生物物理学报　第二十二卷　第五期　二〇〇六年十月
ACTA BIOPHYSICA SINICA　Vol.22　No.5　Oct. 2006

基于滑动窗口的原核转录起始位点计算定位方法

杜耀华，　王正志，　倪青山
（国防科技大学机电工程与自动化学院，长沙 410073）

摘要：转录起始位点的计算定位是基因转录调控研究的重要内容，但现有方法的识别性能较低。文章作者在已有原核启动子识别算法的基础上，提出了一种基于滑动窗口的原核转录起始位点计算定位方法，通过在合理限定的定位范围内对序列进行滑动扫描，来预测转录起始位点的位置。首先根据窗口序列的交迭组分特征和启动子其它特征分别建立二次判别分类器，用其计算对应位置的似然得分，再利用转录起始位点与翻译起始位点的间隔经验分布信息对似然得分进行修正，最后依照似然得分的分布情况由阈值定位算法确定预测位置。对大肠杆菌真实序列数据的测试结果表明，该定位算法可实现对真实转录起始位点位置的有效预测，与已有算法相比，当敏感性指标同为 0.85 左右时，特异性指标可从 0.20 提高至 0.65，从而使得定位准确率提高了约 20 个百分点。

关键词：原核基因组；转录起始位点；计算定位；滑动窗口；交迭组分特征
中图分类号：Q527

0　引　　言

转录起始位点（transcription start site, TSS）的计算定位指的是通过计算的方法给出 TSS 在基因组序列中可能的位置。当前，各种基因组数据库中的注释内容大多为编码蛋白质基因的翻译信息，与转录过程相关的信息还比较少，而转录起始又是整个转录过程的第一步。因此，TSS 的计算定位已经成为丰富基因组注释信息的重要手段，以及进行基因转录调控研究的基本前提。

研究表明，TSS 通常是嘌呤碱基（A 或 G），但位点附近序列的保守性并不强[1,2]。仅凭此单一特征，根本无法在序列中对其进行定位，必须与其它信号相结合。通常地，将从 TSS 上游延伸至其下游的长度为数百碱基的序列区域称为启动子（promoter）。它负责启动从 TSS 处起始的转录过程，是与 TSS 密切关联的转录调控信号。启动子的核心区域位于 TSS 的紧邻上游，鉴于两者在序列中的位置关系，当前的 TSS 定位基本上依赖于对应启动子识别的结果。针对启动子的识别，相关研究已陆续提出多种算法和工具[3,4]。已有的原核启动子识别方法可大致分成两类：基于组成（content）的方法和基于信号（signal）的方法。前者主要利用了启动子序列与背景序列在全局碱基组成上的差异，因此仅限于判断待定序列是否属于启动子区域，无法给出精确位置信息，也无法对 TSS 进行定位[7]。后者则通过发现启动子区域内特异的局部保守模式进行识别，可以对启动子的位置进行预测，并将识别结果的 3' 端位点近似作为 TSS。但由于启动子信号固有的微弱多变特性，即使对结构比较简单的原核启动子，识别算法的结果也不能完全令人满意[5,6]。困扰这类方法的主要问题是识别的特异性较低，从而导致大量的假阳性结果。例如有研究表明，每个真实大肠杆菌 σ^{70} 启动子附近平均预测有 38 个类似的启动子信号[8]。可见，当前算法的 TSS 定位水平还很低，如何增加信号特异性并利用位置信息将是提高定位精确度的关键。

在先前的研究中，我们以大肠杆菌和枯草杆菌启动子为例，提出了一种基于特征筛选的原核启动子识别算法[9]，在启动子的组成特征、信号特征和结构特征等备选特征中筛选出对识别贡献较大的特征组成启动子特征集，以此为基础构建二次判别分析（quadratic discriminant analysis，QDA）分类器，对待定序列是否为启动子序列做出判别。该算法在启动子的核心区域[TSS−60…TSS…TSS+20]内计算特征得分，并将各种特征信息进行综合，提高了启动子信号的特异性，在刀切法（jackknife）测试中获得了高于其它常用算法的识别正确率。

作为后续研究，本文将参考文献[9]中介绍的

收稿日期: 2006-01-05
基金项目：国家自然科学基金项目（60471003）
通讯作者：杜耀华，电话：(0731)4574991
E-mail：qsyaohua@nudt.edu.cn

启动子识别算法扩展应用于原核序列 TSS 的计算定位，其核心思想是把固定的启动子区域改成对应形式为[*S*–60…*S*…*S*+20]的滑动窗口，在基因翻译起始位点（translation start site, TLS）上游一定区域内的每个碱基位置上依次滑动，利用基于启动子特征集建立的二次判别分类器计算窗口序列的分类得分，将其作为窗口内 *S* 位置处碱基的 TSS 似然分值，再用 TSS 与 TLS 间的距离经验分布对此分值进行修正，最后根据待定位区域内分数的分布情况确定 TSS 的位置。为了提高定位的信噪比，算法特别为启动子特征集中的组分特征设计了一组窗口交迭组分变量，基于这些交迭变量建立单独的组分分类器，与由特征集中其余特征建立的分类器一起来综合计算 TSS 的似然分值。对大肠杆菌真实数据集的刀切法测试验证了定位算法的有效性。

1　数据与方法

1.1　数据集的选取

训练窗口分类器的序列数据来自文献[9]中整理的 683 条大肠杆菌 σ^{70} 启动子序列集。剔除位于编码区和 TSS-TLS 距离大于 350 bp 的序列，剩余的 580 条作为训练的正数据集。训练的负数据集则沿用文献[9]中非编码区负集的 612 条序列。训练集中每条序列的长度均为 81 bp，其格式为[TSS–60…TSS…TSS+20]。

与真核生物相比，原核生物的 TSS 与其下游对应 TLS 之间的距离较短。实验证实，在已知的大肠杆菌 TSS 中，位于从 TLS 至其上游 350 bp 区域之内的超过 90%[8,10]。因此，通常可以把 TSS 定位的范围限制在此区域内。根据这一原则，窗口[*S*–60…*S*…*S*+20]中位置 *S* 的可变范围为[TLS–350…TLS]，窗口序列扫过的总范围则对应为[TLS–410…TLS+20]。按此格式，我们依据训练正集来源数据库中提供的启动子对应 TLS 的位置信息，在大肠杆菌全基因组序列[11]中将训练正集的 580 条原长 81 bp 的序列扩展为长 431 bp 的序列，用以组成测试数据集，其中每条序列的 TSS 位置均为已知。

1.2　TSS 计算定位的流程

TSS 计算定位方法的总体流程如图 1 所示。

根据文献[9]中的特征筛选结果，位于非编码区的原核启动子，其特征集由组分特征（4 阶、6 阶词频）和其它特征（复合模式、TSS、-10 区延伸模式、UP 元件、局部转角和结合自由能）两部分组成。我们将原始的组分特征扩展为窗口交迭组分特征，然后分别基于交迭组分特征和其它特征建立二次判别分类器。窗口序列[*S*–60…*S*…*S*+20]中位置 *S* 的 TSS 似然得分（likelihood score）则由两个分类器得分之和组成。似然得分再与来自 TSS-TLS 距离经验分布的位置得分（position score）相结合得到 TSS 最终得分（final score）。根据最终得分的分布，由阈值定位算法即可确定 TSS 的预测位置。

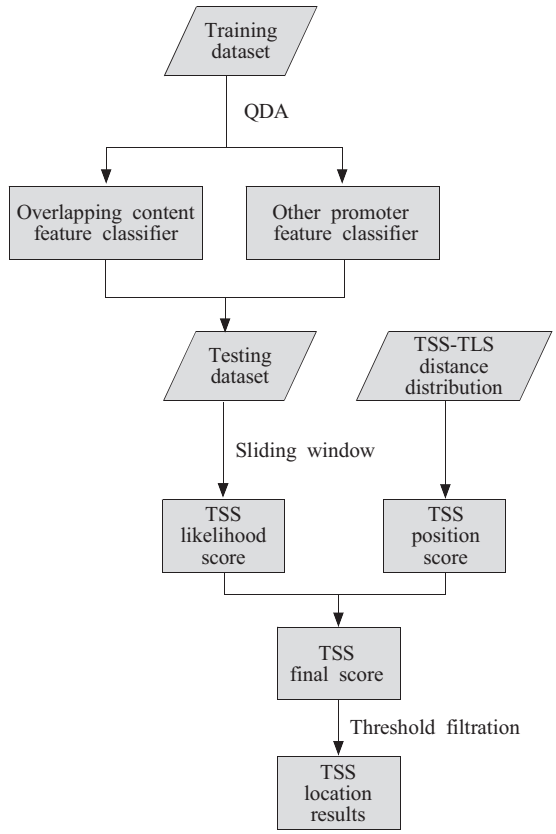

Fig.1　Flow chart of the location algorithm for TSS

1.3　交迭组分特征分类器

由文献[9]中的分析可知，在原核启动子特征集中，高阶词频和复合模式两种特征对启动子识别的贡献明显大于其它特征，进而成为决定 TSS 定位效果的主要因素。复合模式属于信号特征，本身含有位置信息。而词频则属于组分特征，单一的词频分析只能反映计算区域内全局性的组成偏好，受其影响，如果根据原有特征集直接建立滑动窗口分类器，其位置特异性将很低，使得定位结果中含有大量噪声。为了提高信噪比，可用一组计算区间相互交迭的词频特征变量代替原有的全窗口单一词频

特征变量进行计算，并将其从启动子特征集中分离出来，单独建立交迭组分特征分类器。交迭组分变量减少噪声的依据是所谓的“共振”原理[12]：检测结果中的真实信号位置相对固定，而噪声则倾向于随机出现，通过改变模型参数对同一信号区域进行多次检测，真实信号将随检测结果的迭加而增强，而噪声则因相互抵消而减小。依此原理，我们将窗口序列[S–60…S…S+20]划分为 8 个相互交迭的区间，在这些位置固定的区间上分别计算词频组分变量，如图 2 所示。

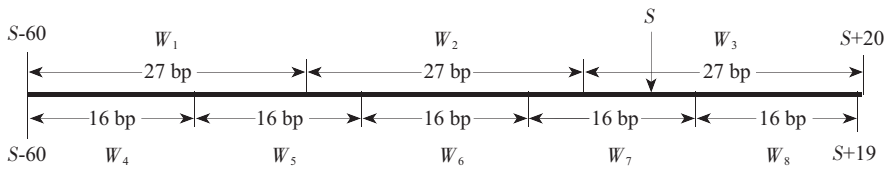

Fig.2 Overlapping content variables in sliding windows

由图 2 可知，组分变量 W_i(i=1,…,3)的计算区间长为 27 bp；W_i(i=4,…,8)的计算区间长为 16 bp。为保证计算具有统计意义，各组分变量对应的词频阶次 k 应满足[12]：

$$N(L-k+1)>4^k \quad (1)$$

其中 N 为参与计算的序列数目，L 为计算区间的长度。对于我们采用的数据集，根据式(1)，可取 k=6。6 阶词频特征的具体计算，以及基于 8 个词频特征变量建立交迭组分特征二次判别分类器的详细方法参见文献[9]。利用同样的方法，根据除去组分特征之后的启动子特征集，还可以建立启动子其它特征的二次判别分类器。

对于窗口序列[S–60…S…S+20]，交迭组分特征分类器和其它特征分类器的二次判别函数计算得到的似然分值，即可作为位置 S 相应的交迭组分特征得分 s_w 和其它特征得分 s_t。如果设位置 S 的 TSS 似然得分为 s_l，则有：

$$s_l=s_w+\alpha s_t \quad (2)$$

其中 α 为权重系数。两类特征得分来自同一窗口序列下的不同启动子特征集，在没有更多先验信息的情况下，不妨取 α=1。

1.4 TSS 与 TLS 的间隔距离分布

在选取数据集时已经提到，原核生物的 TSS-TLS 距离相对较短。以大肠杆菌为例，其 TSS-TLS 距离通常不超过 350 bp。根据训练正集中已知的位置信息，可以计算出每条序列的 TSS-TLS 距离值，由此即可统计得到 TSS-TLS 距离在[0,350]区间上的经验概率分布。统计的直方图以及平滑后的经验分布曲线见图 3。

由图 3 可知，TSS 在区间内并不是等概率分布的，而是更偏向位于离 TLS 较近的位置。图中的

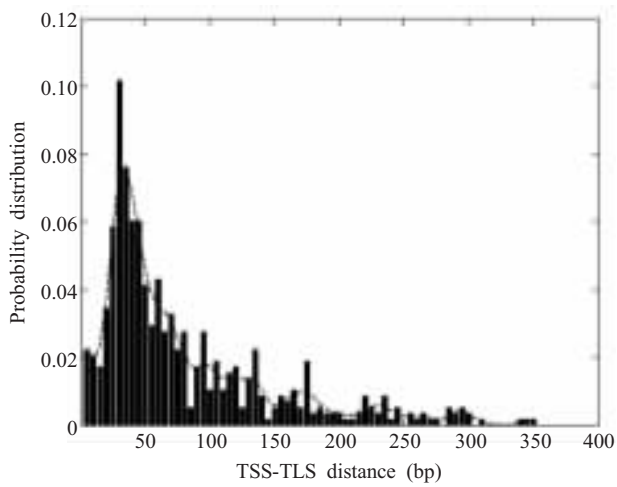

Fig.3 TSS-TLS distance histogram and smoothed empirical probability distribution for *E. coli*

经验分布仅在整数位置取值，设此时离散经验分布函数为 $D(x)$，与相应 TLS 距离为 x 的 TSS 位置得分 $s_d(x)$可由下式计算：

$$s_d(x)=\log[D(x)] \quad (3)$$

$s_d(x)$是 TSS 位置的一种先验信息，可用于对似然得分进行修正。

1.5 TSS 的阈值定位

将窗口序列[S–60…S…S+20]沿测试序列[TLS–410…TLS+20]滑动，每次前进 1 bp，则可计算 TLS 上游[0,350]区间（即[TLS–350…TLS]）内各个位置的 TSS 似然得分 s_l。由文献[9]和式(2)可知，s_l 本质上是 TSS 在对应位置出现概率的一种对数化度量。而由式(3)可知，s_d 也是对 TSS 先验概率的对数化度量。两种概率的相乘在对数变换之后则变成了相加。因此，在[0,350]区间内任取一个与 TLS 距离为 x 的位置，其 TSS 最终得分 s_f 为：

$$s_f(x)=s_l(x)+\beta s_d(x) \tag{4}$$

其中 β 为权重系数。为简单起见，可取 β=1。

根据区间内的 TSS 最终得分，TSS 定位结果由以下的阈值定位算法确定：

1) 引入得分阈值 C_s，扫描整个区间，记录每个 s_f 不低于 C_s 的位置，并将满足条件的位置称为“岛”；

2) 引入间隙阈值 C_α，将间隔距离未超过 C_α 的相邻岛合并；

3) 引入岛长阈值 C_l，淘汰长度小于 C_l 的岛；

4) 对剩余的岛，各岛内 s_f 值最高的位置即为 TSS 的预测位置。

2　结果与讨论

2.1　评价指标

TSS 定位常用的评价指标有敏感性（sensitivity, Sn）、特异性（specificity, Sp）和准确率（accuracy, AC）。定义 *TP* 为真实位点被正确定位的数目，*FP* 为虚假位点被定位为真实位点的数目（假阳性结果），*NP* 为真实位点的数目，则有：

$$Sn=\frac{TP}{NP} \tag{5}$$

$$Sp=\frac{TP}{TP+FP} \tag{6}$$

$$AC\%=\frac{Sn+Sp}{2}\times 100\% \tag{7}$$

2.2　阈值参数的确定

在实际的计算定位中，*Sn* 和 *Sp* 往往是互斥的，不可能同时达到最高，需要根据实际需求在两者之间寻找折中。通常的做法是在保证 *Sn* 达到一定水平的情况下，尽量提高 *Sp*。因此，得分阈值 C_s 可基于训练正集中真实 TSS 的 s_f 值分布情况（见图 4）来确定。

图 4 中的直方图为 580 条训练正集序列中，真实 TSS s_f 值的经验概率分布，平滑曲线则表示逆累积分布函数（inverse cumulative distribution function）。在横轴上选取阈值 C_s 时，其相应的逆累积分布曲线值表示 s_f 值高于此 C_s 的 TSS 在全部真实 TSS 中所占的比例，我们称之为阈值水平（threshold level）。它在一定范围内可近似作为算法所能达到的水平的期望值。在实际应用中，我们就是根据阈值水平来选取得分阈值 C_s 的。

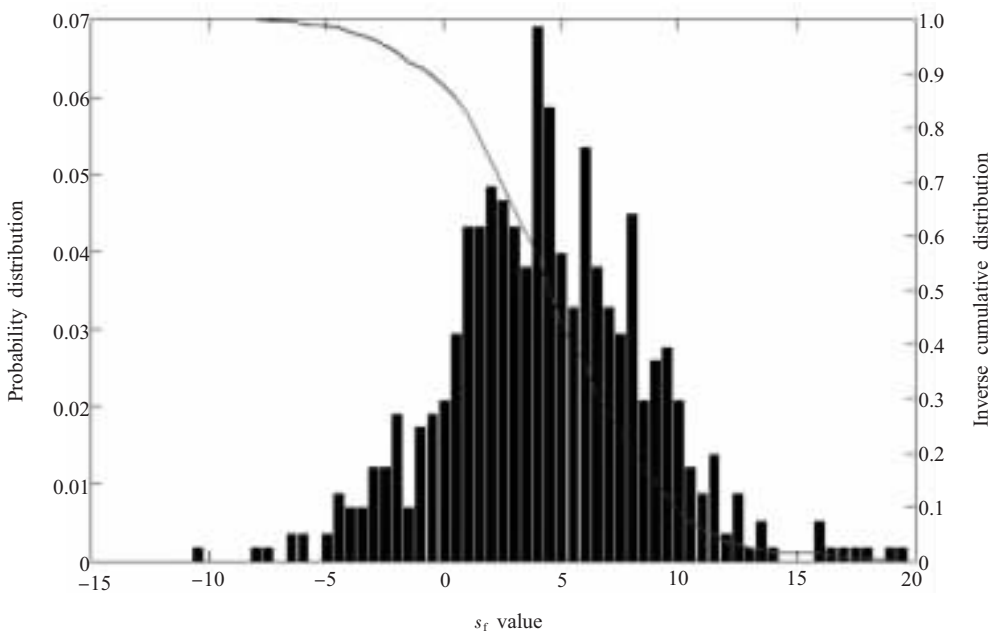

Fig.4　The distribution and inverse cumulative distribution curve of s_f for TSSs in positive training set

根据阈值水平选定得分阈值 C_s 后，间隙阈值 C_α 和岛长阈值 C_l 可用交叉验证（cross-validation）方法[13]，通过在训练集上优化定位准确率 *AC* 来确定，从而使算法达到最佳期望性能。

2.3　测试结果

我们用 **1.1** 中整理的训练集和测试集对 TSS 定位算法进行刀切法测试。刀切法是一种交叉验证方法，其测试结果是算法真实性能的无偏估计[14]。由

于我们的训练正集和测试集序列具有一一对应的关系，因此可按下述过程进行测试：每次从训练正集中留取一条序列不参与训练，利用其余的训练数据（包括正集和负集）训练定位算法，然后对留取序列所对应的测试集序列进行定位测试，并记录预测结果。重复此过程，将训练正集中的序列依次留取一遍，即可实现对测试集序列的全部测试。最后，综合各次的预测结果作为整个测试集的测试结果。

表 1 给出了定位算法在 5 种不同阈值水平 C_t 下的刀切法测试结果。需要指出的是，由于目前对 TSS 信号位点的认识比较有限，定位算法不可能对真实位点的位置做出完全精确的预测。即使是数据库中提供的真实位点也可能会因实验方法的局限而存在一定的位置误差。因此可以定义误差距离 d_t，只要预测位置落入区间[TSS–d_t,TSS+d_t]之内，即认为定位正确。在我们的算法测试中，d_t 取 5 bp。表中还同时列出了每种阈值水平 C_t 对应的得分阈值 C_s，以及由交叉验证获得的最优间隙阈值 C_α 和岛长阈值 C_l。

Table 1 Location results for *E. coli* TSSs in jackknife test

C_t	C_s	C_α (bp)	C_l (bp)	*Sn*	*Sp*	*AC* (%)
0.75	1.89	3	3	0.77	0.79	78.2
0.80	1.31	2	3	0.81	0.74	77.6
0.85	0.61	2	3	0.85	0.65	75.1
0.90	–0.67	1	3	0.88	0.51	69.1
0.95	–2.54	2	3	0.89	0.33	61.3

从表 1 中可知，当 $C_t \leqslant 0.85$ 时，C_t 值与算法实际的 *Sn* 水平大致相当，而 $C_t > 0.85$ 时，随着 C_t 值的增加，算法实际的 *Sn* 水平增长变缓，开始越来越落后于 C_t，其 *Sp* 水平也快速下降。因此，C_t 取 0.85 较为合适，此时算法的 *Sn* 为 0.85，*Sp* 为 0.65，*AC* 达到了 75.1%（误差距离为 5 bp）。文献[8]中提出了一种基于偏序覆盖函数（partial order cover function）的原核启动子识别算法，将预测结果的 3' 端位点作为 TSS。在 *Sn* 为 0.86 时, *Sn* 为 0.20，对应的 *AC* 仅为 53%（误差距离约为 5~15 bp）。与之相比，我们的算法减小了误差距离，在相近的 *Sn* 水平下，大幅度提高了 *Sp*，使得 *AC* 有了较明显的改进。

为了考察交迭组分特征分类器和 TSS-TLS 距离分布对提高定位算法性能所做的贡献，我们定义基于原始窗口分类器（由原始启动子特征集构建）的定位算法为算法Ⅰ，基于交迭组分特征分类器 + 启动子其它特征分类器的算法为算法Ⅱ，基于交迭组分特征分类器 + 启动子其它特征分类器 +TSS-TLS距离分布的算法（即本文的定位算法）为算法Ⅲ，在相同的数据集情况下对三种算法进行刀切法测试，比较其在同一 C_t（≤0.85）下性能指标的变化，结果见表 2。

测试结果表明，在同一 C_t 下，随着交迭组分特征分类器和 TSS-TLS 距离分布的引入，定位算法的各项性能指标（尤其是 *Sp*）有较明显的改进，进一步证实了它们对减少假阳性结果和提高定位信噪比的有效性。

Table 2 Location results of different algorithms for *E. coli* TSSs in jackknife test

C_t	Algorithm Ⅰ			Algorithm Ⅱ			Algorithm Ⅲ		
	Sn	*Sp*	*AC* (%)	*Sn*	*Sp*	*AC* (%)	*Sn*	*Sp*	*AC* (%)
0.70	0.69	0.37	53.2	0.70	0.78	73.9	0.74	0.83	78.3
0.75	0.72	0.32	52.3	0.74	0.74	74.2	0.77	0.79	78.2
0.80	0.76	0.26	50.9	0.78	0.69	73.8	0.81	0.74	77.6
0.85	0.79	0.20	49.4	0.82	0.62	71.8	0.85	0.65	75.1

3 结 论

TSS 的计算定位通常与启动子的识别密切相关，且当前定位算法的预测能力还远不能令人满意。本文在已有原核启动子识别算法的基础上，提出了一种基于滑动窗口的原核序列 TSS 计算定位方法，通过在合理限定的 TSS 定位范围内对序列进行滑动扫描来预测 TSS 位置。为提高定位的信噪比，在计算窗口序列对应位置的似然得分时，综合利用了根据启动子特征集建立的交迭组分特征分类器和其它特征分类器。另外，作为定位依据的各位置 TSS 最终得分中结合了 TSS-TLS 距离分布信息，进一步减少了假阳性结果。对大肠杆菌真实数

据的刀切法测试验证了定位算法预测真实 TSS 位置的能力和相比已有算法的优越性。

应该看到，算法的实际结果与 TSS 的完全精确定位还有一段距离。相信随着对 TSS 信号研究的深入，将有更多的特征信息和数据被发现和利用，这一差距将会不断缩小。另外，相关研究也已经指出，基因组中许多基因对应的 TSS 并不唯一，而是存在多个备选位置。因此当前定位算法得到的假阳性结果中有些可能是潜在的备选 TSS 位点，并不是真正的假阳性，这还需要进一步的实验去证实。最后，定位算法本身也需要不断改进和优化，交迭组分变量的设计、滑动窗口长度及三类得分权重系数 α 和 β 的选取等问题还可深入探讨和研究。这将是我们下一步工作的重点。

参考文献：

[1] Harley C, Reynolds R. Analysis of *E.coli* promoter sequences. *Nucleic Acids Research*, 1987,15(5):2343~2361

[2] Lisser S, Margalit H. Compilation of *E.coli* mRNA promoter sequences. *Nucleic Acids Research*, 1993,21(7):1507~1516

[3] Werner T. Models for prediction and recognition of eukaryotic promoters. *Mammalian Genome*, 1999,10(2):168~175

[4] Ohler U, Niemann H. Identification and analysis of eukaryotic promoters: recent computational approaches. *TRENDS in Genetics*, 2001,17(2):56~60

[5] Hertz G, Stormo G. *Escherichia coli* promoter sequences. Analysis and prediction. *Meth Enzymol*, 1996,273:31~42

[6] Vanet A, Marsanc L, Sagot M. Promoter sequences and algorithmical methods for identifying them. *Res Microbiol*, 1999, 150(9-10):779~799

[7] Gordon L, Chervonenkis A, Gammerman A, Shahmuradov I, Solovyev V. Sequence alignment kernel for recognition of promoter regions. *Bioinformatics*, 2003,19(15):1964~1971

[8] Huerta A, Collado-Vides J. Sigma70 promoters in *Escherichia coli*: specific transcription in dense regions of overlapping promoter-like signals. *J Mol Biol*, 2003,333(2):261~278

[9] 杜耀华, 王正志, 倪青山, 李冬冬. 一种基于特征筛选的原核生物启动子判别分析方法. 生物物理学报, 2006,22(1):39~48

[10] Burden S, Lin YX, Zhang R. Improving promoter prediction for the NNPP2.2 algorithm: a case study using *Escherichia coli* DNA sequences. *Bioinformatics*, 2005,21(5):601~607

[11] Blattner F, Plunkett G, Bloch C, Perna N, Burland V, Riley M, Collado-Vides J, Glasner J, Rode C, Mayhew G, Gregor J, Davis N, Kirkpatrick H, Goeden M, Rose D, Mau B, Shao Y. The complete genome sequence of *Escherichia coli* K-12. *Science*, 1997,277:1453~1462

[12] Zhang MQ. Identification of human gene core promoters in silico. *Genome Research*, 1998,8(3):319~326

[13] Wahba G, Lin Y, Zhang H. Margin like quantities and generalized approximate cross validation for support vector machines. In: Wilson E, Douglas S. Proceeding of the IEEE signal processing society workshop on neural networks for signal processing. Madison, USA: IEEE Computer Society Press, 1999. 12~20

[14] Varma S, Simon R. Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, 2006,7:91

COMPUTATIONAL LOCATION OF TRANSCRIPTION START SITES IN PROKARYOTIC GENOME BASED ON SLIDING WINDOW

DU Yao-hua, WANG Zheng-zhi, NI Qing-shan

(*College of Mechatronics Engineering and Automation, National University of Defense Technology, Changsha 410073, China*)

Abstract: Although a great deal of effort has been undertaken in the area of transcription start site (TSS) computational location due to its essential role in the research of transcription regulation, the problem has not yet been resolved. According to the previous work on prediction algorithm of prokaryotic promoters, a new computational location method for prokaryotic TSSs based on sliding window was proposed. At first, the authors limited the rational searching ranges in genomic sequences based on the prior information of TSSs occurrence. Then the TSS likelihood scores of each possible position in genomic sequences were calculated by two window classifiers which were trained by quadratic discriminant analysis on overlap content features and other promoter features, respectively. The empirical distribution of distances between TSSs and translation start sites (TLSs) was also utilized to amend the likelihood scores. Final location results were achieved through the procedure of threshold filtration on the likelihood score profiles. The testing results on *E. coli* datasets showed that the method could find the putative TSSs efficiently. Compared with other current algorithms, the specificity *Sp* could be improved from 0.20 to 0.65 when the sensitivity *Sn* was about 0.85, which made the location accuracy increasing by about 20 percents.

Key Words: Prokaryotic genome; Transcription start site (TSS); Computational location; Sliding window; Overlap content features

---

This work was supported by a grant from The National Natural Science Foundation of China (60471003)

Received: Jan 5, 2006

Corresponding author: DU Yao-hua, Tel：+86(731)4574991，E-mail：qsyaohua@nudt.edu.cn