Predicting Enhancers in Co-Expressed Gene Sequences

Harshit Maheshwari [#1], Prabhat Pandey [#2]
Advisor: Arnab Bhattacharya

[#1] Indian Institute of Technology, Kanpur, India
[#2] Indian Institute of Technology, Kanpur, India
harshitm@iitk.ac.in,prabhatp@iitk.ac.in,arnabb@cse.iitk.ac.in

November 15, 2013

Abstract

Cis-regulatory elements (enhancers) are considered to be a key component of the gene regulatory process and they are difficult to be located using conventional approaches. Even, there are no computational methods of finding enhancers in a genome sequence. Few attempts have been made to cluster Transcription Factor Binding Sites(TFBSs) and hence predict the potential enhancers but the enhancers generated by this method are prone to errors as it is very difficult to hit the proper density for clustering. Here, we present a tool to predict putative enhancers in co-expressed genes if the tool is provided a decent list of co-expressed genes.

1 Introduction

Transcription may be considered as a process where the rate limiting step is the recruitment of RNA Polymerase II to the promoter. Thus regulation of transcription can be viewed as the mechanism to vary this rate. Since promoter sequence is largely invariant this variation must come from the Enhancer. Enhancer is a cluster of Transcription Factor binding sites and act by combination of transcription factors that are bound to it. It is essential for the control of gene expression. Most of the enhancers are cis-acting and so, also referred as cis-regulatory element.

The cis-regulatory elements are difficult to locate using conventional approaches [2, 9, 8] as they are very small and scattered widely over genome's non-coding region [4]. Studies have found that the regions of high density of Transcription Factor Binding Sites are highly probable of being active regulatory sequence. In [1], 37 regions of the *Drosophila* melanogaster genome with high densities of predicted binding sites were identified for five transcription factors involved in anterior-posterior embryonic patterning. Nine of these clusters overlapped known enhancers. They incorporated conservation of binding-site clustering into a new genome-wide enhancer screen, and predict several hundred new regulatory sequences. Some efforts have been made in this regard but its correctness depends on whether the clustering method is hitting the right density or not.

We have proposed a method to predict potential enhancers in co-expressed genes. We start with predicting Transcription Factor Binding Sites in a genome sequence using P-Match [3] tool. It effectively combines pattern matching and weight matrix approaches. It uses the TRANSFAC\R library of weight matrices as well as sets of aligned known TFBSs collected in the TRANSFAC\R database. We have proposed two different ways to cluster the Transcription Factor Binding Sites to get putative enhancers in a genome sequence. But, simple clustering is expected to give a huge number of putative enhancers than the real enhancers as our algorithms blindly output all the clusters crossing some thresholds which were decided based on patterns in real enhancers. To take care of this issue, we are making use of the hypothesis that co-expressed genes are expected to have similar enhancer. Provided a list of co-expressed genes, the tool outputs the putative enhancers for each gene that are similar to some enhancer in most of the other genes of co-expressed gene set.

The roadmap of the paper is as follows- Section 2 describes the P-Match Algorithm used to predict Transcription Factor Binding Sites (TFBSs) within a sequence. Section 3 discusses two different approaches to cluster TFBSs. Section 4 contains description of an additional filtering used towards the end to get enhancers with strong possibility of existence. Section 5 analyzes the results of our method. Finally, Section 6 contains concluding remarks.

2 Predicting Transcription Factor Binding Sites(TFBSs)

We have implemented P-Match algorithm [3] to predict TFBSs in a sequence. The algorithm is based on simultaneous use of a positional weight matrix (PWM) and a set of aligned TFBSs. It uses matrix library of Nucleotide Frequency Matrix(NFM) provided by TRANSFAC\R which contains the NFMs for 1740 different transcription factors. Nucleotide frequency Matrix contains frequencies of each base(can be one among A,C,G,T) at each position in the regulatory sequence. To construct NFM for a transcription factor, a set of sites is compiled by grouping all sites from the database which are known targets of the selected transcription factor or a family of similar transcription factors. The sites are aligned by using a combination of Gibbs site sampling method [6] and a recursive application of Match program [5] making sure that the core of each site(experimentally found) is included in the alignment. In the end, that window is selected which provides lowest false positives.

The PWMs are computed from the nucleotide frequency matrix using the following formula:

$$w_{i,B} = f_{i,B} \times I_i; B \in \{\text{A,C,G,T}\}, \quad (1)$$

where,

$$I_i = \sum_{B \in \{\text{A,C,G,T}\}} f_{i,B} \times \log_2 f_{i,B} - \sum_{k=1,4} \frac{1}{4} \log_2 \frac{1}{4} \quad (2)$$

So, PWM is essentially 4*(Length of Transcription Factor, f) matrix for each transcription factor, f.

/* We have used a default cut-off of 0.9. */

The P-Match algorithm computes d-score which measures the similarity between a sub-sequence X of length L in a genome and a given transcription factor S. The d-score is calculated using weights of the nucleotides in the individual positions (present in PWM):

$$d = \frac{\text{MaxWeight} - \sum_{i=1,\text{L}} |w_{i,B(X_i)} - w_{i,B(S_i)}|}{\text{MaxWeight}} \quad (3)$$

where $B(X_i)$ and $B(S_i)$ are the nucleotides in ith position of the subsequence X and the site S, respectively. The d-score can take value from 0.0 to 1.0, where 1.0 denotes that there is a perfect match between the subsequence S and the sequence obtained by nucleotides with maximum frequency at each place.

Two different d-scores have been used: d_{matrix} is the d-score calculated for the whole site while d_{core} is the d-score calculated for the five most conserved positions in the PWM. By *'five most conserved positions'* we mean $\text{Top}_5(Max_{i \in \{\text{A,C,G,T}\}}(f))$

We have used a default cut-off score of 0.9 for d_{matrix} and a cut-off score of 0.9 for d_{core}. Only those TFBS are reported and accounted for in the next step of the algorithm which clear both the cut-offs.

3 Enhancer Predicting Algorithm

3.1 Sliding Window Approach

In this method, we slide a window of fixed size from one end to the other end of a gene sequence. In the process of scanning, we check if the region is dense enough (characterized by MINTFBS, ie. minimum number of TFBSs required in a window) and it is added to the enhancer list if satisfies the cut-off criteria. We have taken the MINTFBS to be 15, though there is not any backing towards this hypothesis but a typical enhancer used to have more than 15 TFBSs. The algorithm ensures that the maximum distance between two farthest TFBSs is less than the Sliding Window size.

3.2 Trie-Based Approach

The problem with Sliding Window based algorithm is that it enforces TFBSs to spread close to Window size. A dense cluster with a very short length as compared to Window size can not be the outcome of above algorithm as rather a sparse cluster spread across the sliding window size would be the corresponding cluster outputted by the algorithm. It is even difficult to choose the window size as there is not sufficient studies to provide the average length.

The algorithm we describe in this section takes care of the limitations of the above algorithm. It ensures that all the clusters of length greater than MINTFBS are generated. It starts with pairing up each of the TFBSs to both its neighbours. In the next iteration, the pairs are paired to both their neighbouring pairs and the process builds up a trie in a bottom-up way. During the process the clustered TFBSs that satisfy clustering criteria are added to the potential enhancers list and the clusters in which distance between farthest TFBSs cross the MAXLENGTH are discarded from clustering process. We have chosen MAXLENGTH to be 1 kilo base. In [7], the size of enhancer is said to be ranging from 100 base pairs to a kilo bases with average length of a typical enhancer being 500 base pairs. A cluster left in between discarded clusters is also discarded for the next iterations. The algorithm is described in Algorithm 1-3.

The above algorithm outputs all the possible clusters with at least MINTFBS shared transcription factor binding sites (TFBS) such that the largest distance between any two TFBS does not exceed MAXLENGTH.

Algorithm 1: clusterTFBS

Require: $List < Node > OriginalTrie$

if allAbandoned **then**
  return
end if
$List < Node > trie, trie1, trie2$
$i = 0$;
while $i < trie.Size$ **do**
  if $trie[i].abandoned$ **then**
    $i++$;
  else
    if $(i == trie.size - 1 || trie[i].abandoned)$ **then**
      if $(i == 0 || trie[i-1].abandoned)$ **then**
        $trie[i].abandoned = true$
      else
        $trie1.add(trrie[i])$
      end if
      $i+ = 2$
    else
      $trie1.add(combineNodes(trie[i], trie[i+1]))$
    end if
  end if
end while
$clusterTFBS(trie1)$
$trie = originalTrie.clone$
$i = 0$
while $\{i < trie.size\}$ **do**
  if $trie[i].abandoned$ **then**
    $i++$
  else
    if $i == trie.size - 1 || trie[i+1].abandoned$ **then**
      if $i == 0 || trie[i-1].abandoned$ **then**
        $trie[i].abandoned = true$
        $i+ = 2$
      else
        $trie2.add(trie[i])$
        $i+ = 2$
      end if
    else
      if $i == 0 || trie[i-1].abandoned$ **then**
        $trie2.add(trie[i])$
        $i++$
      else
        $trie2.add(combineNodes(trie[i], trie[i+1])$
        $i+ = 2$
      end if
    end if
  end if
end while
$clusterTFBS(trie2)$

Algorithm 2: CombineNodes

Require: Node $n1$, Node $n2$

Node n
$n.addTFBS(n1.getTFBS)$
$n.addTFBS(n2.getTFBS)$
if $(n.End - n.Start) > MAXLENGTH$ **then**
  $n.abandoned = true$
end if
if $n.sizegfTFBS > MINTFBS$ **then**
  $findPotentialClusters(n1, n2)$
end if
return n

Algorithm 3: findPotentialClusters

Require: Node $n1$, Node $n2$

Node n
$n.addTFBS(n1.getTFBS)$
for $i = 0; i < (MINTFBS - n1.noOfTFBS); i++$ **do**
  $n.addTFBS(n2.getTFBS[i])$
end for
$i = (MINTFBS - n1.noOfTFBS) > 0 ? : 0$
while $i < n2.noOfTFBS$ **do**
  $n.addTFBS(n2.getTFBS[i])$
  if $(n.End - n.Start) > MAXLENGTH$ **then**
    $break$
  end if
  $trieList.add(new\ Cluster(n.getTFBS))$
  $i++$
end while
$n.removeAll()$
$n.addTFBS(n2.getTFBS)$
for $i = n1.getNoOfTFBS - 1;$ $i > (n1.getNoOfTFBS - 1 - (MINTFBS - n2.getNoOfTFBS));$ $i--$ **do**
  $n.addTFBS(n1.getTFBS[i])$
end for
$(n1.getNoOfTFBS - 1 - (MINTFBS -$

$n2.getNoOfTFBS)) > 0?(MINTFBS - n2.getNoOfTFBS) : 0$
while $i >= 0$ **do**
  if $(n.End - n.Start) > MAXLENGTH$ **then**
    $break$
  end if
  $n.addTFBS(n1.getTFBS[i])$
  $trieList.add(new\ Cluster(n.getTFBS()))$
  $i++$
end while

Algorithm 4: isMatched

Require: Enhancer $e1$, Enhancer $e2$
editDistance = $e1.getTFBS.size + e2.getTFBS.size - 2 * intersectionSizeOf(e1.getTFBS, e2.getTFBS)$
if editDistance $=< MAXDIST$ **then**
  return true;
else
  return false;
end if

4 Another Filtering

After we get the potential enhancers from the above algorithm, we apply another filtering algorithm to make our claim stronger. It is observed that co-expressed genes are more likely to contain shared transcription factor binding sites (TFBS) [4]. So, if an enhancer is present in most of the co-expressed genes, it has high probability of actually being an enhancer. Our tool allows a user to provide a set of co-expressed genes and based on cross-gene enhancers, we output a list of enhancers with strong chances of being a real enhancer. As the process of predicting TFBs is not accurate because of possible errors in Position Weight Matrix creation step, we have not used perfect match between enhancers, but a loose match *isMatched*(Enhancer $e1$, Enhancer $e2$). *isMatched* is based on the idea of Hamming distance[6] which captures the dissimilarities between two strings of equal length. The edit distance between two enhancers is calculated as given in Algorithm 4. The non-transitivity property of isMathced enforces checking each pair for similarity. The algorithm is described in Algorithm 5.

Algorithm 5: crossGeneEnhancers

for $(i = 0; i < C.size; i++)$ **do**
  for $(j = 0; j < C[i].size; j++)$ **do**
    for $(l = 0; l < C[k].size; l++)$ **do**
      if $isMatched(C[i][j], C[k][l])$ **then**
        $C[i][j].adjList.add(C[k][l])$
        $C[k][l].adjList.add(C[i][j])$
      end if
    end for
  end for
end for
$list < list < Enhancers >> E$
for $m = 0; m < C.size; m++$ **do**
  $E\{m\} = null$
end for
for $m = 0; m < C.size; m++$ **do**
  for $n = 0; n < C[m].size; n++$ **do**
    if $C[m][n].adjList.size > MINGENEMATCHRQD$ **then**
      $E[m].add(C[m][n])$
    end if
  end for
end for

5 Result

We tested our tool on the following set of co-expressed genes: {SLC35D1, SLC26A2, PAPSS2, WWP2, SERPINH1, UAP1L1} because these genes are expected to be expressed in same space and time and therefore, expected to share enhancers. Following is the result obtained:

Table 1: Number of enhancers predicted for a Gene Set

Gene	Number of Enhancers Predicted
SLC35D1	12
SLC26A2	42
PAPSS2	34
WWP2	9
SERPINH1	50
UAP1L1	52

Using the ChIP-seq technology, powerful predictive tools have been developed to study regulatory sequences and their properties. Using a Support Vector Machine (SVM) framework on gene sequence it is now possible to accurately identify EP300-bound enhancers. First the SVM is trained on experimentally found enhancers and then it is used to find new enhancers that are enriched in both ChIP-seq signal and DNase I hypersensitivity signal in the mouse brain and are located near relevant genes.

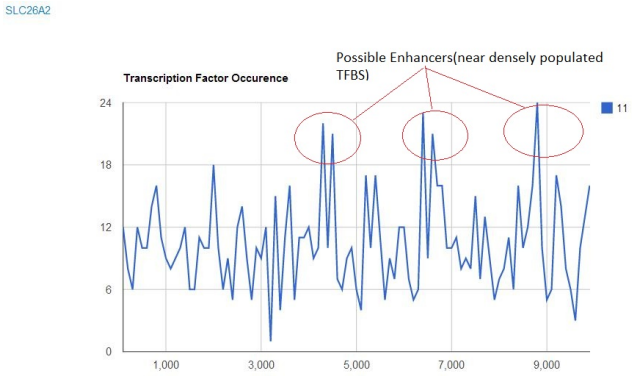

Figure 1: Number of Transcription Factors Found along the Length of the Gene SLC26A2

6 Limitations and Future Work

The tool is not automatically adjustable in a way that user interference is required for better results. For example, a user needs to provide a correct list of co-expressed genes in order to get good quality of results. A biologist generally looks for regulatory regions for co-expressed genes and not a single gene, so in most cases, we can expect a concerned user to be already having a list of co-expressed genes. Apart from that, the quality of results also depends on the upstream and downstream length provided by the user. Though there is not any concrete guidelines for choosing those lengths, but some studies have found that most of the regulatory regions have been observed to be within 2kb of promoters. We still need to test the possible enhancers found using the CRM Activity Database which is a compilation of experimentally validated enhancers and the REDFly database of enhancers.

References

[1] Benjamin P Berman, Barret D Pfeiffer, Todd R Laverty, Steven L Salzberg, Gerald M Rubin, Michael B Eisen, and Susan E Celniker. Computational identification of developmental enhancers: conservation and function of transcription factor binding-site clusters in drosophila melanogaster and drosophila pseudoobscura. *Genome biology*, 5(9):R61, 2004.

[2] Constanze Bonifer. Developmental regulation of eukaryotic gene loci: which cis-regulatory information is required? *Trends in Genetics*, 16(7):310–315, 2000.

[3] DS Chekmenev, C Haid, and AE Kel. P-match: transcription factor binding site search by combining patterns and weight matrices. *Nucleic acids research*, 33(suppl 2):W432–W437, 2005.

[4] Dongseok Choi, Yuan Fang, and William D Mathers. Condition-specific coregulation with¡ i¿ cis¡/i¿-regulatory motifs and modules in the mouse genome. *Genomics*, 87(4):500–508, 2006.

[5] Alexander E. Kel, Ellen Gößling, Ingmar Reuter, Evgeny Cheremushkin, Olga V. Kel-Margoulis, and Edgar Wingender. Matchtm: a tool for searching transcription factor binding sites in dna sequences. *Nucleic acids research*, 31(13):3576–3579, 2003.

[6] Charles E Lawrence, Stephen F Altschul, Mark S Boguski, Jun S Liu, Andrew F Neuwald, and John C Wootton. Detecting subtle sequence signals: a gibbs sampling strategy for multiple alignment. *science*, 262(5131):208–214, 1993.

[7] Gabriela G Loots. Genomic identification of regulatory elements by evolutionary sequence comparison and functional analysis. *Advances in genetics*, 61:269–293, 2008.

[8] Alan M Michelson. Deciphering genetic regulatory codes: a challenge for functional genomics. *Proceedings of the National Academy of Sciences*, 99(2):546–548, 2002.

[9] Len A Pennacchio and Edward M Rubin. Genomic strategies to identify mammalian regulatory sequences. *Nature Reviews Genetics*, 2(2):100–109, 2001.