

工作汇报

BUSINESS POWERPOINT



汇报人：唐玉璇



01

PART ONE

介绍

6mA siter

non-6mA site

nucleotide	A C G T	
Label	1	0
Sequence	ACAGCTAGATCGTTTCGAAATGGCGGGCACATTGGGGACTAG	
Sample	<i>Arabidopsis thaliana</i> (19616 6mA + 19616 non-6mA = 39232) <i>Drosophila melanogaster</i> (10653 6mA + 10653 non-6mA = 21306)	

Dataset

Feature sequence	Label
GTTGACAGCTAGATCGTTTCGAAATGGCGGGCACATTGGGGA	1
AAACTCAATTCCCGCACTATAAACATCGCAAACTAATGATT	0
GAAGCATGAATCTGACTTGGAAATGCGCGTGTTAAAGTGGCT	1
GCCCTCGAGGCGGGTGCCGTAGGCAGCGTTCGCGAGGCGAC	1
AAATAATAAGAAATTTGCCACTCCTAACTTAATGCAACAA	0
TAAGAAGAAGAAATGCGGGGAAAAAGCGGGTGCTGCAGCTG	1
GGTTATAGATTTTAAAAACGAGGTATCGTTCCTTTTATATA	1
CAACACCTTAAAAATTTAATAGAATGCAACTTAACCGAACT	0
CAGACGGACGGACAGTCCGAACGAACGATTCAAACCTCGA	1
ACTGGGTTTTTGGCCCAAGGAGGAAGTTTACAGCATCTAA	0
ATTATTTGCAAAATTTTTGATATAGTGACAATTTTGAAA	1
TACTTTTCTCTTTTGAAAAATAAATCGTGGACTTTAAG	0
TTACAAAGGAAACAAATTATAAAATTATAATATATTTTAGA	0
GGAAATTTCTCAGCCACCGGAGGTATCAGCTCTTAACTTC	1
TGTGAGCTTTCAATACCTTTATTGTTTCTTTTACTCAAT	0
TTTCGACGAAGAAGATGATGATGATGAGGACGGGTTAATG	0
AAATAATCTCAATCTAAGCTATTTCAAGTAACCAACAATACT	0
GCATGCTTTAAAGGATCAGCAGGTGTCATCTTCATGAGAAA	1
ATAAGACCCAAAAAATCAGGAGGTAGAAAGATCTACATGCA	1
TGTTTGTACTCTGTTTGGAGAGACTGTGTCATTTGATAGAG	0

三邻近特征

① onehot编码: $A=[1,0,0,0]$ $C=[0,1,0,0]$ $G=[0,0,1,0]$ $T=[0,0,0,1]$

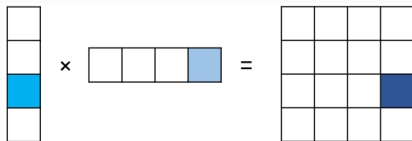


② 二邻近编码: 当前核苷酸与后一个核苷酸组合:

AA, AC, AG, AT, CA, CC, CG, CT, GA, GC, GG, GT, TA, TC, TG, TT

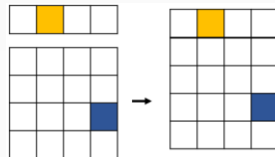
Eg: $GT = (G \text{ 的 onehot 编码})^T \times (T \text{ 的 onehot 编码})$

得到 4×4 的特征向量

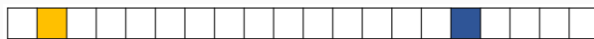


③ 三邻近编码: 当前核苷酸与前一个后一个核苷酸一共64种组合

Eg: CGT



④ 将特征拉平: 最终得到20维的特征向量



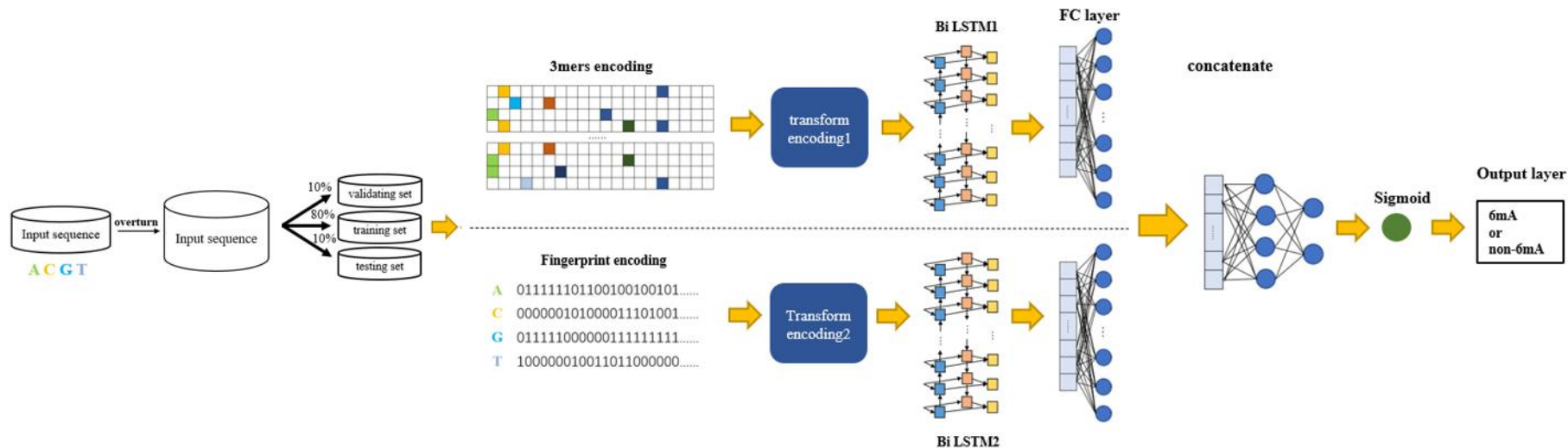
由于第一个核苷酸和最后一个核苷酸分别没有前一个和后一个核苷酸, 因此舍弃, 从第二个核苷酸开始编码, 直到第40个核苷酸

最终每条序列转化为 39×20 的特征向量

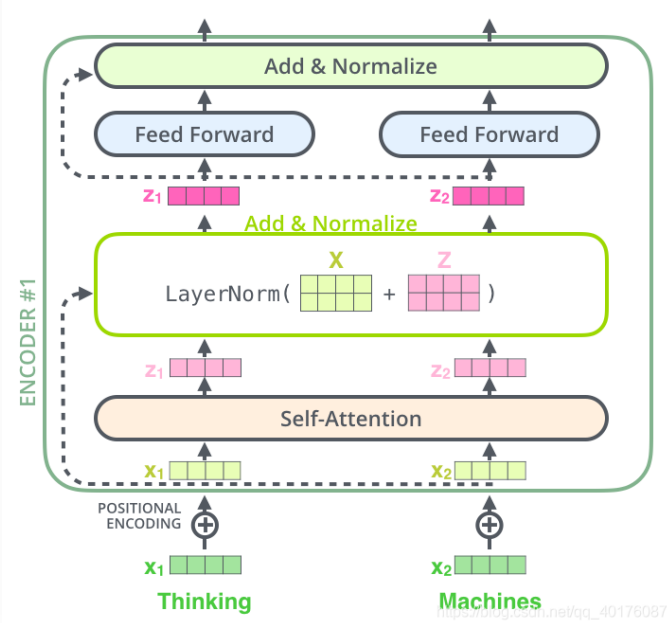
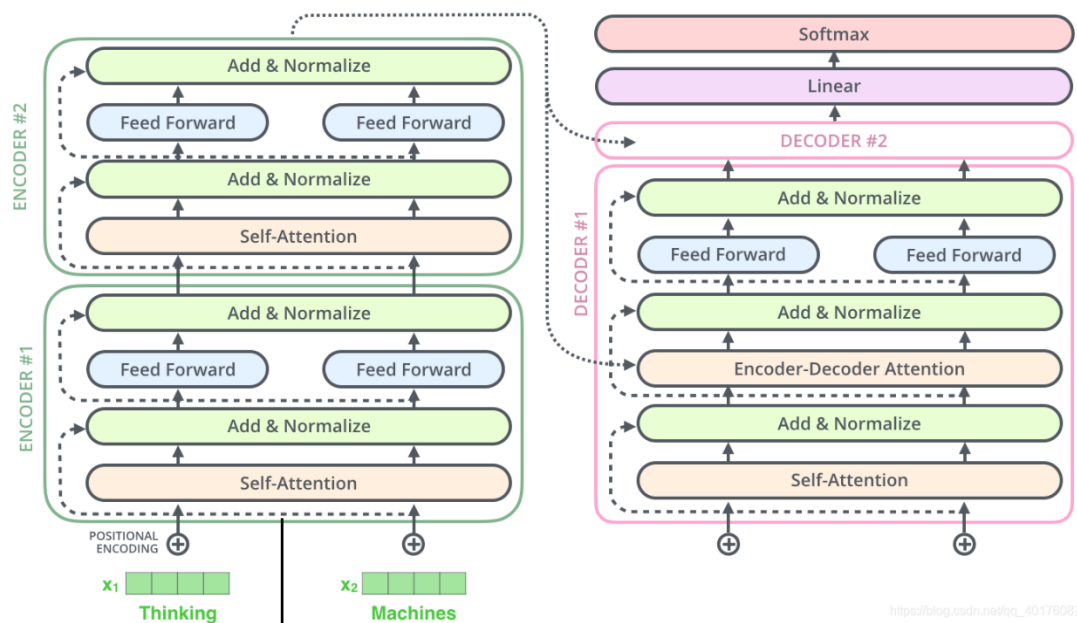
指纹特征

A 011111101100100100101.....
C 000000101000011101001.....
G 011111000000111111111.....
T 100000010011011000000.....

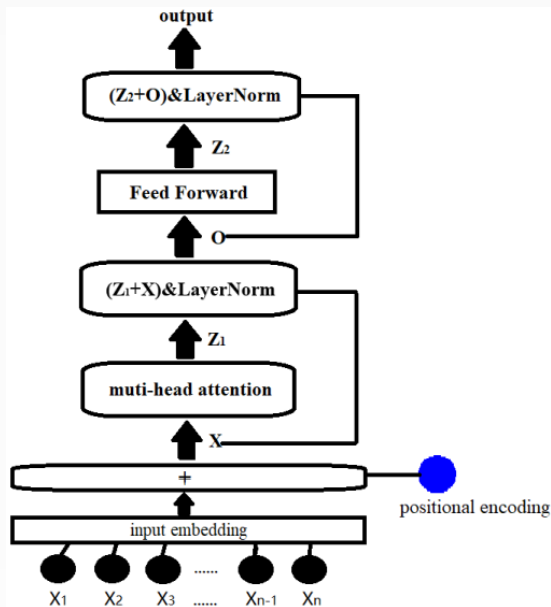
最终转化成 41×71 的特征向量



Overturn: 采用翻转的数据增强方法，既在原始数据集中加上每条序列翻转后的序列，最终得到两倍的数据量

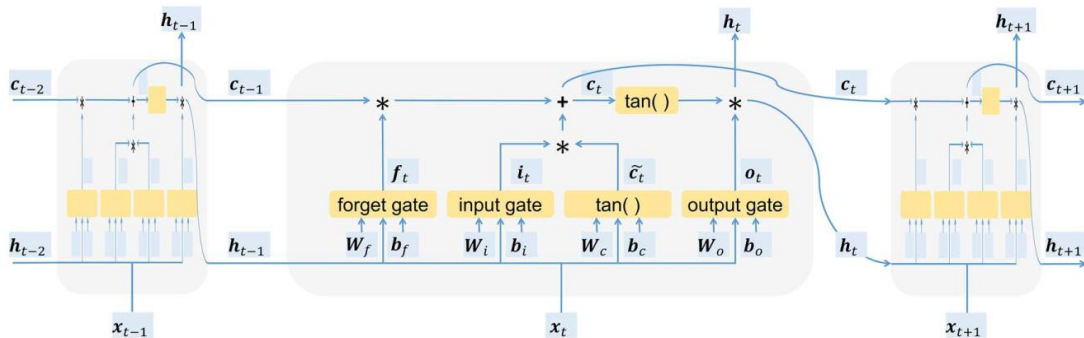


应用transformer的编码层



LSTM作为译码层

+



编码器得到每个词的上下文表示，相当于序列标注

在自然语言处理和计算机视觉任务当中，transformer 在连接长范围数据依赖性方面非常有效，但在 RL 学习中，transformer 难以训练并且容易过拟合。相反，LSTM 在 RL 中已经被证明非常有用。尽管 LSTM 不能很好地捕获长范围的依赖关系，但却可以高效地捕获短范围的依赖关系。

02

PART TWO

实验结果

单独三邻近特征

Dataset	model	MCC
Arabidopsis thaliana	TL	0.817/0.826
Drosophila melanogaster	TL	0.8311/0.841

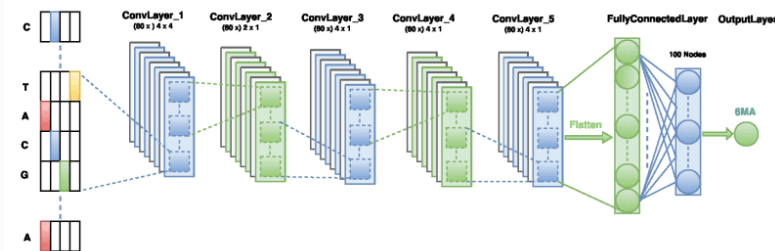
单独指纹特征

Dataset	model	Mcc
Arabidopsis thaliana	LT	0.8145/0.826

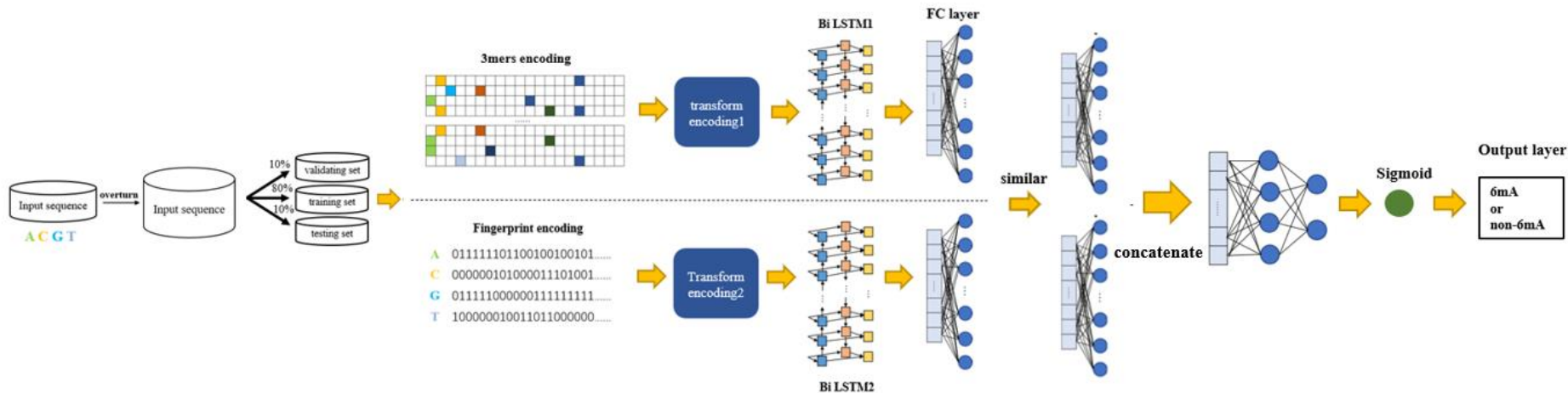
三邻近特征+指纹特征

Dataset	model	Sen	Spe	Mcc
Arabidopsis thaliana	TL	0.8607/0.899	0.9204/0.931	0.8068/0.826
	LT			0.7878/0.826
Drosophila melanogaster	TL	0.8287/0.909	0.9441/0.939	0.7788/0.841
	LT			0.8211/0.841

Dataset	Method	Sen ¹	Spe ¹	Acc ¹	MCC ¹	AUROC	Sen ²	Sen ³
Arabidopsis thaliana	DeepM6A ^a	0.894	0.931	0.913	0.826	0.966	0.920	0.956
	i6mA-DNC ^b	0.846	0.909	0.878	0.757	0.944	0.853	0.912
	iDNA6mA ^c	0.843	0.889	0.866	0.733	0.932	0.833	0.902
	3-mer-LR ^d	0.669	0.728	0.699	0.397	0.773	0.411	0.577
	LA6mA	0.899	0.917	0.909	0.817	0.962	0.912	0.948
	AL6mA	0.862	0.905	0.884	0.768	0.945	0.867	0.927
Drosophila melanogaster	DeepM6A ^a	0.901	0.939	0.920	0.841	0.969	0.930	0.959
	i6mA-DNC ^b	0.869	0.917	0.893	0.787	0.947	0.878	0.916
	iDNA6mA ^c	0.883	0.843	0.863	0.727	0.937	0.846	0.904
	3-mer-LR ^d	0.680	0.702	0.691	0.383	0.753	0.347	0.558
	LA6mA	0.909	0.915	0.912	0.824	0.966	0.921	0.955
	AL6mA	0.840	0.916	0.878	0.758	0.941	0.848	0.920



Supplementary Figure 1. The proposed network architecture of DeepM6A, a method for predicting DNA modification on N6-Adenine.



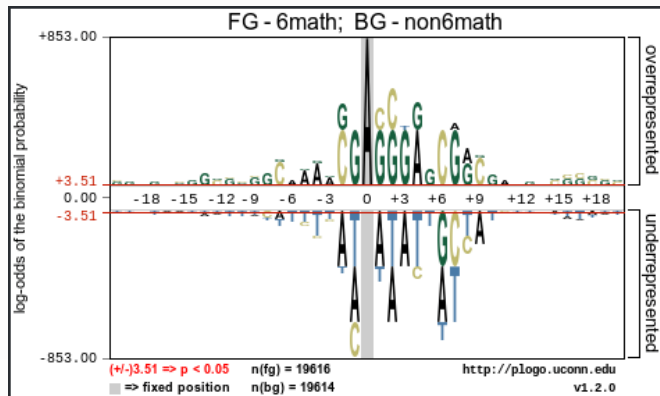
三邻近特征+指纹特征

Dataset	model	Sen	Spe	Mcc
Arabidopsis thaliana	TL	0.8607/0.899	0.9204/0.931	0.7519/ 0.826

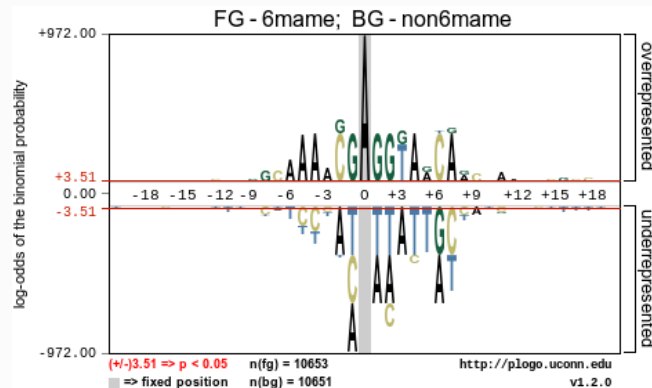
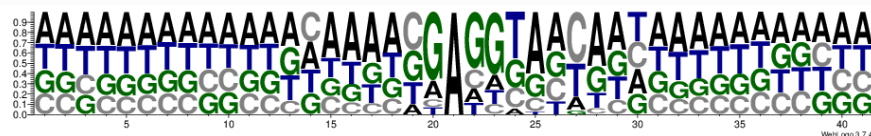
网络不够深，加深网络

➤ 中心为6mA位点的DNA序列

Arabidopsis thaliana



Drosophila melanogaster



中心位点都为核苷酸A，因为只有A才可以被甲基化，当中心位点为6mA位点时，相邻四个核苷酸基本都是CGAGG

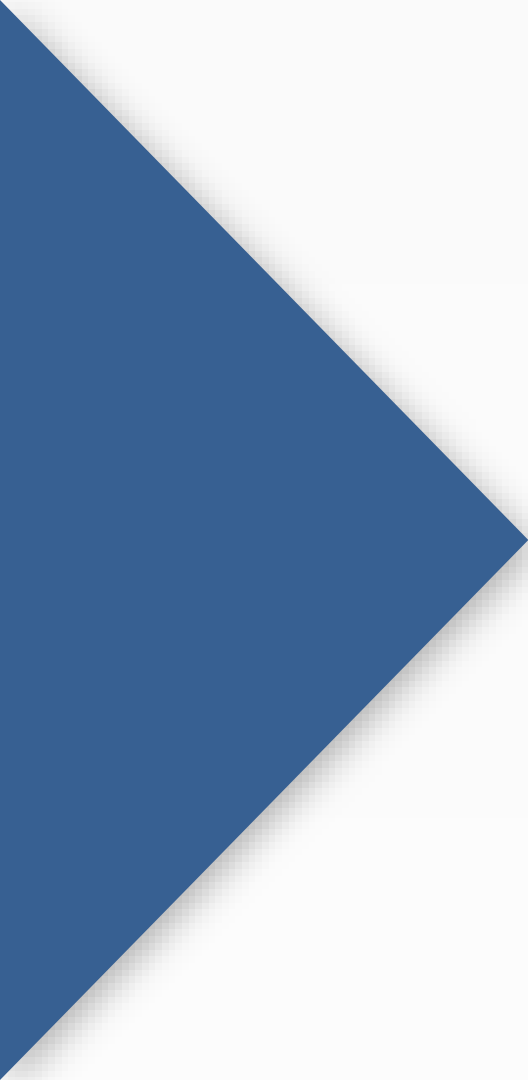
论文汇报

PAPER POWERPOINT



汇报人：唐玉璇

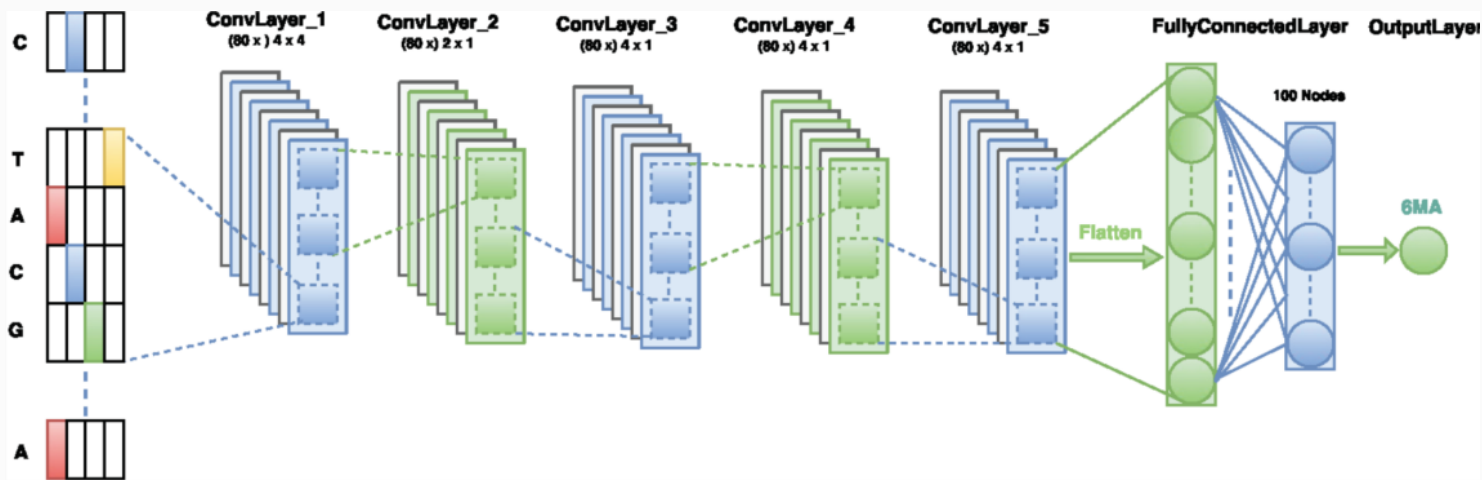


A large blue triangle pointing to the right, located on the left side of the slide.

Elucidation of DNA methylation on *N*6-adenine with deep learning

非腺嘌呤 DNA 甲基化的深度学习阐明

DeepM6A



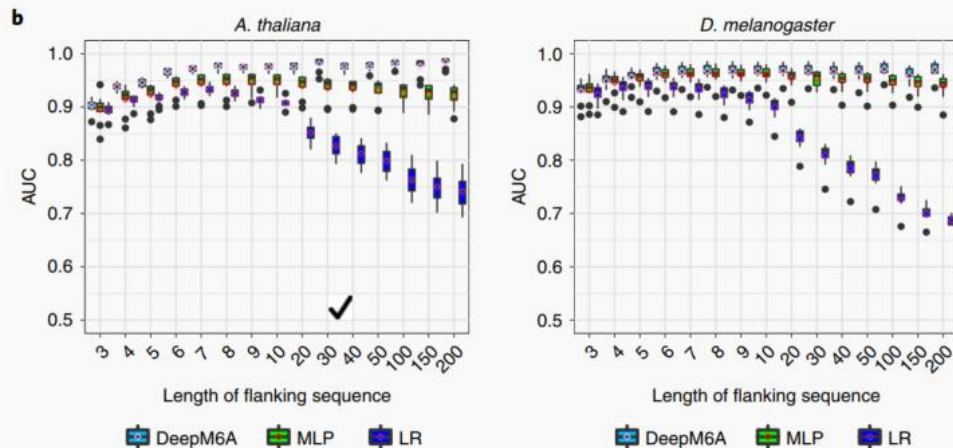
A. thaliana: 19,632个6mA位点序列

D. melanogaster : 10,653个6mA位点序列

E. coli: 33,700个6mA位点序列

再从数据库里随机挑选和正样本相同数量的负样本, 要求负样本和正样本中间的位点至少200bp远

比较位点上下游从3-200dp的预测结果，即序列长度为7-401



MLP: 5个全连接神经网络, 特征是one-hot

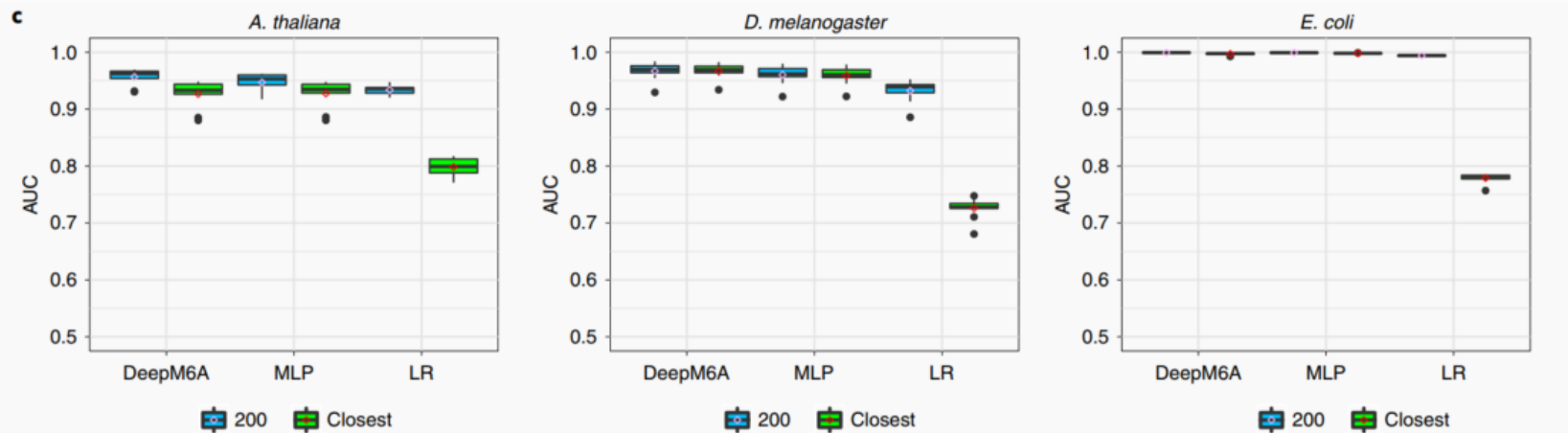
LR: 经典的基于k-mer的逻辑回归模型

6mA 位点的上/下游7-10bp 区域是至关重要的

除了10个 bp 的位置之外, 可能还有其他微妙和或复杂的信号

由于10dp后性能增加, 因此这种远距离的信号可以被 DeepM6A 捕获, 对基于 k-mer的方法是有害的

为了显示 DeepM6A 具有单核苷酸敏感性的稳健性，对于每个6ma位点，选择其最接近的和距离200dp的非N6-甲基化腺嘌呤，三种方法进行性能比较



DeepM6A 可以将其应用于单核苷酸的6MA 预测

将阴性样本混合到阳性样本中以模仿假阳性

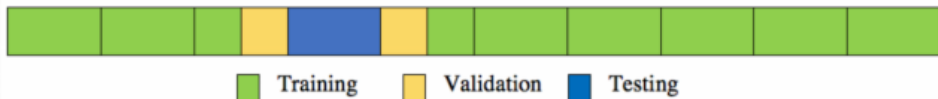
%Label Noise	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
<i>A. thaliana</i>	0.9521	0.9475	0.9407	0.9380	0.9298	0.9296	0.9282	0.9166	0.9054	0.8630
<i>D. melanogaster</i>	0.9619	0.9583	0.9549	0.9493	0.9472	0.9447	0.9343	0.9278	0.9154	0.8665
<i>E. coli</i>	0.9993	0.9987	0.9987	0.9987	0.9984	0.9975	0.9950	0.9916	0.9888	0.9870

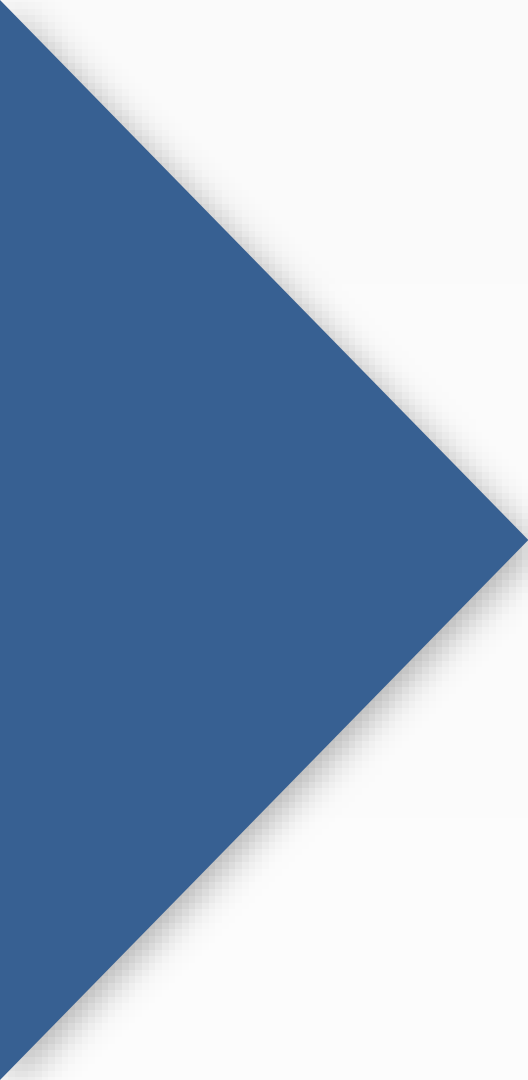
对于 *A. thaliana*, *d. melanogaster* 和 *E. coli*, DeepM6A 的AUC仅分别减少了0.0141, 0.0126和0.0001
当假阳性位点的百分比小于50% 时, DeepM6A仍然可以精确地捕获序列模式

为了促进模型参数的有效调整和合理评估, 采用了三方位数据分割策略

将数据集分割成10个不同的片段, 选择一个片段作为测试集, 验证数据集被设置为相应的最近的上游和下游一半片段的组合, 其余的站点被用于训练。

使用这种空间分割来保证训练和测试部分是严格不重叠的。随机选择训练, 验证和测试数据集的8:1:1比例, 它们的侧翼序列将有可能彼此重叠。预测性能和获得的基序模式可能有些误导



A large blue triangle pointing to the right, located on the left side of the slide.

Predicting protein–peptide binding residues via
interpretable deep learning

通过可解释的深度学习预测蛋白质-多肽绑定定位点

一种可解释的新的基于BERT（Transformer的双向编码器表示法）的对比学习框架 PepBCL来预测仅基于蛋白质序列的蛋白质肽结合残基

在大规模高通量的预测中，计算方法有以下限制：

①大多数蛋白质肽复合物结构是未知的，现有的基于结构的预测方法高度依赖于第三方计算工具预测的结构信息，这容易导致预测效率低下，并带来一些影响预测性能的噪声信息，且很多基于序列的预测方法也依赖于第三方计算工具预测的信息

进化信息来自位置特定评分矩阵 (pssm)，这是通过运行查询蛋白质和大规模蛋白质数据库之间的序列比对生成的

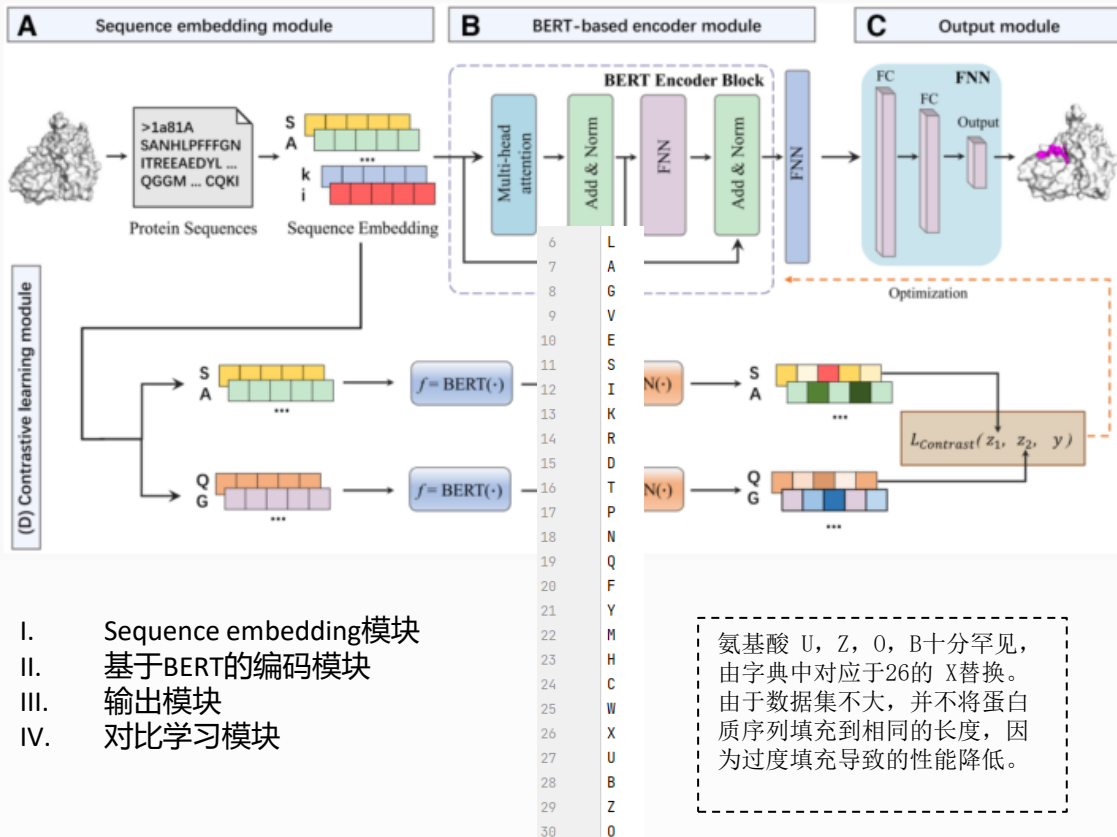
②现有的方法严重依赖手工特征设计来建立模型，对研究人员的专业知识要求很高，而且手工特征设计在一定程度上缺乏适应性

③现有的方法不能很好地解决蛋白质-肽结合残基预测的数据不平衡问题，容易导致整体性能差

④深度学习模型缺乏可解释性

两个数据集分开使用，即由TR1154训练集上训练的模型只在TE125上评估测试，由TR640训练集上训练的模型只在TE639上评估测试。

[illegible]



❑ 先输入查询蛋白质序列, 将每个残基编码成embedding向量

根据定义的词汇字典, 将原始蛋白质序列翻译成数字序列, 其中序列中的每个氨基酸可以被视为句子中的一个单词并映射到数字值, 将编码后的向量嵌入到一个查找表中, 查找表是一个嵌入层

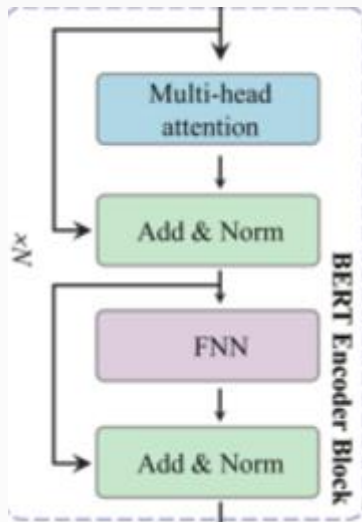
❑ 再通过预训练BERT对embedding矩阵进行编码, 生成高维代表向量

❑ 然后将代表向量输入全连接神经网络层, 可以生成更加密集的表达

❑ 之后通过对比学习模块, 计算和优化训练集中任意两个训练样本的对比损失, 反向传播回BERT模块进行优化, 从而获得绑定位点和非绑定位点更加有区别性的表示。

❑ 最后通过全连接神经网络输出层获得最终的多肽绑定位点概率值

包含了一个多头注意力机制，一个FNN，一个残差连接技术



就是transformer的编码层

多头注意力机制

由多个独立的自注意力机制组成

$$\begin{cases} Q = XW^Q \\ K = XW^K \\ V = XW^V \end{cases}$$

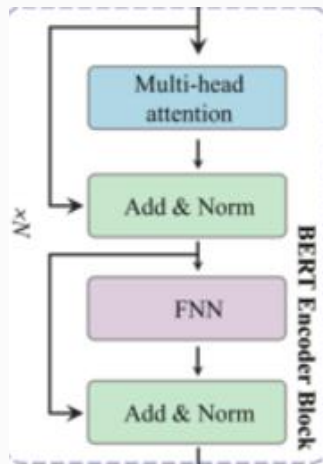
$$\text{Self_Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V,$$

X 为输入序列， W^Q, W^K, W^V 分别为 Q, K, V 对应的权重矩阵， Q, K, V 分别为查询矩阵，键矩阵，价值矩阵

$$\begin{cases} \text{Head}_i = \text{Self_Attention}(XW_i^Q, XW_i^K, XW_i^V), i = 1, \dots, b \\ \text{MultiHead_Attention}(Q, K, V) = [\text{head}_1, \text{head}_2, \dots, \text{head}_b] W^O \\ X_{\text{MultiHead}} = \text{LN}(\text{MultiHead_Attention}(Q, K, V) + X) \end{cases}$$

h 为多头注意力机制的头数， W^O 是一个线性转换层，可以将多头注意力的输出维数映射到嵌入模块的初始嵌入维数
再输入到残差连接和归一化层

多头注意力机制的输出为 $X_{\text{MultiHead}}$ （多头注意力层+残差连接+归一化）



$$\begin{cases} \text{FNN}(X_{\text{MultiHead}}) = \text{gelu}(X_{\text{MultiHead}} W^{(1)}) W^{(2)} \\ X_{\text{FNN}} = \text{LN}(\text{FNN}(X_{\text{MultiHead}}) + X_{\text{MultiHead}}) \end{cases}$$

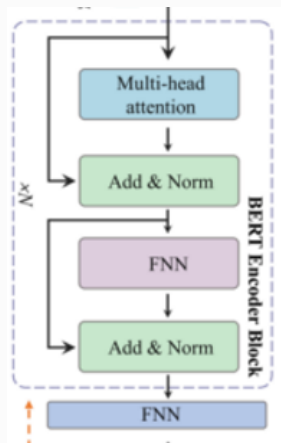
$W^{(1)}$, $W^{(2)}$ 两层线性层的权值, 激活函数为 gelu, 再经过残差连接和归一化层

FNN的输出为 X_{FNN} (FNN层+残差连接+归一化)

由于BERT模型不仅仅只有一层, 它是编码模块的堆积, 因此最后BERT的输出为

$$X^{(i)} = \text{FNN}(\text{MultiHead}(X^{(i-1)})), i = 1, \dots, n$$

n 代表有 n 层编码, $x(i)$ 代表第 i 层编码模块的输出, $X^{(0)}$ 代表初始输入的embedding矩阵, 最终BERT模块的输出为 $X^{(n)}$



由于BERT模型输出维度很高, 为了避免维度冗余, 再连接一个FNN层, 在降低维数的同时, 更好地提取输入序列中氨基酸本身的表示

基于监督式学习的对比学习模块，使相同的类的输入映射到表示空间中邻近点和不同类输入映射到表示空间中的远处。即使同一类的样本具有相似的表示而不同的类样本具有不同的表示，构建了对比损失作为模型的损失函数之一

由于没有将蛋白质序列填充到相同的长度，按批量大小从编码器模块收集表示矩阵。

对于一个batch的一对表示矩阵 z_1, z_2 的损失：

$$\begin{cases} D(z_1, z_2) = 1 - \cos \angle z_1, z_2 > \\ \mathcal{L}_{\text{contrast}}(z_1, z_2, y) = \frac{1}{2}(1-y)D(z_1, z_2)^2 + \frac{1}{2}y\{D_{\max} - D(z_1, z_2)\}^3 \end{cases}$$

Cosine为余弦相似度， $\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|}$ ， $D(z_1, z_2)$ 代表着两个残基代表对之间的距离

当 z_1, z_2 为同类时， $y=0$ ，即 $\mathcal{L}_{\text{contrast}}(z_1, z_2, y) = \frac{1}{2}D(z_1, z_2)^2$

当 z_1, z_2 为不同类时， $y=1$ ， $\mathcal{L}_{\text{contrast}}(z_1, z_2, y) = \frac{1}{2}\{D_{\max} - D(z_1, z_2)\}^3$ 。
 $D_{\max}=2$ 。

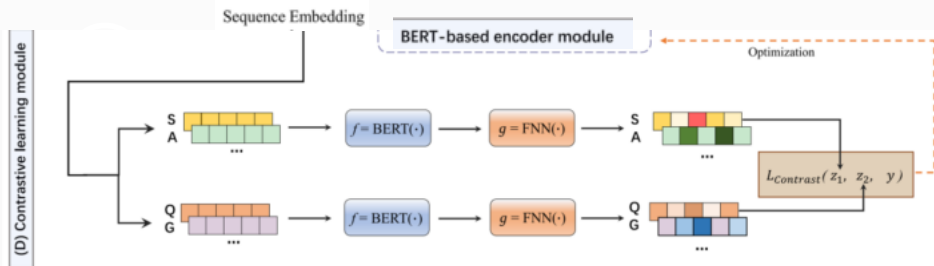
计算一批次残差数据的前半部分和后半部分的对比损失。最小化对比损失，使得从基于BERT编码模块获得一个更有辨别性和好提取的每个残基蛋白质序列表示

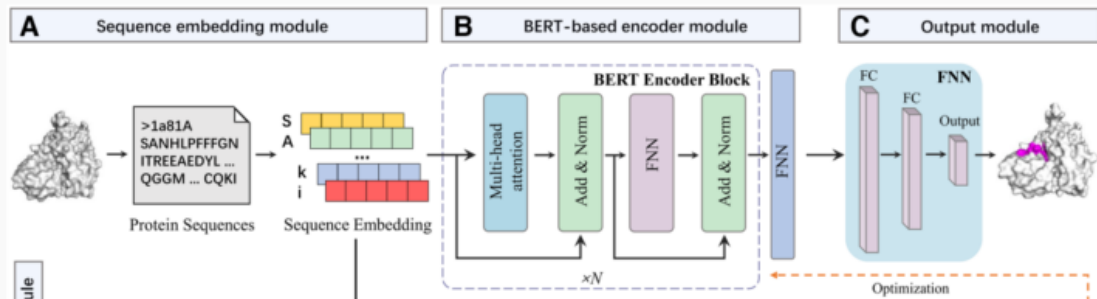
Algorithm 1: Contrastive loss in one batch

Input: X : the input protein sequences, $Label$: the labels of the input protein sequences, $Embed$: Sequence embedding module, $Encode$: BERT-based encoder module, M : the batch size, N : the number of residues in a batch, Z : the representations of residues in a batch, Y : the labels of residues in a batch;

```

for  $i = 1, \dots, M$  do
     $X_{Embed,i} = Embed(X_i)$ 
     $X_{Encode,i} = Encode(X_{Embed,i})$ 
     $Z = \text{concat}(Z, X_{Encode,i})$ 
     $Y = \text{concat}(Y, Label_i)$ 
end
for  $j = 1, \dots, N//2$  do
     $z_j = Z[j]$ 
     $z_{N//2+j} = Z[N//2+j]$ 
     $y = Y[j] \wedge Y[N//2+j]$ 
     $L_1 = L_1 + \mathcal{L}_{\text{contrast}}(z_j, z_{N//2+j}, y)$ 
end
    
```





$$\begin{cases} x_{\text{Encode}} = \text{BERT_based_Encode}(\text{Sequence_Embed}(x)) \\ y_p = \text{FNN}(z), z \in \{x_{\text{Encode},i} | i = 1, \dots, n\} \end{cases}$$

对于输出模块使用交叉熵损失函数 L_{CE} ，一共两个损失函数

$$\begin{cases} L_1 = \sum_i \mathcal{L}_{\text{contrast}}(Z_i, Z_{N//2+i}, y) \\ L_2 = \sum_j L_{\text{CE}}(Z_j, y_j) \\ L = L_1 + L_2 \end{cases}$$

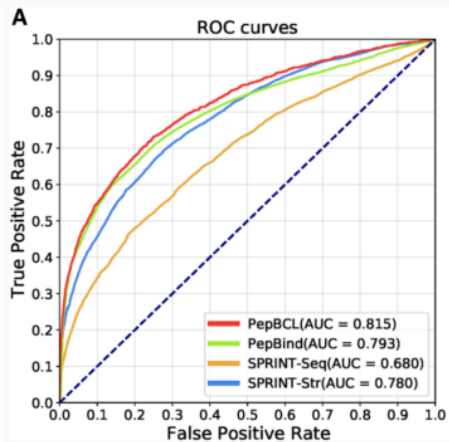
为了避免损失 L_2 的反向传播干扰BERT模块的残基表示学习，导致梯度消失，将表示学习部分和预测部分的优化分开，在训练输出模块时冻结了BERT模块中的参数

```
loss1.backward(retain_graph=True)
```

TE125 test set

Methods	Recall	Specificity	Precision	AUC	MCC
Pepsite*	0.180	0.970	—	0.610	0.200
Peptimap*	0.320	0.950	—	0.630	0.270
SPRINT-Seq	0.210	0.960	—	0.680	0.200
SPRINT-Str*	0.240	0.980	—	0.780	0.290
PepBind	0.344	—	0.469	0.793	0.372
Visual	0.670	0.680	—	0.730	0.170
PepNN-Seq	—	—	—	0.805	0.278
PepNN-Struct*	—	—	—	0.841	0.321
PepBCL (ours)	0.315	0.984	0.540	0.815	0.385

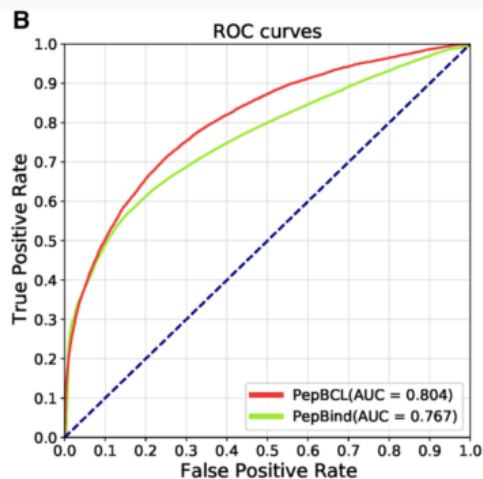
Note: The metrics of other methods were obtained from the corresponding publications and the methods with * are structure-based. The bold font indicates the best performance for each metric.



TE639 test set

Methods	Recall	Specificity	Precision	AUC	MCC
PepBind	0.317	—	0.450	0.767	0.348
PepNN-Seq	—	—	—	0.792	0.251
PepNN-Struct*	—	—	—	0.838	0.301
PepBCL(ours)	0.252	0.983	0.470	0.804	0.312

Note: The metrics of other methods were obtained from the corresponding publications and the methods with * are structure-based. The bold font indicates the best performance for each metric.



PepBCL可以只从蛋白质原始序列提取到更多区别性的特征，也能证明不用蛋白质结构也能够准确识别多肽绑定残基

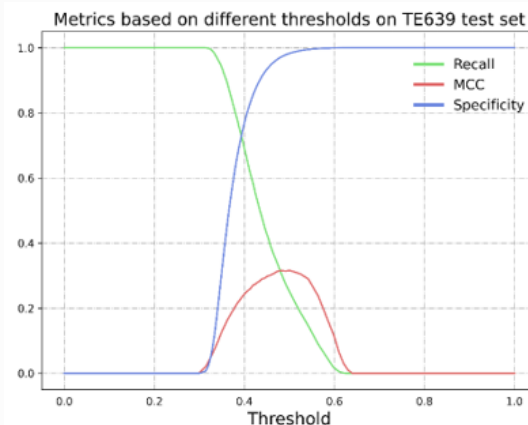
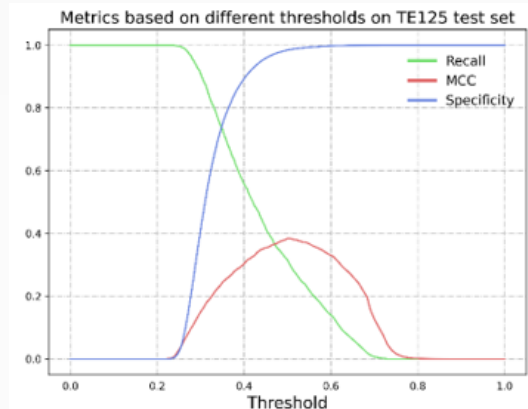
由于现实很多方法会错失蛋白质-多肽结合位点，因此使用不同的概率阈值重新训练模型

TE125 test set

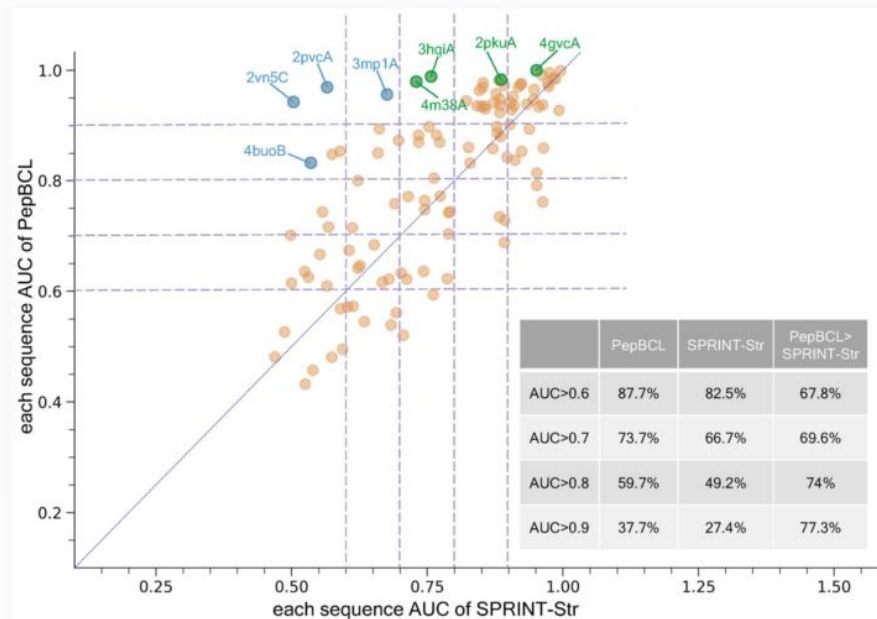
Methods	Recall	Specificity	AUC	MCC
PepBind	0.344	-	0.793	0.372
Visual	0.670	0.680	0.730	0.170
PepBCL(t=0.500)	0.315	0.984	0.815	0.385
PepBCL(t=0.478)	0.351	0.976	0.815	0.374
PepBCL(t=0.358)	0.700	0.780	0.815	0.256

TE639 test set

Methods	Recall	Specificity	AUC	MCC
PepBind	0.317	-	0.767	0.348
PepBCL(t=0.500)	0.252	0.983	0.804	0.312
PepBCL(t=0.470)	0.350	0.960	0.804	0.309
PepBCL(t=0.407)	0.650	0.805	0.804	0.253



由于该模型是在整个数据集上获得了好的性能，为了证明在每一条蛋白质上的预测性能，和基于结构的方法SPRINT-Str在TE125上进行对比



PepBCL的AUC比SPRINT_Str高
PepBCL在单条蛋白质上也有很好的预测性能

对比学习效果

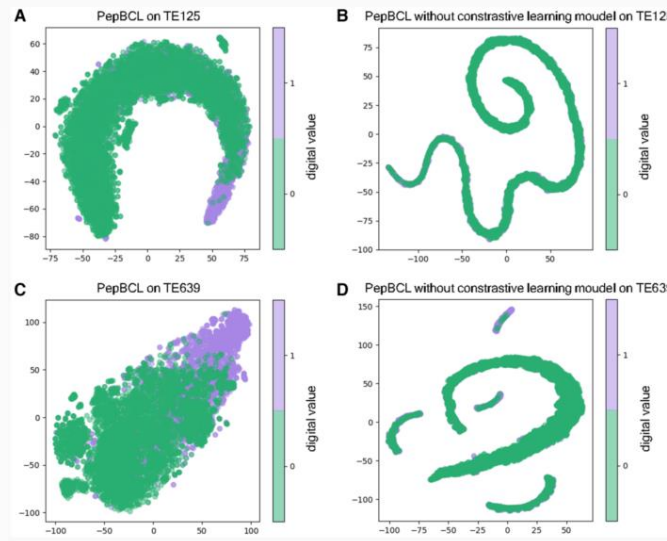
Datasets	Methods	Recall	Specificity	Precision	MCC
TE125	PepBCL	0.315	0.984	0.540	0.385
	PepBCL (no contrast module)	0.485	0.927	0.282	0.322
TE639	PepBCL	0.252	0.983	0.470	0.312
	PepBCL (no contrast module)	0.397	0.941	0.288	0.292

Datasets	Methods	AUC	Difference of AUC	P-value
TE125	PepBCL	0.815	0.009	0.040
	PepBCL (no contrast module)	0.806		
TE639	PepBCL	0.804	0.010	<1e-4
	PepBCL (no contrast module)	0.794		

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

从Spe, Pre看出, 对比学习可以帮助模型准确的识别出更多的非绑定残基, 减少了FP数量

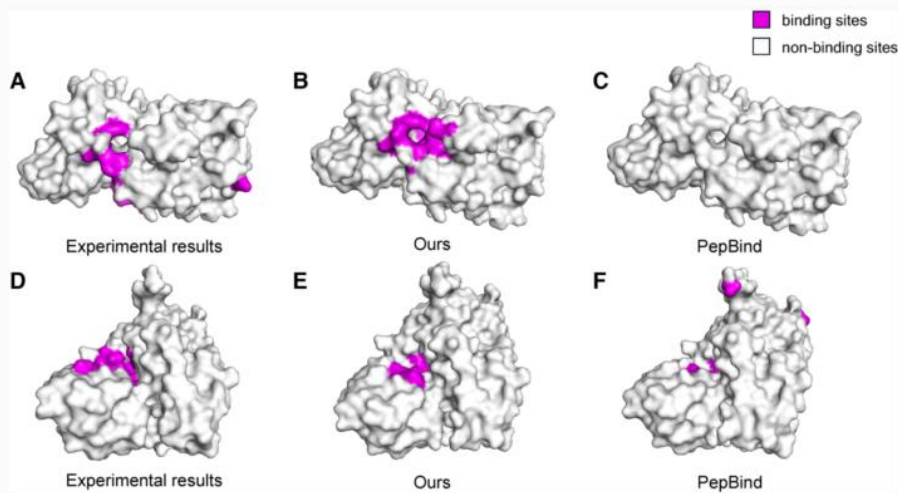


tSNE (T - 分布随机近邻嵌入) 图

主要用于数据展示, 即将不能可视化的多维数据降维到低维 (如2维), 进行数据可视化展示

相比较没有对比学习模块的残基样本的特征空间表示, 有对比学习模块的两种类别分布的更加清晰
证明了对比学习可以在不同类样本中捕捉更多有区别性的信息

从TE125中挑选两个蛋白质 4l3oA, 1fchA, 可视化该模型PepBCL和PepBind的预测结果

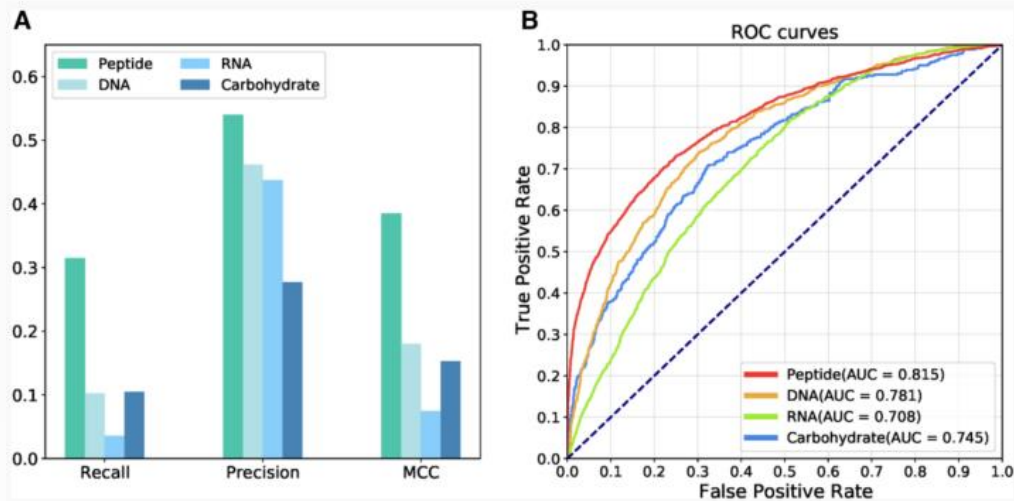


从图上可以看出，PepBCL预测的绑定残基和真实的绑定残基更加相似，预测的更加准确和完整

可能是因为用PepBCL模型获取的特征更好的保留了连续的序列特性，而蛋白质绑定区域通常都是连续的

与其他配体绑定残基的区别

选取了30个DNA绑定的蛋白质 (DNA30)，30个RNA绑定的蛋白质 (RNA30)，30个碳水化合物绑定的蛋白质 (CBH30)



PepBCL对于蛋白质-多肽绑定残基的识别具有特异性，可以更好的捕获多肽绑定残基的具有区别性的信息

从准确度可以看出，多肽、DNA、RNA有一定的相似，可以猜测在未来的研究中这些预测的DNA和RNA绑定残基可能是潜在的多肽绑定残基

与其他手工特征对比

三种手工特征：编码序列特征、进化特征和结构信息特征

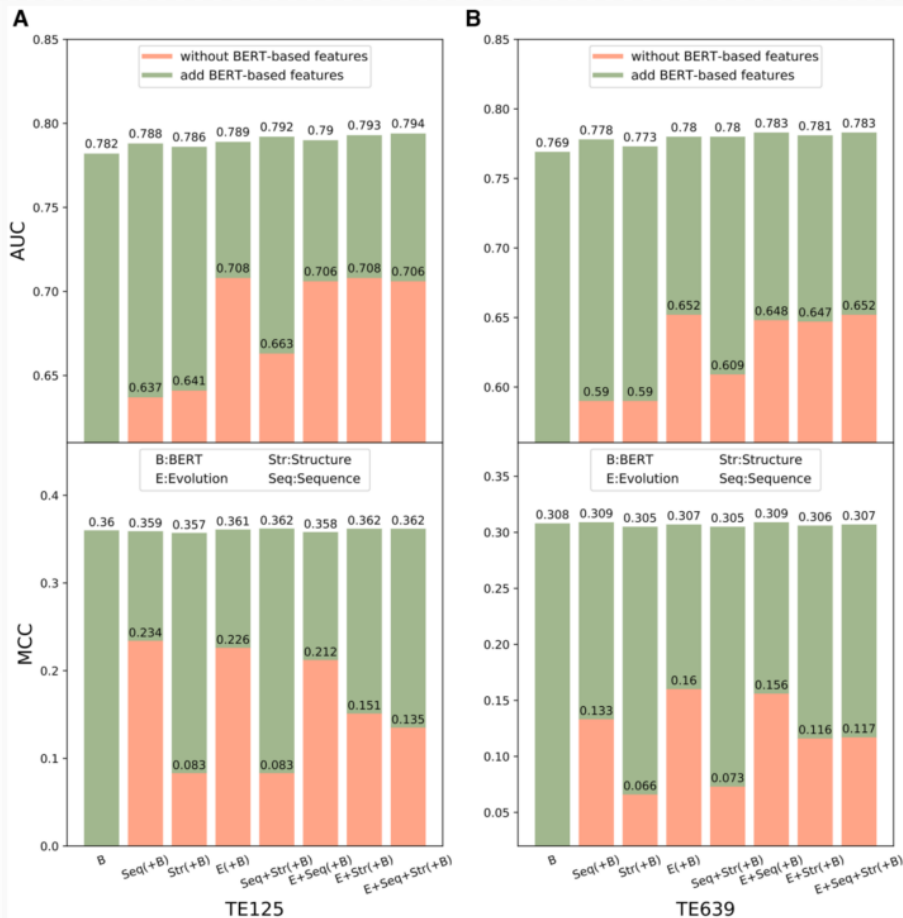
编码序列：onehot，每一个残基由20维的onehot向量表示

进化特征：从通过运行PSI-BLAST生成的PSSM获得

结构信息：二级结构等

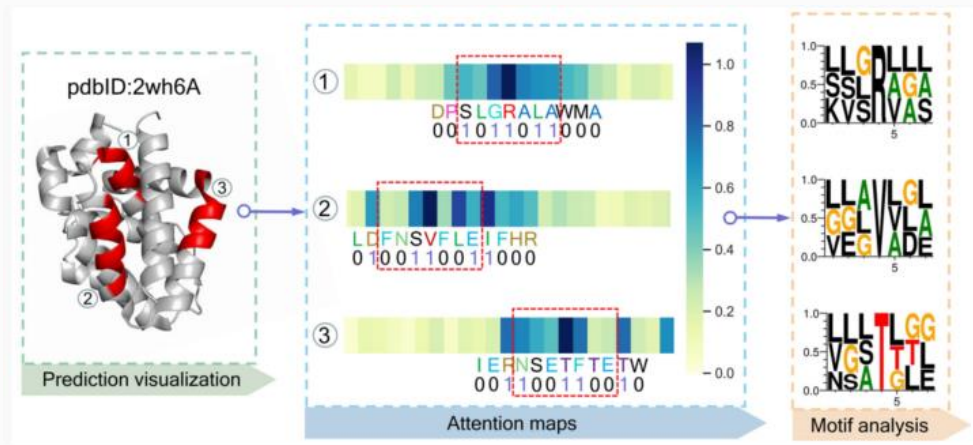
由于随机森林预测性能比其他机器学习好，用随机森林作为分类器进行训练和预测，数据集为Dataset1和Dataset2

所设计的基于BERT的编码器模块能够提取和学习蛋白质序列的高潜在表征，与传统的手工特征相互补充，进一步提高了编码器的性能



模型的可解释性

可视化蛋白质序列绑定区域的注意力分数



Motif分析：通过分析数据集
绑定区域附近的氨基酸种类
分布频率

①中心R周围的所有6个氨基酸出现在Motif分析中
对应序列相同的位置

③中心T周围的6个氨基酸中有4个出现在Motif分析
中对应序列模式相同的位置

说明模型能够学习到绑定残基附件环境的潜在联系

②中心V周围的6个氨基酸中只有2个出现在Motif分析
中对应序列相同的位置

说明模型由于注意机制能够发现新的特定的不同于
通过简单的数据集分析得到的绑定序列模式

解决的问题:

- ✓ 与使用第三方工具预测的信息的现有方法不同，PepBCL 是一种完全基于序列的预测方法，仅使用蛋白质序列进行模型训练和预测，从而加速预测过程并提高计算效率。通过引入一个经过良好训练的蛋白质语言模型，可以自动提取和学习与蛋白质结构和功能相关的高潜在特征表示
- ✓ 一种新的基于对比学习的模块来解决数据不平衡问题
- ✓ 通过可视化蛋白质序列结合区域的注意力评分为我们的模型预测提供了可解释性