

python大作业：Gene comparison

姜戈519111910306

生物信息与统计学系

一.课题灵感

这个项目的灵感来自于本学期的一门专业课计算编程语言，老师在讲动态规划算法时提出的可以用于序列比对算出最优得分，这就引出了一个问题：序列比对

序列比对：是为确定两个或多个序列之间的相似性以至于同源性，而将它们按照一定的规律排列。将两个或多个序列排列在一起，标明其相似之处。序列中可以插入间隔（通常用短横线“-”表示）。

这一方法常用于研究由共同祖先进化而来的序列，特别是如蛋白质序列或DNA序列等生物序列。在比对中，错配与突变相应，而空位与插入或缺失对应。序列比对还可用于语言进化或文本间相似性之类的研究。

基于这个问题，我采用了Needleman/Wunsch算法（一个专用于文本比对的算法），来解决序列比对的评分问题和最佳回溯结果，然后衍生出了一些小功能，比如随机序列生成和数据分析等，从使用方便性考虑我用tinker包设计了一个可视化的小程序，整个程序是在python3.8的anaconda3环境下进行的

二.主要功能及简要介绍

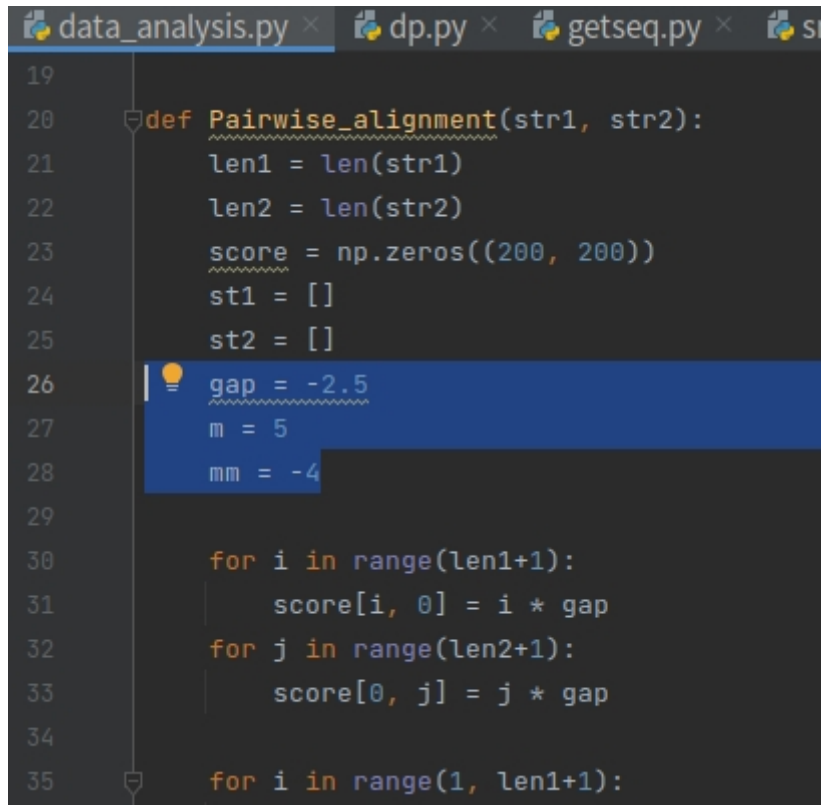
大致可以分成3个

1.序列比对与最佳回溯结果

思路

在生物信息处理中，我们希望找出两条序列之间具有的某种相似性关系，这种寻找生物序列相似性关系的算法就是双序列比对算法。通常利用两个序列之间的字符差异来测定序列之间的相似性，两条序列中相应位置的字符如果差异大，那么序列的相似性低，反之，序列的相似性就高。

- 生物碱基序列由ATCG组成，进行比对时需要设定一个评分的规则，我这里设定的是`match=5, mismatch=-4, gap=-2.5`，如果使用者需要可以自己修改规则，修改规则的位置在`data_analysis.py`文件的26-28行和`dp.py`文件的16-18行：



```
19
20 def Pairwise_alignment(str1, str2):
21     len1 = len(str1)
22     len2 = len(str2)
23     score = np.zeros((200, 200))
24     st1 = []
25     st2 = []
26     gap = -2.5
27     m = 5
28     mm = -4
29
30     for i in range(len1+1):
31         score[i, 0] = i * gap
32     for j in range(len2+1):
33         score[0, j] = j * gap
34
35     for i in range(1, len1+1):
```

```

data_analysis.py x dp.py x getseq.py x sn
2 import numpy as np
3
4 pairwise = "AAATTTCCGG"
5
6
7 def max(a, b):
8     if a >= b:
9         return a
10    else:
11        return b
12
13
14 def Pairwise_alignment(len1, len2, str1, str2):
15     Score = np.zeros((200, 200))
16     gap = -2.5
17     m = 5
18     mm = -4
19
20     for i in range(len1 + 1):
21         Score[i, 0] = i * gap
22     for j in range(len2 + 1):
23         Score[0, j] = j * gap

```

具体的算法实现思路是动态规划

评分函数

- `gap` 表示缺失得分为2.5，`m` 表示匹配得分为5，`mm` 表示非匹配得分设为-4；
- 初始化数组，对于第0层，第0列赋值为`i*gap`；
- 下面的双层 `for` 循环是对二维数组的每个位置算 `score` 值：
 - a. 总体每个位点的得分为：

三个方向的得分=该方向上一位点得分+移动过程得分

最后选取三个方向最高得分作为该位点的得分，以此循环从上到下，从左到右得到整个矩阵的得分。
 - b. 然后经过上一步就知道最右下角的得分肯定是最优得分，因为它是从每种子情况的最优得分得到的。
 - c. 具体的操作就是
 - i. 如果 `str1[i - 1] = str2[j - 1]`，那么 `score[i, j]` 直接等于 `score[i-1, j-1]+m`

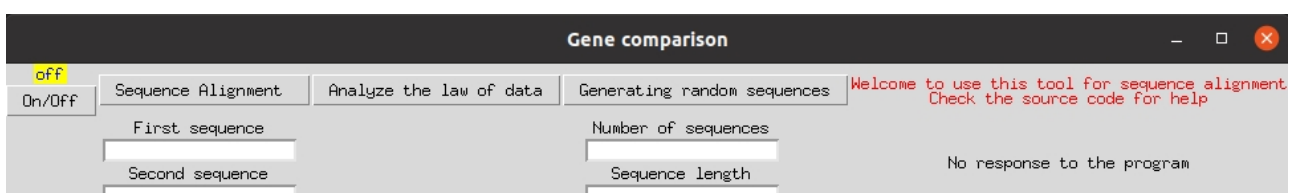
- ii. 如果 `str1[i - 1] != str2[j - 1]`, 那么 `score[i, j]` 直接等于 `score[i-1, j-1]+mm`
- iii. 先进行上面两步, 然后左上角 `score[i, j-1]` 与右上角 `score[i-1, j]` 比较取出较大值, 然后得出来的较大值加上 `gap`, 如果此时 `score[i, j]` 小于这个新的得到的分数, 那么更新 `score[i, j]`, 否则则不更新;
- iv. 最后的结果就是选取三个方向最高得分作为该位点的得分;
- 完整的分数表格形成后最右下角的得分必然就是最佳匹配得分, 写入 `SCORE` 文件即可; 同时返回给小程序
- 但是我们还需要知道它是从哪一条路径得到的这个最优得分。因此需要回溯, 这里就调用了 `printAlign` 函数

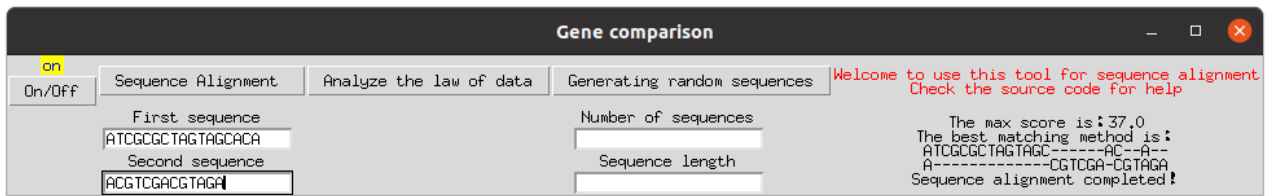
回溯函数

- 回溯的方式就是看每个回溯位点的左上方, 上方和左方最大值位置, 最后就可以得到整个回溯路径;
- `printAlign` 函数输入的参数比较多, `score` 是输入的二维评分数组, `i` 和 `j` 表示的是此时回溯到的评分数组的位置, 初始位置就是从 `(len1, len2)` 开始, `s1` 和 `s2` 是进行的回溯的两条序列, `saln` 和 `raln` 存储的是最后输出的最佳匹配结果
- 三个 `if` 判断是是判断左上, 左, 上哪个得分最高, 最高的就是最优的匹配路径:
 - a. 从右下方开始, 如果最大值出现在上面, 则横向这条序列引入一个GAP ("-"), 纵向这条序列取该处碱基;
 - b. 如果最大值出现在左边, 则纵向这条序列引入一个GAP ("-"), 横向这条序列取该处碱基;
 - c. 如果最大值出现在左上角, 则不引入GAP, 纵向和横向均取该处碱基。
 - d. 当然这个回溯的路径不一定唯一, 当三个位置有两个相同的时候, 两个路径都是可行的, 但为了小程序输出方便, 我用了 `if...elif` 这样就只会筛出一个序列输出

使用手册

运行 `smallprogram.py` 文件, 首先打开 `On`, 然后输入两条序列, 点击上方按钮即可输出最高得分和最佳匹配方式





由于这是一个优秀的文本字符串比对算法，因此不仅可以用来比对碱基序列，类似的文本比对，相似度比对，文字查重，蛋白质序列比对都可以同样进行

2.生成比例为人类染色体的随机序列

思路

人类1号染色体拥有2.3亿个碱基序列，人的很多模拟实验都要用到这些序列来研究特定排序的表达效果和功能，这时候不一定要从NCBI数据库下载真实的数据而只需要类似比例的序列进行模拟处理即可；

所以我就想到了这样一个小功能来帮助研究人员生成特定长度的特定数量的符合人类染色体一行碱基比例的DNA序列

getseqs函数

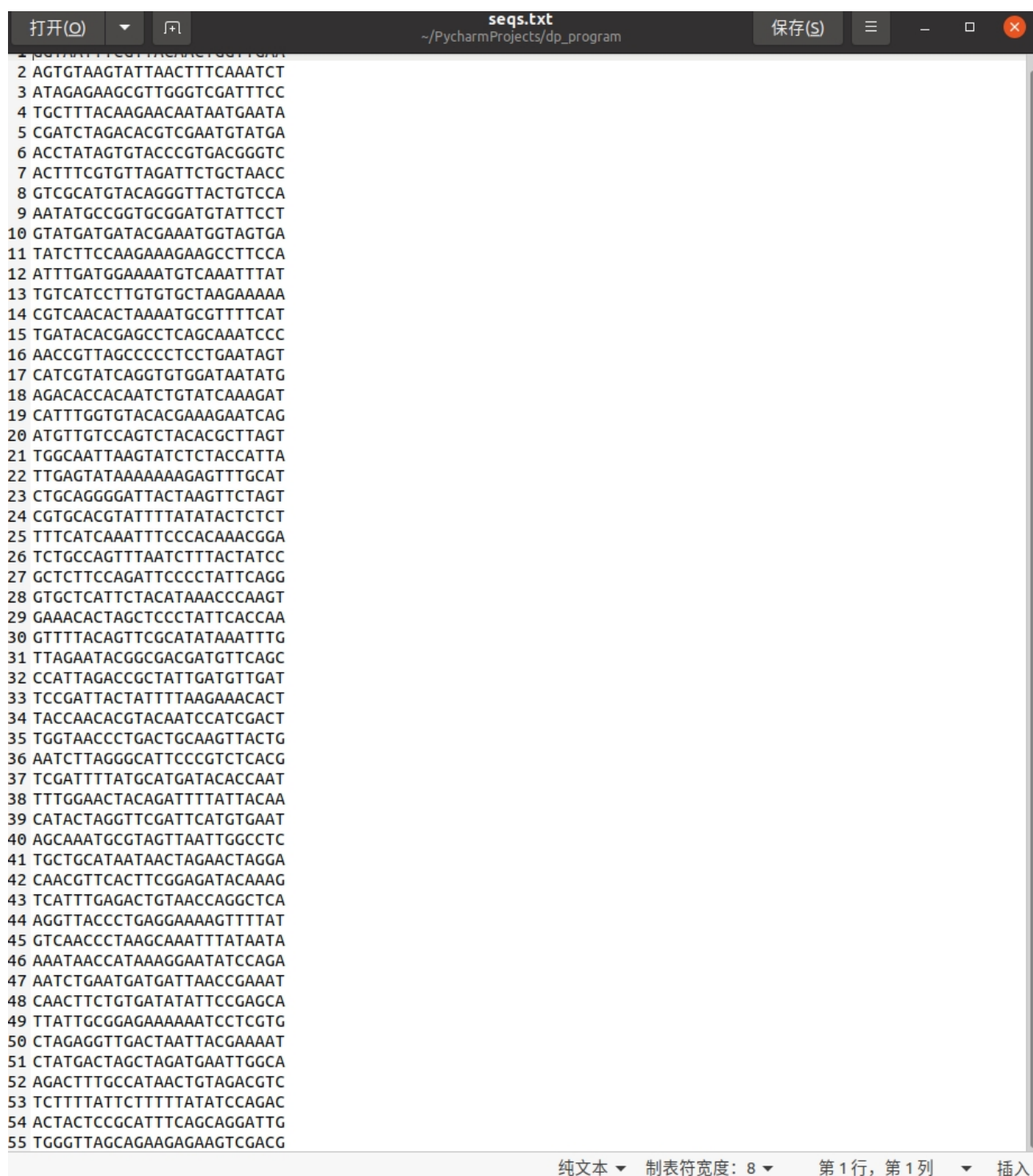
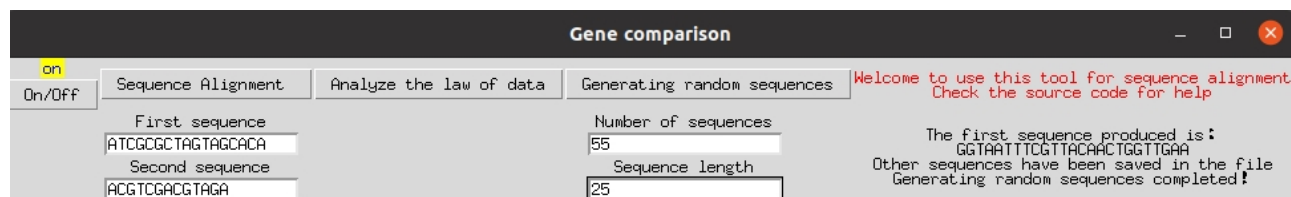
函数具体的实现思路很简单

```
from random import *
pairwise = "AAATTTCCGG"
def getseq(num):
    seq = []
    for i in range(num):
        seq.append(pairwise[randint(0, 9)])
    return seq
```

因为查阅文献可以知道，一号染色体的AT与CG比例为3：2，所以我静态定义了一个字符串 `pairwise`，其中的AT与CG比例正好符合要求，然后用生成随机数的方式从这个字符串里取出元素整合到新生成的序列中，然后执行这个函数规定次数，将所有生成的序列按行写入一个文件 `seqs.txt` 中

使用手册

运行 `smallprogram.py` 文件，首先打开 `On`，然后输入要求的长度和数量，点击上方按钮即可输出第一条符合要求的序列，然后在文件夹中的 `seqs.txt` 文件中即可找到其他所有满足该要求的序列



如果想要生成特定比例的序列只需修改原文件中的pairwise字符串的AT与CG比例即可

3.验证score服从的分布以及数据特征

思路

这个是专业课的一个作业题，生成50条比例符合人类染色体1号序列的序列，然后任一两条之间进行比对生成score文件存储最终得分和seq文件存放最佳比对结果。

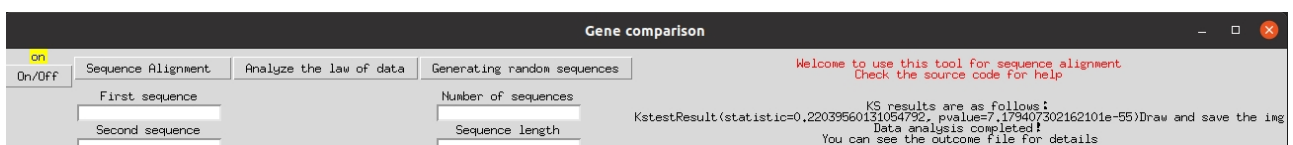
然后对于score文件绘制直方图判断其是否服从高斯分布，并计算score的平均值，标准差和进行k-s检验的结果

具体实现

具体的实现函数和序列比对类似，只是多了loadDta函数读取score.txt文件并进行数据处理，draw_hist函数绘制直方图并存储为outcome.png,然后将数据处理的结果全部输出到outcome.txt文件，并输出k-s检验的结果到软件中

使用手册

运行smallprogram.py文件，首先打开on，然后点击中间上方按钮即可输出k-s检验结果，然后在文件夹中的seq.txt,score.txt,outcome.txt,outcome.png中可以查看详细结果



打开(O)▼

score.txt

保存(S)

≡

—

□

✕

~/PycharmProjects/dp_program

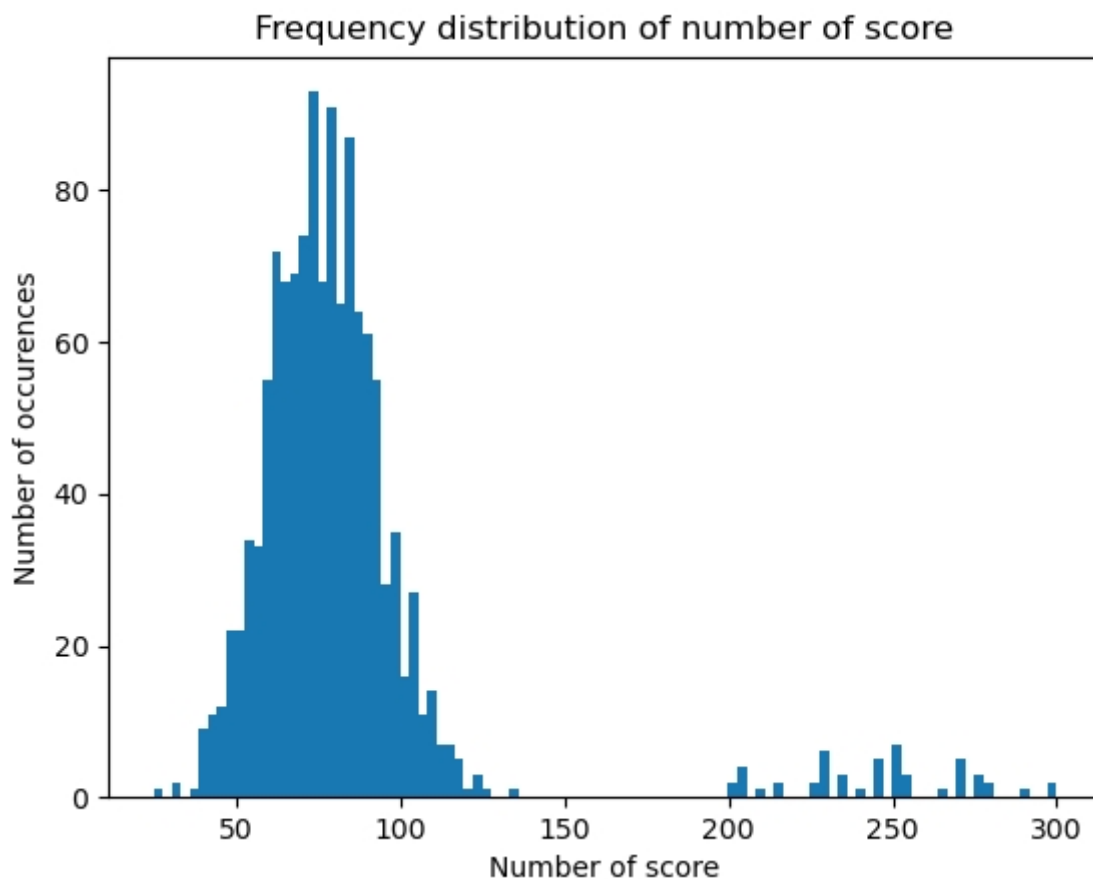
1222 78.0
1223 58.0
1224 82.5
1225 65.0
1226 67.0
1227 61.5
1228 53.0
1229 52.0
1230 70.5
1231 300.0
1232 78.5
1233 54.0
1234 94.5
1235 100.5
1236 117.5
1237 70.5
1238 73.5
1239 80.0
1240 200.0
1241 36.0
1242 68.0
1243 43.5
1244 73.5
1245 60.5
1246 49.0
1247 70.5
1248 265.0
1249 65.0
1250 60.0
1251 71.5
1252 104.5
1253 88.0
1254 80.0
1255 200.0
1256 48.5
1257 73.5
1258 61.0
1259 56.0
1260 72.5
1261 270.0
1262 103.5
1263 98.0
1264 48.0
1265 71.0
1266 245.0
1267 63.5
1268 72.0
1269 81.5
1270 270.0
1271 69.5
1272 71.5
1273 235.0
1274 71.5
1275 230.0

纯文本 ▼ 制表符宽度: 8 ▼ 第 1 行, 第 1 列 ▼ 插入

打开(O)seq.txt~/PycharmProjects/dp_program保存(S)

8883
8884 AAGGCTAATGCCTTATAGTCGTTAGAGCCGACGGTGCCCCAACTTTAGGTCACC 和
8885 AAGGCTAATGCCTTATAGTCGTTAGAGCCGACGGTGCCCCAACTTTAGGTCACC
8886 的序列比对的得分为270.0
8887 最佳匹配结果为
8888 CCACTGGATTTCAACCCCGTGGCAGCCGAGATTGCTGATATTCGTAATCGGAA 和
8889 CCACTGGATTTCAACCCCGTGGCAGCCGAGATTGCTGATATTCGTAATCGGAA
8890
8891 AAGGCTAATGCCTTATAGTCGTTAGAGCCGACGGTGCCCCAACTTTAGGTCACC 和
8892 AAGGTGCGATAGGGAATTCCAGTCTTTTTCTGAATAGTTGGACCCGA
8893 的序列比对的得分为69.5
8894 最佳匹配结果为
8895 ---CCACTGGATT--TCAACCC-----CGTGGCAGC-CG---A--GATT-GC-T-GATAT-TCCGTAATCGGAA 和
8896 AGCCCA--GG-TTGAT-AAGT-CTTTTC-T-G-A--C-CTTAAGGGA-TAGCGTGG---A-----A
8897
8898 AAGGCTAATGCCTTATAGTCGTTAGAGCCGACGGTGCCCCAACTTTAGGTCACC 和
8899 ATAGTTCAAAGAAGCAGGCTTATAGCCTGGGAAGTACGAATGATA
8900 的序列比对的得分为71.5
8901 最佳匹配结果为
8902 CCACTGG--A-----T-TT--CAACCC-----C-G-T-GGCAGCCGAGATTGCTGATATTCGTAATCGGAA 和
8903 --A-TAGTA-AGCATGAAGGTCCGATATTCGG-A--CGAAGAAACCTTGA-T-----A
8904
8905 AAGGTGCGATAGGGAATTCCAGTCTTTTTCTGAATAGTTGGACCCGA 和
8906 AAGGTGCGATAGGGAATTCCAGTCTTTTTCTGAATAGTTGGACCCGA
8907 的序列比对的得分为235.0
8908 最佳匹配结果为
8909 AGCCCAGGTTGATAAGTCTTTTTCTGACCTTAAGGGATAGCGTGGAA 和
8910 AGCCCAGGTTGATAAGTCTTTTTCTGACCTTAAGGGATAGCGTGGAA
8911
8912 AAGGTGCGATAGGGAATTCCAGTCTTTTTCTGAATAGTTGGACCCGA 和
8913 ATAGTTCAAAGAAGCAGGCTTATAGCCTGGGAAGTACGAATGATA
8914 的序列比对的得分为71.5
8915 最佳匹配结果为
8916 -----AGCCC----A-GGTT--G--A-----TAAGTCTTTTTCTGACCTTAAGGGATAGCGTGGAA 和
8917 ATAGTAAG--CATGAAGGG-TCCGAT-ATTCGGACGAAGAAACCTTGAT-----A
8918
8919 ATAGTTCAAAGAAGCAGGCTTATAGCCTGGGAAGTACGAATGATA 和
8920 ATAGTTCAAAGAAGCAGGCTTATAGCCTGGGAAGTACGAATGATA
8921 的序列比对的得分为230.0
8922 最佳匹配结果为
8923 ATAGTAAGCATGAAGGTCCGATATTCGGACGAAGAAACCTTGATA 和
8924 ATAGTAAGCATGAAGGTCCGATATTCGGACGAAGAAACCTTGATA
8925

纯文本 制表符宽度: 8 第 1 行, 第 1 列 插入



三.总结

实现的几个小功能其实都还比较简单，写起来没有啥问题.GUI编程对于所使用的包还不太熟悉，数据分析这一块还可以做很多东西，后续会逐渐补齐这一部分功能。

`tinker`包做这种可视化界面感觉终究是有一些麻烦的，并且美观性和功能性还有所欠缺，后续我会用`django`做一个网页的实现，然后加入更多使用的功能逐步完善这个小程序。

代码文件已上传至作者的github（egotist0），可以在里面获得实现文件，如果感兴趣或想提出一些建议的话,可以给本人的 Github 留言或直接参与修改