Composite Module Analyst: A Fitness-Based Tool for Prediction of Transcription Regulation.

Alexander Kel[1], Tatiana Konovalova[2], Tagir Valeev[3], Evgeny Cheremushkin[3],
Olga Kel-Margoulis[1], Edgar Wingender[1]

[1] BIOBASE GmbH, Halchtersche Str. 33, D-38304 Wolfenbüttel, Germany; ake@biobase.de
[2] Institue of Cytology and Genetics, 10, Lavrentev ave , 630090 Novosibirsk, Russia
[3]A.P. Ershov's Institute of Informatics Systems, 6, Lavrentiev ave.,
630090 Novosibirsk, Russia

Abstract. Functionally related genes involved in the same molecular-genetic, biochemical, or physiological process are often regulated coordinately Such regulation is provided by precisely organized binding of a multiplicity of special proteins (transcription factors) to their target sites (*cis*-elements) in regulatory regions of genes. *Cis*-element combinations provide a structural basis for the generation of unique patterns of gene expression. Here we present a new approach for defining promoter models based on composition of transcription factor binding sites and their pairs. We utilize a multicomponent fitness function for selection of that promoter model fitting best to the observed gene expression profile. We demonstrate examples of successful application of the fitness function with the help of a genetic algorithm for the analysis of functionally related or co-expressed genes as well as testing on simulated data.

1 Introduction

Massive application of microarray measurements of gene expression is a common route in the current studies of disease mechanisms. Hundreds of genes are revealed whose change of expression is associated with the disease. However, changed expression of these genes often represents just an "echo" of the real molecular processes in the cells. Still, the available means of analysis and interpretation of these mechanisms are very limited.

Regulation of gene expression is accomplished through binding of transcription factors (TFs) to distinct regions of DNA. It is clear by now that combinations of transcription factors rather than single transcription factors drive gene transcription and define its specificity. At the level of DNA, the blueprints for assembling of such variable TF complexes on promoter regions may be seen as specific combinations of TF binding sites located in close proximity to each other. We call such structures "composite regulatory modules".

For revealing class-specific composite promoter models of functionally related or coexpressed genes we developed the Composite Module Analyst (CMA) - an integrated computational tool for causal interpretation of gene expression data. This tool applies a novel approach for defining the promoter model based on composition of transcription factor binding sites and their pairs. We utilize a multicomponent fitness function for selection of the promoter model which fits best to the observed gene expression profile. We demonstrate examples of successful application of the fitness function with the help of genetic algorithm for the analysis of functionally related or coexpressed genes as well as testing of the tool on simulated data.

2 Composite Modules

2.1 Databases and tools for identification of TF binding sites

In this study we used three databases: 1) TRANSFAC® is a database on gene regulation [1]. It collects data on transcription factors and their binding sites in promoters and enhancers of eukaryotic genes as well as library of position weight matrices (PWMs). This work has been done with release 8.4. 2) TRANSCompel® (rel.6.4) which contains known composite regulatory elements in mammalian genes [2]. 3) TRANSPRO™ is a database on promoters. Positions of transcription start sites (TSS) in genomes (human, mouse, rat) are collected from EPD, DBTSS and ENSEMBL, providing the possibility to retrieve about 13000 human and about 10000 murine promoters. Potential binding sites for transcription factors were identified by the TRANSFAC accompanying tool Match™ [3] that uses a library of about 500 positional weigh matrices (PWMs) for vertebrate transcription factors (TRANSFAC release 8.4).

2.2 Definition of composite promoter model.

We model the promoter structure by a Boolean function:

$$PS = \theta(b_1, b_2, \ldots b_m), \qquad (1)$$

where b_i are the (0,1) outputs of d independent composite modules (CMs) cm_i (i = 1, 2, …, m).

Each composite module is a triplet (Φ,M,Ψ), where Φ is the set of transcription factors regulating the promoters; M is a set of positional weight matrices (PWMs) that are used to identify potential binding sites for the transcription factors from the set Φ; and Ψ is a set of rules and parameters that are applied to the set M to search for the potential DNA binding sites in promoters. In the current realization of the method we consider the following rules and parameters Ψ:

1) w_i, the length of the window in promoter sequences that covers all sites included in the CM;
2) K_i, the number of individual PWMs in the module,
3) R_i, the number of pairs of PWMs,
4) cut-off value $q^{(k)}_{cut-off}$, relative impact values $\phi^{(k)}$, maximal number of best matches $\kappa^{(k)}$ are assigned to every individual weight matrix k (k=1,K_i),
5) cut-off value $q^{(r)}_{cut-off}$, relative impact values $\phi^{(r)}$, orientation values $o^{(r)}$ and maximal $d^{(r)}_{\max}$ and minimal $d^{(r)}_{\min}$ distances are assigned to every matrix pair r (r=1,R_i) in the CM.

First of all, the program searches for potential TF sites in the promoters using Match™ algorithm by applying the matrices M and corresponding cut-off values. Next, in each sliding window of the length w, the program selects the predefined maximal number of the best matrix matches and checks if the found sites obey the predefined distance and orientation rules. A maximal normalized composite score value cms_i is calculated among all window positions using the following equation:

$$cms_i = \left[\sum_{k=1,K_i} \phi^{(k)} \times \sum_{j=1}^{\kappa^{(k)}} q_j^{(k)}(x) + \sum_{r=1,R_i} \phi^{(r)} \times (q_1^{(r)}(x) + q_2^{(r)}(x)) \right] \Big/ Max\ (cms) \qquad \mathbf{(2)}$$

where $q_i^{(k)}(X) < q^{(k)}_{cut-off}$ and $q_{1,2}^{(r)}(X) < q^{(r)}_{cut-off}$; and distance between matches of two matrices of a pair (r) : $d^{(r)}_{\min} < d^{(r)} < d^{(r)}_{\max}$; *Max(cms)* – is the maximal possible value of *cms.*

A threshold parameter P_i is defined for each composite module. It is used to compute the corresponding Boolean value:

$$b_i = \begin{cases} 0, cms_i < P_i \\ 1, cms_i \geq P_i \end{cases} \qquad \mathbf{(3)}$$

Finally, the promoter score *PS* is calculated using the Boolean function (1). If, for a given promoter, *PS=1* we consider that this promoter matches the defined promoter model.

2.3 Fitness function.

Based on the assumption that promoters of co-regulated genes should share common composition of transcription factor binding sites we propose an approach, which allows defining a composite promoter model for sets of co-regulated genes. We have constructed a fitness function that provides a means to direct search in a wide space of various parameters and to find the optimal promoter model settings. In this approach we take as input a set of promoters of potentially co-regulated genes (set A) and a background set of promoters (set B) and design a fitness function, which is based on several properties of distributions of the θ function and cms_i values in these two promoter sets. As an alternative input we consider a single set of promoters with assigned expression values.

The components of our fitness function are the following:

- R – regression value;
- T – Student t-test value;
- E – specificity and sensitivity value;
- N – normality index;
- P – penalty on the complexity of the mode.

The fitness function is defined as linear combination of these components with the specified relative weights:

$$Z = (aR + bT + cE + dN + eP)/(a + b + c + d + e) \qquad (4)$$

The weights *a, b, c, d, e* can be modified by user. Let's consider the components in detail.

2.4 Regression value

This component shows how well the CM scores $cms_i(n)$ of the promoters (*n=1,NP*-total number of promoters) fit the expression values $\xi(n)$ assigned to the corresponding genes. The $\xi(n)$ values can be assigned either as 1/0 values to the promoters of two corresponding A and B sets of genes or as any monotonies function of the observed differential expression value (for instance as the log ratio of gene expression provided by Affymetrix software). The linear regression is constructed by fitting the $cms_i(n)$ to the curve $\alpha\xi(n) + \beta$. We compute an average of the regression values R^2 for each CM of the promoter model and normalized it to the maximal possible value. It is taken then as the component R in the equation (4) which shows an average fit of the considered composite modules in the promoters of the genes to the differential expression of these genes.

2.5 Student t-test value calculation using fuzzy logic

This component is equals to the value of two-sided (?) Student t-test with different variances [4]. Here we consider two alternative promoter sets, A and B, described above or, in the case of the alternative input, the promoter set is divided into two groups A and B: with high and low differential expression values (in this case the value P_o dividing these two groups is specified by the user).

In order to score the values obtained with the Boolean classifier θ we applied the fuzzy logic approach. For each promoter we define a fuzzy score λ which is based on the scores $\lambda_i = cms_i$ of each component of the predicate of the function θ. The following calculation rules are applied:

$$\begin{gathered} \lambda = \lambda_1\lambda_2 \text{ for logical AND;} \\ \lambda = 1-(1-\lambda_1)(1-\lambda_2) \text{ for logical OR;} \\ \lambda = (1-\lambda) \text{ for logical NOT.} \end{gathered} \qquad \mathbf{(5)}$$

After that, the t-test value T is calculated:

$$T = \left[\frac{\overline{\lambda_A} - \overline{\lambda_B}}{\sqrt{\dfrac{\sigma_A^2}{n_A} + \dfrac{\sigma_B^2}{n_B}}} \right] \Bigg/ T_{\max} \qquad \mathbf{(6)}$$

Here, $n_i\ \overline{\lambda_i}$, σ_i, are the number of promoters in each group; an average value of the fuzzy score λ in each group of promoters and the standard deviation respectively. Since value of this function is not bounded above, it is cut by some high value T_{max} and normalized. The T component of the fitness function shows the difference between two distributions of composite module scores.

2.6 Specificity and sensitivity value

This component of the fitness function regulates the mean error rate. It equals to:

$$E = c(1 - FN) + (1 - c)(1 - FP) \qquad \textbf{(7)}$$

where, FN is the false-negative rate (i.e. proportion of promoters in the group A (with $\xi = 1$, or $\xi \geq P_0$ in the alternative input), while having $PS = 0$), and FP is the false positive rate (i.e. proportion of promoters in the group B (with $\xi = 0$, or $\xi < P_0$ in the alternative input), while having $PS = 0$). The constant *0≤c≤1* is defined by the user and gives relative impact of true positives versus true negatives. It equals to 0.5 by default.

2.7 Normality index of the distribution

This component shows how close the distribution of the composite scores to the normal distribution. Let, $\overline{\lambda}, \sigma$ are the average fuzzy score and their standard deviation for promoters of the group A; *p* is the fraction of promoters whose fuzzy score belongs to $(-\infty; \overline{\lambda} - \sigma) \cup (\overline{\lambda} + \sigma; +\infty)$. Then the *N* component of the fitness function is defined as

$$N = 1 - |0.32 - p| \qquad \textbf{(8)}$$

The normality index allows downgrading promoter models that are characterized by irregular distribution of the scores among the promoters under study. It prevents from overfitting and guarantees stability of the predictions on the control data.

2.8 Penalty on the complexity of the model

This component prevents unnecessary growth of complexity of the promoter model. In the case when the number of composite modules (M), the number of matrices (K_i) and pairs (R_i) in each composite module are minimal (the minimal number is defined by the user), *P* becomes equal to 1, otherwise it decreases according to the growth of the size of the model.

$$P = 1 - \frac{1}{2M + 1}\left(\sum_{i=1}^{M} \left(\frac{K_i - K_{\min}}{K_{\max} - K_{\min} + 1} + \frac{R_i - R_{\min}}{R_{\max} - R_{\min} + 1} \right) + \frac{M - M_{\min}}{M_{\max} - M_{\min} + 1} \right) \qquad \textbf{(9)}$$

Here, max and min values for the parameters *M, K* and *R* are defined by the user of the program based on the expectations of the optimal complexity of the model. The *P* component of the fitness function is used to avoid adding some unnecessary complex composite modules and adding new matrices or pairs that actually don't have any effect on the other components of the fitness function, or their influence is very small.

2.9 Genetic algorithm

In order to identify a composite promoter model best fitting the given data on gene expression we use the fitness function described above and apply an optimization strategy based on genetic algorithm. The algorithm takes as an input two sets of promoters (the set of promoters of differentially expressed genes – A group; and a set of promoters of genes whose expression does not differ significantly between experiment and control – B group) or set of promoters and relative expression values assigned to the corresponding genes. On the first step, the program Match™ is used to identify all potential TF binding sites in these two sets of promoters. It uses a predefined subset of PWMs from the TRANSFAC® database. On this step, normally we use minFN matrix cut-offs (see [3] for details) that allows to identify most of the sites in the sequences. The genetic algorithm works then further with the binding sites found by Match™.

Initially, the program CMA generates the set of random models according to the predefined limits of the maximum and minimal number of matrices, pairs, distances between sites, matrix cut-offs and other parameters. Each model is characterized by its own set of parameters. They are considered as a population of "organisms" subjected to mutations, recombination, selection, and multiplication. During a number of iterations the program finds the best model by optimizing the set of matrices selected, their cut-offs, the relative impact, maximum number of best matches and some other optional parameters such as site orientation and distance range in pairs. The output of the CMA tool is the best discriminative promoter model with the optimized parameters.

3. CMA application

3.1 Testing on the simulated data

First, we did initial testing of the ability of the program to reveal *combinations of single matrices*. To test this we performed a series of simulation experiments. We generated sets of 1000 random nucleotide sequences of the length 1000bp each, "implanted" a certain number of site combinations in random positions of these sequences based on a predefined composite module and let the program "blindly" reveal the composite module back or at least some parts of it.

First of all, we observed that with increase of the size of the matrix library the accuracy of recognition falls linearly (data not shown), but this can be generally recovered by increasing the number of iterations of the genetic algorithm which achieves a recognition accuracy of 100 %. We also observed that the algorithm is able to recapture not only CMs containing unique matrices but also modules comprising matrices that are similar to matrices for different TFs (e.g. V$GATA3_03 and V$MYOD_01). We found that the optimal parameters of the algorithm are the following. The ration of the population size to the number of iterations should be between 0.1 and 2; the optimal selection pressure is about 40% (so the best 40% of the population goes into the next generation) and the optimal mutation levels is about 30% of population and recombination level about 60%. With these optimal parameters we performed further testing. The results of the testing are shown in the Tables 1-3.

Table 1. Revealing combination of 3 single matrices (V$AP4_01, V$E2F_02, V$GATA1_02) in the window of 200bp. We implanted sites into 30% of the sequences (set A); other sequences in the set are taken as background (set B). Scores of the implanted sites are high (optimum to reduce false positives) or low (optimum to reduce false negatives). Sites were implanted probabilistically, so the probability that a site is implanted in a given sequence (*P*) varies from 0.3 to 0.9. +++ indicates that all implanted matrices were revealed, ++– that only 2 were found, in 100 iterations; population size was 200.

Site scores	Probability of site insertion		
	0.9	0.6	0.3
High	+++	+++	+++
Low	+++	++–	++–

The result of this simulation shows that the CMA program is able to determine implanted matrices correctly in most cases, even if just a few sequences of set A contain all of the sites of the module (e.g. for *P*=0.3, only abut 8 sequences out of 300 contain all 3 sites).

Microarray data usually contain a high level of noise and reveal no clear differentiation between "changed" and "unchanged" genes. The expression values vary considerably. In order to test the ability of our method to deal with such kind of noisy data we generated test data with various ratios of differential expression values (e) and the random variance of the expression value Δe (as a measure of noise). The result of this simulation (see Table 2) shows that the CMA was able to reveal correct CM in cases of Δe < e and Δe ~ e, although the fitness is decreasing in the second case. In case of Δe > e the method was able to reveal correctly only 2 matrices used for the sites implantation.

Table 2. Testing of ability of CMA program to analyze noisy expression data. 30% of the random sequences were considered as «changed genes». Sites were implanted in these sequences only (*P*=0.9). Matrices were the same as in the Table 1. High cut-offs were used for implantation. We assigned to each "changed gene" an expression value randomly generated from the interval [10, 10+x]. Other "nonchanged" genes got the lower expression value from the interval [0,x]. By varying x = 5,10,15 we simulate 3 variants of the data with gradually increased noise.

Expression parameters	Fitness (*Z*)	CM
Δe < e (x=5)	0,6	+++
Δe ~ e (x=10)	0,4	+++
Δe > e (x=15)	0,3	++–

Next, we tested the functionality of the program on revealing *pairs* of matrices that reflect composite elements composed of two closely situated sites. One pair of sites was implanted in to the 30% of random sequences (pair: V$AP4_01 / V$MEF2_01; distance vary: 5-25; cut-offs "high"). After that we tried to reveal this CM back by varying the parameters of the search (see Table 3). When the frequency of the pairs in the sequences is high (*P*=0.9), the program is able to reveal the correct matrix pair under a wide range of search parameters. If frequency is low, it becomes more difficult to predict the correct structure of the CM without enough knowledge on the expected structure which can help to set optimal parameters of the search. Anyway, in the most of the cases the program is able to predict correctly the components of the CM even its fine structure is not correctly predicted.

Table 3. Testing of ability of CMA program to reveal implanted pairs of matrices. (++) indicates that the correct pair was found, ++ that two single matrices were revealed and they are correct components of the pair.

Parameters of the search of	Probability of site insertion		
the CM structure	0,9	0,6	0,3
1 pair	(++)	(++)	(++)
1-3 pair	(++)	(++)	(++)
0-1 pair, 0-2 single matrices	(++)	++	++

3.2 Analysis of promoters of co-regulated genes with the help of CMA

In order to check the ability of CMA to reveal known composite promoter modules we analyzed a set of promoters of T-cell specific genes that have been shown to be regulated by a very specific type of composite elements: NF-AT/AP-1. The set includes genes for several interleukins 1,4,5,8; their receptors, signaling molecules such as TNF-alpha, IFN and others. In our earlier paper [5] we showed that promoters of these genes contain many copies of the NF-AT/AP-1 composite elements. Here, we would like to check if CMA would be able to reveal these composite elements without having explicit knowledge about their composition. Results of this test are shown in the Table 4 and Fig.1.

Table 4. Example of CMA output of the composite promoter model revealed in T-cell specific promoters by optimization the fitness function with the help of genetic algorithm. Set A contains 26 promoters of an average length 1000bp; background set B contains 250 randomly generated sequences with the same nucleotide composition as in the set A. The model consists of the predicate of two small CMs (K1,K2) connected by the logical OR. Each CM is represented by one pair of matrices. NF-AT/AP-1 pair is revealed by the program.

```
----------------------------------------------------------------------------------
   Score  Predicate      CMcut-off          CM (Matrices/pairs)
----------------------------------------------------------------------------------
0.716741    (K1|K2)  K1: P=0.240461  V$AP1_Q4@0.872000<*V$NFAT_Q6@0.791000,3,1..14
                     K2: P=0.240461  V$PU1_Q6@0.940000**V$AP1_Q4@0.772000,1,12..15

                         Components of the fitness function
--------------------------------------------------------------------------
      Elements         R         T         E         N         P     Total
--------------------------------------------------------------------------
         Value  0.327396  0.508021  0.774441  0.973846  1.000000
        Weight  0.200000  0.200000  0.200000  0.200000  0.200000  1.000000
Weighted value  0.065479  0.101604  0.154888  0.194769  0.200000  0.716741
```

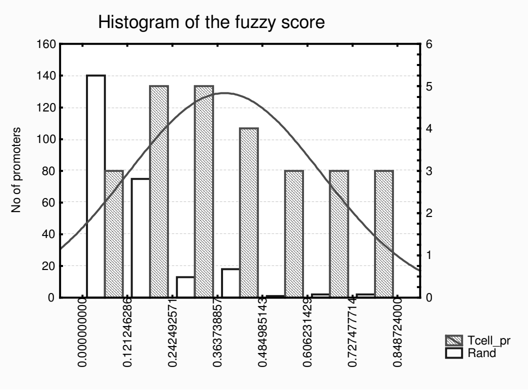

Fig. 1. Histograms of the fuzzy scores of the composite promoter model (K1|K2) computed for the promoters of T-cell specific genes. The majority of the T-cell promoters are characterized by the high values of the fuzzy score.

In another example we performed an analysis of gene expression data on yeast cell cycle taken from the paper of Spellman with co-authors [6]. We selected 5 sets of genes according to their expression in different cell cycle phases: genes specific for G1, S phases and S/G2, M/G1, G2/M transitional states. We retrieved promoters of the length 1100bp (-1000, +100) for these genes and applied CMA program to find cell cycle phase specific composite promoter models.

The parameters of the search were set to reveal a model, which consists of a single CM containing sites for 3 single matrices co-localized in a window of 200bp . The total number of yeast matrices considered was 31. To estimate how the results are consistent with the known facts we compared them with the experimental data on ChIP (chromatin immunoprecipitation) analysis and microarray gene expression data summarized in a recent publication [7]. In Table 5 we present the list of TFs whose weight matrices have been included by CMA into the best promoter models and compared these factors with the factors associated with the corresponding cell cycle phase in the mentioned paper [7].

Table 5. Comparison of TFs predicted by CMA and experimental known TFs regulating different yeast cell cycle phases.

TF	G1	S	S/G2	M/G1	G2/M
FKH1		░ +	░ +		░
FKH2		▒ +	▒ +		▒ +
MBP1	▒ +				
MCM1			+	▒ +	▒ +
ROX1					+
STB1	▒ +	+		▒ +	
STE1			+	▒	
SWI4	▒ +	░			
SWI5		+		▒ +	
SWI6	▒ +	▒ +			
NDD1					▒

Factors found by our program for specific phases are marked with "+". Gray color of the table cells indicates this factor was shown experimentally to be involved in regulating genes in the corresponding cell cycle phases. Factors FKH1 and FKH2 are very similar to each other; also the same can be said about SWI4 and SWI6. Sometimes our program found one of them instead of another; such cases are marked with light-gray. One can see a very good agreement of the predictions made by CMA with the experimental knowledge.

4 Conclusion

In this paper we describe a novel method for analysis and interpretation of gene regulatory regions. The method identifies composite modules – stable combinations of TF binding sites that are shared by the most of the co-regulated promoters. It is generally accepted that such modules are responsible for function-specific regulation of transcription of genes in genome.

Recently, a number of approaches identifying composite motifs that help to discover new regulatory sites for yet unknown transcription factors, were described [8-10]. But such "ab initio" motif finding methods are limited by the length of sequences and may not be suitable for the analysis of long regulatory regions of higher eukaryotic organisms. An important source to identify transcription factor binding sites is the TRANSFAC® database [1]. Novel methods have been developed for the identification of composite modules by utilizing this information [11-13]. In comparison with them the method described here has several advantages, such as (a) the ability to work with data of microarray experiments; (b) optimization not only of the matrix sets, but also of cut-off values for each matrix; (c) search for pairs of matrices, selecting best distance and orientation.

Testing on simulated data shows that our method is able to correctly reveal CMs that are overrepresented in the set of sequences and can be used to analyze data and propose factor combinations that are playing key roles in transcriptional regulation in the given biological context.

Acknowledgements

Parts of the work were funded by grant of the German Ministry of Education and Research (BMBF) together with BioRegioN GmbH “BioProfil”, grant no. 0313092; EU grants “TRANSISTOR” and “COMBIO” and INTAS grant No:03-51-5218.

References

1. Matys, V., Fricke, E., Geffers, R., Gossling, E., Haubrock, M., Hehl, R., Hornischer, K., Karas, D., Kel, A.E., Kel-Margoulis, O.V., Kloos, D.U., Land, S., Lewicki-Potapov, B., Michael, H., Munch, R., Reuter, I., Rotert, S., Saxel, H., Scheer, M., Thiele, S. and Wingender, E. (2003) TRANSFAC: transcriptional regulation, from patterns to profiles. *Nucleic Acids Res*., **31**, 3576-3579.
2. Kel-Margoulis, O., Kel, A.E., Reuter, I., Deineko, I.V., and Wingender, E. (2002) TRANSCompel: a database on composite regulatory elements in eukaryotic genes. *Nucleic Acids Res.,* **30**, 332-334.
3. Kel, A.E., Gossling, E., Reuter, I., Cheremushkin, E., Kel-Margoulis, O.V., and Wingender, E. (2003) MATCH: A tool for searching transcription factor binding sites in DNA sequences. *Nucleic Acids Res.,* **31**, 3576-3579.
4. Goldberg, D. E. (1989) Genetic Algorithms in Search. Optimization and Machine Learning, Kluwer Academic Publishers. Boston, MA.
5. Kel, A., Kel-Margoulis, O., Babenko, V., and Wingender, E. (1999) Recognition of NFATp/AP-1 composite elements within genes induced upon the activation of immune cells. *J. Mol. Biol.,* **288**, 353-376.
6. Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D., Futcher, B (1998) Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Mol Biol Cell*, **9**, 3273-3297.
7. Kato, M., Naoya, H., Nilanjana, B., Futcher, B., Zhang, M.Q. (2004) Identifying combinatorial regulation of transcription factors and binding motifs. *Genome Biology,* **5**, R56.
8. van Helden, J., Rios, A. F., and Collado-Vides, J. (2000) Discovering regulatory elements in non-coding sequences by analysis of spaced dyads. *Nucleic Acids Res.,* **28**, 1808-1818.
9. Guha Thakurta, D. and Stormo, G. D. (2001) Identifying target sites for cooperatively binding factors. *Bioinformatics*, **17**, 608-621.
10. Eskin, E., and Pevzner, P. A. (2002) Finding composite regulatory patterns in DNA sequences. *Bioinformatics*, **18**:Suppl. 1: S354-S363.
11. Kel-Margoulis, O.V., Ivanova, T.G., Wingender, E., Kel, A.E. (2002) Automatic annotation of genomic regulatory sequences by searching for composite clusters. *Pac Symp Biocomput.,* 2002, 187-198.
12. Aerts, S., Thijs, G., Coessens B., Staes, M., Moreau, Y., De Moor, B., (2003) TOUCAN : Deciphering the Cis-Regulatory Logic of Coregulated Genes. *Nucl Acids Res.*, **31**, 1753-1764.
13. Sinha, S., van Nimwegen, E. and Siggia, E.D. (2003) A probabilistic method to detect regulatory modules. *Bioinformatics,* **19, Supl 1**, i292–i301.