Isomorph

0.2

Generated by Doxygen 1.8.6

Mon Jun 22 2015 19:28:50

Contents

1 Main Page **1**

2 Hierarchical Index **3**

2.1 Class Hierarchy . . . 3

3 Class Index **5**

3.1 Class List . . . 5

4 File Index **7**

4.1 File List . . . 7

5 Class Documentation **9**

5.1 isomorph::CountEstimator::AlgoParams Struct Reference . . . 9

5.1.1 Detailed Description . . . 9

5.1.2 Member Data Documentation . . . 10

5.1.2.1 alignments . . . 10

5.1.2.2 eff_num_reads . . . 10

5.1.2.3 paired_end . . . 10

5.1.2.4 paired_reads . . . 10

5.1.2.5 pairs . . . 10

5.1.2.6 qNameToID . . . 10

5.1.2.7 reads . . . 10

5.1.2.8 single_reads . . . 10

5.1.2.9 transcripts . . . 10

5.2 Args Struct Reference . . . 11

5.3 isomorph::CountEstimator Class Reference . . . 11

5.3.1 Detailed Description . . . 12

5.3.2 Member Function Documentation . . . 12

5.3.2.1 calculate_read_count . . . 12

5.3.2.2 estimate_abundances . . . 12

5.3.2.3 output_result . . . 12

5.3.2.4 preprocess_data . . . 12

5.4 isomorph::CountEstimator::CountResult Struct Reference . . . 13

5.4.1 Detailed Description . . . 13
5.5 isomorph::EMEstimator Class Reference . . . 13
5.5.1 Detailed Description . . . 15
5.5.2 Member Function Documentation . . . 15
5.5.2.1 create_paired_end . . . 15
5.5.2.2 create_single_end . . . 15
5.5.2.3 EMAlgorithm . . . 15
5.5.2.4 estimate_abundances . . . 15
5.5.2.5 output_result . . . 15
5.5.2.6 precalc_posteriors . . . 15
5.5.2.7 preprocess_data . . . 16
5.6 isomorph::EMEstimator::EMParams Struct Reference . . . 16
5.6.1 Detailed Description . . . 17
5.6.2 Constructor & Destructor Documentation . . . 17
5.6.2.1 EMParams . . . 17
5.6.3 Member Data Documentation . . . 17
5.6.3.1 eff_num_reads . . . 17
5.6.3.2 insert_mean . . . 17
5.6.3.3 insert_stdev . . . 17
5.6.3.4 paired_end . . . 17
5.6.3.5 paired_reads . . . 17
5.6.3.6 pairs . . . 18
5.6.3.7 qNameToID . . . 18
5.6.3.8 reads . . . 18
5.6.3.9 single_reads . . . 18
5.6.3.10 transcripts . . . 18
5.7 isomorph::EMEstimator::EMResult Struct Reference . . . 18
5.7.1 Detailed Description . . . 18
5.8 isomorph::Estimator Class Reference . . . 18
5.8.1 Detailed Description . . . 19
5.8.2 Member Function Documentation . . . 19
5.8.2.1 estimate_abundances . . . 19
5.9 isomorph::FastAData Struct Reference . . . 19
5.9.1 Detailed Description . . . 20
5.10 isomorph::FastQData Struct Reference . . . 20
5.10.1 Detailed Description . . . 20
5.11 isomorph::PairedRead Struct Reference . . . 20
5.11.1 Detailed Description . . . 21
5.11.2 Member Data Documentation . . . 21
5.11.2.1 id . . . 21

5.11.2.2 left_fa_id . . . 21
5.11.2.3 left_phred . . . 21
5.11.2.4 left_seq . . . 21
5.11.2.5 pi_x_n . . . 22
5.11.2.6 right_fa_id . . . 22
5.11.2.7 right_phred . . . 22
5.11.2.8 right_seq . . . 22
5.12 isomorph::Read Struct Reference . . . 22
5.12.1 Detailed Description . . . 22
5.13 isomorph::Reader Class Reference . . . 22
5.13.1 Detailed Description . . . 23
5.13.2 Member Function Documentation . . . 23
5.13.2.1 read_fasta . . . 23
5.13.2.2 read_fastq . . . 23
5.13.2.3 read_sam . . . 24
5.14 isomorph::SamData Struct Reference . . . 24
5.14.1 Detailed Description . . . 25
5.15 isomorph::SingleRead Struct Reference . . . 25
5.15.1 Detailed Description . . . 26
5.15.2 Member Data Documentation . . . 26
5.15.2.1 fa_id . . . 26
5.15.2.2 id . . . 26
5.15.2.3 phred . . . 26
5.15.2.4 pi_x_n . . . 26
5.15.2.5 seq . . . 26

6 File Documentation **27**

6.1 src/count_estimator.h File Reference . . . 27
6.1.1 Detailed Description . . . 28
6.2 src/em_estimator.cpp File Reference . . . 28
6.2.1 Detailed Description . . . 29
6.3 src/em_estimator.h File Reference . . . 29
6.3.1 Detailed Description . . . 30
6.4 src/estimator.h File Reference . . . 30
6.4.1 Detailed Description . . . 31
6.5 src/main.cpp File Reference . . . 31
6.5.1 Detailed Description . . . 32
6.6 src/paired_read.h File Reference . . . 32
6.6.1 Detailed Description . . . 33
6.7 src/read.h File Reference . . . 34

6.7.1 Detailed Description . . . 34
6.8 src/single_read.h File Reference . . . 35
6.8.1 Detailed Description . . . 35
6.9 src/utility.cpp File Reference . . . 36
6.9.1 Detailed Description . . . 36
6.10 src/utility.h File Reference . . . 37
6.10.1 Detailed Description . . . 38

Chapter 1

Main Page

Isomorph is a software for RNA transcript abundance estimation. It works with de novo assembled transcripts and supports different modes of operation.

At its core, it contains an EM algorithm that tries to optimize and find parameters of a robust statistical model proposed in the papers

"RNA-Seq gene expression estimation with read mapping uncertainty"

http://bioinformatics.oxfordjournals.org/content/26/4/493.fullm

and "RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome"

http://www.biomedcentral.com/1471-2105/12/323.

Isomorph, besides the above described statistical model,

also supports simple read count models described in the associated master thesis.

It requires seqan (http://www.seqan.de/), an open source C++ library for more efficient handling

of input and output data.

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

isomorph::CountEstimator::AlgoParams 9
Args 11
isomorph::CountEstimator::CountResult 13
isomorph::EMEstimator::EMParams 16
isomorph::EMEstimator::EMResult 18
isomorph::Estimator 18
    isomorph::CountEstimator 11
    isomorph::EMEstimator 13
isomorph::FastAData 19
isomorph::FastQData 20
isomorph::Read 22
    isomorph::PairedRead 20
    isomorph::SingleRead 25
isomorph::Reader 22
isomorph::SamData 24

Chapter 3

Class Index

3.1 Class List

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

isomorph::CountEstimator::AlgoParams 9
Args 11
isomorph::CountEstimator 11
isomorph::CountEstimator::CountResult 13
isomorph::EMEstimator 13
isomorph::EMEstimator::EMParams 16
isomorph::EMEstimator::EMResult 18
isomorph::Estimator 18
isomorph::FastAData 19
isomorph::FastQData 20
isomorph::PairedRead 20
isomorph::Read 22
isomorph::Reader 22
isomorph::SamData 24
isomorph::SingleRead 25

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

src/count_estimator.h
: Declares the count estimator class 27

src/em_estimator.cpp
: CountEstimator class methods are implemented here 28

src/em_estimator.h
: Declares the em estimator class 29

inherited by other estimator classes 30

src/main.cpp
: Parses input parameters and runs the approriate estimator 31

src/paired_read.h
: Declares paired_end read header 32

src/read.h
: Declares a generic read header 34

src/single_read.h
: Declares single_end read header 35

src/utility.cpp
: Utility data and functions used by other classes are implemented here 36

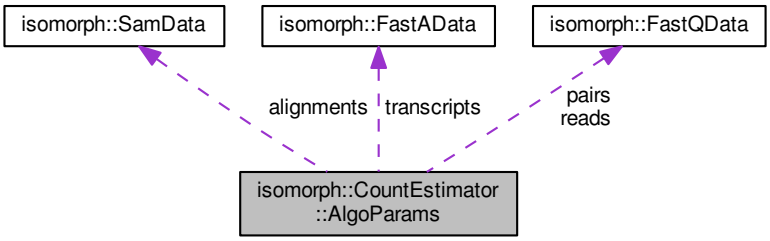
src/utility.h
: Utility data and functions used by other classes are declared here 37

Chapter 5

Class Documentation

5.1 isomorph::CountEstimator::AlgoParams Struct Reference

Collaboration diagram for isomorph::CountEstimator::AlgoParams:



Public Attributes

- std::unordered_map
  < std::string, int > qNameToID
- FastQData reads
- FastQData pairs
- FastAData transcripts
- SamData alignments
- std::vector< std::unique_ptr
  < SingleRead > > single_reads
- std::vector< std::unique_ptr
  < PairedRead > > paired_reads
- int eff_num_reads
- bool paired_end

5.1.1 Detailed Description

The struct encapsulates all the parameters used by count algorithm and passed to it from the outside.

5.1.2 Member Data Documentation

5.1.2.1 SamData isomorph::CountEstimator::AlgoParams::alignments

The alignments of the reads to reconstructed transcripts.

5.1.2.2 int isomorph::CountEstimator::AlgoParams::eff_num_reads

This is the number of reads that have a mapping and are not ignored.
This will be gone in future versions when single_reads and paired_reads
will be containing only information for valid reads.

5.1.2.3 bool isomorph::CountEstimator::AlgoParams::paired_end

If paired-end is used, this is set to true.

5.1.2.4 std::vector<std::unique_ptr<PairedRead> > isomorph::CountEstimator::AlgoParams::paired_reads

If paired reads are used, this is where their information regarding
mappings will be stored for EM processing.

5.1.2.5 FastQData isomorph::CountEstimator::AlgoParams::pairs

If paired-end is used, reads coming from the second file are stored here.

5.1.2.6 std::unordered_map<std::string, int> isomorph::CountEstimator::AlgoParams::qNameToID

Serves to map read names collected from fastq files to corresponding integer indices of reads.

5.1.2.7 FastQData isomorph::CountEstimator::AlgoParams::reads

Reads from file. Encapsulates seqan data structures.

5.1.2.8 std::vector<std::unique_ptr<SingleRead> > isomorph::CountEstimator::AlgoParams::single_reads

If single reads are used, this is where their information regarding
mappings will be stored for EM processing.

5.1.2.9 FastAData isomorph::CountEstimator::AlgoParams::transcripts

Stores transcript information.
The documentation for this struct was generated from the following file:

- src/count_estimator.h

5.2 Args Struct Reference

Public Attributes

- bool **paired_end**
- CharString **reads**
- CharString **pairs**
- CharString **transcripts**
- bool **use_count**

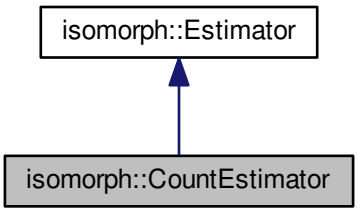
The documentation for this struct was generated from the following file:

- src/main.cpp

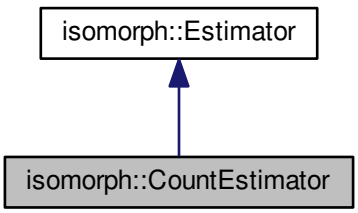
5.3 isomorph::CountEstimator Class Reference

`#include <count_estimator.h>`

Inheritance diagram for isomorph::CountEstimator:



Collaboration diagram for isomorph::CountEstimator:



Classes

- struct AlgoParams
- struct CountResult

Public Member Functions

- virtual void estimate_abundances (seqan::CharString reads, seqan::CharString transcripts, seqan::CharString pairs="")

Private Member Functions

- void preprocess_data (const seqan::CharString &transcripts, const seqan::CharString &reads, const seqan::CharString &pairs, const std::string &output_dir, AlgoParams &params)
- void calculate_read_count (const AlgoParams &params, CountResult &result)
- void output_result (const FastAData &transcripts, const CountResult &result, const std::string &output_file)

5.3.1 Detailed Description

CountEstimator class.

This class estimates transcript abundances using very simple

ideas of counting number of reads mapped to a transcript as well as

the number of bases mapped to a certain isoform and reports the result.

5.3.2 Member Function Documentation

5.3.2.1 void isomorph::CountEstimator::calculate_read_count (const AlgoParams & *params,* CountResult & *result*) [private]

Calculates the count of reads mapped to each transcript.

For multi mapping reads, only the first mapping is considered valid.

5.3.2.2 void isomorph::CountEstimator::estimate_abundances (seqan::CharString *reads,* seqan::CharString *transcripts,* seqan::CharString *pairs* = " ") [virtual]

Implementation of estimate_abundances virtual method.

The method estimates transcript abundances according to the strategy of this class.

Parameters

reads	path to the reads fastq file
transcripts	path to the reconstructed transcripts fasta file
pairs	path to the paired end file, if paired reads are used

Implements isomorph::Estimator.

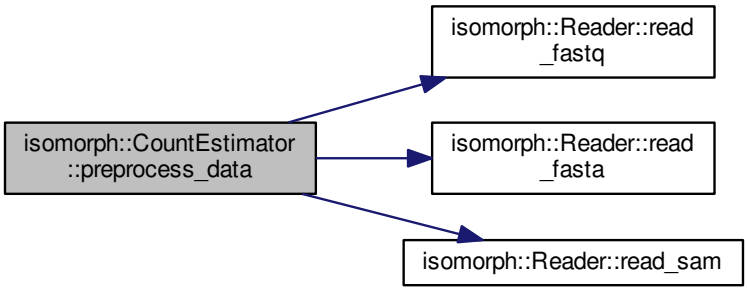
5.3.2.3 void isomorph::CountEstimator::output_result (const FastAData & *transcripts,* const CountResult & *result,* const std::string & *output_file*) [private]

Outputs the result to a file called isomorph.abundances.fasta.

5.3.2.4 void isomorph::CountEstimator::preprocess_data (const seqan::CharString & *transcripts,* const seqan::CharString & *reads,* const seqan::CharString & *pairs,* const std::string & *output_dir,* AlgoParams & *params*) [private]

Does rudimentary data preprocessing.

Here is the call graph for this function:



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

- src/count_estimator.h
- src/count_estimator.cpp

5.4 isomorph::CountEstimator::CountResult Struct Reference

Public Attributes

- std::vector< double > **counts**

5.4.1 Detailed Description

The resulting τ values are stored here and forwarded to postprocessing (if done) and output.

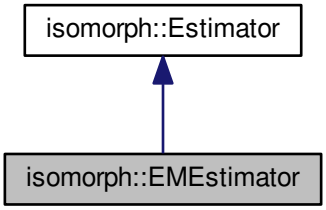
The documentation for this struct was generated from the following file:

- src/count_estimator.h

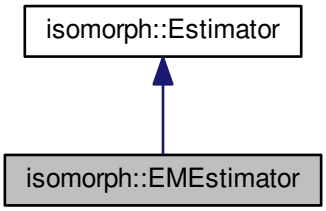
5.5 isomorph::EMEstimator Class Reference

```
#include <em_estimator.h>
```

Inheritance diagram for isomorph::EMEstimator:



Collaboration diagram for isomorph::EMEstimator:



Classes

- struct EMParams
- struct EMResult

Public Member Functions

- virtual void estimate_abundances (seqan::CharString left_pairs, seqan::CharString right_pairs, seqan::CharString transcripts)

Private Member Functions

- void preprocess_data (const seqan::CharString &transcripts, const seqan::CharString &reads, const seqan::CharString &pairs, const std::string &output_dir, EMParams &params)
- void EMAlgorithm (EMParams &params, EMResult &result)
- void precalc_posteriors (const EMParams &params, std::vector< std::vector< double > > &posteriors)
- void create_paired_end (const SamData &alignments, EMParams &params)
- void create_single_end (const SamData &alignments, EMParams &params)
- void output_result (const FastAData &transcripts, const EMResult &result, const std::string filename)

5.5.1 Detailed Description

EMEstimator class.

This class estimates transcript abundances using a robust statistical model

by applying the EM algorithm to obtain maximum likelihood estimates

of the model parameters of the model.

5.5.2 Member Function Documentation

5.5.2.1 void isomorph::EMEstimator::create_paired_end (const SamData & *alignments*, EMParams & *params*) [private]

Parses the SAM mapping file and creates information for paired reads and their mappings.

5.5.2.2 void isomorph::EMEstimator::create_single_end (const SamData & *alignments*, EMParams & *params*) [private]

Parses the SAM mapping file and creates information for single reads and their mappings.

5.5.2.3 void isomorph::EMEstimator::EMAlgorithm (EMParams & *params*, EMResult & *result*) [private]

The core EM algorithm is done here.

5.5.2.4 void isomorph::EMEstimator::estimate_abundances (seqan::CharString *left_pairs*, seqan::CharString *right_pairs*, seqan::CharString *transcripts*) [virtual]

Implementation of estimate_abundances virtual method.

The method estimates transcript abundances according to the strategy of this class.

Parameters

reads	path to the reads fastq file
transcripts	path to the reconstructed transcripts fasta file
pairs	path to the paired end file, if paired reads are used

Implements isomorph::Estimator.

5.5.2.5 void isomorph::EMEstimator::output_result (const FastAData & *transcripts*, const EMResult & *result*, const std::string *filename*) [private]

Outputs the EM algorithm result to a file called isomorph.abundances.fasta.

Output format is:

>transcript_name

τ TPM_value

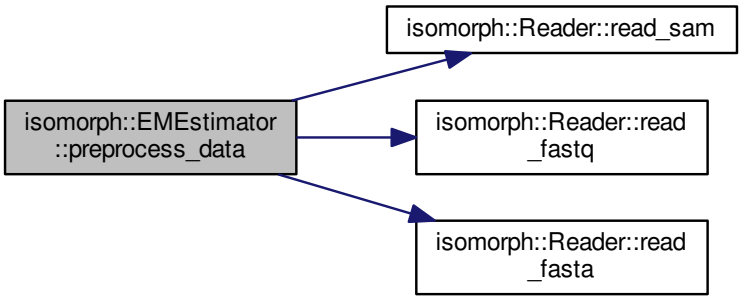
5.5.2.6 void isomorph::EMEstimator::precalc_posteriors (const EMParams & *params*, std::vector< std::vector< double > > & *posteriors*) [private]

The method precalculates some probabilities that don't need to be recalculated anymore.

5.5.2.7 void isomorph::EMEstimator::preprocess_data (const seqan::CharString & *transcripts*, const seqan::CharString & *reads*, const seqan::CharString & *pairs*, const std::string & *output_dir*, **EMParams** & *params*) [private]

Method does preprocesing of data, elimination of extra stuff and creates necessary data structures for the EM to operate on. Most of the ideas are described in the original RSEM paper as well as in the follow-up.

Here is the call graph for this function:

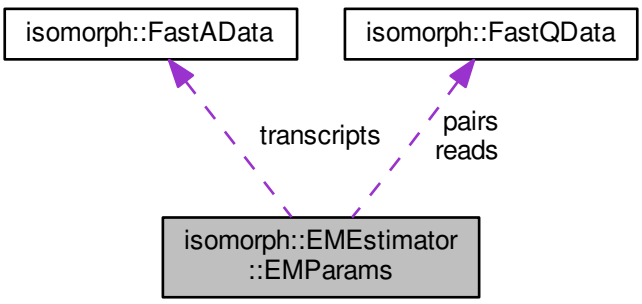


The documentation for this class was generated from the following files:

- src/em_estimator.h
- src/em_estimator.cpp

5.6 isomorph::EMEstimator::EMParams Struct Reference

Collaboration diagram for isomorph::EMEstimator::EMParams:



Public Member Functions

- EMParams ()

Public Attributes

- std::unordered_map < std::string, int > qNameToID
- FastQData reads
- FastQData pairs
- FastAData transcripts
- std::vector< std::unique_ptr < SingleRead > > single_reads
- std::vector< std::unique_ptr < PairedRead > > paired_reads
- int eff_num_reads
- bool paired_end
- double insert_mean
- double insert_stdev

5.6.1 Detailed Description

The struct encapsulates all the parameters used by EM algorithm and passed to it from the outside.

5.6.2 Constructor & Destructor Documentation

5.6.2.1 isomorph::EMEstimator::EMParams::EMParams ( ) [inline]

The constructor that sets some of the information to invalid values.

5.6.3 Member Data Documentation

5.6.3.1 int isomorph::EMEstimator::EMParams::eff_num_reads

This is the number of reads that have a mapping and are not ignored.

This will be gone in future versions when single_reads and paired_reads

will be containing only information for valid reads.

5.6.3.2 double isomorph::EMEstimator::EMParams::insert_mean

When paired-end is used, estimate of the mean of the insert sizes is stored here.

5.6.3.3 double isomorph::EMEstimator::EMParams::insert_stdev

When paired-end is used, estimate of the standard deviation of the insert sizes is stored here.

5.6.3.4 bool isomorph::EMEstimator::EMParams::paired_end

If paired-end is used, this is set to true.

5.6.3.5 std::vector<std::unique_ptr<PairedRead> > isomorph::EMEstimator::EMParams::paired_reads

If paired reads are used, this is where their information regarding

mappings will be stored for EM processing.

5.6.3.6 FastQData isomorph::EMEstimator::EMParams::pairs

If paired-end is used, reads coming from the second file are stored here.

5.6.3.7 std::unordered_map<std::string, int> isomorph::EMEstimator::EMParams::qNameToID

Serves to map read names collected from fastq files to corresponding integer indices of reads.

5.6.3.8 FastQData isomorph::EMEstimator::EMParams::reads

Reads from file. Encapsulates seqan data structures.

5.6.3.9 std::vector<std::unique_ptr<SingleRead> > isomorph::EMEstimator::EMParams::single_reads

If single reads are used, this is where their information regarding
mappings will be stored for EM processing.

5.6.3.10 FastAData isomorph::EMEstimator::EMParams::transcripts

Stores transcript information.
The documentation for this struct was generated from the following file:

- src/em_estimator.h

5.7 isomorph::EMEstimator::EMResult Struct Reference

Public Attributes

- std::vector< double > **relative_expressions**

5.7.1 Detailed Description

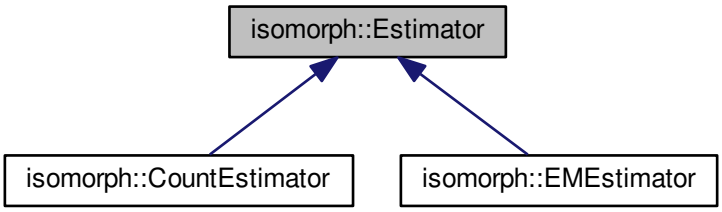
The resulting τ values are stored here and forwarded to postprocessing (if done) and output.
The documentation for this struct was generated from the following file:

- src/em_estimator.h

5.8 isomorph::Estimator Class Reference

```
#include <estimator.h>
```

Inheritance diagram for isomorph::Estimator:



Public Member Functions

- virtual void estimate_abundances (seqan::CharString reads, seqan::CharString transcripts, seqan::CharString pairs="")=0

5.8.1 Detailed Description

Estimator class.

Abstract base class all estimator classes must inherit.

5.8.2 Member Function Documentation

5.8.2.1 virtual void isomorph::Estimator::estimate_abundances (seqan::CharString *reads*, seqan::CharString *transcripts*, seqan::CharString *pairs* = " ") [pure virtual]

This method must be implemented in the classes inheriting this one.

Parameters

reads	path to the reads fastq file
transcripts	path to the transcripts fasta file
pairs	path to the paired end fastq file, if used

Implemented in isomorph::CountEstimator, and isomorph::EMEstimator.

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

- src/estimator.h

5.9 isomorph::FastAData Struct Reference

```
#include <utility.h>
```

Public Attributes

- seqan::StringSet
  < seqan::CharString > **ids**

- seqan::StringSet
  < seqan::Dna5String > **seqs**

5.9.1 Detailed Description

FastAData struct. Encapsulates seqan fasta file data classes for easier handling.

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

- src/utility.h

5.10 isomorph::FastQData Struct Reference

`#include <utility.h>`

Public Attributes

- seqan::StringSet
  < seqan::CharString > **ids**
- seqan::StringSet
  < seqan::Dna5String > **seqs**
- seqan::StringSet
  < seqan::CharString > **phred**

5.10.1 Detailed Description

FastQData struct. Encapsulates seqan fastq file data classes for easier handling.

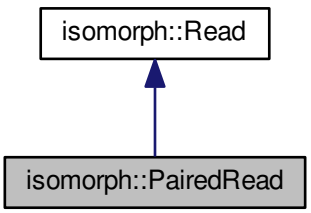
The documentation for this struct was generated from the following file:

- src/utility.h

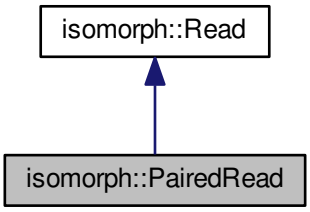
5.11 isomorph::PairedRead Struct Reference

`#include <paired_read.h>`

Inheritance diagram for isomorph::PairedRead:



Collaboration diagram for isomorph::PairedRead:



Public Attributes

- int id
- seqan::CharString left_fa_id
- seqan::Dna5String left_seq
- seqan::CharString left_phred
- seqan::CharString right_fa_id
- seqan::Dna5String right_seq
- seqan::CharString right_phred
- std::vector< std::tuple< int, int, int, int, seqan::CharString, seqan::CharString > > pi_x_n

5.11.1 Detailed Description

Paired read class.

All the data necessary for paired read analysis is placed here.

5.11.2 Member Data Documentation

5.11.2.1 int isomorph::PairedRead::id

Read id given to it by isomorph for easier indexing.

5.11.2.2 seqan::CharString isomorph::PairedRead::left_fa_id

The full read ID string from first fastq file.

5.11.2.3 seqan::CharString isomorph::PairedRead::left_phred

Corresponding phred quality symbols..

5.11.2.4 seqan::Dna5String isomorph::PairedRead::left_seq

Sequence of mate from the first fastq file.

5.11.2.5 std::vector<std::tuple<int, int, int, int, seqan::CharString, seqan::CharString> > isomorph::PairedRead::pi_x_n

The reduced space of transcripts this read maps well to

together with mapped read sequence from SAM file.

5.11.2.6 seqan::CharString isomorph::PairedRead::right_fa_id

The full ID string of the second mate.

5.11.2.7 seqan::CharString isomorph::PairedRead::right_phred

Corresponding phred quality symbols.

5.11.2.8 seqan::Dna5String isomorph::PairedRead::right_seq

Sequence of the second mate.

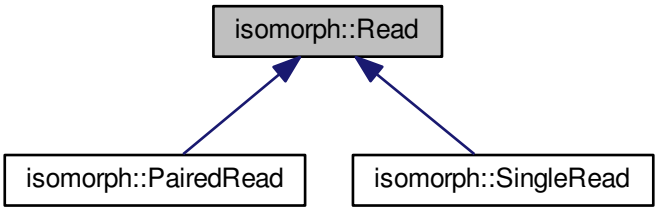
The documentation for this struct was generated from the following file:

- src/paired_read.h

5.12 isomorph::Read Struct Reference

`#include <read.h>`

Inheritance diagram for isomorph::Read:



5.12.1 Detailed Description

A generic read class.

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

- src/read.h

5.13 isomorph::Reader Class Reference

`#include <utility.h>`

Public Member Functions

- int read_sam (seqan::CharString filename, SamData *data)
- int read_fasta (seqan::CharString filename, FastAData *data)
- int read_fastq (seqan::CharString filename, FastQData *data)

5.13.1 Detailed Description

Reader class that encapsulates methods for data input and output.

5.13.2 Member Function Documentation

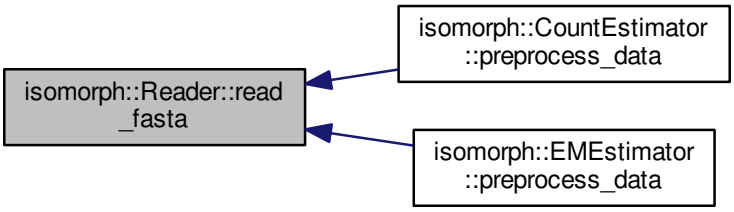
5.13.2.1 int isomorph::Reader::read_fasta (seqan::CharString *filename,* FastAData * *data*)

Reads a fasta file.

Parameters

filename	path to the file
data	pointer to fastadata struct that will be filled.

Here is the caller graph for this function:



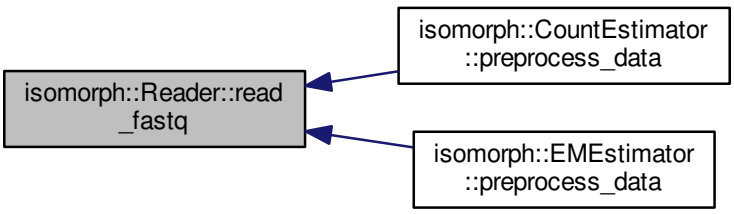
5.13.2.2 int isomorph::Reader::read_fastq (seqan::CharString *filename,* FastQData * *data*)

Reads a fastq file.

Parameters

filename	path to the file
data	pointer to fastqadat struct that will be filled.

Here is the caller graph for this function:



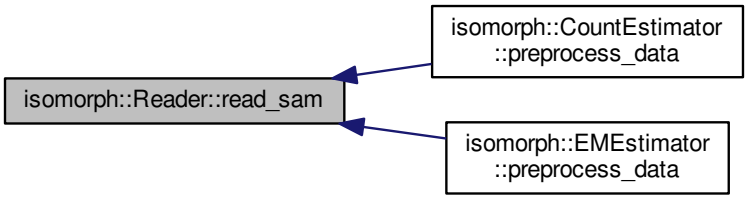
5.13.2.3 int isomorph::Reader::read_sam (seqan::CharString *filename*, SamData * *data*)

Reads SAM alignment file.

Parameters

filename	path to the file
data	Pointer to samdata struct that will be filled.

Here is the caller graph for this function:



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

- src/utility.h
- src/utility.cpp

5.14 isomorph::SamData Struct Reference

`#include <utility.h>`

Public Attributes

- seqan::BamHeader **header**
- std::vector
  < seqan::BamAlignmentRecord > **records**

5.14.1 Detailed Description

SamData struct. Encapsulates seqan fsam file data classes for easier handling.

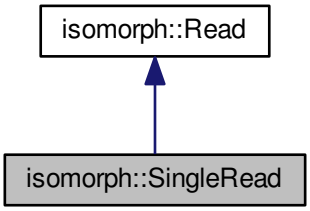
The documentation for this struct was generated from the following file:

- src/utility.h

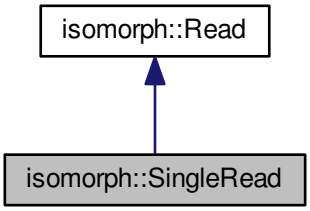
5.15 isomorph::SingleRead Struct Reference

`#include <single_read.h>`

Inheritance diagram for isomorph::SingleRead:



Collaboration diagram for isomorph::SingleRead:



Public Attributes

- int id
- seqan::CharString fa_id
- seqan::Dna5String seq
- seqan::CharString phred
- std::vector< std::tuple< int, int > > pi_x_n

5.15.1 Detailed Description

A single-end read class.

All the data necessary for single-read analysis is placed here.

5.15.2 Member Data Documentation

5.15.2.1 seqan::CharString isomorph::SingleRead::fa_id

Full read ID from the fastq file.

5.15.2.2 int isomorph::SingleRead::id

Read id given to it by isomorph for easier indexing.

5.15.2.3 seqan::CharString isomorph::SingleRead::phred

Associated phred quality symbols.

5.15.2.4 std::vector<std::tuple<int, int> > isomorph::SingleRead::pi_x_n

The reduced space of the transcripts this read maps well to.

5.15.2.5 seqan::Dna5String isomorph::SingleRead::seq

Read sequence from the fastq file.

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

- src/single_read.h

Chapter 6

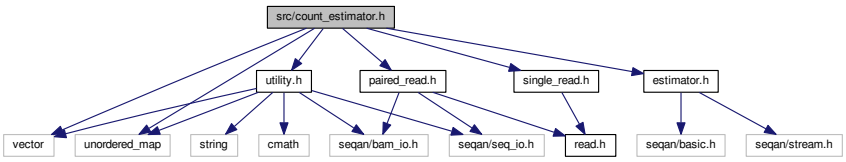
File Documentation

6.1 src/count_estimator.h File Reference

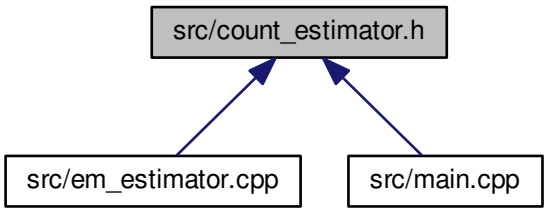
Declares the count estimator class.

```
#include <vector>
#include <unordered_map>
#include "utility.h"
#include "estimator.h"
#include "single_read.h"
#include "paired_read.h"
```

Include dependency graph for count_estimator.h:



This graph shows which files directly or indirectly include this file:



Classes

- class isomorph::CountEstimator
- struct isomorph::CountEstimator::AlgoParams
- struct isomorph::CountEstimator::CountResult

6.1.1 Detailed Description

Declares the count estimator class.

Author

Pavlovic:Dario

Version

Revision 0.2

The count estimator header class is declared here.

Date

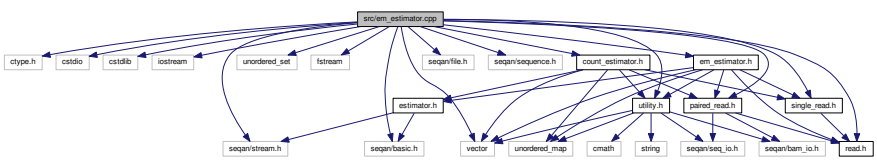
Tuesday, June 16, 2015

6.2 src/em_estimator.cpp File Reference

CountEstimator class methods are implemented here.

```
#include <ctype.h>
#include <cstdio>
#include <cstdlib>
#include <iostream>
#include <vector>
#include <unordered_set>
#include <fstream>
#include <seqan/stream.h>
#include <seqan/basic.h>
#include <seqan/file.h>
#include <seqan/sequence.h>
#include "em_estimator.h"
#include "utility.h"
#include "count_estimator.h"
#include "read.h"
#include "single_read.h"
#include "paired_read.h"
```

Include dependency graph for em_estimator.cpp:



6.2.1 Detailed Description

CountEstimator class methods are implemented here. EMEstimator class methods are implemented here.

Author

Pavlovic:Dario

Version

Revision 0.2

Date

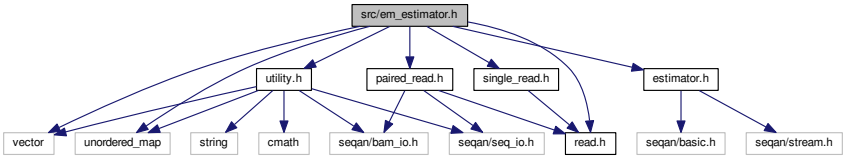
Tuesday, June 16, 2015

6.3 src/em_estimator.h File Reference

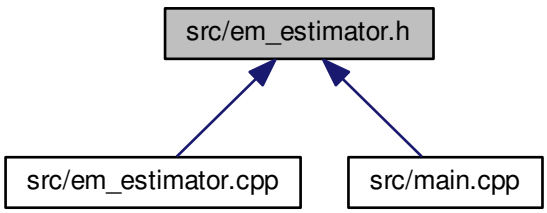
Declares the em estimator class.

```
#include <vector>
#include <unordered_map>
#include "utility.h"
#include "estimator.h"
#include "read.h"
#include "single_read.h"
#include "paired_read.h"
```

Include dependency graph for em_estimator.h:



This graph shows which files directly or indirectly include this file:



Classes

- class isomorph::EMEstimator
- struct isomorph::EMEstimator::EMParams
- struct isomorph::EMEstimator::EMResult

6.3.1 Detailed Description

Declares the em estimator class.

Author

Pavlovic:Dario

Version

Revision 0.2

The em estimator header class is declared here. This is the core header of isomorph.

Date

Tuesday, June 16, 2015

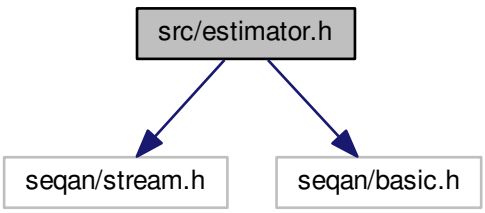
6.4 src/estimator.h File Reference

Declares a generic abstract estimator base class

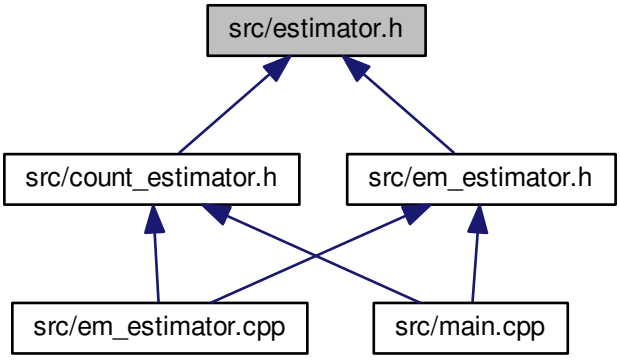
inherited by other estimator classes.

```
#include <seqan/stream.h>
#include <seqan/basic.h>
```

Include dependency graph for estimator.h:



This graph shows which files directly or indirectly include this file:



Classes

- class isomorph::Estimator

6.4.1 Detailed Description

Declares a generic abstract estimator base class
inherited by other estimator classes.

Author

Pavlovic:Dario

Version

Revision 0.2

This is the abstract base class header that other estimators
must include and inherit to keep everything according to the strategy pattern.

Date

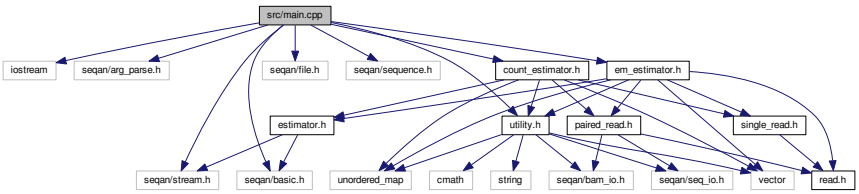
Tuesday, June 16, 2015

6.5 src/main.cpp File Reference

Parses input parameters and runs the approriate estimator.

```
#include <iostream>
#include <seqan/arg_parse.h>
#include <seqan/stream.h>
#include <seqan/basic.h>
#include <seqan/file.h>
#include <seqan/sequence.h>
#include "count_estimator.h"
#include "utility.h"
#include "em_estimator.h"
```

Include dependency graph for main.cpp:



Classes

- struct Args

Functions

- int **parse_arguments** (int argc, char **argv, Args &args)
- int **main** (int argc, char **argv)

6.5.1 Detailed Description

Parses input parameters and runs the approriate estimator.

Author

Pavlovic:Dario

Version

Revision 0.2

This file starts the program, parses input flags by using seqan's arg_parse module and runs the selected estimator on the input.

Date

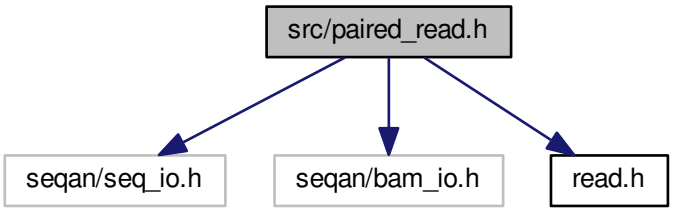
Tuesday, June 16, 2015

6.6 src/paired_read.h File Reference

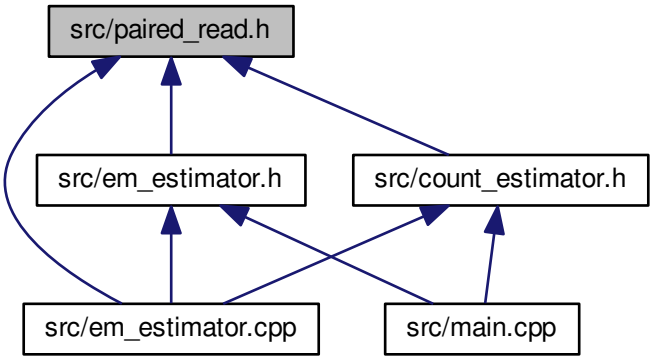
Declares paired_end read header.

```
#include <seqan/seq_io.h>
#include <seqan/bam_io.h>
#include "read.h"
```

Include dependency graph for paired_read.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct isomorph::PairedRead

6.6.1 Detailed Description

Declares paired_end read header.

Author

Pavlovic:Dario

Version

Revision 0.2

Class modeling paired-end reads is declared here.

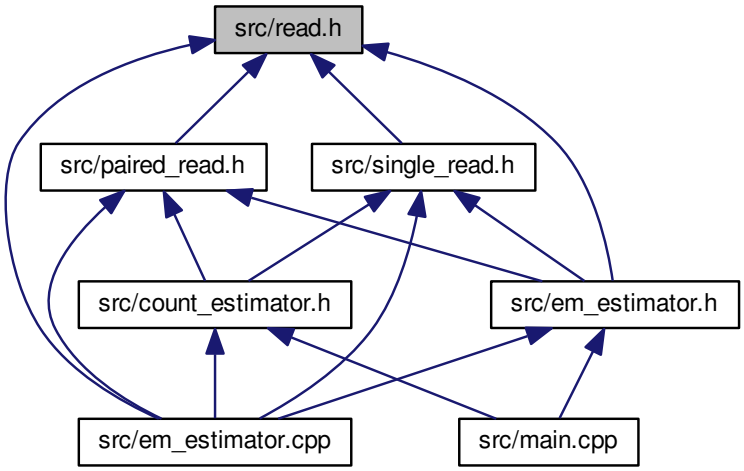
Date

Tuesday, June 16, 2015

6.7 src/read.h File Reference

Declares a generic read header.

This graph shows which files directly or indirectly include this file:



Classes

- struct isomorph::Read

6.7.1 Detailed Description

Declares a generic read header.

Author

Pavlovic:Dario

Version

Revision 0.2

A generic class modeling a read is declared here.

Date

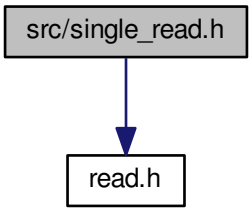
Tuesday, June 16, 2015

6.8 src/single_read.h File Reference

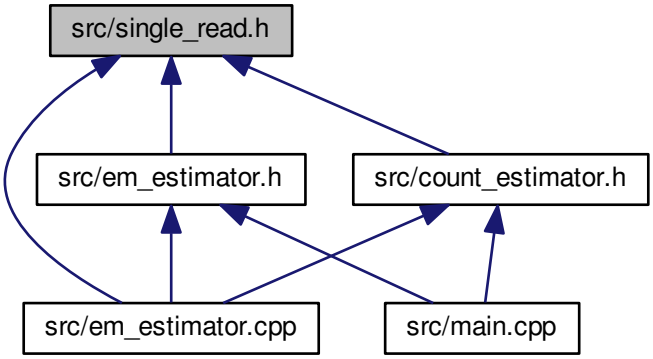
Declares single_end read header.

```
#include "read.h"
```

Include dependency graph for single_read.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct isomorph::SingleRead

6.8.1 Detailed Description

Declares single_end read header.

Author

Pavlovic:Dario

Version

Revision 0.2

Class modeling single-end reads is declared here.

Date

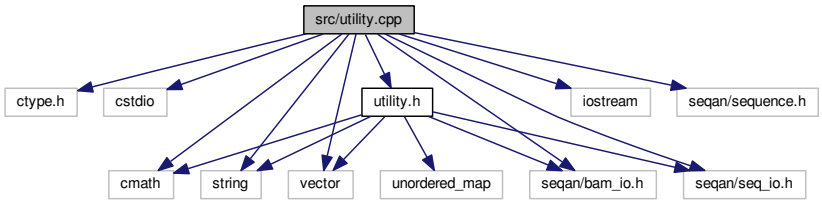
Tuesday, June 16, 2015

6.9 src/utility.cpp File Reference

Utility data and functions used by other classes are implemented here.

```
#include <ctype.h>
#include <cstdio>
#include <cmath>
#include <iostream>
#include <string>
#include <vector>
#include <seqan/bam_io.h>
#include <seqan/seq_io.h>
#include <seqan/sequence.h>
#include "utility.h"
```

Include dependency graph for utility.cpp:



6.9.1 Detailed Description

Utility data and functions used by other classes are implemented here.

Author

Pavlovic:Dario

Version

Revision 0.2

Date

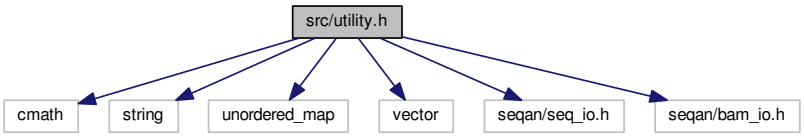
Tuesday, June 16, 2015

6.10 src/utility.h File Reference

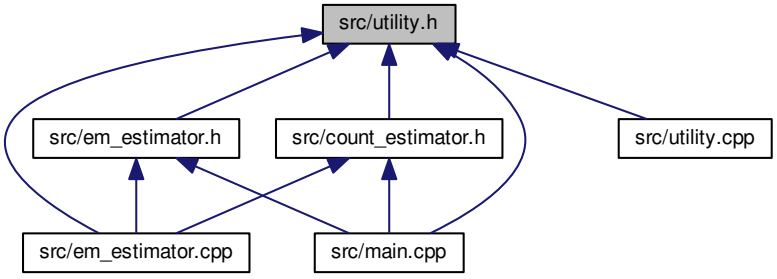
Utility data and functions used by other classes are declared here.

```
#include <cmath>
#include <string>
#include <unordered_map>
#include <vector>
#include <seqan/seq_io.h>
#include <seqan/bam_io.h>
```

Include dependency graph for utility.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct isomorph::FastQData
- struct isomorph::FastAData
- struct isomorph::SamData
- class isomorph::Reader

Functions

- std::string **isomorph::execute_command** (const char *cmd)
- double **isomorph::prob_normal** (double mean, double stdev, double x)
- void **isomorph::estimate_insert_size** (const SamData &alignments, std::pair< double, double > &params)
- void **isomorph::run_alignment** (const std::string &reads, const std::string &pairs, const std::string &transcripts, const std::string &output_dir, const bool paired_end)
- void **isomorph::print_sam_alignment_records** (const std::vector< seqan::BamAlignmentRecord > &records)
- char **isomorph::reverse_complement** (const char &c)

Variables

- const double **PI** = 3.14159265358979

6.10.1 Detailed Description

Utility data and functions used by other classes are declared here.

Author

Pavlovic:Dario

Version

Revision 0.2

Date

Tuesday, June 16, 2015