Pendekatan *Dynamic Programming* untuk Menyelesaikan *Sequence Alignment*

Ray Andrew Obaja Sinurat - 13515073
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13515073@std.stei.itb.ac.id

***Abstract*—*Sequence Alignment* merupakan cara dalam menyusun urutan DNA, RNA, atau protein untuk menentukan adanya kesamaan fungsional atau struktural diantara urutan tersebut. Sequence Alignment dapat dipermudah dengan menggunakan *dynamic programming*.**

***Keywords*—DNA, RNA, sequence, dynamic, programming, algorithm.**

I. PENDAHULUAN

Algoritma sudah sangat pelak dikaitkan jika berbicara tentang informatika. Banyak ilmu-ilmu yang dipelajari di dunia menggunakan informatika sebagai sarana dalam mempermudah pemrosesan, kalkulasi, visualisasi, dan bahkan prediksi.

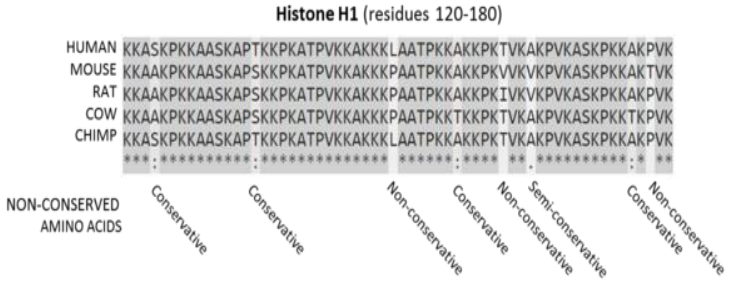
Ada banyak macam dari strategi algoritma, contohnya adalah *brute force*, *greedy*, *divide and conquer*, *decrease and conquer*, *branch and bound*, *backtrack*, *dynamic programming*, dan masih banyak lagi.

Berbagai macam strategi diciptakan untuk menyelesaikan masalah yang berbeda-beda. Suatu strategy bisa saja menjadi optimisasi dari strategi yang lain. Hal ini dilakukan untuk mendapat kompleksitas algoritma yang rendah sehingga ketika dieksekusi, waktu yang diperoleh juga akan seminimum mungkin.

Strategi algoritma sering sekali dipakai untuk memecahkan suatu masalah dari disiplin ilmu lain, salah satu contohnya adalah *bioinformatics. Bioinformatics* merupakan disiplin ilmu untuk megembangkan metode dan aplikasi untuk menganalisis data-data biologi. *Bioinformatics* menggabungkan *computer science*, statistic, matematika, dan rekayasa untuk menganalisis dan mengartikan data-data biologi.

Sequence alignment merupakan salah satu penerapan algoritma yang digunakan dalam *bioinformatics.* Hal ini digunakan agar susunan urutan DNA atau RNA atau protein agar menentukan adanya kesamaan fungsional atau struktural.

Banyak cara untuk melakukan analisis terhadap *sequence alignment*, seperti menggunakan *brute force*, program dinamis, dan juga probabilistik. Namun, pada makalah kali ini, hanyalah yang menggunakan program dinamis yang akan dibahas.



Gambar 1.1 *sequence alignment, produced by ClustalO*
(sumber : https://en.wikipedia.org/wiki/Sequence_alignment)

II. DASAR TEORI

A. Program Dinamis

Metode pemecahan masalah dengan cara menguraikan solusi menjadi sekumpulan langkah atau tahapan sedemikian sehingga solusi dari persoalan dapat dipandang dari serangkaian keputusan yang saling berkaitan.

Karakteristik dari persoalan program dinamis adalah :

a. Persoalan dapat dibagi menjadi beberapa tahap, yang pada setiap tahap hanya diambil satu keputusan
b. Masing-masing tahap terdiri dari sejumlah status yang berhubungan dengan tahap tersebut
c. Hasil dari keputusan yang diambil pada setiap tahap ditransofrmasikan dari status yang bersangkutan ke status berikutnya pada tahap berikutnya
d. Ongkos pada suatu tahap meningkat secara teratur dengan bertambahnya jumlah tahapan
e. Ongkos pada suatu tahap bergantung pada ongkos tahap-tahap yang sudah berjalan dan ongkos pada tahap tersebut
f. Keputusan terbaik pada suatu tahap bersifat independen terhadap keputusan yang dilakukan pada tahap sebelumnya
g. Ada hubungan rekursif yang mengidentifikasikan kepututsan terbaik untuk setiap kasus pada tahap k memberikan keputusan terbaik untuk setiap status pada

tahap k+1

h. Prinsip optimalitas berlaku pada persoalan tersebut

Pendekatan yang dilakukan pun dapat berbeda yaitu :

a. Maju (forward atau top-down) dimana program dinamis akan memulai dari tahap awal hingga tahap akhir
b. Mundur (backward atau bottom-up) dimana program dinamis akan memulai dari tahap paling akhir hingga tahap awal

(Diktat Kuliah IF2211 Strategi Algoritma, Rinaldi Munir)

Program dinamis biasanya membentuk sebuah penyimpanan sementara suatu state agar menjadi data bagi state selanjutnya. Program dinamis juga biasanya diperoleh dari pendekatan secara matematika sehingga biasanya ada formula matematika untuk memecahkannya.

B. Sequence Alignment

Alignment adalah ketika dua teks representasi dari urutan DNA atau protein dibandingkan secara bersamaan sehingga elemen yang sangat mirip akan ditempatkan sedemikian rupa sehingga sejajar satu sama lain. Banyak sekali tugas-tugas yang dapat diselesaikan pada disiplin ilmu *bioinformatics* jika sukses dalam *alignment.*

Alighment merupakan hal yang esensial karena jika kita berhasil melakukan *aligment* sedemikian rupa, maka akan terdapat *traces* dan gambaran dari DNA atau protein yang sedang diteliti. Jika terdapat kemiripan dengan DNA atau protein yang pernah diteliti, maka akan lebih mempersingkat waktu dan mempermudah pengerjaan.

Setiap elemen yang terdapat di dalam *traces* bisa saja *match* atau *gap*. Keadaan *match* terjadi jika elemen ke-j dari kedua *sequence* sama ataupun terjadi kemiripan. Dimana di DNA dikenal dengan purin dan pirimidin. Sesama purin yaitu A dan G akan memiliki kemiripan secara kimiawi, begitu juga dengan T dan C. Sedangkan keadaan *gap* adalah ketika terjadi ketidakcocokan antara elemen dari kedua sekuens. Hal ini terjadi karena berbeda tipe sehingga tidak mirip secara kimiawi.

Selain kedua hal diatas, *trace* juga dapat merepresentasikan berbagai hal seperti *substitution*, *deletion,* dan *insertion.*

a. Substitution
   J K S J M N
   J K M J M N

b. Deletion
   J K S J M N
   J K S J – N

c. Insertion
   AB-D
   ABCD

Pada algoritma program dinamis didapatkan istilah *traceback. Traceback* merupakan tahapan-tahapan yang penyusunan solusi setelah tabel berhasil dikalkulasikan. Jika menggunakan top-down, maka *traceback* akan dimulai dari akhir tabel. Jika menggunakan bottom-up, maka traceback akan dimulai dari awal tabel.

III. PROGRAM DINAMIS UNTUK MEMUDAHKAN *SEQUENCE ALIGNMENT*

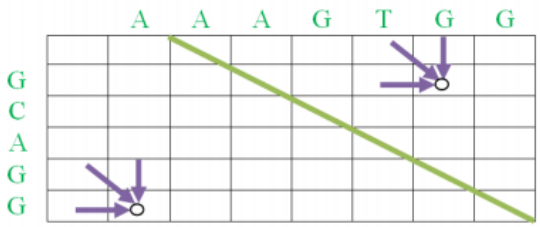
Sebagai contoh akan dicoba kecocokan antara rantai DNA “AAAGTGG” dan “GCAGG”

- Jenis-jenis algoritma :

A. *Weighted Alignment* (WA)

Ide dari *Weighted Alignment* adalah mengambil bobot sedemikian rupa sehingga aliran pergerakan mendekati diagonal seperti pada gambar 3.1.

Pada *Weighted Alignment*, nukleotida yang sama diharapkan terkumpul pada diagonal *aligment matrix.* Sehingga disimpulkan, apapun yang lebih dekat dengan diagonal pasti memiliki bobot yang lebih rendah dibandingkan yang lain.



Gambar 3.1 Weight Matrix
(source : uu.diva-portal.org)

Formula dari *Weighted Alignment* adalah :

$$S[i,j] = \max\{S[i-1,j] - \frac{1}{m-i+1}\beta, S[i,j-1] - \frac{1}{n-j+1}\beta, S[i-1,j-1] + \varrho\delta(x_i,y_j)\},$$

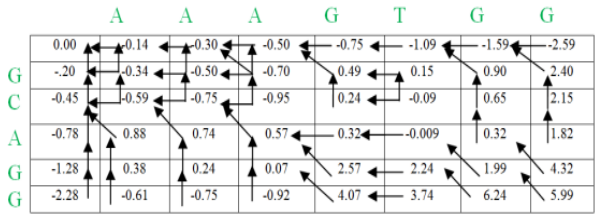
dimana :

$$\varrho = \max\{\frac{1}{m-i+1}, \frac{1}{n-j+1}\}.$$

dengan menentukan bobot seperti di atas, maka kita akan membuat pergerakan bobot bergeser kearah diagonal. Sehingga ketika membuat kolom pertama dan baris pertama kita dapat menggunakan formula :

$$
\begin{aligned}
S[0,0] &= 0, \\
S[0,j] &= S[0,j-1] - \frac{1}{n-j+1}\beta, \\
S[i,0] &= S[i-1,0] - \frac{1}{m-i+1}\beta.
\end{aligned}
$$

Weight Matrix memiliki ketentuan dalam mengisi tabel yakni sebagai berikut :



Gambar 3.2 *Weight Matrix Filling Step*
(source : uu.diva-portal.org)

Sehingga diperolehlah proses *traceback* dari *Weighted Alignment* sebagai berikut :

Gambar 3.3 *Weight Matrix TraceBack*
(source : uu.diva-portal.org)

Hasil perhitungan :

1. Jika menggunakan jalur bawah, maka diperoleh :

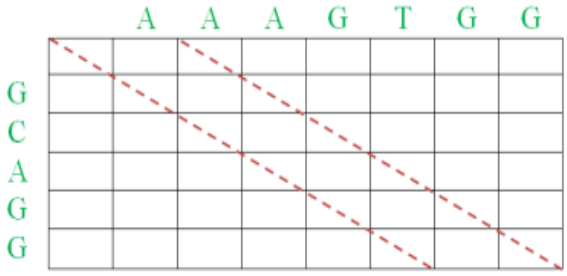
G	C	φ	φ	A	φ	φ	G	G
φ	φ	A	A	A	G	T	G	G
-	-	-	-	+	-	-	+	+
1/5	1/4	1/7	1/6	4/3	1/4	1/3	2	4

2. Jika menggunakan jalur atas, maka diperoleh :

φ	φ	G	C	A	Φ	φ	G	G
A	A	φ	φ	A	G	T	G	G
-	-	-	-	+	-	-	+	+
1/7	1/6	1/5	1/4	4/3	1/4	1/3	2	4

B. Diagonal Aligment

Dari pendekatan ini, diperlukan garis diagonal pada tabel. Diagonal yang dimaksudkan tidaklah pojok ke pojok dikarenakan adanya perbedaan panjang antara kedua rantai yang akan di-*align*.



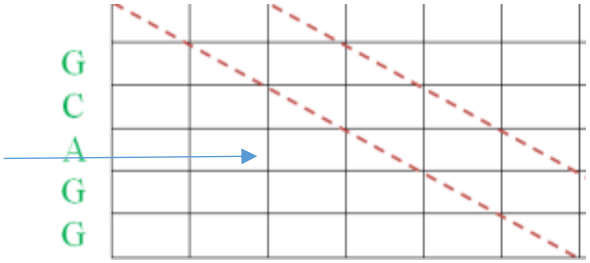
Gambar 3.4 Diagonal Matrix
(source : uu.diva-portal.org)

Untuk mendapatkan bobot yang sesuai pertama-tama tabel harus diinisalisasi dengan :

$$S[0,0] = 0,\ S[i,0] = -\beta \times i,\ S[0,j] = -\beta \times j,$$

Namun, untuk mendapatkan bobot dari sel yang lain, maka tabel harus dibagi menjadi beberapa daerah yang berbeda perhitungannya :

a. Sel yang berada pada segitiga bawah

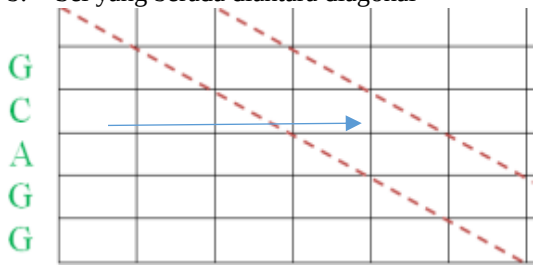


Gambar 3.5 Diagonal Matrix Lower Triangle
(source : uu.diva-portal.org)

memiliki formula sebagai berikut :

$$S[i,j] = \max\{S[i-1,j], S[i,j-1] - \beta, S[i-1,j-1] + \delta(x_i, y_j)\}.$$

b. Sel yang berada diantara diagonal

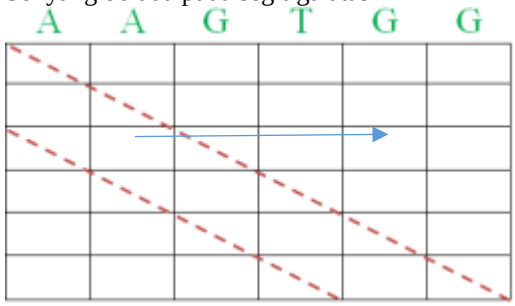


Gambar 3.6 Diagonal Matrix
(source : uu.diva-portal.org)

memiliki formula sebagai berikut :

$$S[i,j] = \max\{S[i-1,j] - \beta, S[i,j-1] - \beta, S[i-1,j-1] + \delta(x$$

c. Sel yang berada pada segitiga atas

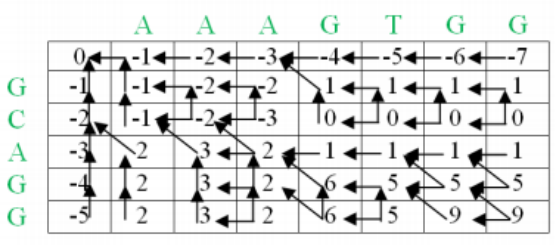


Gambar 3.7 Diagonal Matrix Upper Triangle
(source : uu.diva-portal.org)

memiliki formula sebagai berikut :

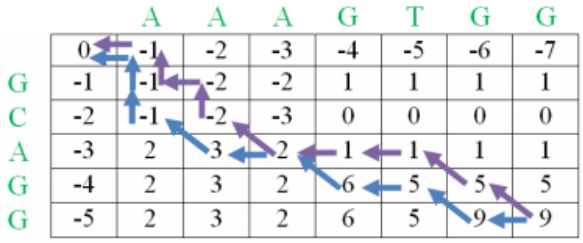
$$S[i,j] = \max\{S[i-1,j] - \beta, S[i,j-1], S[i-1,j-1] + \delta(x_i$$

Diagonal Matrix memiliki ketentuan dalam mengisi tabel yakni sebagai berikut :



Gambar 3.8 Diagonal Matrix Filling Step
(source : uu.diva-portal.org)

Sehingga diperolehlah proses *traceback* dari *Diagonal Alignment* sebagai berikut :



Gambar 3.9 Diagonal Matrix TraceBack
(source : uu.diva-portal.org)

Hasil perhitungan :

1. Jika menggunakan jalur bawah, maka diperoleh :

φ	G	C	A	φ	G	φ	G	Φ
A	φ	φ	A	A	G	T	G	G
-	+	+	+	-	+	-	+	+
1	0	0	4	1	4	1	4	0

2. Jika menggunakan jalur atas, maka diperoleh :

φ	G	φ	C	A	φ	φ	G	G
A	φ	A	φ	A	G	T	G	G
-	+	-	+	+	-	+	+	+
1	0	1	0	4	1	0	4	4

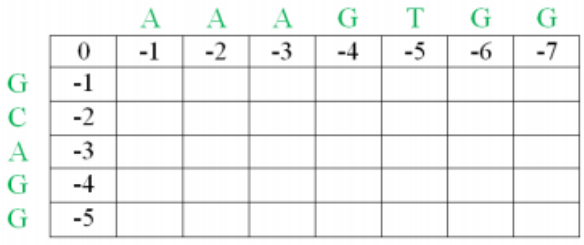
C. Global Alignment

Disebut juga Needleman-Wusch Algorithm, dilakukan dengan cara melakukan *align* pada kedua sekuens secara global dan mencakup keseluruhan sekuens.

Cara melakukan inisialisasi adalah sebagai berikut :

$$S[0,0] = 0,\ S[i,0] = -\beta \times i,\ S[0,j] = -\beta \times j.$$

Sehingga diperolehlah :

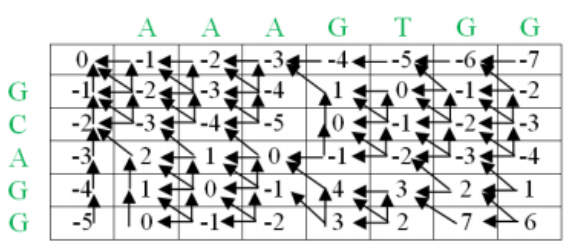


Gambar 3.10 Global Alignment Matrix
(source : uu.diva-portal.org)

Untuk sel yang lain, berlaku formula :

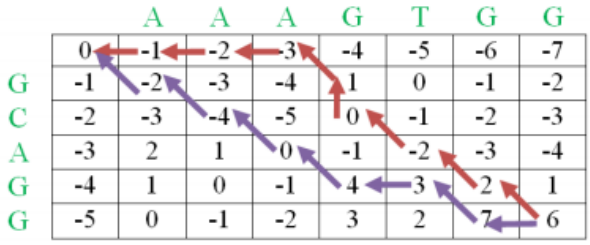
$$S[i,j] = \max\{S[i-1,j] - \beta, S[i,j-1] - \beta, S[i-1,j-1] + \delta(x_i, y_j)\}.$$

Pada Global Alignment, memiliki ketentuan dalam mengisi tabel yakni sebagai berikut :



Gambar 3.11 Global Alignment Matrix Filling Step
(source : uu.diva-portal.org)

Sehingga diperolehlah proses traceback dari *Global alignment* sebagai berikut :



Gambar 3.11 Global Alignment Matrix TraceBack

(source : uu.diva-portal.org)

Hasil perhitungan :

1. Jika menggunakan jalur bawah, maka diperoleh :

G	C	A	G	φ	G	φ
A	A	A	G	T	G	G
-	-	+	+	-	+	-
2	2	4	4	1	4	1

2. Jika menggunakan jalur atas, maka diperoleh :

φ	φ	φ	G	C	A	G	G
A	A	A	G	φ	T	G	G
-	-	-	+	-	-	+	+
1	1	1	4	1	2	4	4

- Kompleksitas Algoritma

Meskipun melakukan penyimpanan dan perhitungan pada tabel menggunakan formula yang ada, ketiganya tetap memiliki kompleksitas yang sama yaitu O(mn), dimana m adalah banyaknya karakter dari rantai pertama, dan n adalah banyaknya karakter dari rantai kedua.

Sehingga algoritma ini masih bisa diselesaikan secara cepat dalam waktu yang masih *polynomial*.

- Pseudocode

A. Weight matrix

```
if i = 1 or j = 1 then
  if i > 0 then
    for i' ← 1 to i do
      print xi' φ
    end
  end
  if j > 0 then
    for j' ← 1 to j do
      print φ yj'
    end
  end
  return
end
if S[i, j]=S[i − 1, j − 1] + max{ 1/(m−i+1) ,
1/(n−j+1)}δ(xi , yj) then
 WEIGHTEDALIGNMENTOUTPUT(X,Y,S,i-1,j-1)
  print (xi , yj )
else if S[i, j] = S[i − 1, j] − β / (m−i+1) then
  WEIGHTEDALIGNMENTOUTPUT(X,Y,S,i-1,j)
  print (xi , φ )
else
   WEIGHTEDALIGNMENTOUTPUT(X,Y,S,i,j-1)
  print (φ , yj)   end
```

B. Diagonal Matrix

```
S[0, 0] ← 0 for j ← 1 to n do
  S[0, j] ← −β × j
end
for i ← 1 to m do
  S[i, 0] ← −β × i
  for j ← 1 to n do
   if (i < j and j < i + (n − m)) then
    S[i, j] = max{ S[i − 1, j] − β, S[i, j − 1] − β, S[i − 1, j
− 1] + δ(xi , yj ) }
   else if (j >= i + (n − m)) then S[i, j] = max{S[i − 1, j]
− β, S[i, j − 1], S[i − 1, j − 1] + δ(xi , yj )}
   else S[i, j] = max{S[i−1, j], S[i, j−1] − β, S[i−1, j−1] +
δ(xi ,yj )}
   end
  end
end
Output S[m, n]
```

C. Global Alignment

```
S[0, 0] ← 0
for j ← 1 to n do
  S[0, j] ← −β × j
end
for i ← 1 to m do
  S[i, 0] ← −β × i
  for j ← 1 to n do
    S[i, j] = max { S[i − 1, j] − β, S[i, j − 1] − β, S[i − 1, j
− 1] + δ(xi , yj )}
  end
end
Output S[m, n]
```

IV. KESIMPULAN

Program dinamis merupakan salah satu strategi algoritma untuk memecahkan masalah dengan menguraikan solusi menjadi kumpulan langkah. Program dinamis dapat digunakan untuk membantu memecahkan *Sequence Alignment* untuk memudahkan disiplin ilmu *bioinformatics* untuk memyusun rangkaian DNA atau protein sedemikian rupa sehingga mendapat kesamaan fungsional atau struktrural. Terdapat banyak sekali jenis-jenis dalam *sequence alignment* mulai dari Global Alignment, Weighted Alignment, Diagonal Alignment, Single Alignment, dan masih banyak lagi.

V. UCAPAN TERIMA KASIH

Pertama – tama saya mengucapkan terima kasih kepada Tuhan Yang Maha Esa yang telah melimpahkan berkat dan kasih-Nya sehingga makalah Strategi Algoritma ini dapat diselesaikan tepat waktu. Saya juga mengucapkan terima kasih kepada kedua orang tua saya yang selalu memberi dukungan dan doa restu kepada saya sehingga dapat menempuh pendidikan sampai saat ini. Tak lupa saya juga mengucapkan terima kasih kepada Dr. Ir. Rinaldi Munir, M.T, Dr. Masayu Leylia Khodra, S.T., M.T., dan Dr. Nur Ulfa Maulidevi, S.T., M.Sc., yang berperan sebagai dosen mata kuliah IF 2211 Strategi Algoritma sehingga dengan ilmu pengetahuan seputar Strategi Algoritma, saya dapat membuat dan menyelesaikan makalah ini. Saya juga tidak lupa mengucapkan terima kasih kepada teman-teman dalam membantu saya menentukan topik untuk makalah ini. Terima kasih.

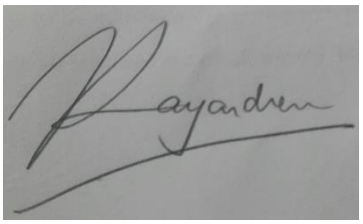
REFERENSI

[1] Munir, Rinaldi. Diktat Kuliah IF2211 Strategi Algoritma, 2009.
[2] Alimehr, Leila. The Performance of Sequence Alignment Algorithms, Uppala Universitet, 2013.
[3] http://www.cs.tau.ac.il/~rshamir/algmb/01/scribe03/lec03.pdf. Diakses tanggal 18 Mei 2017.
[4] https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-096-algorithms-for-computational-biology-spring-2005/lecture-notes/lecture5_newest.pdf. Diakses tanggal 18 Mei 2017.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 18 Mei 2017



Ray Andrew Obaja Sinurat – 13515073
Teknik Informatika 2015
Institut Teknologi Bandung

LAMPIRAN

Visualisasi dengan bahasa C++ untuk algoritma Global Alignment

```
PS C:\Users\user\Downloads\SequenceAlignment\lel> g++ nw.cpp main.cpp -o main
PS C:\Users\user\Downloads\SequenceAlignment\lel> ./main AAAGTGG GCAGG -p

Dynamic programming matrix:

        A   A   A   G   T   G   G
    0  -1  -2  -3  -4  -5  -6  -7
G  -1  -1  -2  -3  -1  -2  -3  -4
C  -2  -2  -2  -3  -2  -2  -3  -4
A  -3   0   0   0  -1  -2  -3  -4
G  -4  -1  -1  -1   2   1   0  -1
G  -5  -2  -2  -2   1   1   3   2

Traceback matrix:

    A A A G T G G
  n - - - - - - -
G | \ - - \ - - -
C | | \ - | \ - -
A | \ \ \ - - - -
G | | | | \ - - -
G | | | | | \ \ -

AAAGTGG
GCAG-G-
```