

My Project

Generated by Doxygen 1.8.14

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	Namespace Documentation	7
4.1	HannesOOPProject Namespace Reference	7
4.1.1	Detailed Description	7
5	Class Documentation	9
5.1	CCP_Hannes.Analysis Class Reference	9
5.2	CCP_Hannes.Bead Class Reference	9
5.2.1	Detailed Description	10
5.2.2	Member Function Documentation	10
5.2.2.1	attachGlycans()	10
5.3	CCP_Hannes.Builder Class Reference	10
5.4	CCP_Hannes.Cytokine Class Reference	11
5.4.1	Detailed Description	11
5.5	CCP_Hannes.DecoderCell Class Reference	12
5.5.1	Detailed Description	12
5.5.2	Member Function Documentation	12

5.5.2.1	expressLectins()	12
5.6	CCP_Hannes.Glycan Class Reference	14
5.6.1	Detailed Description	14
5.7	CCP_Hannes.Lectin Class Reference	14
5.7.1	Detailed Description	15
5.8	CCP_Hannes.Simulation Class Reference	15
5.8.1	Detailed Description	15
5.9	CCP_Hannes.Sphere Class Reference	15
5.9.1	Detailed Description	16
5.10	CCP_Hannes.Well Class Reference	16
5.10.1	Detailed Description	16
5.10.2	Member Function Documentation	17
5.10.2.1	borderControl()	17
5.11	CCP_Hannes.Well_list Class Reference	17
5.11.1	Detailed Description	18
5.11.2	Member Function Documentation	18
5.11.2.1	addBead()	18
5.12	CCP_Hannes.Well_npArray Class Reference	18

Chapter 1

Namespace Index

1.1 Namespace List

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

[HannesOOPProject](#)

This project aims to simulate the binding of immune cells to artificial beads decorated with molecules that induce a biological response of the cells

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

CCP_Hannes.Analysis	9
CCP_Hannes.Builder	10
CCP_Hannes.Cytokine	11
CCP_Hannes.Glycan	14
CCP_Hannes.Lectin	14
CCP_Hannes.Simulation	15
CCP_Hannes.Sphere	15
CCP_Hannes.Bead	9
CCP_Hannes.DecoderCell	12
CCP_Hannes.Well	16
CCP_Hannes.Well_list	17
CCP_Hannes.Well_npArray	18

Chapter 3

Class Index

3.1 Class List

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

CCP_Hannes.Analysis	9
CCP_Hannes.Bead Bead is a subclass of Sphere and represents beads (made of PMMA, or acrylic glass) loaded with glycan structures	9
CCP_Hannes.Builder	10
CCP_Hannes.Cytokine Objects of class Cytokine are produced if objects of the classes Bead and DecoderCell contain- ing the correct pair of objects of the classes Glycan and Lectin , respectively, as described by the cytokine dictionary	11
CCP_Hannes.DecoderCell DecoderCell is a subclass of Sphere and represents immune cells (such as THP-1 cells) that express Lectins which bind to Glycan structures and engage in immune response by secreting cytokines	12
CCP_Hannes.Glycan Objects of the class Glycan are attached to objects of the class Bead	14
CCP_Hannes.Lectin Objects of the class Lectin are attached to objects of the class DecoderCell	14
CCP_Hannes.Simulation Dictionary is based on Geijtenbeek & Gringhuis C-type lectin receptors in the control of T helper cell differentiation Nature Reviews Immunology volume 16, pages 433–448 (2016)	15
CCP_Hannes.Sphere Sphere serves as superclass for the two spherical objects introduced below	15
CCP_Hannes.Well The class Well describes the sample, in which binding events take place	16
CCP_Hannes.Well_list Subclass of Well using built-in python lists as containers for objects of the classes Bead and DecoderCell	17
CCP_Hannes.Well_numpyArray	18

Chapter 4

Namespace Documentation

4.1 HannesOOPProject Namespace Reference

This project aims to simulate the binding of immune cells to artificial beads decorated with molecules that induce a biological response of the cells.

4.1.1 Detailed Description

This project aims to simulate the binding of immune cells to artificial beads decorated with molecules that induce a biological response of the cells.

In nature, immune cells express receptors (so-called lectins) binding to carbohydrate structures, or glycans, on the surface of body's own cells or pathogens. Depending on the nature of the glycan ligand, cells express specific cytokines, e. g. the pro-inflammatory IL-6. The idea of the biochemical is to investigate the biological response of a monocytic cell line which recombinantly expresses a defined set of lectin receptors upon binding to glycan ligands that are chemically linked to acrylic glass beads in well-defined amounts.

In the following, the translation of real-world dimensions for this project will be explained.

Cells from a typical monocytic cell line are about 15 μm in diameter, while beads typically are few μm in size. Therefore, 1 pt in this program corresponds to 10 μm in reality. The (round) basis of a well has an area of 0.36 qcm . Here, we consider a quadratic basis, hence $x = y = 0.06\text{cm}^2 = 600\text{pt}$. Typical volume is 25 μl . Hence, the height of the liquid in a well is

$$h = \frac{25\mu\text{l}}{b} = \frac{25 \cdot 10^{-8}\text{m}^3}{0.36\text{cm}^2} = 41.67 \cdot 10^{-4}\text{m} = 420\text{pt}.$$

The incubation time of the cells with beads amounts to 30 minutes, or 1800 s. In order to determine to many steps that correlates to, the Stokes-Einstein equation was utilized:

$$x^2 = \frac{k_B \cdot T}{6r\pi\eta} t = \frac{1.381 \cdot 10^{-23}\text{J/K} \cdot 298\text{K}}{6 \cdot 5\mu\text{m} \cdot \pi \cdot 0.891\text{mPa} \cdot \text{s}} = 9.8 \cdot 10^{-13} \frac{\text{m}^2}{\text{s}} \cdot t$$

At 1 s, a particle of 10 μm in aqueous solution would diffuse:

$$\sqrt{x^2} = \sqrt{4.9 \cdot 10^{-13} \frac{\text{m}^2}{\text{s}} \cdot 1\text{s}} = 7 \cdot 10^{-7}\text{m}^2.$$

Therefore, a typical experiment of 30 minutes corresponds to $1800\text{s} \cdot 0.7\mu\text{m/s} = 1260\mu\text{m} = 126$ steps.

Chapter 5

Class Documentation

5.1 CCP_Hannes.Analysis Class Reference

Public Member Functions

- `def __init__(self)`
- `def countCytokines(self, simulationResults)`
- `def plotCytokines(self, simulationResults)`

Public Attributes

- `cytokineAmount`
- `cytokineNames`

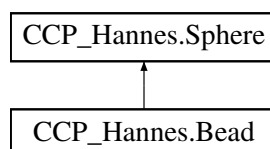
The documentation for this class was generated from the following file:

- `CCP_Hannes.py`

5.2 CCP_Hannes.Bead Class Reference

[Bead](#) is a subclass of [Sphere](#) and represents beads (made of PMMA, or acrylic glass) loaded with glycan structures.

Inheritance diagram for `CCP_Hannes.Bead`:



Public Member Functions

- `def __init__ (self, args)`
- `def attachGlycans (self, glycan_names_list, glycan_types_list, density_percentage)`
Method to attach [Glycan](#) objects to [Bead](#) objects.

Public Attributes

- `glycan_density`
- `glycan`

5.2.1 Detailed Description

[Bead](#) is a subclass of [Sphere](#) and represents beads (made of PMMA, or acrylic glass) loaded with glycan structures.

5.2.2 Member Function Documentation

5.2.2.1 attachGlycans()

```
def CCP_Hannes.Bead.attachGlycans (
    self,
    glycan_names_list,
    glycan_types_list,
    density_percentage )
```

Method to attach [Glycan](#) objects to [Bead](#) objects.

Every [Bead](#) objects contains one type of [Glycan](#) with a given density. Parameters `glycan_names_list` and `glycan_types_list` are handed over to the constructor of [Glycan](#).

Parameters

in	<code>glycan_names_list</code>	List of names of all Glycans to be added to the Bead . List may contain any number of member, including 0
in	<code>see</code>	documentation for class Glycan .
in	<code>see</code>	documentation for class Glycan .

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

- `CCP_Hannes.py`

5.3 CCP_Hannes.Builder Class Reference

Public Member Functions

- `def __init__ (self, x, y, z)`

- def **buildWell** (self, container_type, n_beads, n_cells)
- def **buildBead** (self, i, ID, glycan_name_string, glycan_type_string, density_percentage)
- def **buildDecoderCell** (self, i, ID, lectin_name_string, density_percentage)

Public Attributes

- **x**
- **y**
- **z**
- **well**

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

- CCP_Hannes.py

5.4 CCP_Hannes.Cytokine Class Reference

Objects of class [Cytokine](#) are produced if objects of the classes [Bead](#) and [DecoderCell](#) containing the correct pair of objects of the classes [Glycan](#) and [Lectin](#), respectively, as described by the cytokine dictionary.

Public Member Functions

- def **__init__** (self, name, coordinates_list)

Public Attributes

- **name**
- **coordinates**

5.4.1 Detailed Description

Objects of class [Cytokine](#) are produced if objects of the classes [Bead](#) and [DecoderCell](#) containing the correct pair of objects of the classes [Glycan](#) and [Lectin](#), respectively, as described by the cytokine dictionary.

Parameters

in	<i>name</i>	Name of the cytokine.
in	<i>coordinates_list</i>	List containing the three values for x, y, and z, where the Cytokine was produced.

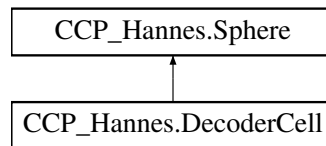
The documentation for this class was generated from the following file:

- CCP_Hannes.py

5.5 CCP_Hannes.DecoderCell Class Reference

[DecoderCell](#) is a subclass of [Sphere](#) and represents immune cells (such as THP-1 cells) that express Lectins which bind to [Glycan](#) structures and engage in immune response by secreting cytokines.

Inheritance diagram for CCP_Hannes.DecoderCell:



Public Member Functions

- `def __init__(self, args)`
- `def expressLectins(self, lectin_list, density_percentage)`
Method to attach [Lectin](#) objects to [DecoderCell](#) objects.

Public Attributes

- `lectin_density`
- `lectin`

5.5.1 Detailed Description

[DecoderCell](#) is a subclass of [Sphere](#) and represents immune cells (such as THP-1 cells) that express Lectins which bind to [Glycan](#) structures and engage in immune response by secreting cytokines.

Right	Center	Left
10	10	10
1000	1000	1000

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

5.5.2 Member Function Documentation

5.5.2.1 `expressLectins()`

```

def CCP_Hannes.DecoderCell.expressLectins (
    self,
    lectin_list,
    density_percentage )
  
```


Method to attach [Lectin](#) objects to [DecoderCell](#) objects.

Every [DecoderCell](#) objects contains one type of [Lectin](#) with a given density. Parameter `lectins_list` is handed over to the constructor of [Lectin](#).

Parameters

in	<i>lectin_list</i>	see documentation for class Lectin .
----	--------------------	--

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

- CCP_Hannes.py

5.6 CCP_Hannes.Glycan Class Reference

Objects of the class [Glycan](#) are attached to objects of the class [Bead](#).

Public Member Functions

- `def __init__ (self, glycan_names_list, glycan_types_list)`

Public Attributes

- **name**
- **type**

5.6.1 Detailed Description

Objects of the class [Glycan](#) are attached to objects of the class [Bead](#).

They represent glycan structures which are divided into different types.

Parameters

in	<i>glycan_names_list</i>	List of Strings containing the "real" names of Glycan structures to be added to the Bead . List may contain any number of members, including 0
in	<i>glycan_types_list</i>	List of Strings containing the respective types of Glycan structures. These types determine the outcome of the interaction between Bead and DecoderCell as specified in the cytokine dictionary. If the this list does not contain the same number of member as <i>glycan_names_list</i> , the program terminates.

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

- CCP_Hannes.py

5.7 CCP_Hannes.Lectin Class Reference

Objects of the class [Lectin](#) are attached to objects of the class [DecoderCell](#).

Public Member Functions

- `def __init__(self, lectin_list)`

Public Attributes

- `name`

5.7.1 Detailed Description

Objects of the class [Lectin](#) are attached to objects of the class [DecoderCell](#).

They represent lectin receptors which recognize certain sets of glycans on other cells, beads, viruses a.s.o.

Parameters

in	<i>lectin_list</i>	List of Strings containing the names of Lectin receptors to be added to the DecoderCell . These names appear in the dictionary. List may contain any number of members, including 0.
----	--------------------	--

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

- `CCP_Hannes.py`

5.8 CCP_Hannes.Simulation Class Reference

Dictionary is based on Geijtenbeek & Gringhuis C-type lectin receptors in the control of T helper cell differentiation Nature Reviews Immunology volume 16, pages 433–448 (2016)

5.8.1 Detailed Description

Dictionary is based on Geijtenbeek & Gringhuis C-type lectin receptors in the control of T helper cell differentiation Nature Reviews Immunology volume 16, pages 433–448 (2016)

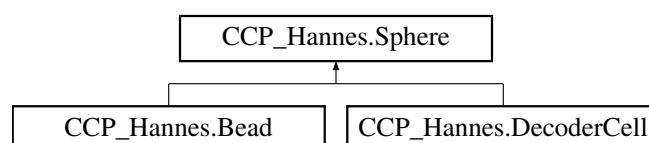
The documentation for this class was generated from the following file:

- `CCP_Hannes.py`

5.9 CCP_Hannes.Sphere Class Reference

[Sphere](#) serves as superclass for the two spherical objects introduced below.

Inheritance diagram for `CCP_Hannes.Sphere`:



Public Member Functions

- `def __init__ (self, ID_string)`

Public Attributes

- **ID**
- **coordinates**

5.9.1 Detailed Description

[Sphere](#) serves as superclass for the two spherical objects introduced below.

Parameters

in	<i>ID</i>	Identifier
	<i>coordinates</i>	Empty. Will be filled when added to Well .

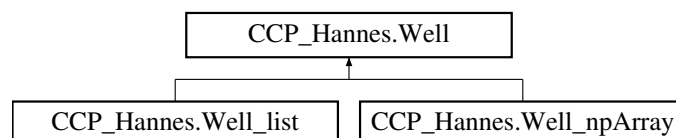
The documentation for this class was generated from the following file:

- CCP_Hannes.py

5.10 CCP_Hannes.Well Class Reference

The class [Well](#) describes the sample, in which binding events take place.

Inheritance diagram for CCP_Hannes.Well:



Public Member Functions

- `def __init__ (self, x, y, z)`
- `def borderControl (self, coordinates, dx, dy, dz)`
The method borderControl ensures that objects' coordinates don't exceed the well's size.

5.10.1 Detailed Description

The class [Well](#) describes the sample, in which binding events take place.

It serves as a superclass for two subclasses of which one uses NumPy Arrays as a container for objects of the classes [Bead](#) and [DecoderCell](#) while the other one uses built-in Python lists for this purpose.

Parameters

in	x,y,z	Size of the well.
----	-------	-------------------

5.10.2 Member Function Documentation

5.10.2.1 borderControl()

```
def CCP_Hannes.Well.borderControl (
    self,
    coordinates,
    dx,
    dy,
    dz )
```

The method borderControl ensures that objects' coordinates don't exceed the well's size.

If a step in randomWalk would lead to a forbidden value, the respective step value will be set to 0 and the object won't move this turn as if repelled from the well's borders.

Parameters

in	<i>coordinates</i>	The object's coordinates before the move.
in	<i>dx,dy,dz</i>	The randomly chosen next steps.
out	<i>dx,dy,dz</i>	The revised next steps

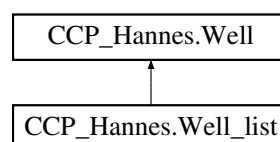
The documentation for this class was generated from the following file:

- CCP_Hannes.py

5.11 CCP_Hannes.Well_list Class Reference

Subclass of [Well](#) using built-in python lists as containers for objects of the classes [Bead](#) and [DecoderCell](#).

Inheritance diagram for CCP_Hannes.Well_list:



Public Member Functions

- def **__init__** (self, x, y, z, n_beads, n_cells)
- def **addBead** (self, i, bead, glycan_name_string, glycan_type_string, density_percentage)
Method to add objects of class [Bead](#) to the bead list.
- def **addDecoderCell** (self, i, decoderCell, lectin_name_string, lectin_type_string)

Public Attributes

- **size**
- **beads**
- **decoderCells**

5.11.1 Detailed Description

Subclass of [Well](#) using built-in python lists as containers for objects of the classes [Bead](#) and [DecoderCell](#).

5.11.2 Member Function Documentation

5.11.2.1 addBead()

```
def CCP_Hannes.Well_list.addBead (
    self,
    i,
    bead,
    glycan_name_string,
    glycan_type_string,
    density_percentage )
```

Method to add objects of class [Bead](#) to the bead list.

Coordinates of the [Bead](#) are randomly chosen between 0 and size of the [Well](#) in the respective dimension.

Parameters

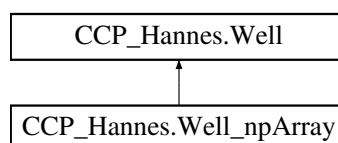
in	<i>i</i>	Not used here, but necessary for convenient switching between container types.
in	<i>bead</i>	Object of class Bead to be added.
in		

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

- CCP_Hannes.py

5.12 CCP_Hannes.Well_npArray Class Reference

Inheritance diagram for CCP_Hannes.Well_npArray:



Public Member Functions

- `def __init__ (self, x, y, z, n_beads, n_cells)`
- `def addBead (self, i, bead, glycan_name_string, glycan_type_string, density_percentage)`
- `def addDecoderCell (self, i, decoderCell, lectin_name_string, lectin_type_string)`

Public Attributes

- `size`
- `beads`
- `decoderCells`

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

- `CCP_Hannes.py`

