Predicting splicing sites using Bayes Network

jiafeng Liao

Bioinformatics,Huazhong University of Science and Technology

June 2,2020

Abstract

Motivation: Owing to the complete sequencing of human and many other genomes, huge amounts of DNA sequence data have been accumulated. In bioinformatics, an important issue is how to predict the complete structure of genes from the genomic DNA sequence, especially the human genome. A crucial part in the gene structure prediction is to determine the precise exonintron boundaries, i.e. the splice sites, in the coding region.
Results: First,I conduct a χ^2-test to find the dependence between i-th and j-th bases using Training Set.Then,I expanded the dependency graph (which is usually a graph with cycles thatmake probabilistic reasoning very difficult, if not impossible)into a Bayesian network (which is a directed acyclic graph that facilitates statistical reasoning).
Availablity:the code is presented at github:https://github.com/JiaPei2433/Bayes-Network.git

1 Training Set

For donor site,we choose a slide window of 18 bases,which 9 bases are from upstream of donor site and 9 bases are from downstream of donor site. For acceptor site,we choose a slide window of 36 bases,which 27 bases are from upstream of acceptor site and 9 bases are from downstream of acceptor site.I make a weblogo chart **Fig.1**.

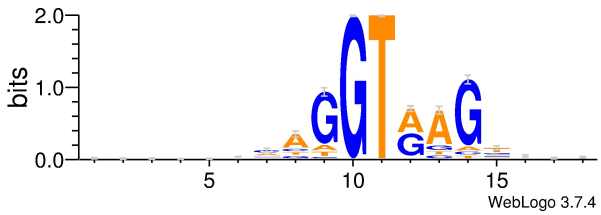

Figure 1: *Weblogo of pos donor*

2 Method

2.1 Test of dependency and the contingency tables

To perform the null hypothesis test of independence on a pair of nucleotides at the i-th and the j -th positions of a splice site, we formed a 4 Œ 4 contingency table, as shown in Table 1, by counting the observed number Ymn of DNA sequences where the i-th nucleotide X_i was m and the j -th nucleotide X_j was n (for simplicity, we have code A, T, C, G as 0,1,2,3).The contingency table is showed in table1.

Table 1: *contingency table*

X	A	T	C	G	total
A	Y_{11}	Y_{12}	Y_{13}	Y_{14}	Y_{1c}
T	Y_{21}	Y_{22}	Y_{23}	Y_{24}	Y_{2c}
C	Y_{31}	Y_{32}	Y_{33}	Y_{34}	Y_{3c}
G	Y_{41}	Y_{42}	Y_{43}	Y_{44}	Y_{4c}
total	Y_{r1}	Y_{r2}	Y_{r3}	Y_{r4}	Y

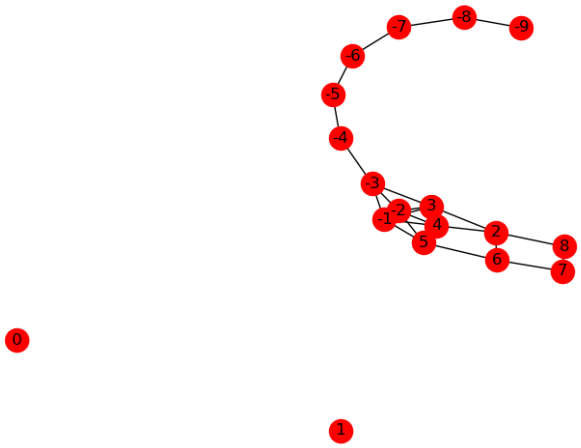

Figure 2: *Bayes Network of donor*

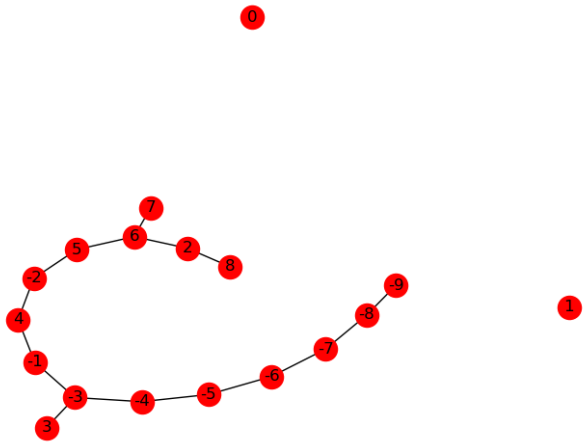

Figure 3: *Spanning Tree of donor*

The formula of χ^2 is as follows:

$$\chi^2(X_i, X_j) = \sum_{m=1}^{4}\sum_{n=1}^{4}\frac{(Y_{mn} - E_{mn})^2}{E_{mn}} \qquad (1)$$

where

$$E_{mn} = Y_{mc}Y_{rn}/Y$$

To determine the rejection region for the null hypothesis, we have specified a numerical value α for the Type I error of the test, according to a χ^2-distribution with degrees of freedom (4 1)ů(4 1) = 9, and then the critical point, K, was computed as follows:

P (null hypothesis is rejected when it is true) = P ($\chi^2(X_i, X_j)$ K | null hypothesis) = α
For donor sites,we set α=55.4491.

2.2 Bayes Network

We have got the value of χ^2 between i_th and j_th bases.We draw the Bayes Network using the packages $Networkx$.The Network is showed in **Fig.2**:

2.3 Expanded Bayes Network

Although cycle Bayes can fully represent the intrinsic dependency showed in **Fig.2**,it is not the most suitable for statistical reasoning.So we transfer Bayes Network into Tree modelthat is one-order subtree.I construct a spanning tree under the conditions of χ^2-value.The maximum spanning is derived by using Prime algorithm.The spanning tree is in **Fig.3**.

The core code using Python can been in **Fig.4**:

```
##create a tree containing all the edges
def get_all_edges(chart):
    node=list()
    tree=list()
    len=chart.shape[0]#the row numbers of chart
    for i in range(len):
        for j in range(i+1,len):
            if chart[i,j]:
                node=[str(i-9),str(j-9),chart[i,j]]

                tree.append(node)
    return(tree)

def Prim(nodes, edges):
    '''  基于最小堆实现的Prim算法  '''
    element = defaultdict(list)
    for start, stop, weight in edges:
        element[start].append((weight, start, stop))
        element[stop].append((weight, stop, start))
    all_nodes = set(nodes)
    used_nodes = set(nodes[0])
    usable_edges = element[nodes[0]][:]
    heapify(usable_edges)
    # 建立最小堆
    MST = []
    while usable_edges and (all_nodes - used_nodes):
        weight, start, stop = heappop(usable_edges)
        if stop not in used_nodes:
            used_nodes.add(stop)
            MST.append((start, stop, weight))
            for member in element[stop]:
                if member[2] not in used_nodes:
                    heappush(usable_edges, member)
    return MST
```

Figure 4: *Core code of Prim algorithm*

2.4 Training probabilistic tree model

According to the maximum spanning tree,we can scores a sequcence $GGAAAGAAGATGGCAGGG$

using the conditional probability.The formula is as follows:

$$P(X_{-9},X_{-8}...X_7,X_8) = P(X_1)P(X_2)P(X_{-3}|X_{-9}))$$
$$P(X_{-8}|X_{-9}))...P(X_8|X_7)P(X_3|X_7) \tag{2}$$

Firstly ,I count up 18 site probablistic matrixs encoding A,T,C,G as **0,1,2,3**.Secondly,I calculate 15 conditional probability matrixs according the tree model and contingency tables of $i-th$ and $j-th$ position.

2.5 Bayes Network model

According to last step,I create the scoring method for Bayes Network model.I train the positive model as well as negative model using Training SetM_t and M_f.An unknown sequence S can then be classified by calculating the probability ratio:

$$\frac{p(s|M_t)}{p(s|M_f)} \tag{3}$$

where P (S | MT) and P (S | MF) are the probability of having the tested potential splice site S based on the true splice site model MT and the probability based on the false splice site model MF, respectively. With an empirically determined threshold score T , the tested potential splice site S will be claimed real if the log-odds score is no less than T ; otherwise,it will be claimed pseudo.The core code of scoring is as follows:

```
#calculate the probability using Model
def cal_pro_M(M,count_base_prob,string):
    string=list(string.lower())#convert str into a list
    p=cal_stable_pro(count_base_prob,string)      #calcu
    for couple in couples:
        key=''.join(couple)#key to index 2-D list
        start=int(couple[0])+9
        stop=int(couple[1])+9
        if string[start]=='a':
            if string[stop]=='a':
                p=p*M[key][0][0][0]
            elif string[stop]=='t':
                p = p * M[key][0][0][1]
            elif string[stop]=='c':
                p = p * M[key][0][0][2]
            else:
                p = p * M[key][0][0][3]
        elif string[start]=='t':
            if string[stop]=='a':
                p=p*M[key][0][1][0]
            elif string[stop]=='t':
                p = p * M[key][0][1][1]
            elif string[stop]=='c':
                p = p * M[key][0][1][2]
            else:
                p = p * M[key][0][1][3]
        elif string[start]=='c':
            if string[stop]=='a':
                p=p*M[key][0][2][0]
            elif string[stop]=='t':
                p = p * M[key][0][2][1]
            elif string[stop]=='c':
                p = p * M[key][0][2][2]
            else:
                p = p * M[key][0][2][3]
        else:
            if string[stop]=='a':
                p=p*M[key][0][3][0]
            elif string[stop]=='t':
                p = p * M[key][0][3][1]
            elif string[stop]=='c':
                p = p * M[key][0][3][2]
            else:
                p = p * M[key][0][3][3]
    return(p)
```

Figure 5: *core code of tree model*

3 Results

Firstly,I extract the positive and negative sequences containing 18 bases from Testing Set and score them using my model,respectively.To plot ROC curve conveniently,I tag 1 for positive scores,whereas tag 0 for negative scores.So I get two .xls file containing scores for positive and negative data.Then,I use the pROC packages to plot the ROC curve and analyize the Sn-Sp properity.

4 Dissucsion

The representation of the donor (acceptor) site by a window around the exon/intron (intron/exon) boundary has been studied extensively. We found that the window from 9 bases upstream to 9 bases downstream of the exon/intron boundary best represents the donor site and the window from 27 bases upstream to 9 bases downstream of the intron/exon boundary best represents the acceptor site.The interdependency between base positions in the representative window of a splice site can be seen from the dependency graphs of the donor and the acceptor splice sites. As shown in **Fig.2**, strong interdependency among bases

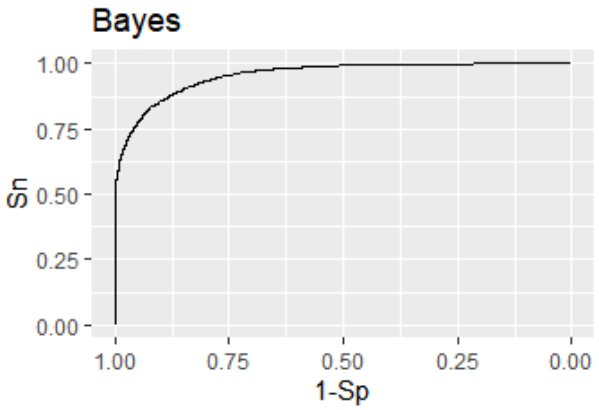

Figure 6: *ROC curve of donor*

$D_3, D_4 D_5, D\_1, D_7, D_6, D_5, D_{-2}$ of the donor site was observed and conforms to the consensus region of the donor site as indicated in Table 4. This implies that the spliceosome binds the donor site mainly on the bases downstream of the exon/intron boundary which conforms to our biological knowledge derived from experiments.

Reference

1.Te-Ming Chen1,Chung-Chin Lu1,and Wen-Hsiung Li.Te-Ming Chen1,Chung-Chin Lu1,and Wen-Hsiung Li.2005,471482.doi:10.1093/bioinformatics/bti025

2.Deyou Cai,Arthur Delcher,Ben Kao1 and Simon Kasif,Modeling splice sites with Bayes networks. 2000,152158. **3**.