International Journal of Bioinformatics Research
ISSN: 0975-3087 & E-ISSN: 0975-9115, Volume 5, Issue 1, 2013, pp.-289-293.
Available online at http://www.bioinfopublication.org/jouarchive.php?opt=&jouid=BIJ0000001

MINING SIGNIFICANT GENE BICLUSTERS FROM DNA MICROARRAY

RAUT S.A.* AND SATHE S.R.

Department of Computer Science & Engineering, Visvesvaraya National Institute of Technology, Nagpur- 440 001, MS, India.
*Corresponding Author: Email- rautsa@gmail.com

Received: November 26, 2013; Accepted: December 26, 2013

Abstract- Significant gene biclusters are important in medical science in many ways. They can be used in drug discovery, identification of severe diseases, finding the gene pathways and many more. We are using two algorithms to find the final significant biclusters from the DNA microarrays. In first algorithm, transform discrete values are used while in second algorithm, actual numerical values are used as an input. The results are tested on both synthetic as well as on real database. The output for the real database i.e. Yeast Cell Cycle is discussed at the end.

Keywords- Biclusters, DNA Microarrays, Gene Expression Matrix

Citation: Raut S.A. and Sathe S.R. (2013) Mining Significant Gene Biclusters from DNA Microarray. International Journal of Bioinformatics Research, ISSN: 0975-3087 & E-ISSN: 0975-9115, Volume 5, Issue 1, pp.-289-293.

Introduction

In all the living organisms, gene is one of the most important part of the cells. These genes after undergoing certain processes produce proteins. Each protein within the body has a specific role. Some proteins are involved in structural support, while others are involved in bodily movement, or some are busy in defense activity against germs. These proteins are responsible for every biological activity of living organisms. To study the proteins is one of the most complex process. Hence, Bioinformaticians do the research on genes which are origin of these proteins. Scientists produced many important conclusions after a deep study of these genes which ultimately helps to understand the complex processes of proteins. Gene expression is the process by which information from a gene is used in the synthesis of a functional gene product that is proteins. Other than understanding of proteins, analysis of gene expression contributes to many important bioinformatics activities, they are shortly listed as follows:

- Finding genes, locating coding regions, predicting function: automate
- Sequence, structure, function, evolution (FESS relationships)
- Metabolic genotype, phenotype, redundancy
- Genes to Pathways; Genes to biological knowledge
- Assigning gene sets to different species: homologs vs paralogs
- Finding conserved proteins common to all life
- Expression profiles, relation to metabolic pathways / genetic networks
- Gene synteny between species: gene adjacency in genomes.

The study of gene expression starts with gene expression matrix. These matrices are the big matrices, where number of rows are genes and number of columns are conditions. As per biological observations, any biological activity is only present with group of genes and group of conditions. Hence to catch only involved genes and conditions (which may contribute lot of biological significance), algorithm must select the bicluster having only involved genes and conditions responsible for specific biological activity. In this paper, we try to catch these significant biclusters using two algorithms.

The paper further talks in detail about above subject. It is divided into Background, Literature survey, Algorithm, Experimental studies, and Discussion on generated outcome.

Background

Every cell in living organism, contain instructions for every structure (protein) and processes in our body. The instructions are present in a material called DNA. The transformation of DNA to RNA (transcription) and RNA to protein (translation) is called the central dogma of molecular biology. DNA is the heredity material of the cell. A gene is a small segment of DNA, found in a small section of the chromosome. Transcription and translation are highly regulated processes, with constantly changing environments. The 30,000 genes of the human genome can express hundreds of thousands of proteins, each with a specific role to play. Transformation of genes to proteins is considered to be one of the most complicated biological processes. It is even more complicated than genomics because an organism's genome is more or less constant, whereas the proteome differs from cell to cell and from time to time.

Distinct genes are expressed in different cell types, which means that even the basic set of proteins that are produced in a cell needed to be identified. Recent advances in bioinformatics brings a revolution in understanding of the molecular mechanisms underlying normal and dysfunctional biological processes. Proteins are directly or indirectly responsible for all these biological process. It is really difficult to study the most complicated(proteins) form of genes than to study directly the genes. Hence, we try to find many important answers by studying or analysis of originate of proteins i.e. genes. The analysis of genes and their expression will help to understand the complex processes, under which each gene in the DNA sequence is “expressed”, i.e. when, where, and to what extent the gene is stimulated to produce the protein (encoding).

DNA Microarray

DNA microarray is a high throughput technology used in molecular biology and in medicines. It is very powerful technology and can measure the performance of thousands of genes simultaneously. DNA microarrays works as per principle of Watson-Crick base pairing rule. Two types of experiment can be done using DNA Microarray, one is time course experiment and second is comparative experiment. In time course experiment, we note the changes found in terms of expressions by the genes after each time course as experimental conditions. For example, we want to see the effect of injection of some drug to certain cell. Then rows of datasets will be number of genes and column will be timestamp of some interval. Each element in the dataset is an expression of specific gene at respective time, we consider it as effect of drug at various time span on genes. Second experiment is of comparative nature i.e. infected cell vs. normal cell. We can note down the expressions of normal cell and then compare it with expression of infected cell. This information is very helpful to find which genes are affected, we can have detailed study like how to treat these genes, what is effect of different drugs on them and many more.

For generating final datasets of gene expression, microarrays follows three steps:

Sample Preparation and Labeling

It involves, extraction of mRNA from tissue of interest, conversion from mRNA to cDNA and labeling of this cDNA. Labeling is important as it founds detection of which cDNA are bound to which microarray.

Sample Hybridization and Washing

In hybridization, DNA probes on the microarrays and labeled DNA target, forms heteroduplexes according to Watson-Crick base pairing rule. After hybridization, microarray chip is washed to eliminate any excess labeled sample other than DNA complementary probes.

Image Scanning and its Processing

A hybridized array is scanned to produce a microarray image. Labeled samples with dyes emit detectable light when simulated by a laser. Detectable emitted light by target DNA strands are bound to their complementary probes. This scanned output is a monochrome image.

The scanned image can be considered as an input for microarray information analysis. The analysis further categorized as low level analysis and high level analysis. In low level analysis, spot quantization matrices are generated. Knowledge extraction, is done in high level analysis. The set of spot quantization matrices are summarized to form single gene expression data matrix (dataset). The above complete process is shown in pictorial form [Fig-1].

Gene Expression Matrix

The results of microarray experiment are often produced in terms of matrix, called as Gene Expression Matrix, in which rows represents genes and columns represents various time points or different environmental conditions. An element of a matrix represents the logarithm of the relative abundance of mRNA of a gene under specific condition.

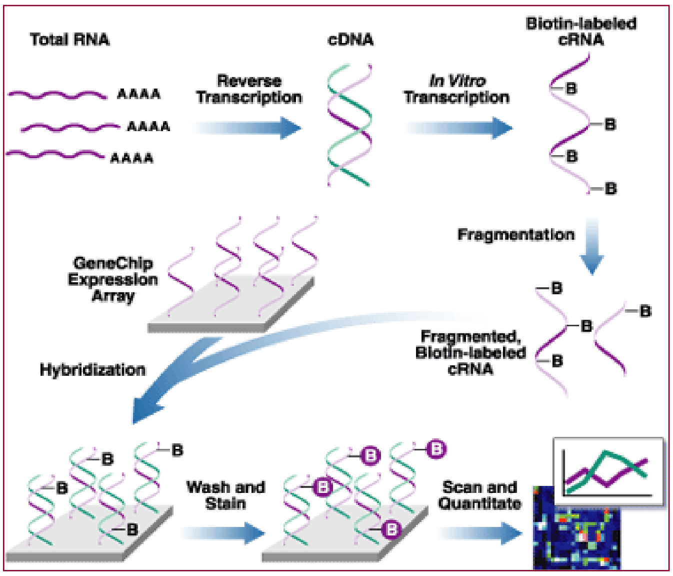

Fig. 1- Generation of Gene Expression Matrix from DNA Microarray

Gene expression matrices have been extensively analyzed by two dimensions: the gene dimension and condition dimension. These analysis correspond, respectively, to analyze the expression patterns of samples by comparing the rows in the matrix, and to analyze the expression patterns of samples by comparing the columns in the matrix. For generating results we worked on gene dimensions, where rows are genes and columns are the environmental conditions. If *A* is a gene expression matrix with *M* rows and *N* columns, then the goal is to find a sub-matrix *A’* with *M’* rows and *N’* columns form *A*.

A(M,N) A’(M’,N’) where M’ are some selected rows in M and N’ are some selected columns in N. These M’ and N’ are the genes and conditions resp. selected together which shows some relation or pattern in between them. The pattern which we are looking are explained in [Fig-2]. All such genes are co-expressed genes under certain same conditions. We try to find such sub-matrices which will be responsible for some important clue for molecular biology.

Problem Definition

To get sub-matrix from original gene matrix, so that significant biological information can be gained is the only objective of our procedure. The goals can be further listed as follows:

- Find out the groups of all co-expressed genes.
- Observe the behavior of all group of genes as per conditions.
- List out the significant observations

To Contribute to the Mining of Genes

We consider a data matrix *A* called as gene data matrix having X rows and Y columns, Here, X = $\{x_1,x_2,\ldots x_n\}$, and Y = $\{y_1,y_2\ldots y_n\}$. The submatrix A_{IJ} = (I, J) is, a subset of rows I=$\{i_1,i_2,\ldots i_k\}$ (I ⊆ X and k ≤ n), and a subset of columns J = {j1, j2,..js} (J ⊆ Y and s ≤ m). A (I, J) can be defined as a k by s submatrix from the matrix A. Our aim is to find such A (I, J) meaningful submatrices which contributes to the biological knowledge.

The objective of an algorithm is to extract coherent and maximum size biclusters, i.e., a maximum group of genes with a maximum groups of conditions where the genes exhibit highly correlated activities over a range of conditions.

Literature Survey

Traditional technique used for finding out co-expressed genes from gene expression matrix was clustering. The detailed discussion of clustering can be found in Raut, et al [1,2]. There are some drawbacks in the application of clustering on gene expression matrix as follows. Clustering methods can be applied to either the rows or the columns of the data matrix, separately and hence can derive a global model which finds either group of genes with all conditions or group of conditions with all genes. But, the fact according to biological findings is most of the patterns [Fig-2] are common to the group of genes only under specific experimental conditions and remain independent for the rest of conditions. These local patterns need to be identified as they help to solve many biological problems. Clustering cannot find these local patterns and hence needed biclustering so that some group of genes with some group of conditions, which are highly co-related, can be estimated from the given gene matrix. Vast literature is available on usage of biclustering for extraction of information on gene expression matrix. According to [3], all biclustering methods/ algorithms are designed to find certain patterns from gene expression matrix, they can be specified as:

1. Biclusters with constant values [Fig-2](a).
2. Biclusters with constant values on rows or columns [Fig-2](b) & (c).
3. Biclusters with coherent values (Additive) [Fig-2](d).
4. Biclusters with coherent values (Multiplicative) [Fig-2](e).
5. Biclusters with coherent evolution [Fig-2](f).

5	6	7	8
1	2	3	4
2	3	4	5
6	7	8	9

(a)

0	1	1	1
1	2	1	2
1	2	1	2
1	3	2	4

(b)

1	3	4	1
3	5	7	0
5	7	10	1
6	7	8	4

(c)

5	5	5	5
5	5	5	5
5	5	5	5
5	5	5	5

(d)

5	5	5	5
6	6	6	6
7	7	7	7
8	8	8	8

(e)

5	6	7	8
5	6	7	8
5	6	7	8
5	6	7	8

(f)

Fig. 2- Types or pattern of Gene Expression

Earlier research papers develop algorithms which are able to find one or more patterns out of five, but not all. Some considering [Fig-2](a-e) are the patterns of one type and [Fig-2](f). is of another type. Hence, the algorithms develop either considers all patterns i.e. from [Fig-2](a-e) or an algorithm which considers only pattern [Fig-2](f). Hartigan [4], Tibshirani, et al [5], Getz, et al [6], Segal, et al [7,8], Sheng, et al [9], Lazzeroni and Owen [10], Kluger, et al [11], Tang et al [12], Ayadi, et al [13], Xiangchao Gan et al [14] and the most important Cheng and Church [15] discussed their algorithms to find pattern of [Fig-2] and many more gave excellent contribution for discovering patterns for gene expression matrix.

Our Contribution and Algorithmic Strategy

We tried to find significant clusters using two algorithms. In first algorithm, as the size of matrix is large, we transform the values of matrix as transform values and examine it as coherent values. The output of the first algorithm is group of some genes with some conditions. In second algorithm, we take the input as all those groups formed by first algorithm and find exact biclusters from the those groups. The detailed method is as follows:

Algorithm-I

The input for the first algorithm is gene expression matrix *A* having number rows *I* and number of columns *J*.

aij is the expression of gene *I* for condition *J*. Synthetic data is noise prune, but, real time data always has some noise, hence instead of considering matrix as a real numbers, we transform the input matrix of real numbers into discrete matrix having only three values 1, 0, -1. The rules to convert given matrix values are as follows:

$$B\,(I',J') = \begin{cases} -1, \text{if } A\,(I,J) - A\,(I,J+1) > 1; \\ 0, \text{if } A\,(I,J) - A\,(I,J+1) = 0; \\ 1, \text{if } A\,(I,J) - A\,(I,J+1) < 1; \end{cases}$$

where, A (I,J) is the input matrix and B (I',J') is the transformed matrix for ex. [Fig-3](a) shows original matrix while [Fig-3](b) shows its transform equivalent matrix.

There is a necessity of calculation of S for each generated biclusters, as each bicluster may not be substantial bicluster. We calculate *S* as per Ayadi, et al [13], and find its validity. In our experiment, for synthetic as well as for real dataset we fix the value of acceptable *S* is 0.8. So, if the generated bicluster is having calculated score equal or more than 0.8, that bicluster considered to be valid for further processing. The output of the first algorithm is collection of all such valid biclusters.

1.2	2.7	3.9	0.6
3.1	5.2	7.1	0.2
5.4	6.8	9.7	1.3
6.3	7.3	8.2	4.3

(a)

1	1	-1
1	1	-1
1	1	-1
1	1	-1

(b)

Fig. 3 (a)- Sample Data Matrix; **(b)-** Processing on Sample Data Matrix

Algorithm II

All the biclusters which are with valid *S* as Spearman's ratio will be selected, but these are raw biclusters, to select more significant biclusters, we are using actual values of C i.e. original expression matrix values as input for second algorithm. In the second algorithm, we take all valid biclusters having original (not transformed i.e. not 0, 1 & -1) values and find significant group of genes with group of conditions as an output. The algorithmic steps are as follows:

Input: C, which are intermediate set of Biclusters.

Output: Significant Biclusters S.

```
for each C1 in C do
        for each j in J of C1 do
subtract/divide rest of columns in J of C1 to get IM(X,Y).
        end for
find constant/additive/multiplicative subclusters
for IM to form final S1.
S= U S1.
end for
```

Three separate functions are written to identify constant(includes [Fig-2](a) & (b), additive [Fig-2](c) & (d) and multiplicative [Fig-2](e) pattern exists for input biclusters.

5	5	5	5
5	5	5	5
5	5	5	5
5	5	5	5

5	5	5	5
6	6	6	6
7	7	7	7
8	8	8	8

(a)

0	0	0
0	0	0
0	0	0
0	0	0

0	0	0
0	0	0
0	0	0
0	0	0

(b)

Fig. 4 (a)- Sample Data Matrix; **(b)-** Processing on Sample Data Matrix

[Fig-4](a) & (b) are the examples of constant biclusters, so for constant biclusters, if we subtract column from other rest of columns we always get constant values as a output.

5	6	7	8
5	6	7	8
5	6	7	8
5	6	7	8

5	6	7	8
1	2	3	4
2	3	4	5
6	7	8	9

(a)

-1	-2	-3
-1	-2	-3
-1	-2	-3
-1	-2	-3

-1	-2	-3
-1	-2	-3
-1	-2	-3
-1	-2	-3

(b)

Fig. 5(a)- Sample Data Matrix; **(b)-** Processing on Sample Data Matrix

[Fig-5](a) are the input biclusters, [Fig-5](b) is the output of Algorithm II. As group of genes have same value for group of conditions, these can be formed to an cluster.

0.3	0.9	0.6	1.2
0.5	1.5	1	2
0.6	1.8	1.2	2.4
0.9	2.7	1.8	3.6

(a)

3	2	4
3	2	4
3	2	4
3	2	4

(b)

Fig. 6(a)- Sample Data Matrix; **(b)-** Processing on Sample Data Matrix

In [Fig-6](a), pattern is multiplicative then by dividing column from other rest of columns we get group of genes with similar group of conditions and hence can be group as a cluster[Fig-6](b).

Experimental Study

We applied our designed algorithms on both real and simulated data.

Simulated Data

We have used two simulated data, first of dimension 420 X 70 and second of dimension 50 X 20. Both datasets are randomly generated and embedded with constant, additive as well as multiplicative non-overlapping biclusters. As the algorithm I working, is based on greedy strategy, all the embedded rows for the specified biclusters are traced correctly but in groups, i.e. suppose a bicluster is having number of rows as 2,4,5,8,9,10,13,14,15 and number of columns as 3,4,5,8,12,15,17,19 then instead of one cluster i.e. all rows to be in one biclusters we get two biclusters with number of columns. Then these all biclusters are submitted to algorithm II, to get exact rows and columns. By the end of algorithm II, we get biclusters (may be single bicluster divided into number of groups with group of columns). Then, finally by the close observation of all the biclusters we may regroup splitted biclusters.

We are able to extract all the biclusters that are embedded with both the simulated datasets.

Real Dataset

We have used Yeast Cell Cycle dataset as a real dataset [15]. It contains 2884 genes and 17 conditions. The Yeast Cell Cycle is 6000 X 17 dataset but we have considered publicly available [15] i.e. 2884 X 17 as a dataset for generating the output of our algorithm. To find the biological relevance, we use web tool, FuncAssociate 2.0 [16], complete GO:0000001 to GO:2000911 are annotated for Yeast Cell Cycle. The file is quite big, hence we show the output with respect to GO:0000002 (attached as File-I). We apply our both algorithms on above Yeast Cell Cycle, the output of algorithm is attached with the paper(as supplementary File-III attached). We further explain our outcomes with reference to GO:0000002. The observations are mentioned after analyzing output of second algorithm (as supplementary File-IV attached). In the dataset in place of actual gene name we have numbers from 1,2,3,….,2884. The mapping of actual genes to numbers is in a file as supplementary File-II.

Discussion

For real dataset we have choosen Yeast Cell Cycle dataset as it is publicly available. The first algorithm is applied on the dataset of size 2884 X 17, this dataset is transformed into three discrete values like -1,0,1. The algorithm is of greedy nature and hence biclusters are generated as split up form. The meaning is if a bicluster is having 30 genes and 10 conditions, the first algorithm output shows 3-4 biclusters, having 30 genes pervade on these biclusters. But all the genes are correctly selected in all those 3-4 biclusters. Now as the input is transformed may be all relevant genes as per transformed values can group to form the biclusters.

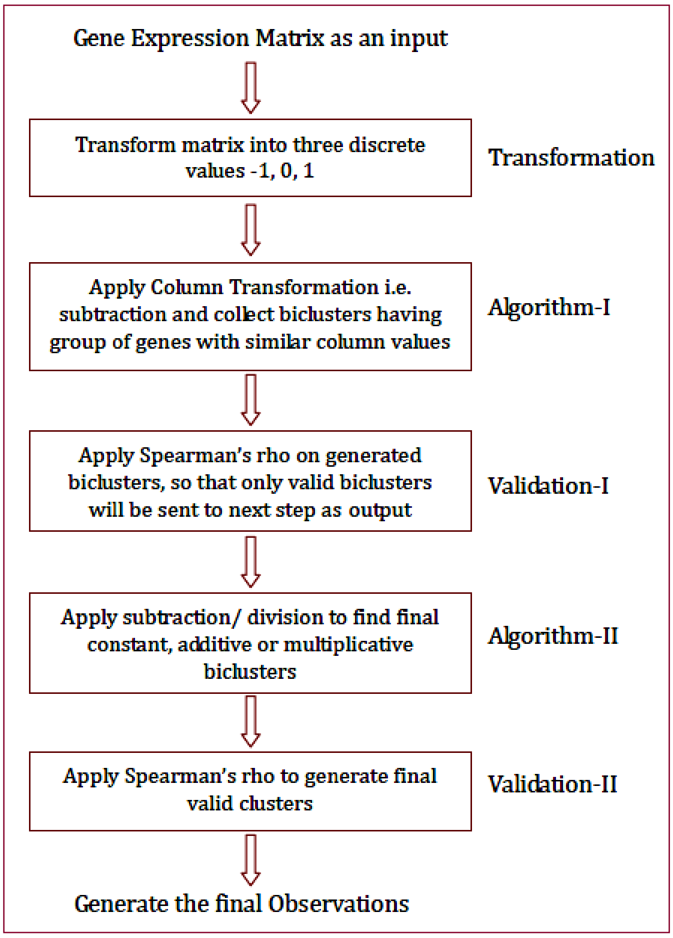
Gene Expression Matrix as an input
Transform matrix into three discrete values -1, 0, 1
Transformation
Apply Column Transformation i.e. subtraction and collect biclusters having group of genes with similar column values
Algorithm-I
Apply Spearman's rho on generated biclusters, so that only valid biclusters will be sent to next step as output
Validation-I
Apply subtraction/ division to find final constant, additive or multiplicative biclusters
Algorithm-II
Apply Spearman's rho to generate final valid clusters
Validation-II
Generate the final Observations

Fig. 7- Pictorial representation of an Algorithm

We find Spearman's rho value for each bicluster, to select only the valid biclusters and delete all irrelevant biclusters. Now, we send all the valid and tested biclusters as an input to second algorithm. The second algorithm take the intermediate biclusters generated by first algorithm and generates final biclusters. Hear also we use Spearman's rho to find only relevant biclusters. The flow of the complete procedure can be shown as pictorial diagram in [Fig-7]. The detailed discussion of result are explained in supplementary files, File-I, File-II, File-III, and File-IV.

The problem with applied algorithm is its greedy nature, and thus in place of receiving single bicluster for meaningful output, we receive 2-4 biclusters and later by post processing we have to combine all 2 -4 biclusters to be one. Hence, in future we like to improve the algorithm so that only one meaningful bicluster will be generated for single meaningful output.

Acknowledgement

We are thankful to Cheng & Church [15], publically available dataset on Yeast Cell Cycle. We generate our observations based on this datasets. We are thankful for all who contribute for the same topic and because of them large literature is available for the topic. Last but not least we are thankful to our department for all the needful assistance.

Conflicts of Interest: None declared.

References

[1] Raut S.A., Sathe S.R., Raut A. (2010) *IEEE International Conference on Bioinformatics and Biomedical Technology* 97-100.

[2] Raut S.A., Sathe S.R., Raut A. (2010) *Journal of Computer Science and Engineering*, 4(1), 11-17.

[3] Madeira S.C., Teixeira M.C., Sá-Correia I., Oliveira A.L. (2010) *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 7(1), 153-165.

[4] Hartigan J.A. (1972) *Journal of the American Statistical Association*, 67(337), 123-129.

[5] Tibshirani R., Hastie T., Eisen M., Ross D., Botstein D. and Brown P. (1999) *Clustering methods for the analysis of DNA microarray data*, Technical report, Department of Health Research and Policy, Department of Genetics and Department of Biochemistry, Stanford University.

[6] Getz G., Levine E., Domany E. (2000) *Proceedings of the National Academy of Sciences*, 97(22), 12079-12084.

[7] Segal E., Taskar B., Gasch A., Friedman N., Koller D. (2001) *Bioinformatics*, 17(1), S243-S252.

[8] Segal E., Battle A., Koller D. (2003) *Proceedings of the Pacific Symposium on Biocomputing*, 8, 89-100.

[9] Sheng Q., Moreau Y., De Moor B. (2003) *Bioinformatics*, 19(2), ii196-ii205.

[10] Lazzeroni L., Owen A. (2002) *Statistica Sinica*, 12(1), 61-86.

[11] Kluger Y., Basri R., Chang J.T., Gerstein M. (2003) *Genome Research*, 13(4), 703-716.

[12] Tang C., Zhang L., Zhang A., Ramanathan M. (2001) *IEEE 2nd International Symposium on Bioinformatics and Bioengineering Conference*, 41-48.

[13] Ayadi W., Elloumi M., Hao J.K. (2012) *Knowledge-Based Systems*, 35, 224-234.

[14] Gan X., Liew A.W., Yan H. (2008) *BMC Bioinformatics*, 9(1), 209.

[15] Cheng Y., Church G.M. (2000) *Eighth International Conference on Intelligent Systems for Molecular Biology*, 8, 93-103.

[16] Berriz G.F., King O.D., Bryant B., Sander C., Roth F.P. (2003) *Bioinformatics*, 19(18), 2502-2504.