Parallel Spectral Clustering for the segmentation of cDNA Microarray Images

Sandrine Mouysset, Ronan Guivarch, Joseph Noailles, and Daniel Ruiz

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 method 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 and 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 [7], [9], [3]. 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 [10], [6]. 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 and should not depend on many input parameters. Spectral methods, and in particular the spectral clustering algorithm introduced by Ng-Jordan-Weiss [5], are useful when considering no a priori shaped subsets of data. 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 [8], [2] were recently suggested, mainly

Sandrine Mouysset
University of Toulouse - UPS - IRIT, 118 Route de Narbonne, 31062 Toulouse, France
e-mail: sandrine.mouysset@irit.fr

Ronan Guivarch, Joseph Noailles and Daniel Ruiz
University of Toulouse - INPT(ENSEEIHT) - IRIT, 2 rue Camichel, 31071 Toulouse, France
e-mail: {ronan.guivarch,joseph.noailles,daniel.ruiz}@enseeiht.fr

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 Parallel Spectral Clustering: justifications

2.1 Spectral Clustering

Let's first give some notations and recall the Ng-Jordan-Weiss algorithm [5]. Let's consider a microarray image I of size $l \times m$. 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 $A \in \mathbb{R}^{n\times n}$ with $n = l \times m$ defined by equation (1).
2. Construct the normalized matrix: $L = D^{-1/2}AD^{-1/2}$ with $D_{i,i} = \sum_{r=1}^{n} A_{ir}$,
3. Assemble the matrix $X = [X_1 X_2 .. X_k] \in \mathbb{R}^{n\times k}$ by stacking the eigenvectors associated with the k largest eigenvalues of L,
4. Form the matrix Y by normalizing each row in the $n \times k$ matrix X,
5. Treat each row of Y as a point in $\mathbb{R}^k$, 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 pixel of coordinates (i, j) and (r, s) for $i, r \in \{1, .., l\}$ and $j, s \in \{1, .., m\}$. After a normalization step, the k largest eigenvectors are extracted. So every data point I_{ij} is plotted in a spectral embedding space of $\mathbb{R}^k$ and the clustering is made in this space by applying K-means method. Finally, thanks to an equivalence relation, the final partition of data set is 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 some 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 affin-

ity A_{ir} is defined as follows:

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

where σ is the affinity parameter and the distance d between the pixel (ij) and (rs) 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 Method

By exploiting the block structure of microarrays, clustering can be made on subdomains by breaking up the data set into data subsets with respect to their geometrical coordinates in a straightforward way. With an appropriate Gaussian affinity parameter and a method to determine the number of clusters, each processor applies independently the spectral clustering (Algorithm 1) on a subset of data points and provides a local partition on this data subset. Based on these local partitions, a grouping step ensures the connection between subsets of data and determines a global partition thanks to the following transitive relation: $\forall I_{i_1 j_1}, I_{i_2 j_2}, I_{i_3 j_3} \in I$,

$$\text{If } I_{i_1 j_1}, I_{i_2 j_2} \in C^1 \text{ and } I_{i_2 j_2}, I_{i_3 j_3} \in C^2 \text{ then } C^1 \cup C^2 = P \text{ and } I_{i_1 j_1}, I_{i_2 j_2}, I_{i_3 j_3} \in P \quad (3)$$

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(a) on several microarray images of the Saccharomyces cerevisiae database from the Stanford Microarray database[1] like the one in Fig.1(b).

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. Additionally, the analysis of each subproblem is made from the extraction of eigenvectors in the scaled affinity sub-matrices, keeping in mind that one eigenvector is needed for each identified cluster in the corresponding sub-image. In that respect, the parallel approach enables us to decrease drastically the cost of this eigenvector computation.

[1] http://smd.stanford.edu/index.shtml

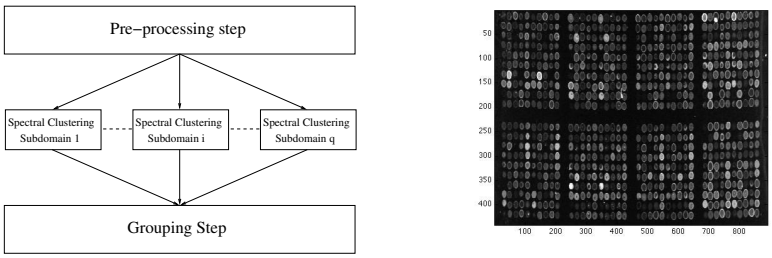

Fig. 1 Principle of the parallel strategy for microarray image : (a) Principle of parallel spectral clustering and (b) Block structure of microarray image.

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 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 [1] 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:

$$D_{\max} = \max_{\substack{1\leq i,r\leq l \\ 1\leq j,s\leq m}} d(I_{ij}, I_{rs}).$$

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

$$\sigma = \frac{D_{\max}}{n^{\frac{1}{p}}}, \tag{4}$$

in which $n = l \times m$ 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 [1]. 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.2(a). The ratios, noted r_{ij}, between the Frobenius norm of the off-diagonal blocks (ij) and

that 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:

$$\hat{k} = argmin \sum_{i \neq j} r_{ij}. \tag{5}$$

Numerically, the corresponding loop to test several values of k until satisfying (5) 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 4×2 greyscaled spotted microarray image which corresponds to one subdomain, the original data set and its clustering result are plotted in Fig.2(b) for $k = 8$.

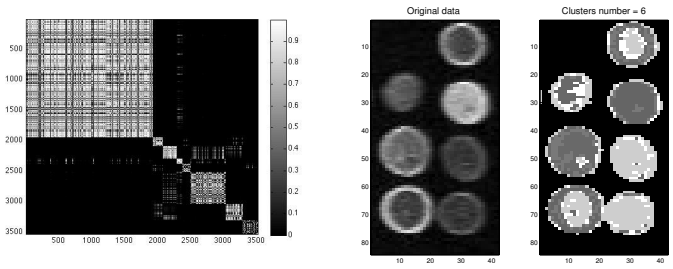

Fig. 2 Clustering on one sub-domain made by 4×2 greyscaled spotted microarray image (3500 pixels): (a) Block structure of the indexed affinity matrix for $k = 8$ and (b) Original data and its clustering result.

3.3 Parallel Implementation of the Spectral Clustering Algorithm

The FORTRAN 90 implementation of the parallel Spectral Clustering Algorithm follows the Master-Slave paradigm with the MPI library to perform the communications between processors (algorithms 2 and 3).

Algorithm 2 Parallel Algorithm: Slave

1: Receive the sigma value and its data subset from the Master (MPI_RECV)
2: Perform the Spectral Clustering Algorithm on its subset
3: Send the local partition and its number of clusters to the Master (MPI_SEND)

Algorithm 3 Parallel Algorithm: Master

1: Pre-processing step
   1.1 Read the global data and the parameters
   1.2 Split the data into subsets regarding the geometry
   1.3 Compute the affinity parameter σ
2: Send the sigma value and the data subsets to the other processors (MPI CALL)
3: Perform the Spectral Clustering Algorithm on subset 1
4: Receive the local partitions and the number of clusters from each processor (MPI CALL)
5: Grouping Step
   5.1 Gather the local partitions in a global partition thanks to the transitive relation
   5.2 Give as output a partition of the whole data set S and the final number of clusters k

4 Numerical Experiments

The numerical experiments were carried out on the Hyperion supercomputer[2] of the CICT. 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 a decomposition in 64 subboxes, the clustering result is plotted in Fig.3. The original microarray image of 392931 pixels which represents 8 blocks of 100 spots is plotted on the left of the figure. After the grouping step, the parallel spectral clustering result has determined 11193 clusters which are plotted on the right of Fig.3. Compared to the original data set, the shapes of the various hybridization spots are well described.

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

We give in Table 1, for each distribution, the number of points on each processor, the time in seconds to compute σ defined by (4), the time in the parallel Spectral Clustering step, the time of the grouping phase, 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.

[2] http://www.calmip.cict.fr/spip/spip.php?rubrique90

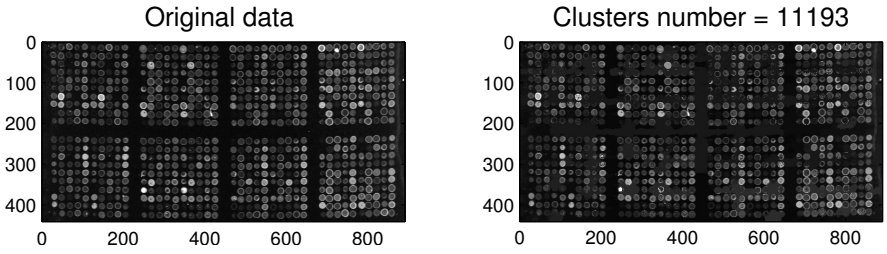

Fig. 3 Original microarray image and its clustering result.

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 $S_q = \frac{T_{18}}{T_q}$.

We can notice in Fig.4(a) 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.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.4 (b), these parallel (plain line) and total (dashed line) computational costs.

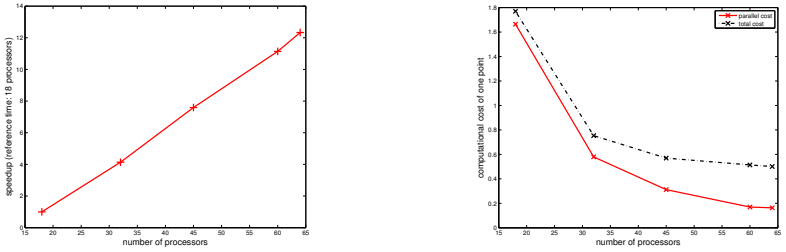

Fig. 4 Performances of the parallel part : (a) Speedup with the 18 processors time as reference and (b) Parallel and total computational costs.

We can observe from Table 1 that the less points we have by 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 can study some techniques for sparsifying the Gaussian affinity matrix: some sparsification techniques, such as thresholding the affinity between data points, could also be introduced to speed up the algorithm when the subdomains are still large enough. 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 [4]. To reduce the time of the sequential parts, we could investigate parallelization of the computation of the σ parameter and ability to separate the spotted microarray image in sub-images.

References

1. S. Mouysset, J. Noailles, and D. Ruiz. Using a Global Parameter for Gaussian Affinity Matrix in Spectral Clustering. In *Lecture Notes in Computer Science*, pages 378–390. Springer-Verlag, juin 2008.
2. W.-Y. Chen, S. Yangqiu, H. Bai, C.-J. Lin, and E. Y. Chang. Parallel Spectral Clustering in Distributed Systems. *Preprint of IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.
3. N. Giannakeas and D. Fotiadis. Image Processing and Machine Learning Techniques for the Segmentation of cDNA Microarray Images, 2008.
4. R. Lehoucq, D. Sorensen, and C. Yang. *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*. Siam, 1998.
5. A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: analysis and an algorithm. *Proceedings in Advance Neural Information Processing Systems*, 2002.
6. L. Rueda and L. Qin. A new method for DNA microarray image segmentation. *Image Analysis and Recognition*, pages 886–893, 2005.
7. L. Rueda and J. Rojas. A Pattern Classification Approach to DNA Microarray Image Segmentation. *Pattern Recognition in Bioinformatics*, pages 319–330, 2009.
8. Y. Song, W. Chen, H. Bai, C. Lin, and E. Chang. Parallel spectral clustering. In *Proceedings of European Conference on Machine Learning and Pattern Knowledge Discovery*. Springer, 2008.
9. V. Uslan, O. Bucak, and B. Cekmece. Microarray image segmentation using clustering methods. *Mathematical and Computational Applications*, 15(2):240–247, 2010.
10. Y. Yang, M. Buckley, and T. Speed. Analysis of cDNA microarray images. *Briefings in bioinformatics*, 2(4):341, 2001.