Bioinformation by Biomedical Informatics Publishing Group *open access*

www.bioinformation.net **Prediction Model**

Finding distinct biclusters from background in gene expression matrices

Zhengpeng Wu [1, $], Jiangni Ao [1, $], Xuegong Zhang[1, *]

[1]Bioinformatics Division, TNLIST and Department of Automation, Tsinghua University, Beijing 100084, China; [$]These authors contributed equally to this work; Xuegong Zhang* - E-mail: zhangxg@tsinghua.edu.cn; *Corresponding author

received December 15, 2007; accepted December 30, 2007; published online December 30, 2007

Abstract:
Biclustering, or the discovery of subsets of samples and genes that are homogeneous and distinct from the background, has become an important technique in analyzing current microarray datasets. Most existing biclustering methods define a bicluster type as a fixed (predefined) pattern and then trying to get results in some searching process. In this work, we propose a novel method for finding biclusters or 2-dimensional patterns that are significantly distinct from the background without the need for pre-defining a pattern within the bicluster. The method named Distinct 2-Dimensional Pattern Finder (D2D) is composed of an iterative reordering step of the rows and columns in the matrix using a new similarity measure, and a flexible scanning-and-growing step to identify the biclusters. Experiments on a large variety of simulation data show that the method works consistently well under different conditions, whereas the existing methods compared may work well under some certain conditions but fail under some other conditions. The impact of noise levels, overlapping degrees between clusters and different setting of parameters were also investigated, which indicated that the D2D method is robust against these factors. The proposed D2D method can efficiently discover many different types of biclusters given that they have distinctive features from the background. The computer program is available upon request.

Keywords: gene expression matrices; simulation; biclusters; Distinct 2-Dimensional (D2D); noise

Background:

Clustering is a family of techniques very useful in finding meaningful subsets from datasets in an unsupervised manner. Hartigan (1972) **[1]** proposed a technique for clustering objects and coordinates simultaneously, which was the origin of the so-called biclustering problem. With the rapid accumulation of high-throughput genomic and proteomic data such as microarray-based gene expression data, biclustering algorithms are becoming important tools in data mining.

Microarray techniques measure the expression levels of a large number of genes simultaneously under tens or hundreds of conditions **[2]**. Traditional clustering techniques analyze microarray data along with two directions, clustering genes and clustering conditions **[3]**. However, from a biological viewpoint, usually only a subset of genes participates in a particular cellular process. So finding a sub-matrix distinguishable from background becomes a worthwhile task, where the biclustering problem emerges naturally.

Many methods have been developed within the microarray research area to solve the biclustering problem (e.g., see the review of Madeira and Oliveira 2004) **[4]**. Cheng and Church (2000) **[5]** were the first to apply biclustering methodology to microarray expression data. They proposed a deletion-addition algorithm (CC) with greedy nature to find a given number of δ-biclusters, whose mean squared residues (MSRs) are under some given thresholds. Ben-Dor *et al* (2002) **[6]** proposed another greedy iterative algorithm (OPSM) to identify statistical significant order preserving sub-matrices. Ihmels *et al* (2002, 2004) **[7, 8]** derived an iterative signature algorithm (ISA) using several initial gene sets to find up- or down-regulated biclusters. Murali and Kasif (2003) **[9]** introduced the concept of 'xMotif' which is a bicluster with coherent evolution on the rows and proposed a random algorithm (xMotif) to find the largest 'xMotif'. Prelic *et al* (2006) **[10]** systematically compared some different prominent biclustering methods, and also provided a divide-and-conquer biclustering method (Bimax) aimed at finding up- or down-regulated biclusters. However, Bimax needs discretization of expression data into a binary matrix beforehand. Liu and Wang (2007) [11] proposed a novel similarity score to evaluate a bicluster and developed a deletion-addition algorithm (RMSBE) to find optimal biclusters using their similarity score.

All these methods follow a common basic strategy: they first define a bicluster type as a fixed pattern and then try to get results with a searching procedure. This strategy has its inherent limitations. Since many possible types of biclusters may occur under different biological scenarios, it is not easy in practice to predefine the pattern one wants to search for in a microarray dataset. This is reflected by the fact that some existing methods can perform well in some sets of data but may not be suitable for other conditions **[10, 11]**. This situation makes it difficult for the biologist to make the appropriate selection. Therefore, a method that is more flexible in the requirement for a bicluster pattern definition and suits more conditions is needed.

Similar to traditional unsupervised learning problems, a key issue in biclustering is the proper definition of the biclusters. From the study of many microarray datasets and the existing publications on biclustering methods, we perceive that the various kinds of biclusters that biologists are interested in have one key feature in common, no matter what specific patterns they follow. This feature is that biclusters should be homogeneous and be significantly distinguishable from the background. Based on this understanding, we proposed a new method for finding biclusters from microarray data. We call it as Distinct 2-Dimensional Pattern Finder (or D2D for short) as a bicluster can be viewed as a distinctive 2-dimensional block embedded in the microarray data matrix. We compared the method with some representative existing methods under a large variety of simulated situations. Experimental results show that the proposed method performs consistently well under all the experimental conditions, whereas the performance of other methods vary with the datasets.

Methodology and results:

Definitions and notations

Basic notations

Let G be a set of genes, C a set of conditions, and $E(G,C)$ the expression matrix, where $G=\{1,2,\ldots,m\}$ and $C=\{1,2,\ldots,n\}$. The element e_{ij} of $E(G,C)$ represents the expression level of gene i under condition j.

The aim of biclustering is to extract the sub-matrix $E(G',C')$ (or sub-matrices) of $E(G,C)$ meeting some criteria, which is identified by gene subset G' of G and condition subset C' of C.

Biclusters

The definition of ‘cluster’ in biclustering methodology is different from the ‘cluster’ used by classical clustering. The aim of classical clustering is to cluster heterogeneous objects into homogeneous groups. While in the biclustering problem, the aim is to extract some ‘good’ sub-matrices rather than partition any objects into groups. We could only describe some properties of sub-matrices as the criteria to define the biclusters.

Concerning the biological interpretation, there are mainly two kinds of biclusters according to their appearances **[10]**. The first is the simplest situation, constant biclusters. A perfect constant bicluster is a sub-matrix, in which all the entries are almost the same. The second is the natural generalization of constant biclusters, additive biclusters. An additive bicluster is a sub-matrix, whose rows or columns form an arithmetical progression. When subtracted by a certain row and column from the rows and columns of the additive biclusters respectively, the bicluster will be transformed into a constant bicluster. Generally speaking, a bicluster is always masked by noise. Figure 1 illustrates these two situations.

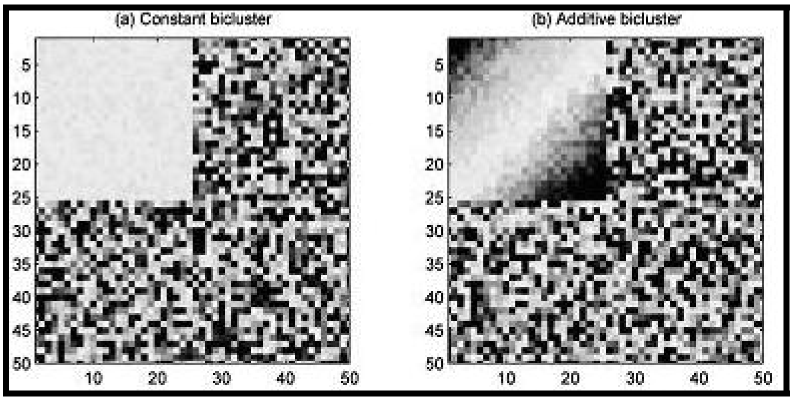


Figure 1: Definition of biclusters. (a) Constant bicluster; (b) Additive bicluster

Local distance definition

We will introduce a new distance definition named ‘local distance’ to evaluate the similarity between different entries as follows. For two genes p and q, whose row vectors are $[e_{p,1},\ldots,e_{p,m}]$ and $[e_{q,1},\ldots,e_{q,m}]$ respectively, we calculate their difference vector $R_{pq}=[R_{pq,1},\ldots,R_{pq,m}]=[(e_{p,1}-e_{q,1}),\ldots,(e_{p,m}-e_{q,m})]$. Then we have the following investigations.

If genes p and q originate from the same constant bicluster, their row vectors would be similar to each other, and the difference vector R_{pq} would have many values close to zero (i.e. the histogram of R_{pq} will be concentrated). On the contrary, if genes p and q come from different constant biclusters, the difference vector R_{pq} would have many values far from zero (i.e. the histogram of R_{pq} would be more dispersed from zero).

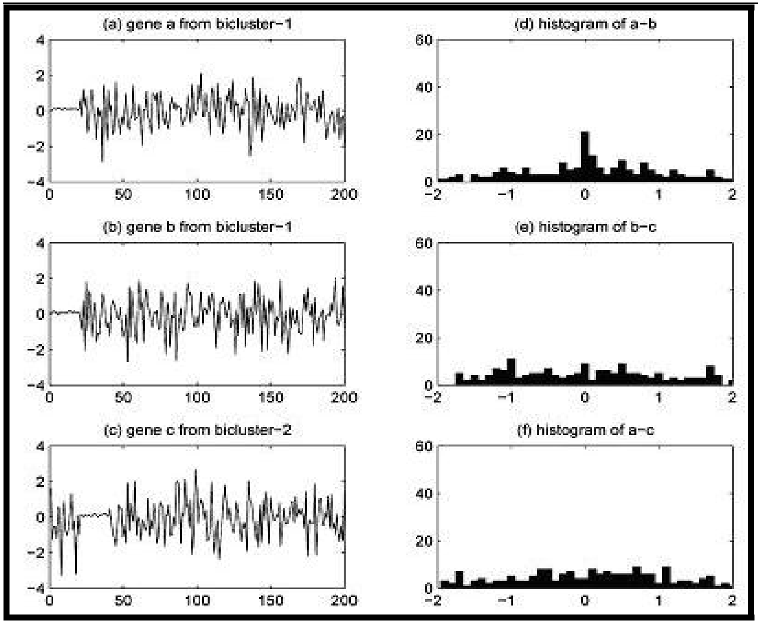


Figure 2: Illustration of local distance definition - Left three figures show three genes' expression values. Genes *a* and *b* are from the same bicluster and gene *c* originates from another bicluster. Right three figures show the histograms of (*a*−*b*), (*b*−*c*) and (*a*−*c*) respectively

As shown in Figure 2, the background is sampled from normal distribution N(0,1) and the bicluster is sampled from normal distribution N(0.1,0.1). Genes *a* and *b* are from the same bicluster and gene *c* comes from another bicluster. Obviously the histogram of (*a-b*) is more concentrated around zero than that of (*b-c*) and (*a-c*).

If we set a threshold called δ, and then count the number of elements of R_{pq} whose absolute values are less than δ, denoting it by N_{pq}, we will get the distance between genes *p* and *q* as given in equation (1) (see supplementary material).

As shown in Figure 2, if two genes come from the same bicluster, their difference vector would have more values close to zero than that of genes from different biclusters. Correspondingly the NR_{pq} will be bigger and DR_{pq} will be smaller. A very similar definition is made for distance between columns as shown in formula (2) (given under supplementary material).

As this kind of distance depends only on a small portion of elements of the rows or columns, it reflects the local (versus global) property of the rows or columns. We named it as local distance. Next, we will present D2D separately according to its framework in the following paragraphs.

Algorithms

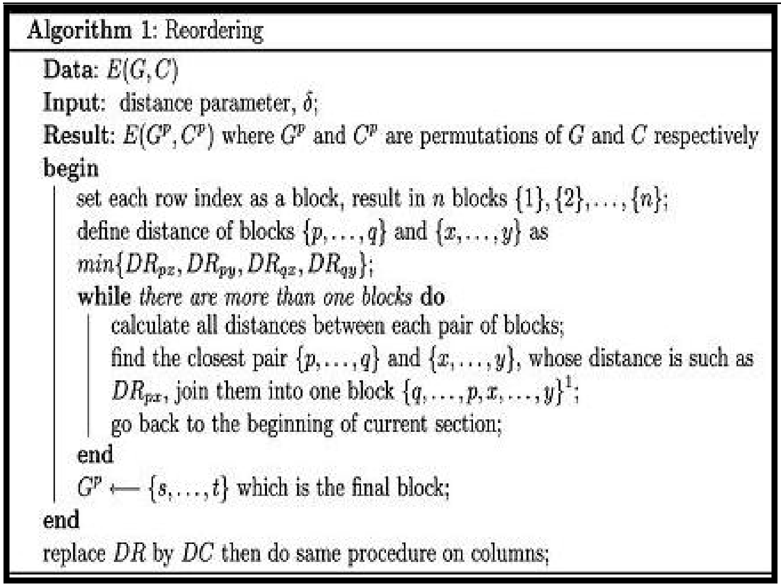
In this section we present a two-step strategy to tackle the biclustering problem.

Reordering

In the first place, we rearrange the data matrix to make the rows or columns sharing more similarities aggregate together. This novel reordering strategy is carried out under local distance as defined above.

For a given $m*n$ data matrix, the sketch of reordering is as follows: (1) start with each row of the data matrix as a block; (2) calculate the local distance of every two blocks and merge the nearest two blocks as a new block. The process of mergence occurs only once and we will get (m-1) blocks here; (3) repeat the above process until all the rows are merged into one block; (4) apply the same procedure on the columns of the data matrix.

In step 2, two points should be emphasized. Firstly, the local distance between blocks is a little different from formula (1) and (2). For two blocks ($p,\ldots,q$) and ($x,\ldots,y$), we are only interested in the edge of the blocks: *p*, *q*, *x* and *y*. We define the distances between blocks ($p,\ldots,q$) and ($x,\ldots,y$) as min(DR_{px}, DR_{py}, DR_{qx}, DR_{qy}). Secondly, when we merge two blocks, there is a special 'head to head' rule to maintain the structure of the parental blocks. Figure 3 illustrates the details of this procedure.



```
Algorithm 1: Reordering
Data: $E(G,C)$
Input: distance parameter, $\delta$;
Result: $E(G^p,C^p)$ where $G^p$ and $C^p$ are permutations of $G$ and $C$ respectively
begin
    set each row index as a block, result in $n$ blocks $\{1\},\{2\},\dots,\{n\}$;
    define distance of blocks $\{p,\dots,q\}$ and $\{x,\dots,y\}$ as
    $min\{DR_{px},DR_{py},DR_{qx},DR_{qy}\}$;
    while there are more than one blocks do
        calculate all distances between each pair of blocks;
        find the closest pair $\{p,\dots,q\}$ and $\{x,\dots,y\}$, whose distance is such as
        $DR_{px}$, join them into one block $\{q,\dots,p,x,\dots,y\}^1$;
        go back to the beginning of current section;
    end
    $G^p \longleftarrow \{s,\dots,t\}$ which is the final block;
end
replace $DR$ by $DC$ then do same procedure on columns;
```

Figure 3: Pseudo-code of reordering approach

Further modification
To identify the additive biclusters, we need to transform additive biclusters into constant biclusters. Ihmels *et al* (2002, 2004) **[7, 8]** and Liu and Wang (2007) **[11]** utilized a strategy using reference genes and conditions to solve this problem.

In an additive bicluster, the expression values of genes fluctuate in the similar way as the reference gene and condition. Thus, if we know the reference gene *i* and reference condition *j,* transformation ‘$b_{ij}=e_{ij}-e_{i'j}-e_{ij'}$’ will transform the additive bicluster into a constant bicluster. In fact, we usually do not know the reference gene *i* and reference condition *j* beforehand, so we would have to try every gene and condition as references, which would be a large amount of work. We can also do some random sampling to reduce the computation time **[7, 8, 11]**, which may lose some good results.

Scanning and growing
After reordering the data matrix, we have aggregated together the similar entries of the original matrix as closely as possible. In the next step, we devised a systematic scanning-and-growing procedure based on similarity to identify the biclusters.

Concerning of the real situation, there are two sub-categories of biclusters: the ones that have different brightness from the background, which correspond to up- or down-regulated expression patterns; and the ones that are more homogeneous compared to the background, which correspond to similarly expressed patterns.

For the first sub-category, the dominant feature is the mean value within the sub-matrix. For the second sub-category, the dominant feature is the consistence within the sub-matrix, and we used the mean squared residue (MSR) proposed by Cheng and Church (2000) **[5]** as the evaluation of consistence within a bicluster as explained in equation (3) and equation (4) (see supplementary material for equations).

As the score *H(I,J)=0* means that the expressions of the bicluster fluctuate in unison we can apply this score to find constant or additive biclusters.

D2D deals with the above two situations simultaneously. In real situations, this generality would cover more complex bicluster patterns resulting from the combination of above two sub-categories.

The sketch of scanning process is to sample around the matrix and get the distribution of some key parameters. We then choose those sub-matrices that are statistically significant as the local optimal bicluster named ‘seeds’. In the growing procedure, we optimize the seeds trying to get the global optimized results.

Scanning
In this step we aim to identify those local optimal sub-matrices as seeds. In detail, we use a template, which is a small window of size (*a*, *b*), to scan the data matrix, calculate and record the mean values and MSR within every window. By doing this we get two distributions of the mean value and MSR. In order to get the sub-matrices distinguished from the background, we choose those sub-matrices which are statistically significant as seeds. Specifically, we use three parameters to choose the statistically significant sub-matrices: p_{ml} and p_{mh} are the two tail probabilities of mean value distribution where the former is the lower and the latter is the higher; p_{vl} is the tail probability of MSR distribution. p_{ml} and p_{mh} correspond to m_l and m_h respectively as shown in figure 4a; p_{vl} corresponds to v_l as shown in figure 4b.

We choose the sub-matrices whose mean values are less than m_l or greater than m_h or whose MSR are less than p_{vl} as seeds.

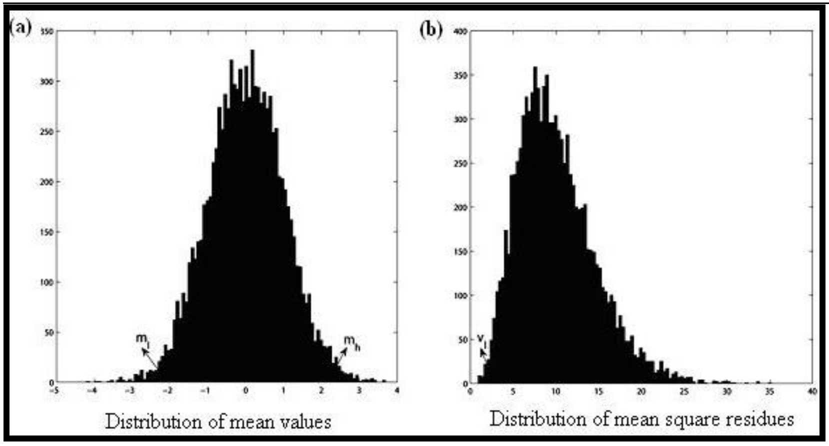
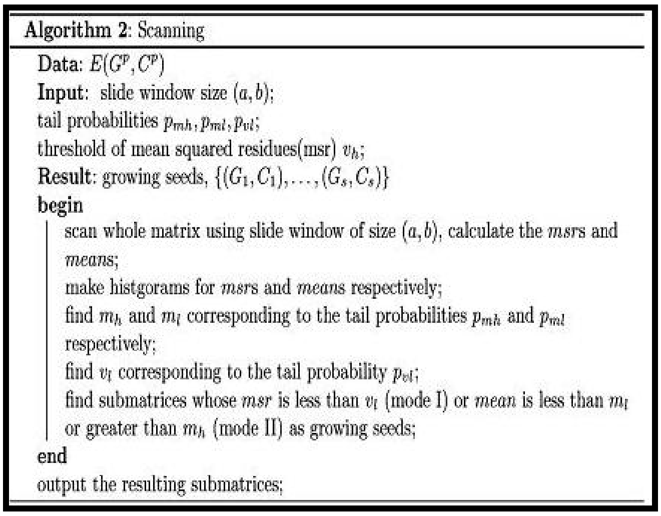
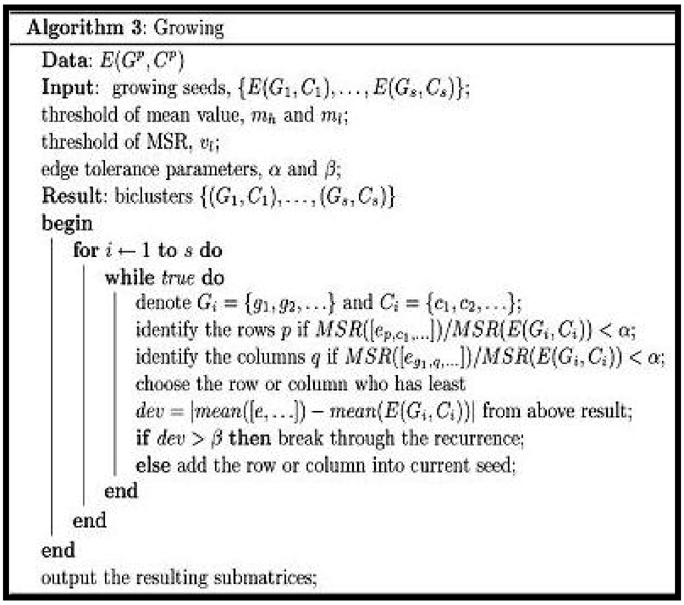


Figure 4: Parameters used for scanning: (a) is the distribution of mean values and (b) is the distribution of MSR. m_l and m_h correspond to the tail probabilities p_{ml} and p_{mh} respectively. v_l correspond to the tail probability p_{vl}



```
Algorithm 2: Scanning
  Data: $E(G^p, C^p)$
  Input: slide window size $(a, b)$;
  tail probabilities $p_{mh}, p_{ml}, p_{vl}$;
  threshold of mean squared residues(msr) $v_h$;
  Result: growing seeds, $\{(G_1, C_1), \ldots, (G_s, C_s)\}$
  begin
      scan whole matrix using slide window of size $(a, b)$, calculate the $msrs$ and
      $means$;
      make histgorams for $msrs$ and $means$ respectively;
      find $m_h$ and $m_l$ corresponding to the tail probabilities $p_{mh}$ and $p_{ml}$
      respectively;
      find $v_l$ corresponding to the tail probability $p_{vl}$;
      find submatrices whose $msr$ is less than $v_l$ (mode I) or $mean$ is less than $m_l$
      or greater than $m_h$ (mode II) as growing seeds;
  end
  output the resulting submatrices;
```

Figure 5: Pseudo-code of scanning approach



```
Algorithm 3: Growing
  Data: $E(G^p, C^p)$
  Input: growing seeds, $\{E(G_1, C_1), \ldots, E(G_s, C_s)\}$;
  threshold of mean value, $m_h$ and $m_l$;
  threshold of MSR, $v_l$;
  edge tolerance parameters, $\alpha$ and $\beta$;
  Result: biclusters $\{(G_1, C_1), \ldots, (G_s, C_s)\}$
  begin
      for $i \leftarrow 1$ to $s$ do
          while $true$ do
              denote $G_i = \{g_1, g_2, \ldots\}$ and $C_i = \{c_1, c_2, \ldots\}$;
              identify the rows $p$ if $MSR([e_{p,c_1,\ldots}])/MSR(E(G_i, C_i)) < \alpha$;
              identify the columns $q$ if $MSR([e_{g_1,q,\ldots}])/MSR(E(G_i, C_i)) < \alpha$;
              choose the row or column who has least
              $dev = |mean([e, \ldots]) - mean(E(G_i, C_i))|$ from above result;
              if $dev > \beta$ then break through the recurrence;
              else add the row or column into current seed;
          end
      end
  end
  output the resulting submatrices;
```

Figure 6: Pseudo-code of growing approach

Growing

In this step, the seeds grow through adding certain rows and columns based on similarity. For a given row or column, we are only interested in its elements corresponding to the seed, so we will just call these elements as rows or columns briefly. The sketch of the growing process is as follows: (1) We identify the rows and columns which fluctuate like the seed to some extent. Specifically, we calculate the ratio of variance of the row/column and MSR of the seed; then choose a cut-off α to filter. Only those rows/columns whose ratio is less than α is identified; (2) Within the filtered rows and columns, we choose the one whose mean value is least deviated from the mean value of the seed and add it to the seed; (3) For the new seed, we repeat the above process until all the remaining rows or columns' deviations are beyond another cut-off β. Here the cut-off α and β are parameters used to evaluate the similarity of the newly concerned row/column and the seed. The detailed algorithms of D2D are shown in Figure 3, Figure 5 and Figure 6.

Parameter selections

In the reordering procedure, selection of parameter δ depends on the noise level within data matrix. Due to our computational simulations (data not shown), we recommend to choose δ from (0, 0.3) after normalizing data matrix. There is a balance for the selection of size of template. If the size is too large, we may lose many good results; otherwise if the size is too small, it is difficult to identify meaningful results. The proper size of template is about 5% of the dimensions of data matrix. p_{ml}, p_{mh} and p_{vl} work just as the tail probabilities in statistics; we choose them as 0.05, 0.95 and 0.1 respectively. α and β describe the tolerance of fluctuation within the biclusters. In our experiments, we take α as 10^2 or 10^3 and β as 0.25.

Testing

The datasets used in this study include public data (Prelic *et al* 2006 **[10]**) and the data synthesized by ourselves. We collected the data so that they can represent a wide variety of different situations. We compared D2D with some existing representative algorithms following the way stated by Prelic *et al* (2006) **[10]**. The compared algorithms are CC **[5]**, OPSM **[6]**, xMotif **[9]**, ISA **[7, 8]**, Bimax **[10]** and RMSBE **[11]**. The program BicAT **[12]** was used to implement all these methods except RMSBE, which we used the codes downloaded from the author's website.

Evaluation

In order to assess performances of different biclustering methods, the following gene matching score was proposed by Prelic *et al* (2006) **[10]** as given in equation (5) (see supplementary material). The above score only evaluates from the genes dimension, so there would be some deviation from the aim of biclustering. An alternative criterion is given in equation (6) (see supplementary material).This criterion is more intuitive than the previous one. The numerator is the area of the overlapping part of two biclusters and the denominator is the total area of two biclusters. In order to compare D2D with the reference methods in (Prelic *et al* 2006 **[10]**), we use the first criterion on the Prelic data, and the latter one on our new synthetic datasets.

Datasets

The data of Prelic *et al*. include constant-type and additive-type biclusters, with biclusters of different degrees of overlap, and of different levels of noise. Their design mainly focused on the biclusters with up- or down-regulated expression values. As in a real biological process, a set of genes participating in a pathway may exhibit similar expression values which are close to the average values. So the up- or down-regulated model may not be appropriate for this scenario. We also designed some datasets to study the performances of different algorithms under this situation.

Our datasets design also considered two situations, constant and additive biclusters. For the constant situation, the matrix with implanted constant bi-clusters is generated in four steps: (1) generate a 200*200 matrix A as background through sampling from uniform distribution (0, 1); (2) generate 10 biclusters of size 20*20 whose mean values are ranged from 0.2 to 2; (3) add noise to these biclusters through sampling from uniform distribution (-σ, σ); (4) implant the 10 biclusters into A without overlapping. For the additive situation, the procedure is similar. But here we let the biclusters have an increasing trend on the rows and columns with the difference between two adjacent rows or columns being 0.05.

Synthetic datasets I

Following the process proposed by Prelic *et al* (2006) **[10]**, we compared the performance of D2D and RMSBE with the reference methods. Through systematic experiments, we chose the best set of parameters of the RMSBE for the comparison. The experimental results of other methods are taken from Prelic *et al* (2006).

Figure 7 summarizes the performances of the considered methods with respect to constant biclusters. The datasets used in Figure7a are designed for assessing the sensitivity to noise whereas the datasets used in Figure 7b are for assessing the accommodation to degree of overlap.

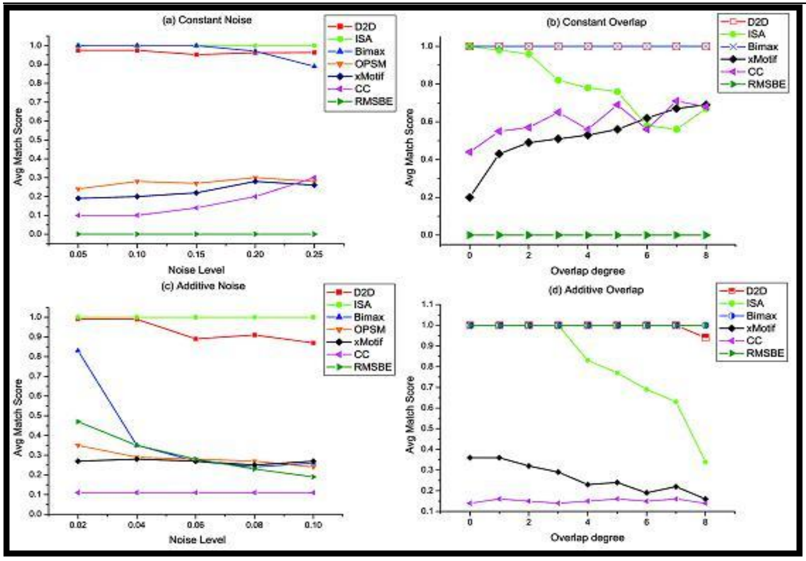


Figure 7: Results of experiments on synthetic datasets I (a) for the constant biclusters with increasing noise level; (b) for constant biclusters with increasing degree of overlap and without noise; (c) for additive biclusters with increasing noise level; (d) for additive biclusters with increasing degree of overlap and without noise

As shown in Figure 7, ISA, Bimax and D2D outperform the other algorithms in most cases. Bimax could identify more than 90% of all the implanted biclusters except for additive biclusters implanted in high noisy background. ISA's result is affected by degree of overlapping heavily. RMSBE could identify all the hidden biclusters as shown in Figure 7d but can hardly find targets in other situations. This is because the similarity score of RMSBE is not very appropriate for the situation where the noise levels within target biclusters and within background are similar. The other methods all behave differently and are affected by noise and degree of overlapping to some extent due to their different models or strategies. Through the comparison results, we can see that the D2D could identify most implanted biclusters and is insensitive to noise and degree of overlapping.

Synthetic datasets II

As in the real situation, the distribution of background is usually broad, so the distinct up- or down-regulation patterns may not be very representative. We designed some datasets with some biclusters whose expression values are similar to the background but the biclusters are more consistent than the background. According to the multiple-seeds growing procedure used by D2D, it is insensitive to degree of overlapping of implanted biclusters as already shown on Synthetic Datasets I. Therefore we only focused on the influence of noise level in the experiments on Synthetic Datasets II.

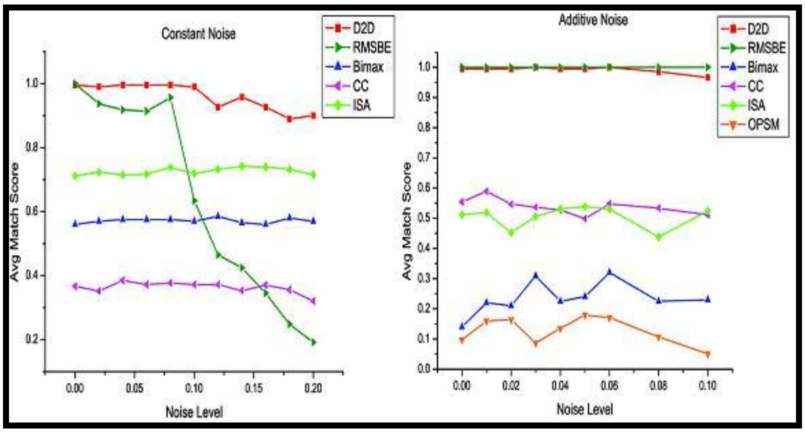


Figure 8: Results of experiments on synthetic datasets II (a) constant biclusters with increasing noise level; (b) additive biclusters with increasing noise level

As OPSM aims to find order-preserving sub-matrices which may not be appropriate for constant biclusters, we did not test its performance on constant scenario. These experiments are shown in Figure 8. The performance of RMSBE on Datasets II is much better than on Datasets I. This is because the implanted biclusters are more consistent than the background and this situation is more appropriate for its assumption. In the constant bicluster scenario, its results deteriorate with increasing noise level. CC's performance also declines due to the same reason. The results of ISA and Bimax are almost the same, around 0.5; this is because they mainly identify the up-regulated biclusters. When it comes to additive datasets, RMSBE and D2D work better in this scenario which demonstrates that the strategy of reference gene and reference condition works well. Through the experiments under this scenario, we can see that the performance of D2D is always among the top methods.

Discussion:

The present study proposes a novel strategy named D2D for tackling the biclustering problem. This novel strategy divides the process of biclustering into a two-step process. At first, a reordering of the data matrix is implemented by defining a new distance, local distance. Then, we use the reordered matrix to find the final biclusters through a scanning-and-growing process. By comparing with several other existing representative methods, we demonstrate the advantage of D2D in several respects, such as flexibility of target pattern definition, insensitivity to the noise and degree of overlapping, and interpretable selection of parameters.

Different methods focus on different features of the data matrix. For example, Bimax and ISA mainly focus on the up- or down-regulation within the data matrix, so its performance on the dataset in Prelic *et al.* **[10]** is good, but it cannot recover the biclusters which are neither up nor down compared to the background. Other methods such as CC, xMotif and RMSBE perform well on finding the consistent biclusters, but are substantially poor on the representative up- or down-regulation datasets.

D2D works well on all the above datasets. The results demonstrate its accommodation for several factors. Besides the novel distance definition, the two-stage procedure adds more flexibility to the biclustering process. One can choose different parameters combinations to get different result with flexible features. In addition, D2D can be used as a pre-processing step to other reference methods. Because most other methods have greedy nature, adopting the reordering procedure before biclustering may help them to get better results.

Conclusions:

This study proposes a novel method D2D for finding distinct biclusters from gene expression data. Comparing with several representative existing methods, D2D accommodates different kinds of bicluster patterns by using a novel local distance and adopting more flexible algorithmic structure. The experiments on synthetic datasets show that D2D is robust to noise levels, overlapping degrees between clusters and different settings of parameters. Further experiments on real microarray datasets and the biological analysis of the results are being undertaken and will be available soon.

Author's contribution:

ZW carried out the original algorithm design, especially the reordering procedure, and participated in the experiments; JA devised the scanning-and-growing procedure and implemented all the experiments. ZW and JA drafted the manuscript. XZ initiated the project, proposed the basic motivation of this study and guided the experiments and manuscript preparation. All authors have read and approved the final draft.

Acknowledgement:

This work is partially supported by NSFC grants 60575014 and 30625012 and the National 863 project 2006AA02Z325.

References:

[01] J. A. Hartigan, *J.A.S.A.*, 67: 123 (1972)
[02] M. Schena, *et al.*, *Science*, 270: 467 (1995) [PMID: 7569999]
[03] M. B. Eisen, *et al.*, *Proc. Natl. Acad. Sci.*, 95: 14863 (1998) [PMID: 9843981]
[04] S. C. Madeira, *et al.*, *IEEE/ACM Trans Comput. Biol. Bioinform.*, 1: 24 (2004) [PMID: 17048406]
[05] Y. Z. Cheng, *et al.*, *Pro. Int. Conf. Intell. Syst. Mol. Biol.*, 8: 93 (2000) [PMID: 10977070]
[06] A. Ben-Dor, *et al.*, *J. Comput. Biol.*, 10: 373 (2003) [PMID: 12935334]
[07] J. Ihmels, *et al.*, *Nat. Genet.*, 31: 370 (2003) [PMID: 12134151]
[08] J. Ihmels, *et al.*, *Bioinformatics*, 20: 1993 (2004) [PMID: 15044247]
[09] T. M. Murali, *et al.*, *Pac. Symp. Biocomput.*, 77 (2003) [PMID: 12603019]
[10] A. Prelic, *et al.*, *Bioinformatics*, 22: 1122 (2006) [PMID: 16500941]
[11] X. W. Liu, *et al.*, *Bioinformatics*, 23: 50 (2007) [PMID: 17090578]
[12] S. Barkow, *et al.*, *Bioinformatics*, 22: 1282 (2006) [PMID: 16551664]
[13] A. Tanay, *et al.*, *Bioinformatics*, 18: 136S (2002) [PMID: 12169541]
[14] C. Li, *et al.*, *Genome Biol.*, 2: 32 (2001) [PMID: 11532216]

Edited by A. M. Khan, T. W. Tan & S. Ranganathan
Citation: Wu *et al.*, Bioinformation 2(5): 207-215 (2007)

Supplementary material

$$DR_{pq} = \frac{1}{NR_{pq}} = \frac{1}{\#\{|R_{pq,i}| < \delta\}} \qquad (1)$$

$$DC_{pq} = \frac{1}{NC_{pq}} = \frac{1}{\#\{|C_{pq,i}| < \delta\}} \qquad (2)$$

In above formula, C_{pq}=[$C_{pq,1}$,…,$C_{pq,n}$]=[($e_{1,p}$- $e_{1,q}$),…,($e_{n,p}$- $e_{n,q}$)] is the difference vector between column vectors [$e_{1,p}$,…,$e_{n,p}$]^T and [$e_{1,q}$,…,$e_{n,q}$]^T.

$$r(e_{ij}) = e_{ij} - e_{iJ} - e_{Ij} + e_{IJ} \qquad (3)$$

$$H(I,J) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} r(e_{ij})^2 \qquad (4)$$

Where e_{ij} is an element of bicluster, e_{iJ} is the mean value of row *i* of bicluster, similarly e_{Ij} is the mean value of column *j* of bicluster, and e_{IJ} is mean of whole bicluster. $r(e_{ij})$ is the residue of e_{ij} and $H(I,J)$ is the mean of the squared residues of the whole bicluster.

$$S(G_1,G_2) = \frac{1}{|M_1|} \sum_{(G_1,C_1)\in M_1} \max_{(G_2,C_2)\in M_2} \frac{|G_1 \cap G_2|}{|G_1 \cup G_2|} \qquad (5)$$

Here M_1 and M_2 are two sets of biclusters. M_1 is the set of implanted biclusters designed in the simulation (the ground truth) and M_2 is the resulting set of an algorithm (the prediction). The value of $S(G_1,G_2)$ reflects to what extent the predicted biclusters recover the ground truth.

$$S = \frac{1}{|M_1|} \sum_{(G_1,C_1)\in M_1} \max_{(G_2,C_2)\in M_2} \frac{|G_1 \cap G_2||C_1 \cap C_2|}{|G_1||C_1| + |G_2||C_2| - |G_1 \cap G_2||C_1 \cap C_2|} \qquad (6)$$