POCO: discovery of regulatory patterns from promoters of oppositely expressed gene sets

Matti Kankainen¹ and Liisa Holm¹,²,*

¹Institute of Biotechnology and ²Department of Biosciences, Division of Genetics, University of Helsinki, PO Box 56 (Vilinkkaari 5), FIN-00014, Helsinki, Finland

Received February 15, 2005; Revised March 18, 2005; Accepted April 13, 2005

ABSTRACT

Functionally associated genes tend to be co-expressed, which indicates that they could also be co-regulated. Since co-regulation is usually governed by transcription factors via their specific binding elements, putative regulators can be identified from promoter sets of (co-expressed) genes by screening for over-represented nucleotide patterns. Here, we present a program, POCO, which discovers such over-represented patterns from either one or two promoter sets. Typical microarray experiments yield up- and down-regulated gene sets that may represent, for example, distinct defense pathways. Assuming that a functional transcription factor cannot simultaneously both up- and down-regulate the gene sets, its binding element should respectively be over- and under-represented in the corresponding promoter sets. This idea is implemented in POCO, which tests the hypothesis that the distributions of a pattern differ among three sets of promoters: up-regulated, down-regulated and randomly-chosen. In the program, pattern discovery is based on explicit enumeration of all possible patterns on the alphabet (A, C, G, T and N). The mean occurrences and SDs of the patterns are estimated using bootstrapping and their significance is assessed using ANOVA F-statistics, Tukey’s honestly significantly difference test and P-values. The program is freely available at http://ekhidna.biocenter.helsinki.fi/poco.

INTRODUCTION

One important biological phenomenon that can be explored by using high-throughput genomic techniques combined with computational methods is gene regulation: which genes are regulated and what causes the regulation. Gene expression is typically regulated by two mechanisms working in concert. Chromatin remodeling controls closely located genes, whereas transcription factors activate or repress more distantly located genes. In chromatin remodeling, genes are regulated by altering the structure of DNA blocks, thus controlling the accessibility of the genes to the regulatory factors (1). In the second mechanism, regulated genes are identified via specific nucleotide patterns situated within the regulatory regions of the target genes (2). If a set of similarly expressed genes are distantly located, then the regulation is more likely to be governed by transcription factors, via their patterns, than by chromatin remodeling.

The usual computational pipeline to discover the regulatory transcription factors begins with the generation of the co-expressed gene groups, e.g. by gathering the gene groups from expression data. The second step is to analyze the promoter regions of these gene groups in order to detect over-represented nucleotide patterns that could indicate a co-regulation phenomenon. The final step is then to link the over-represented pattern(s) with a particular transcription factor(s) to create the hypothesis. Various algorithms and tools have been designed to screen for over-represented patterns, and thus also the associated regulatory factors, from an input promoter set. These tools fall into two main categories: probabilistic sequence models (3–5) and pattern enumerators (6–8). The advantage of probabilistic sequence model tools, such as Gibbs samplers, is the ability to find long and general patterns, but the disadvantage is the possibility of failing to find the correct one as a result of getting trapped in a locally optimal solution (8). Pattern enumeration tools are always guaranteed to find the globally optimal solution, because they analyze every pattern in the input set. However, they also have drawbacks, such as reporting multiple hits from a single original pattern, a limited pattern vocabulary and a relatively short pattern length. While one method is more suitable for one purpose and the other is more suitable for another, their goal is the same—to report biologically functional patterns.

To address the question of how to find biologically functional patterns, we have developed a program that can analyze
either one or two input promoter sets. When the program is used to analyze two promoter sets, it cross-validates the found patterns. The cross-validation can be a powerful method to detect the true patterns from distinct cellular pathways or from oppositely expressed gene sets. For example, defense-related genes are needed when the cell is under pathogen attack, but genes needed during an attack of one pathogen can be harmful during an attack of another, and vice versa. Hence, when the first gene group is activated, the second must be repressed and when the second gene group is activated, the first must be repressed. An example is provided by the competition between salicylic acid and jasmonic acid defense-pathways in Arabidopsis thaliana (9). Assuming that both gene groups are regulated by transcription factors and that a single transcription factor cannot simultaneously both up- and down-regulate these given groups, there should be specific transcription factors and characteristic patterns for both gene groups. Otherwise, a common transcription factor would produce similar regulation for both groups and compromise the desired gene expressions. With these hypotheses, differently expressed genes belonging to distinct pathways can be exploited to filter unimportant patterns and to find biologically functional patterns.

METHODS
Principles of POCO
The basic idea of pattern discovery programs is that patterns occurring surprisingly often in the promoters of co-expressed genes can have a regulatory function. Our extended version of this idea is that the same regulatory pattern should not exist in oppositely expressed gene groups. In other words, biological patterns are expected to be over-represented in the promoters of one gene group and under-represented in the other, relative to each other and to the background promoter collection.

Overview of POCO
The program can be described in three parts. First, all patterns composed of the alphabet A, C, G, T and N (N is the wildcard and stands for any of A, C, G or T) that occur in the input promoter sequences or in their reverse complements and have the desired maximum length or length smaller than this are stored into suffix-trie (10). The N-degenerated wildcard forms of patterns grouped into the five groups and reported to the user are automatically generated, must be in FASTA format. Other parameters: ‘promoter length’ is the length of the sequences retrieved for the background calculation as well as the truncation length for the input sequences (shorter sequences are accepted). ‘Motif length’ is the maximum number of nucleotides in the searched patterns. ‘Motifs to report’ is the number of patterns grouped into the five groups and reported to the user (all discovered patterns can be downloaded from the all-file). ‘Min occurrence’ excludes patterns that occur in less than this number of promoters. For example, if the value is 4 and the input set has 20 promoters, then patterns occurring in 4 or more promoters are analyzed. ‘Chosen background’ is the used background organism; the full background promoter collection consists of the promoter sequences of all known genes in this organism. Since comparison points for the significance of a pattern are generated from the real promoter sequences, constraints to remove duplicated TA-repeats, poly(A) or poly(T) etc. are not required. Currently, there are seven organisms available: Anopheles gambiae, A.thaliana, Caenorhabditis elegans, Drosophila melanogaster, Homo sapiens, Mus musculus and Saccharomyces cerevisiae, which are the same as in the POBO-server (12). ‘Number of promoters to pick-out’ and ‘Number of samples to generate’ are bootstrapping parameters. ‘Number of promoters to pick-out’ should be equal or less than the size of the smaller input rather than raw occurrences leads to normally distributed samples and allows the application of statistical tests for equal size data. It can also be noted that the simulations allow the elimination of uninteresting and repeated patterns with statistical tests, because patterns that occur multiple times in few promoters tend to have bigger SDs than patterns that are more evenly distributed.

Third, similarities and dissimilarities between the pattern distributions are evaluated and the patterns are grouped using ANOVA, honestly significantly difference (HSD) test and P-values. ANOVA measures the differences between the means of more than two groups, and its null hypothesis is that the means are the same (13). ANOVA is calculated from the bootstrap estimated means and deviations and its result, the F-score, gives the magnitude of the difference (larger F-scores indicate larger difference than smaller ones). The program also groups the patterns into five groups by calculating the HSD-test (Q-value) for each pair of the three datasets and by comparing results (13). The significance of the patterns is then reported to the users with P-values, which is the probability to find a larger F-score from the data by chance (13). In the program, P-values are calculated from a standardized f-distribution, where each F-score, and pattern, has a corresponding P-value and the interesting P-values locate in the extreme right-hand tail. The P-values are calculated using the observed F-score distribution of analyzed patterns, which is translated to correspond to the standardized f-distribution, if necessary. As a result, P-values that are highly consistent with those in the random data are reported (for detailed explanation of the statistics and equations see Supplementary Material).

Inputs to POCO
To start POCO, users give their promoter sequence set(s) and set a few parameters (see Supplementary Figure 2). The input promoter set(s), for which reversible complementary strands are automatically generated, must be in FASTA format. Other parameters: ‘promoter length’ is the length of the sequences retrieved for the background calculation as well as the truncation length for the input sequences (shorter sequences are accepted). ‘Motif length’ is the maximum number of nucleotides in the searched patterns. ‘Motifs to report’ is the number of patterns grouped into the five groups and reported to the user (all discovered patterns can be downloaded from the all-file). ‘Min occurrence’ excludes patterns that occur in less than this number of promoters. For example, if the value is 4 and the input set has 20 promoters, then patterns occurring in 4 or more promoters are analyzed. ‘Chosen background’ is the used background organism; the full background promoter collection consists of the promoter sequences of all known genes in this organism. Since comparison points for the significance of a pattern are generated from the real promoter sequences, constraints to remove duplicated TA-repeats, poly(A) or poly(T) etc. are not required. Currently, there are seven organisms available: Anopheles gambiae, A.thaliana, Caenorhabditis elegans, Drosophila melanogaster, Homo sapiens, Mus musculus and Saccharomyces cerevisiae, which are the same as in the POBO-server (12). ‘Number of promoters to pick-out’ and ‘Number of samples to generate’ are bootstrapping parameters. ‘Number of promoters to pick-out’ should be equal or less than the size of the smaller input
promoter set, whereas ‘Number of samples to generate’ depends on ‘Motif length’ parameter (see Supplementary Material), but basically higher values yield more accurate results.

**Outputs of POCO**

The output (see Supplementary Figure 2) contains F-score sorted patterns that are assigned into five groups by comparing the results of the HSD-tests. The five groups are groups 1 and 2, patterns over-represented only either in the first (1) or the second (2) input promoter set; group 3, patterns over-represented in both input promoter sets; and groups 4 and 5, patterns over-represented in the first input promoter set and under-represented in the second (4) or vice versa (5). When ‘Min occurrence’ is used, the over-represented patterns must occur at least in this number of promoters. The output contains the following information: the number of promoters with the pattern, total occurrence, input and background bootstrap means and SDs, t-test values, F-score, Q-, Z- and P-values. All reported patterns are linked to the POBO-server (12), which can be used to perform more demanding bootstrap simulations, to disambiguate the wildcards (N) or to map the locations of the patterns in the promoters.

**Availability and running the program**

POCO is written in the C++ language, a MySQL (http://www.mysql.com/) database is used to store the background promoters and the P-values are calculated using the DCDFLIB-package (http://www.netlib.org/random). In the server, some parameters are limited: number of characters in input sets =31 000, Motif length =8, Number of promoters to pick-out =20 and Number of samples to generate =300. Typical run times, when using the maximum parameters, are a few hours. The server, help-pages, backgrounds (MySQL dump-files) and the source code to run POCO locally are available at http://ekhidna.biocenter.helsinki.fi/poco/.

**Artificial data tests**

For comparison of Z-scores and F-scores and randomized data, see Supplementary Material.

**RESULTS**

**Example data**

To demonstrate the functionality of the program and to give an example how to use it, we reanalyze *A. thaliana* microarray datasets from a transgenic plant constitutively over-expressing WRKY70 (14). In the experiment, it was observed that a set of defense-related genes were either up-regulated (24 genes) or down-regulated (10 genes) compared with the control plant. WRKY is a plant-specific transcription factor superfamily and it is the second largest transcription factor superfamily of the *A. thaliana* consisting of 74 proteins (15,16). The WRKY factor has been reported to bind onto patterns like TTTGAC(C/T), which are called the W-box (15). Their biological role is to regulate genes involved in developmental processes, responses to various stresses and plant defense via the salicylic acid pathway (14). We retrieved the promoter sequences of the reported genes (from the ATG) and analyzed them using the following parameter settings: Motif length, 8 bp; Promoter length, 1500 bp; Number of promoters to pick-out, 20; Number of samples to generate, 10000; Min occurrence, 1; and Chosen background, *A. thaliana*.

**Discovered patterns from the WRKY70-example**

Table 1 shows patterns over-represented in the up-regulated and under-represented in the down-regulated promoter set. An interesting finding is that three out of the five patterns (patterns at rank 1, 2 and 5 and corresponding t-scores and randomized data, all reported patterns are linked to the POBO-server (12), which can be used to perform more demanding bootstrap simulations, to disambiguate the wildcards (N) or to map the locations of the patterns in the promoters.

**Availability and running the program**

POCO is written in the C++ language, a MySQL (http://www.mysql.com/) database is used to store the background promoters and the P-values are calculated using the DCDFLIB-package (http://www.netlib.org/random). In the server, some parameters are limited: number of characters in input sets =31 000, Motif length =8, Number of promoters to pick-out =20 and Number of samples to generate =300. Typical run times, when using the maximum parameters, are a few hours. The server, help-pages, backgrounds (MySQL dump-files) and the source code to run POCO locally are available at http://ekhidna.biocenter.helsinki.fi/poco/.

**Artificial data tests**

For comparison of Z-scores and F-scores and randomized data, see Supplementary Material.

**RESULTS**

**Example data**

To demonstrate the functionality of the program and to give an example how to use it, we reanalyze *A. thaliana* microarray datasets from a transgenic plant constitutively over-expressing WRKY70 (14). In the experiment, it was observed that a set of defense-related genes were either up-regulated (24 genes) or down-regulated (10 genes) compared with the control plant. WRKY is a plant-specific transcription factor superfamily and it is the second largest transcription factor superfamily of the *A. thaliana* consisting of 74 proteins (15,16). The WRKY factor has been reported to bind onto patterns like TTTGAC(C/T), which are called the W-box (15). Their biological role is to regulate genes involved in developmental processes, responses to various stresses and plant defense via the salicylic acid pathway (14). We retrieved the promoter sequences of the reported genes (from the ATG) and analyzed them using the following parameter settings: Motif length, 8 bp; Promoter length, 1500 bp; Number of promoters to pick-out, 20; Number of samples to generate, 10000; Min occurrence, 1; and Chosen background, *A. thaliana*.
for the binding of some C2H2 zinc finger proteins (17,18). Another interesting finding is that the flanking sequences of these patterns resemble the known W-box. Also, a pattern TTGACTNA (at rank 39, F-score 8239.46, P-value 1 \times 10^{-5}) that is highly similar to the known W-box is in the results of group 4. The data indicate a possible presence of an over-represented W-box in the promoters of the up-regulated genes that is missing in the down-regulated promoter set. Therefore, it is possible that these patterns bind WRKY-factors and presumably the over-expressed WRKY70. The two other patterns at ranks 3 and 4 (P-values 3.2 \times 10^{-5} and 3.3 \times 10^{-5}) resemble a pattern C(A/T)TCATTC (Figure 1b), which was found to be over-represented within the promoters of the up-regulated genes in a chitin-treated plant (19). Chitin induces defense-related responses in plant and the proposed hypothesis was that the transcription factor that binds onto this pattern could co-operate with WRKY (19).

Table 2 shows patterns under-represented in the up-regulated and over-represented in the down-regulated WRKY70 promoter set. In the table, patterns at rank 2 and 4 (P-values 7.6 \times 10^{-5} and 2.1 \times 10^{-5}) resemble a pattern TAGTNCTGT (Figure 1c) that was demonstrated to be the regulatory element of several auxin-responsive genes in soybean (20). Auxin is a plant signaling molecule that is involved in most major growth responses, such as cell extension, division and differentiation (21). It has also been shown that auxin is engaged in plant defense, where it interacts with the jasmonic acid defense pathway (22). The pattern at rank 3 (P-value 1.0 \times 10^{-4}) has similarities with a pattern AGATCCAA (Figure 1d), which was shown to bind novel GATA-type transcription factors (AGPs) (23). The expression of one of them, AGP1, was up-regulated by wounding and this protein has been proposed to be a wounding specific regulator (23). The patterns at ranks 1 and 5 (P-values 2.9 \times 10^{-5} and 2.5 \times 10^{-4}) did not show strong similarity to previously reported binding elements.

As a summary, on the one hand the patterns in the promoters of up-regulated genes (Table 1) can be associated with a WRKY-mediated salicylic acid defense pathway. On the other hand, patterns discovered in the promoters of down-regulated genes (Table 2) can be associated with wounding and with another plant defense pathway, the jasmonic acid pathway. Therefore, the results suggest that the genes in the down-regulated gene cluster of the WRKY70-experiment could belong to distinct and competing defense pathways, which are silenced to favor other defense pathways. This hypothesis is also supported by experimental data, which show that the jasmonic acid pathway is indeed activated under different stresses and stimuli than the salicylic acid pathway (9).

**DISCUSSION**

The program demonstrated in this article can be used to find biologically significant putative regulatory patterns. Our opinion is that the program is well suited for a subset of biological experiments where different promoter sets can be obtained and where competition really takes place. This situation can arise, for example, in comparisons of competitive pathways, such as defense and stress, and in comparisons of oppositely expressed gene sets.

**SUPPLEMENTARY MATERIAL**

Supplementary Material is available at NAR Online.

**ACKNOWLEDGEMENTS**

The authors thank Günter Brader for providing the data, Christopher Wilton for fixing the language, and Alvis Brazma and the group members for helpful discussion. This work was supported by a grant from the Ministry of Education to M.K. Funding to pay the Open Access publication charges for this article was provided by Academy of Finland.

Conflict of interest statement. None declared.

**REFERENCES**


