the whole data set implicitly reduces the Gaussian affinity to diagonal subblocks (after permutations).

For the *4* x *2* greyscaled spotted microarray image which corresponds to one subdomain of 3500 pixels, the original data set and its clustering result are plotted in Fig.4 for *k=8*.

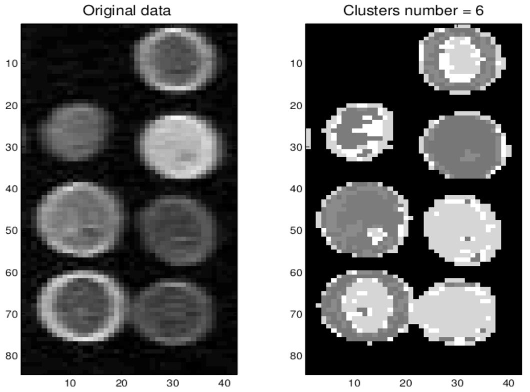

Fig.4 Clustering on one subdomain: 4x2 greyscaled spotted microarray image and its clustering result

3.3 Parallel Implementation of the Spectral Clustering Algorithm

The FORTRAN 90 implementation of the parallel Spectral Clustering Algorithm follows the MasterSlave paradigm with the MPI library to perform the communications between processors (algorithms 2 and 3). Classical routines from LAPACK library [ANDERSON, 1999] are used to compute selected eigenvalues and eigenvectors of the normalized affinity matrix A for each subset of data points.

```
Algorithm 2. Parallel Algorithm: Slave
  1: Receive the value and its data subset from the Master processor (MPI CALL)
  2: Perform the Spectral Clustering Algorithms on its subset
  3: Send the local partition and its number of clusters to the Master processor (MPI CALL)
```

```
Algorithm 3. Parallel algorithm: Master
  1: Pre-processing step
    1.1 Read the global data and the parameters.
    1.2 Split the data into $q$ subsets regarding the geometry
    1.3 Compute the affinity parameter $\sigma$ with the formula (5). The bandwidth of the overlapping is fixed to $3 \times \sigma$ .
  2: Send the sigma value and the data subsets to the other processors (MPI SEND)
  3: Perform Spectral clustering algorithm on subset
  4: Receive the local partitions and the number of clusters from each processor (MPI RECV)
  5: Grouping step
    5.1 Gather the local partitions in a global partition thanks to the transitive relation (3)
    5.2 Give as output a partition of the whole image $I$ and the final number of clusters $k$ are given.
```

4. Numerical Experiments

The numerical experiments were carried out on the Hyperion supercomputer of the CICT. With its 352 bi-Intel ”Nehalem” EP quad-core nodes it can develop a peak of 33TFlops. Each node has 4.5 GB memory dedicated for each of the cores and an overall of 32 GB fully available memory on the node that is shared between the cores.

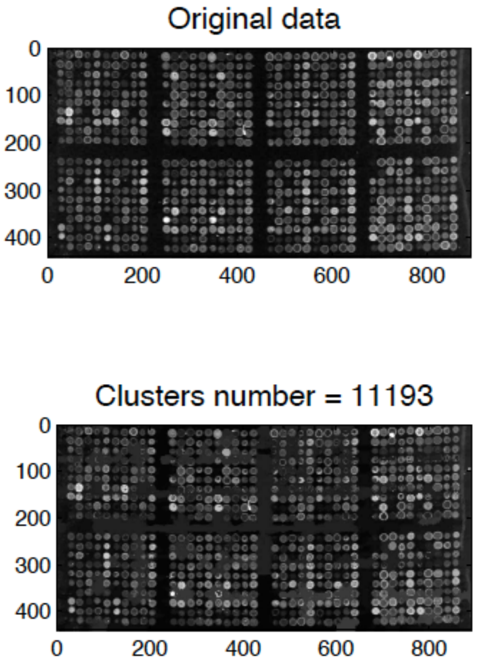

Fig.5 Original microarray image and its clustering result

For our tests, the domain is successively divided in q={18,32,45,60,64} subboxes. The timings for each step of parallel algorithm are measured. We test this Parallel Spectral Clustering on one microarray image from the Stanford Microarray Database. For decomposition in 64 subboxes, the original microarray image of 392931 pixels which represents 8 blocks of 100 spots and the clustering result are plotted in Fig.5. After the grouping step, the parallel spectral clustering result has determined 11193 clusters. Compared to the original data set, the shapes of the various hybridization spots are well described.

We give in Table 1, for each distribution, the number of points on each processor, the time in seconds to compute σ defined by (5), the time in the parallel Spectral Clustering step, the time of the grouping phase and the total time and the memory consumption in GigaOctets. The first remark is that the total time decreases drastically when we increase the number of processors. Logically, this is time of the parallel part of the algorithm (step 3) that decreases while the two other steps (1 and 5), that are sequential, remain practically constant. To study the performance of our parallel algorithm, we compute the speedup. Because we cannot have a result with only one processor in order to have a sequential reference (lack of memory), we take the time with the 18 processors, the minimum number of processors in order to have enough memory by processor. The speedup for q processors will then be defined as T_{18}/T_q .

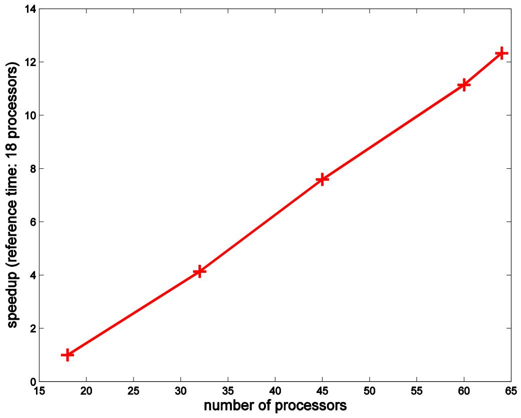

Fig.6: Performances of the parallel part: Speedup with the 18 processors time as reference

We can notice in Fig.6 that the speedups increase faster than the number of processors: for instance, from 18 to 64 processors, the speedup is 12 although the number of processors grows only with a ratio 3.55. This good performance is confirmed if we draw the mean computational costs per point of the image.

Table 1 **Microrarray image segmentation results for different splittings.**

Number of proc.	Number of points	Time σ	Time parallel SC	Time Grouping	Total Time	Memory Cons.
18	22000	1413	36616	892	38927	7.75
32	12500	1371	7243	794	9415	2.50
45	9000	1357	2808	953	5127	1.30
60	6800	1360	1153	972	3495	0.74
64	6300	1372	1030	744	3157	0.64

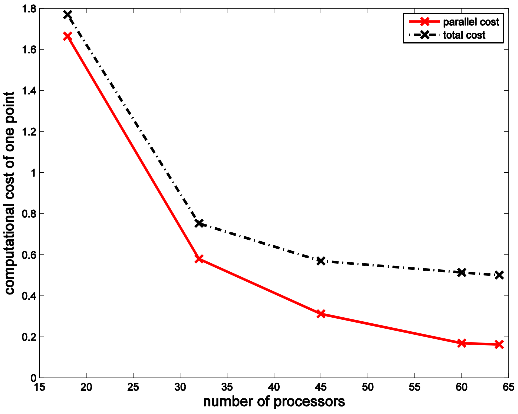

Fig.7 Parallel and total computational costs

We define, for a given number of processors, the parallel computational cost (resp. total computational cost) the time spent in the parallel part (parallel Spectral Clustering part) (resp. total time) divided by the average number of points on each subdomain. We give in Fig.7, these parallel (plain line) and total (dashed line) computational costs.

We can observe from Table 1 that the fewer points we have per subset, the faster we go and the decreasing is better than linear. This can be explained by the non-linearity of our problem which is the computation of eigenvectors from the Gaussian affinity matrix. There are much better gains in general when smaller subsets are considered.

5 Conclusion

With the domain decomposition strategy and heuristics to determine the choice of the Gaussian affinity parameter and the number of clusters, the parallel spectral clustering becomes robust for microarray image segmentation and combines intensity and shape features. The numerical experiments show the good behaviour of our parallel strategy when increasing the number of processors and confirm the suitability of our method to treat microarray images.

However, we find two limitations: the lack of memory when the subset given to a processor is large and the time spent in the sequential parts which stays roughly constant and tends to exceed the parallel time with large number of processors.

To reduce the problem of memory but also to reduce the spectral clustering time, we studied sparsification techniques [MOUYSSET, 2013] in the construction of affinity matrix by dropping some components that correspond to points at a distance larger than a threshold. A threshold based on uniform distance was defined for any kind of data distribution. This distance could be considered as a limit threshold to preserve the clustering results.

We validate this approach in Matlab by showing that the number of non zero of the affinity matrix decreases with still some good results in terms of spectral clustering and even some gains in the time spent to compute the affinity matrix.
These results are confirmed when we use sparsification with our parallel spectral clustering solver. We can show that we are able to reduce significantly the size of the affinity matrix without loosing the quality of the segmentation solution.

With sparse structures to store the matrix, we will also gain a lot of memory. However, we may have to adapt our eigenvalues solver and use for example ARPACK library [LEHOUCQ, 1998]. To reduce the time of the sequential parts, we could also investigate parallelization of the computation of the σ parameter and the ability to separate the spotted microarray image in sub-images.

6 Acknowledgment

This work was performed using HPC resources from CALMIP (Grant 2012-p0989).

7 References

[ANDERSON, E. *et al*. 1999] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, et al. *LAPACK Users' guide*. Society for Industrial Mathematics, 1999.

[BELKIN, M. *et al*. 2002] M. Belkin and P. Niyogi. *Laplacian eigenmaps and spectral techniques for embedding and clustering*. Advances in Neural Information Processing Systems, 2002.

[CHEN, W.-Y. *et al*. 2010] W.-Y. Chen, Y. Song, S. Yangqiu, H. Bai, C.-J. Lin, and E. Y. Chang. *Parallel Spectral Clustering in Distributed Systems*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011.

[FWOLKES, C.*et al*.2010] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. *Spectral grouping using the Nytrom method*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004.

[GIANNAKEAS,N. *et al* 2008] N. Giannakeas and D. Fotiadis. *Image Processing and Machine Learning Techniques for the Segmentation of cDNA Microarray Images*. Handbook of research on advanced techniques in diagnostic images and biomedical application, 2008.

[GLEZ-PENA, D. et al 2009] D. Glez-Peña; F. Díaz, J.M. Hernández; J.M. Corchado and F. Fdez-Riverola. *geneCBR: a translational tool for multiple-microarray analysis and integrative information retrieval for aiding diagnosis in cancer research*. BMC Bioinformatics, 2009.

[LEHOUCQ,R. *et al 199*8] R. Lehoucq, D. Sorensen, and C. Yang. *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*. SIAM, 1998.

[Ng, A.Y. *et al*. 2002] A. Y. Ng, M. I. Jordan, and Y. Weiss. *On spectral clustering: analysis and an algorithm*. Proceedings in Advance Neural Information Processing Systems, 2002.

[MOUYSSET, S. *et al*. 2008] S. Mouysset, J. Noailles, and D. Ruiz. *Using a Global Parameter for Gaussian Affinity Matrix in Spectral Clustering*, Lecture Notes in Computer Science, Springer-Verlag, 2008.

[MOUYSSET, S. *et al*. 2010] S. Mouysset, J. Noailles, and D. Ruiz. *On an Interpretation of Spectral Clustering via Heat Equation and Finite Elements Theory*. Lecture Notes in Engineering and Computer Science, 2010.

[MOUYSSET, S. *et al*. 2013] S. Mouysset and R. Guivarch. *Sparsification on Parallel Spectral Clustering*. Lecture Notes in Computer Science, Springer-Verlag, 2013 (to appear).

[RUEDA, I. *et al*. 2005] L. Rueda and L. Qin. *A new method for DNA microarray image segmentation*. Image Analysis and Recognition, 2005.

[RUEDA, I. *et al*. 2009] L. Rueda and J. Rojas. *A Pattern Classification Approach to DNA Microarray Image Segmentation*. Pattern Recognition in Bioinformatics, pages 319–330, 2009.

[SONG, Y. *et al*. 2008] Y. Song, W.-Y. Chen, H. Bai, C. Lin, and E. Chang. *Parallel spectral clustering*. Proceedings of European Conference on Machine Learning and Pattern Knowledge Discovery. Springer, 2008.

[USLAN, V. *et al*. 2010] V. Uslan, O. Bucak, and B. Cekmece. *Microarray image segmentation using clustering methods*. Mathematical and Computational Applications, 2010.

[YANG, Y. *et al*. 2001] Y. Yang,M. Buckley, and T. Speed. *Analysis of cDNA microarray images*. Briefings in bioinformatics, 2001.