利用贝叶斯网络预测真核生物剪接位点

摘要：在真核生物基因转录过程中会发生 mRNA 的剪接，其剪切位点在不同基因组和不同 mRNA 中都具有保守性。通过已有的实验已经证实剪接位点中的 donor 位点后通常是 GT 序列碱基，因此我们根据这个特征利用算法来识别剪接位点。目前已有许多机器学习的算法来进行位点识别，如支持向量机 SVM，贝叶斯网络，WMM，WAM 算法。本文通过贝叶斯网络算法预测真核生物剪接位点中的 donor 位点，并通过 python 实现。利用 KR set （ Kulp & Reese)作为训练集， BG set（Burset & Guigo）作为测试集，对模型进行训练和检验。

关键词：真核生物剪接位点预测，donor 位点，机器学习，贝叶斯网络

1. 引言

真核生物剪接发生在基因转录的过程中，在许多真核细胞中，基因的编码序列被一段段非编码序列隔开，这些割裂基因的编码序列被称为外显子，而非编码序列被称为内含子，在多细胞生物中，例如人类中，有内含子的基因数目非常多，每个基因所含的内含子也很多（最多的情况下可以达到 362 个）。当一个有内含子的基因转录时，RNA 最初包括了这些内含子。这些内含子被除去以得到成熟的 mRNA，内含子被除去的过程称为剪接。内含子和外显子边界的序列可以帮助细胞识别内含子并将其切除。这些剪接序列大多数存在于内含子中。这些序列根据它们相对于内含子末端的位置被称为 3’剪接位点（acceptor 位点）和 5’剪接位点（donor 位点）[1]。

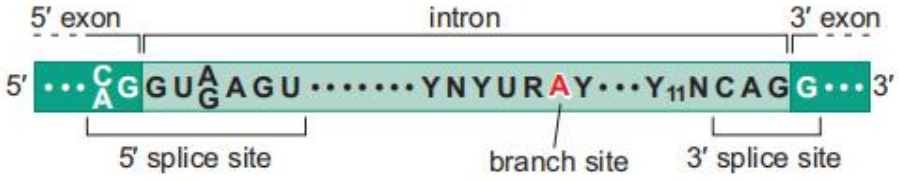


图 1，mRNA 序列内含子与外显子边界。这里是 mRNA 序列，所以 5’剪切位点，即 donor 位点边界为 GU,与我们已知的基因序列为 GT 相同。他们即为 donor 位点的共有序列（consensus sequences）[2]

本文利用 donor 位点的共有序列，通过机器学习算法贝叶斯网络提取其附近序列特征，从而用于真核基因剪接位点预测。贝叶斯网络是基于贝叶斯的一种方法，相对于 WAM 只考虑了相邻碱基之间的相关性，贝叶斯网络取了剪接位点周围很长序列，对每两个位点之间的相关性进行分析，即考虑了所有碱基之间的相关性，对于可信度高的位点之间关系保留，可信度低的位点之间关系舍去，构建一

个位点之间相关性的网络。

贝叶斯网络(Bayesian network)，又称信念网络(Belief Network)，或有向无环图模型(directed acyclic graphical model)，是一种概率图模型，于 1985 年由 Judea Pearl 首先提出。它是一种模拟人类推理过程中因果关系的不确定性处理模型，其网络拓朴结构是一个有向无环图(DAG)[3]。

2. 目标

本次上机实操目的为通过 python 实现贝叶斯网络预测真核剪切位点，从而加深对机器学习和贝叶斯网络的理解，通过其预测性能对贝叶斯网络算法进行性能评估，在后续与其他算法进行比较，学习了解贝叶斯网络的优劣势，并在实践的过程中锻炼编程能力。

3. 假设

我们已知 donor 位点边界是由 GT 碱基序列组成，但由于碱基序列非常长，每两个碱基中出现 GT 碱基的概率（$P = \frac{1}{4} * \frac{1}{4} = \frac{1}{16}$），对于真核生物碱基序列长度在 Mb 以上数量来说，出现 GT 碱基的数量会非常高，所以仅依此来判断是否为剪接位点并不可靠，会出现非常多的假阳性。同时对于生物中剪接序列的蛋白质来说，由于序列特别长，仅仅只识别 GT 序列不足以正确切割外显子与内含子。所以我们考虑到 donor 位点附近的碱基，把它们纳入 donor 位点特征中，用一条更长的序列来进行预测，而他们之间一定存在某种相关性，使剪接序列具有特异性而能被蛋白质识别。在本文的贝叶斯网络模型中，我们假设，剪接位点附近序列中各个碱基之间都可能存在相关性，通过卡方检验得到位点之间相关性检验，可信度高的则认为存在相关性，可信度低的不认为存在相关性，舍去，从而构建一个相关性的网络。

4. 算法原理

4.1 贝叶斯网络拓扑结构的构建

4.1.1 有向无环图[3]

贝叶斯网络的有向无环图中的节点表示随机变量，它们可以是可观察到的变量，或隐变量、未知参数等。认为有因果关系（或非条件独立）的变量或命题则用箭头来连接。若两个节点间以一个单箭头连接在一起，表示其中一个节点是“因(parents)”，另一个是“果(children)”，两节点就会产生一个条件概率值。

总而言之，连接两个节点的箭头代表此两个随机变量是具有因果关系，或非条件独立。把某个研究系统中涉及的随机变量，根据是否条件独立绘制在一个有向图中，就形成了贝叶斯网络。其主要用来描述随机变量之间的条件依赖，用圈表示随机变量(random variables)，用箭头表示条件依赖(conditional dependencies)。

例如，假设节点 E 直接影响到节点 H，即 E→H，则用从 E 指向 H 的箭头建立

结点 E 到结点 H 的有向弧(E,H)，权值(即连接强度)用条件概率 P(H|E)来表示，如下图所示：

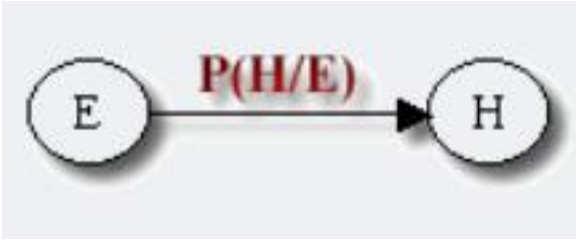


图 1 有向无环图

4.1.2 卡方检验

贝叶斯网络算法通过卡方检验来判断各个位点之间的相关性。

$X_i \backslash X_j$	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

图 2 一条核苷酸序列的两个独立位点之间的列联表，用于卡方检验。

卡方检验的公式如下[4]：

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

且可知：

$$\sum_{m=1}^{4} Y_{mc} = \sum_{n=1}^{4} Y_{rn} = Y. \quad (2)$$

$$E_{mn} = Y_{mc}Y_{rn}/Y \quad (3)$$

4.1.3 假设检验

这里假设检验的零假设为 Xi 位点和 Xj 位点独立，对应的自由度为(4-1)*(4- 1)=9。否定零假设的条件如下：

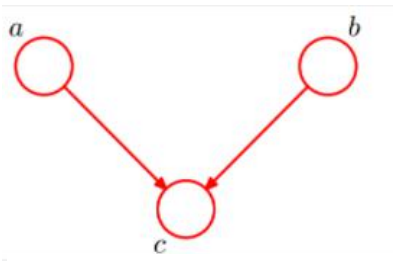
$$P\ (\text{null hypothesis is rejected when it is true})$$
$$= P(\chi^2(X_i, X_j) \geq K \mid \text{null hypothesis}) = \alpha$$

即我们需要设定一个显著性概率，大于这个值则说明假设可信，即认为两个位点之间独立，小于这个值则说明两个位点之间存在关联。一般这个值我们设为 0.05。

4.1.4 构建有向无环图

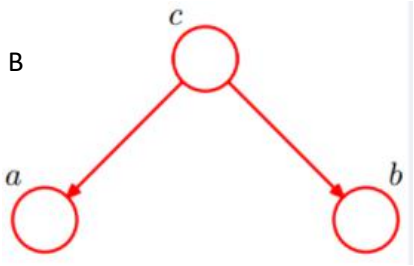
通过上述构建的位点之间的关联，构建有向无环图。在贝叶斯网络中，一般有三种形式：

A



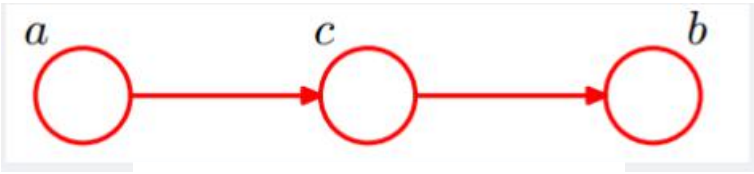
P(a,b,c) = P(a)*P(b)*P(c|a,b)

B



P(a,b,c) = P(a)*P(b)*P(c|a)*P(c|b)

C



P(a,b,c)=P(a)*P(c|a)*P(b|c)

图 3 贝叶斯网络中有向无环图的三种结构与计算公式。A，head-to-head。B，tail-to-tail。C，head-to-tail。

4.2 模型参数学习

构建了模型之后，利用训练集的数据，对构建的有向无环图进行参数学习，得到有向无环图中的条件概率。在贝叶斯网络中有两种参数学习的方法，即最大似然估计（MLE）和贝叶斯估计。

贝叶斯估计方法与传统的统计方法最大的差别就是两者对不确定性的看法上的区别. 后者把概率简单的看作是频率的无限趋进，而前者认为不确定性是人们对事物的一种认知程度，这种认知程度是由原来的主观知识和观察到的现象共同决定的. 因此 BE 方法将待估参数作为一个随机变量，在学习过程中，充分利用参数的先验分布和参数的选取规则，即就是考虑先验知识的影响，从而比 MLE 更为合理[5]。

4.3 计算打分

贝叶斯网络的判别函数与 WAM 算法相似，因为它们都是基于贝叶斯公式得到的判别函数，贝叶斯网络的判别函数如下：

$$\mathrm{Score}_M(S) = \log\left[\frac{P(S \mid M_{\mathrm{T}})}{P(S \mid M_{\mathrm{F}})}\right]$$

其中 MT 代表的是阳性数据训练得到的模型，MF 代表的是阴性数据训练得到的

模型。

5. 算法实现过程

在 python 中有专门用于构建贝叶斯网络的库 pmgpy，本文就利用 pmgpy 库来实现贝叶斯网络算法预测真核生物剪接位点。

5.1 数据提取

考虑到需要尽可能找到位点之间的相关性，在本文中选取了 donor 位点前 9 个碱基与后 9 个碱基，共 20 个碱基序列，得到其相关性。

5.2 贝叶斯网络结构学习

在 pgmpy 库中有 PC 算法、HillClimbSearch 算法、ExhaustiveSearch 算法和 MmhcEstimator，可以用于贝叶斯网络结构的学习，具体可在 pgmpy 的官方文档中查看不同算法的用法，这些算法用法都差不多，本文选用 PC 算法进行结构学习，后续再比较与爬山算法（HillClimbSearch）的区别。

在模型训练的数据中，阳性数据有 2379 条（没有 2381 条序列的原因可能为当获取的序列为 20 个碱基时，可能有两条序列包含了非 A、T、C、G 序列从而被去除），由于阴性数据过多导致程序运行特别慢，且阴性数据过多对模型的训练不一定会造成很好的结果，所以选用平衡的阴性数据进行模型学习。

在 PC 算法的 estimate 函数参数中

```
estimate(ci_test='chi_square',max_cond_vars=5,significance_level=0.01,variant='stable',n_jobs=- 1,show_progress=True)
```

其中第一个就包括了检验用的方法为卡方检验建立列联表，而 significance_level 则设定显著性阈值，从而构建了两个位点之间是否存在相关性。

通过 graphviz 包对上述学习得到的模型进行可视化。

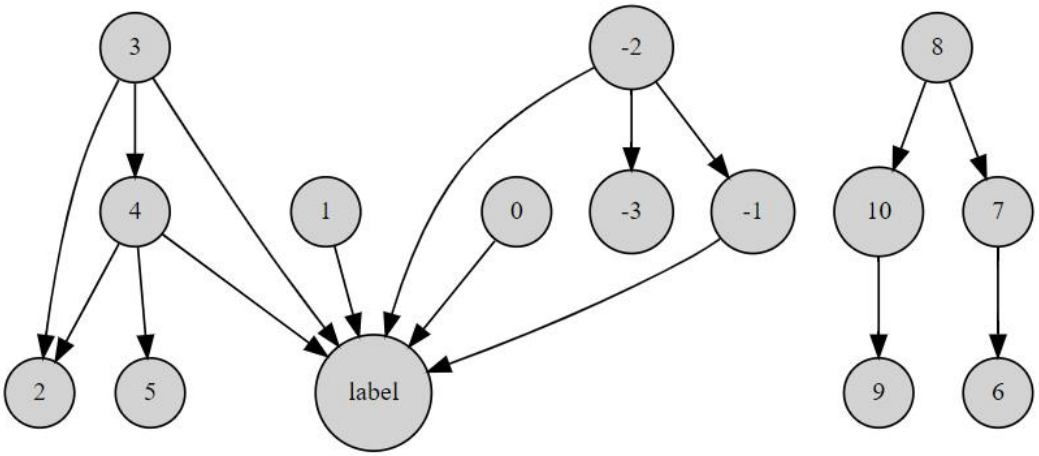


图 4 贝叶斯网络模型可视化。在这个图中可以得到哪几个位点之间存在相关性，便于后续操作。其中 0 位点为 G，1 位点为 T，二者与 label 的关系符合预期。

5.3 贝叶斯网络参数学习

得到模型后我们要进行贝叶斯网络的参数学习，得到每条边的条件概率，参数学习方法上述讲过，利用最大似然估计或者贝叶斯网络估计。本文先用贝叶斯网络估计来得到贝叶斯网络中的参数。

CPD of -1:

-2	-2(a)	-2(c)	-2(g)	-2(t)
-1(a)	0.1530511811023622	0.24061032863849766	0.210989010989011	0.1305699481865285
-1(c)	0.07381889763779527	0.23943661971830985	0.18021978021978022	0.13575129533678756
-1(g)	0.6751968503937008	0.25704225352112675	0.4307692307692308	0.49533678756476685
-1(t)	0.09793307086614174	0.26291079812206575	0.17802197802197803	0.23834196891191708

图 5 贝叶斯网络部分参数。通过贝叶斯网络估计算法得到的位点-1 与-2 之间的条件概率可视化。

5.4 利用训练的模型对测试集打分

在上述获得贝叶斯网络的模型后，对测试集进行预测，利用测试集的数据（缺失了 label 项）来得到 label 项的可能，每个序列会得到两个结果，即是剪接位点序列（label_1）和非剪接位点序列（label_2）的概率值。

	label_0	label_1
0	0.006173	0.993827
1	0.004695	0.995305
2	0.001795	0.998205
3	0.000000	1.000000
4	0.045455	0.954545

图 6 预测类别可能性部分值。

得到两个类别的概率值后通过 4.3 的公式进行打分，得到每条序列的分数。

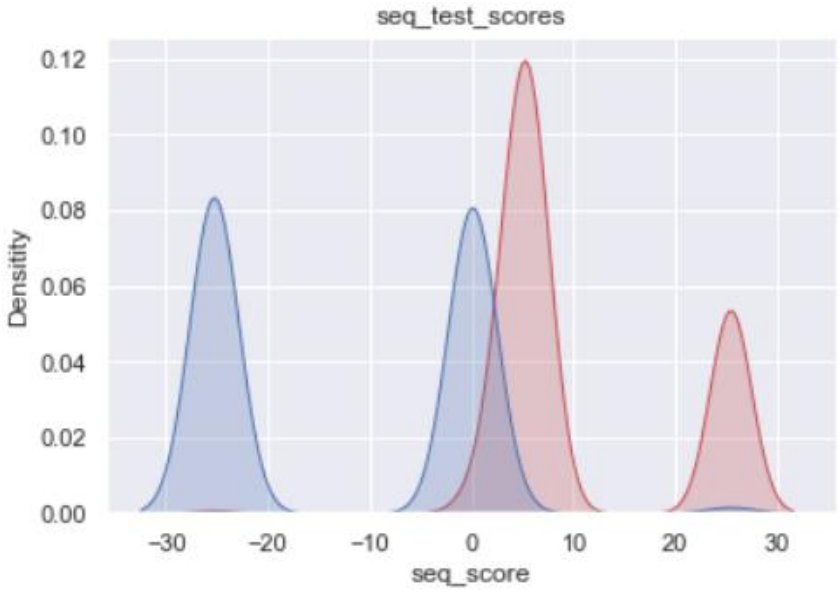
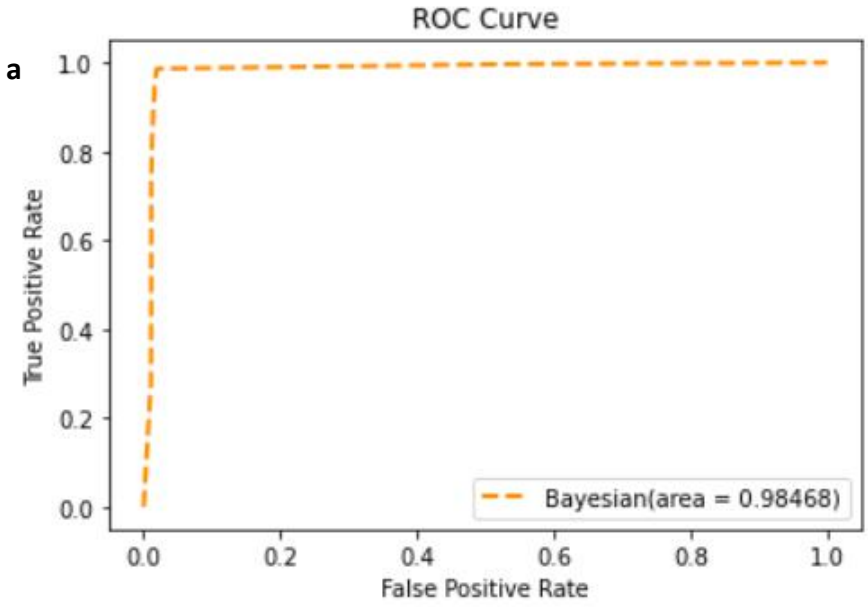


图 7 对测试集进行打分值与密度图。红色为阳性。蓝色为阴性。可以看到二者有重合的分数。

5.5 预测性能

在上述得到测试集的分数后，即可通过设定改变阈值得到该模型预测的 ROC 曲线图与 PR 图。



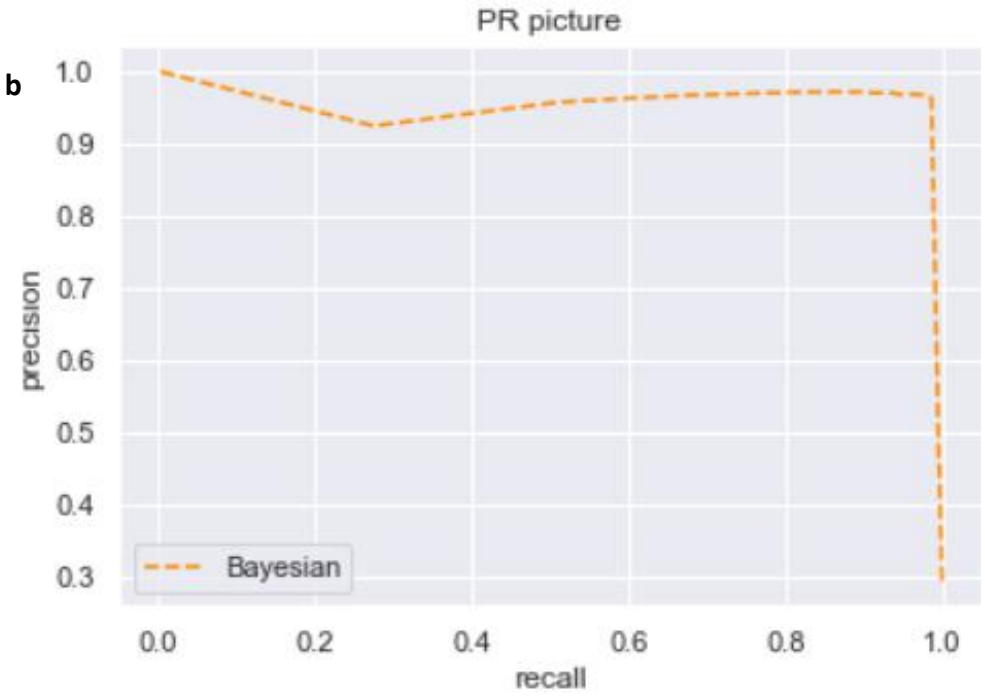


图 8 贝叶斯网络的 ROC 图与 PR 图。a，ROC 图。贝叶斯网络的模型 AUC 面积为 0.984，预测性能良好。**b，PR 图，**可以看到在右上角时预测性能良好。

6. 模型训练所用算法比较

上述代码实现过程中训练所用的算法为 PC 算法，下面用 HillClimbSearch 算法实现并与 PC 算法进行比较。

爬山算法适用于 5 个节点以上的结构学习，爬山搜索实现贪婪的局部搜索，从 DAG 开始（默认：断开 DAG），并通过迭代执行最大程度提高分数的单边缘操作继续进行。一旦找到本地最大值，搜索将终止[6]。所以本质上爬山算法应该能找到模型的最优结构并进行预测。

因为爬山算法找到的位点之间关联很多，所以在后面预测时如果也是取 20 个碱基的话，占用的内存会超过 8G（超过我电脑最大内存）无法运行，所以在进行爬山算法进行运算时我只选取了 donor 位点前 3 个碱基和后 4 个碱基，共 9 个碱基进行运算。训练集阳性样本与阴性样本比为 1：1，且阴性样本第四、第五位也是 GT 序列。

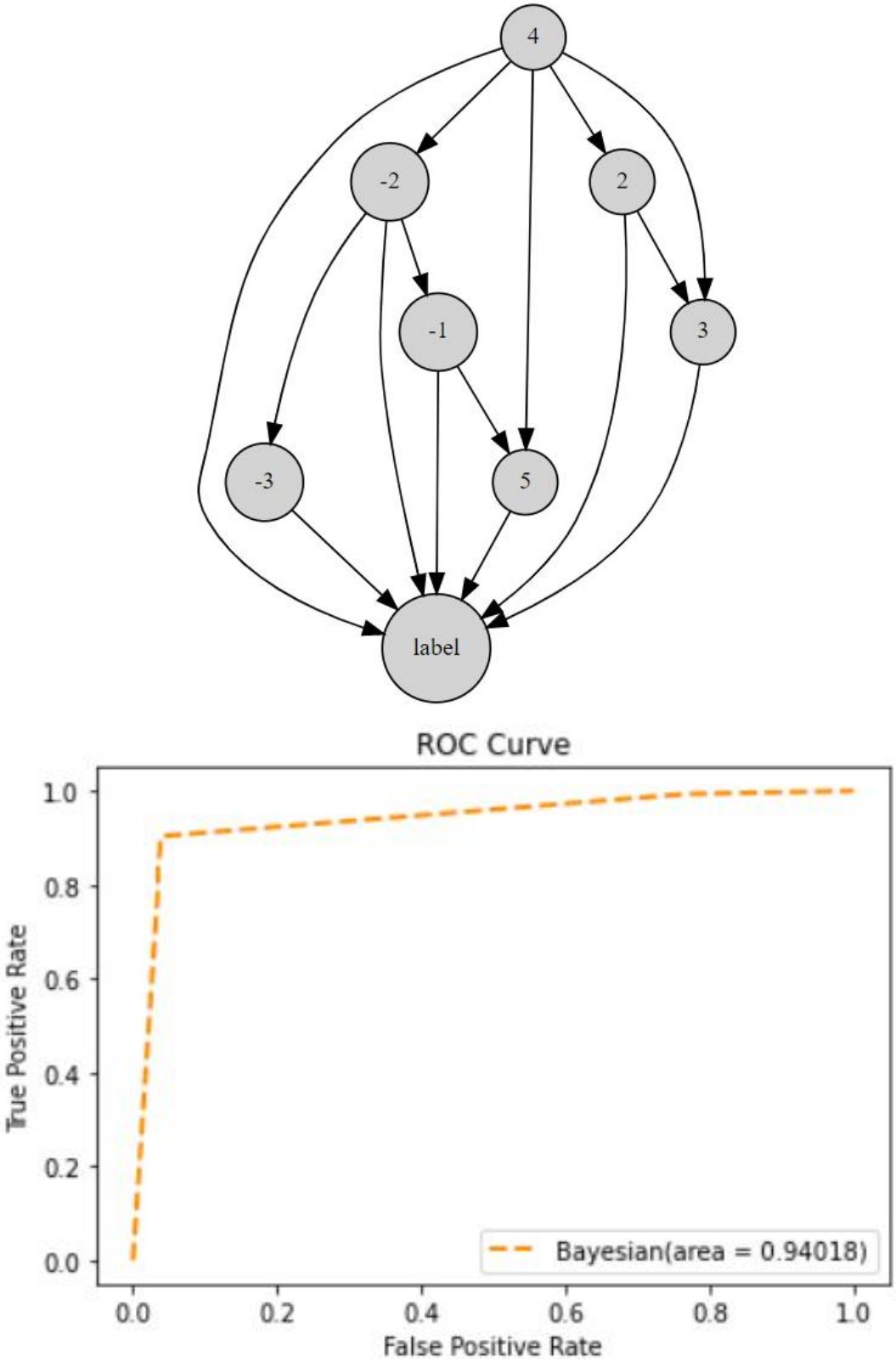


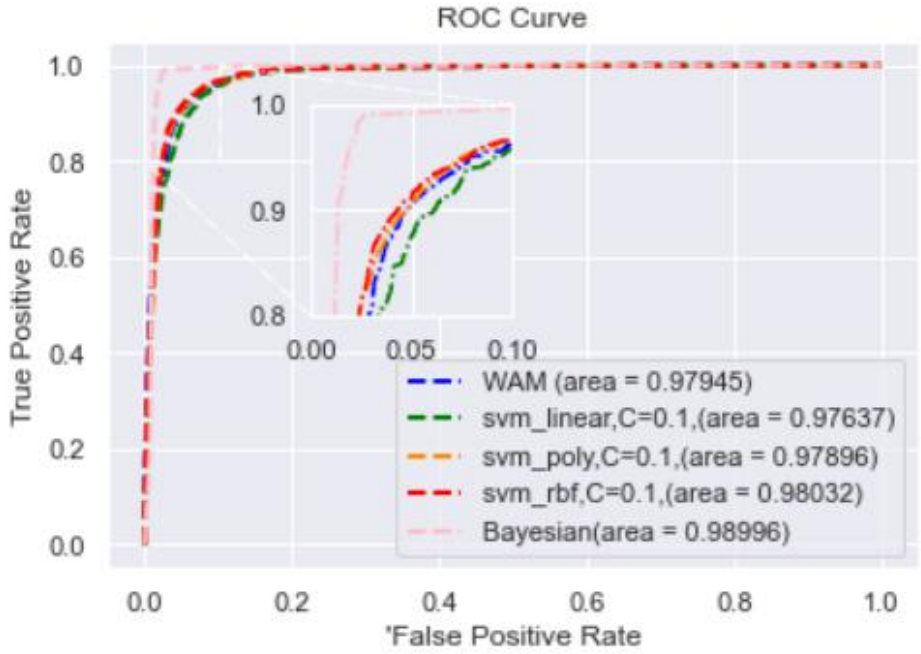
图 9 爬山算法预测得到的贝叶斯网络结构与预测性能。

爬山算法得到的预测性能在 0.94 左右，这有一部分原因在于训练样本中隐形样本第 4、5 位也是 GT 序列，导致阳性样本失去了这个强特征性，所以预测性能明显低于 PC 算法。但 PC 算法性能仍然优于爬山算法很多。

同时，虽然 PC 算法训练和预测时间都比较快，而爬山算法虽然训练模型很快，但预测时我只选用了 4155 条序列，但却运行了 9 个半小时，时间成本太高，性能却没达到预期，所以综合来看在利用贝叶斯网络预测真核生物剪接位点时用 PC 算法无论是内存、速度、还是性能都远远优于爬山算法。

7. 讨论

将贝叶斯网络算法与其他算法进行比较。



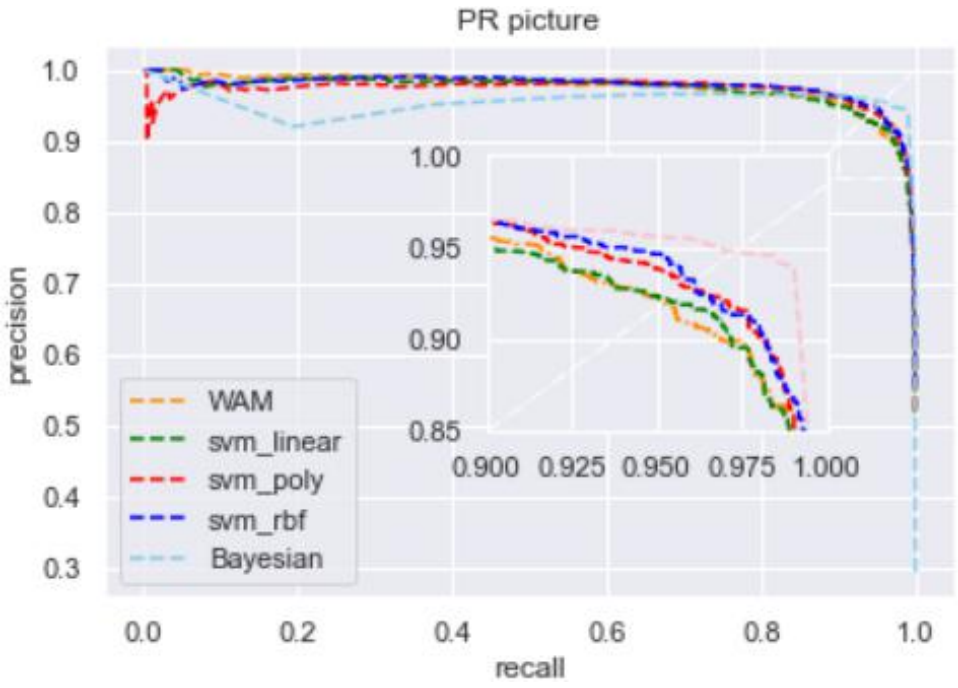


图 10 **贝叶斯网络算法与其他算法预测性能比较。**三种模型的预测性能都在 0.98 左右，ROC 曲线基本重合。

将贝叶斯网络算法的测试结果用 ROC 与 PR 图来描述，同时和 SVM 的三个核函数以及 WAM 算法进行比较。贝叶斯网络的准确率达到了 0.99，高于所有的其他比较算法，可见贝叶斯网络具有极好的性能。贝叶斯网络在朴素贝叶斯和理想贝叶斯之间进行了极其高效的平衡，既保证了准确性，也保证了速度和参数规模。

8. 数据获取

本文算法实现的完整代码与产出图片、报告电子档可在 github 上获取。获取链接如下：
https://github.com/xuweialan/Splicing-site-prediction/tree/main/Bayesian

9. 参考资料

1. 分子生物学书本 Molecular+Biology+of+the+Gene+(7th+Edition) P505
2. 分子生物学书本 Molecular+Biology+of+the+Gene+(7th+Edition) P469
3. https://blog.csdn.net/v_july_v/article/details/40984699
4. hen T , Lu C , Li W . Prediction of splice sites with dependency graphs and their expanded bayesian networks[J]. Bioinformatics, 2005,21(4):471-482.
5. https://zhuanlan.zhihu.com/p/355765619
6. https://blog.csdn.net/weixin_41599977/article/details/90453613