Progeny

1.0

Generated by Doxygen 1.8.7

Wed Jul 23 2014 10:00:21

Contents

Chapter 1

Progeny Documentation

Author

Boris Bulanek National Radiation Protection Institute, Bartoskova 28, 140 00, Praha 4 boris.bulanek@suro.cz 00420 226 518 279

Date

02/19/13

1.1 About the program

Program Progeny is created in order to provide simple and fast radon progeny concentration estimation. From the knowledge of the concentration of radon progenies user can estimate the activity of filter or the signal detected from particular detector with known efficiency for given time interval of data aquisition.

As an input file is used href="http://www.sqlite.org">Sqlite database.

The program uses some additional packages:

- UI (User Interface) framework Qt
- a package for minimization GSL
- collection of C++ tools Boost
- database software Sqlite

1.2 Installation

A single executable file **progeny.exe** is created for Windows (7 or XP) users using static libraries linking.

1.2.1 Linux

All of needed packages have to be in official repositories of your distribution as all of them are open source programs. For example in case of Arch Linux distribution, you have only to write down command like:

```
sudo pacman -S cmake boost boost-libs gsl qt sqlite3 sqlitebrowser.
```

After installation you have to go to the src directory and e.g. follow these steps:

```
mkdir build
cd build
qmake(-qt4) ..
```

```
make
./progeny &
```

Compilation is made using shared libraries.

1.2.2 Sqlite database

All the information from binary file and some additional parameters can be stored in a simple database called `Sqlite`. There exist graphical tool for working with a Sqlite database called `Sqlitebrowser`. Following conventions are used for input sqlite tables: Two tables *info_1* and *info_2* are used as an input data tables. The content of tables entries is following. The content of *info_1* table are information connected with filtration through filter. The content of *info_2* table are information about filter measurement with particular detector.

- info_1
  - `filtration_time` - Time of filtration through filter in seconds
  - `air_volume` - Amount of air filtered (liters)
  - `filter_efficiency` - Efficiency to catch progenies on filter (no distinction between types of progenies)
  - `id` - Identification of measurement needed for *info_2* table with the same table entry
  - `measurement_datetime` - Voluntary information about date and time of measurement
- info_2
  - `signal` - Signal (number of events) obtained from detector measurement
  - `startTime` - Time of beginning of measurement with particular detector. The time from the end of filtration (in seconds)
  - `timeDelta` - Time of measurement with particular detector (in seconds)
  - `detector_efficiency` - Efficiency of detecting signal (events) with particular detector for particular signal..
  - `type` - Type of measurement progenies
    - 0 RnA + RnC (e.g. summary alpha)
    - 1 RnA
    - 2 RnB
    - 3 RnC
    - 4 RnB+RnC (e.g. summary beta)
  - `id` Identification of measurement with particular device for assignment to the *info_1* table entry with the same `id`

1.3 Program user guide

1.3.1 Algorithm

1.3.2 Running program

```
If you have any question, please hesitate and send me a mail to: boris.bulanek@suro.cz
```

Chapter 2

Namespace Index

2.1 Namespace List

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

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

Chapter 4

Data Structure Index

4.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 5

Namespace Documentation

5.1 Ui Namespace Reference

5.1.1 Detailed Description

Author

Boris Bulanek () National Radiation Protection Institute, Bartoskova 28, 140 00, Praha 4 boris.bulanek@suro.cz . 226 518 279

Date

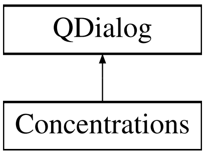
01/30/13

Chapter 6

Data Structure Documentation

6.1 Concentrations Class Reference

Inheritance diagram for Concentrations:



Public Member Functions

- **Concentrations** (QWidget ∗parent=0)
- Concentrations & **showTable** ()
- Concentrations & **showConcentrations** ()

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

- /home/boris/dokumenty/SURO/medipix/programy/Progeny_UI/src/concentrations.h

6.2 DataHandle Class Reference

Public Member Functions

- const vector< Data > & **getData** ()
- int **createDataFromTxt** (string name)
- int **openDb** (string name)
- int **getConfigurationData** (const string &confDataName)
- int **chiSquareComputeGSL** (double ∗initialParameters)
- int **createChiSquareInputData** ()
- double **computeACCorrection** ()
- DataHandle & **setDataVec** (const vector< Data > dataVec)
- const gsl_vector ∗ **getResults** ()
- const gsl_matrix ∗ **getCovMat** ()
- const QSqlDatabase & **getDb** ()
- void **setDatabasePath** (const string &databasePath)
- const string & **getDatabasePath** () const

- void **setDatabaseName** (const string &databaseName)
- const string & **getDatabaseName** () const
- const MainWindowData & **getMainWindowData** ()
- const TimeDependenceWindowData & **getTimeDependenceWindowData** ()
- DataHandle ∗ **setMainWindowData** (MainWindowData &data)
- DataHandle ∗ **setTimeDependenceWindowData** (TimeDependenceWindowData &data)

Static Public Member Functions

- static DataHandle ∗ **getInstance** ()
- static ProgenyMatrix ∗ **getProgeny** ()

Static Public Attributes

- static bool **IS_NEW**
- static double **TIME_NOT_A**

Friends

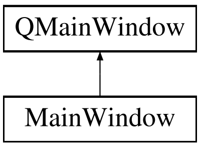
- ostream & **operator**<< (ostream &stream, const DataHandle &dataHandle)
- ostream & **operator**<< (ostream &stream, const DataHandle ∗dataHandle)

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

- /home/boris/dokumenty/SURO/medipix/programy/Progeny_UI/src/datahandle.h

6.3 MainWindow Class Reference

Inheritance diagram for MainWindow:



Public Member Functions

- **MainWindow** (QWidget ∗parent=0)
- void **setMainData** ()
- void **getMainData** ()

Static Public Attributes

- static QString **TITLE**

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

- /home/boris/dokumenty/SURO/medipix/programy/Progeny_UI/src/mainwindow.h

6.4 MainWindowData Struct Reference

Data Fields

- string **name**
- string **dbName**
- double **lambda** [3]
- double **concentrations** [3]
- double **filt_time**
- double **eff_filter**
- double **volume**

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

- /home/boris/dokumenty/SURO/medipix/programy/Progeny_UI/src/datahandle.h

6.5 ProgenyMatrix Class Reference

Public Member Functions

- **ProgenyMatrix** (const double ∗lambda, const double T=0)
- void **setCoeficients** (const double ∗lambda, const double T=0)
- vector< double > **getConcFromInfAlphas** (const double ∗par)
- vector< double > **getConcFromAlphas** (const double ∗par, const double &timeDelta)
- double **getActivityFilter** (const int &which, const double ∗conc, const double &aTime, const double &volumeFiltered, const double &timeFiltration)
- double **getNumParticles** (const int &which, const double ∗conc, const double &aTime, const double time↩Delta=0)
- double **getNumParticles** (const int &which, const double ∗conc, const double &aTime, const double time↩Delta, const double volumeFiltered, double timeFiltration=0)
- vector< double > **getProgenyWihoutFilter** (const double ∗N, const double t)
- vector< double > **getNumCreatedParticles** (const double ∗N, const double t, const double timeDelta)
- vector< double > **getNumParticles** (const double ∗N, const double timeDelta)
- void **test** ()

Friends

- ostream & **operator**<< (ostream &stream, const ProgenyMatrix &progenyMatrix)
- ostream & **operator**<< (ostream &stream, const ProgenyMatrix ∗progenyMatrix)

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

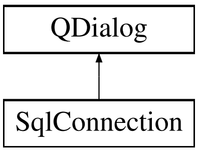
- /home/boris/dokumenty/SURO/medipix/programy/Progeny_UI/src/progenyMatrix.hh

6.6 SqlConnection Class Reference

Class for showing, selecting sql database. Class for insertion and obtaining data is SqlHandle.

```
#include <sqlconnection.h>
```

Inheritance diagram for SqlConnection:



Public Slots

- void **updateTableFromCommand** ()
- void **showTable** (const QString &t)
- void **on_actionFetchDb_triggered** ()
- void **on_actionInsertRow_triggered** ()
- void **on_actionDeleteRow_triggered** ()
- void **currentChanged** ()

Signals

- void **statusMessage** (const QString &message)

Public Member Functions

- **SqlConnection** (QWidget ∗parent=0)
- bool **isOpenDb** ()
- virtual void **accept** ()
- const QTableView ∗ **getInfo1Table** () const
- int **getSqlEntry** (const int ID)
- void **insertRow** ()
- void **deleteRow** ()
- void **updateActions** ()
- void **showDbTable** ()
- void **onlyForSave** ()
- void **onlyForOpen** ()

Static Public Attributes

- static bool **IS_SAVE**

6.6.1 Detailed Description

Class for showing, selecting sql database. Class for insertion and obtaining data is SqlHandle.

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

- /home/boris/dokumenty/SURO/medipix/programy/Progeny_UI/src/sqlconnection.h

6.7 SqlHandle Class Reference

The SqlHandle class for manipulating with sql entries.

```
#include <sqlhandle.h>
```

Public Member Functions

- void **createMainTables** ()
- void **createSecondTable** (const int ID)
- void **insertIntoMainInfoTable** (const MainWindowData &mainWindowData)
- void **deleteMeasurement** (const int ID)
- const vector< Data > **getSqlData** (const int ID)
- Data **getDataFromQuery** (const QSqlQuery &query, const QSqlRecord &record)

Static Public Member Functions

- static SqlHandle ∗ **getInstance** ()
- static void **insertIntoSecondTable** (const int ID)
- static void **insertIntoThirdTable** (const int ID, const int measurement)

6.7.1 Detailed Description

The SqlHandle class for manipulating with sql entries.

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

- /home/boris/dokumenty/SURO/medipix/programy/Progeny_UI/src/sqlhandle.h

6.8 TimeDependenceWindowData Struct Reference

Data Fields

- double **efficiencyDetection** [3]
- double **activityTime**
- double **detectedParticlesTimeRange** [2]
- double **activities** [3]
- double **detectedParticles** [3]
- double **plotXRange** [2]

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

- /home/boris/dokumenty/SURO/medipix/programy/Progeny_UI/src/datahandle.h

Index