RIThM

Generated by Doxygen 1.8.11

Contents

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

Class Documentation

2.1 DATA_PREPROCESSING Class Reference

Related Functions

(Note that these are not member functions.)

- MatrixXd csv2MatrixXd (const std::string &path)

  Read a csv file and convert to an instance of Eigen MatrixXd.
- std::vector< int > csv2Vec (const std::string &path, int lineNum)

  Read a csv file and convert to an instance of Eigen MatrixXd.
- MatrixXd scaleMatrix (const MatrixXd &m, int t1, int t2)

  Compute scale matrix (covariance matrix) for multivariate time series.

2.1.1 Friends And Related Function Documentation

2.1.1.1 MatrixXd csv2MatrixXd (const std::string & *path*) [related]

Read a csv file and convert to an instance of Eigen MatrixXd.

Read a csv file and convert to an instance of Eigen MatrixXd (RowMajor).

Parameters

in	*path*	Path to csv file

Returns

an instance of MatrixXd containing the date from the csv file

2.1.1.2 std::vector< int > csv2Vec (const std::string & *path*, int *lineNum*) [related]

Read a csv file and convert to an instance of Eigen MatrixXd.

Read a specified line of a csv file and convert to an std::vector<int>.

Parameters

in	*path*	Path to csv file
in	*lineNum*	Number of the line to be read in.

Returns

an instance of MatrixXd containing the date from the csv file

2.1.1.3 MatrixXd scaleMatrix (const MatrixXd & *m,* int *t1,* int *t2*) [related]

Compute scale matrix (covariance matrix) for multivariate time series.

Compute scale matrix (covariance matrix) for a time interval [t1,t2) of a multivariate time series.

Parameters

in	*m*	Multivariate time series. Cols are variables, rows are time points.
in	*t1*	Beginning of time interval (This point is included in interval)
in	*t2*	End of time interval (This point in NOT included in interval)

Returns

an instance of MatrixXd containing the date from the csv file

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

- /home/schwarze/Desktop/RIThM.h

2.2 DISPLAY_SEQUENCE Class Reference

Functor for printing permutations or combinations (Credit: Howard Hinnant).

`#include <RIThM.h>`

Public Member Functions

- **DISPLAY_SEQUENCE** (unsigned l)
- template<class It >
  bool **operator()** (It first, It last)
- **operator std::uint64_t** () const

2.2.1 Detailed Description

Functor for printing permutations or combinations (Credit: Howard Hinnant).

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

- /home/schwarze/Desktop/RIThM.h

2.3 ENTROPY_CALCULATION Class Reference

Related Functions

(Note that these are not member functions.)

- template<typename Derived >
  double logDet (const MatrixBase< Derived > &m)

  Compute log-determinant of a positive-definite square matrix.
- template<typename Derived >
  double stableLogDet (const MatrixBase< Derived > &m)

  Compute log-determinant of a positive-definite square matrix.
- double CauchySummand (int n)

  Compute a size-dependent summand for entropy of a system with multivariate Cauchy distribution.
- double GaussSummand (int n)

  Compute a size-dependent summand for entropy of a system with multivariate Gauss distribution.
- template<typename Derived >
  double entropy (const MatrixBase< Derived > &m, double sizeSummand)

  Compute entropy for multivariate system from scale matrix and size summand.
- template<typename Derived >
  double stableEntropy (const MatrixBase< Derived > &m, double sizeSummand)

  Compute entropy for multivariate system from scale matrix and size summand.
- template<typename Derived >
  double atomicLogDet (const MatrixBase< Derived > &m)

  Compute the sum of log-determinants of size-1 subsystems.
- template<typename Derived >
  double atomicEntropy (const MatrixBase< Derived > &m, double sizeSummand)

  Compute mean entropy for size-1 subsystems of a system from its scale matrix.
- template<typename Derived >
  double atomOutputLogDet (const MatrixBase< Derived > &m, const std::vector< int > &kernel, const std↩
  ::vector< int > &output)

  Compute sum of log-determinant for subsystems consisting of output system and a single kernel node (atom).
- template<typename Derived >
  double stableAtomOutputLogDet (const MatrixBase< Derived > &m, const std::vector< int > &kernel, const std::vector< int > &output)

  Compute sum of log-determinant for subsystems consisting of output system and a single kernel node (atom).
- template<typename Derived >
  double atomOutputEntropy (const MatrixBase< Derived > &m, const std::vector< int > &kernel, const std↩
  ::vector< int > &output, double sizeSummand)

  Compute sum of atom-output entropies.

- template<typename Derived >
  double stableAtomOutputEntropy (const MatrixBase< Derived > &m, const std::vector< int > &kernel, const std::vector< int > &output, double sizeSummand)

  Compute sum of atom-output entropies.

- template<typename Derived >
  double meanLogDet (const MatrixBase< Derived > &m, int k, int sampleSize, std::tr1::mt19937 &gen)

  Compute mean subsystem log-determinant from a system's scale matrix.

- template<typename Derived >
  double meanStableLogDet (const MatrixBase< Derived > &m, int k, int sampleSize, std::tr1::mt19937 &gen)

  Compute mean subsystem log-determinant from a system's scale matrix.

- template<typename Derived >
  std::vector< double > allLogDets (const MatrixBase< Derived > &m, int k, int sampleSize, std::tr1::mt19937 &gen)

  Compute list of log-determinants for subsystems of fixed size.

- template<typename Derived >
  std::vector< double > allStableLogDets (const MatrixBase< Derived > &m, int k, int sampleSize, std::tr1↩
  ::mt19937 &gen)

  Compute list of log-determinants for subsystems of fixed size.

- template<typename Derived >
  double meanCauchyEntropy (const MatrixBase< Derived > &m, int k, int sampleSize, std::tr1::mt19937 &gen)

  Compute mean subsystem Cauchy entropy from a system's scale matrix.

- template<typename Derived >
  double meanGaussEntropy (const MatrixBase< Derived > &m, int k, int sampleSize, std::tr1::mt19937 &gen)

  Compute mean subsystem Gauss entropy from a system's scale matrix.

- template<typename Derived >
  double meanStableCauchyEntropy (const MatrixBase< Derived > &m, int k, int sampleSize, std::tr1↩
  ::mt19937 &gen)

  Compute mean subsystem Cauchy entropy from a system's scale matrix.

- template<typename Derived >
  double meanStableGaussEntropy (const MatrixBase< Derived > &m, int k, int sampleSize, std::tr1::mt19937 &gen)

  Compute mean subsystem Gauss entropy from a system's scale matrix.

- template<typename Derived >
  std::vector< double > allCauchyEntropies (const MatrixBase< Derived > &m, int k, int sampleSize, std↩
  ::tr1::mt19937 &gen)

  Compute list of subsystem Cauchy entropies from scale matrix.

- template<typename Derived >
  std::vector< double > allGaussEntropies (const MatrixBase< Derived > &m, int k, int sampleSize, std::tr1↩
  ::mt19937 &gen)

  Compute list of subsystem Gauss entropies from scale matrix.

- template<typename Derived >
  std::vector< double > allStableCauchyEntropies (const MatrixBase< Derived > &m, int k, int sampleSize, std::tr1::mt19937 &gen)

  Compute list of subsystem Cauchy entropies from scale matrix.

- template<typename Derived >
  std::vector< double > allStableGaussEntropies (const MatrixBase< Derived > &m, int k, int sampleSize, std::tr1::mt19937 &gen)

  Compute list of subsystem Gauss entropies from scale matrix.

2.3.1 Friends And Related Function Documentation

2.3.1.1 template<typename Derived > std::vector< double > allCauchyEntropies (const MatrixBase< Derived > & *m*, int *k*, int *sampleSize*, std::tr1::mt19937 & *gen*) [related]

Compute list of subsystem Cauchy entropies from scale matrix.

Compute list of subsystem entropies for fixed-size subsystems with variables with Cauchy distribution. If system is large, consider using `allRescaledCauchyEntropies()`.

Parameters

in	*m*	Scale matrix (instance of Eigen::MatrixXd)
in	*k*	size of subsystems
in	*sampleSize*	number of samples for making list. If 0, make list of all subsystems
in	*gen*	random-number generator passed by reference (instance of std::tr1::mt19937)

Returns

list of subsystem Cauchy entropies (an instance of std::vector<double>)

2.3.1.2 template<typename Derived > std::vector< double > allGaussEntropies (const MatrixBase< Derived > & *m*, int *k*, int *sampleSize*, std::tr1::mt19937 & *gen*) [related]

Compute list of subsystem Gauss entropies from scale matrix.

Compute list of subsystem entropies for fixed-size subsystems with variables with Gauss distribution. If system is large, consider using `allRescaledGaussEntropies()`.

Parameters

in	*m*	Scale matrix (instance of Eigen::MatrixXd)
in	*k*	size of subsystems
in	*sampleSize*	number of samples for making list. If 0, make list of all subsystems
in	*gen*	random-number generator passed by reference (instance of std::tr1::mt19937)

Returns

list of subsystem Gauss entropies (an instance of std::vector<double>)

2.3.1.3 template<typename Derived > std::vector< double > allLogDets (const MatrixBase< Derived > & *m*, int *k*, int *sampleSize*, std::tr1::mt19937 & *gen*) [related]

Compute list of log-determinants for subsystems of fixed size.

Compute log-determinants for subsystems of fixed size from a system's scale matrix. If system is large, consider using `allRescaledLogDets()`.

Parameters

in	*m*	Scale matrix (instance of Eigen::MatrixXd)
in	*k*	size of subsystems
in	*sampleSize*	number of samples for making list. If 0, make list of all subsystems
in	*gen*	random-number generator passed by reference (instance of std::tr1::mt19937)

Returns

list of subsystem log-determinant (instance of std::vector<double>)

2.3.1.4 template<typename Derived > std::vector< double > allStableCauchyEntropies (const MatrixBase< Derived > & *m*, int *k*, int *sampleSize*, std::tr1::mt19937 & *gen*) [related]

Compute list of subsystem Cauchy entropies from scale matrix.

Compute list of subsystem entropies for fixed-size subsystems with variables with Cauchy distribution. Uses `stableEntropy()` to avoid overflow.

Parameters

in	*m*	Scale matrix (instance of Eigen::MatrixXd)
in	*k*	size of subsystems
in	*sampleSize*	number of samples for making list. If 0, make list of all subsystems
in	*gen*	random-number generator passed by reference (instance of std::tr1::mt19937)

Returns

list of subsystem Cauchy entropies (an instance of std::vector<double>)

2.3.1.5 template<typename Derived > std::vector< double > allStableGaussEntropies (const MatrixBase< Derived > & *m*, int *k*, int *sampleSize*, std::tr1::mt19937 & *gen*) [related]

Compute list of subsystem Gauss entropies from scale matrix.

Compute list of subsystem entropies for fixed-size subsystems with variables with Gauss distribution. Uses `stableEntropy()` to avoid overflow.

Parameters

in	*m*	Scale matrix (instance of Eigen::MatrixXd)
in	*k*	size of subsystems
in	*sampleSize*	number of samples for making list. If 0, make list of all subsystems
in	*gen*	random-number generator passed by reference (instance of std::tr1::mt19937)

Returns

list of subsystem Gauss entropies (an instance of std::vector<double>)

2.3.1.6 template<typename Derived > std::vector< double > allStableLogDets (const MatrixBase< Derived > & *m*, int *k*, int *sampleSize*, std::tr1::mt19937 & *gen*) [related]

Compute list of log-determinants for subsystems of fixed size.

Compute log-determinants for subsystems of fixed size from a system's scale matrix. Uses `stableLogDet()` to avoid overflow.

Parameters

in	*m*	Scale matrix (instance of Eigen::MatrixXd)
in	*k*	size of subsystems
in	*sampleSize*	number of samples for making list. If 0, make list of all subsystems
in	*gen*	random-number generator passed by reference (instance of std::tr1::mt19937)

Returns

list of subsystem log-determinant (instance of std::vector<double>)

2.3.1.7 template<typename Derived > double atomicEntropy (const MatrixBase< Derived > & *m*, double *sizeSummand*) [related]

Compute mean entropy for size-1 subsystems of a system from its scale matrix.

Compute mean entropy for size-1 subsystems of a system from its scale matrix and size summand.

Parameters

in	*m*	Scale matrix (instance of Eigen::MatrixXd)
in	*sizeSummand*	Size-dependent summand to entropy for size-1 system (see CauchySummand or GaussSummand)

Returns

mean of atomic entropies (double)

2.3.1.8 template<typename Derived > double atomicLogDet (const MatrixBase< Derived > & *m*) [related]

Compute the sum of log-determinants of size-1 subsystems.

Compute the sum of log-determinants of size-1 subsystems from scale matrix by taking the sum of logs of diagonal elements of the scale matrix.

Parameters

in	*m*	Scale matrix (instance of Eigen::MatrixXd)

Returns

sum of log-determinants of size-1 subsystems (double)

2.3.1.9 template<typename Derived > double atomOutputEntropy (const MatrixBase< Derived > & *m,* const std::vector< int > & *kernel,* const std::vector< int > & *output,* double *sizeSummand*) [related]

Compute sum of atom-output entropies.

Compute sum of subsystem entropies, where each subsystem consists of all output nodes and one kernel node. For large output sets, consider using `stableAtomOutputEntropy()` instead.

Parameters

in	*m*	Scale matrix (instance of Eigen::MatrixXd)
in	*kernel*	List of indices corresponding to kernel nodes (instance of std::vector<int>)
in	*output*	List of indices corresponding to output nodes (instance of std::vector<int>)
in	*sizeSummand*	Size-dependent summand to entropy for size-1 system (see CauchySummand or GaussSummand)

Returns

sum of atom-output entropies

2.3.1.10 template<typename Derived > double atomOutputLogDet (const MatrixBase< Derived > & *m,* const std::vector< int > & *kernel,* const std::vector< int > & *output*) [related]

Compute sum of log-determinant for subsystems consisting of output system and a single kernel node (atom).

Compute sum of subsystem log-determinant, where each subsystem consists of all output nodes and one kernel node. For large output sets, consider using `stableAtomOutputLogDet()`.

Parameters

in	*m*	Scale matrix (instance of Eigen::MatrixXd)
in	*kernel*	List of indices corresponding to kernel nodes (instance of std::vector<int>)
in	*output*	List of indices corresponding to output nodes (instance of std::vector<int>)

Returns

sum of subsystem log-determinants

2.3.1.11 double CauchySummand (int *n*) [related]

Compute a size-dependent summand for entropy of a system with multivariate Cauchy distribution.

An estimate of the entropy of a system of random variables with a Cauchy law is the sum of a `0.5*logDet` of the system's scale matrix and a term that only depends on the size of the system.

For formula for entropy estimator see page 70 in: Nadarajah & Kotz: "Mathematical Properties of the Multivariate t Distribution" Acta Applicandae Mathematicae (2005) 89: 53–74.

Parameters

in	*n0*	Size of system

Returns

Size-dependent summand for entropy (double)

2.3.1.12 template<typename Derived > double entropy (const MatrixBase< Derived > & *m*, double *sizeSummand*) [related]

Compute entropy for multivariate system from scale matrix and size summand.

Compute entropy for multivariate system from scale matrix and size summand. (See `CauchySummand()` or `GaussSummand()` for explanation of size-dependent summands.) CAUTION: May lead to overflow or underflow for large system sizes. To avoid, use `stableEntropy` instead in those cases.

Parameters

in	*m*	Scale matrix (instance of Eigen::MatrixXd)
in	*sizeSummand*	Size-dependent summand to entropy (see CauchySummand or GaussSummand)

Returns

entropy (double)

2.3.1.13 double GaussSummand (int *n*) [related]

Compute a size-dependent summand for entropy of a system with multivariate Gauss distribution.

An estimate of the entropy of a system of random variables with a Gauss law is the sum of a `0.5*logDet` of the system's scale matrix and a term that only depends on the size of the system.

Parameters

in	*n0*	Size of system

Returns

Size-dependent summand for entropy (double)

2.3.1.14 template<typename Derived > double logDet (const MatrixBase< Derived > & *m*) [related]

Compute log-determinant of a positive-definite square matrix.

Compute log-determinant of a positive-definite square matrix using `determinant()` from Eigen::Dense. CAUTION: May lead to overflow or underflow for large matrices. To avoid, use `stableLogDet()` instead in those cases.

Parameters

in	*m*	Positive-definite square matrix (Eigen::MatrixXd)

Returns

Log-determinant of matrix (double)

2.3.1.15 template<typename Derived > double meanCauchyEntropy (const MatrixBase< Derived > & *m*, int *k*, int *sampleSize*, std::tr1::mt19937 & *gen*) [related]

Compute mean subsystem Cauchy entropy from a system's scale matrix.

Compute mean subsystem entropy for subsystems of fixed size and variables with Cauchy distribution. If system is large, consider using `meanRescaledCauchyEntropy()`.

Parameters

in	*m*	Scale matrix (instance of Eigen::MatrixXd)
in	*k*	size of subsystems
in	*sampleSize*	number of samples for computing sample mean. If 0, compute population mean
in	*gen*	random-number generator passed by reference (instance of std::tr1::mt19937)

Returns

mean subsystem Cauchy entropy

2.3.1.16 template<typename Derived > double meanGaussEntropy (const MatrixBase< Derived > & *m*, int *k*, int *sampleSize*, std::tr1::mt19937 & *gen*) [related]

Compute mean subsystem Gauss entropy from a system's scale matrix.

Compute mean subsystem entropy for subsystems of fixed size and variables with Gauss distribution. If system is large, consider using `meanRescaledGaussEntropy()`.

Parameters

in	*m*	Scale matrix (instance of Eigen::MatrixXd)
in	*k*	size of subsystems
in	*sampleSize*	number of samples for computing sample mean. If 0, compute population mean
in	*gen*	random-number generator passed by reference (instance of std::tr1::mt19937)

Returns

mean subsystem Gauss entropy

2.3.1.17 template<typename Derived > double meanLogDet (const MatrixBase< Derived > & *m*, int *k*, int *sampleSize*, std::tr1::mt19937 & *gen*) [related]

Compute mean subsystem log-determinant from a system's scale matrix.

Compute mean log-determinants for subsystems of fixed size. If system is large, consider using `meanRescaledLogDet()`.

Parameters

in	*m*	Scale matrix (instance of Eigen::MatrixXd)
in	*k*	size of subsystems
in	*sampleSize*	number of samples for computing sample mean. If 0, compute population mean
in	*gen*	random-number generator passed by reference (instance of std::tr1::mt19937)

Returns

mean subsystem log-determinant

2.3.1.18 template<typename Derived > double meanStableCauchyEntropy (const MatrixBase< Derived > & *m*, int *k*, int *sampleSize*, std::tr1::mt19937 & *gen*) [related]

Compute mean subsystem Cauchy entropy from a system's scale matrix.

Compute mean subsystem entropy for subsystems of fixed size and variables with Cauchy distribution. Uses `stableEntropy()` to avoid overflow.

Parameters

in	*m*	Scale matrix (instance of Eigen::MatrixXd)
in	*k*	size of subsystems
in	*sampleSize*	number of samples for computing sample mean. If 0, compute population mean
in	*gen*	random-number generator passed by reference (instance of std::tr1::mt19937)

Returns

mean subsystem Cauchy entropy

2.3.1.19 template<typename Derived > double meanStableGaussEntropy (const MatrixBase< Derived > & *m*, int *k*, int *sampleSize*, std::tr1::mt19937 & *gen*) [related]

Compute mean subsystem Gauss entropy from a system's scale matrix.

Compute mean subsystem entropy for subsystems of fixed size and variables with Gauss distribution. Uses `stableEntropy()` to avoid overflow.

Parameters

in	*m*	Scale matrix (instance of Eigen::MatrixXd)
in	*k*	size of subsystems
in	*sampleSize*	number of samples for computing sample mean. If 0, compute population mean
in	*gen*	random-number generator passed by reference (instance of std::tr1::mt19937)

Returns

mean subsystem Gauss entropy

2.3.1.20 template<typename Derived > double meanStableLogDet (const MatrixBase< Derived > & *m*, int *k*, int *sampleSize*, std::tr1::mt19937 & *gen*) [related]

Compute mean subsystem log-determinant from a system's scale matrix.

Compute mean log-determinants for subsystems of fixed size. Uses `stableLogDet()` to avoid overflow.

Parameters

in	*m*	Scale matrix (instance of Eigen::MatrixXd)
in	*k*	size of subsystems
in	*sampleSize*	number of samples for computing sample mean. If 0, compute population mean
in	*gen*	random-number generator passed by reference (instance of std::tr1::mt19937)

Returns

mean subsystem log-determinant

2.3.1.21 template<typename Derived > double stableAtomOutputEntropy (const MatrixBase< Derived > & *m*, const std::vector< int > & *kernel*, const std::vector< int > & *output*, double *sizeSummand*) [related]

Compute sum of atom-output entropies.

Compute sum of subsystem entropies, where each subsystem consists of all output nodes and one kernel node. Uses `stableEntropy()` to avoid overflow.

Parameters

in	*m*	Scale matrix (instance of Eigen::MatrixXd)
in	*kernel*	List of indices corresponding to kernel nodes (instance of std::vector<int>)
in	*output*	List of indices corresponding to output nodes (instance of std::vector<int>)
in	*sizeSummand*	Size-dependent summand to entropy for size-1 system (see CauchySummand or GaussSummand)

Returns

sum of atom-output entropies

2.3.1.22 template<typename Derived > double stableAtomOutputLogDet (const MatrixBase< Derived > & *m*, const std::vector< int > & *kernel*, const std::vector< int > & *output*) [related]

Compute sum of log-determinant for subsystems consisting of output system and a single kernel node (atom).

Compute sum of subsystem log-determinant, where each subsystem consists of all output nodes and one kernel node. Uses `stableLogDet()` to avoid overflow.

Parameters

in	*m*	Scale matrix (instance of Eigen::MatrixXd)
in	*kernel*	List of indices corresponding to kernel nodes (instance of std::vector<int>)
in	*output*	List of indices corresponding to output nodes (instance of std::vector<int>)

Returns

sum of subsystem log-determinants

2.3.1.23 template<typename Derived > double stableEntropy (const MatrixBase< Derived > & *m*, double *sizeSummand*) [related]

Compute entropy for multivariate system from scale matrix and size summand.

Compute entropy for multivariate system from scale matrix and size summand. (See `CauchySummand()` or `GaussSummand()` for explanation of size-dependent summands.) To avoid overflow, we use `stableLogDet()`.

Parameters

in	*m*	Scale matrix (instance of Eigen::MatrixXd)
in	*sizeSummand*	Size-dependent summand to entropy (see CauchySummand or GaussSummand)

Returns

entropy (double)

2.3.1.24 template<typename Derived > double stableLogDet (const MatrixBase< Derived > & *m*) [related]

Compute log-determinant of a positive-definite square matrix.

Compute log-determinant of a positive-definite square matrix. This function depends on compiler settings: If compiled without flag, it uses `determinant()` from Eigen::Dense on `m/m.mean()` and rescales result after taking the logarithm. This avoids overflow in the determinant. If compiled with -D DYER, it uses a GitHub gist by Chris Dyer that uses an LU decomposition with partial pivoting. (In initial tests, this was 2.5% faster than using `determinant()`.)

Parameters

`in`	*m*	Positive-definite square matrix (Eigen::MatrixXd)

Returns

Log-determinant of matrix (double)

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

- /home/schwarze/Desktop/RIThM.h

2.4 HELPER_FUNCTIONS Class Reference

Related Functions

(Note that these are not member functions.)

- class DATA_PREPROCESSING print (T message)
  Print like in python. Why? Because I am lazy.
- template<typename T >
  void printVec (const std::vector< T > &path)
  Print an instance of std::vector.
- template<typename T >
  void printTime (T title, std::chrono::time_point< std::chrono::_V2::high_resolution_clock > t0, std::chrono::time_point< std::chrono::_V2::high_resolution_clock > t1)
  Print elapsed time between t0 and t1.
- template<typename T >
  void printNanoTime (T title, std::chrono::time_point< std::chrono::_V2::high_resolution_clock > t0, std::chrono::time_point< std::chrono::_V2::high_resolution_clock > t1)
  Print elapsed time between t0 and t1 in nanoseconds.
- std::chrono::time_point< std::chrono::_V2::high_resolution_clock > now ()
  Shorthand for current system time.
- template<class It >
  unsigned display (It begin, It end)
  Print elements in iterable to shell with comma separation.
- std::vector< int > range (int lowerLimit, int upperLimit)
  Constructor for std::vector<int> that is equivalent to `range(lowerLimit, upperLimit, 1)` in python.
- std::vector< int > complementVec (int n, std::vector< int > &vec)
  Vector with all integers up to `n` that are not in `vec`.
- template<typename T >
  void saveVec (const std::string &filename, std::vector< T > &vec)
  Save content of an std::vector to a file.

2.4.1 Friends And Related Function Documentation

2.4.1.1 std::vector< int > complementVec (int *n*, std::vector< int > & *vec*) [related]

Vector with all integers up to `n` that are not in `vec`.

Vector with all integers up to `n` that are not in `vec`. Useful for constructing sets of kernel nodes for a given set of output nodes.

Parameters

in	*n*	
in	*vec*	

Returns

vector with all integers up to `n` that are not in `vec`

2.4.1.2 template<class It > unsigned display (It *begin*, It *end*) [related]

Print elements in iterable to shell with comma separation.

Print elements in iterable to shell with comma separation.

Parameters

in	*begin*	Iterable.begin()
in	*end*	Iterable.end()

Returns

Number of elements in iterable

2.4.1.3 std::vector< int > range (int *lowerLimit*, int *upperLimit*) [related]

Constructor for std::vector<int> that is equivalent to `range(lowerLimit, upperLimit, 1)` in python.

Constructor for std::vector<int> that is equivalent to `range(lowerLimit, upperLimit, 1)` in python.

Parameters

in	*lowerLimit*	
in	*upperLimit*	

Returns

standard vector with elements in range [lowerLimit, upperLimit)

2.4.1.4 template<typename T > void saveVec (const std::string & *filename,* std::vector< T > & *vec*) [related]

Save content of an std::vector to a file.

Save content of an std::vector to a file.

Parameters

in	*filename*	
in	*vec*	

Returns

void

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

- /home/schwarze/Desktop/RIThM.h

2.5 LOG_MINORS Class Reference

Functor for returning list of logi-determinants of principal submatrices.

`#include <RIThM.h>`

Public Member Functions

- **LOG_MINORS** (MatrixXd a0, int l0)
- template<class It >
  bool **operator()** (It first, It last)
- **operator std::vector**< **double** > () const

2.5.1 Detailed Description

Functor for returning list of logi-determinants of principal submatrices.

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

- /home/schwarze/Desktop/RIThM.h

2.6 STABLE_LOG_MINORS Class Reference

Functor for returning list of log-determinants of principal submatrices via `stableLogDet()`.

`#include <RIThM.h>`

Public Member Functions

- **STABLE_LOG_MINORS** (MatrixXd a0, int l0)
- template<class It >
  bool **operator()** (It first, It last)
- **operator std::vector**< **double** > () const

2.6.1 Detailed Description

Functor for returning list of log-determinants of principal submatrices via `stableLogDet()`.

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

- /home/schwarze/Desktop/RIThM.h

2.7 STABLE_SUM_LOG_MINORS Class Reference

Functor for computing the sum of log-determinants of principal submatrices via `stableLogDet()`.

```
#include <RIThM.h>
```

Public Member Functions

- **STABLE_SUM_LOG_MINORS** (MatrixXd a0, int l0)
- template<class It >
  bool **operator()** (It first, It last)
- **operator double** () const

2.7.1 Detailed Description

Functor for computing the sum of log-determinants of principal submatrices via `stableLogDet()`.

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

- /home/schwarze/Desktop/RIThM.h

2.8 SUM_LOG_DET_REDUNDANCY Class Reference

Functor for computing sum of subsystem redundancies from log-determinants.

```
#include <RIThM.h>
```

Public Member Functions

- **SUM_LOG_DET_REDUNDANCY** (MatrixXd a0, std::vector< int > output0, int subSize0)
- template<class It >
  bool **operator()** (It first, It last)
- **operator double** () const

2.8.1 Detailed Description

Functor for computing sum of subsystem redundancies from log-determinants.

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

- /home/schwarze/Desktop/RIThM.h

2.9 SUM_LOG_MINORS Class Reference

Functor for computing the sum of log-determinants of principal submatrices.

`#include <RIThM.h>`

Public Member Functions

- **SUM_LOG_MINORS** (MatrixXd a0, int l0)
- template<class It >
  bool **operator()** (It first, It last)
- **operator double** () const

2.9.1 Detailed Description

Functor for computing the sum of log-determinants of principal submatrices.

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

- /home/schwarze/Desktop/RIThM.h

2.10 SUM_STABLE_LOG_DET_REDUNDANCY Class Reference

Functor for computing sum of subsystem redundancies from log-determinants obtained via `stableLogDet()`.

`#include <RIThM.h>`

Public Member Functions

- **SUM_STABLE_LOG_DET_REDUNDANCY** (MatrixXd a0, std::vector< int > output0, int subSize0)
- template<class It >
  bool **operator()** (It first, It last)
- **operator double** () const

2.10.1 Detailed Description

Functor for computing sum of subsystem redundancies from log-determinants obtained via `stableLogDet()`.

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

- /home/schwarze/Desktop/RIThM.h

Index