OPTIMAL ALGORITHM FOR FINDING DNA MOTIFS WITH NUCLEOTIDE ADJACENT DEPENDENCY*

FRANCIS Y.L. CHIN HENRY C.M. LEUNG M.H. SIU S.M. YIU
Department of Computer Science, University of Hong Kong, Pokfulam
Hong Kong

Abstract: Finding motifs and the corresponding binding sites is a critical and challenging problem in studying the process of gene expression. String and matrix representations are two popular models to represent a motif. However, both representations share an important weakness by assuming that the occurrence of a nucleotide in a binding site is independent of other nucleotides. More complicated representations, such as HMM or regular expression, exist that can capture the nucleotide dependency. Unfortunately, these models are not practical (with too many parameters and require many known binding sites). Recently, Chin and Leung introduced the SPSP representation which overcomes the limitations of these complicated models. However, discovering novel motifs in SPSP representation is still a NP-hard problem. In this paper, based on our observations in real binding sites, we propose a simpler model, the Dependency Pattern Sets (DPS) representation, which is simpler than the SPSP model but can still capture the nucleotide dependency. We develop a branch and bound algorithm (DPS-Finder) for finding optimal DPS motifs. Experimental results show that DPS-Finder can discover a length-10 motif from 22 length-500 DNA sequences within a few minutes and the DPS representation has a similar performance as SPSP representation.

1 Introduction

A *gene* is a segment of DNA that can be decoded to produce functional products like protein. To trigger the decoding process, a molecule, called *transcription factor*, will bind to a short region (*binding site*) preceding the gene. One kind of transcription factor can bind to more than one binding site. These binding sites usually have similar patterns and are collectively represented by a *motif*. Finding motifs and the corresponding binding sites from a set of DNA sequences is a critical step for understanding how genes work.

There are two popular models to represent a motif, string representation [4,6,10,11,16,17,19-22] and matrix representation [2,8,12-14]. String representation uses a length-l string of symbols (or nucleotides) 'A', 'C', 'G' and 'T' to represent a motif of length l. To improve the descriptive power of the representation, IUPAC symbols [6,20,22] can be introduced into the string to represent choices of symbols at a particular position (e.g. 'K' denotes 'G' or 'T'). Matrix representation further improves the descriptive power by using position weight matrices (PWMs) or position specific scoring matrices (PSSMs) to represent a motif. PWMs and PSSMs are matrices of size $4 \times l$ with the j-th column, which has four elements corresponding to the four nucleotides, effectively giving the occurrence probability of each of the four nucleotides at position j. While the matrix representation model appears superior, the solution space for PWMs

* The research was supported in parts by the RGC grant HKU 7120/06E.

and PSSMs is huge, which consists of $4l$ real numbers, and thus, algorithms generally either produce a sub-optimal motif matrix [2,8,12,13] or take too long to run when the motif is longer than 10 [15].

However, both the string and the matrix representations share an important common weakness: they assume that the occurrence of each nucleotide at a particular position of a binding site is independent of the occurrence of nucleotides at other positions. This assumption may not represent the actual situation. According to the analysis of wild-type and mutant Zif268 (Egr1) zinc fingers by Bulyk et al [5], it gives compelling evidence that nucleotides of transcription factor binding sites should not be treated independently, and a more realistic motif model should be able to describe nucleotide interdependence. Man and Stormo [18] have arrived at a similar conclusion in their analysis of Salmonella bacteriophage repressor Mnt: they found that interactions of Mnt with nucleotides at positions 16 and 17 of the 21 bp binding site are in fact not independent.

When there are sufficient number of *known* binding sites of a transcription factor, people can use some complex representations, e.g. the hidden Markov model (HMM) [24], Bayesian network [3] or enhanced PWM [9], to represent nucleotide interdependence. However, when we want to discover novel motif or describe a motif with only a few known binding sites, the input data may not contain enough information for deriving the hidden motif. Chin and Leung overcame the problem by introducing the SPSP representation [7], a generalized model of string representation and matrix representation, that can model the adjacent dependency of nucleotides with much less parameters than HMM and regular expression. Since the SPSP representation is simple, it can be used to discover novel motifs even if there are only five DNA sequences containing the binding sites of the transcription factor. However, like other models, discovering novel motifs in SPSP representation is a NP-hard problem. No efficient algorithm exists that can guarantee finding the hidden motif in reasonable amount of time.

After studying the binding sites of real biological data, we found that many motifs can be described by a simpler model. In this paper, we further simplify the SPSP representation to the Dependency Pattern Sets (DPS) representation. DPS representation is a generalized model of string representation, which can model adjacent nucleotide dependency. Although it has a lower descriptive power than SPSP representation, experimental results on real biological data showed that it has almost the same performance as SPSP representation. Besides, since DPS representation uses fewer parameters to describe a motif, it is possible to find the "optimal" motif in reasonable amount of time. We have introduced a branch and bound algorithm DPS-Finder that guarantees finding the "optimal" motif. In practice, DPS-Finder takes only a few minutes to discover a length-10 motif from 20 length-600 DNA sequences. For other approaches such as HMM, it may take hours or even days for a dataset of similar size.

This paper is organized as follows. In Section 2, we describe the DPS representation and the scoring function for determine the "optimal" motif in a set of DNA sequences. We introduce the branch and bound algorithm DPS-Finder in Section 3. Experimental

results on real biological data comparing DPS-Finder with some popular software are given in Section 4, followed by concluding remarks in Section 5.

2 Problem Definition

2.1 DPS Representation

Motif is an abstract model for a set of binding sites with similar patterns. For example, the transcription factor CSRE [25], which activates the gluconeogenic structural genes, can bind to the following binding sites.

CGGA**T**G**A**A**T**GG
CGGA**T**G**A**A**T**GG
CGGA**T**G**AA**GG
CGGA**C**G**G**A**T**GG
CGGA**C**G**G**A**T**GG

Note that there is dependence between the fifth and the seventh symbols, and the binding sites "CGGATGAATGG" occurs twice in the DNA sequences. The string representation models these binding sites by the length-11 string "CGGA**Y**G**R**A**W**GG" where 'Y' denotes 'T' or 'C', 'R' denotes 'A' or 'G' and 'W' denotes 'A' or 'T'. However, this representation has a problem that the strings "CGGA**T**G**G**A**T**GG", "CGGA**T**G**G**A**A**GG", "CGGA**C**G**A**A**T**GG", "CGGA**C**G**A**A**A**GG" and "CGGA**C**G**G**A**A**GG" are also considered as binding sites (false positives). Instead of modeling the CSRE motif by one string, the SPSP representation uses a pattern P and a set of score S (negative of logarithm of the occurrence probability) to represent the CSRE motif as follows.

$$P = (\text{CGGA})\begin{pmatrix}\text{TGA}\\ \text{CGG}\end{pmatrix}(\text{A})\begin{pmatrix}\text{A}\\ \text{T}\end{pmatrix}(\text{GG}) \text{ and}$$

$$S = (-\log(1))\begin{pmatrix}-\log(0.6)\\ -\log(0.4)\end{pmatrix}(-\log(1))\begin{pmatrix}-\log(0.2)\\ -\log(0.8)\end{pmatrix}(-\log(1))$$

A length-11 string is considered as a binding site of CSRE if it matches with P and its score (sum of corresponding entries) is at most some threshold, say 3.1. For example, the score of the binding site "CGGA**TGA**A**T**GG" is -log(1)+ -log(0.6) + -log(1) + -log(0.8) + -log(1) = 1.05 < 3.1. The score of a non-binding site string "CGGA**CGG**A**A**GG" is -log(1)+ -log(0.4) + -log(1) + -log(0.2) + -log(1) = 3.6 > 3.1. The string "**T**GGA**TGA**A**T**GG" does not match with P, so it is not a binding site. In this example, the SPSP representation can model the motif with no false positive.

Although SPSP representation can describe the motif well, it is difficult to determine the score S for novel motifs (motifs with no known binding site) in real biological data. A challenge is to have a simpler model, which describes real motifs using fewer parameters than the SPSP representation while having fewer false positives than string

representation. We observed that using only the pattern P without S, we already can describe most real motifs. For example, if we consider those strings matching with P as binding sites, we only have one false positive "CGGA**CGG**AA**A**GG" (instead of five for the string representation).

Apart from this, SPSP representation allows a motif having any number of wildcard pattern sets (positions with more than one possible choice of patterns, i.e. brackets with more than one pattern in it). For example, the following pattern P is allowed.

$$P = (\mathrm{C})\begin{pmatrix}\mathrm{GG}\\ \mathrm{GT}\\ \mathrm{TT}\end{pmatrix}(\mathrm{A})\begin{pmatrix}\mathrm{T}\\ \mathrm{C}\end{pmatrix}(\mathrm{G})\begin{pmatrix}\mathrm{A}\\ \mathrm{G}\end{pmatrix}(\mathrm{A})\begin{pmatrix}\mathrm{A}\\ \mathrm{T}\end{pmatrix}(\mathrm{GG})$$

Since the binding sites of a motif should be conserved in most positions, the number of wildcard pattern sets should be small. We found that allowing at most two wildcard pattern sets is enough for describing most motifs. Based on the above observations, we define the *Dependency Pattern Sets* (DPS) representation as follows.

> A DPS representation P contains a list of patterns sets P_i, $1 \le i \le L$, where at most **two** are wildcard pattern sets P_i containing 2 to k length-l_i patterns $P_{i,j}$ of symbols 'A', 'C', 'G' and 'T', $l_i \le l_{max}$ where the Hamming distance between these patterns is at most $d_{\max}$. Each of the other pattern set P_i contains **exactly one** length-l_i pattern $P_{i,1}$ and $\sum_i l_i = l$. A length-l string $\sigma = \sigma_1\sigma_2\ldots\sigma_L$ where $|\sigma_i| = l_i$ is considered as a binding site of P if $\sigma_i \in P_i$, $1 \le i \le L$.

2.2 Scoring Function and Problem Definition

Given a set of DNA sequences T with X length-l substrings bound by the same transcription factor, we should find many candidate motifs having different number of binding sites in T. In order to discover the hidden motif, we should have a scoring function for comparing different motifs. Given two motifs P_1 and P_2, a naive scoring function is to count the number of binding sites represented by the motifs, that is, P_1 is more likely to be the hidden motif if P_1 have more binding sites than P_2 in the set of sequences T. However, this scoring function has a weakness that it has not considered the number of possible binding sites for P_1 and P_2. Consider the following motifs.

$$P_1 = (\mathrm{C})\begin{pmatrix}\mathrm{AT}\\ \mathrm{CG}\\ \mathrm{GT}\end{pmatrix}(\mathrm{CC})\begin{pmatrix}\mathrm{A}\\ \mathrm{T}\end{pmatrix}(\mathrm{TC}) \text{ and } P_2 = (\mathrm{ACG})\begin{pmatrix}\mathrm{AG}\\ \mathrm{CT}\end{pmatrix}(\mathrm{AAA})$$

Even P_1 has slightly more binding sites than P_2, we cannot conclude that P_1 is more likely to be the hidden motif because P_1 has more possible binding site patterns ($3 \times 2 = 6$ patterns) than P_2 (2 patterns). In order to have a fair comparison, given a motif P with b binding sites in T, we calculate the probability (p-value) that P has b or more binding sites in T by chance based on a background model. Under the assumption that the hidden motif should have an unexpectedly large number of binding sites, a motif P with small p-

value is likely to be the hidden motif. The *p*-value of a motif can be calculated as follows [7].

Let B be the background model for the non-binding region of the DNA sequences T and $B(\sigma)$ be the probability that a length-l string σ occurs in a particular position in T. B can be a Markov Chain or an uniform distribution etc. Given a DPS motif P with w possible binding sites $s_1, s_2, \ldots, s_w$, the probability that P has a binding site at a particular position in T is $\sum_{i=1}^{w} B(s_i)$. Assuming the probability that motif P has a binding site at any positions in T are independent, the probability that P has b or more binding sites in T is

$$p-\text{value}(P) = \sum_{j=b}^{X} \left[\binom{X}{j} \left(\sum_{i=1}^{w} B(s_i) \right)^{j} \left(1 - \sum_{i=1}^{w} B(s_i) \right)^{X-j} \right] \quad (1)$$

Based on the scoring function in Eq(1), we define the motif discovering problem as follows.

> Given a set of DNA sequences T, the background model B and the motif length l, we want to discover a length-l DSP motif P with the minimum *p*-value.

3 DPS-Finder Algorithm

In this section, we introduce the DPS-Finder Algorithm for solving the motif discovery problem described in Section 2. DPS-Finder Algorithm first constructs a l-factor tree [1], a suffix tree with all nodes of depth > l being removed, to represent all possible motifs in the input sequences T with different positions of the wildcard pattern sets. For each possible motif P, it finds the set of patterns in each wildcard pattern set that minimizes *p*-value(P) using a branch and bound approach. Experiments showed that DPS-Finder Algorithm has to deal in the best case only 25% of the number of cases to be dealt by the brute force algorithm.

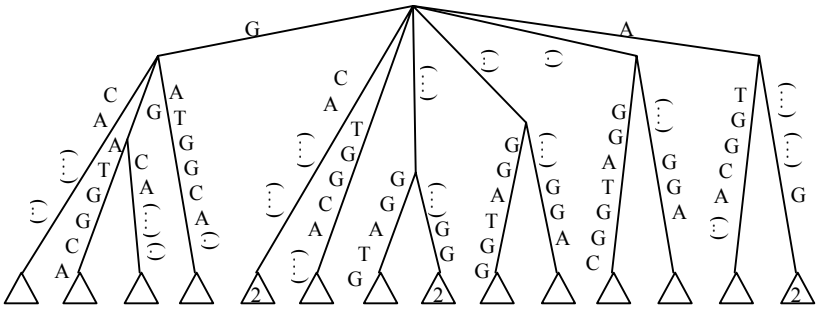

Figure 1. The 8-factor tree of the sequences "CA(…)(…)GGATGGCA(…)(…)GG". For examples, the pattern "(CA)(…)(…)", "(A)(…)(…)(G)" and "(…)(…)(GG)" occur twice in the sequences.

3.1 Factor Tree Representation

In order to discover the optimal motif, we should consider all possible positions ($C(2 + l - 2l_{max}, 2) = O(l^2)$) of the wildcard pattern sets. For example, when the motif length l is 8 and the maximum wildcard pattern length l_{max} is 3, the length-8 substring "CGCAGGTG" (binding site of the AC transcription factor) can be a binding site of motifs in the following formats, (…)(…)(TG), (…)(A)(…)(G), (…)(AG)(…), (C)(…)(…)(G), (C)(…)(G)(…) and (CG)(…)(…), where (…) represents a wildcard pattern set of length 3. Note that motifs with wildcard pattern shorter than 3 or with one wildcard pattern set only have also been considered in the above formats. For example, (…)(AGG)(..) and (…)AGGTG are special case of the motif (…)(AG)(…). When we find the optimal motif in form of (…)(AG)(…), we have also considered motifs in form of (…)(AGG)(..) and (…)AGGTG. Since it takes O(l) time to convert a length-l substring to a motif and there are X length-l substrings in T, brute force method takes $O(Xl^3)$ time to get the list of $O(Xl^2)$ possible forms of motif.

However, when a motif of a substring is considered, we can easily get another motif for the adjacent substring by shifting one symbol. For example, when the motif (CG)(…)(…) of the substring "CGCAGGTG" in the input sequence "…CA**CGCAGGTG**GG…" is considered, by shifting one symbol, we will get another motif (G)(…)(…)(G) for the substring "GCAGGTGG". When we represent the input sequence in the form of "…CA**CGCAGGTG**GG…", each length-8 sliding window containing the two length-3 brackets represents one possible motif. Based on this observation, DPS-Finder Algorithm constructs a generalized l-factor tree [1] of $O(l^2)$ (represent the $O(l^2)$ motifs for a length-l substring) length-$O(X)$ sequences (input sequences with some positions represented by brackets) to represent the $O(Xl^2)$ possible motifs. A l-factor tree is a suffix tree [23] of height l where each path from the root to a leaf represents a length-l substring occurring in the input sequence. Figure 1 shows a factor tree of height-8 for the sequence "CA(…)(…)GGATGGCA(…)(…)GG". Since constructing the generalized l-factor tree takes $O(Xl^2)$ time [1] only, DPS-Finder Algorithm speeds up the process by a factor of $O(l)$ when compares with the brute force algorithm.

3.2 Branch and Bound Approach

Each leaf of the l-factor tree represents a candidate motif. These candidate motifs may not be in DPS representation because they may have more than k patterns in their wildcard pattern sets. Therefore, giving a candidate motif P, we have to reduce the number of patterns in each of its wildcard pattern set to at most k and at the same time, to minimize the p-value. Although this problem is NP-hard when the value of k is large (see Appendix), in practice we usually consider motifs with small k (e.g. $k = 4$) and finding the optimal motif is still feasible.

When refining a candidate motif P to a motif P' in DSP representation with the minimum p-value(P'), we perform a depth-first-search to check all possible combinations

of patterns in the two wildcard pattern sets of P. We first pick two patterns, each forms a wildcard pattern set of P. Then we pick more patterns for P' until k patterns have been selected for each wildcard pattern set. In the selection process, we consider patterns with increasing order of p-values. After picking a new pattern P_i, the additional number of binding sites covered by P' is upper bounded by the number of binding sites covered by P_i. Therefore, in many cases, we can stop picking new patterns because the p-value of the refined motif P' must not be smaller than the suboptimal motif we have already found.

Apart from applying a branch and bound approach on refining each candidate motif P, we also apply similar approach on checking the $O(Xl^2)$ candidate motifs. We first refine those candidate motifs with two patterns, each forms a wildcard pattern set, covering the largest number of binding sites. Since the number of binding sites covered by a candidate motif is upper bounded by the total number of binding sites covered by the top-k patterns in its wildcard pattern sets, many candidate motifs can be pruned out.

4 Experimental Results

We compared the performance of some popular motif discovering algorithms, i.e. Weeder [19], MEME [13] and SPSP-Finder [7], with DSP-Finder on the yeast data set in SCPD [25]. SCPD contains information of the motif patterns and the binding sites for a set of transcription factors of yeast. For each transcription factor, we chose the 600 base pairs in the upstream of those genes bound by the transcription factor as the input sequences T. Given the motif length, the four algorithms were used to discover the hidden motif in T.

Weeder and MEME used string representation and matrix representation to model a motif respectively. Both of them could not model the nucleotide dependency in motifs. SPSP-Finder, used the SPSP representation, can model the nucleotide dependency in motifs. However, all these algorithms applied a heuristic approach which cannot guarantee finding the "optimal" motifs.

In the experiments, DSP-finder used an order-0 Markov chain calculated based on the input sequence to model the non-binding regions. The width of a wildcard pattern set was at most 3 ($l_{\max} = 3$), the Hamming distance between patterns in a wildcard pattern set was at most 1 ($d = 1$) and there were at most 4 patterns in a wildcard pattern set. The experimental results were shown in Table 1. All algorithms finished in 10 minutes for each dataset. Note that we have not listed out those motifs which could not be discovered by any of the algorithms.

In general, SPSP-Finder and DSP-Finder has better performance than the other algorithms because they can model nucleotide dependency. DSP-Finder performs better than SPSP-Finder when finding motif of MCM1 because DSP-Finder guarantees finding the motif with the lowest p-value while SPSP-Finder is trapped in local minimum.

DSP-Finder performs worse than MEME and SPSP-Finder in two cases, the HAP2/3/4 and SFF datasets. For the HAP2/3/4 dataset, there was nucleotide dependency between the fifth and the sixth nucleotides. However, since the Hamming distance

between the possible patterns is 2, DSP-Finder could not discover the motif in our setting (d = 1). DSP-Finder could not discover the motif of SFF while MEME was successful because there were no strong bias at most positions of this motif. In these cases, a matrix representation can model the motif better than a string representation, i.e. Weeder also fails in this case.

Table 1. Experimental results on yeast data.

Name	Pattern	Weeder	MEME	SPSPFinder	DPSFinder
13nt	ACGAGGCTTACCG	-	-	(ACGA)(GGCT)(TACC)(G)	(A)(CGA)(GGC)(TTACCG)
ACE2	GCTGGT	-	-	(GCTG)(GT)	[GCA GCT][GGT CGT]
ADR1	TCTCC	-	TCTCC	(TCTC)(C)	(TC)[TCC TTC TGC]
AP1	TTANTAA	-	-	(TTA)[G C](TAA)	(TTA)[GTA CTA](A)
CCBF	CNCGAAA	CACGAAA	-	(C)[A C G T](CGAA)(A)	[CAC CGC][GAA CAA](A)
CPF1	TCACGTG	CACGTG	TCACGTG	(CACG)(TA)	[TCA CCA GCA](C)[GTG GAG]
CSRE	CGGAYRRAWGG	-	-	(CGGA)[TGA TAA CGG](A)[A T](GG)	(CGG)[ATG ATA][AAT AAA AAG](GG)
CuRE	TTTGCTC	TTTGCTCA		(TTT)[GCT TCG](C)	(T)[TTG GTG](CTC)
GATA	CTTATC	CTTATC	-	(CTTA)(TC)	[CTT CTA](ATC)
HAP2/3/4	CCAATCA	-	-	(CCAA)[TC TG CC](A)	-
LEU	CCGNNNNCGG	CCGGGACCGG	CCGGAACCGG	(CGG)[A G](ACCG)(G)	(CC)[GGT GGG][ACC CCC](GG)
MAT2	CRTGTWWWW	CATGTAATTA	-	[GA GT TA](AATT)[AC TA TC TG][A C]	(CAT)[GTA GTT][ATT AAT]
MCM1	CCNNNWWRGG	CCCGTTTAGG	CCTAATTAGG	-	(CCTA)[AAA AAC AAT][GGG TGG AGG]
SFF	GTMAACAA	-	GTCAACAA	-	-
UASCAR	TTTCCATTAGG	-	-	(T)[GCCC TCAC TCCA](TT)[AGCG AGGA]	(AT)[TTC TGC](CAT)(TAG)

Motifs of transcription factors that cannot be found by any algorithms were not shown in this table. ‘M’ stands for ‘A’ or ‘C’, ‘N’ stands for any nucleotide. ‘R’ stands for ‘A’ or ‘G’, ‘W’ stands for ‘A’ or ‘T’, ‘Y’ stands for ‘C’ or ‘T’.

5 Conclusion

In this paper, we introduced the DPS representation to capture the nucleotide dependency in a motif, which is simpler than the SPSP representation. We also developed a branch and bound algorithm DPS-Finder to locate the optimal DPS motif. Experimental results on real biological datasets show that DPS-Finder is efficiency and the DPS representation

is powerful enough to capture most of the real motifs. Further directions include extending the model and the algorithm to local motif pairs or non-linear motifs.

References

1. J. Allali and M.F. Sagot. The at most k-deep factor tree. *Internal Report Institut Gaspard Monge, University of Marne-la-Vallee*, IGM 2004-03, Juillet 2004.
2. T. Bailey and C. Elkan. Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Machine Learning*, 21:51—80, 1995.
3. Y. Barash, G. Elidan, N. Friedman and T. Kaplan. Modeling Dependencies in Protein-DNA Binding Sites. *RECOMB*, 28—37, 2003.
4. J. Buhler and M. Tompa. Finding motifs using random projections. *RECOMB*, 69—76, 2001.
5. M.L. Bulyk, P.L.F. Johnson and G.M. Church. Nucleotides of transcription factor binding sites exert interdependent effects on the binding affinities of transcription factors. *Nuc. Acids Res.*, 30:1255—1261, 2002.
6. F. Chin and H. Leung. An Efficient Algorithm for String Motif Discovery. *APBC*, 79—88, 2006.
7. F. Chin and H. Leung. DNA Motif Representation with Nucleotide Dependency. *TCBB* (to appear)
8. F. Chin, H. Leung, S.M. Yiu, T.W. Lam, R. Rosenfeld, W.W. Tsang, D. Smith and Y. Jiang. Finding Motifs for Insufficient Number of Sequences with Strong Binding to Transcription Factor. *RECOMB*, 125—132, 2004.
9. S. Hannenhalli and L.S. Wang. Enhanced Position Weight Matrices Using Mixture Models. *Bioinformatics*, 21(Supp 1):i204—i212, 2005.
10. U. Keich and P. Pevzner. Finding motifs in the twilight zone. *RECOMB*, 195—204, 2002.
11. S. Kielbasa, J. Korbel, D. Beule, J. Schuchhardt and H. Herzel. Combining frequency and positional information to predict transcription factor binding sites. *Bioinformatics*, 17:1019—1026, 2001.
12. C. Lawrence, S. Altschul, M. Boguski, J. Liu, A. Neuwald and J. Wootton . Detecting subtule sequence signals: a Gibbs sampling strategy for multiple alignment. *Science* 262:208—214, 1993.
13. C. Lawrence and A. Reilly. An expectation maximization (em) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins: Structure, Function and Genetics*, 7:41—51, 1990.
14. H. Leung and F. Chin. Discovering Motifs with Transcription Factor Domain Knowledge. *PSB*, 472—483, 2007.
15. H. Leung and F. Chin. Finding Exact Optimal Motif in Matrix Representation by Partitioning. *Bioinformatics*, 22:ii86—ii92, 2005.
16. M. Li, B. Ma, L. Wang. Finding similar regions in many strings. *Journal of Computer and System Sciences*, 65:73—96, 2002.
17. S. Liang. cWINNOWER Algorithm for Finding Fuzzy DNA Motifs. *Computer Society Bioinformatics Conference*, 260—265, 2003.

18. T.K. Man and G.D. Stormo. Non-independence of Mnt repressor-operator interaction determined by a new quantitative multiple fluorescence relative affinity (QuMFRA) assay. *Nuc. Acids Res.*, 29:2471—2478, 2001.
19. G. Pavesi, P. Mereghetti, F. Zambelli, M. Stefani, G. Mauri and G. Pesole. MoD Tools: regulatory motif discovery in nucleotide sequences from co-regulated or homologous genes. *Nuc. Acids Res.*, 34:566—570, 2006.
20. G. Pesole, N. Prunella, S. Liuni, M. Attimonelli and C. Saccone. Wordup: an efficient algorithm for discovering statistically significant patterns in dna sequences. *Nucl. Acids Res.*, 20(11):2871—2875, 1992.
21. P. Pevzner and S.H. Sze. Combinatorial approaches to finding subtle signals in dna sequences. *ISMB*, 269—278, 2000.
22. S. Sinha S and M. Tompa. A statistical method for finding transcription factor binding sites. *ISMB*, 344—354, 2000.
23. E. Ukkonen. On-line construction of suffix trees. *Algorithmica*, 14(3):249—260, 1995.
24. X. Zhao, H. Huang and T.P. Speed. Finding Short DNA Motifs Using Permuted Markov Models. *RECOMB*, 68—75, 2004.
25. J. Zhu and M. Zhang. SCPD: a promoter database of the yeast Saccharomyces cerevisiae. *Bioinformatics*, 15:563—577, 1999. http://cgsigma.cshl.org/jian/

Appendix

In this section, we prove that the Candidate Motif Refinement Problem is NP-hard.

> **Candidate Motif Refinement (CMR) Problem:** given a motif P, reducing the size of P's wildcard pattern sets to at most k with the minimum p-value.

We prove it by reducing the Balanced Complete Bipartite Subgraph problem, which is NP-hard, to this problem.

> **Balanced Complete Bipartite Subgraph (BCBS) Problem:** given a bipartite graph $G = (V,E)$ and a positive integer k, we want to determine if there are two disjoint subsets $V_1, V_2 \subseteq V$ such that $|V_1| = |V_2| = k$ and $u \in V_1$, $v \in V_2$ implies that $\{u,v\} \in E$.

Given a BCBS Problem, we construct a motif P as follows: Let $l_{\max}$ be the smallest integer such that $4^{l_{\max}} \geq k|V|$. Each vertex v_i of G is represented by a unique length-$l_{\max}$ string $s(v_i)$. The candidate motif P is a length-$2l_{\max}$ pattern with exactly two wildcard pattern sets, each contains length-$l_{\max}$ string $s(v_i)$, representing the vertices in one partite of G. There are |E| length-$2l_{\max}$ input DNA sequences T. $s(v_i)s(v_j)$ is an input DNA sequence if and only if $\{v_i, v_j\} \in E$.

Under the restriction that the size of the wildcard pattern sets is at most k, the refined motif P' has the minimum p-value when the concatenation of each pair of patterns in the two wildcard pattern sets of size k exists in the input DNA sequences T (i.e. P' has exactly k^2 binding sites). Therefore, the BCBS problem can be solved by solving the

CMR problem and check if refined motif *P'* has exactly k^2 binding sites.

■