实验九 制作人类基因剪接位点的GT-AG序列标识

一、实验目的

对于真核基因来说，几乎所有内含子DNA序列5'端起始的两个核苷酸总是5'-GT-3'，而3'端的最后两个核苷酸始终是5'-AG-3'。由于这两个碱基序列的高度保守性和广泛存在性，将其称为GT-AG法则，即5'-GT---AG-3'。利用序列标识（sequence logo），可以把GT-AG法则通过图形表示出来。而WebLogo便是一个灵活方便的制作序列标识的工具。针对人类基因组中22号染色体（chr22）上的所有基因，利用集成在Galaxy中的WebLogo和网络版的WebLogo，制作剪接位点的GT-AG序列标识。

1. 学习和掌握剪接位点的GT-AG法则。
2. 学习和掌握Galaxy的基本使用方法。
3. 学习和掌握WebLogo的使用方法。
4. 学习和掌握序列标识的含义。

二、实验内容——图形化操作

1. 打开Galaxy网站。通过搜索引擎搜索“Galaxy UCSC”，或者直接在浏览器的地址栏中输入网址，打开 https://main.g2.bx.psu.edu/ 即可。
2. 获取chr22上基因的内含子数据。打开Get Data工具集中的UCSC Main工具，调整参数提取人类hg19基因组中22号染色体（chr22）上的所有内含子信息，以BED格式进行存储。
3. 提取内含子上的剪接位点信息。
4. 提取供体位点的信息。打开Operate on Genomic Intervals工具集中的Get flanks工具，利用刚刚获取的内含子数据，调整Region为Around Start，调整Location of the flanking region/s为Upstream，设置Offset为17，设置Length of flanking region(s)为32，提取出包含供体位点2bp在内、同时上下游各延伸15bp的坐标。
5. 提取受体位点的信息。打开Operate on Genomic Intervals工具集中的Get flanks工具，利用刚刚获取的内含子数据，调整Region为Around End，调整Location of the flanking region/s为Downstream，设置Offset为-17，设置Length of flanking region(s)为32，提取出包含受体位点2bp在内、同时上下游各延伸15bp的坐标。
6. 获取剪接位点附近的序列。
7. 提取供体位点的序列。打开Fetch Sequences工具集中的Extract Genomic DNA工具，利用供体位点的坐标信息，提取出以FASTA格式保存的供体位点的序列。
8. 提取受体位点的序列。打开Fetch Sequences工具集中的Extract Genomic DNA工具，利用受体位点的坐标信息，提取出以FASTA格式保存的受体位点的序列。
9. 对序列进行多序列比对。此处提取的序列已经是根据坐标比对好的序列，没有必要再单独进行多序列比对了。
10. 制作序列标识。
11. 制作供体位点的序列标识。打开Motif Tools中的Sequence Logo工具，利用供体位点的序列，调整参数，制作供体位点的序列标识。
12. 制作受体位点的序列标识。打开Motif Tools中的Sequence Logo工具，利用受体位点的序列，调整参数，制作受体位点的序列标识。
13. WebLogo的使用。
14. 下载剪接位点的序列。从Galaxy中下载提取出的供体位点和受体位点的序列，保存到本地计算机上。
15. 打开WebLogo3网站。通过搜索引擎搜索“WebLogo”，或者直接在浏览器的地址栏中输入地址，打开 http://weblogo.threeplusone.com 即可。

16. 制作剪接位点的序列标识。在creat页面，上传供体位点或受体位点的序列，适当修改参数，就可制作出精美的序列标识。
17. 尝试对其他染色体或全基因组上的基因的剪接位点制作序列标识，进一步熟悉在Galaxy和WebLogo的使用方法。

三、实验内容——命令行操作

1. 配置环境。安装 conda 、 bioconda，新建环境(略)。
2. 安装软件。

```
# bedtools: a powerful toolset for genome arithmetic
conda install bedtools
# WebLogo: A Sequence Logo Generator
conda install weblogo
```

3. 下载数据。

```
# hg38, refGene, 基因数据
mysql -h genome-mysql.cse.ucsc.edu -u genome -D hg38 -N -A -e 'select
chrom,txStart,txEnd,name2,score,strand from refGene' > genes_hg38.bed
# hg38, refGene, 外显子数据
mysql -h genome-mysql.cse.ucsc.edu -u genome -D hg38 -N -A -e 'select
chrom,exonStarts,exonEnds,name2,score,strand from refGene' > genes_hg38_tmp.txt
awk 'BEGIN {OFS="\t"}; { n=split($2, a, ","); split($3, b, ","); for(i=1; i<n; ++i) print $1,
a[i], b[i], $4, $5, $6 }' genes_hg38_tmp.txt | sort | uniq > exons_hg38.bed
# hg38, 基因组信息
mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -e "select chrom, size from
hg38.chromInfo"  > hg38.genome
# hg38, 基因组序列
# wget --timestamping 'http://hgdownload.cse.ucsc.edu/goldenPath/hg38/bigZips/hg38.fa.gz' -O
hg38.fa.gz
# gunzip -c hg38.fa.gz > hg38.fa
```

4. 数据处理。

```
# 获取最终的外显子数据
bedtools sort -i exons_hg38.bed | bedtools merge -s -i stdin > exons.bed
# 提取内含子坐标（注意：此处不使用-s，为什么？）
bedtools sort -i genes_hg38.bed | bedtools subtract -a stdin -b exons.bed > introns.bed
# 提取供体位点和受体位点的坐标
bedtools flank -i introns.bed -g hg38.genome -l 15 -r 0 -s | awk 'BEGIN {OFS="\t"}; { if($2<$3)
print; }' | bedtools slop -i stdin -g hg38.genome -l 0 -r 17 -s > donor.bed
bedtools flank -i introns.bed -g hg38.genome -l 0 -r 15 -s | awk 'BEGIN {OFS="\t"}; { if($2<$3)
print; }' | bedtools slop -i stdin -g hg38.genome -l 17 -r 0 -s > acceptor.bed
# 提取供体位点和受体位点的序列（（此处如果不使用-s，结果会有什么变化？））
bedtools getfasta -fi hg38.fa -bed donor.bed -s > donor.fa
bedtools getfasta -fi hg38.fa -bed acceptor.bed -s > acceptor.fa
# 制作供体位点和受体位点的序列标识
weblogo -i -15 -F png -s large --resolution 300 < donor.fa > donor.png
weblogo -i -17 -F png  -s large --resolution 300 < acceptor.fa > acceptor.png
```

5. 参考资料
    - Programmatic access to the Genome Browser

- bedtools: a powerful toolset for genome arithmetic
- bedtools at GitHub
- Defining genomic regions
- WebLogo 3: User's Manual