DNATool

Generated by Doxygen 1.9.5

Chapter 1

eda-lvl5-dnatool README

- Fecha de entrega: 10/05/2022
- Grupo: 7
  - DALZOTTO, Rafael
  - HEIR, Alejandro Nahuel

22.08 - Algoritmos y Estructuras de Datos - Ingeniería Electrónica - ITBA

1.1 Respuestas

1.1.1 Mejores alineamientos para los mystery genomes:

- Mystery genome 1 - OV806939
  - Delta B.1.617.2 - OM202516
  - Puntaje óptimo: 29826
- Mystery genome 2 - MT345875
  - Beta B.1.351 - MZ314996
  - Puntaje óptimo: 29865
- Mystery genome 3 - OU772970
  - Delta B.1.617.2 - OM202516
  - Puntaje óptimo: 29840

Los alineamientos óptimos correspondientes se encuentra en `/resources/test/results/`

1.2 Bonus points

- La complejidad computacional del algoritmo Needleman-Wunsch es O(mn) debido a que el calculo de las direcciones (o scores) de la matriz de alineacion depende de los largos de ambas secuencias.

  Sin embargo logramos reducir sustancialmente el uso de memoria, gracias a aprovechar cada bit en los elementos de la matriz de direcciones (DirectionMat) y debido a que no usamos una matriz de scores, resultando en un uso de memoria de no mas de 250 MB para comparar dos secuencias de unos 30000 pares de base (resultados experimentales). En cambio al usar una matriz de scores de `int32_t`, el uso de memoria se eleva a mas de 3 GB.

- Al cambiar los valores en el sistema de puntaje, cambia el puntaje final de alineamiento óptimo y tambien puede cambiar el alinieamiento optimo en si

- Comparamos los genomas de SARS-CoV-2003 y MERS-CoV con Wuhan reference genome
  - Puntaje de MERS-CoV vs Wuhan reference genome: 7240
  - Puntaje de SARS-CoV-2003 vs Wuhan reference genome: 18645
  - Los alineamientos óptimos correspondientes se encuentran en `/resources/test/results/`

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Class Documentation

4.1 DirCell Struct Reference

DirCell contains 4 directions in a single byte.

`#include <DirectionMat.h>`

Public Attributes

- uint8_t d0: 2
- uint8_t d1: 2
- uint8_t d2: 2
- uint8_t d3: 2

4.1.1 Detailed Description

DirCell contains 4 directions in a single byte.

4.1.2 Member Data Documentation

4.1.2.1 d0

`uint8_t DirCell::d0`

4.1.2.2 d1

`uint8_t DirCell::d1`

4.1.2.3 d2

```
uint8_t DirCell::d2
```

4.1.2.4 d3

```
uint8_t DirCell::d3
```

The documentation for this struct was generated from the following file:

- DirectionMat.h

4.2 DirectionMat Class Reference

```
#include <DirectionMat.h>
```

Public Member Functions

- DirectionMat (const size_t rows, const size_t cols)
  Constructs a new Direction Mat object.
- ~DirectionMat ()
  Destroys the Direction Mat object.
- uint8_t at (size_t row, size_t col)
  Reads a given DirectionMap cell.
- void set (uint8_t val, size_t row, size_t col)
  Writes a given DirecitonMat cell.
- void print ()
  Prints the given DirectionMat, used for debugging.

Public Attributes

- const size_t rows
- const size_t cols

4.2.1 Constructor & Destructor Documentation

4.2.1.1 DirectionMat()

```
DirectionMat::DirectionMat (
            const size_t rows,
            const size_t cols )
```

Constructs a new Direction Mat object.

Parameters

rows	Total rows
cols	Total cols

4.2.1.2 ~DirectionMat()

```
DirectionMat::~DirectionMat ( )
```

Destroys the Direction Mat object.

4.2.2 Member Function Documentation

4.2.2.1 at()

```
uint8_t DirectionMat::at (
            size_t row,
            size_t col )
```

Reads a given DirectionMap cell.

Parameters

row	
col	

Returns

uint8_t

4.2.2.2 print()

```
void DirectionMat::print ( )
```

Prints the given DirectionMat, used for debugging.

Parameters

mat	initialized DirectionMat

4.2.2.3 set()

```
void DirectionMat::set (
            uint8_t val,
            size_t row,
            size_t col )
```

Writes a given DirecitonMat cell.

Parameters

val	Data to store
row	
col	

4.2.3 Member Data Documentation

4.2.3.1 cols

```
const size_t DirectionMat::cols
```

4.2.3.2 rows

```
const size_t DirectionMat::rows
```

The documentation for this class was generated from the following files:

- DirectionMat.h
- DirectionMat.cpp

Chapter 5

File Documentation

5.1 Aligner.cpp File Reference

Genetic sequence aligner, based on Needleman-Wunsch algorithm.

```
#include "Aligner.h"
#include <iostream>
#include <cstdint>
#include <algorithm>
#include <vector>
```

Enumerations

- enum Direction { VERTICAL , DIAGONAL , HORIZONTAL }

Functions

- void initMat (DirectionMat &mat)

  Initializes a DirectionMat for storing Needleman-Wunsch directions.
- int32_t fillMat (DirectionMat &mat, const string &seq1, const string &seq2)

  Completes a DirectionMat based on two given genetic sequences.
- void buildAligment (DirectionMat &mat, const string &seq1, const string &seq2, array< string, 3 > &output)

  Builds the optimal alignment for two genetic sequences and an already filled DirectionMat.
- int32_t getGlobalAlignment (const string &seq1, const string &seq2, array< string, 3 > &alignment)

  Runs aligner for two acgt-like strings.

5.1.1 Detailed Description

Genetic sequence aligner, based on Needleman-Wunsch algorithm.

Authors

DALZOTTO, Rafael; HEIR, Alejandro - Grupo 7

Version

0.1

Date

2022-05-08

Copyright

5.1.2 Enumeration Type Documentation

5.1.2.1 Direction

```
enum Direction
```

Enumerator

VERTICAL	
DIAGONAL	
HORIZONTAL	

5.1.3 Function Documentation

5.1.3.1 buildAligment()

```
void buildAligment (
            DirectionMat & mat,
            const string & seq1,
            const string & seq2,
            array< string, 3 > & output )
```

Builds the optimal alignment for two genetic sequences and an already filled DirectionMat.

Parameters

mat	
seq1	String with one acgt-like genetic sequence.
seq2	String with one acgt-like genetic sequence.
output	3-string array to store the optimal alignment.

5.1.3.2 fillMat()

```
int32_t fillMat (
            DirectionMat & mat,
            const string & seq1,
            const string & seq2 )
```

Completes a DirectionMat based on two given genetic sequences.

Parameters

mat	
seq1	String with one acgt-like genetic sequence.
seq2	String with one acgt-like genetic sequence.

Returns

int32_t Optimal score

5.1.3.3 getGlobalAlignment()

```
int32_t getGlobalAlignment (
            const string & seq1,
            const string & seq2,
            array< string, 3 > & alignment )
```

Runs aligner for two acgt-like strings.

Parameters

seq1	String with one acgt-like genetic sequence.
seq2	String with one acgt-like genetic sequence.
alignment	3-string array to store the optimal alignment.

Returns

int32_t Optimal alignment score

5.1.3.4 initMat()

```
void initMat (
            DirectionMat & mat )
```

Initializes a DirectionMat for storing Needleman-Wunsch directions.

Parameters

rows	
cols	

5.2 Aligner.h File Reference

Genetic sequence aligner, based on Needleman-Wunsch algorithm.

```
#include <string>
#include <array>
#include "DirectionMat.h"
```

Functions

- void initMat (DirectionMat &mat)

  Initializes a DirectionMat for storing Needleman-Wunsch directions.
- int32_t fillMat (DirectionMat &mat, const std::string &seq1, const std::string &seq2)
- void buildAligment (DirectionMat &mat, const std::string &seq1, const std::string &seq2, std::array< std::string, 3 > &output)
- int32_t getGlobalAlignment (const std::string &seq1, const std::string &seq2, std::array< std::string, 3 > &alignment)

5.2.1 Detailed Description

Genetic sequence aligner, based on Needleman-Wunsch algorithm.

Authors

DALZOTTO, Rafael; HEIR, Alejandro - Grupo 7

Version

0.1

Date

2022-05-08

Copyright

Copyright (c) 2022 - 22.08 EDA - ITBA

5.2.2 Function Documentation

5.2.2.1 buildAligment()

```
void buildAligment (
            DirectionMat & mat,
            const std::string & seq1,
            const std::string & seq2,
            std::array< std::string, 3 > & output )
```

5.2.2.2 fillMat()

```
int32_t fillMat (
            DirectionMat & mat,
            const std::string & seq1,
            const std::string & seq2 )
```

5.2.2.3 getGlobalAlignment()

```
int32_t getGlobalAlignment (
            const std::string & seq1,
            const std::string & seq2,
            std::array< std::string, 3 > & alignment )
```

5.2.2.4 initMat()

```
void initMat (
            DirectionMat & mat )
```

Initializes a DirectionMat for storing Needleman-Wunsch directions.

Parameters

rows	
cols	

5.3 Aligner.h

Go to the documentation of this file.

```
1
11 #ifndef ALIGNER_H
12 #define ALIGNER_H
13
14 #include <string>
15 #include <array>
16
17 #include "DirectionMat.h"
18
19 void initMat(DirectionMat &mat);
20
21 int32_t fillMat(DirectionMat &mat, const std::string &seq1, const std::string &seq2);
22
23 void buildAligment(DirectionMat &mat, const std::string &seq1, const std::string &seq2,
24                    std::array<std::string, 3> &output);
25
26 int32_t getGlobalAlignment(const std::string &seq1, const std::string &seq2,
27                            std::array<std::string, 3> &alignment);
28
29 #endif
```

5.4 DirectionMat.cpp File Reference

Implements 2 bit-size matrix cells.

```
#include "DirectionMat.h"
```

5.4.1 Detailed Description

Implements 2 bit-size matrix cells.

Authors

DALZOTTO, Rafael; HEIR, Alejandro - Grupo 7

Version

0.1

Date

2022-05-08

Copyright

Copyright (c) 2022 - 22.08 EDA - ITBA

5.5 DirectionMat.h File Reference

Implements 2 bit-size matrix cells optimized for memory usage.

```
#include <iostream>
```

Classes

- struct DirCell

  DirCell contains 4 directions in a single byte.
- class DirectionMat

5.5.1 Detailed Description

Implements 2 bit-size matrix cells optimized for memory usage.

Authors

DALZOTTO, Rafael; HEIR, Alejandro - Grupo 7

Version

0.1

Date

2022-05-08

Copyright

5.6 DirectionMat.h

Go to the documentation of this file.

```
1
11 #ifndef DIRECTIONMAT_H
12 #define DIRECTIONMAT_H
13
14 #include <iostream>
15
19 typedef struct
20 {
21     uint8_t d0 : 2;
22     uint8_t d1 : 2;
23     uint8_t d2 : 2;
24     uint8_t d3 : 2;
25 } DirCell;
26
27 class DirectionMat
28 {
29 public:
30     DirectionMat(const size_t rows, const size_t cols);
31     ~DirectionMat();
32
33     uint8_t at(size_t row, size_t col);
34     void set(uint8_t val, size_t row, size_t col);
35     void print();
36
37     const size_t rows;
38     const size_t cols;
39
40 private:
41     DirCell *data;
42
43     size_t compactCols;
44 };
45
46 #endif
```

5.7 GBReader.cpp File Reference

Sequence extractor for GenBank-like files.

```
#include "GBReader.h"
#include <algorithm>
```

Functions

- bool isNotACGT (char c)

  Checks if a given character is not 'a', 'c', 'g' or 't'.
- bool getSequence (string &output, istream &file)

  Extracts an acgt-like string from a GenBank file.

5.7.1 Detailed Description

Sequence extractor for GenBank-like files.

Authors

DALZOTTO, Rafael; HEIR, Alejandro - Grupo 7

Version

0.1

Date

2022-05-08

Copyright

5.7.2 Function Documentation

5.7.2.1 getSequence()

```
bool getSequence (
            string & output,
            istream & file )
```

Extracts an acgt-like string from a GenBank file.

Parameters

output	String to store the acgt sequence
file	istream file handler.

Returns

true Success

false Failure

5.7.2.2 isNotACGT()

```
bool isNotACGT (
            char c )
```

Checks if a given character is not 'a', 'c', 'g' or 't'.

Parameters

c	

Returns

true It's not

false It is

5.8 GBReader.h File Reference

Sequence extractor for GenBank-like files.

```
#include <iostream>
```

Functions

- bool getSequence (std::string &output, std::istream &file)

5.8.1 Detailed Description

Sequence extractor for GenBank-like files.

Authors

DALZOTTO, Rafael; HEIR, Alejandro - Grupo 7

Version

0.1

Date

2022-05-08

Copyright

5.8.2 Function Documentation

5.8.2.1 getSequence()

```
bool getSequence (
            std::string & output,
            std::istream & file )
```

5.9 GBReader.h

Go to the documentation of this file.

```
1
12 #ifndef GBREADER_H
13 #define GBREADER_H
14
15 #include <iostream>
16
17 bool getSequence(std::string &output, std::istream &file);
18
19 #endif
```

5.10 main.cpp File Reference

Runs the genetic sequence aligner.

```
#include <iostream>
#include <fstream>
#include <chrono>
#include "GBReader.h"
#include "Aligner.h"
```

Functions

- void printElapsedFrom (time_point< steady_clock > &timestamp)
- int main (int argc, char **argv)

5.10.1 Detailed Description

Runs the genetic sequence aligner.

Authors

DALZOTTO, Rafael; HEIR, Alejandro - Grupo 7

Version

0.1

Date

2022-05-08

Copyright

5.10.2 Function Documentation

5.10.2.1 main()

```
int main (
            int argc,
            char ** argv )
```

5.10.2.2 printElapsedFrom()

```
void printElapsedFrom (
            time_point< steady_clock > & timestamp )
```

5.11 main_test.cpp File Reference

Tests for the genetic sequence aligner.

```
#include <iostream>
#include <string>
#include <array>
#include <fstream>
#include <sstream>
#include "GBReader.h"
#include "Aligner.h"
```

Functions

- bool testMatInit (DirectionMat &mat, DirectionMat &expectedMat)
- bool testFillMat (DirectionMat &mat, const string &s1, const string &s2, int32_t expectedScore, DirectionMat &expectedMat)
- bool testAlignment (DirectionMat &mat, const string &s1, const string &s2, array< string, 3 > &expected↩ Alignment)
- int main ()

5.11.1 Detailed Description

Tests for the genetic sequence aligner.

Authors

DALZOTTO, Rafael; HEIR, Alejandro - Grupo 7

Version

0.1

Date

2022-05-08

Copyright

Copyright (c) 2022 - 22.08 EDA - ITBA

5.11.2 Function Documentation

5.11.2.1 main()

```
int main ( )
```

5.11.2.2 testAlignment()

```
bool testAlignment (
            DirectionMat & mat,
            const string & s1,
            const string & s2,
            array< string, 3 > & expectedAlignment )
```

5.11.2.3 testFillMat()

```
bool testFillMat (
            DirectionMat & mat,
            const string & s1,
            const string & s2,
            int32_t expectedScore,
            DirectionMat & expectedMat )
```

5.11.2.4 testMatInit()

```
bool testMatInit (
            DirectionMat & mat,
            DirectionMat & expectedMat )
```

5.12 README.md File Reference

Index