An Algorithm for Identifying Novel Targets of Transcription Factor Families: Application to Hypoxia-inducible Factor 1 Targets

Yue Jiang[1], Bojan Cukic[1], Donald A. Adjeroh[1], Heath D. Skinner[2], Jie Lin[1], Qingxi J. Shen[3] and Bing-Hua Jiang[2]

[1]Lane Department of Computer Science and Electrical Engineering. [2]Mary Babb Randolph Cancer Center, and Department of Microbiology, Immunology and Cell Biology, West Virginia University, Morgantown, WV 26506, U.S.A. [3]Department of Biological Sciences, University of Nevada, Las Vegas, NV 89154, U.S.A.

Abstract: Efficient and effective analysis of the growing genomic databases requires the development of adequate computational tools. We introduce a fast method based on the suffix tree data structure for predicting novel targets of hypoxia-inducible factor 1 (HIF-1) from huge genome databases. The suffix tree data structure has two powerful applications here: one is to extract unknown patterns from multiple strings/sequences in linear time; the other is to search multiple strings/sequences using multiple patterns in linear time. Using 15 known HIF-1 target gene sequences as a training set, we extracted 105 common patterns that all occur in the 15 training genes using suffix trees. Using these 105 common patterns along with known subsequences surrounding HIF-1 binding sites from the literature, the algorithm searches a genome database that contains 2,078,786 DNA sequences. It reported 258 potentially novel HIF-1 targets including 25 known HIF-1 targets. Based on microarray studies from the literature, 17 putative genes were confirmed to be upregulated by HIF-1 or hypoxia inside these 258 genes. We further studied one of the potential targets, COX-2, in the biological lab; and showed that it was a biologically relevant HIF-1 target. These results demonstrate that our methodology is an effective computational approach for identifying novel HIF-1 targets.

Introduction

In the past decade, we have witnessed unprecedented advances in genomic databases. The completion of the human genome project has provided us with sequence information on human genes, along with their regulatory sequences.[1] With the large amount of genomic information, developing efficient and effective computational tools to analyze such huge genomic data has become an important challenge. One important application of such analysis is in gene finding. Some programs for gene finding are designed to predict an entire gene sequence.[2–6] However, a majority of them are designed to identify some specific gene segments, such as promoters,[7,8] enhancers,[7] exons and CpG islands.[8]

Given the special role of transcription factors in gene expression, the identification of transcription factor targets is an important task.[9–15] A transcription factor controls and regulates gene expression by binding to a particular promoter or enhancer region of the gene. DNA fragment lengths for a transcription factor binding vary from 5 to 25 base pairs. However, a larger region of regulatory elements is involved in gene expression. Thus, in addition to the transcription factor binding site, other sequences may play important roles in gene expression. Therefore, more sophisticated approaches need to be explored in order to accurately identify the relevant sequences that control gene expression. Methods based on frequency of *k*-tuples and exhaustive pattern search have been proposed.[14] Methods that use both global and local alignments to predict transcription factors, and that considers the binding of transcription factors and *cis*-regulatory elements were previously described.[8,13]

Suffix tree based methods have been used in pattern discovery problems in biology. While exact pattern occurrences were considered in,[16] detecting transcription factor binding sites using suffix trees were considered in,[17,18] based on a method for suffix-tree based inexact pattern matching initially described in.[19] Essentially, inexact (*k*-mismatch) pattern matching was performed progressively: starting from the root, the method performs an exhaustive comparison of all the symbols on each branch that start from the node against the current position in the pattern, until up to *k* positions mismatch on the path, or the pattern is exhausted. The time requirement of the algorithms is exponential with respect to

Correspondence: Yue Jiang, West Virginia University, Morgantown, WV 26506, U.S.A. Email: yue@csee.wvu.edu

the length of the pattern and the size of the symbol alphabet, which makes the approach impractical for moderately sized sequences, or large number of sequences. In this work, we also use suffix trees as the basis for pattern matching, and consider only exact pattern matching. A key difference in our approach is the consideration of the practical implementation of this important data structure for environments with huge genomic databases, potentially involving millions of sequences, or billions of base pairs.

In this study, we develop a new methodology for identifying novel targets of hypoxia inducible factor 1 (HIF-1) based on the suffix tree data structure. The methodology includes the following four steps. Step1: Construct the suffix tree using a set of promoter sequences from known HIF-1 targets as training genes. Then we extract common patterns that occur in every training gene at least once from the suffix tree. Step 2: Using the common patterns and known HIF-1 binding site sequences to identify all potential HIF-1 target genes from the genome database. Step 3: Process the potential HIF-1 targets by positional analysis to select those targets with predicted HIF-1 DNA binding site and common patterns from above at the 5′ region upstream of the promoter. Step 4: Analyze the accuracy of the prediction for HIF-1 targets. Step 2 and Step 3 together ensure that interested motifs are located only in the 5′ upstream promoter region. This approach may be extended to identify potential novel targets of other transcription factors since they share similar characteristics for binding to the DNA sequence.

We use the suffix tree data structure in the first and second steps.[20] Given a string $S[1..n]$ of length n, a suffix tree is a rooted tree with n leaves, whereby the i-th leaf node corresponds to the suffix $S[i..n]$, each edge in the tree is a substring, and no two edges out of a node start with the same character. There are two advantages in using a suffix tree in complex string matching problems. One is the possibility of finding common patterns from multiple strings in linear time, and the other is the potential to search for multiple patterns in multiple strings in linear time (with respect to the length of the concatenated strings). The storage requirement is also linear. Table 1 lists the popular linear time search algorithms commonly used to search multiple patterns against a sequence (multiple sequences). Each algorithm in the table is described in detail in.[20] Assume k is the number of patterns; $m_i(0 < i < k)$ is the length of a pattern; M is the total length of patterns; M' is the total length of output patterns; n is the length of a sequence; σ is the total number of individual character in the sequence.

Table 1. Comparison of common string match algorithms.

Algorithm	Preprocessing	Search
Rabin-Karp	$\Theta(M)$	$O(nM)$
Aho-Corasick	$\Theta(M)$	$O(n + M')$
Knuth-Morris-Pratt	$O(M)$	$O(nM)$
Boyer-Moore	$O(M + \sigma)$	$O(nM)$
Suffix tree	$\Theta(n)$	$\Theta(M)$

The Table 1 compares several available string match algorithms when searching with multiple patterns (i.e. set of patterns) against a sequence. From the table, we can see that the suffix tree is the worst with respect to preprocessing time, but it outperforms all the others at the search phase. The $\Theta(n)$ preprocessing and $\Theta(M)$ search of suffix tree is not achievable by any of the other algorithms. The other methods would preprocess each requested string on input, and then take $O(n)$ or more worst case time to search for the string (n can be huge compared to M in our case). Thus, in theory, the suffix tree is efficient in both time and space, and has been used in different applications, such as in multiple genome alignment[21] and in the identification of sequence repeats.[22] However, there is still the difficulty of practical implementation of suffix trees suitable for analysis of huge datasets. A major contribution of this work is the development of a simple and innovative methodology for using suffix trees, which makes it feasible to use them on large genomic databases. We apply the method to the problem of finding novel targets of HIF-1 transcription factor, using a database containing millions of sequences, or billions of base pairs.

Materials and Methods

General methodology

The general methodology used in this study is illustrated in Figure 1. In brief, 1) A suffix tree is constructed using the set of training genes. A set of common patterns that occur on all training genes at least once is extracted from the suffix tree. 2) Using the multiple patterns (including the common patterns from the previous step and other

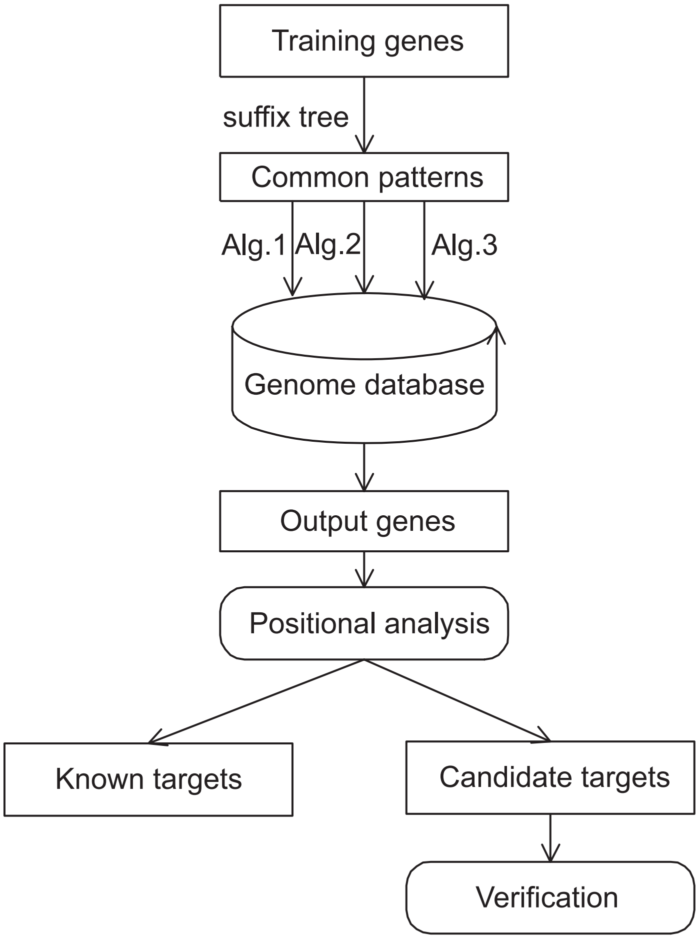

Figure 1. The outline of general methodology. The training genes of known HIF-1 targets are built into a suffix tree, and a set of common patterns are extracted from the suffix tree. Common patterns (including the set of common patterns and consensus sequences) are used to search the human genome database using the suffix tree algorithm. Using positional analysis, we analyze the output genes according to the relative locations of HIF-1 binding sites in the genes, and define the output genes with HIF-1 binding sites upstream of translational start site as potential HIF-1 targets. The potential HIF-1 targets are divided into two groups, known HIF-1 target genes and the candidate target genes. Finally, the candidate novel target genes are validated using available microarray data in the literature and tested in the biological lab.

known patterns such as HIF-1 binding sites (see Table 2) and consensus sequences from the literature, the genome database is searched by applying suffix tree algorithms. This generates the output sequences. 3) Positional analysis is performed on each output sequence according to the functional DNA fragments at the specific locations of the sequence. 4) The output targets from the positional analysis are grouped into known target genes and candidate targets. 5) The candidate target genes are further verified by doing biological experiments in the laboratory and by using available microarray data in the literature.

Selection of training genes

We used 21 known HIF-1 target genes, and download all available DNA sequences near HIF-1 binding sites from NCBI Nucleotide database (Table 2). In NCBI Nucleotide GenBank, there are gene features for each gene in the annotation database.[45] We extract 25 different DNA subsequences containing promoter and flanking sequence from these 21 HIF-1 target genes according to the feature information provided in GenBank. The length of subsequence for each HIF-1 target gene training sequence could be different. In these 25 subsequences, there are four genes: HO1, LDHA, EPO, and ENO1 with two different subsequences. Only one subsequence is used for each gene in the remaining 17 HIF-1 target genes. Thus, the known HIF-1 target genes are 21, and the subsequences are 25. We used leave-*k*-out cross-validation method[46] to select appropriate number of training gene subsequences for this study. Twenty-five HIF-1

Table 2. The HIF-1 binding sequences from 21 known HIF-1 target genes.

Gene	Subsequences			Ref.
α_{1B}AR	5′-CAGGCGA	CGTG	CTGCCGGG-3′	23,24
ADM	5′-CCCGTGGCAAA	CGTG	TTC-3′	24
	5′-GACAAA	CGTG	TCTAGCGTGAT-3′	24
	5′-ACAAA	CGTG	TCTAGCGTGAT-3′	25
ALDA	5′-CCCCCTCGGA	CGTG	ACTCGGACCAC-3′	25
	5′-GA	CGTG	ACT-3′	25
	5′-CTTCA	CGTG	CGGGGACCAGGGACCGT-3′	26
	5′-GGGATGTGGTCCGAGT	CACG	TCCG-3′	26
ET-1	5′-CGGGTCTTATCTCCGGCTG	CACG	TTGCCTGTGGGTGACTAATCACACAATAA-3′	26
ENO1	5′-GGCCA	CGTG	CGCCGCCTGCGCCTGCG-3′	26
	5′-AGGGCCGGA	CGTG	GGGCCCC-3′	26
	5′-ACGCTGAGTG	CGTG	CGGGACTCGGAGTACGTGACGGA-3′	26
	5′-CGCA	CGTG	GCCCCGGACACGCAGC-3′	26
EPO	3′-GCCCTA	CGTG	CTGTCTCACACAGCCTGTCTGAC-5′	26,27
	3′-GCCCTA	CGTG	CTGTCTCACACAGCCTGTCTGACCTACCGG-5′	28
	3′-GGGGCTGCTGCAGA	CGTG	CTGTCTCACACAGCCTGTCTGAC-5′	29
	3′-GCCCTA	CGTG	TCTCACACAGCCTGTCTGAC-5′	29
	5′-TGAGACAG	CACG	TAGGGC-3′	30
	5′-GCCCTA	CGTG	CTGCCTCGCAT-3′	26,27
	5′-GCTGGGCCCTA	CGTG	CTGTCTCACACAGCCTGTCT-3′	26,27
	5′-CCTA	CGTG	CTGTCTCACACAGCCT-3′	26,27
GLUT1	5′-TGGGTCCACAGG	CGTG	C-3′	31
	5′-CAGG	CGTG	CCGTCTGACACGCATC-3′	32
HO-1	5′-GAGCGGA	CGTG	CTGGCGTGGCACGTCCTCTC-3′	33
IGFBP1	3′-CAACTA	CGTG	CTCTGG-5′	34
	5′-GCAGGA	CGTG	CTCTGGGGGGCACACATAGCT-3′	34
	3′-TGCCCA	CGTG	CTGGCA-5′	34
	3′-GACACA	CGTG	CTTTCT-5′	34
	3′-GACACA	CGTG	CTTCCT-5′	34
LDHA	5′-ACA	CGTG	GGTTCCCGCACGTCCGC-3′	27
	5′-GTGGGAGCCCAGCGGA	CGTG	CGGGAA-3′	27
	5′-CACA	CGTG	GGTTCCCGCACGTCCG-3′	26
iNOS	5′-GTGACTA	CGTG	CTGCCTAGGGGCCACTGCC-3′	35
	5′-AGTGACTA	CGTG	CTGCCTAGG-3′	28
p35srj	5′-GTGTGCG	CGTG	GTGCCATACGGGACGT-GCAGCTACGTGCCCA-3′	30
FKL	5′-CCGGGTAGCTGGCGTA	CGTG	CTGCAG-3′	24
PGK1	5′-GA	CGTG	ACAAACGAAGCCGCACGTC-3′	27
	5′-CGCGT	CGTG	CAGGACGTGACAAATGGAAGTAGCACGTC-3′	

(*Continued*)

Table 2. (*Continued*)

Gene	Subsequences			Ref.
	5′-GTGAGA	CGTG	CGGCTTCCGTTTG-3′	24
	5′-CTGCCGA	CGTG	CGCTCCGGAG-3′	24
TF	5′-TTCCTG	CACG	TACACACAAGCGCACGTATTTC-3′	36
	5′-GTGTGATTGT	CGTG	GTAGTGGATTCCATGC-3′	36
	5′-A	CGTG	CGCTTTGTGTGTACGTGC-3′	36
TR	5′-AGCGTA	CGTG	CCTC-3′	36
	5′-CGCGAGCGTA	CGTG	CCTCAGG-3′	36
	5′-AGCGTA	CGTG	CCTCAGGAAGTGACGCACAGCCCCCCTG-3′	36
	5′-GGTGTA	CGTG	CGGAAGGAAGTGACGTAGATCCAGAGGG-3′	36
VEGF	5′-CCACAGTGCATA	CGTG	GGCTCCAACAGGTCCTCTT-3′	27
FLT-1	5′-TTGAGGAACAA	CGTG	GAATTAGTGTCATCGTAAAT-3′	37
	5′-TTGAGGAACAA	CGTG	GAATTAGTGTCATAGCAAAT-3′	37
Met	5′-TTAGCGGAGA	CGTG	GGAGAGGCCGAGAGCAAAGCTCGCG-3′	38
	5′-ACCTTGT	CGTG	GGCGGGGCAGAGGCGGGAGGAAACGC-3′	38
	5′-CAGACA	CGTG	CTGGGGCGGGCAGG-3′	38
	5′-CAGCGCG	CGTG	TGGGAAGGGGCGGAGGGAGTGC-3′	38
	5′-GGAGCGCG	CGTG	TGGTCC-3′	38
Nip3	5′-CCCGCGCACGCGCCGCA	CGTG	CCGCACGCGCCCCGCG-3′	39
RTP801	5′-ACGTTGCTTA	CGTG	CGCCCGG-3′	40

Abbreviations: α_{1B}AR, α_{1B} adrenergic receptor; ADM, adrenomedullin; ALDA, aldolase A; ET-1, endothelin-1; ENO1, enolase 1; EPO, erythropoietin; GLUT1, glucose transporter 1; HO-1, heme oxygenase 1; IGFBP1, insulin-like growth-factor binding protein 1; LDHA, lactate dehydrogenase A; iNOS, inducible nitric oxide synthase; PFKL, phosphofructokinase L; PGK1, phosphoglycerate kinase 1; PKM, pyruvate kinase M; TF, transferrin; TR, transferring receptor; VEGF, vascular endothelial growth factor; FLT-1, VEGF receptor. Note, in the above table, several sequences has "CACG" that is the complementary sequence of "CGTG".

gene subsequences are used in this analysis. We denote the 25 HIF-1 target gene subsequences as SET25. The following steps are used: Step 1: 15 training subsequences are randomly selected from SET25. Step 2: these 15 training subsequences are built into a suffix tree and then a set of common patterns that occur at least once in each gene are extracted from the suffix tree. Step 3: these common patterns and HIF-1 binding sites are used to search against SET25. Step 4: the number of the output genes is determined and the accuracy of the approach is calculated. Step 5: Steps 1 to 4 are repeated 1000 times, and the average results are recorded. Similarly, the above procedure is repeated using different numbers of training genes, namely 10, 12, 18, and 20 HIF-1 target gene subsequences. We obtained similar detection accuracy by using 15 and 18 training sequences, and lower detection accuracy using 10 and 12 training sequences. The detection accuracy using 20 training genes is slightly higher. However, the number of common patterns using 20 training genes is much smaller, which could lead to more potential false HIF-1 target genes in the prediction. Thus, we randomly selected 15 training genes in this study. The selected 15 known HIF-1 target gene subsequences are listed and their length of training subsequence are indicated inside parentheses: α_{1B}AR(3494), ADM(2356), ALDA(3586), ET-1(1329), ENO1(2312), GLUT1(480), HO-1(908), IGFBP1(1930), LDHA(6166), iNOS(1588), PFKL(699), TFR(365), VEGF(2362), FLT-1(2371), and c-met(3020).

Suffix tree algorithms for searching genome database

To facilitate the practical application of suffix trees on the huge genome database, we use a sliding window method which significantly improved the speed of the algorithms and reduced computer memory requirement. The basic idea is to sequentially analyze smaller chunks of the database based on a chosen window size. Considering a simple example using the string “CACGTGTTATGG” as shown in Figure 2, we wish to determine whether “TT” is in the string. The length of the longest pattern is two in the string. If the machine is able to process five characters at a time, a fixed window of five characters is adopted, and an overlap of one character is needed (overlapping size = the length of longest pattern –1). The window slides from the left to right with the movement size of four characters (movement size = window size—overlapping size). In the first phase, a substring of five characters “CACGT” is read, and used to construct a suffix tree to be searched using the pattern “TT”. In the next phase, the last character “T” from the previous phase is kept, and a substring “TGTTA” should be used to construct a suffix tree. The same process is performed until the search condition is met or the whole string is read.

For a short string, the advantage of using the sliding window may not be obvious. However, the sliding window method becomes extremely important when the string is long and the available computer memory is limited. For example, for large DNA sequences with 5,000,000 base pairs or a concatenation of several DNA sequences, the sliding window method has a noticeable advantage. The sliding window is particularly useful when the whole database (10, 268, 238, 630 base pairs in our case) is needed to be built into a suffix tree. The whole database can be viewed as a large string formed by concatenating all the DNA sequences in the database.

In this section, we describe the algorithms used to search the huge genome database to identify the potential novel target candidates. We use both the common patterns from the training genes (Table 3), and known HIF-1 binding sites (Table 2) as criteria in this search. If a gene contains all the common patterns and one of the HIF-1 transcription factor binding sites, then the gene is selected as an output gene. The stage of searching the huge genome database is a major bottleneck in finding potential novel transcription factor targets. Thus, three algorithms are proposed for this task. We refer to these three algorithms as *Algorithm* 1, *Algorithm* 2 and *Algorithm* 3, respectively.

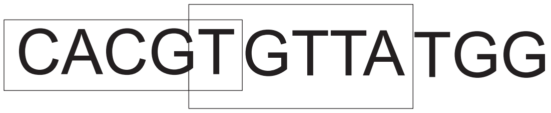

Figure 2. Sliding window method.

Algorithm 1 constructs one suffix tree for each sequence, then uses the common patterns to search against each suffix tree. Algorithm 1 is described as follows:

Algorithm 1

```
set number of characters to be processed $w_s = 8000$
    (note: we assume 8000 characters are processed at one time)
compute length of longest common pattern (overlap size).
**for** each sequence, $S_i$, in database **do**
    set overlap string $O_s$ to empty
    **while** not end of sequence $S_i$ **do**
        set $S_{tmp} = |O_s| + w_s$ characters of $S_i$
        construct a suffix tree, $ST$, for the subsequence $S_{tmp}$
        use multiple patterns search against the suffix tree $ST$
        record the search result
        determine the content of overlap string $O_s$
        update position for next $w_s$ characters from $S_i$
    **end while**
**end for**
```

Algorithm 2 uses the common patterns to build a suffix tree (ST_c), then uses the individual sequences (S_i) in the database to search against the suffix tree, ST_c.

Algorithm 2

```
set number of characters to be processed $w_s = 8000$
    (note: we assume 8000 characters are processed at one time)
calculate $L_1$, the length of the shortest pattern among the multiple patterns
calculate $L_2$, the length of the longest pattern among the multiple patterns
concatenate all the multiple patterns into one sequence, $S_c$
construct a suffix tree, $ST_c$, for $S_c$
```

Table 3. The set of 105 common patterns from 15 training genes.

AAAC	**AGGC**	**CCCTT**	**CTTC**	**GCGA**	**GGGAG**	**TCCA**
AACT	AGGGA	CCGGG	CTTG	GCGT	GGGC	TCCCC
AAGCA	ATCC	CCTC	GAAA	GCTA	GGGGC	TCCG
AAGG	CAAG	CCTG	GAAC	GCTC	GGGT	TCCTG
AAGT	CACA	CCTT	GACC	GCTGG	GGTC	TCTT
ACAC	CACC	CGGA	GAGCC	GCTTC	GGTG	TGAC
ACAG	CACG	CGGG	GAGGA	GGAA	GTCCT	TGAG
ACCC	CAGA	CGTG	GAGT	GGAC	GTGA	TGCCT
ACCT	CAGCA	CTAG	GATC	GGAGC	GTGCT	TGCG
ACGC	CAGCC	CTAT	GATG	GGAT	GTGT	TGCTG
AGAA	CAGGC	CTCA	GCAC	GGCC	TAAA	TGGC
AGAGC	CCAGC	CTCCC	GCAG	GGCG	TAGGG	TGGG
AGCAG	CCAT	CTGC	GCCA	GGCTG	TATA	TGTG
AGCCT	CCCAG	CTGGC	GCCC	GGCTT	TCAGG	TTCA
AGGAC	CCCCA	CTGT	GCCT	GGGAA	TCAT	TTCT

```
for each sequence, $S_i$, in database do
    set overlap string $|O_s|$ to empty
    while not end of sequence $S_i$ do
        set $S_{tmp} = O_s + w_s$ characters of $S_i$
        for each pattern $P_p$ in $S_{tmp}$ whose length is from $L_1$ to $L_2$ do
            search $P_p$ against $ST_c$
            record the search result
        end for
        determine content of overlap string $O_s$
        update position for next $w_s$ characters from $S_i$
    end while
end for
```

Algorithm 3 builds a suffix tree for the concatenation of all sequences (denoted ST_d), and another suffix tree for a concatenation of the common patterns (denoted ST_c). Then, the suffix tree ST_c is used to search against the suffix tree, ST_d.

Algorithm 3

```
concatenate all the multiple patterns into one sequence, $S_c$
construct a suffix tree, $ST_c$, from $S_c$
concatenate all the database sequences into one sequence, $S_d$
construct a suffix tree, $ST_d$, from $S_d$
use $ST_c$ to search against $ST_d$
record the search result
```

Algorithm 3 constructs a suffix tree for the entire database and stores it for later search. If Algorithm 3 is applied to a huge database such as the genome database, the suffix tree ST_d is built from all the sequences in the database. Thus, it requires a powerful machine with a huge memory. If we have such a machine that can be used to build a suffix tree for all the database sequences, this algorithm certainly would have some advantages: the whole database only needs to be built into a suffix tree once, and the database can be stored as one big suffix tree. It can be used to search different pattern sets as many times as one may wish. In this case, the search process is very fast, since the time used is linear with respect to the length of concatenated common patterns.

The proposed algorithms utilize the sliding window method to build a suffix tree (except for Algorithm 3). The processed DNA sequence is in FASTA format. A line of FASTA format DNA sequence contains 80 characters except the ending line. Thus, the sliding window algorithm process 100 lines (8000 characters) at a time, for a fixed window size of 8000 characters.

Positional analysis

Using Algorithm 1 and Algorithm 2, we searched the genome database. The output genes from both algorithms were the same. The only difference was the time each required. We further analyze the output genes using positional analysis.

A typical schematic diagram of a target gene activated by HIF-1 is shown in Figure 3. It is known that HIF-1 has the consensus binding site "RCGTG" (R stands for any of the four nucleotides: A, C, G, and T) at its target genes.[41–44] All the known HIF-1 binding sites are at the 5′ region upstream of the promoter sequence, that is, in 5′ enhancer region, except erythropoietin (EPO) which contains HIF-1 binding site in the 3′ enhancer region. From the information provided by the annotation databases in GenBank, it is quite difficult to obtain the stop site of gene coding sequence. Therefore, in the positional analysis, we only select the potential HIF-1 candidate targets that contain HIF-1 binding sites in the 5′ region upstream of the promoter.

To identify genes that have the HIF-1 binding site in the 5′ region upstream of the promoter, we need to find the HIF-1 binding site which is in the 5′ enhancer region from the target gene sequences. Letting V_s denote 5′ region upstream of the promoter, three methods are used to extract V_s from gene sequence based on the feature tables provided in the GenBank annotation database.[45] Method 1: For those gene sequences with the available enhancer sequence and position in the feature table, we extract the enhancer DNA sequence as V_s. Method 2: For those gene sequences with the available promoter region and sequence in the feature table, V_s is the DNA sequence of the 5′ region upstream of the promoter plus the promoter region. Method 3: For the remaining gene sequences with no information on either the promoter or enhancer sequence, we search for the first position of the beginning of "CDS", "TATA" box, or "CAAT" box sequences, called E_e. Then, we extract DNA sequence from 5′ end to E_e as V_s. After determining V_s by using the above three methods, we use Boyer-Moore fast string matching algorithm[20] to search whether the HIF-1 binding site "RCGTG" is inside V_s.

Lab verification

Human prostate cancer cells, PC-3 cells were cultured in RPMI 1640 supplemented with 10% fetal bovine serum (Intergen, Purchase, NY), 0.2 units/ml human insulin (Sigma, St. Louis, MO), 50 units/*ml* penicillin, and 50 *mg/ml* streptomycin (Invitrogen, Carlsbad, CA). These cells were seeded in a 12-well plate overnight, and transfected with the indicated plasmids using lipofectamine (Sigma) per the manufacturer's instructions. Briefly, COX-2 reporter plasmid (0.4 μg) containing a 960-bp human COX-2 promoter with the potential HIF-1 binding site was co-transfected with β-gal plasmid, and the control vector, HIF-1 dominant negative construct, or HIF-1α expression plasmid using 2 μl Lipofectamine per well in serum-free Opti-MEM media (Invitrogen, Carlsbad, CA) for 30 *min*. The transfection solution was then added to the cells, and incubated with cells for 4.5 *h*. The cells were then washed and cultured in the medium for 36 *h*. The cells were collected and analyzed using luciferase analysis buffer (Promega, Madison, WI). Luciferase activity was measured using a moonlight luminometer, and β-gal activity was measured as a control using the above cellular extracts. The relative luciferase activity was the ratio of luc/β-gal with the value normalized to the control as described previously.[27,49]

Results

In this study, we have used HIF-1 target genes as a model system, and developed a new methodology for identifying the novel HIF-1 target genes. Using a training set of 15 known HIF-1 target genes, we have obtained 238 potential HIF-1 targets including

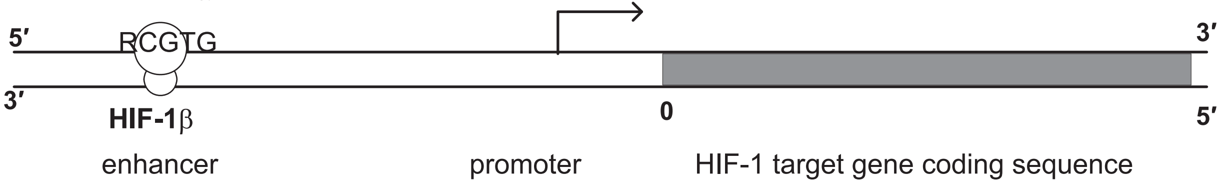

Figure 3. The regulation of a typical HIF-1 target gene. A HIF-1 target gene codes for a specific protein. The promoter is located immediately upstream of the coding sequence for the protein for regulating the gene expression. The enhancer is located upstream of the promoter with different lengths of spacing and with HIF-1 binding site. HIF-1 consists of HIF-1α and HIF-1β subunits. HIF-1α and HIF-1β can dimerize, and bind to the enhancer region to increase its promoter activity. HIF-1 commonly has the binding site "RCGTG" in the enhancer region.[41,42,44]

25 known HIF-1 targets from a large genome database. Although suffix trees have been around for some time, the key innovation in our approach is how to use them efficiently on a large database, using a standard personal computer. Our proposed method is particularly efficient, handling a large database of 2,078,786 DNA sequences with a total of 10,268,238,630 base pairs on a PC with 2.8 GHz, and 512 RAM. This confirms the feasibility of the proposed methodology. In addition, through literature search, 17 putative novel targets are verified by microarray data to be upregulated by HIF-1 or hypoxia. We also considered COX-2, one of the potential new targets proposed by our algorithm, and confirmed that COX-2 is a biologically relevant HIF-1 target gene. These results further demonstrate that this new methodology is effective in predicting novel HIF-1 targets.

Common patterns from training genes

To obtain the common patterns of HIF-1 target genes, we built a suffix tree using the randomly selected 15 known HIF-1 target training genes. From the suffix tree, we extracted a set of 105 common patterns that occurred in all training genes at least once. We fixed the minimum length at 4 base pairs. These are listed in Table 3.

Comparison of algorithms for searching genome database

The suffix tree data structure is constructed in linear time using Ukkonen's linear time algorithm.[20] The three algorithms proposed all have the same overall theoretical running time complexity. Each requires linear time, with respect to the total size of the database (i.e. length of all the concatenated database sequences). We consider the algorithms in terms of the suffix tree construction time, search time using the suffix tree, and memory requirement for the two stages. This is summarized in Table 4.

In terms of running time, the major difference is how much time each algorithm spends in constructing the suffix tree(s), or in searching while using the constructed suffix tree(s). For instance, while Algorithm 1 and 3 spend more time in constructing the suffix tree $O(n_s l_s)$, they spend less time in searching on the suffix tree $O(n_p l_p)$, where n_s = number of sequences in the database, n_p = number of common patterns, l_s = average

Table 4. Average case complexity for the proposed algorithms.

Complexity		Alg. 1	Alg. 2	Alg. 3
Time	Construction	$O(n_s l_s)$ $O(\|S_d\|)$	$O(n_s l_s + n_p l_p)$	$O(n_p l_p) = O(\|S_c\|)$
	Search	$O(n_s n_p l_p)$ $O(n_s\|S_c\|)$	$O(n_p l_p)$	$O(n_s l_s) = O(\|S_d\|)$
	Total	$O(n_s l_s + n_s n_p l_p) \approx O(n_s l_s)$	$O(n_s l_s + n_p l_p) \approx O(n_s l_s)$	$O(n_s l_s + n_p l_p) \approx O(n_s l_s)$
Space	Construction	$O(w_s)$	$O(n_s l_s + n_p l_p)$	$O(n_p l_p)$
	Search	$O(l_p)$	$O(l_p)$	$O(w_s)$
	Total	$O(w_s + l_p) \approx O(w_s)$	$O(n_s l_s + n_p l_p) \approx O(n_s l_s)$	$O(n_p l_p) + O(w_s) \approx O(w_s)$

*Average case complexity for the proposed algorithms (n_s = number of sequences in the database, n_p = number of common patterns, l_s = average length of a sequence, l_p = average length of a pattern, w_s = number of characters processed at one time (size of the sliding window), S_c is the concatenation of all the multiple patterns, and S_d is the concatenation of all the sequences in the database).

length of a sequence, and l_p = average length of a pattern. The reverse is the case for Algorithm 2. The overall time complexity (combining tree construction and searching) remains the same for the algorithms.

The memory requirement is, however, quite different for the three algorithms. For Algorithm 2, the advantage is that we only need to build a suffix tree for the multiple patterns once, then use it throughout the whole search. Algorithm 3 for instance requires extra memory proportional to the size of the entire database. It is obvious that Algorithm 2 should be the fastest and most practical if we do not have a powerful machine to support Algorithm 3. This is because, on average, the total length of the common patterns (i.e. after concatenation) is usually shorter than the length of a gene sequence, and the preprocessing time to build the suffix tree is quite short. Moreover, the suffix tree for the common patterns only needs to be built once. In practice, Algorithm 2 is the fastest of the three algorithms, although it has the same space complexity as Algorithm 1.

Algorithm 1 and 2 are more practical for those who do not have a supercomputer with huge memory. For instance, in our case, computational experiments were carried out on a Pentium 4 PC with 2.8 GHz and 512 MB memory. Thus, we implemented Algorithms 1 and 2, and use them to search the genome database.

The nucleotide database was divided into approximately 6 equal parts (based on the number of sequences). Algorithm 1 and Algorithm 2 were executed separately on these 6 parts of the database. The comparative results are shown in the Table 5. As can be observed, in each part of the database, Algorithm 2 processed more DNA sequences and more bytes per minute than Algorithm 1. On average, Algorithm 2 is about 36% faster than Algorithm 1.

Output genes from genome database

The final output genes after processing for the positional analysis are divided into two groups: the mammalian group contains genes from mammals, such as human, rat and bovine; the other group contains genes from non-mammals, such as virus and plant. Within the potential novel targets, the same gene in different species is counted as one gene. One of the goals is to find genes that may have important implications in human health and disease research. Thus, further analysis of the genes in the mammalian group was conducted. A total of 258 distinct genes were identified.

Verification of candidate targets

After applying positional analysis to the output genes, the remaining genes are called candidate targets. We further characterize the candidate targets using three approaches: by using known HIF-1 target genes in the literature, by microarray data from literature search, and by biological lab verification.

Verification of potential novel HIF-1 targets using known HIF-1 targets

In our final output, there are 25 known HIF-1 targets identified. Inside these 25 known output targets, there are 15 HIF-1 targets that are used for the training analysis. Additional six genes in the predicted output were also known HIF-1 targets: cyclin G2, p21(WAF), PGK, TGFα, Nip3, and trefoil factor. These 25 HIF-1 targets are shown in Table 6.

Table 5. Comparative results for Algorithm 1 and Algorithm 2.

DB	Sequences	Size (MB)	Avg./Seq (symbols)	Avg. speed (sequences/min)		Avg. speed (KB/min)	
				Alg. 1	Alg. 2	Alg. 1	Alg. 2
1st/6	346,466	643	1,856	216	521	401	967
2nd/6	346,464	1511	4,360	196	279	851	1,216
3rd/6	346,464	2125	6,134	195	216	1,196	1,325
4th/6	346,464	2691	7,766	169	169	1,312	1,312
5th/6	346,464	1638	4,728	189	299	894	1,414
6th/6	346,464	1661	4,793	197	285	944	1,366
Total	2,078,786	10,268	4,940	192	262	948	1,294

The validation of candidate novel HIF-1 targets using available microarray data

In a follow-up literature search, additional 17 putative novel HIF-1 targets from the output list were confirmed to be upregulated by HIF-1 or hypoxia by the microarray data. These targets are shown in Table 7. This result showed that our predicted novel HIF-1 targets can be found as upregulated targets of HIF-1 and hypoxia, further confirming the accuracy of our prediction.

Laboratory validation of a candidate novel HIF-1 target

We selected one of the candidate HIF-1 targets identified as described above to be tested in the biology laboratory. The verified gene was human cyclooxygenase-2 (COX-2) gene. There are two reasons for selecting COX-2. First, COX-2 is important in biological function such as tumor growth and angiogenesis. Second, the availability of COX-2 promoter construct (kindly provided by Dr. Jian Li, Harvard University, MA). It is difficult to obtain promoter constructs for each gene in our final output. COX-2 was a putative target at the time the experiment was carried out (See,[47] but its regulation by HIF-1 has been recently published independently.[48]

Table 6. The 25 HIF-1 known target genes in the final output.

Gene name	Accession#	ID
ALDA	X06351*, X12447, J05517	1
α_{1B}AR	D32045, AF116943*, X51585	2
DEC1	AB043885	3
cyclin G2	AF549495	4
ET-1	S76970*	5
EPO	M11319	6
HO-1	U70472*	7
c-met	AF046925*	8
IGFBP1	AY434089*, M74587, M59316	9
LDHA	U13679*, Y00309	10
PFKL	M61210*	11
iNOS	AJ308545, L23806 (AY445095*)	12
FLT-1	AJ224863*	13
ENO1	X16287*	14
p21(WAF)	U24170	15
p35srj	AF129290	16
ETS-1	L20682	17
TFR	X04664*	18
VEGF	M63971, AF095785*	19
ADM	S73906*, D78349	20
GLUT1	U82755*	21
PGK	X15339, AF335419	22
TGFα	AL732598	23
Nip3	AF283504	24
trefoil factor	AB038162	25

*Indicates the training set of 15 HIF-1 target genes.

It is known that HIF-1 target genes are regulated at the transcriptional level by triggering their promoter activity. Therefore, to determine whether HIF-1 expression plays a role in COX-2 transcriptional activation, PC-3 prostate cancer cells were transfected with a COX-2 promoter reporter containing a 960-bp human COX-2 promoter with the potential HIF-1 binding site. Expression of HIF-1 dominant negative construct specifically inhibited HIF-1 activity, and inhibited the COX-2 reporter activity in a dose-dependent manner (Fig. 4a). This result indicates that HIF-1 activity is required for COX-2 transcriptional activation. In order to determine whether HIF-1 is sufficient to induce COX-2 transcriptional activation, HIF-1α expression plasmid was co-transfected with the COX-2 reporter. The expression of HIF-1α in PC-3 cells induced HIF-1 expression and COX-2 reporter activity in a dose-dependent manner (Fig. 4b). Thus, HIF-1α is also sufficient to induce COX-2 transcriptional activation. This data demonstrates that COX-2 is a functional HIF-1 target. These result further shows that our methodology is effective in identifying HIF-1 novel targets. Lab verification indicates that HIF-1 is essential in regulating COX-2 transcriptional activation.

While there are certainly many potential HIF-1 targets in the final output, we performed experiments on COX-2. The complete list of output genes is in the supplementary files. We hope that the results of this work will spur others to run the required biological experiments to validate the genes from the final list and to test these potential HIF-1 targets.

Discussion

The basic methodology in this study is as follows: 1) extract common patterns from the known

Table 7. The 17 putative novel targets identified to be upregulated by HIF-1 or hypoxia based on microarray data through literature search.

Accession#	Gene name	Ref.
AY282416	Interleukin 8	50
M11567	Angiogenin	50
AY339617	carbohydrate sulfotransferase 1	51
AL121586	fer-1-like 4 (*C. elegans*)	51
AF050157	hypothetical protein	51
AF157623	serine protease	51
AJ400879	ribosomal protein L27a	51
AY428630	neuroblastoma RAS viral oncogene	51
U06950	tumor necrosis factor, lymphotoxin	51
AK038789	B-cell CLL/lymphoma	51
AK549495	cyclin-dependent kinase	51
AY149618	heat shock 70 kDa protein 1A	53
AY149619	heat shock 70 kDa protein 1A	53
NM_003670	BHLHB2	52
NM_017817	RAS oncogene	52
NM_009320	solute carrier family 6	53
AF055066	MHC class I	53

gene sequences; 2) use the set of common patterns to search the genome database; 3) analyze the target genes according to the specific gene's feature in the database.

The methodology proposed here is to identify HIF-1 novel target genes using a combination of the specific HIF-1 binding sequence "RCGTG" and the common patterns. Our approach can be applied to other transcription factors. The transcription factors generally have common DNA binding sequences such as activator protein 1 (AP-1),[38] and nuclear factor-kappaB (NF-*k*B).[39] AP-1 has the common binding site "TGACTCA".[54] NF-*k*B has the common binding site "CAAGGAGGGAA TTCCCGAGT."[55,56]

The methodology may be extended to study other functional genes because many genes are conserved across widely divergent species with similar functions. Genes with similar functions may have similar structure and sequences. Genes belonging to the same family commonly share specific sequences and/or consensus sequences. The idea is to generate the common patterns from known genes, then to use these common patterns to search for unknown novel targets. Thus, steps 1 and 2 may be applied to novel function prediction based on gene structure. We use the annotation database in GenBank which is available to the public. Apart from transcription factors studied here, the databases can be used to study other functional DNA segments, such as exons, introns, miRNAs, and 5´UTRs. For a different kind of gene, step three needs to be changed to adapt to the specific gene's feature, but the basic idea remains the same.

Furthermore, the approach may potentially be applied to other genes that have known consensus sequences and common regulatory patterns. The suffix tree method can be applied to general gene clustering and classification that needs to group and categorize similar genes together. An improvement in the results (for instance, further filtering the output target genes) could be obtained by combining the proposed suffix tree approach with statistical models.

Although the suffix tree data structure is used for exact string matching in this study, the suffix tree analysis can be further developed for inexact string matching problems.[20] The inexact matching such as *k-mismatch* is an inexact pattern matching problem: identify all the occurrences of pattern P in text T which allowing k characters of mismatch

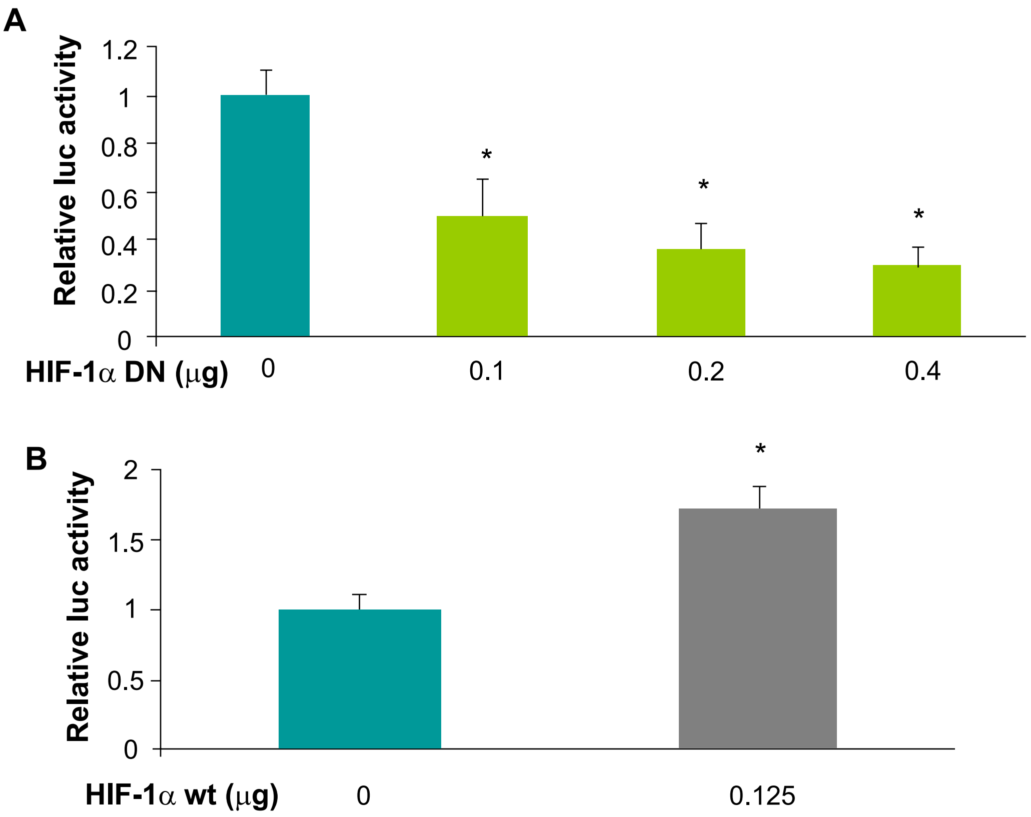

Figure 4. Effect of HIF-1 expression on COX-2 transcriptional activation. PC-3 prostate cancer cells were seeded into 6 well plates a day before the transfection. **a**) To determine whether HIF-1 activity is required for COX-2 transcriptional activation, the cells were co-transfected with COX-2 promoter luciferase reporter (PXP4/COX-2), pCMV-β-gal, and pcDNA3 vector or pcDNA3-HIF-1 dominant negative plasmid. **b**) To determine whether HIF-1 expression is sufficient to induce COX-2 transcriptional activation, the cells were co-transfected with the COX-2 promoter reporter, pCMV-β-gal, and pcDNA3 vector or pcDNA3-HIF-1α wild type expression plasmid. The cells were cultured for 36 h after transfection. The relative luciferase activity was determined by the ratio of luciferase/β-gal activity, and normalized to the vector control (100%). *Indicates the significant difference when the value is compared to the control ($p < 0.01$).

of pattern P. *k-mismatch* is very useful to find functional similarities (or gene mutations) among genes in bioinformatics.[17,18,20] In DNA sequences, mutation, insertion or deletion of nucleotide(s) happens frequently across different species or different individuals where the functional signals may not show up exactly. MicroRNA (miRNA) are a class of small non-coding RNAs with 21 to 23 base pair in length with hairpin structure, that play important roles in regulating post-transcription mRNA expression in animals and plants. Identification of miRNAs using computational methods is successful.[57] Most of computational prediction of novel MiRNA is based on phylogenetic conservation and structure similarity in closely related species, such as human,[57,58,60] animal,[57,60] insect,[57,59] and plants.[57] It would be interesting and useful to extend this suffix tree method to identify the potential targets of miRNAs in the future study. Taken together, the approach proposed here may be used as a general methodology to identify novel gene targets of a given transcription factor, and to study other gene function and regulation in the future.

Acknowledgments

We thank Dr. Jian Li (Harvard University, MA) for providing human COX-2 promoter construct. This work was supported by Grants CA109460 and CA78230 from National Cancer Institute.

Disclosure

The authors report no conflicts of interest.

References

1. Venter JC, Adams MD, Myers EW, et al. The sequence of the human genome. *Science*. 2001;291:1304–1351.
2. Burge C, Karlin S. Prediction of complete gene structures in human genomic DNA. *J Mol Biol*. 1997;268:78–94.

3. Eddy SR. Profile hidden Markov models. *Bioinformatics*. 1998;14: 755–763.
4. Reese MG, Kulp D, Tammana H, Haussler D. Genie—gene finding in Drosophila melanogaster. *Genome Res*. 2000;10:529–538.
5. Snyder EE, Stormo GD. Identification of coding regions in genomic DNA sequences: an application of dynamic programming and neural networks. *Nucleic Acids Res*. 1993;21:607–613.
6. Solovyev VV. Finding Genes by Computer: Probabilistic and Discriminative Approaches. In Jiang T, Xu J, Zhang MQ. (eds.), *Current Topics in Computational Molecular Biology*. The MIT Press; 2002. p. 201–248.
7. Zhang MQ. Promoter analysis of co-regulated genes in the yeast genome. *Comput Chem*. 1999;23:233–250.
8. Zhang MQ. Computational Methods for Promoter Recognition. In Jiang T, Xu J, Zhang MQ. (eds.), *Current Topics in Computational Molecular Biology*. The MIT Press: 2002; p. 249–267.
9. Brazma A, Vilo J, Ukkonen E, Valtonen K. Data mining for regulatory elements in yeast genome. *Proc Int Conf Intell Syst Mol Biol*. 1997;5:65–74.
10. Bulyk ML. Computational prediction of transcription-factor binding site locations. *Genome Biol*. 2003;5:201.
11. Liu R, Agarwal P. Computational identification of transcription factors involved in early cellular response to a stimulus. *J Bioinform Comput Biol*. 2005;3:949–964.
12. Taverner NV, Smith JC, Wardle FC. Identifying transcriptional targets. *Genome Biol*. 2004;5:210.
13. Zhang W, Ruan J, Ho TH, You Y, Yu T, Quatrano RS. Cis-regulatory element based targeted gene finding: genome-wide identification of abscisic acid- and abiotic stress-responsive genes in Arabidopsis thaliana. *Bioinformatics*. 2005;21:3074–3081.
14. Pesole G, Prunella N, Liuni S, Attimonelli M, Saccone C. WORDUP: an efficient algorithm for discovering statistically significant patterns in DNA sequences. *Nucleic Acids Res*. 1992;20:2871–2875.
15. Zhang MQ. Computational prediction of eukaryotic protein-coding genes. *Nat Rev Genet*. 2002;3:698–709.
16. Apostolico A, Bock ME, Lonardi S, Xu X. Efficient detection of unusual words. *J Comput Biol*. 2000;7:71–94.
17. Pavesi G, Mauri G, Pesole G. An algorithm for finding signals of unknown length in DNA sequences. *Bioinformatics*. 2001;(17 Suppl 1)S207–S214.
18. Pavesi G, Mereghetti P, Mauri G, Pesole G. Weeder Web: discovery of transcription factor binding sites in a set of sequences from co-regulated genes. *Nucleic Acids Res*. 2004;32:W199–W203.
19. Marsan L, Sagot MF. Algorithms for extracting structured motifs using a suffix tree with an application to promoter and regulatory site consensus identification. *J Comput Biol*. 2000;7:345–362.
20. Gusfield D. *Algorithms on Strings, Trees, and Sequences*. Cambridge University Press. 1997.
21. Hohl M, Kurtz S, Ohlebusch E. Efficient multiple genome alignment. *Bioinformatics*. 2002;(18 Suppl 1):S312–S320.
22. Kurtz S, Schleiermacher C. REPuter: fast computation of maximal repeats in complete genomes. *Bioinformatics*. 1999;15:426–427.
23. Eckhart AD, Yang N, Xin X, Faber JE. Characterization of the alpha1B-adrenergic receptor gene promoter region and hypoxia regulatory elements in vascular smooth muscle. *Proc Natl Acad Sci U S A*. 1997;94:9487–9492.
24. Semenza GL, Roth PH, Fang HM, Wang GL. Transcriptional regulation of genes encoding glycolytic enzymes by hypoxia-inducible factor 1. *J Biol Chem*. 1994;269:23757–23763.
25. Nguyen SV, Claycomb WC. Hypoxia regulates the expression of the adrenomedullin and HIF-1 genes in cultured HL-1 cardiomyocytes. *Biochem Biophys Res Commun*. 1999;265:382–386.
26. Semenza GL, Jiang BH, Leung SW, et al. Hypoxia response elements in the aldolase A, enolase 1, and lactate dehydrogenase A gene promoters contain essential binding sites for hypoxia-inducible factor 1. *J Biol Chem*. 1996;271:32529–32537.
27. Jiang BH, Rue E, Wang GL, Roe R, Semenza GL. Dimerization, DNA binding, and transactivation properties of hypoxia-inducible factor 1. *J Biol Chem*. 1996;271:17771–17778.
28. Melillo G, Musso T, Sica A, Taylor LS, Cox GW, Varesio L. A hypoxia-responsive element mediates a novel pathway of activation of the inducible nitric oxide synthase promoter. *J Exp Med*. 1995;182: 1683–1693.
29. Yamashita K, Discher DJ, Hu J, Bishopric NH, Webster KA. Molecular regulation of the endothelin-1 gene by hypoxia. Contributions of hypoxia-inducible factor-1, activator protein-1, GATA-2, AND p300/CBP. *J Biol Chem*. 2001;276:12645–12653.
30. Bhattacharya S, Michels CL, Leung MK, Arany ZP, Kung AL, Livingston DM. Functional role of p35srj, a novel p300/CBP binding protein, during transactivation by HIF-1. *Genes Dev*. 1999;13:64–75.
31. Chen C, Pore N, Behrooz A, Ismail-Beigi F, Maity A. Regulation of glut1 mRNA by hypoxia-inducible factor-1. Interaction between H-ras and hypoxia. *J Biol Chem*. 2001;276:9519–9525.
32. Okino ST, Chichester CH, Whitlock JP Jr. Hypoxia-inducible mammalian gene expression analyzed in vivo at a TATA-driven promoter and at an initiator-driven promoter. *J Biol Chem*. 1998;273: 23837–23843.
33. Lee PJ, Jiang BH, Chin BY, et al. Hypoxia-inducible factor-1 mediates transcriptional activation of the heme oxygenase-1 gene in response to hypoxia. *J Biol Chem*. 1997;272:5375–5381.
34. Tazuke SI, Mazure NM, Sugawara J, et al. Hypoxia stimulates insulin-like growth factor binding protein 1 (IGFBP-1) gene expression in HepG2 cells: a possible model for IGFBP-1 expression in fetal hypoxia. *Proc Natl Acad Sci U S A*. 1998;95:10188–10193.
35. Palmer LA, Semenza GL, Stoler MH, Johns RA. Hypoxia induces type II NOS gene expression in pulmonary artery endothelial cells via HIF-1. *Am J Physiol*. 1998;274:L212–L219.
36. Rolfs A, Kvietikova I, Gassmann M, Wenger RH. Oxygen-regulated transferrin expression is mediated by hypoxia-inducible factor-1. *J Biol Chem*. 1997;272:20055–20062.
37. Gerber HP, Condorelli F, Park J, Ferrara N. Differential transcriptional regulation of the two vascular endothelial growth factor receptor genes. Flt-1, but not Flk-1/KDR, is up-regulated by hypoxia. *J Biol Chem*. 1997;272:23659–23667.
38. Pennacchietti S, Michieli P, Galluzzo M, Mazzone M, Giordano S, Comoglio PM. Hypoxia promotes invasive growth by transcriptional activation of the met protooncogene. *Cancer Cell*. 2003;3:347–361.
39. Bruick RK. Expression of the gene encoding the proapoptotic Nip3 protein is induced by hypoxia. *Proc Natl Acad Sci U S A*. 2000;97: 9082–9087.
40. Leonard MO, Cottell DC, Godson C, Brady HR, Taylor CT. The role of HIF-1 alpha in transcriptional regulation of the proximal tubular epithelial cell response to hypoxia. *J Biol Chem*. 2003;278:40296–40304.
41. Semenza GL. Hypoxia, clonal selection, and the role of HIF-1 in tumor progression. *Crit Rev Biochem Mol Biol*. 2000;35:71–103.
42. Semenza GL. HIF-1: mediator of physiological and pathophysiological responses to hypoxia. *J Appl Physiol*. 2000;88:1474–1480.
43. Semenza GL. HIF-1 and tumor progression: pathophysiology and therapeutics. *Trends Mol Med*. 2002;8:S62–S67.
44. Semenza GL. Targeting HIF-1 for cancer therapy. *Nat Rev Cancer*. 2003;3:721–732.
45. Benson DA, Karsch-Mizrachi I, Lipman DJ, Ostell J, Wheeler DL. GenBank: update. *Nucleic Acids Res*. 2004;32 Database issue: D23–D26.
46. Stone M. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society (Series B)*. 1974;36:111–147.
47. Jiang Y. A General Method for Genome Search and Discovery. *MSCS Thesis, West Virginia University*. 2004.
48. Csiki I, Yanagisawa K, Haruki N, et al. Thioredoxin-1 modulates transcription of cyclooxygenase-2 via hypoxia-inducible factor-1alpha in non-small cell lung cancer. *Cancer Res*. 2006;66:143–150.
49. Jiang BH, Jiang G, Zheng JZ, Lu Z, Hunter T, Vogt PK. Phosphatidylinositol 3-kinase signaling controls levels of hypoxia-inducible factor 1. *Cell Growth Differ*. 2001;12:363–369.
50. Jones N, Jiang BH, Ivy A, Agani FH. Hypoxia-inducible factor 1 (HIF-1) regulates invasion of uveal melanoma cells during hypoxia. *Clin Exp Metastasis*. 2006;23:87–96.

51. Manalo DJ, Rowan A, Lavoie T, et al. Transcriptional regulation of vascular endothelial cell responses to hypoxia by HIF-1. *Blood*. 2005; 105:659–669.
52. Wang V, Davis DA, Haque M, Huang LE, Yarchoan R. Differential gene up-regulation by hypoxia-inducible factor-1alpha and hypoxia-inducible factor-2alpha in HEK293T cells. *Cancer Res*. 2005;65:3299–3306.
53. Greijer AE, van der GP, Kemming D, et al. Up-regulation of gene expression by hypoxia is mediated predominantly by hypoxia-inducible factor 1 (HIF-1). *J Pathol*. 2005;206:291–304.
54. Dragan AI, Liu Y, Makeyeva EN, Privalov PL. DNA-binding domain of GCN4 induces bending of both the ATF/CREB and AP-1 binding sites of DNA. *Nucleic Acids Res*. 2004;32:5192–5197.
55. Glasgow JN, Wood T, Perez-Polo JR. Identification and characterization of nuclear factor kappaB binding sites in the murine bcl-x promoter. *J Neurochem*. 2000;75:1377–1389.
56. Shetty S, Graham BA, Brown JG, et al. Transcription factor NF-kappaB differentially regulates death receptor 5 expression involving histone deacetylase 1. *Mol Cell Biol*. 2005;25:5404–5416.
57. Berezikov E, Cuppen E, Plasterk RH. Approaches to microRNA discovery. *Nature genetics*, 2006;38 Suppl:S2–S7.
58. Nam JW, Shin KR, Han J, Lee Y, Kim VN, Zhang BT. Human microRNA prediction through a probabilistic co-learning model of sequence and structure. *Nucleic Acids Res*. 2005;33(11):3570–3581.
59. Xu Y, Zhou X, Zhang W. MicroRNA prediction with a novel ranking algorithm based on random walks. *Bioinformatics*. 2008;24:50–58.
60. Xie X, Lu J, Kulbokas EJ, et al. Systematic discovery of regulatory motifs in human promoters and 3' UTRs by comparison of several mammals. *Nature*. 2005;434(7031):338–345.