

navicom

Generated by Doxygen 1.7.1

Wed May 27 2015 15:07:21

## Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class Hierarchy . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>2</b>
2.1	Class List . . . . .	2
<b>3</b>	<b>Class Documentation</b>	<b>2</b>
3.1	navicom::displayConfig::DisplayConfig Class Reference . . . . .	2
3.1.1	Detailed Description . . . . .	2
3.1.2	Member Function Documentation . . . . .	3
3.2	navicom::navidata::NaviAnnotations Class Reference . . . . .	3
3.2.1	Detailed Description . . . . .	3
3.3	navicom::navicom::NaviCom Class Reference . . . . .	4
3.3.1	Detailed Description . . . . .	6
3.3.2	Member Function Documentation . . . . .	6
3.4	navicom::navidata::NaviData Class Reference . . . . .	10
3.4.1	Detailed Description . . . . .	11
3.4.2	Member Function Documentation . . . . .	11
3.5	navicom::navidata::NaviSlice Class Reference . . . . .	11
3.5.1	Detailed Description . . . . .	12

## 1 Class Index

### 1.1 Class Hierarchy

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

<b>navicom::displayConfig::DisplayConfig</b>	<b>2</b>
<b>navicom::navicom::NaviCom</b>	<b>4</b>
<b>navicom::navidata::NaviData</b>	<b>10</b>
<b>navicom::navidata::NaviAnnotations</b>	<b>3</b>
<b>navicom::navidata::NaviSlice</b>	<b>11</b>

## 2 Class Index

### 2.1 Class List

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

<a href="#"><code>navicom::displayConfig::DisplayConfig</code></a> ( <a href="#"><code>DisplayConfig</code></a> class to set the color gradients configuration in <a href="#"><code>NaviCell</code></a> )	<a href="#">2</a>
<a href="#"><code>navicom::navidata::NaviAnnotations</code></a> (Enhance <a href="#"><code>NaviData</code></a> to contain annotations and associate annotations values with samples )	<a href="#">3</a>
<a href="#"><code>navicom::navicom::NaviCom</code></a> ( <a href="#"><code>NaviComm</code></a> class to handle data and display them in a standardized way on <a href="#"><code>NaviCell</code></a> maps )	<a href="#">4</a>
<a href="#"><code>navicom::navidata::NaviData</code></a> (Custom class to store the data and be able to access rows and columns by name )	<a href="#">10</a>
<a href="#"><code>navicom::navidata::NaviSlice</code></a> (A slice from a <a href="#"><code>NaviData</code></a> array )	<a href="#">11</a>

## 3 Class Documentation

### 3.1 `navicom::displayConfig::DisplayConfig` Class Reference

[`DisplayConfig`](#) class to set the color gradients configuration in [`NaviCell`](#).

#### Public Member Functions

- `def __init__`  
*Initialise a color gradient configuration.*
- `def __repr__`

#### Public Attributes

- `na_color`
- `zero_color`
- `na_size`
- `zero_size`
- `use_absolute_values`
- `step_count`
- `colors`
- `color`

#### 3.1.1 Detailed Description

[`DisplayConfig`](#) class to set the color gradients configuration in [`NaviCell`](#).

Definition at line [51](#) of file [displayConfig.py](#).

### 3.1.2 Member Function Documentation

**3.1.2.1** `def navicom::displayConfig::DisplayConfig::__init__ ( self, step_count = 3, color_gradient = [ "00FF00", FF0000, zero_color = "ffffff", na_color = "ffffff", zero_size = 0, na_size = 0, use_absolute_values = False )`

Initialise a color gradient configuration.

#### Parameters

- step\_count* number of steps for the color gradients. A step for NAs is automatically attributed.
- color\_gradient* a list of colors of length 2 or *step\_count*. If length 2 a gradient is built, if the length is *step\_count* the list is used for the colors.
- zero\_color* an hexadecimal string for the color of the zero, only visible if *step\_count* is odd

Definition at line 60 of file [displayConfig.py](#).

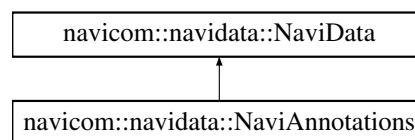
The documentation for this class was generated from the following file:

- [navicom/displayConfig.py](#)

## 3.2 navicom::navidata::NaviAnnotations Class Reference

Enhance [NaviData](#) to contain annotations and associate annotations values with samples.

Inheritance diagram for navicom::navidata::NaviAnnotations:



### Public Member Functions

- `def __init__`

### Public Attributes

- `categoriesPerAnnotation`
- `samplesPerCategory`
- `old_annots`

### 3.2.1 Detailed Description

Enhance [NaviData](#) to contain annotations and associate annotations values with samples. Also reduce continuous data with to many levels to a limited number of interval levels.

Definition at line 210 of file [navidata.py](#).

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

- navicom/navidata.py

### 3.3 navicom::navicom::NaviCom Class Reference

NaviComm class to handle data and display them in a standardized way on NaviCell maps.

#### Public Member Functions

- def [\\_\\_init\\_\\_](#)  
*Initialize a Navicell communication object.*
- def **listData**
- def **listAnnotations**
- def [\\_\\_repr\\_\\_](#)
- def **nameData**
- def **getDataName**
- def [getDataTuple](#)  
*Return tuple corresponding to the data name or tuple.*
- def [getData](#)  
*Return the NaviData entity corresponding to the data name or tuple.*
- def [loadData](#)  
*Load data from a .txt or .ncc file containing several datas, or from a .tsv, .ncd or .nca file containing data from one method.*
- def [bindNaviData](#)  
*Bind NaviData to the [NaviCom](#) object in order to use it.*
- def **defineUniformData**
- def [newProcessedData](#)  
*Update adequate arrays when processed data are generated.*
- def [quantifyMutations](#)  
*Transform the qualitative mutation datas into a quantitative one, where 1 means a mutation and 0 no mutation.*
- def [defineModules](#)  
*Defines the modules to use and which module each gene belongs to.*
- def [averageModule](#)  
*Perform module averaging for every modules for one data type.*
- def [pcaComp](#)  
*Run pca on the data and create a color matrix with the 3 principal components in the three main colors.*
- def [exportData](#)  
*Export data to NaviCell, can be processed data.*

- def [checkBrowser](#)  
*Check if the browser is opened or open it.*
- def [exportAnnotations](#)  
*Export samples annotations to NaviCell.*
- def [configureDisplay](#)  
*Changes the Color and Size Configuration for the datatable to the one precised by the user.*
- def [display](#)  
*Display data on the NaviCell map.*
- def [resetDisplay](#)  
*Reset the data and samples selections in NaviCell.*
- def [resetAnnotations](#)
- def [selectAnnotations](#)
- def [processSampleSelection](#)  
*Process a list of samples or groups to a list of samples/groups names exportable to NaviCell or to "all\_groups"/"all\_samples" for heatmap and barplot, and select the correct groups in NaviCell.*
- def [processGroupsName](#)  
*Process a group selection string and return the names of the individual groups to select and the corresponding values selected.*
- def [displayMethylome](#)  
*Display the methylation data as glyphs or heatmap on the NaviCell map, with mRNA expression of gene CNV as map staining.*
- def [displayTranscriptome](#)  
*Display one transcriptome data as map staining, with optionnaly some extra displays (samples as heatmap or barplot, mutations as glyphs, a glyph for the most highly expressed genes).*
- def [generateDistributionData](#)  
*Compute distribution of values for all genes for one type of data.*
- def [colorsOverlay](#)  
*Create a dataset where values are colors.*
- def [saveAllData](#)  
*Save all data in an .ncc file.*
- def [saveData](#)  
*Save the data in a file that can be exported to NaviCell or imported in [NaviCom](#).*

### Public Attributes

- **nv**
- **name**
- **exported\_annotations**
- **browser\_opened**
- **biotypes**
- **display\_config**
- **processings**
- **data**
- **exported\_data**
- **data\_names**
- **associated\_data**
- **annotations**
- **modules**
- **associated\_modules**
- **hsid**
- **hdid**
- **bid**

#### 3.3.1 Detailed Description

NaviComm class to handle data and display them in a standardized way on NaviCell maps.

Definition at line 28 of file [navicom.py](#).

#### 3.3.2 Member Function Documentation

**3.3.2.1** `def navicom::navicom::NaviCom::__init__ ( self, map_url =  
'https://navicell.curie.fr/navicell/maps/cellcycle/master/index.php',  
fname = "", modules_dict = "", browser_command = "firefox %s", display_config  
= DisplayConfig() )`

Initialize a Navicell communication object.

#### Parameters

*map\_url* URL of the NaviCell map

*fname* name of the data file to load

*modules\_dict* name of the module definition file (.gmt) to load

*browser\_command* command to open the browser

Definition at line 38 of file [navicom.py](#).

**3.3.2.2** `def navicom::navicom::NaviCom::colorsOverlay ( self, red = "uniform", green = "uniform", blue = "uniform", processing = "" )`

Create a dataset where values are colors.

The color is calculated according to three datasets.

#### Parameters

**red** data name or tuple (processing, method)

**green** data name or tuple (processing, method)

**blue** data name or tuple (processing, method)

Definition at line 950 of file [navicom.py](#).

**3.3.2.3** `def navicom::navicom::NaviCom::defineModules ( self, modules_dict = "" )`

Defines the modules to use and which module each gene belongs to.

#### Parameters

**modules\_dict** Either a dict indexed by module name or a file name with the description of each module (.gmt [file](#): tab delimited, first column module name, second column description, then list of entities in the module)

Definition at line 290 of file [navicom.py](#).

**3.3.2.4** `def navicom::navicom::NaviCom::display ( self, perform_list, default_samples = "all: 1.0", colors = "", module = "", reset = True )`

Display data on the NaviCell map.

**perform\_list** (list of 2-tuples): each tuple must contain the name of the data to display and the mode of display ("glyphN\_(color|size|shape)", "barplot", "heatmap" or "map\_staining"). Barplots and heatmaps cannot be displayed simultaneously. Several data types can be specified for heatmaps. Specifying "glyph" (without number) will automatically select a new glyph for each data using the same properties (shape, color or size) in glyphs (maximum of 5 glyphs).

#### Parameters

**colors** range of colors to use (NOT IMPLEMENTED YET) **default\_samples** (str or list of str) : Samples to use. Only the first sample is used for glyphs and map staining, all default\_samples from the list are used for heatmaps and barplots. Use 'all\_samples' to use all default\_samples or ['annot1:...:annotn', 'all\_groups'] to use all groups corresponding to the combinations of annot1...annotn.

Definition at line 539 of file [navicom.py](#).



**3.3.2.5** `def navicom::navicom::NaviCom::displayMethylome ( self, samples = "all: 1.0",  
processing = "raw", background = "mRNA", methylation = "glyph" )`

Display the methylation data as glyphs or heatmap on the NaviCell map, with mRNA expression of gene CNV as map staining.

#### Parameters

*background* should genes, mRNA or no data be used for the map staining

*processing* should the processed data be used

Definition at line 836 of file [navicom.py](#).

**3.3.2.6** `def navicom::navicom::NaviCom::displayTranscriptome ( self, dataName, group =  
"all: 1.0", samplesDisplay = "", samples = list (), binsNb = 10 )`

Display one transcriptome data as map staining, with optionnaly some extra displays (samples as heatmap or barplot, mutations as glyphs, a glyph for the most highly expressed genes).

- *dataName* (str or tuple): name or identifier of the data.
- *group* (str): Identifier of the group to display
- *samplesDisplay* (str): Channel where the individual samples should be displayed (heatmap or barplot)
- *samples* (list or str): list of samples to display, or a string specifying how such a list should be built ('quantiles' to get the distribution of values)
- *nbOfSamples* (int): number of individual samples to display, ignored if samples is a list

Definition at line 877 of file [navicom.py](#).

**3.3.2.7** `def navicom::navicom::NaviCom::exportData ( self, method, processing = "raw",  
name = "" )`

Export data to NaviCell, can be processed data.

#### Parameters

*method* name of the method to export

*processing* "" to export raw data, processing method to export processed data. See 'averageModule' and 'pcaComponent'

Definition at line 380 of file [navicom.py](#).

**3.3.2.8** `def navicom::navicom::NaviCom::generateDistributionData ( self, dataName, group, binsNb = 10 )`

Compute distribution of values for all genes for one type of data.

Use the same scale for all genes. The distribution is centered on 0 if it is included, so that it is easy to see if a gene is over- or under-expressed.

Definition at line 896 of file [navicom.py](#).

**3.3.2.9** `def navicom::navicom::NaviCom::loadData ( self, fname = "data/Ovarian_Serous_Cystadenocarcinoma_TCGA_Nature_2011.txt", keep_mutations_nan = False )`

Load data from a .txt or .ncc file containing several datas, or from a .tsv, .ncd or .nca file containing data from one method.

#### Parameters

*fname* name of the file from which the data should be loaded

*keep\_mutations\_nan* whether nan in mutations data should be considered as no mutation (False) or missing value (True)

Definition at line 141 of file [navicom.py](#).

**3.3.2.10** `def navicom::navicom::NaviCom::quantifyMutations ( self, method, keep_nan = False )`

Transform the qualitative mutation datas into a quantitative one, where 1 means a mutation and 0 no mutation.

#### Parameters

*keep\_nan* Should nan values be converted to 0 (no mutations) or kept as missing data

Definition at line 267 of file [navicom.py](#).

**3.3.2.11** `def navicom::navicom::NaviCom::saveAllData ( self, folder = "" )`

Save all data in an .ncc file.

Does not save the distribution nor color data.

Definition at line 1004 of file [navicom.py](#).

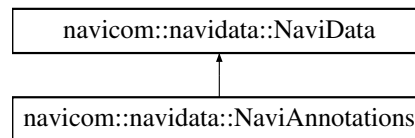
The documentation for this class was generated from the following file:

- [navicom/navicom.py](#)

### 3.4 navicom::navidata::NaviData Class Reference

Custom class to store the data and be able to access rows and columns by name.

Inheritance diagram for navicom::navidata::NaviData:



#### Public Member Functions

- def `__init__`
- def `__getitem__`
- def `__iter__`
- def `__next__`
- def `__repr__`
- def `makeData`

*Builds a string suitable for NaviCell Web Service from a python matrix of gene/sample values or a NaviCom object.*

- def `exportToNaviCell`

*Export data to a NaviCell map.*

- def `saveData`

*Save the [NaviData](#) datas in a file that can be used in NaviCell, but can also be loaded as [NaviData](#).*

#### Public Attributes

- `processing`
- `method`
- `biotype`
- `data`
- `rows`
- `rows_names`
- `columns`
- `columns_names`
- `inColumns`
- `annotations`
- `annotations_names`
- `inRows`
- `samples`
- `samples_names`
- `genes`
- `genes_names`
- `dType`
- `itermode`
- `index`
- `iter_mode`

### 3.4.1 Detailed Description

Custom class to store the data and be able to access rows and columns by name. *data* (list or array) : Values of the data to insert in the [NaviData](#) object. Must be convertible into a numpy array.

#### Parameters

- rows\_list* names of the rows (samples names)
- columns\_list* names of the columns (genes names)
- processing* name of the computer processing applied to the data
- method* name of the experimental method used to get the original ("raw") data
- dType* "data" or "annotations", whether the [NaviData](#) object contains datas or annotations (Note : this should be left to default, this is used by [NaviAnnotations](#) to change some internal variables)

Definition at line 54 of file [navidata.py](#).

### 3.4.2 Member Function Documentation

#### 3.4.2.1 def navicom::navidata::NaviData::makeData ( *self*, *hugo\_map* = "" )

Builds a string suitable for NaviCell Web Service from a python matrix of gene/sample values or a Navi-Com object.

Matrix format:

- first line is: GENE word followed by a tab separated list of sample names,
- each line begins with an gene name and must be followed by a tab separated list of gene/sample values.

Remove genes not present in *hugo\_map* if provided.

Definition at line 156 of file [navidata.py](#).

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

- [navicom/navidata.py](#)

## 3.5 navicom::navidata::NaviSlice Class Reference

A slice from a [NaviData](#) array.

#### Public Member Functions

- def [\\_\\_init\\_\\_](#)
- def [\\_\\_getitem\\_\\_](#)
- def [\\_\\_setitem\\_\\_](#)
- def [\\_\\_iter\\_\\_](#)
- def [\\_\\_repr\\_\\_](#)
- def [\\_\\_add\\_\\_](#)

### Public Attributes

- **data**
- **ids**

#### 3.5.1 Detailed Description

A slice from a [NaviData](#) array.

Definition at line 272 of file [navidata.py](#).

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

- [navicom/navidata.py](#)

## Index

- `__init__`
  - `navicom::displayConfig::DisplayConfig`, 2
  - `navicom::navicom::NaviCom`, 5
- `colorsOverlay`
  - `navicom::navicom::NaviCom`, 6
- `defineModules`
  - `navicom::navicom::NaviCom`, 6
- `display`
  - `navicom::navicom::NaviCom`, 6
- `displayMethylome`
  - `navicom::navicom::NaviCom`, 7
- `displayTranscriptome`
  - `navicom::navicom::NaviCom`, 7
- `exportData`
  - `navicom::navicom::NaviCom`, 7
- `generateDistributionData`
  - `navicom::navicom::NaviCom`, 8
- `loadData`
  - `navicom::navicom::NaviCom`, 8
- `makeData`
  - `navicom::navidata::NaviData`, 10
- `navicom::displayConfig::DisplayConfig`, 1
  - `__init__`, 2
- `navicom::navicom::NaviCom`, 3
  - `__init__`, 5
  - `colorsOverlay`, 6
  - `defineModules`, 6
  - `display`, 6
  - `displayMethylome`, 7
  - `displayTranscriptome`, 7
  - `exportData`, 7
  - `generateDistributionData`, 8
  - `loadData`, 8
  - `quantifyMutations`, 8
  - `saveAllData`, 8
- `navicom::navidata::NaviAnnotations`, 2
- `navicom::navidata::NaviData`, 9
  - `makeData`, 10
- `navicom::navidata::NaviSlice`, 11
- `quantifyMutations`
  - `navicom::navicom::NaviCom`, 8
- `saveAllData`
  - `navicom::navicom::NaviCom`, 8