Master Thesis

Generated by Doxygen 1.8.4

Sun Jul 7 2013 15:26:06

Contents

Chapter 1

Namespace Index

1.1 Namespace List

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

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

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

Chapter 5

Namespace Documentation

5.1 PRPSEvolution Namespace Reference

Classes

- struct Normalizer
- struct Constants
- struct System

Enumerations

- enum **NormalizatioMethodes** { **Native**, **B**, **CMPLX**, **RND** }

Variables

- const int **ANTENNA_AMOUNT** = 8
- const int **EXPECTED_LINES_CALIBRATION_FILE** = 4
- const int **EXPECTED_VALUES_CALIBRATION_FILE** = ANTENNA_AMOUNT
- const int **EXPECTED_LINES_COORD_FILE** = ANTENNA_AMOUNT
- const int **EXPECTED_VALUES_COORD_FILE** = 3
- const int **EXPECTED_LINES_SYSTEM_INI_FILE** = 2
- const int **MAT_ROWS** = 3
- const int **MAT_COLS** = 10
- const int **CALIBRATION_POINTS_AVAILABLE** = 4
- const int **EXPECTED_LINES_MEASUREMENT_FILE** = ANTENNA_AMOUNT
- const int **EXPECTED_VALUES_MEASUREMENT_FILE** = 2
- const int **DATA_NV** = 65535
- const int **DEFAULT_MIN_GROUP_SIZE** = 4

5.1.1 Detailed Description

This file contains structures and classes belonging to the system itself

Chapter 6

Class Documentation

6.1 PRPSEvolution::Permutate::AntennaPermutations< N_MAT, T > Struct Template Reference

Static Public Member Functions

- static void **dump_matrix** (NRmatrix< T > mat)
- static void **dump_matrix_2_file** (std::ofstream &f, NRmatrix< T > mat)

Public Attributes

- std::array< NRmatrix< T >, N_MAT > **mat**
- std::array< std::string, N_MAT > **names**

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

- trunk/libPermutate/permutate.h

6.2 PRPSEvolution::Exceptions::Permutation::Calculation Class Reference

Inheritance diagram for PRPSEvolution::Exceptions::Permutation::Calculation:

Collaboration diagram for PRPSEvolution::Exceptions::Permutation::Calculation:

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

- trunk/include/PRPSEvolutionPermutationExceptions.h

6.3 PRPSEvolution::Exeptions::Permutation::Calculation Class Reference

Inheritance diagram for PRPSEvolution::Exeptions::Permutation::Calculation:

Collaboration diagram for PRPSEvolution::Exeptions::Permutation::Calculation:

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

- trunk/include/PRPSEvolutionPermutationExeptions.h

6.4 PRPSEvolution::Constants Struct Reference

Public Member Functions

- **Constants** (const PRPSEvolution::Constants &c)

Public Attributes

- double **a_1**
- double **a_2**
- double **lambda**
- double **f_mess**
- double **c_0**

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

- trunk/libPRPSSystem/prpsevolutionsystem.h

6.5 PRPSEvolution::Positioning::CoordContainer< N, T > Struct Template Reference

Collaboration diagram for PRPSEvolution::Positioning::CoordContainer< N, T >:

Public Types

- typedef T **value_type**

Public Member Functions

- template<typename T1 >
  CoordContainer (T1 init)
- T & **operator[ ]** (std::size_t i)

Public Attributes

- T **x_** [N]
- T **y_** [N]
- T **z_** [N]

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

- trunk/include/coords.h

6.6 PRPSEvolution::Exceptions::FileIO::FileNotFound Class Reference

Inheritance diagram for PRPSEvolution::Exceptions::FileIO::FileNotFound:

Collaboration diagram for PRPSEvolution::Exceptions::FileIO::FileNotFound:

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

- trunk/include/PRPSEvolutionFIOExceptions.h

6.7 PRPSEvolution::Exeptions::FileIO::FileNotFound Class Reference

Inheritance diagram for PRPSEvolution::Exeptions::FileIO::FileNotFound:

Collaboration diagram for PRPSEvolution::Exeptions::FileIO::FileNotFound:

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

- trunk/include/PRPSEvolutionFIOExeptions.h

6.8 PRPSEvolution::Exceptions::Permutation::Init Class Reference

Inheritance diagram for PRPSEvolution::Exceptions::Permutation::Init:

Collaboration diagram for PRPSEvolution::Exceptions::Permutation::Init:

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

- trunk/include/PRPSEvolutionPermutationExceptions.h

6.9 PRPSEvolution::Exeptions::Permutation::Init Class Reference

Inheritance diagram for PRPSEvolution::Exeptions::Permutation::Init:

Collaboration diagram for PRPSEvolution::Exeptions::Permutation::Init:

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

- trunk/include/PRPSEvolutionPermutationExeptions.h

6.10 PRPSEvolution::Exeptions::FileIO::MalformedInput Class Reference

Inheritance diagram for PRPSEvolution::Exeptions::FileIO::MalformedInput:

Collaboration diagram for PRPSEvolution::Exeptions::FileIO::MalformedInput:

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

- trunk/include/PRPSEvolutionFIOExeptions.h

6.11 PRPSEvolution::Exceptions::FileIO::MalformedInput Class Reference

Inheritance diagram for PRPSEvolution::Exceptions::FileIO::MalformedInput:

Collaboration diagram for PRPSEvolution::Exceptions::FileIO::MalformedInput:

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

- trunk/include/PRPSEvolutionFIOExceptions.h

6.12 PRPSEvolution::Normalizer< N, T > Struct Template Reference

Public Member Functions

- Normalizer (NormalizatioMethodes method)

- std::array< T, N > **complexNorm** (const std::array< T, N > &p, const std::array< T, N > &a)
- std::array< T, N > **randNorm** ()
- std::array< T, N > normalize (std::array< T, N > phase, std::array< T, N > amp)

Public Attributes

- NormalizatioMethodes **Method**

6.12.1 Constructor & Destructor Documentation

6.12.1.1 template<std::size_t N, typename T > PRPSEvolution::Normalizer< N, T >::Normalizer (NormalizatioMethodes *method*) [inline]

Constructor

Parameters

in	*method*	Selects the Normalization function

6.12.2 Member Function Documentation

6.12.2.1 template<std::size_t N, typename T > std::array<T, N> PRPSEvolution::Normalizer< N, T >::normalize (std::array< T, N > *phase,* std::array< T, N > *amp*) [inline]

Calculates the normalizations

Parameters

in	*phase*	The measured phase data
in	*amp*	The measured amplitude data

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

- trunk/libNormalizer/normalizer.h

6.13 PRPSEvolution::Exceptions::General::NotImplemented Struct Reference

```
#include <PRPSEvolutionGeneralExceptions.h>
```

Inheritance diagram for PRPSEvolution::Exceptions::General::NotImplemented:

Collaboration diagram for PRPSEvolution::Exceptions::General::NotImplemented:

Public Member Functions

- const char ∗ **what** () const noexcept

6.13.1 Detailed Description

Throw this if a Method is not implemented

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

- trunk/include/PRPSEvolutionGeneralExceptions.h

6.14 PRPSEvolution::Exceptions::FileIO::Output Class Reference

Inheritance diagram for PRPSEvolution::Exceptions::FileIO::Output:

Collaboration diagram for PRPSEvolution::Exceptions::FileIO::Output:

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

- trunk/include/PRPSEvolutionFIOExceptions.h

6.15 PRPSEvolution::Exeptions::FileIO::Output Class Reference

Inheritance diagram for PRPSEvolution::Exeptions::FileIO::Output:

Collaboration diagram for PRPSEvolution::Exeptions::FileIO::Output:

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

- trunk/include/PRPSEvolutionFIOExeptions.h

6.16 PRPSEvolution::Calibration::performCalibration< N_ANTA, N_CALPOS, T > Struct Template Reference

```
#include <calib.h>
```

6.16.1 Detailed Description

template<std::size_t N_ANTA, std::size_t N_CALPOS, typename T>struct PRPSEvolution::Calibration::performCalibration< N_ANTA, N_CALPOS, T >

This will perform the calibration stuff

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

- trunk/libCalibration/calib.h

6.17 PRPSEvolution::Permutate::permuteAntennas< N_ANTA, N_ANTPERM, T > Struct Template Reference

```
#include <permutate.h>
```

Collaboration diagram for PRPSEvolution::Permutate::permuteAntennas< N_ANTA, N_ANTPERM, T >:

Public Member Functions

- **permuteAntennas** (const PRPSEvolution::Constants c)
- int rCoordFile ()
- int computePermutations (const PRPSEvolution::Constants &co)
- template<std::size_t NN, std::size_t MM>
const NRmatrix< T > computeMatrix (const int ref, const int a1, const int a2, const int a3, const PRPSEvolution::Constants &co)
- NRmatrix< T > **compute_d_k0_Mat** ()
- void **dumpConfigurationsToFile** ()
- void dump_matrices_2_file ()

Public Attributes

- int **ref**
- PRPSEvolution::Constants **systemConstants**
- Positioning::CoordContainer
  < N_ANTA, T > **AntennaCoordinates**
- std::array
  < AntennaPermutations
  < N_ANTPERM, Doub >, N_ANTA > **configurations**
- NRmatrix< T > **d_k0_mat**

6.17.1 Detailed Description

template<std::size_t N_ANTA, std::size_t N_ANTPERM, typename T>struct PRPSEvolution::Permutate::permuteAntennas< N_ANTA, N_ANTPERM, T >

This will collect some stuff for calculating the permutation of the antennas

6.17.2 Member Function Documentation

6.17.2.1 template<std::size_t N_ANTA, std::size_t N_ANTPERM, typename T > template<std::size_t NN, std::size_t MM> const NRmatrix< T > PRPSEvolution::Permutate::permuteAntennas< N_ANTA, N_ANTPERM, T >::computeMatrix (const int *ref,* const int *a1,* const int *a2,* const int *a3,* const PRPSEvolution::Constants & *co*)

This method will compute all the possible permutations based on the given reference antenna

See Also

ref

Parameters

in	*ref*	The reference antenna
in	*a1*	First antenna
in	*a2*	Second antenna
in	*a3*	Third antenna

6.17.2.2 template<std::size_t N_ANTA, std::size_t N_ANTPERM, typename T > int PRPSEvolution::Permutate::permuteAntennas< N_ANTA, N_ANTPERM, T >::computePermutations (const PRPSEvolution::Constants & *co*)

This method handles the computation of the antenna permutations

Parameters

in	*co*	Constant structure with the system constants we need

See Also

PRPSEvolution::Constants

6.17.2.3 template< std::size_t N_ANTA, std::size_t N_ANTPERM, typename T > void **PRPSEvolution::Permutate::permuteAntennas**< N_ANTA, N_ANTPERM, T >::dump_matrices_2_file ( )

This method will dump all the Antennas to an output file

6.17.2.4 template< std::size_t N_ANTA, std::size_t N_ANTPERM, typename T > int **PRPSEvolution::Permutate::permuteAntennas**< N_ANTA, N_ANTPERM, T >::rCoordFile ( )

Load the csv-file containing the coordinates and store it into the container.

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

- trunk/libPermutate/permutate.h

6.18 PRPSEvolution::Solve::PostProcessing Class Reference

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

- trunk/libSolve/solve.h

6.19 PRPSEvolution::Solve::PreProcessing< N_ANTA, N_Configs, T, T_Measure > Class Template Reference

Public Member Functions

- PreProcessing (const std::array< AntennaPermutations< Permutate::MAX_PERMUTATION_AMOUNT, Doub >, N_ANTA > &, const NRmatrix< T > &)

Public Attributes

- std::vector< NRmatrix< T > > matricesForSolution
- std::vector< NRvector< T > > **vectorsForSolution**
- std::vector< std::string > **ConfigurationNames**

6.19.1 Constructor & Destructor Documentation

6.19.1.1 template< std::size_t N_ANTA, std::size_t N_Configs, typename T , typename T_Measure > **PRPSEvolution::Solve::PreProcessing**< N_ANTA, N_Configs, T, T_Measure >::**PreProcessing** (const std::array< **AntennaPermutations**< Permutate::MAX_PERMUTATION_AMOUNT, Doub >, N_ANTA > & *precalculatedMatrices*, const NRmatrix< T > & *d_k0s*)

Construct the object an perform neccessary PreProcessing steps.

1. Read out the measurements from the given interface (e.g. a file)
2. Normalize everything
3. Select the matrices for further processing
4. Fill the matrices with the information

5. Precalculate the

$$c_{k0}$$

   -Vector

6. Store matrices to make them availiable in the next steps

6.19.2 Member Data Documentation

6.19.2.1 template<std::size_t N_ANTA, std::size_t N_Configs, typename T, typename T_Measure> std::vector<NRmatrix<T> > **PRPSEvolution::Solve::PreProcessing**< N_ANTA, N_Configs, T, T_Measure >::matricesForSolution

determines how many Configurations will solved for the Wavenumber

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

- trunk/libSolve/solve.h

6.20 PRPSEvolution::Solve::ProblemDimensions Struct Reference

Static Public Attributes

- static const int **WholeTomatoeApproach** = 7
- static const int **Sphere** = 10
- static const int **Rosenbrock** = 15

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

- trunk/libSolve/solve.h

6.21 PRPSEvolution::Solve::Process Class Reference

```
#include <solve.h>
```

Public Member Functions

- Process ()
- **Process** (const Process &p)
- double **getLastSolutionFitness** ()
- template<typename T >
  T findSolution (const NRmatrix< Doub > &A_selected, const NRvector< Doub > &b_selected, ESStrategy strategy, int seed)
- int **sq** (int i)
- void setMinSolutionFitness (double value)
- void **setSeed** (unsigned int value)

6.21.1 Detailed Description

Find solutions for the possible matrices

6.21.2 Constructor & Destructor Documentation

6.21.2.1 PRPSEvolution::Solve::Process::Process ( ) [inline]

a Pointer to the object containing the fitness functions Constructor

6.21.3 Member Function Documentation

6.21.3.1 template< typename T > T PRPSEvolution::Solve::Process::findSolution (const NRmatrix< Doub > & *A_selected,* const NRvector< Doub > & *b_selected,* ESStrategy *strategy,* int *seed*) [inline]

Set the ES-Strategy

Parameters

in	*Strategy*	The selected strategy Find a Solution for a given pair of matrices
in	*A_selected*	The matrix A to use in this solution
in	*b_selected*	The c_k0' vector for this solution

Returns

The solution

6.21.3.2 void PRPSEvolution::Solve::Process::setMinSolutionFitness (double *value*) [inline]

Sets the min. solution fitness we want to achieve.

Parameters

in	*value*	The new value for the solution fitness

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

- trunk/libSolve/solve.h

6.22 PRPSEvolution::Solve::solveresult_t< T_Store, T_Return > Struct Template Reference

```
#include <solveresult.h>
```

Public Attributes

- T_Store values
- T_Return fitness
- int iterations
- int duration
- bool converged

6.22.1 Detailed Description

template<typename T_Store, typename T_Return>struct PRPSEvolution::Solve::solveresult_t< T_Store, T_Return >

Stores the final state of a solution

6.22.2 Member Data Documentation

6.22.2.1 template<typename T_Store , typename T_Return > bool **PRPSEvolution::Solve::solveresult_t**< T_Store, T_Return >::converged

Indicates whether the build in convergence criterium was applied, or not

6.22.2.2 template<typename T_Store , typename T_Return > int **PRPSEvolution::Solve::solveresult_t**< T_Store, T_Return >::duration

The processing time for this solution

6.22.2.3 template<typename T_Store , typename T_Return > T_Return **PRPSEvolution::Solve::solveresult_t**< T_Store, T_Return >::fitness

The fitness value

6.22.2.4 template<typename T_Store , typename T_Return > int **PRPSEvolution::Solve::solveresult_t**< T_Store, T_Return >::iterations

The amount of iterations needed for this result

6.22.2.5 template<typename T_Store , typename T_Return > T_Store **PRPSEvolution::Solve::solveresult_t**< T_Store, T_Return >::values

Whrere the result is stored

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

- trunk/libSolve/solveresult.h

6.23 PRPSEvolution::System Struct Reference

Collaboration diagram for PRPSEvolution::System:

Public Member Functions

- System (const PRPSEvolution::System &s)
- int **rPRPSIniFile** ()

Public Attributes

- PRPSEvolution::Constants **constants**
- std::string **fn**

6.23.1 Constructor & Destructor Documentation

6.23.1.1 PRPSEvolution::System::System (const **PRPSEvolution::System** & *s*) [inline]

copy constructor

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

- trunk/libPRPSSystem/prpsevolutionsystem.h

6.24 PRPSEvolution::Solve::Ueber9000< T > Struct Template Reference

```
#include <solve.h>
```

Public Member Functions

- Ueber9000 ()
- **Ueber9000** (const Ueber9000 &me)
- Ueber9000 (const NRmatrix< T > A_selected, const NRvector< T > c_k0_selected)
- double WholeTomatoeApproach (const ChromosomeT< double > &x)
- double WholeTomatoeApproach (const NRmatrix< T > &A, const ChromosomeT< double > &x, const NRvector< T > &b)
- double WavenumberVariation (const ChromosomeT< double > &n)
- double PositionVariation (const ChromosomeT< double > &pos)
- double fitnessSphere (const ChromosomeT< double > &c)
- double fitnessRosenbrock (const ChromosomeT< double > &c)
- double fitnessAckley (const std::vector< double > &x)

Public Attributes

- double(Ueber9000< double >::* **evaluate**)(const ChromosomeT< double > &)
- int **Dimension**
- NRmatrix< T > **A**
- NRvector< T > **c_k0**

6.24.1 Detailed Description

template<typename T>struct PRPSEvolution::Solve::Ueber9000< T >

Collect the fitness functions. Make sure they are static so we can function-pointer to them.

6.24.2 Constructor & Destructor Documentation

6.24.2.1 template<typename T> PRPSEvolution::Solve::Ueber9000< T >::Ueber9000 ( ) [inline]

Default constructor

6.24.2.2 template<typename T> PRPSEvolution::Solve::Ueber9000< T >::Ueber9000 (const NRmatrix< T > *A_selected*, const NRvector< T > *c_k0_selected*) [inline]

Construct Ueber9000 to use the WholeTomatoeApproach as fitness function

Parameters

in	*A_selected*	The matrix A for this Solution
in	*c_k0_selected*	The vector c_k0 for this Solution

6.24.3 Member Function Documentation

6.24.3.1 template< typename T > double PRPSEvolution::Solve::Ueber9000< T >::fitnessAckley (const std::vector< double > & *x*) [inline]

The infamous Ackley-function

6.24.3.2 template< typename T > double PRPSEvolution::Solve::Ueber9000< T >::fitnessRosenbrock (const ChromosomeT< double > & *c*) [inline]

The Rosenbrock implementation

6.24.3.3 template< typename T > double PRPSEvolution::Solve::Ueber9000< T >::fitnessSphere (const ChromosomeT< double > & *c*) [inline]

This ist the fitness function used in the EA algorithm

6.24.3.4 template< typename T > double PRPSEvolution::Solve::Ueber9000< T >::PositionVariation (const ChromosomeT< double > & *pos*) [inline]

Approach 3 based on the thoughts of by S. Winter

6.24.3.5 template< typename T > double PRPSEvolution::Solve::Ueber9000< T >::WavenumberVariation (const ChromosomeT< double > & *n*) [inline]

Approach 2 based on the thoughts of S. Winter. Here we want to optimize the wavenumbers

6.24.3.6 template< typename T > double PRPSEvolution::Solve::Ueber9000< T >::WholeTomatoeApproach (const ChromosomeT< double > & *x*) [inline]

This method basically wraps around the real WholeTomatoeApproach-function. Maps the function so that it can be used with the evaluate-method

Parameters

in	*x*	The vector x

6.24.3.7 template< typename T > double PRPSEvolution::Solve::Ueber9000< T >::WholeTomatoeApproach (const NRmatrix< T > & *A*, const ChromosomeT< double > & *x*, const NRvector< T > & *b*) [inline]

This approach will solve calculate the 10x3 matrix described in the Master-Thesis of C.Gnip Basically solves the linear equation

$$r = \mathbf{A}\mathbf{x} - \mathbf{b}$$

Parameters

in	*A*	The 10x3 Matrix that ist used in this solution
in	*x*	The vector containing the variables
in	*b*	Representing the vector b

Returns

The residuum of the equation system representing the "Fitness" of the given Solution in

See Also

x

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

- trunk/libSolve/solve.h

Chapter 7

File Documentation

7.1 trunk/include/PRPSError.h File Reference

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

Variables

- const int **PRPSError::FileIO::okay** = 0
- const int **PRPSError::FileIO::generalError** = -1
- const int **PRPSError::FileIO::fnf** = -2
- const int **PRPSError::FileIO::inputmalformed** = -3
- const int **PRPSError::okay** = 0
- const int **PRPSError::general** = -1
- const int **PRPSError::critical** = 10

7.1.1 Detailed Description

Date

2013|Jun|18 This file cantains definitions belonging to the PRPSError-namespace. It is split into sub-namespaces for keeping thing nicely small.

7.2 trunk/include/PRPSEvolution.h File Reference

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

Namespaces

- PRPSEvolution

Constant Groups

- PRPSEvolution

Variables

- const int **PRPSEvolution::ANTENNA_AMOUNT** = 8
- const int **PRPSEvolution::EXPECTED_LINES_CALIBRATION_FILE** = 4
- const int **PRPSEvolution::EXPECTED_VALUES_CALIBRATION_FILE** = ANTENNA_AMOUNT
- const int **PRPSEvolution::EXPECTED_LINES_COORD_FILE** = ANTENNA_AMOUNT
- const int **PRPSEvolution::EXPECTED_VALUES_COORD_FILE** = 3
- const int **PRPSEvolution::EXPECTED_LINES_SYSTEM_INI_FILE** = 2
- const int **PRPSEvolution::MAT_ROWS** = 3
- const int **PRPSEvolution::MAT_COLS** = 10
- const int **PRPSEvolution::CALIBRATION_POINTS_AVAILABLE** = 4
- const int **PRPSEvolution::EXPECTED_LINES_MEASUREMENT_FILE** = ANTENNA_AMOUNT
- const int **PRPSEvolution::EXPECTED_VALUES_MEASUREMENT_FILE** = 2
- const int **PRPSEvolution::DATA_NV** = 65535
- const int **PRPSEvolution::DEFAULT_MIN_GROUP_SIZE** = 4

7.2.1 Detailed Description

Date

2013|Jun|18 This file collects definitions belonging to the PRPSEvolution-namespace. Especially const. defines.

7.3 trunk/include/PRPSEvolutionFIOExeptions.h File Reference

```
#include <iostream>
#include <exception>
```

Include dependency graph for PRPSEvolutionFIOExeptions.h:

Classes

- class PRPSEvolution::Exeptions::FileIO::FileNotFound
- class PRPSEvolution::Exeptions::FileIO::MalformedInput
- class PRPSEvolution::Exeptions::FileIO::Output

Namespaces

- PRPSEvolution

Constant Groups

- PRPSEvolution

Variables

- PRPSEvolution::Exeptions::FileIO::FileNotFound **PRPSEvolution::Exeptions::FileIO::FNFExeption**
- PRPSEvolution::Exeptions::FileIO::MalformedInput **PRPSEvolution::Exeptions::FileIO::MalformedInputExeption**
- PRPSEvolution::Exeptions::FileIO::Output **PRPSEvolution::Exeptions::FileIO::OutputExeption**

7.3.1 Detailed Description

Exeptions of file IO

7.4 trunk/include/PRPSEvolutionGeneralExeptions.h File Reference

```
#include <iostream>
#include <exception>
#include <string>
```

Include dependency graph for PRPSEvolutionGeneralExeptions.h:

Namespaces

- PRPSEvolution

Constant Groups

- PRPSEvolution

7.4.1 Detailed Description

Contains the exeption used in the PRPSEvolution - Project

7.5 trunk/include/PRPSEvolutionSolveExeptions.h File Reference

```
#include <iostream>
#include <exception>
```

Include dependency graph for PRPSEvolutionSolveExeptions.h:

Namespaces

- PRPSEvolution

Constant Groups

- PRPSEvolution

7.5.1 Detailed Description

Contains the exeption used in the PRPSEvolution - Project

7.6 trunk/libCalibration/calib.h File Reference

```
#include "nr3/nr3.h"
#include "nr3/svd.h"
#include "../include/coords.h"
#include "../include/PRPSEvolution.h"
#include "../include/PRPSEvolutionCalibrationExceptions.h"
#include "../include/PRPSError.h"
#include "../libPRPSSystem/prpsevolutionsystem.h"
```

Include dependency graph for calib.h: This graph shows which files directly or indirectly include this file:

Classes

- struct PRPSEvolution::Calibration::performCalibration< N_ANTA, N_CALPOS, T >

Namespaces

- PRPSEvolution

Constant Groups

- PRPSEvolution

7.6.1 Detailed Description

Date

2013|Jun|25

7.7 trunk/libNormalizer/normalizer.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <iostream>
#include "../include/PRPSEvolutionGeneralExceptions.h"
#include "../include/PRPSEvolutionFIOExceptions.h"
#include <complex>
#include <array>
#include <random>
```

Include dependency graph for normalizer.h: This graph shows which files directly or indirectly include this file:

Classes

- struct PRPSEvolution::Normalizer< N, T >

Namespaces

- PRPSEvolution

Constant Groups

- PRPSEvolution

Enumerations

- enum **NormalizatioMethodes** { **Native**, **B**, **CMPLX**, **RND** }

7.7.1 Detailed Description

Collects normalizations for the input data

7.8 trunk/libPermutate/permutate.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <iterator>
#include <iostream>
#include <algorithm>
#include <array>
#include <string>
#include "../include/coords.h"
#include "../include/PRPSEvolution.h"
#include "../include/PRPSEvolutionPermutationExceptions.h"
#include "../include/PRPSEvolutionFIOExceptions.h"
#include "../include/PRPSError.h"
#include "../libPRPSSystem/prpsevolutionsystem.h"
#include "nr3/nr3.h"
#include "nr3/svd.h"
```

Include dependency graph for permutate.h: This graph shows which files directly or indirectly include this file:

Classes

- struct PRPSEvolution::Permutate::AntennaPermutations< N_MAT, T >
- struct PRPSEvolution::Permutate::permuteAntennas< N_ANTA, N_ANTPERM, T >

Namespaces

- PRPSEvolution

Constant Groups

- PRPSEvolution

Functions

- int **PRPSEvolution::Permutate::Factorial** (int x)
- template<typename Iterator >
  bool **PRPSEvolution::Permutate::next_combination** (const Iterator first, Iterator k, const Iterator last)

Variables

- const int **PRPSEvolution::Permutate::MAX_PERMUTATION_AMOUNT** = 35

7.8.1 Detailed Description

Date

2013|Jun|25

7.9 trunk/libSolve/solve.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <iterator>
#include <iostream>
#include <algorithm>
#include <array>
#include <chrono>
#include <string>
#include <random>
#include "../include/coords.h"
#include "../include/PRPSEvolution.h"
#include "../include/PRPSEvolutionSolveExceptions.h"
#include "../include/PRPSEvolutionFIOExceptions.h"
#include "../include/PRPSEvolutionGeneralExceptions.h"
#include "../include/PRPSError.h"
#include "../libPermutate/permutate.h"
#include "../libPRPSSystem/prpsevolutionsystem.h"
#include "../libNormalizer/normalizer.h"
#include "nr3/nr3.h"
#include "nr3/svd.h"
#include <EALib/ChromosomeCMA.h>
#include <SharkDefs.h>
#include <EALib/PopulationT.h>
#include <EALib/ObjectiveFunction.h>
#include "solveresult.h"
```

Include dependency graph for solve.h: This graph shows which files directly or indirectly include this file:

Classes

- struct PRPSEvolution::Solve::ProblemDimensions
- struct PRPSEvolution::Solve::Ueber9000< T >
- class PRPSEvolution::Solve::PreProcessing< N_ANTA, N_Configs, T, T_Measure >
- class PRPSEvolution::Solve::Process
- class PRPSEvolution::Solve::PostProcessing

Namespaces

- PRPSEvolution

Constant Groups

- PRPSEvolution

Enumerations

- enum **SelectBy** { **ConditionNumber**, **Random**, **AllPossible** }

- enum **ESStrategy** { **PRPSEvolution::Solve::OnePlusOne**, **PRPSEvolution::Solve::MuPlusLambda**, **PRPSEvolution::Solve::MuCommaLambda** }

Variables

- const int **PRPSEvolution::Solve::nConfigsForProcessing** = 1

7.9.1 Detailed Description

Date

2013|Jun|25

7.10 trunk/libSolve/solveresult.h File Reference

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

Classes

- struct PRPSEvolution::Solve::solveresult_t< T_Store, T_Return >

Namespaces

- PRPSEvolution

Constant Groups

- PRPSEvolution

7.10.1 Detailed Description

Date

2013|Jul|5

7.11 trunk/test/AntennaConfiguration.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <array>
#include <iostream>
#include <exception>
#include <fstream>
#include <sstream>
#include <string>
#include <chrono>
#include <future>
#include <thread>
#include "AntennaConfiguration.h"
#include "../libPermutate/permutate.h"
#include "../libPRPSSystem/prpsevolutionsystem.h"
#include "../libCalibration/calib.h"
#include "../libSolve/solve.h"
#include "../libSolve/solveresult.h"
#include "../include/PRPSEvolutionGeneralExceptions.h"
#include <EALib/ChromosomeCMA.h>
```

Include dependency graph for AntennaConfiguration.cpp:

Functions

- int **main** (int argc, char ∗argv[])

Variables

- const int **SOLUTION_AMOUNT** = 100

7.11.1 Detailed Description

This File contains the main() of the AntennaApp-Project

Index