Predicting Gene Regulatory Elements in Silico on a Genomic Scale

Alvis Brazma, Inge Jonassen, Jaak Vilo and Esko Ukkonen

Genome Res. 1998 8: 1202-1215
Access the most recent version at doi:10.1101/gr.8.11.1202

LETTER

Predicting Gene Regulatory Elements in Silico on a Genomic Scale

Alvis Brāzma,[1] Inge Jonassen,[2] Jaak Vilo,[3,4] and Esko Ukkonen[3]

[1]European Molecular Biology Laboratory (EMBL) Outstation–Hinxton, European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SD, UK; [2]Department of Informatics, University of Bergen, Høyteknologisenteret, N5020 Bergen, Norway; [3]Department of Computer Science, FIN-00014 University of Helsinki, Helsinki, Finland

We performed a systematic analysis of gene upstream regions in the yeast genome for occurrences of regular expression-type patterns with the goal of identifying potential regulatory elements. To achieve this goal, we have developed a new sequence pattern discovery algorithm that searches exhaustively for a priori unknown regular expression-type patterns that are over-represented in a given set of sequences. We applied the algorithm in two cases, (1) discovery of patterns in the complete set of >6000 sequences taken upstream of the putative yeast genes and (2) discovery of patterns in the regions upstream of the genes with similar expression profiles. In the first case, we looked for patterns that occur more frequently in the gene upstream regions than in the genome overall. In the second case, first we clustered the upstream regions of all the genes by similarity of their expression profiles on the basis of publicly available gene expression data and then looked for sequence patterns that are over-represented in each cluster. In both cases we considered each pattern that occurred at least in some minimum number of sequences, and rated them on the basis of their over-representation. Among the highest rating patterns, most have matches to substrings in known yeast transcription factor-binding sites. Moreover, several of them are known to be relevant to the expression of the genes from the respective clusters. Experiments on simulated data show that the majority of the discovered patterns are not expected to occur by chance.

Completely sequenced genomes, together with the emerging DNA microarray technologies enabling the measurement of the gene expression levels in cell cultures (Schena et al. 1995; for a survey, see Ramsay 1998), are opening new possibilities for studying gene regulation. The sequencing of the first eukaryotic genome (the yeast *Saccharomyces cerevisiae*) was completed in 1996 (Goffeau et al. 1996; Mewes et al. 1997). Data about the expression levels of almost all of the ~6000 yeast genes have been obtained (DeRisi et al. 1997; Velculescu et al. 1997; Wodicka et al. 1997) during 1997. In particular, DeRisi et al. (1997) measured the relative expression levels of the yeast genes at seven consecutive time points (in 2-hr intervals) during a shift from anaerobic to aerobic metabolism (diauxic shift). They showed that some of the genes that are known to be involved in metabolic pathways related to the diauxic shift underwent a very significant change in their expression level during the shift. By treating the expression measurements as a time series, it is possible to cluster genes according to similarities in their expression profiles. It may be hypothesized that at least some of the genes in a cluster are regulated by similar mechanisms.

The transcription regulation mechanisms in eukaryotic genomes are not well understood. Evidently, however, an essential role is played by transcription factors, which can bind to particular DNA sequences, called transcription factor-binding sites, believed to be about 5–25 bp long. In yeast, these sites are usually within several hundred base pairs upstream of the respective ORFs (Mellor 1993).

Regular expression type patterns, as well as nucleotide distribution matrices, have both been used for describing transcription factor-binding sites, (e.g., see Bucher 1990; Ghosh 1990; Chen et al. 1995; Wingender et al. 1996). Inference of such descriptions from the sequences that are assumed to contain a site for a particular transcription factor is a difficult problem as the consensus of the different binding sites of the same transcription factor is often rather weak. Algorithms have been proposed for inferring such descriptions from sets of relatively small number of sequences (about 20) in which all

[4]Corresponding author.
E-MAIL vilo@cs.helsinki.fi; FAX 358 9 708 44441.

 8:1202–1215 ©1998 by Cold Spring Harbor Laboratory Press ISSN 1054-9803/98 $5.00; www.genome.org

or almost all of the sequences are known to contain the site for the respective transcription factor (e.g., see Stormo and Hartzell 1989; Wolfertstetter et al. 1996; van Helden et al. 1998). More recently, van Helden et al. (1998) and Yada et al. (1998) have proposed methods for the discovery of putative transcription factor-binding sites from larger data sets. Yada et al. (1998) applied their method to analyze about 400 human promotor sequences.

Apparently, an even more difficult problem is identifying potential binding sites or other regulatory elements from sets of sequences only suspected to contain such elements. In this report, we consider the case when only a small portion of the sequences in the given set may actually contain a common regulatory element, and the total number of sequences may be up to thousands. In this setting, it may not be possible to infer precise binding site descriptions; still, if the number of sequences containing a common regulatory element is larger than would be expected by chance, it may be possible to obtain hints about sequence properties of such an element and in which particular sequences it may be present.

An obvious difficulty in attacking this problem is the computational complexity of the algorithmic problem of discovering interesting sequence patterns in a large collection of sequences only some of which may contain a common pattern. Ultimately the results of such discoveries should be taken as predictions that must be verified by independent, that is, wet biology, means. Still, some validation can be obtained by comparing the discovered site descriptions to the transcription factor database entries, or by statistical means by comparing the distribution of the discovered patterns to the distribution in simulated data.

Pattern discovery methods basically fall into two groups; sequence-driven and pattern-driven methods (for a survey, see Brāzma et al. 1998a,b). Algorithms in the first group normally work by combining the results of pairwise sequence comparisons to form patterns that match the subsets of the sequences. These algorithms are too slow to find patterns that occur in arbitrarily sized subsets of thousands of sequences. Pattern-driven algorithms work by enumerating or searching a predefined pattern class to find patterns and their occurrence frequencies. In these methods, one needs a very fast method for locating all matches of each pattern from the search space. Special data structures and pattern occurrence lists have been used for this purpose, but the methods have been limited to the analysis of smaller data sets.

We have developed a new, more powerful, pattern discovery algorithm that is able to discover various subclasses of regular expression type patterns of unlimited length common to as few as ten sequences from thousands. We used this algorithm for predicting regulatory elements from gene upstream regions in the yeast *S. cerevisiae*.

We considered two cases. First, we looked for patterns that occur more frequently in the gene upstream regions than in randomly chosen regions in the yeast genome. For each pattern present in at least 10 sequences (from >12,000), we calculated a score equal to the ratio of the number of upstream regions that contain the pattern divided by the number of random regions (of the same length and number) that contain the pattern, and rated the patterns according to this ratio.

In the second case, we used information from the yeast genome expression data (DeRisi et al. 1997) to cluster the genes according to their expression profiles. After clustering the upstream regions (treating the expression measurements as time series) we selected characteristic clusters according to some rigorous criteria. We hypothesized that some of the genes in a cluster may contain binding sites for the same transcription factors or other common regulatory elements. We used our algorithm to look for patterns that are over-represented in each cluster as compared with other upstream regions.

We systematically compared the high-scoring patterns that we discovered to the transcription factor-binding site descriptions for the yeast in TRANSFAC database (Wingender et al. 1996). We found that most of the discovered patterns (both from the total set of upstream regions and from the clusters) have matches to substrings of genome regions that contain transcription factor-binding sites. We also compared the distribution of patterns present in upstream regions to the distribution of the patterns that can be discovered in random regions of the genome and showed that the distributions are rather different. The comparison with the TRANSFAC database as well as the overall statistics of the discovered patterns suggest that many of the discovered patterns can be important for the expression profile of the particular clusters of genes or for the transcription or translation initiation in general.

RESULTS

First, we describe the pattern discovery in the complete set of yeast gene upstream regions, then the clustering of the yeast gene expression data, and finally, the results obtained by pattern discovery

from within the subsets of upstream regions of genes sharing similar expression profiles.

We considered three different types of patterns: (P1) substring patterns (i.e., words in the alphabet A, T, G, C); (P2) substring patterns with wild cards (of fixed length); and (P3) patterns with character groups [such patterns can be represented as words over IUPAC code (Corhish-Bowden 1984) characters; here we will use a more explicit notation].

We denote wild-card positions by a dot in the pattern (e.g., TA.A), and the group positions by enlisting all possible characters in square brackets (e.g., T[AT]A). A wild-card position is group position [ATCG], that is, all characters are allowed. For instance, pattern A[TG].C matches all strings that contain a substring beginning with A, followed by either T or G, followed by any character, followed by C. In practice, for reasons of efficiency, we restrict ourselves to various subclasses of these pattern classes (e.g., limiting the number of possible wild cards or group symbols). The implementation of the algorithm, results, data, and additional images are available on the worldwide web at http://www.cs.Helsinki.FI/~vilo/Yeast/.

Discovering Patterns from the Total Set of Upstream Regions

We extracted upstream regions relative to all ORFs, as annotated in the MIPS Yeast genome database (Mewes et al. 1997). Concretely, we extracted seven sets of upstream regions of length 100 from the positions −100 to 0, −150 to −50, −200 to −100, −250 to −150, −300 to −200, −350 to −250, and −400 to −300, a set of regions of length 300 from positions −300 to 0, and a set of regions of length 600 from positions −600 to 0 (all positions are relative to the start codon of the ORF; see Methods). Also we extracted two sets of sequences of the same number and length from randomly selected locations of the same chromosome. These sets of random regions were used as random samples of the yeast genome sequences (the nucleotide and dinucleotide distribution in the random regions reflected that in the genome in general) (1) to compare the upstream regions to random regions for identifying patterns that are more frequent in upstream regions than in the genome in general and (2) to compare the two random sets against each other for testing whether the pattern occurrence statistics resulting from the comparison of upstream and random regions can be explained by chance.

We analyzed these data sets for occurrences of patterns. We presented each pattern that occurred at least 10 times in upstream or random regions as a dot in a two-dimensional plot (see Fig. 1, left column). The vertical axis shows the number of upstream regions, and the horizontal axis the number of random regions, where the pattern is present.

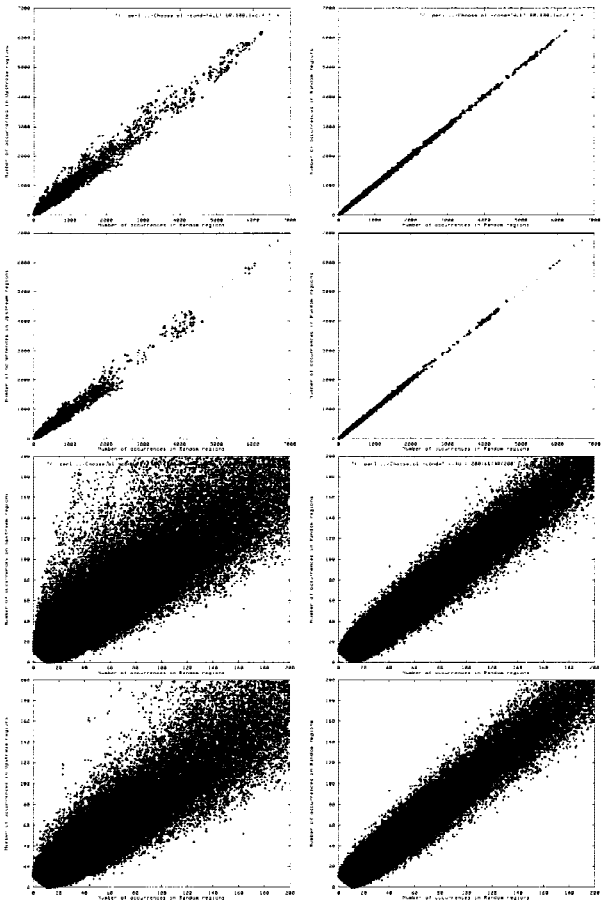

Figure 1 The distribution of all patterns (of unrestricted length) with at most one wild-card symbol in the regions −250 to −150 (upstream from the ORFs) and randomly chosen genomic regions of length 100 bp. Dots in graphs in the *left* correspond to patterns that occur in *x* sequences from the random regions (along horizontal axis) and *y* sequences from the upstream regions (vertical axis). In graphs on the *right*, the upstream regions are replaced by another set of random regions; therefore, these plots show the expected statistics if the regions are chosen at random. (*Top row*) All patterns with at least 10 occurrences. (*Second row*) Subset of top row with all patterns containing at least two characters C or G and not containing any of the substrings AAAA, TTTT, ATAT, or TATA. (*Bottom two rows*) Same plots as in the first two rows, but only including patterns with at most 200 occurrences in upstream or random regions (i.e., zoomed to the lower left corner).

Hence a dot in plot location (x; y) indicates that there is a pattern that occurs in x random regions and y upstream regions. The patterns deviating from the diagonal, and particularly, being above the diagonal, are the ones that can distinguish the upstream regions from the random regions (and, therefore, are likely to distinguish the upstream regions from the genome in general), in contrast to the patterns that fall close to the diagonal and thus occur with the same frequency in upstream and random regions. The dots farthest above the diagonal correspond to the patterns that are potential candidates for regulatory elements. For each pattern we calculated a score as defined by equation (2) in Methods, which is essentially the number of occurrences in the upstream regions divided by the sum of the number of occurrences in the random regions and a correcting constant.

A control experiment (right column in Fig. 1) was done to estimate whether the difference in pattern frequencies observed for upstream versus random sequence segments could be explained by chance. In the control experiments, we compared two sets of random regions. The pattern occurrence statistics obtained when comparing the upstream regions to the random regions is rather different from the statistics obtained when comparing two sets of random regions. We also tested that this considerable difference can be explained neither by higher AT content in the upstream regions, nor by poly(A), poly(T), or poly(AT) patterns. To achieve this goal, we plotted the patterns containing at least two characters C or G and not containing any of the substrings AAAA, TTTT, ATAT, or TATA. The difference between the plots remained essentially as strong (see Fig. 1). Therefore, we conclude that the distribution of patterns in the upstream regions differs from the distribution in regions. In particular, there are some specific patterns that occur considerably more often in upstream regions than in random regions.

The best distinction (as judged by visual inspection) between upstream and random regions by substring patterns was achieved for upstream regions of length 100 when counting matches only on the gene's strand. [The use of only one strand can be justified because of the very distinct distribution of different bases in a region of 300 bp upstream from the start of the gene (see Fig. 3, below, in Methods).] Similar differences were observed for all considered lengths and region relative positions. We also experimented with the three sets of sequences of length 600 and 300 bp, analyzing substring patterns on either strand; and the sequences of length 100, analyzing the patterns that contain up to one wild card. Some results for patterns with at most one wild-card symbol from regions of length 100 bp at upstream positions −250 to −150 are shown in Figure 1.

Many of the top-scoring patterns, particularly, for the region −250 to −150, are effectively poly(T) sequences. Still, as mentioned above, these trivial poly(T) patterns cannot explain the differences in the pattern occurrence statistics compared with random genomic regions; therefore, overall, the patterns not containing poly(T) sequences are significant. We removed from the list of discovered patterns the ones that contain substrings TTTT or AAAA (and additionally the patterns ending in the wild-card—we call the remaining patterns nontrivial) and the list of the 20 remaining highest scoring patterns are given in Table 1 (the numbering of the patterns is given for the total list of patterns including the trivial ones).

We compared the groups of highest scoring nontrivial patterns from each of the seven regions of length 100 bp of various distances with the respective ORFs. We used the program Pratt (Jonassen 1997) to try to find patterns that would be a consensus for a substantial number of patterns for each group. More concretely, we took the 20 highest scoring patterns and used Pratt to discover patterns matching at least 6 patterns. It turned out that only for regions −150 to −50, the highest scoring pattern groups have a relatively good consensus pattern GATG.G.T, the region −200 to −100 has two consensus patterns, T.ACCCG and CGGGT.A, which are mutually symmetric, and the region −250 to −150 has the consensus ACCCG (note that it is a subpattern of T.ACCCG). No significant consensus patterns have been found for other regions.

We also matched the 50 highest scoring nontrivial patterns for each of the regions against all the transcription factor-binding site descriptions given in the TRANSFAC (Wingender et al. 1996) database for the yeast. The results of the exact matches are given in the Table 2 (by an exact match, we mean that the discovered pattern exactly matched a substring in the binding site description). Note that although the highest scoring patterns from neighboring regions are not necessarily similar themselves, the number of coinciding binding sites (from TRANSFAC) matched by patterns from two regions show a considerable correlation with the distance between the positions of the regions.

The complete list of the discovered patterns is available on the World Wide Web.

Table 1. Highest Scoring Nontrivial Patterns with (at Most) One Wild-Card Symbol

No.[a]	Pattern	Score[b]	$N+$[c]	$N-$[d]
	A. Regions −100..0			
2	AAG.AAACAAA	6.54	37	1
6	A.TAAGAACA	5.79	27	0
8	A.AATAGGA	5.61	43	3
9	AAGAAA.CAAA	5.58	26	0
12	GTAACAA.C	5.36	25	0
13	AAA.AACTTA	5.36	25	0
20	ACAAC.TAA	5.09	39	3
21	AG.AAACAAA	5.06	64	8
23	ACAAACAA.A	4.97	48	5
26	AATAGTA.A	4.92	77	11
32	AATAGTATA	4.77	27	1
34	TCACTAC.T	4.72	22	0
35	CAAACA.ACA	4.72	22	0
37	ACA.ATAGA	4.72	55	7
42	AGAGA.ATA	4.63	54	7
47	AATAAACAA.A	4.59	26	1
50	AAAG.ACAAG	4.57	35	3
52	CTAAGAA.A	4.55	53	7
56	A.AAGGGAAG	4.51	21	0
57	CAAA.TAAC	4.50	48	6
	B. Regions −250..−150			
14	TTACCCGC	6.22	29	0
58	GT.ACCCG	5.59	54	5
71	T.ACCCGC	5.48	42	3
126	CGGGTA.T	5.06	64	8
141	G.TACCCG	4.97	48	5
165	CGGGTAA.A	4.87	47	5
178	GTTACCCG	4.83	37	3
305	TACAT.TATA	4.43	65	10
353	TTTCTC.TTT	4.32	46	6
372	TTACCCG	4.30	119	23
379	TTTCCTGT.T	4.29	20	0
405	CTCATCTC.T	4.24	24	1
425	TCACGTGA	4.20	28	2
427	T.ATATATTC	4.20	28	2
454	CGGGTAA	4.12	114	23
460	TGTGT.GAT	4.08	19	0
465	ATTACCCG.A	4.08	19	0
474	G.ACATATAT	4.06	23	1
485	TA.GTAAAC	4.05	27	2
500	TTTCTCT.TT	4.03	47	7

Matches were only allowed on the *W* (gene) strand.
[a]No. of the pattern enumerating them decreasingly by scores (before trivial patterns were removed).
[b]From equation 2.
[c]No. of upstream regions matching the pattern.
[d]No. of random sequences matching the pattern.

Clustering the Gene Expression Data

DeRisi et al. (1997) studied the relative expression rate changes of yeast genes during the diauxic shift. They inoculated yeast cells from an exponentially growing yeast culture into fresh medium and after some initial period, harvested samples at seven 2-hr intervals, isolated their mRNA, and prepared fluorescently labeled cDNA. Two different fluorescent moieties were used—one for cells harvested in each of the successive time points, the other for reference, from cells harvested at the first time point. The cDNAs from each time point, together with the reference cDNA were hybridized to the microarray with ~6400 DNA sequences representing ORFs of the yeast genome. Measurement of the relative fluorescence intensity for each of the ~6400 × 7 elements reflect the relative abundance of the corresponding mRNA in each cell population. The measurement data is available on the Internet.

We used the data from these yeast gene expression studies (DeRisi et al. 1997) and clustered all the genes by similarities in their expression profiles in several alternative ways. To achieve this goal, we developed and implemented a simple algorithm based on discretizing the time series of the measurement space into a simplified form and then clustering these simple time series. Some rigorous selection criteria were used for defining good clusters (for details, see Methods). This produced 32 different clusters containing from 10 to 77 ORFs each and 11 clusters containing at least 25 ORFs (see Table 3).

The most significant changes in gene expression rates during the diauxic shift occurred during the last two time points. This significance is reflected in the clusters that we obtained (although some fluctuations at earlier time points occur for smaller groups of genes, which may be due to noise). Many of the constructed clusters strongly overlap. From the 11 clusters of at least 25 ORFs each, in 8 clusters, the expression level is increasing in the time point 6, in 2 it is decreasing, and in 1 it is unchanged.

Discovering Patterns from the Gene Clusters

We studied whether clusters of genes with similar expression profiles can help to discover sequence patterns putatively describing transcription factor-binding sites. For each cluster, we compared the corresponding upstream regions of length 300 bp against all other upstream regions. The algorithm was used to find the highest scoring patterns containing up to three wild cards. The patterns were

Table 2. Matches to TRANSFAC Binding Sites for the 50 Best Patterns Found for Each 100-bp Upstream Region

	−100	−150	−200	−250	−300	−350	−400
Y$ARS1_03	.						
Y$ARSH4_02	.				+	+	
Y$CAR1_01							+
Y$CAR2_01							+
Y$CDC2_01			.	.			
Y$CDC9_01			+	+			
Y$CEN12_01					+	+	+
Y$CEN6_01					+	+	+
Y$CENIV_01					+	+	+
Y$CFES_01							.
Y$CHA1_04							.
Y$CSVIII_02							+
Y$CTA1_01				+	+	+	
YSCYC1_12		.	+	+	.	.	
Y$CYC1_14						+	+
Y$DDR2_02						.	
Y$G3PDH_01				.	.	+	+
Y$GAL1_03							+
Y$GAL1_04		.	+	+	.	.	
Y$GAL1_06							+
Y$GAL1_14		.	+	+	.	.	+
Y$GAL2_03				+	+	.	+
Y$HO_06		+	+				
Y$HO_07							.
Y$ICL1_01							+
Y$MAL61_02			.				
Y$MES1_01				+	+	+	+
Y$PDC1_02							.
Y$PGK_01					.		
Y$PHO8_02		.	+	+	.	.	
Y$POX1_01			.	.	.	.	
Y$RAP_01			+	+	+		+
Y$RP51A_01							+
Y$RPL16A_01						+	+
Y$RRNA_01		+	+	+	+	+	
Y$RRNA_02			+	+			
Y$STE6_02				+			
Y$SUC2_02				.			
Y$TEF2_01						+	+
Y$TOP2_01							+
Y$TRP1_01		+	+	+	+		
Y$TRP5_01			+	+			
Y$X40_01			+	+	+	+	
Y$Y30_01			.	+	+		

For each 100-bp region starting at the seven different positions upstream from ORFs, the 50 highest scoring nontrivial patterns were matched (in substring sense) against the yeast transcription factor binding sites as given in the TRANSFAC (Wingender et al. 1996) database. The first column gives the binding site identifier in TRANSFAC that is matched by one of the best patterns from any of these sets.

(+) At least one of the respective patterns matches exactly the corresponding TRANSFAC site.

(.) A pattern matches only the reverse-complement of the TRANSFAC site.

Table 3. Summary Information about Pattern Scores in the Clusters and Random Sets

Cluster name	No. of genes	Score range for the 10 best patterns	Score for the best pattern in the resp. random set
C^r(3,4)(000010)	77	3.80–2.80	1.70
C^r(5,2,4)(000020)	55	3.99–2.96	1.94
C(5,2,4)(0000021)	41	3.77–2.95	2.12
C(5,2,4)(0000022)	38	7.15–4.11	2.09
C^r(3,5)(000010)	38	3.50–3.17	2.73
C(5,2,4)(0000012)	37	2.87–2.52	2.42
C^r(5,3,5)(000020)	37	3.60–3.08	2.12
C^r(5,2,3)(00002-1)	25	4.00–3.89	3.86
C^r(3,3)(000001)	26	3.55–3.29	4.33
C(5,2,6)(00000-1-2)	27	3.69–3.21	3.13
C(5,3,6)(00000-1-2)	25	4.00–3.89	3.86

For explanation of the cluster names, see Methods. The first eight clusters consist of genes the expression level of which increase at time point 6; the last two of genes the expression level of which decrease at time point 6. The statistics include all patterns (trivial variants were not removed).

matched on either strand and ranked by the score given by equation 1 in Methods.

To evaluate the overall significance of the result, we picked for each cluster a random subset (of the same size) of upstream regions from the total set of genes, and analyzed this set exactly the same way as the cluster. We found that, for 10 clusters out of 11 containing at least 25 sequences and for all clusters containing at least 30 sequences, the scores of the best patterns found from clusters is better than for the best patterns found from the randomly picked sets (see Table 3; Fig. 2).

The largest clusters (>30 sequences) correspond to the expression profiles with increase in the expression level at time point 6, and, for each of these clusters, high-scoring patterns containing the substring CCCC are found [CCCC has score 1.9 for cluster C^r(5,2,4)(000020)]. The cluster C(5,2,4)(0000022) with 38 sequences contains patterns that are standing out as the highest in comparison with the pattern scores for the random set of the size 38. The highest scoring patterns are given in the Table 4 (note that in Table 4 we have removed trivial variants of the patterns, e.g., patterns ending with wild-card characters). Pattern CCCCT (and its reverse complement AGGGG) is the highest scoring for the cluster C^r(5,2,4)(000020) (containing 55 sequences) matching 64% (35 out of 55) of sequences in the cluster and 21% (1280 out of 5921) of remaining upstream regions, thus getting a score of 2.95. Other high-scoring patterns in this cluster

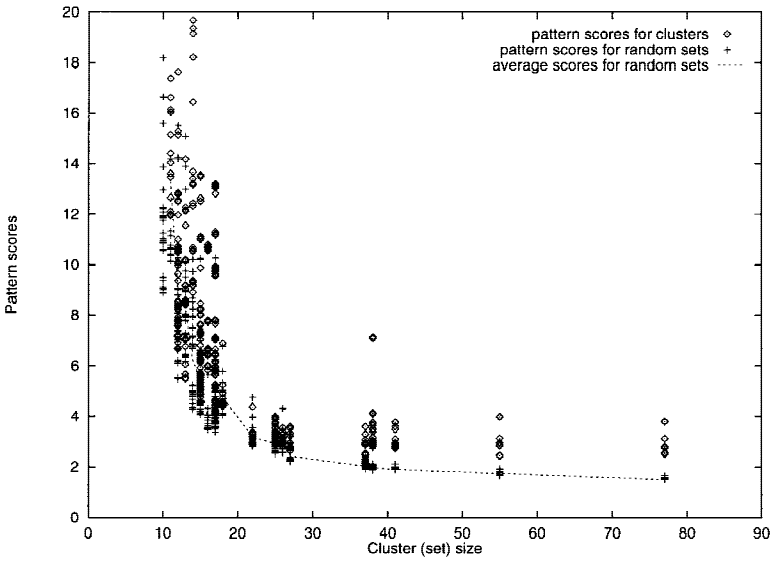

Figure 2 The plots of the scores of the 30 best patterns found from the clusters of upstream sequences from genes with similar expression profiles and of random sets of the upstream sequences of the same size. The dotted line is the average score of the 30 best patterns found from the random sets of the respective sizes. For the sets of 30 sequences and more, the pattern scores from the random sets of the upstream sequences are stabilizing and are considerably lower than for 30 best pattern scores for the respective clusters.

Table 4. Highest Scoring Patterns for the Cluster ***C***(5,2,4)(0000022)

Pattern	N+[a]	Total+[b]	Score[c]	TRANSFAC (exact matches)
A. Highest score in experiment allowing patterns to have at most 3 wild cards and no group characters				
CCCCT..T	22	27	7.09	Y$DDR2_01, Y$DDR2_02, Y$TPI_02
A..AGGGG	22	27	7.09	–
GGGGC	20	27	4.09	Y$GAL2_02, Y$SUC2_02, Y$RRNA_01 Y$ERG11_01
GCCCC	20	27	4.09	Y$CYB2_02
G..GGGG	19	28	3.73	Y$CYC1_04, Y$CYC1_05, Y$CYC1_06
CCCC..C	19	28	3.73	Y$GAL3_01, Y$MAL2R_01
CCCC...T	25	42	3.65	Y$SUC2_01, Y$DDR2_01, Y$DDR2_02 Y$TPI_02, Y$GAL3_01, Y$GAL4_01 Y$MAL2R_01, Y$MAL63_01, Y$PDC1_02 Y$HAP4_01
A...GGGG	25	42	3.65	Y$SUC2_02, Y$RRNA_01, Y$ERG11_01 Y$MEL1_02, Y$FPS1_01
CCCCT	25	38	3.03	Y$DDR2_01, Y$DDR2_02, Y$TPI_02
AGGGG	25	38	3.03	Y$CAR1_02
CCCT..TT	19	22	2.95	Y$DDR2_01
AA..AGGG	19	22	2.95	–
GGG.TG	20	21	2.93	–
CA.CCC	20	21	2.93	Y$GAL1_04, Y$CYC1_12, Y$GAL1_14 Y$DDR2_02, Y$TPI_02
B. Highest score in experiment allowing patterns having at most one group character with two alternative letters (all pairs allowed)[e]				
CCCCT[GT]	20	28	3.86	Y$DDR2_01, Y$DDR2_02, Y$TPI_02
CCCCT[AT]	20	24	3.58	Y$DDR2_02, Y$TPI_02
[CG]CCCC	24	47	3.27	Y$CYB2_02, Y$GAL2_02, Y$SUC2_02, Y$RRNA_01, Y$ERG11_01
CCCC[CT]	29	58	2.94	Y$DDR2_01, Y$DDR2_02, YTPI_02, YSUC2_02, Y$CAR1_02, Y$ERG11_01
[AG]CCCC	29	48	2.90	Y$CYB2_02, Y$DDR2_02, YTPI_02, YCYC1_04, Y$CYC1_05, Y$CYC1_06, Y$GAL2_02, Y$SUC2_02, Y$RRNA_01, Y$CAR1_02, Y$ERG11_01, Y$GAL1_15

Trivial pattern variants were removed, e.g., patterns ending with a wild-card character.
[a]No. of upstream regions matching the pattern.
[b]Total number of matches in the upstream regions.
[c]Normalized version of pattern score.
[d]TRANSFAC entries matching the pattern.
[e]Best patterns from experiment 2 not also found in experiment 1.

include C..CCC.T (score 2.88), T.C..CCC (score 2.85), and T.AGGG (score 2.27). Furthermore, the pattern CCCCT was also among the 10 highest scoring patterns found for the clusters C^r(3,5)(000010), C^r(5,3,5)(000020), and C(5,2,4)(0000022). These four clusters strongly overlap (17 ORFs are in all four clusters). DeRisi et al. (1997) describe a set containing seven genes (see Fig. 5C in DeRisi et al. 1997) out of which five are contained in our cluster C^r(5,2,4)(000020). They note the presence of the pattern CCCCT in the upstream regions of each gene in their set and that it is known to be a stress-responsive motif.

We also analyzed the upstream regions of the genes in the clusters having expression level decrease at time point 6, and found that they contain patterns with matches to binding sites for the RAP1 factor, which is known to be related to the stringent control of ribosomal protein gene transcription in *S. cerevisiae* (Moehle and Hinnebusch 1991). Some of

the patterns found in the upstream regions of genes in the clusters *C*(5,2,6)(00000-1-2) (27 sequences) and *C*(5,3,6)(00000-1-2) (25 sequences) match substrings in the sites for REB1 and BAF1 proteins, that are repressors (Diffley and Stillman 1988; Wang et al. 1990), which corresponds well with the fact that the clusters contain genes with decreasing expression level.

The complete list of patterns discovered from the clusters is available on the worldwide web.

DISCUSSION

The results of the analysis of the complete set of gene upstream regions show that, given a genome with annotated genes, some transcription factor-binding site (or other regulatory element) descriptions can be generated without any background knowledge about the transcription factors of the organism. As far as we know, these are the first results from an automatic method for the discovery of possible regulatory elements applied to a complete genome, when no background information about transcription factors is used. Additionally, we have used expression level data to find groups of genes with similar expression profiles and searched for patterns common in the upstream regions of genes in each cluster by a fully automatic and rigorous method. Earlier work by other authors includes methods for finding transcription factor binding sites from smaller sets of sequences upstream from coregulated genes taken from the literature and methods for clustering smaller sets of genes by expression patterns. Also, combinations of transcription factor binding sites that tend to occur together have been studied in (Brāzma et al. 1997; Wagner 1998).

A survey of available software for transcription factor-binding site inference from sets of sequences is given in (Frech et al. 1997). The algorithms have been tested on sets of about 20 sequences all known to contain a common motif, such as TATA or CCAAT boxes. One of the first and still most used methods is based on the information content or expectation maximization for constructing binding site profiles (Schneider et al. 1986; Stormo and Hartzell 1989; Cardon and Stormo 1992). Profiles are generally better representations of binding sites than regular expression type patterns because they can give more accurate descriptions. At the same time, it is more difficult to discover profiles from data sets that include sequences that may not contain the specific binding site, because profiles easily can incorporate this noise in the model.

A recent algorithm for finding binding-site descriptions was published by Wolfertstetter et al. (1996). This algorithm is based on searching for all the substrings of fixed length (the default length is 7) that match at least the given portion (the default value 90%) of the given sequences, possibly with one mismatch. Examples are given when the binding-site descriptions have been discovered from sets of about 20 sequences almost all known to contain the respective binding site. An anchored alignment-based method that can be used for discovering binding site descriptions was also developed (Frech et al. 1993; Quandt et al. 1995). In a more recent work, van Helden et al. (1998) have developed a new algorithm based on finding *n*-tuples that are over-represented in the set of given upstream regions, in comparison to some precalculated background distribution. By applying the algorithm to sets of yeast gene upstream regions that are known to be coregulated [from their roles in the yeast metabolic pathway and also using gene expression studies of (DeRisi et al. 1997)], they found the binding sites of the transcription factors involved in the regulation of these genes. On the other hand, they also noted that in some cases substring patterns are not sufficient for describing some binding sites known to be present in these upstream regions.

Clustering of genes by their expression pattern similarity has been done previously (Michaels et al. 1998; Wen et al. 1989) for 112 genes from rat spinal cord. The genes were grouped in five clusters, and it was shown that this clustering correlates with the gene functional classes. In their yeast gene expression studies, DeRisi et al. (1997) selected several small groups of approximately five to six genes with similar expression profiles and showed that they belong to similar regulatory pathways and some contain binding sites for relevant transcription factors.

Two principal differences between earlier approaches and ours are the size of the data sets and the uncertainty about the presence of real patterns in the data. We do not assume any a priori knowledge about how many of the sequences share common patterns (there may be only 10–20 such sequences from 6000). Dealing with larger and noisier data sets was possible because of the development of a new, more powerful pattern discovery algorithm. At the same time, it should be noted that because of the high noise level in our data set, we cannot infer precise regulatory element descriptions, but rather give hints about the descriptions and locations of such elements in the genome. Our approach is essentially a data mining approach: We use systematic, algorithmic methods with as little human in-

tervention and as few informal steps as possible, to completely automatically obtain hints about regulatory elements in the genome in presence of rather noisy data.

The fact that patterns that are very similar to known yeast transcription factor-binding sites can be discovered from the complete set of about 6000 gene upstream regions automatically by analysis of sequence information only was quite unexpected to us. It raises the possibility that some other discovered sequence patterns that do not have matches in TRANSFAC may be yet unknown binding sites or other meaningful signals.

We were less surprised that our results showed that known transcription factor-binding sites could be discovered automatically from upstream regions of genes clustered by similarity in their expression profiles. It shows, however, that these data are sufficiently clean to enable fully automatic clustering and subsequent sequence pattern discovery. Note that genes sharing similar expression profiles are not guaranteed to be regulated by the same transcription factors for at least two reasons: first, because of the possible noise in the expression data, and the subjectivity of the clustering procedure, and second, because even the perfect coincidence in the expression profiles does not necessarily imply the same regulation mechanism. For instance, it is possible that a cluster of genes with similar expression profiles can be further split into two or more clusters, each of which share a common binding site. Whether this is the case, is an interesting question for further research.

The results of our pattern discovery should be interpreted as possible indications of putative binding sites, the functionality of which can be ultimately confirmed only by experimental methods. Nevertheless, the comparison of the discovered pattern occurrence statistics to that in the randomly selected upstream regions suggests, that, even if these patterns do not describe binding sites, they may represent other signals important for the particular expression profiles of the respective genes.

METHODS

Preparing the Data Sets

As mentioned above, we took all sequences of length 100, 300 and 600 bp upstream to those annotated ORFs in the yeast genome database MIPS (Mewes et al. 1997) that have expression measurements reported in (DeRisi et al. 1997; this constitutes almost all ORFs in MIPS annotated as likely to be functional). Two sets of 6215 randomly chosen genomic regions were also generated. Long repeated parts in the sequences may distort pattern count statistics. Therefore, we searched for long repeated patterns and removed the sequences in which they were contained. The removed sequences contained mostly the repeats related to retrotransposons or similar upstream regions from Srp1p/Tip1p family and constituted only a small part (<1.2%) of the original set. There remained 6184 upstream sequences. We also searched the sets of randomly chosen regions for long repeated patterns, but none were found, and, therefore, no sequences were removed from these sets.

We studied the nucleotide frequencies in upstream regions and observed that the two strands have noticeably different distributions of nucleotides A and T within 300 bp from ORF (Fig. 3, plots for bases C and G are available on the World Wide Web). On the basis of this observation, we decided to treat the two strands separately in the case of sequences of length 100, while for sequences of length 300 and 600, the reverse complements were included in the analysis.

Pattern Rating

The pattern rating is based on comparing the number of matching positive sequences (e.g., upstream regions, or upstream regions from genes with a specific expression profile) and the number of matching negative sequences (e.g., random genomic regions, or regions upstream from the genes with different expression profiles). Let us denote by S_+ the set of the positive sequences and by S_- the set of the negative sequences. There are basically two different ways how we use the negative sequences: (A) S_- is the complete set of sequences from which we want to distinguish S_+; (B) S_- is a sample of the set of all negative sequences. In both cases, however, noise may be present in the sense that the split into

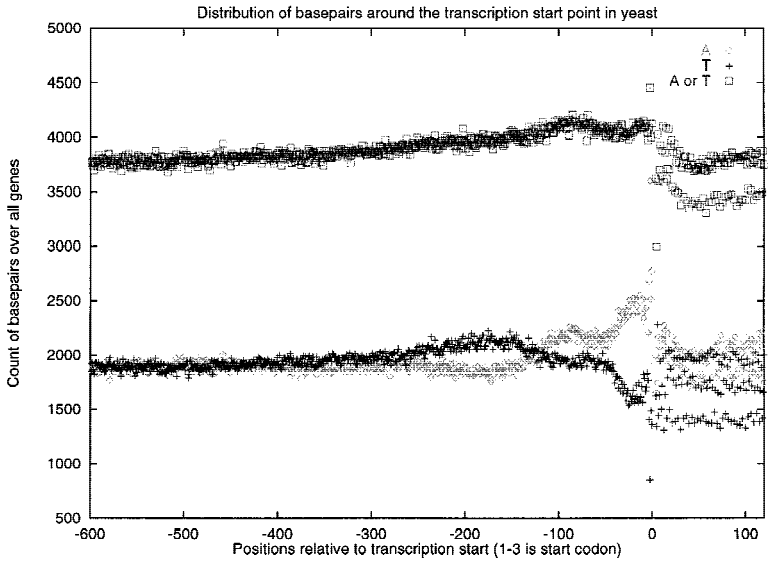

Figure 3 Distribution of bases A and T in the neighborhood of the translation start points in yeast. (◇) A; (+) T; (□) A or T. The sequences from the gene's strand are aligned on the start codon ATG at the positions 1–3.

S_+ and S_- may be imperfect. In our experiments, (A) corresponds to the discovery of patterns from gene clusters, while (B) corresponds to the discovery of patterns from the total set of upstream regions. Moreover, particularly in the case of (B), S_- may actually be a sample of a larger set of sequences (the complete genome) containing S_+ as a subset, and our goal is to find patterns specific to the sequences in S_+ (upstream regions).

Let t be some threshold, that is, a positive integer (in practice $t = 10$). If P is a pattern and $M(P)$ is the set of sequences matching P, then we can define the score as

$$R_t(P, S_+, S_-) = \begin{cases} \dfrac{|S_+ \cap M(P)|}{|S_- \cap M(P)|}, & \text{if } |S_+ \cap M(P)| \geq t \text{ and } |S_- \cap M(P)| > 0 \\ \infty, & \text{if } |S_+ \cap M(P)| \geq t \text{ and } |S_- \cap M(P)| = 0 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

A normalized version of this scoring is $R_t(P; S_+; S_-) \cdot (|S_-|/|S_+|)$. When S_- is the set of all negative sequences (case A), this tends to work well. In the case when S_- is a sample of a larger set (case B), the scoring function (1) over-rates the patterns that have very few matches in S_- (for instance if $|S_- \cap M(P)| = 0$, this scoring essentially extrapolates the fact that there are no matches in the sample set, to the whole set). Therefore, for these cases we introduce another rating function by adding a correcting constant c to $|S_- \cap M(P)|$, thus

$$R_c(P, S_+, S_-) = \frac{|S_+ \cap M(P)|}{|S_- \cap M(P)| + c} \quad (2)$$

The constant c can be chosen in different ways. We chose the value that would make the scores equal for the best discovered patterns for the -250 to -150 regions, which had respectively 0 and 1 matches in the negative set. This gave $c = 4\frac{2}{3}$.

Pattern Discovery Algorithm

For analyzing the complete set of all upstream regions and the upstream regions of gene clusters, we developed a new pattern discovery algorithm, which allows one to enumerate and produce systematically and exhaustively different kinds of patterns of unrestricted length in classes P1, P2, and P3 defined above. The program that was implemented on the basis of this algorithm is capable of analyzing large data sets (thousands of sequences) to find relatively simple patterns (substring patterns or patterns with a small number of wild cards or group characters) that occur in at least a given small number of sequences. (The data set for upstream and random sequences of length 600 bp together with their reverse complements had total size of about 16 Mb.) Alternatively, more complex patterns can be generated for smaller sets of data. Another feature of our method is that we are able to use positive and negative examples simultaneously. In our studies, the negative examples were either the random genomic sequences, or all upstream regions not belonging to the selected cluster.

Our pattern discovery algorithm is based on a data structure called suffix trie. It is a simplified but more resource demanding version of the well-known suffix tree (McCreight 1976; Ukkonen 1995). We construct a suffix trie for our set of sequences G. Our construction procedure is inspired by the lazy algorithm of (Giegerich and Kurtz 1995) for generating suffix trees. The resulting trie represents all the patterns (in the chosen class of patterns) that are present in some sequence in G. The nodes of the trie are labeled with symbols from the pattern representation alphabet: the individual nucleotides, wild cards, or character groups (all nonempty subsets of the alphabet, equal to the full IUPAC coding). The labels on the path from the root to any internal node spell out the pattern associated with the node. Thus we call the tree the pattern trie.

At each node we maintain an occurrence list that gives all the positions of input sequence where the corresponding pattern matches it. The tree is generated starting from the root. The root corresponds to the empty pattern λ whose occurrence list contains all positions of the input string. The tree is extended by generating the nodes in tree in breadth-first order, level by level. For a node n with associated pattern P, every legal extension Pc of P is generated by inserting a new child with label c for node n. The occurrence lists for each new node are computed from the occurrence list of its parent by checking for each occurrence of P in the input sequence if it can be extended to an occurrence of Pc.

If there are restrictions on the patterns to be generated, then the children of nodes not satisfying the restrictions are not generated. Depending on the pattern language and on the nature of the restrictions, by discarding these children, one can still guarantee finding all patterns satisfying the restrictions. Some of the restrictions that can be used during the pattern generation include, for example: construct only the patterns having at least t occurrences in the input sequences; construct only patterns matching at least t different input sequences; construct only the patterns that match at least t of the input sequences corresponding to positive sequences (and output those that minimize the occurrences in negative sequences).

Variations of this conceptually simple pattern generation algorithm are possible. For instance, one needs to decide in which order the nodes are processed. If all nodes at level k are processed before the nodes at level k+1, then information about which patterns at level k fulfill the requirements can be used to decide which nodes to explore at level k+1. For example, for the pattern ATCG to have at least t occurrences, both patterns ATC and TCG need to have at least t occurrences. Alternatively, one can detect nodes with identical occurrence lists (where each occurrence of a pattern is represented by its end position). Observing that the subtrees below these nodes will be identical, one can avoid duplicate work. We implemented this latter method by using a priority queue so that the patterns are studied in the order of their occurrence frequencies (i.e., size of the occurrence lists) and by using a tailor-made data structure for rapidly finding identical occurrence lists.

We considered patterns of the form $P = A_1A_2, \ldots, A_p$, where $A_1, \ldots, A_p$ are nonempty subsets of symbols from the alphabet $\Sigma = \{A,T,G,C\}$. In the experiments in which we analyzed the distribution of all patterns in upstream versus random sequences, we used the substring patterns, patterns with at most one wild-card symbol, and patterns with one possible group symbol. We recorded exhaustively all occurrences of all patterns in chosen pattern classes to be able to analyze the distribution of the patterns.

In the experiments in which we compared the patterns in upstream regions for clusters of similarly behaving genes to all other upstream regions, we allowed the patterns to contain single characters and up to three wild-card positions. We re-

quired every pattern to match at least half of the upstream regions in the cluster.

The time and space requirement of our algorithm for generating all different substrings from n characters long input strings is $O(n^2)$. By restricting ourselves to patterns occurring at least t times we can speed up the algorithm in practice. The larger is t, the faster is the algorithm. For more complex pattern classes with many wild cards and group characters the complexity of the algorithm can grow exponentially as the number of patterns matching the required number of sequences tends to grow fast. We chose the different pattern classes for different size data sets so that the exhaustive pattern generation algorithms would take approximately one, or at most a few hours to complete on Sun and DEC workstations with 512 MB of memory. The space requirement, that is, the memory used for storing all the possible patterns from the pattern class, was the most important practical limitation, as we had to store all the patterns including the ''unnecessary'' ones. This might be avoided by studying and reporting only the most promising patterns. The algorithm is described in detail in Vilo (1998) and the software is available at http://www.cs.Helsinki.FI/~vilo/bio/.

Clustering of the Expression Data

The measurements of DeRisi et al. (1997) were used for clustering genes on the basis of their expression rates r_i (the relative fluorescence intensity as compared to the reference sample) at seven consecutive time points $i = 1, \ldots, 7$ at 2-hr intervals. We observed that the expression levels of almost all genes varied at most within a factor of 10.

Next, we developed a method to divide the genes into clusters on the basis of the shape of the time-series $r_1, \ldots, r_7$ that is formed by the expression levels associated with the gene. Genes with similar shape should be classified in the same clusters. In DeRisi et al. (1997) it was observed that in repeated measurements, the expression levels stayed the same within a factor of 2 in 95% of the genes. This observation indicates that the classification of the time-series shapes is meaningful but it should not be too fine.

We classified the shapes by replacing each rate r_i (which is a real number $\geqslant 0$) with a discrete value representing an interval of reals that contains r_i. For symmetry reasons, we first replaced each r_i with its base-2 logarithm $d_i = \log r_i$. Values d_i can be positive and negative reals. Then we fixed a division of the real axis into a small number of intervals, symmetric to the origin. We used three or five intervals defined by one or two thresholds as follows. By fixing one threshold h we defined intervals $(-\infty, -\log h]$, $(-\log h, \log h)$; and $[\log h, +\text{infinity})$. The discrete characters (the names of the intervals) used to represent these intervals are -1, 0, and 1 respectively. The discretized presentation of the time series $r_1, \ldots, r_7$ is now obtained by replacing each r_i by the name a_i of the interval that contains d_i. This gives sequence $a_1, \ldots, a_7$ in three-character alphabet. Similarly, by fixing two thresholds h_1 and h_2 we defined five intervals $(-\infty, -\log h_2]$, $(-\log h_2, -\log h_1]$, $(-\log h_1, \log h_1)$, $[-\log h_1, \log [h_2)$, and $[\log h_2, +\infty)$, with respective names -2; -1; 0; 1; and 2. The corresponding discretized version $a_1, \ldots, a_7$ of $r_1, \ldots, r_7$ is a sequence in five-character alphabet.

Now we can complete the definition of our clustering scheme: the genes that have the same discretized sequence $a_1, \ldots, a_7$, for fixed thresholds, belong to the same cluster. We denote individual clusters as $C(3,h)(a_1, \ldots, a_7)$, or $C(5,h_1,h_2)(a_1, \ldots, a_7)$, where the first part defines the intervals for discretization and the second part the particular cluster. An example of a time series that belongs to cluster $C(5,4,8)(0000120)$ is shown in Figure 4.

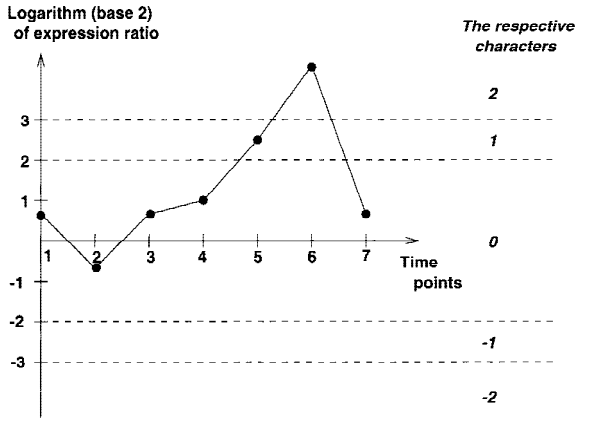

Figure 4 Discretizing the continuous measurement space. An example of time series that belongs to cluster C(5; 4; 8)(0000120).

Besides the discretization described above, we used similar discretizing technique for relativized version of the time series; that is, we defined six differences $d'_i = d_i + 1 - \log(r_i + 1 = r_i)$ and then defined the clusters by $C_r(n, h_1, \ldots, h_{\lfloor n/2 \rfloor})(a_1, \ldots, a_6)$, where each a_i is computed from d'_i instead of d_i. The reason for such clustering is that it can be hypothesized, that not the amount of the gene expression product, but the rate of the gene expression change is affected directly.

In this way, we transformed the time series of the measurements into words of length seven or six over the alphabet of three or five characters. We used 10 different ways to discretize the measurement space by choosing different combinations of threshold values from 2, 3, through 8. For the original time-series we used five intervals, and for the relativized time series three and five intervals. For finding the resulting gene clusters we implemented a program that enumerated all possible sequences $a_1, \ldots, a_7$ and found which ORFs have expression profiles that map into each sequence. We selected the clusters with between 10 and 100 ORFs. In the case of five intervals, we selected the clusters with expression profiles containing at least one symbol -2 or 2 and thus showing noticeable variation during the diauxic shift. Later we selected from them only the clusters with at least 25 sequences, because for these clusters the discovered pattern ranking consistently differed from the patterns in random sets of the same size. The resulting clusters of genes with similar expression profiles are summarized in Table 3.

ACKNOWLEDGMENTS

We thank A. Thanaraj for carefully reading the manuscript and valuable comments. The authors benefited substantially from discussions with J. ColladoVides, R. Durbin, J. van Helden, A. Robinson, and G. Stormo. J. DeRisi was helpful in providing and discussing the gene expression data. We used a data mining tool Decisionhouse from Quadstone Ltd. for exploring the gene expression data, and valuable advice was provided by A. Ewing and N. Skillings. The authors also wish

to thank anonymous referees for valuable suggestions. A.B. was supported by the BIOVIS and BioStandards project from the European Union. I.J. has been supported by grant 111032/410 from the Norwegian Research Council. J.V. was supported by grant 8745 from the Academy of Finland.

The publication costs of this article were defrayed in part by payment of page charges. This article must therefore be hereby marked "advertisement" in accordance with 18 USC section 1734 solely to indicate this fact.

REFERENCES

Brāzma, A., J. Vilo, E. Ukkonen, and K. Valtonen. 1997. Data mining for regulatory elements in yeast genome. In *Proceedings of Fifth International Conference on Intelligent Systems for Molecular Biology*, pp. 65–74. AAAI Press, Menlo Park, CA.

Brāzma, A., I. Jonassen, I. Eidhammer, and D. Gilbert. 1998a. Approaches to automatic discovery of patterns in biosequences. *J. Comp. Biol.* 5 277–304.

Brāzma, A., I. Jonassen, J. Vilo, and E. Ukkonen. 1998b. Pattern discovery in biosequences. In *Proceedings of the Fourth International Colloquium on Grammar Inference, Lecture Notes in Artificial Intelligence*, vol. 1433, pp. 255–270, Springer, New York, NY.

Bucher, P. 1990. Weight matrix description of four eukaryotic RNA polymerase II promoter elements derived from 502 unrelated promoter sequences. *J. Mol. Biol.* 212: 563–578.

Cardon, L.R. and G.D. Stormo. 1992. Expectation maximization algorithm for identifying protein-binding sites with variable lengths from unaligned DNA fragments. *J. Mol. Biol.* 223: 159–170.

Chen, Q.K., G.Z. Hertz, and G.D. Stormo. 1995. MATRIX SEARCH 1.0: A computer program that scans DNA sequences for transcriptional elements using a database of weight matrices. *Comput. Appl. Biosci.* 11: 563–566.

Corhish-Bowden, A. 1984. Nomenclature for incompletely specified bases in nucleic acid sequence: Recommendations 1984. *Nucleic Acids Res.* 13:3021–3030.

DeRisi, J.L., V.R. Iyer, and P.O. Brown. 1997. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science* 278: 680–686.

Diffley, J.F. and B. Stillman. 1988. Purification of a yeast protein that binds to origins of DNA replication and a transcriptional silencer. *Proc. Natl. Acad. Sci.* 85: 2120–2124.

Frech, K., G. Herrmann, and T. Werner. 1993. Computer-assisted prediction, classification, and delimitation of protein binding sites in nucleic acids. *Nucleic Acids Res.* 21: 1655–1664.

Frech, K., K. Quandt, and T. Werner. 1997. Software for the analysis of DNA sequence elements of transcription. *Comput. Appl. Biosci.* 13: 89–97.

Ghosh, D. 1990. A relational database of transcriptional factors. *Nucleic Acids Res.* 18: 1749–1756.

Giegerich, R. and S. Kurtz. 1995. A comparison of imperative and purely functional suffix tree constructions. *Sci. Comput. Program.* 25 187–218.

Goffeau, A., B.G. Barrell, H. Bussey, R.W. Davis, B. Dujon, H. Feldmann, F. Galibert, J.D. Hoheisel, C. Jacq, M. Johnston, E.J. Louis, H.W. Mewes, Y. Murakami, P. Philippsen, H. Tettelin, and S.G. Oliver. 1996. Life with 6000 genes. *Science* 274: 546–567.

Jonassen, I. 1997. Efficient discovery of conserved patterns using a pattern graph. *Comput. Appl. Biosci.* 13: 509–522.

McCreight, E.M. 1976. A space-economical suffix tree construction algorithm. *J. Assoc. Comput. Mach.* 23: 262–272.

Mellor, J. 1993. Multiple interactions control the expression of yeast genes. In *The eukaryotic genome, organisation and regulation* (ed. P. Broda, S.G. Oliver, and P.F.G. Sims), pp. 275–320. Cambridge University Press, Cambridge, UK.

Mewes, H.W., K. Albermann, M. B-ahr, D. Frishman, A. Gleissner, J. Hani, K. Heumann, K. Kleine, A. Maierl, S.G. Oliver, F. Pfeiffer, and A. Zollner. 1997. Overview of the yeast genome. *Nature* (Suppl) 387: 7–65.

Michaels, G.S., D.B. Carr, M. Askenazi, S. Fuhrman, X. Wen, and R. Somogyi. 1998. Cluster analysis and data visualization of large scale gene expression data. In *Proceedings of Pacific Symposium on Biocomputing '98*, pp. 42–53. World Scientific, Singapore.

Moehle, C.M. and A.G. Hinnebusch. 1991. Association of rap1 binding sites with stringent control of ribosomal protein gene transcription in Saccharomyces cerevisiae. *Mol. Cell Biol.* 11: 2723–2735.

Quandt, K., K. Frech, H. Karas, E. Wingender, and T. Werner. 1995. MatInd and MatInspector: New fast and versatile tools for detection of consensus matches in nucleotide sequence data. *Nucleic Acids Res.*23: 4878–4884.

Ramsay, G. 1998. DNA chips: State-of-the-art. *Nature Biotechnol.* 16: 40–44.

Schena, M., D. Shalon, R.W. Davis, and P.O. Brown. 1995. Quantitative monitoring of gene expression patterns with a complimentary DNA microarray. *Science* 270: 467–470.

Schneider, T.D., A. Erenfeucht, G.D. Stormo, and L. Gold. 1986. Information content of binding sites on nucleotide sequences. *J. Mol. Biol.* 188: 415–431.

Stormo, G. and G.W. Hartzell III. 1989. Identifying protein-binding sites from unaligned DNA fragments. *Proc. Natl. Acad. Sci.* 86: 1183–1187.

Ukkonen, E. 1995. On-line construction of suffix trees. *Algorithmica* 14: 249–260.

van Helden, J., B. André, and J. Collado-Vides. 1998. Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *J. Mol. Biol.* 281 827–842.

Velculescu, V.E., L. Zhang, W. Zhou, J. Vogelstein, M.A. Basarai, D.E. Bassett Jr., P. Hieter, B. Vogelstein, and K.W. Kinzler. 1997. Characterization of the yeast transcriptosome. *Cell* 88: 243–251.

Vilo, J. 1998. Discovering frequent patterns from strings. Technical Report C-1998-9, Department of Computer Science, University of Helsinki, Finland.

Wagner, A. 1998. Distribution of transcription factor binding sites in the yeast genome suggests abundance of coordinately regulated genes. *Genomics* 50: 293–295.

Wang, H., P.R. Nicholson, and D.J. Stillman. 1990. Identification of a Saccharomyces cerevisiae DNA-binding protein involved in transcriptional regulation. *Mol. Cell Biol.* 10: 1743–1753.

Wen, X., S. Fuhrman, G.S. Michaels, D.B. Carr, S. Smith, J.L. Bakerand, and R. Somogyi. 1989. Identifying protein-binding sites from unaligned DNA fragments. *Proc. Natl. Acad. Sci.* 86: 1183–1187.

Wingender, E., P. Dietze, H. Karas, and R. Knuppel. 1996. TRANSFAC: A database of transcriptional factors and their DNA binding sites. *Nucleic Acids Res.* 24: 238–241.

Wodicka, L., H. Dong, M. Mittmann, M.-H. Ho, and D.J. Lockhart. 1997. Genome-wide expression monitoring in Saccharomyces cerevisiae. *Nat. Biotechnol.* 15: 1359–1367.

Wolfertstetter, F., K. Frech, G. Herrmann, and T. Werner. 1996. Identification of functional elements in unaligned nucleic acid sequences by a novel tuple search algorithm. *Comput. Appl. Biosci.* 12: 71–80.

Yada, T., Y. Totoki, M. Ishikawa, K. Asai, and K. Nakai. 1998. Automatic extraction of motifs represented in the hidden Markov model from a number of DNA sequences. *Bioinformatics* 14: 317–325.

Received July 6, 1998; accepted in revised form September 18, 1998.