ChIP-seq Analysis and Spectra Predicting

崔皓玮 何啓成 阳尚志

ChIP-seq Analysis

摘要

MACS(Model-based analysis of ChIP-seq)，是一种能在ChIP-seq数据中识别出转录因子结合位置的算法，该算法最终给出结合位置富集峰值及统计可信度。

前言

ChIP-seq用于在全基因组范围内绘制转录因子结合位点和组蛋白修饰状态。ChIP通常在转录因子结合位点或组蛋白标记位置周围产生75-300bp的DNA片段，而高通量测序通常从ChIP DNA片段的5'端产生数千万至数亿个25-75 bp序列（称为short reads）。ChIP-seq数据分析是将short reads映射回参考基因组，short reads富集区表示转录因子结合或组蛋白标记的位点。MACS是一种从ChIP-seq数据中识别short reads富集区的计算方法，在ChIP-seq和下游分析中得到广泛应用。

数据集与方法:

step1，得到chip-seq产生的数据

如果已经是带有注释的reads文件格式，如bam、bed、sam等: 无须再做处理

如果是由测序直接产生的fastq文件，需要对照参考基因组，生成带有注释的reads文件格式

使用bowtie实现

1. 生成参考基因组索引(ebwt格式)

```
bowtie-build reference_genomeFileName index_prefixFileName
```

2. 转化

```
bowtie -x index_prefixFileName -U inputfastq -S output.sam
```

3. (可选)进一步将sam文件转成bam文件

```
samtools view -bS input.sam -o output.bam
```

使用bowtie2实现

1. 生成参考基因组索引(bt2格式)

```
bowtie2-build reference_genomeFileName index_prefixFileName
```

2. 转化

```
bowtie2 -x index_prefixFileName -U inputfastq -S output.sam
```

例如：从chip_dmel.fastq.gz和input_dmel.fastq.gz得到chip_dmel.bam和input_dmel.bam，并对读入的序列进行检查

```
##序列质量检查
for sample in chip_dmel input_dmel ; do
    fastx_quality_stats -i <(gunzip -c ${sample}.fastq.gz) -o
${sample}_stats.txt
    fastq_quality_boxplot_graph.sh -i ${sample}_stats.txt -o
${sample}_quality.png -t ${sample}
    fastx_nucleotide_distribution_graph.sh -i ${sample}_stats.txt -o
${sample}_nuc.png -t ${sample}
    rm ${sample}_stats.txt
done


##原始读入计数
for sample in chip_dmel input_dmel ; do
    echo -en $sample"\t"
    gunzip -c ${sample}.fastq.gz | awk '((NR-2)%4==0)
{read=$1;total++;count[read]++}END{for(read in count){if(!max||count[read]>max)
{max=count[read];maxRead=read};if(count[read]==1){unique++}};print
total,unique,unique*100/total,maxRead,count[maxRead],count[maxRead]*100/total}'
done
```

```
##读入长度检查
for sample in chip_dmel input_dmel ; do
    echo -en $sample"\t"
    gunzip -c ${sample}.fastq.gz | awk '((NR-2)%4==0)
{count[length($1)]++}END{for(len in count){print len}}'
    LEN=36
    gunzip -c ${sample}.fastq.gz | awk -vLEN=$LEN '{if((NR-2)%2==0){print
substr($1,1,LEN)}else{print $0}}' | gzip > ${sample}_36bp.fastq.gz
done



##对读入进行匹配
for sample in chip_dmel input_dmel; do
    gunzip -c ${sample}_36bp.fastq.gz | bowtie -q -m 1 -v 3 --sam --best --
strata dm5/d_melanogaster_fb5_22 - > ${sample}.sam
done

gunzip -c chip_dmel_36bp.fastq.gz | bowtie -q -m 1 -v 3 --sam --best --strata
dm5/d_melanogaster_fb5_22 - > chip_dmel.sam
samtools view -Sb chip_dmel.sam > chip_dmel_nonSorted.bam
samtools sort chip_dmel_nonSorted.bam -T chip_dmel -o chip_dmel.bam

for sample in chip_dmel input_dmel; do
    samtools view -Sb ${sample}.sam > ${sample}_nonSorted.bam
    samtools sort ${sample}_nonSorted.bam -T ${sample} -o ${sample}.bam
    samtools index ${sample}.bam
    rm ${sample}.sam ${sample}_nonSorted.bam
done
```

step2，使用macs3处理数据

```
macs3 callpeak -t TreatmentGroup.bam -c ControlGroup.bam -f FileType(Here is
BAM) -g GenomeType(here is hs==human) -n test(OutputFileName) -B -q 0.01(q-value
screen)
```

- 当测序深度较大或者基因组大小比较小的时候，由于算法原因，可能会报错
  - 错误信息

    > MACS3 needs at least 100 paired peaks at + and - strand to build the model, but can only find 0! Please make your MFOLD range broader and try again.

  - 解决办法——取消算法的model过程（如果取消此过程，则无法生成.r分析结果脚本）

    ```
    macs3 callpeak -t TreatmentGroup.bam -c ControlGroup.bam -f FileType -g
    GenomeType -n test -B -q 0.01 --nomodel --extsize 147
    ```

step3，输出：

macs3通常会生成peaks.xls、summits.bed、treat_pileup.bdg、peaks.narrowPeak、model.r文件

peaks.xls

chr	start	end	length	abs_summit	pileup	-log10(pvalue)	fold_enrichment	-log10(qvalue)	name
chr22	16058732	16058878	147	16058805	1	5.53245	1.99516	3.15162	test_peak_1
chr22	16162899	16163045	147	16162972	1	5.53245	1.99516	3.15162	test_peak_2
chr22	16189662	16189808	147	16189735	1	5.53245	1.99516	3.15162	test_peak_3
chr22	16399905	16400051	147	16399978	1	5.53245	1.99516	3.15162	test_peak_4
chr22	16405614	16405760	147	16405687	1	5.53245	1.99516	3.15162	test_peak_5
chr22	16494710	16494856	147	16494783	1	5.53245	1.99516	3.15162	test_peak_6
chr22	17024434	17024580	147	17024507	1	5.53245	1.99516	3.15162	test_peak_7
chr22	17025618	17025764	147	17025691	1	5.53245	1.99516	3.15162	test_peak_8
chr22	17255469	17255794	326	17255570	10	18.1355	9.89864	15.3046	test_peak_9
chr22	17372727	17373186	460	17373043	19	43.5026	18.946	39.9038	test_peak_10
chr22	17392311	17392644	334	17392495	36	74.8967	32.4823	70.3694	test_peak_11

1. 文件内容分为以下模块
    - **chr**: 表示染色体（chromosome）的名称或编号。
    - **start**: 峰的起始位置（start position），即峰的左边界位置。
    - **end**: 峰的结束位置（end position），即峰的右边界位置。
    - **length**: 峰的长度，即峰的范围（end - start + 1）。
    - **abs_summit**: 峰的峰顶位置（summit position），即峰的最高点位置。
    - **pileup**: 在峰区域内测量到的测序 reads 的堆积数量。
    - **-log10(pvalue)**: 对应于峰区域的 p-值的负对数（以底数为 10），表示峰的统计显著性水平。数值越大，表示 p-值越小，峰越显著。
    - **fold_enrichment**: 峰的富集倍数，表示峰区域中的 ChIP 信号与对照组（control）信号相比的倍数。较高的富集倍数表示该峰在 ChIP 实验中的显著信号。
    - **-log10(qvalue)**: 对应于峰区域的 q-值的负对数（以底数为 10），用于校正多重假设检验中的错误发现率（FDR）。数值越大，表示 q-值越小，峰越可靠。
    - **name**: 峰的名称或标识符。
2. 可以编写bash文件，使用文本处理工具（如awk、sed、grep等）或者脚本编程语言（如Python、R）来读取、解析和处理该文件。可以根据需要提取感兴趣的信息，进行过滤、排序、合并等操作。

summits.bed

chr22	16058804	16058805	test_peak_1	3.15162
chr22	16162971	16162972	test_peak_2	3.15162
chr22	16189734	16189735	test_peak_3	3.15162
chr22	16399977	16399978	test_peak_4	3.15162
chr22	16405686	16405687	test_peak_5	3.15162
chr22	16494782	16494783	test_peak_6	3.15162
chr22	17024506	17024507	test_peak_7	3.15162
chr22	17025690	17025691	test_peak_8	3.15162
chr22	17255569	17255570	test_peak_9	15.3046
chr22	17373042	17373043	test_peak_10	39.9038
chr22	17392494	17392495	test_peak_11	70.3694
chr22	17398503	17398504	test_peak_12	28.7906
chr22	17539156	17539157	test_peak_13	20.3718

summits.bed：这是MACS3生成的峰值顶部位置的BED格式文件，每行表示一个峰值的染色体、起始位置和终止位置。可以使用BED文件处理工具（如bedtools）或者脚本编程语言来处理该文件，如合并重叠的峰值、计算峰值长度、计算峰值的覆盖范围等。

```
bedtools merge -i input.bed > merged.bed
# 合并重叠区

bedtools sort -i input.bed > sorted.bed
# 排序

bedtools bedlength -i input.bed > lengths.txt
# 计算峰区长度
```

treat_pileup.bdg

```
chr22   0         16052569          0.00000
chr22   16052569          16052716          1.00000
chr22   16052716          16058731          0.00000
chr22   16058731          16058878          1.00000
chr22   16058878          16103660          0.00000
chr22   16103660          16103807          1.00000
chr22   16103807          16110195          0.00000
chr22   16110195          16110342          1.00000
chr22   16110342          16114618          0.00000
chr22   16114618          16114765          1.00000
chr22   16114765          16162898          0.00000
chr22   16162898          16163045          1.00000
```

这是MACS3生成的处理组的叠加文件，以BEDGraph格式存储了每个位置的叠加值。可以使用BEDGraph处理工具（如bedtools、bedGraphToBigWig）或者脚本编程语言来处理该文件，如生成可视化文件wig/bigwig计算峰值的平均叠加值、进行区域操作等。

```
bedGraphToBigWig input.bdg chrom.sizes output.bw
# 转为bigwig

bedtools map -a regions.bed -b input.bdg -c 4 -o mean > output.bdg
#根据BED文件对BDG文件进行区域操作

awk '$4 >= 0.5' input.bdg > filtered.bdg
#将BDG文件中的值按照某个阈值进行筛选
```

- 转成bigwig或wig后，可以通过IGV等进行可视化分析

peaks.narrowPeak

```
chr22   16058731        16058878        test_peak_1     31      .       1.99516 5.53245 3.15162 73
chr22   16162898        16163045        test_peak_2     31      .       1.99516 5.53245 3.15162 73
chr22   16189661        16189808        test_peak_3     31      .       1.99516 5.53245 3.15162 73
chr22   16399904        16400051        test_peak_4     31      .       1.99516 5.53245 3.15162 73
chr22   16405613        16405760        test_peak_5     31      .       1.99516 5.53245 3.15162 73
chr22   16494709        16494856        test_peak_6     31      .       1.99516 5.53245 3.15162 73
chr22   17024433        17024580        test_peak_7     31      .       1.99516 5.53245 3.15162 73
chr22   17025617        17025764        test_peak_8     31      .       1.99516 5.53245 3.15162 73
chr22   17255468        17255794        test_peak_9     153     .       9.89864 18.1355 15.3046 101
```

这是MACS3生成的峰值结果的压缩窄峰格式文件，通常用于基因组注释和后续分析。可以使用文本处理工具或者脚本编程语言来读取和解析该文件，提取感兴趣的注释信息，如靠近的基因、功能区域等。

model.r

```
# R script for Peak Model
#   -- generated by MACS
p ← c(0.021549946001992445,0.024628509716562795,0.028199643625464402,0.03238649027728008,0.0343567710546051,0.033864200860273
84,0.03250963282586289,0.03127820734003475,0.032879060471611335,0.03177077753436601,0.033002203020194146,0.033371630665942585,
0.033125345568776964,0.03238649027728008,0.03386420086027384,0.03472619870035354,0.035341911443267614,0.035341911443267614,0.0
3423362850602229,0.03645019438051294,0.03558819654043324,0.03484934124893636,0.03423362850602229,0.03411048595743947,0.0342336
2850602229,0.033248488117359774,0.03583448163759887,0.03620390928334731,0.03718904967200982 5,0.03583448163759887,0.03521876889
4684796,0.03632705183193012,0.03657333692909575,0.03620390928334731,0.03620390928334731,0.03669647947767857,0.0348493412489363
6,0.035095626346101985,0.03497248379751917,0.03632705183193012,0.03731219222059263 5,0.03731219222059263 5,0.039775043192248914,
0.039528758095083286,0.0396519006436661,0.039159330449334846,0.03928247299791766,0.03928247299791766,0.03891304535216922,0.038
54361770642078,0.03718904967200982 5,0.03706590712342701,0.03620390928334731,0.03706590712342701,0.037558477317758264,0.0373121
92220592635,0.036942764574844196,0.03805104751208952,0.03891304535216922,0.039159330449334846,0.039528758095083286,0.040513898
4837458,0.03903618790075203,0.03866676025500359,0.03891304535216922,0.03903618790075203,0.039528758095083286,0.038913045352169
22 0.03731219222059263 5 0.03645019438051294 0.03645019438051294 0.035341911443267614 0.03657333692909575 0.03657333692909575 0
```

这是MACS3生成的模型文件，通常用于后续的差异分析。该文件包含了建模过程中使用的参数和统计模型。可以使用R语言加载该文件，并根据需要进行模型的解析、参数的提取和进一步的统计分析。

```
Rscript FileName_model.r
```

结果：

在Github上有此次简单分析的代码，其中`output.fa`，`sep_index.bed`为位点信息`Otest.txt`为序列信息。可运行得到：

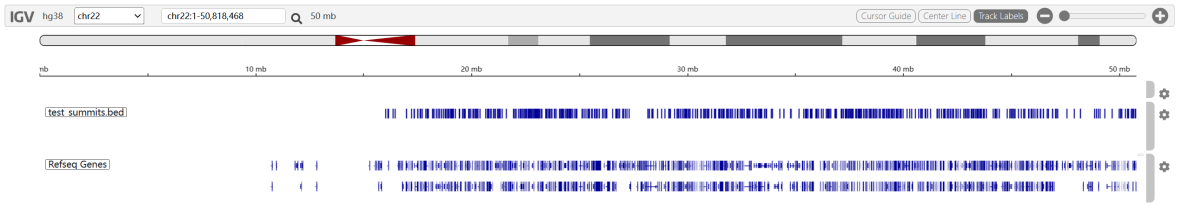
```
bedtools getfasta -fi ./BioData/hg19/chr22.fa -bed seq_index.bed -fo output.fa
```

ChIP_procce.sh脚本用法：

```
bash CHip_procces.sh -t data_file -c control_data -g data_type_of_bio -n
outputfile_name
```

此脚本会另外询问是否调用xls_processor.sh对結果进行處理。

最后得到一xls文件，其中包含基因组中可能的结合位。以下为峰信息在IGV上的可视化结果



讨论

chip-seq流程生成的序列很有可能是蛋白质结合序列，有待进一步实验证明。结合的蛋白质可能是转录因子或组蛋白，需要进一步的筛选。

贡献

阳尚志负责macs3数据分析流程的构建，崔皓玮负责生成可供macs3使用的.bam文件，何啓成负责xls_processor.sh的编写。报告由3人共同完成。

参考文献

Feng, J., Liu, T., Qin, B. et al. Identifying ChIP-seq enrichment using MACS. Nat Protoc 7, 1728–1740 (2012). https://doi.org/10.1038/nprot.2012.101

使用pDeep2预测质谱的准确度检验

摘要

pDeep2使用深度学习的方法预测蛋白质谱。本文将HLA_A0101免疫多肽的真实谱图和pDeep2预测谱图视为两个单位向量，用它们的点积来表示质谱的相似程度，检验了pDeep2预测质谱的准确度，证明其可以用于蛋白质组学研究中。

前言

蛋白质翻译后修饰（PTM）的调节对生物体起到非常重要的作用，基于串联质谱（MS/MS）自下而上的蛋白质组学是目前分析样品中PTM的主要方法。修饰肽的鉴定和修饰位点的定位依赖于理论质谱和实验质谱的比较，因此有必要开发出准确预测修饰肽MS/MS质谱的方法。pDeep2是使用迁移学习来训练的模型，我们需要检验它预测质谱的准确度，之后才能将其应用到科研中。

数据集与方法

数据：HLA免疫多肽质谱数据文件`M20151015_HLA_A0101_1e8ceq_biorep1_techrep1.mzML`，以及从中提取出来的离子肽段文件`HLA_A0101_ionslist.csv`

方法：

step1: 将.csv文件转换成pDeep2软件能够处理的格式

```
#转换.csv文件的格式
file_name="HLA_A0101_data/HLA_A0101_ionslist.csv"
echo -e "peptide\tmodification\tcharge" > "HLA_A0101_data/HLA_A0101_peptide.txt"
while read line
do
i=0
j=0
while true
do

c="${line:$i:1}"
if [ "$c" = "/" ]
then
        break
elif [ "$c" = "[" ]
then
        ((j++))
elif [ "$c" = "]" ]
then
        ((j--))
elif [ $j -eq 0 ]
then
        echo -e "$c\c" >> "HLA_A0101_data/HLA_A0101_peptide.txt"
fi

((i++))
done

((i++))
echo -e "\t\t${line:$i:$((${#line}-$i))}" >>
"HLA_A0101_data/HLA_A0101_peptide.txt"

done < $file_name
```

step2:用pDeep2软件预测给定肽段的质谱，可以调整相对碰撞能量的大小以提高预测准确度，一般在0.3左右都可以

```
#使用pDeep2进行预测，-e 0.27 设定相对碰撞能量为0.27（可以根据情况调整）
python pDeep-master/pDeep2/predict.py -e 0.27 -i QE -in
HLA_A0101_data/HLA_A0101_peptide.txt -out HLA_A0101_data/HLA_A0101_predict.txt
```

step3:调整文件格式，得到.mgf文件

```
#调整文件格式，得到.mgf文件
file_name="HLA_A0101_data/HLA_A0101_predict.txt"
echo -e "\c" > HLA_A0101_data/HLA_A0101_predict.mgf
while read line
do
        if [ "${line:0:6}" = "TITLE=" ] || [ "${line:0:8}" = "pepinfo=" ]
        then
                x=`expr index "$line" "||"`
                line="${line:0:$((x-1))}/${line:$((x+1)):$((${#line}-x-1))}"
        fi

        if [ "${line:0:6}" = "TITLE=" ]
        then
                line="NAME=${line:6:$((${#line}-6))}"
        fi
        echo "$line" >> HLA_A0101_data/HLA_A0101_predict.mgf
done < $file_name
```

step4:将.mgf转换成.msp

```
#调用R语言程序，将.mgf转换成.msp
Rscript Mgf2msp.r HLA_A0101_data/HLA_A0101_predict.mgf $PWD

#修改格式，使文件能够被spectrast处理
file_name="HLA_A0101_data/HLA_A0101_predict.msp"
echo -e "\c" > HLA_A0101_data/HLA_A0101_Predict.msp
while read line
do
        if [ "${line:0:10}" = "Num Peaks:" ]
        then
                line="Num peaks:${line:10:$((${#line}-10))}"
        fi
        echo $line >> HLA_A0101_data/HLA_A0101_Predict.msp
done < $file_name
rm $file_name
```

Mgf2msp.r的内容：

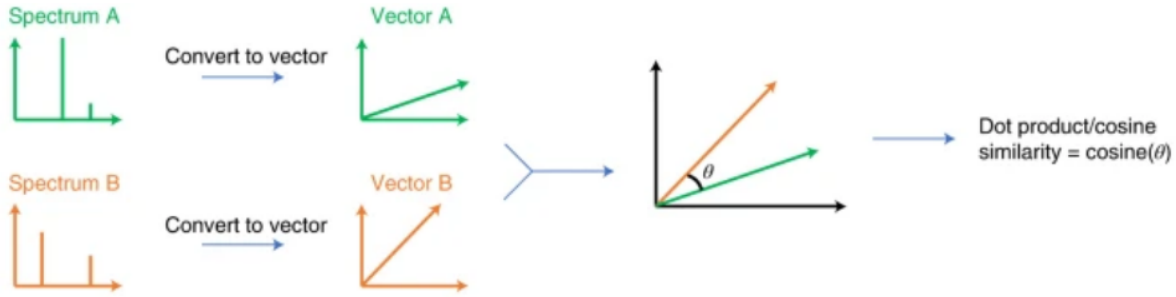
```
args<-commandArgs(trailingOnly = TRUE)
library(IDSL.FSA)
mgf2msp(path=args[2],args[1])
```

step5: 使用spectrast进行搜索与建库，得到`M20151015_HLA_A0101_1e8ceq_biorep1_techrep1.txt`

```
#使用spectrast进行搜索与建库
spectrast -cNHLA_A0101_data/raw HLA_A0101_data/HLA_A0101_Predict.msp
spectrast -sLHLA_A0101_data/raw.splib
HLA_A0101_data/M20151015_HLA_A0101_1e8ceq_biorep1_techrep1.mzML
spectrast -sR -sEtxt -sLHLA_A0101_data/raw.splib
HLA_A0101_data/M20151015_HLA_A0101_1e8ceq_biorep1_techrep1.mzML
```

step6:

将两个质谱转换为两个单位向量，用两向量角度的余弦（即这两个单位向量的点积）来表示质谱的相似程度




```
#每条肽段有一个Dot值，画出Dot值的分布直方图
python draw1.py
```

draw1.py的内容：

```
#画出Dot值分布直方图
import matplotlib.pyplot as plt
import numpy as np


x=[]
for i in range(101):
    x.append(i/100.0)

y=[0]*101
HEAD=1
for line in
open("HLA_A0101_data/M20151015_HLA_A0101_1e8ceq_biorep1_techrep1.txt",'r'):
    if HEAD==1:
        HEAD=0
        continue
    t=line.split()
    y[int(float(t[3])*100)]+=1

plt.bar(x,y,0.01)

plt.savefig('HLA_A0101_data/HLA_A0101_result.jpg')
plt.close()
```

step7:

```
#对于一个肽段，以离子质量为横坐标，离子相对强度为纵坐标，对于质谱实验得到的真实谱库和pDeep2预测的谱库分别画出离子分布图，比较Dot值较高的肽段的离子峰是否能够对应上，以此判断pDeep2预测的准确度
echo -e "\c" > "HLA_A0101_data/HLA_A0101_Dot.txt"
id=0
HEAD=1
while read line
do
        if [ $HEAD -eq 1 ]
        then
                HEAD=0
                continue
        fi

        i=`expr index "$line" /`
        ((i+=2))
        while [ "${line:$i:1}" = " " ]
        do
                ((i++))
        done

        j=$i
        while [ "${line:$j:1}" != " " ]
        do
                ((j++))
        done

        Dot=${line:$i:$((j-i))}
        #忽略Dot值较小的结果（认为是噪声）
        if [ `echo "$Dot>=0.82" | bc` -eq 1 ]
        then
                ((id++))

                i=0
                while [ "${line:$i:1}" != " " ]
                do
                        ((i++))
                done
                echo -e "$id ${line:0:$i} \c" >>
"HLA_A0101_data/HLA_A0101_Dot.txt"
                sss="${line:0:$i}"

                j=`expr index "$line" /`
                i=$j
                while [ "${line:$i:1}" != " " ]
                do
                        ((i--))
                done
                ((i++))

                while [ "${line:$j:1}" != " " ]
                do
                        ((j++))
                done

                echo "${line:$i:$((j-i))}" >> "HLA_A0101_data/HLA_A0101_Dot.txt"

                #画出分布图
```

```
                python draw2.py
HLA_A0101_data/M20151015_HLA_A0101_1e8ceq_biorep1_techrep1.mgf $sss
HLA_A0101_data/raw.sptxt "${line:$i:$((j-i))}"
"HLA_A0101_data/ion_pictures/${id}_${Dot}.jpg" $Dot
        fi

done < "HLA_A0101_data/M20151015_HLA_A0101_1e8ceq_biorep1_techrep1.txt"
```

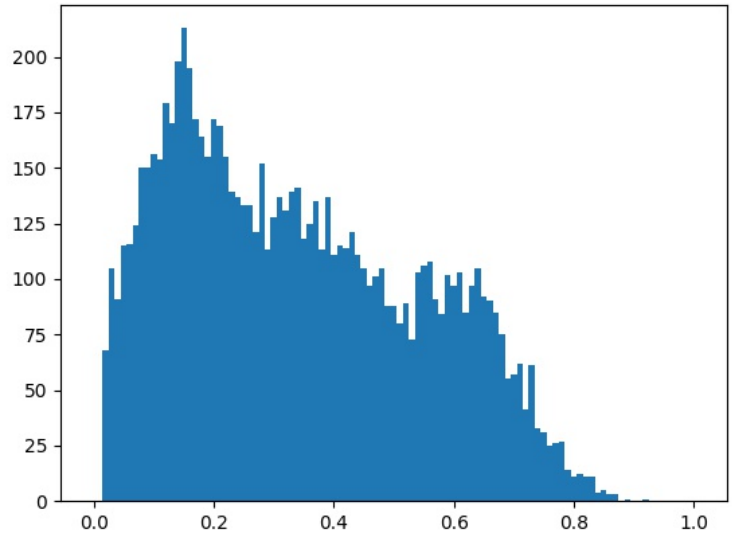
draw2.py的内容：

```
#每条肽段有一个Dot值，画出Dot值的分布直方图
import sys
import re
import matplotlib.pyplot as plt

f=open(sys.argv[1],"r")


str=f.read()
i=re.search("TITLE="+sys.argv[2],str).end()
i=i+re.search("CHARGE=",str[i:]).end()
i=i+re.search("\n",str[i:]).end()

Max1=0
Max2=0
x1=[]
y1=[]
x2=[]
y2=[]
while str[i:i+8]!="END IONS":
    while (str[i].isdigit())==False:
        i+=1
    j=i+re.search(" ",str[i:]).start()
    x1.append(float(str[i:j]))
    while str[j]==' ':
        j+=1
    y1.append(float(str[j:j+re.search("\n",str[j:]).start()]))
    Max1=max(Max1,y1[-1])
    i=j+re.search("\n",str[j:]).end()

f.close()

f=open(sys.argv[3],"r")
str=f.read()
i=re.search("Name: "+sys.argv[4],str).end()
i=i+re.search("NumPeaks:",str[i:]).end()
while str[i]==' ':
    i+=1

num=0
while str[i].isdigit():
    num=num*10+int(str[i])
    i+=1
for k in range(num):
    while (str[i].isdigit())==False:
        i+=1
```

```
    j=i+re.search("\t",str[i:]).start()
    x2.append(float(str[i:j]))
    while str[j]=='\t':
        j+=1
    y2.append(-float(str[j:j+re.search("\t",str[j:]).start()]))
    Max2=min(Max2,y2[-1])
    i=j+re.search("\n",str[j:]).end()

#离子强度的归一化：将最高峰都设为10000
for i in range(len(y1)):
    y1[i]=y1[i]/Max1*10000
for i in range(len(y2)):
    y2[i]=y2[i]/Max2*(-10000)

plt.vlines(x1,0,y1)
plt.vlines(x2,0,y2)
for i in range(len(x1)):
    j=0
    while True:
        if x1[i]-0.1<x2[j] and x2[j]<x1[i]+0.1:
            plt.vlines(x1[i],0,y1[i],colors='r') #匹配上的两条直线标为红色，未匹配上的
直线标为蓝色
            break
        j+=1
        if j>=len(x2):
            break

for i in range(len(x2)):
    j=0
    while True:
        if x2[i]-0.1<x1[j] and x1[j]<x2[i]+0.1:
            plt.vlines(x2[i],0,y2[i],colors='r')
            break
        j+=1
        if j>=len(x1):
            break

plt.title('Dot='+sys.argv[6])
plt.savefig(sys.argv[5])
plt.close()

f.close()
```

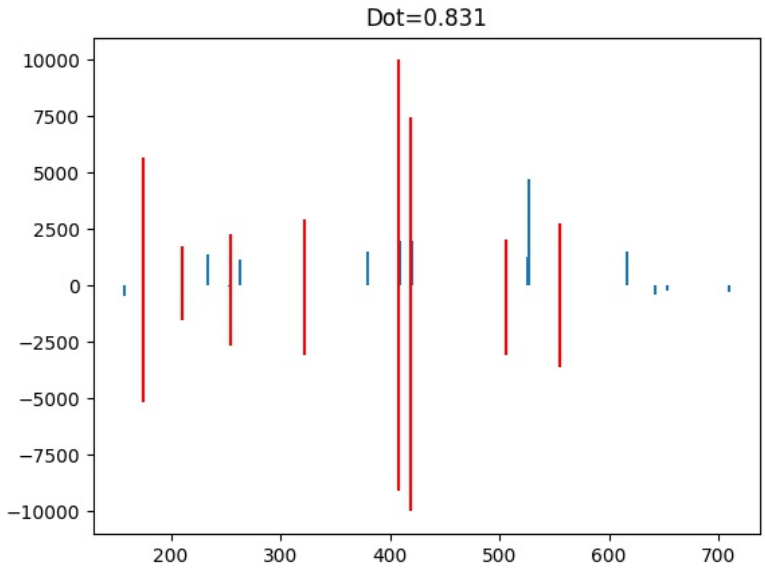
结果

`HLA_A0101_data/HLA_A0101_result.jpg` 如下所示，可以看到大致显现出2个峰（左边Dot值接近0的是噪声），与pDeep2开发者给出的期望情况相符。

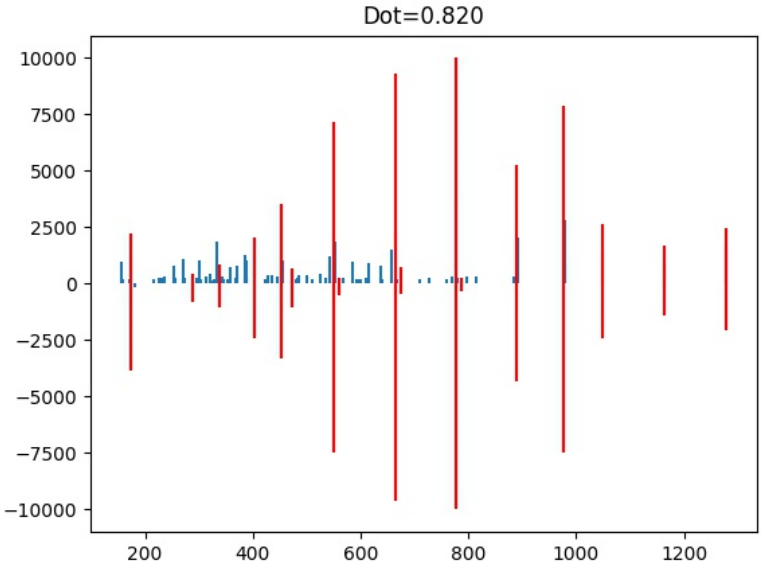


在`HLA_A0101_data/ion_pictures/`文件夹中产生了一些图片，对比了质谱实验得到的真实谱库和pDeep2预测的谱库的离子分布图，红线表示对应的离子峰。可以看到，除去一些噪声，pDeep2预测的结果与实验结果符合程度较高。

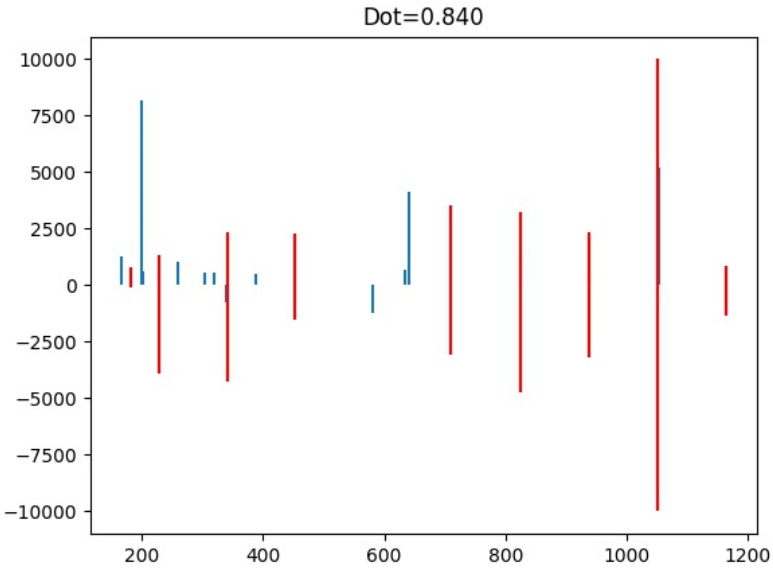
1_0.831.jpg:



27_0.820.jpg:



38_0.840.jpg:



讨论

pDeep2能够较为准确地预测出HLA_A0101质谱的结果，因此可以在蛋白质组学的研究中用于鉴定蛋白质的翻译后修饰。

贡献

崔皓玮负责数据分析流程的构建，何啓成、阳尚志负责流程的调试检验。

参考文献

- Wen-Feng Zeng, Xie-Xuan Zhou, Wen-Jing Zhou, Hao Chi, Jian-feng Zhan, Si-Min He. Anal. Chem. 2019, 91, 15, 9724-9731.
  MS/MS Spectrum Prediction for Modified Peptides Using pDeep2 Trained by Transfer Learning.https://doi.org/10.1021/acs.analchem.9b01262
- Li, Y., Kind, T., Folz, J. *et al.* Spectral entropy outperforms MS/MS dot product similarity for small-molecule compound identification. *Nat Methods* **18**, 1524–1531 (2021). https://doi.org/10.1038/s41592-021-01331-z

附录

项目核心代码等文件在Github上：https://github.com/misaka19260817/ChIP-seq_Analysis-and-Spectra_Predicting