DNA (Евдокимов)

Описание программы

Задача программы состоит в выравнивании двух последовательностей, которое используется в биоинформатике при построении выравниваний аминокислотных или нуклеотидных последовательностей в Python.
Используется алгоритм Нидлмана-Вунша.
Для начала по заданным данным генерируются 2 последовательности ДНК. Затем высчитываются коэфициенты для подсчета матрицы схожести: d - штраф за разрыв равный длине самой длинной последовательности, A = коэфициент схожести символов равный квадрату штрафа за разрыв.
Затем непосредственно составляется матрица. В конце концов по конечной матрицы мы можем востановить необходимое нам выравнивание.

Тесты и эксперементы

(Можно самостоятельно запустить программу) Проверим программу на разных данных:

Тест 1

Вот наши последовательности
GCACT
GGTGA
Проводим выравнивание
—GCACT
GGTG-A–
Объединяем
GGTGCACT

Тест 2

Вот наши последовательности
TCGGAGGTTT
GTACAGCTTG
Проводим выравнивание
-T-CGGAG-GTT-T
GTAC–AGC-TTG-
Объединяем
GTACGGAGCGTTGT

Тест 3

Вот наши последовательности
TGAAAGGTATCGAATTCCCTAACTACGGTA
ACATTAGCCTTCCCGACCTCAAAAGGCCAAGTCCT
Проводим выравнивание
TGA-A–AG–GTAT–CGA–ATTC------CCTAA–CTACGGTA
–ACATTAGCC-T-TCCCGACC–TCAAAAGGCC-AAGTC–C--T-
Объединяем
TGACATTAGCCGTATCCCGACCATTCAAAAGGCCTAAGTCTACGGTA

Тест 4

Протестируем программу на последовательности длинной 1000. Чтобы оценить правильность выравнивания, возьмем две одинаковые последовательности, и у второй изменим 5 случайных символов. Следовательно выравнивание верно, если длина выравнивания не больше чем на 5 привышает длину оригинала. Результат в файле.

Тест 5

На 10000 символах программа работала 5 минут и, проведя ту же проверку, получаем положительный результат. Результат в файле.

Тест 6

Наконец тест на 10^5 символов. Программа работала более 24 часов и в конце концов на моем компьютере не хватило оперативной памяти, так для хранения требуется 10 гб.

Выводы

Смотря на результаты тестов, можно утверждать, что длина последовательности влияет на загруженность оперативной памяти. Необходимая память для обработки равна n * m байт. Один из вариантов решения данной проблемы может быть разделение матрицы на два отдельных файла, тогда они не будут засорять оперативную память