Paired-End Adapter Finder

Generated By Rmarkdown

Fri January 5 2017

Chapter 1

Paired-End Adapter Finder v16

The Paired-End Adapter Finder identify two consensus adapter sequences from Reads 1 and Reads 2 of paired-end sequencing by using the Needleman-Wunsh algorithm to align the two reads and determining which region of the sequence is actually an adapter. The user has to input two fastq format files, corresponding to Read 1 and Read 2, the input file can be 4-line FASTQ file or multi-line FASTQ file, and the result will be the consensus adapter sequences for both the Adapters in Read 1 and Read 2 respectively of the paired-end sequencing.

- Author
  - Rayan Gan and Farhan Tahir
- Date
  - Jan 2017

Chapter 2

Class Index

2.1. Class List

Chapter 3

Class Documentation

3.1. Input Class Reference

Class transforms the sequences and format line of input file to be used in finding the Adapter Sequencer
#include "Input.h"

- Public Member Functions
  - int complementInput(string&);
  - string reform(string, bool&);

3.1.1. Detailed Description

The Input Class transforms the sequences to be used in the Adapter Sequencer. The Input Class is responsible to reverse complement Read 2 before dynamic programming is done.

- Author
  - Rayan Gan and Farhan Tahir

- Date
  - Jan 2017

3.1.2. Member Function Documentation

3.1.2.1. int Input::complementInput(string& seq)

Reverse complement the input for sequences from the second input file.

Parameters

Parameter	Description
seq	The sequence to be reversed and complemented

3.1.2.2. string Input::reform(string file, bool &fourline)

Reform multi-line FASTQ file into 4-line FASTQ file and return the name of new file of 4-line FASTQ file after reformation process succeed.

Parameters

Parameter	Description
file	Input file (multi-line FASTQ file)
fourline	Value whether the file is 4-line FASTQ file or multi-line FASTQ file

The documentation for this class was generated from the following file:
- export/home/farhan/NetBeansProjects/PEAdapterDev/Input.h
- export/home/farhan/NetBeansProjects/PEAdapterDev/Input.cpp

3.2. NW Class Reference

The NW Class corresponds to the Needleman-Wunsch algorithm which is used to align the two reads. *#include "NW.h"*

- Public Member Functions
  - int nw (string seq_1, string seq_2, string &seq_1_al, string &seq_2_al, int debug)
- Public Attributes
  - int **colmax**
  - int **count**
  - int ** **F**
  - int **Fx**
  - int **Fy**
  - double **percentage**
  - int **rowmax**
  - char ** **traceback**
  - char ** **tracebackscore**
- Private Member Functions
  - void clear()
  - void dpm_init(int ** **F**, char ** **traceback**, int L1, int L2, int d)
  - int nw_align (int ** **F**, char ** **traceback**, string seq_1, string seq_2, string &seq_1_al, string &seq_2_al, int d)
  - void print_al (string &seq_1_al, string &seq_2_al)
  - void print_matrix (int ** **F**, string seq_1, string seq_2)
  - void print_traceback (char ** **traceback**, string seq_1, string seq_2)
  - void verifyPercentage (string seq_1_al, string seq_2_al)

3.2.1 Detailed Description

The NW Class corresponds to the Needleman-Wunsch algorithm which is used to align the two reads. The Needleman-Wunsch algorithm is implemented thorought dynamic programming. Read 1 is aligned against the reverse complement of Read 2 to obtain the best alignment. The nucleotides at the right end of Read 1 and upper-end of Read 2 form the adapter sequences for Adapter 1 and Adapter 2 respectively.

- Author
  - Rayan Gan and Farhan Tahir
- Date
  - Jan 2017

3.2.2 Member Function Documentation

3.2.2.1 void NW::clear ( ) [private]

Clears the dynamic programming and traceback matrices

3.2.2.2 void NW::dpm_init (int ** F, char ** traceback, int L1, int L2, int d) [private]

Initialize the dynamic programming matrix and traceback matrix

Parameters

Parameter	Description
F	The dynamic programming matrix
traceback	The traceback matrix
L1	Length of Read 1
L2	Length of Read 2
d	Gap Penalty

3.2.2.3 int NW::nw (string seq_1, string seq_2, string & seq_1_al, string & seq_2_al, int debug)

Main function in the NW (p. 8) class which calls other functions to perform dynamic programming on Reads 1 and Reads 2

Parameters

Parameter	Description
seq_1	Read 1 in the first FASTQ file
seq_2	Reversed-complemented Read 2 from the second FASTQ file
seq_1_al	Read 1 aligned to Reversed-complemented Read 2
seq_2_al	Reversed-complemented Read 2 aligned to Read 1

3.2.2.4 int NW::nw_align (int F, char traceback, string seq_1, string seq_2, string & seq_1_al, string & seq_2_al, int d) [private]

Runs the Needleman-Wunsch Alignment to get the best alignment

Parameters

Parameter	Description
F	The dynamic programming matrix
traceback	The traceback matrix
seq_1	Read 1 in the first FASTQ file
seq_2	Reversed-complemented Read 2 from the second FASTQ file
seq_1_al	Read 1 aligned to Reversed-complemented Read 2
seq_2_al	Reversed-complemented Read 2 aligned to Read 1
d	Gap Penalty

3.2.2.5 void NW::print_al (string & seq_1_al, string & seq_2_al) [private]

Prints out the aligned sequences

Parameters

Parameter	Description
seq_1_al	Read 1 aligned to Reversed-complemented Read 2
seq_2_al	Reversed-complemented Read 2 aligned to Read 1

3.2.2.6 void NW::print_matrix (int **F, string seq_1, string seq_2) [private]

Prints out the dynamic programming matrix

Parameters

Parameter	Description
F	The dynamic programming matrix
seq_1	Read 1 in the first FASTQ file
seq_2	Reversed-complemented Read 2 from the second FASTQ file

3.2.2.7 void NW::print_traceback (char **traceback, string seq_1, string seq_2) [private]

Prints out traceback matrix

Parameters

Parameter	Description
traceback	The traceback matrix
seq_1_al	Read 1 aligned to Reversed-complemented Read 2
seq_2_al	Reversed-complemented Read 2 aligned to Read 1

3.2.2.8 void NW::verifyPercentage (string seq_1_al, string seq_2_al) [private]

Determines the traceback percentage

Parameters

Parameter	Description
seq_1_al	Read 1 aligned to Reversed-complemented Read 2
seq_2_al	Reversed-complemented Read 2 aligned to Read 1

3.2.3 Member Data Documentation

3.2.3.1 int NW::colmax

Traceback value at first column of dynamic programming.

3.2.3.2 int NW::count

Counts the number of runs of NW.

3.2.3.3 int ** NW::F

Stores the dynamic programming values.

3.2.3.4 int NW::Fx

Keeps track of the length of sequence 1.

3.2.3.5 int NW::Fy

Keeps track of the length of sequence 2.

3.2.3.6 double NW::percentage

Percentage of matches in best traceback.

3.2.3.7 int NW::rowmax

Largest value at last row of dynamic programming.

3.2.3.8 char ** NW::traceback

Stores the traceback characters.

3.2.3.9 char * NW::tracebackscore

Stores the best traceback characters.

The documentation for this class was generated from the following files:
- export/home/farhan/NetBeansProjects/PEAdapterDev/NW.h
- export/home/farhan/NetBeansProjects/PEAdapterDev/NW.cpp

3.3. CS Class Reference

The CS Class corresponds to the Consensus Sequence which is used obtain the adapter sequences from the two reads.

#include "CS.h"

- Public Member Functions
  - void calc_phred ()
  - void checkConfidence (double conf, int &confTrue, int adapLenCount)
  - void cs (string seq_1, int max)
  - void print_cs (int opt)
  - void print_nucCount_phred ()
- Static Private Attributes
  - static const char nuclist [4] = {'A','C','G','T'}

3.3.1 Detailed Description

The CS (p. 5) Class corresponds to the Consensus Sequence which is used obtain the adapter sequences from the two reads. The Consensus Sequence for both Read 1 and Read 2 adapters are determined based on the traceback of the dynamic programming algorithm. The Consensus Sequence is searched based on how well Read 1 and Read 2 align to each other. The ends of Read 1 and Read 2 which do not align form adapters and are added to the nucleotide count. The final Consensus Sequence is based on the nucleotide count.

- Author
  - Rayan Gan and Farhan Tahir

- Date
  - Jan 2017

3.3.2 Member Function Documentation

3.3.2.1 void CS::calc_phred ( )

Calculates the Phred scores of each nucleotide in the nucleotide count

3.3.2.2 void CS::checkConfidence (double conf, int & confTrue, int adapLenCount)

Checks the confidence level of the consensus sequences

Parameters

Parameter	Description
conf	Confidence Level
confTrue	Does the sequence meet the confidence level?

3.3.2.3 void cs(string seq_1, int max, int & c1);

Main function in CS class which performs the algorithm to find the consensus sequences

Parameters

Parameter	Description
seq_1	Read 1 and Complemented Read 2 from FASTQ file
max	Length at which alignment is the best
c1	Length of adapter sequence detected in the read

3.3.2.4 void CS::print_cs (int c, int opt)

Prints out the Consensus Sequence

Parameters

Parameter	Description
c	The highest length of adapter sequence from all adapter sequence detected
opt	Option to choose if Read 1 or Read 2 (0 - Read 1, 1 - Read 2)

3.3.2.5 void CS::print_nucCount_phred ( )

Prints out the nucleotide count and Phred scores

3.3.3 Member Data Documentation

3.3.3.1 int CS::adapterLength [private]

Length of adapter

3.3.3.2 int CS::adapterPos [private]

Position of adapter

3.3.3.3 int CS::Confidence[20] [private]

Confidence of adapter

3.3.3.4 char CS::consensus[20] [private]

Consensus sequence of adapter.

3.3.3.5 int CS::nucleotidecount[4][20] [private]

2D-array of number of nucleotides at each posiiton in the adapter.

3.3.3.6 const char CS::nuclist = {'A','C','G','T'} [static], [private]

List of nucleotides

3.3.3.7 double CS::phred[4][20] [private]

2D-array of Phred score of each nucleotide in the adapter.

The documentation for this class was generated from the following files:
- export/home/farhan/NetBeansProjects/PEAdapterDev/CS.h
- export/home/farhan/NetBeansProjects/PEAdapterDev/CS.cpp