利用 Python 进行生信分析 | 基础 + 实战

——Python for Bioinformatics

赵华男

2022 一起加油

项目一　序列文件的处理

序列储存格式的介绍

序列储存格式的介绍–FASTA 文件格式

- **FASTA 格式**
  - 一种用于表示核苷酸序列或多肽序列的文本格式；其中碱基对或氨基酸用单个字母来表示
  - 允许在序列前添加序列名及注释
  - 该格式已成为生物信息学领域的一项标准
- **FASTA 文件各行记录信息如下:**
  - 第一行
    - 由大于号">" 开头的任意文字说明，用于序列标记
    - 为了保证后续分析软件能够区分每条序列，单个序列的标识必须是唯一的
  - 第二行
    - 序列本身；只允许使用既定的核苷酸或氨基酸编码符号。
    - 通常核苷酸符号大小写均可，而氨基酸常用大写字母。
    - 注意有些程序对大小写有明确要求。一般每行 60-80 个字母。

```
1  >HWI-D00433:463:HNT7JBCXX:1:1101:19071:2193 1:N:0:TTCTCCAT
2  CCTACGGGACGCATCAGTGAGGAATATTGGTCAATGGACGCGAGTCTGAACCAGCCAAG
3  AGCGTGAAGGATGAAGGCCCGATGGGTTGTAAACCTCTTTTATCTGGGAATAAAACGTG
4  CACGTGTGGTATTTTGTATGTACCATAAGAATAAGTATCGGCTAACTCCGTGCCAGCAG
5  CGCGGTAATACGGAGGATCCGAGCGTTATCCGGATTTATTGGGTTTAAAGGGTGCGCAG
6  CGGTCTGTTA
7  >HWI-D00433:463:HNT7JBCXX:1:1101:1713:2316 1:N:0:TTCTCCAT
8  CCTACGGGGTTCACCAGTAGGGAATCTTCCACAATGGGCGAAAGCCTGATTGAGCAAAG
9  CGCGTGTTTTAAGAAGGTCTTCGGATCGTAAAACCCTGTTGTTAGAGAAGAAAGTGCGT
10 CGCGTAACTGTTCACGTTTCTACTGTATCTAACAAGAAAGCACCGGCTAACTACGTTCG
```

https://zhuanlan.zhihu.com/p/363971040

序列储存格式的介绍–FASTA 文件格式

```
# legend server
cd /home/zhaohuanan/3.project/2022_Other_projects/2022-09-25_Prepared_data/FASTA
rsync -avzP rsync://hgdownload.cse.ucsc.edu/goldenPath/mm39/chromosomes/ .
# check一下md5
md5sum -c md5sum.txt

(base) PS C:\Users\vagrant\PythonForBioinformatics> rsync -avzP rsync://hgdownload.cse.
ucsc.edu/goldenPath/mm39/chromosomes/chrM.fa.gz .
receiving incremental file list
chrM.fa.gz
          5,385 100%    5.14MB/s    0:00:00 (xfr#1, to-chk=0/1)

sent 43 bytes  received 5,492 bytes  299.19 bytes/sec
total size is 5,385  speedup is 0.97

# 接下来我们需要将genome生成一下，未确定的基因组区域暂不考虑，只看确定的基因组
# 测试命令
echo `seq 1 1 19` X Y M | awk 'BEGIN {printf "cat "}{for(i=1; i<=NF;i++){printf "chr"$i".fa.gz "}}'
# 写入genome文件
echo `seq 1 1 19` X Y M | awk 'BEGIN {printf "cat "}{for(i=1; i<=NF;i++){printf "chr"$i".fa.gz "}}' | \
    sh > genome_ucsc_mm39.fa.gz
zcat genome_ucsc_mm39.fa.gz | grep -v N | less # linux/window
zcat < genome_ucsc_mm39.fa.gz | grep -v N | less # macos
```

序列储存格式的介绍–FASTQ 文件格式

- 第一行
  - 以 @ 开头
  - 后面是 reads 的 ID 以及其他信息
  - 例如下一页例中 HWUSI-EAS100R 代表 Illumina 设备名称,
  - 6 代表 flowcell 中的第六个 lane,73 代表第六个 lane 中的第 73 个 tile
  - 941:1973 代表该 read 在该 tile 中的 x:y 坐标信息;
  - #0, 若为多样本的混合作为输入样本, 则该标志代表样本的编号, 用来区分个样本中的 reads/
  - /1 代表 paired end 中的前一个 read。
- 第二行
  - read 的序列
  - 紧接着下面两行代表该 read 的质量
- 第三行
  - 以“+”开头, 跟随着该 read 的名称 (一般于 @ 后面的内容相同), 但有时可以省略, 但“+”一定不能省

https://baike.baidu.com/item/fastQ 格式 (rmspace)

序列储存格式的介绍–FASTQ 文件格式

- **第四行**
  - 代表 reads 的质量。这一行可以详细说一下!
  - Illumina 测序仪是按照荧光信号来判断所测序的碱基是哪一种的，例如红黄蓝绿分别对应 ATCG，那么一旦出现一个紫色的信号该怎么判断呢?
  - 因此对每个结果都有一个概率的问题。起初 sanger 中心用 Phred quality score 来衡量该 read 中每个碱基的质量，既-10lgP #(lg 意为 log10)
    - 其中 P 代表该碱基被测序错误的概率
    - 如果该碱基测序出错的概率为 0.001，则 Q 应该为 30，那么 30+33=63，那么 63 对应的 ASCii 码为 '?'，则在第四行中该碱基对应的质量代表值即为 '?'
    - ASCII https://baike.baidu.com/item/ASCII

```
1  @SEQ_ID
2  GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
3  +
4  !''*((((****))%%%++)(%%%%).1***-+*''))**55CCF
5
6  # 例如在NCBI看到的FASTQ格式如下
7  @HWUSI-EAS100R:6:73:941:1973#0/1
8  GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTT
9  +HWUSI-EAS100R:6:73:941:1973#0/1
10 !''*((((****))%%%++)(%%%%).1***-+*''))**55CCF
```

序列储存格式的介绍–FASTQ 文件格式

```
# from illumina
@E00591:528:HHVW3CCX2:2:1101:17360:2170 1:N:0:GATCAGCG:ACTA
TGGAGTGAGTACGGTGTGCGTTGAAGTCCTCGTTGTCTTGTTGGCAGGGGTCTGCACCCGGGAGCCCCCGTTCTATATCATCACTGAGATCATGACCTACGGGAACCTCCTCGACTACC
+
JJJJAFFFFJFJJJAFFJJJJFJFFJJFFFJF7A<FJ7JF-<F-7A7<JF7<FJJJ-F<JJJJFJJJJJJ<<AJFJAJJ<7AJFFFJ-<7AFFJJJA-<FF--7FJ-F7-7FA7F<FFJ
# from mgi
@Beta12AdemL1C001R00100000184/1
GTGTCCGTATCTTCACCCCACCACAAACTATTAGCTTTAGAAAGGAAAAGAAAAACCACAACAAAACAGTGTGTGTGTACCAAAGACTTATATGTGCATAAGCAAAAGCAAACAATAGC
+
ICIDIIIDEDIDBIEIIIIEIIEIEEEID@DDE4IDDDEID>EIIEEEEIEEEEEIIEIEEIECEEIE4DIDIDDC6DEIIE@E6EI5CBDEDICIIECEEIIAEEEIIDEEIE:CD:I
# from sra
@fig7_untreated_rep1.1 1 length=150
TTACAAGACTGNTGTATTAGTTTATACTACAAGGACAGGCCCATTNGNNTNTNTTNTTTTNAANTAGGGANATAGTTGGTATTAGGATTAGNATTGTTGTGAAGTATAGTANGGATGCT
+fig7_untreated_rep1.1 1 length=150
AFFFFFFFFFF#FFFFFFFFFFFFFFFFFFFFFFFFFFFF/F#F##F#F#FF#FFFF#FF#FFFFFF#FFFFFFFFFFFFFFFFFFFA#FFFFFFFFFFFFFFFFFF#FFFFFF
```

FASTQ 文件的操作

FASTQ 文件的操作

- **FASTQ 文件的操作**
  - 读取 FASTQ 文件并以 FASTA 格式输出
  - 解析 FASTQ 的质量值，计算 Q30 比例
  - 根据 FastQC 报告对 FASTQ 文件进行截取
  - 根据 FastQC 报告，过滤低质量的 lane,tile 数据

FASTA 文件的操作

FASTA 文件的操作

- **FASTA 文件的操作**
  - 读取 FASTA 文件，并将其中 U 替换成 T
  - 读取 FASTA 文件，并输出反向互补序列
  - 计算基因组序列的长度
  - 计算基因组各染色体的平均 GC 含量
  - 计算基因组中 N 的总长度（effective length）

Docstring-reST

```python
def func(arg1, arg2):
    """Summary line.

    Extended description of function.

    :param int arg1: Description of arg1.
    :param str arg2: Description of arg2.
    :raise: ValueError if arg1 is equal to arg2
    :return: Description of return value
    :rtype: bool

    :example:

    >>> a=1
    >>> b=2
    >>> func(a,b)
    True
    """

    if arg1 == arg2:
        raise ValueError('arg1 must not be equal to arg2')

    return True
```

Docstring-Google Style

```python
def func(arg1, arg2):
    """Summary line.

    Extended description of function.

    Args:
        arg1 (int): Description of arg1
        arg2 (str): Description of arg2

    Returns:
        bool: Description of return value

    Raises:
        AttributeError: The ``Raises`` section is a list of all
            exceptions that are relevant to the interface.
        ValueError: If `arg2` is equal to `arg1`.

    Examples:
        Examples should be written in doctest format, and should
            illustrate how to use the function.
        >>> a=1
        >>> b=2
        >>> func(a,b)
        True
    """
    if arg1 == arg2:
        raise ValueError(
            'arg1 must not be equal to arg2'
        )

    return True
```

Docstring-Numpy Style

```python
def func(arg1, arg2):
    """Summary line.

    Extended description of function.

    Parameters
    ----------
    arg1 : int
        Description of arg1
    arg2 : str
        Description of arg2

    Returns
    -------
    bool
        Description of return value

    Raises
    ------
    AttributeError
        The ``Raises`` section is a list of all exceptions
        that are relevant to the interface.
    """
    """
    ValueError
        If `arg2` is equal to `arg1`.

    See Also
    --------
    otherfunc: some other related function

    Examples
    --------
    These are written in doctest format, and should
    use the function.

    >>> a=1
    >>> b=2
    >>> func(a,b)
    True
    """

    if arg1 == arg2:
        raise ValueError(
            'arg1 must not be equal to arg2'
        )
    return True
```

Q & A!

Thank you!