## Contents

1. **AIMS2 Hierarchical Index** ................................. 1
   1.1 AIMS2 Class Hierarchy .................................. 1

2. **AIMS2 Data Structure Index** ............................ 3
   2.1 AIMS2 Data Structures ................................... 3

3. **AIMS2 Data Structure Documentation** .................. 5
   3.1 AIMS2 Class Reference .................................. 5
   3.2 CAbObject Class Reference ............................. 7
   3.3 CAgObject Class Reference ............................. 9
   3.4 CAPC Class Reference ................................. 12
   3.5 CBCell Class Reference ................................ 14
   3.6 CCCreateEnv Class Reference .......................... 17
   3.7 CCCreateNDevEnv Class Reference ..................... 19
   3.8 CDevEnv Class Reference ................................ 21
   3.9 CDDiscrete3DSpace Class Reference ................... 23
   3.10 CDDiscrete3DSpace::cmpPos Struct Reference ........ 28
   3.11 CImmuneCell Class Reference ......................... 29
   3.12 CLivingObject Class Reference ....................... 32
   3.13 CObjectCreator Class Reference ..................... 34
   3.14 CPepptide Class Reference ........................... 37
   3.15 CPosition Class Reference ............................. 40
   3.16 CReactEnv Class Reference ............................ 42
   3.17 CReactEnvNTissue Class Reference ................... 44
   3.18 CStatistics Class Reference .......................... 49
   3.19 CTCCell Class Reference ................................ 51
   3.20 CTCtissue Class Reference ........................... 54
Chapter 1

AIMS2 Hierarchical Index

1.1 AIMS2 Class Hierarchy

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

AIMS2 ................................................................. 5
CDiscrete3DSpace ................................................. 23
  CCreateNDevEnv .............................................. 19
    CCreateEnv ................................................ 17
    CDevEnv ................................................... 21
  CReactEnvNTissue ........................................... 44
    CReactEnv ................................................ 42
    CTissue ................................................... 54
CDiscrete3DSpace::cmpPos ........................................ 28
CLivingObject ................................................... 32
  CABObject ................................................... 7
  CAgObject .................................................... 9
    CImmuneCell .............................................. 29
  CAPC .......................................................... 12
  CBCell ........................................................ 14
  CTCell ........................................................ 51
COBJECTCreator .................................................. 34
CPeptide .......................................................... 37
CPosition ........................................................ 40
CStatistics ....................................................... 49
Chapter 2

AIMS2 Data Structure Index

2.1 AIMS2 Data Structures

Here are the data structures with brief descriptions:

AIMS2 (The Simulator class) .......................................................... 5
CABObject ......................................................................................... 7
CAGObject ......................................................................................... 9
CAPC (CAPC models antigen presenting cells) ..................................... 12
CBCell ............................................................................................... 14
CCreateEnv (Class modelling the bone marrow) ..................................... 17
CCreateNDevEnv .................................................................................. 19
CDevEnv (Class modelling the thymus) ..................................................... 21
CDiscrete3DSpace (Abstract class modelling a discrete three dimensional space) ............................................ 23
CDiscrete3DSpace::cmpPos (Comparison method for two CPosition(p. 40) objects needed for STL map) .................................................................................. 28
CImmuneCell ....................................................................................... 29
CLivingObject ..................................................................................... 32
CObjectCreator .................................................................................. 34
CPeptide ............................................................................................. 37
CPosition (A class that indicates a position in the 3D space) .................... 40
CReactEnv (Class modelling the peripheral lymphoid organs) .................. 42
CReactEnvNTissue ............................................................................... 44
CStatistics .......................................................................................... 49
CTCell ................................................................................................. 51
CTissue (Class modelling the body tissue) .............................................. 54
Chapter 3

AIMS2 Data Structure 
Documentation

3.1 AIMS2 Class Reference

The Simulator class.
#include <aims2.h>

Collaboration diagram for AIMS2:

Public Member Functions

- AIMS2()
- virtual ~AIMS2()
- int getNumberSelfSequences()
- int getNumberMHCmolecules()
- void initSelf()
- void runSimulation ()

### 3.1.1 Detailed Description

The Simulator class.
AIMS2 means "Another IMmune System Simulator".

**Author:**
Christin Seifert

### 3.1.2 Constructor & Destructor Documentation

#### 3.1.2.1 AIMS2::AIMS2 ()

Constructor for AIMS2 objects.

#### 3.1.2.2 virtual AIMS2::~AIMS2 () [virtual]

Destructor for AIMS2 objects.

### 3.1.3 Member Function Documentation

#### 3.1.3.1 int AIMS2::getNumberMHCmolecules ()

Get number of different MHC molecules.

**Returns:**
number of different MHC molecules in the simulator

#### 3.1.3.2 int AIMS2::getNumberSelfSequences ()

Get number of self sequences.

**Returns:**
number of different self sequences in the simulator

#### 3.1.3.3 void AIMS2::initSelf ()

Do the initial creation of self sequences and MHC molecules. These self sequences and MHC molecules are of Class **CPeptide** (p.37) and created randomly.

#### 3.1.3.4 void AIMS2::runSimulation ()

This is the main loop of the system. All environments are started from within this loop.

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

- aims2.h
3.2 CAbObject Class Reference

```
#include <cabobject.h>
```

Inheritance diagram for CAbObject:

```
+--- CLivingObject
   |   CAbObject
       +--- CPeptide
```

Collaboration diagram for CAbObject:

```
+--- CLivingObject
   |   CPeptide
       +--- m_paratope
           +--- CAbObject
```

### Public Member Functions
- `CAbObject ()`
- `CAbObject (CPeptide paratope)`
- `virtual ~CAbObject ()`
- `CPeptide getParatope ()`
- `void setParatope (CPeptide sequence)`
- `virtual void print ()`

### Protected Attributes
- `CPeptide m_paratope`
- `int lengthParatope`

### 3.2.1 Detailed Description

This is a antibody object (can exists without a cell), is emitted by B plasma cells. Models: Immunglobuline.

**Author:**
Christin Seifert

### 3.2.2 Constructor & Destructor Documentation

#### 3.2.2.1 CAbObject::CAbObject ()

Constructor for CAbObject objects.
3.2.2.2 CAbObject::CAbObject (CPeptide paratope)

Constructor for CAbObject objects.

Parameters:
paratope the paratope of the antibody

3.2.2.3 virtual CAbObject::~CAbObject () [virtual]

Destructor for CAbObject objects.

3.2.3 Member Function Documentation

3.2.3.1 CPeptide CAbObject::getParatope ()

Get the binding part of the antibody - the paratope modelled by m_paratope.

Returns:
the paratope of the antibody

3.2.3.2 virtual void CAbObject::print () [virtual]

Printing state of the object (values of all attributes).
Implements CLivingObject (p. 33).

3.2.3.3 void CAbObject::setParatope (CPeptide sequence)

Set m_paratope to sequence.

Parameters:
sequence the sequence the paratope should be set to

3.2.4 Field Documentation

3.2.4.1 int CAbObject::lengthParatope [protected]

lengthParatope stores the length of the paratope. Is the same for all objects and should coincide with m_paratope->length. BCR, TCR and antigen epitope must be of same length.

3.2.4.2 CPeptide CAbObject::m_paratope [protected]

m_paratope is a member variable of class CPeptide(p. 37). This models the binding part of the antibody.

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

• cabobject.h
3.3 CAgObject Class Reference

#include <cagobject.h>

Inheritance diagram for CAgObject:

```
+------------------
|                  |
|   CAgObject      |
|                  |
+------------------+
|                  |
|   CLivingObject  |
|                  |
+------------------+
```

Collaboration diagram for CAgObject:

```
+------------------
|                  |
|   CAgObject      |
|                  |
+------------------+
|                  |
|   CPeptide       |
|                  |
|                 / m_epitope
|                  |
+------------------+
```

Public Member Functions

- CAgObject()
- CAgObject(CPeptide epitope)
- CAgObject(CAgObject &ag)
- virtual ~CAgObject()
- CPeptide getEpitope()
- void setEpitope(CPeptide sequence)
- virtual void print()

Data Fields

- bool isSelfAg

Protected Attributes

- CPeptide m_epitope
- int lengthEpitope

3.3.1 Detailed Description

This is a antigen object (can exist without cells). Models: Pathogens

Author:

Christin Seifert
3.3.2 Constructor & Destructor Documentation

3.3.2.1 CAgObject::CAgObject ()
Constructor for CAgObject objects.

3.3.2.2 CAgObject::CAgObject (CPeptide epitope)
Constructor for CAgObject objects.

3.3.2.3 CAgObject::CAgObject (CAgObject & ag)
Copy-Constructor for CAgObject objects.
Parameters:
  ag the antigen that should be copied

3.3.2.4 virtual CAgObject::~CAgObject () [virtual]
Destructor for CAgObject objects.

3.3.3 Member Function Documentation

3.3.3.1 CPeptide CAgObject::getEpitope ()
Get the binding part of the antigen - the epitope modelled by m_epitope.
Returns:
  the epitope of the antigen

3.3.3.2 virtual void CAgObject::print () [virtual]
Printing state of the object (values of all attributes).
Implements CLivingObject (p. 33).

3.3.3.3 void CAgObject::setEpitope (CPeptide sequence)
Set m_epitope to sequence.
Parameters:
  sequence the sequence the epitope should be set to

3.3.4 Field Documentation

3.3.4.1 bool CAgObject::isSelfAg
isSelfAg indicates whether the antigen belongs to the body itself and should therefore not trigger a reaction of the immune system.
3.3.4.2 int CAgObject::lengthEpitope [protected]

lengthEpitope holds the length of the epitope. Is the same for all objects and should coincide with m_epitope->length. Paratope of antibodies, BCR and TCR must be of the same length must be of the same length.

3.3.4.3 CPeptide CAgObject::m_epitope [protected]

m_epitope is a member variable of class CPeptide(p.37). This modells the binding part of the antigen.

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

- cagobject.h
3.4 CAPC Class Reference

CAPC models antigen presenting cells.

#include <capc.h>

Inheritance diagram for CAPC:Collaboration diagram for CAPC:

![Collaboration diagram for CAPC]

Public Member Functions

- CAPC ()
- virtual ~CAPC ()
- void setMHCMolecule (CPeptide peptide)
- CPeptide getMHCMolecule ()
- CPeptide getPresentedPeptide ()
- bool isPeptidePresented ()
- void eatAndPresent (CPeptide peptide)
- void print ()

3.4.1 Detailed Description

CAPC models antigen presenting cells.

This cell class unites the function of APC (macrophages, dentritic cells) and cells that do macroinocytosis (dentritic cell). Phagocytosis is not possible, characteristic bacterial bitstrings must be known.

Author:
Christin Seifert

3.4.2 Constructor & Destructor Documentation

3.4.2.1 CAPC::CAPC ()

Constructor for CAPC objects.

3.4.2.2 virtual CAPC::~CAPC () [virtual]

Destructor for CAPC objects.
3.4.3 Member Function Documentation

3.4.3.1 void CAPC::eatAndPresent (CPeptide peptide)

This models the makropinocytosis (affiliation of extra cellular material not depending on any
specific receptor) and the presenting of the material (peptide) on surface. It’s the main function
of an APC (antigen presenting cell).

Parameters:
peptide the peptide the APC affiliates and afterwards presents on its surface

3.4.3.2 CPeptide CAPC::getMHC Molecule ()

Get the MHC molecule which is on surface of the APC.

Returns:
the MHC molecule on surface of the APC

3.4.3.3 CPeptide CAPC::getPresentedPeptide ()

Get the presented peptide. Make sure to test the bool function isPeptidePresented() (p. 13)
first. If no peptide is presented the function returns NULL.

Returns:
the peptide the cell is presenting on its surface

3.4.3.4 bool CAPC::isPeptidePresented ()

Indicates whether the APC is currently presenting a peptide.

Returns:
true if the APC is presenting a peptide, false otherwise

3.4.3.5 void CAPC::print () [virtual]

Printing state of the object (values of all attributes).
Implements CImmuneCell (p. 30).

3.4.3.6 void CAPC::setMHC Molecule (CPeptide peptide)

Set the MHC molecule which is on surface of the APC.

Parameters:
peptide the sequence the MHC molecule of the APC should be set to

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

- capc.h
3.5 CBCell Class Reference

#include <cbcell.h>

Inheritance diagram for CBCell:

Collaboration diagram for CBCell:

Public Member Functions

- CBCell ()
- virtual ~CBCell ()
- CBCell (CBCell &bccell)
- void setMHC (Molecule CPeptide peptide)
- CPeptide getMHC (Molecule peptide)
- void setBCR (CPeptide bcr)
- CPeptide getBCR ()
- CPeptide getPresentedPeptide ()
- double getAffinityBCRtoPeptide (CPeptide peptide)
- int getRemainingSomHypSteps ()
- void decrementRemainingSomHypSteps ()
- void mutateBCR ()
- void eatAndPresent (CPeptide peptide)
- virtual void print ()

3.5.1 Detailed Description

B Cell which is responsible for humorale immunity.

Author:
Christin Seifert

3.5.2 Constructor & Destructor Documentation

3.5.2.1 CBCell::CBCell ()

Constructor for CBCell objects.

3.5.2.2 virtual CBCell::~CBCell () [virtual]

Destructor for CBCell objects.
### 3.5.2.3 CBCell::CBCell (CBCell & bcell)

Copy-Constructor for CBCell objects.

**Parameters:**
- `bcell` the bcell that should be copied

### 3.5.3 Member Function Documentation

#### 3.5.3.1 void CBCell::decrRemainingSomHypSteps ()

Decrease the remaining somatic hypermutation steps of the B cell.

#### 3.5.3.2 void CBCell::eatAndPresent (CPeptide peptide)

This models the makropinocytosis (affiliation of extra cellular material not depending on any specific receptor) and the presenting of the material (peptide) on surface. It's the main function of an bcell (antigene presenting cell).

**Parameters:**
- `peptide` the peptide the bcell affiliates and afterwards presents on its surface

#### 3.5.3.3 double CBCell::getAffinityBCRToPeptide (CPeptide peptide)

Get the affinity value of BCR to the given peptide. Only if the value extends the threshold AFFINITY_REACTION_THRESHOLD the B cell is able to bind the presented peptide to the receptor.

**Parameters:**
- `peptide` the sequence of which the affinity to the BCR should be calculated

**Returns:**
- the affinity value in range from 0 to 1

#### 3.5.3.4 CPeptide CBCell::getBCR ()

Get the sequence of the BCR.

**Returns:**
- the bcr

#### 3.5.3.5 CPeptide CBCell::getMHCmolecule ()

Get the MHC molecule which is on surface of the B cell.

**Returns:**
- the MHC molecule
3.5.3.6 CPeptide CBCell::getPresentedPeptide ()

Get the presented peptide. Make sure to check the state of the cell first, to know whether the cell is presenting a peptide. Functions return NULL of no peptide is presented.

Returns:
the peptide the cell is presenting on its surface

3.5.3.7 int CBCell::getRemainingSomaticSteps ()

Get the remaining steps the cell is doing somatic hypermutation (cell is in state SOM_HYP).

Returns:
the remaining steps of somatic hypermutation

3.5.3.8 void CBCell::mutateBCR ()

Take the BCR and mutate it randomly (point mutations only). Mutation rate is set in file globals.h(p.??) with BCR_MUTATION_Rate. This defines each which bit in average a mutation takes place. For instance, if BCR_MUTATION_Rate equals 4 each 4th bit will be mutated in average.

3.5.3.9 virtual void CBCell::print () [virtual]

Printing state of the object (values of all attributes).
Implements CImmuneCell (p. 30).

3.5.3.10 void CBCell::setBCR (CPeptide bcr)

Set the BCR which is on surface of the B cell.

Parameters:
bcr the sequence the BCR should be set to

3.5.3.11 void CBCell::setMHCMolecule (CPeptide peptide)

Set the MHC receptor structure which is on surface of the B cell.

Parameters:
peptide the sequence the MHC molecule of the bcell should be set to

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

• cbcell.h
3.6 CCreateEnv Class Reference

Class modelling the bone marrow.
#include <ccreateenv.h>
Inheritance diagram for CCreateEnv: Collaboration diagram for CCreateEnv:

Public Member Functions

- CCreateEnv (CObjectCreator *oc, CTissue *ts)
- virtual ~CCreateEnv ()
- void run ()

Friends

- class CStatistics

3.6.1 Detailed Description

Class modelling the bone marrow.

Environment class for creation of cells. Here only self-antigens and immature b cells move around.

Author:
Christin Seifert

3.6.2 Constructor & Destructor Documentation

3.6.2.1 CCreateEnv::CCreateEnv (CObjectCreator * oc, CTissue * ts)

Constructor for CCreateEnv objects. This one is needed to pass the CObjectCreator(p.34) object from AIMS2(p.5) class. CObjectCreator(p.34) object is allowed to exist only once because of the stored self sequences.

Parameters:
- oc the CObjectCreator(p.34) object used to create all needed CLivingObjects (as b cells and self-antigens).
- ts the CTissue(p.54) object (the only one of this class within the simulator) passed by AIMS2(p.5) class used to move mature b cells to.

3.6.2.2 virtual CCreateEnv::~CCreateEnv () [virtual]

Destructor for CCreateEnv objects.

3.6.3 Member Function Documentation

3.6.3.1 void CCreateEnv::run () [virtual]

Run one simulation step in creation environment, consisting of
• creation of cells – createCells() (p. 24)
• negative selection of b cells – negSelectionOfBCells() including removal of self-reacting b cells – selectOutIfReact()
• decreasing the life time of all objects and killing the death ones – doLifeCycle() (p. 20)
• do one step of maturation of BCells – matureCells()
• move mature cells to reaction environment – moveMatureCellsToEnv() (p. 20)

Implements CDiscrete3DSpace (p. 26).

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

• createenv.h
3.7 CCreateNDevEnv Class Reference

#include <createndevenv.h>

Inheritance diagram for CCreateNDevEnv:

```
   CDiskrete3DSpace
    |__________________________
    |                          |
    |                          | CCreateNDevEnv
    |                          |<------------------------
    |                          |  CCreateEnv  CDevEnv
```

Collaboration diagram for CCreateNDevEnv:

```
   CDiskrete3DSpace
    |__________________________
    |                          |
    |                          | CCreateNDevEnv
```

Public Member Functions

- CCreateNDevEnv ()
- ~CCreateNDevEnv ()

Protected Member Functions

- virtual void doLifeCycle ()
- virtual void moveMatureCellsToEnv ()=0

Protected Attributes

- vector< CPosition > m_matureCells

3.7.1 Detailed Description

Superclass of CCreateEnv (p.17) and CDevEnv (p.21). Combines attributes and methods of environments that create cells.

Author:
Christin Seifert

3.7.2 Constructor & Destructor Documentation

3.7.2.1 CCreateNDevEnv::CCreateNDevEnv ()

Constructor for CCreateNDevEnv objects.
3.7.2.2 CCreateNDevEnv::~CCreateNDevEnv ()

Destructor for CCreateNDevEnv objects.

3.7.3 Member Function Documentation

3.7.3.1 virtual void CCreateNDevEnv::doLifeCycle () [protected, virtual]

Decrease the life time of all objects and kill the ones with expired life time. Mature cells if needed and remember mature cells in vector m_matureCells.
Implements CDiscrete3Dspace (p.25).

3.7.3.2 virtual void CCreateNDevEnv::moveMatureCellsToEnv () [protected, pure virtual]

Move mature cells to either CTissue(p.54) or CReactEnv(p.42).

3.7.4 Field Documentation

3.7.4.1 vector<CPosition> CCreateNDevEnv::m_matureCells [protected]

Vector of all positions where mature cells were found.
The documentation for this class was generated from the following file:

* ccreatendevenv.h
3.8 CDevEnv Class Reference

Class modelling the thymus.

#include <cdevenv.h>

Inheritance diagram for CDevEnv: Collaboration diagram for CDevEnv:

Public Member Functions

- CDevEnv (CObjectCreator *oc, CReactEnv *re)
- virtual ~CDevEnv ()
- void run ()

Friends

- class CStatistics

3.8.1 Detailed Description

Class modelling the thymus.

Environment for development of T cells. It is the place of positive and negative selection of T cells. It is ignored that T stemm cells are created in the bone marrow, they are created within this environment – only the development of the receptors is important. Here only self-aggs move around.

Author:
Christin Seifert

3.8.2 Constructor & Destructor Documentation

3.8.2.1 CDevEnv::CDevEnv (CObjectCreator * oc, CReactEnv * re)

Constructor for CDevEnv objects. This one is needed to pass the CObjectCreator(p.34) object from AIMS2(p.5) class. CObjectCreator(p.34) object is allowed to exist only once because of the stored self sequences.

Parameters:
- oc the CObjectCreator(p.34) object used to create all needed CLivingObjects (as t cells and self-antigens)
- re the CReactEnv(p.42) object (the only one of this class within the simulator) passed by AIMS2(p.5) class used to move mature t cells to

3.8.2.2 virtual CDevEnv::~CDevEnv () [virtual]

Destructor for CDevEnv objects.
3.8.3 Member Function Documentation

3.8.3.1 void CDevEnv::run () [virtual]

Run one simulation step in creation environment, consisting of

- creation of cells – createCells() (p. 24)
- positive and negative selection – posAndNegSelectionOfTCells()
- decrease the life time of all objects and kill the death ones, mature cells – doLifeCycle() (p. 20)
- move mature cells to reaction environment – moveMatureCellsToEnv() (p. 20)
- move objects round in space – doMovements() (p. 25)

Implements CDiscrete3DSpace (p. 26).

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

- cdelevn.h
3.9 CDiscr3DSpace Class Reference

Abstract class modelling a discrete three dimensional space.

#include <cdiscr3dSPACE.h>

Inheritance diagram for CDiscr3DSpace:

```
CDiscr3DSpace
  CCreateNDevEnv
  CReactEnvNTissue
  CCreateEnv
  CDevEnv
  CReactEnv
  CTissue
```

Public Member Functions

- CDiscr3DSpace()
- virtual ~CDiscr3DSpace()
- virtual void run() = 0
- virtual CLivingObject* getObjectAtPosition(const CPosition pos)
- virtual bool isPositionFree(CPosition pos)
- virtual int getNumFreePlaces()
- virtual void addObjectAtPosition(CLivingObject* object, CPosition pos)
- virtual void printObjects()
- virtual void deleteObject(CLivingObject* object)
- virtual void deleteObjectAtPosition(CPosition pos)
- virtual void removeObjectAtPosition(CPosition pos)
- virtual CPosition getRandomFreePosition()

Protected Types

- typedef map<CPosition, CLivingObject*, cmpPos> Pos2LivObjMap

Protected Member Functions

- virtual void createCells() = 0
- virtual CPosition getRandomPosition()
- virtual void doMovements()
- virtual void doLifeCycle() = 0

Protected Attributes

- int xMax
- int xMin
- int yMax
- int yMin
- int zMax
- int zMin
- int freePlaces
- Pos2LivObjMap m_objects

Generated on Sat Feb 14 15:59:37 2004 for AIM32 by Doxygen
3.9.1 Detailed Description

Abstract class modelling a discrete three dimensional space.

A class providing functionality of a discrete 3D space with objects of class `CLivingObject` (p. 32) (and inherited classes) in it. The space has a defined dimension. Objects are handled in an stl map.

Author:
Christin Seifert

3.9.2 Member TypeDef Documentation

3.9.2.1 `typedef map<CPosition, CLivingObject*, cmpPos> CDiscrete3DSpace::Pos2LivObjMap [protected]`

A stl map with keytype `CPosition` (p. 40) and value type Pointer to `CLivingObject` (p. 32) using `cmpPos` (p. 28) as ordering operator.

3.9.3 Constructor & Destructor Documentation

3.9.3.1 `CDiscrete3DSpace::CDiscrete3DSpace ()`
Constructor for `CDiscrete3DSpace` objects.

3.9.3.2 `virtual CDiscrete3DSpace::~CDiscrete3DSpace () [virtual]`
Destructor for `CDiscrete3DSpace` objects.

3.9.4 Member Function Documentation

3.9.4.1 `virtual void CDiscrete3DSpace::addObjectAtPosition (CLivingObject * object, CPosition pos) [virtual]`
Add a given object to a given position. Be sure to call `isPositionFree` first.

Parameters:
- `object` object that should be added in the space.
- `pos` position to which the object should be added in the space.

3.9.4.2 `virtual void CDiscrete3DSpace::createCells () [protected, pure virtual]`
Create cells randomly.

3.9.4.3 `virtual void CDiscrete3DSpace::deleteObject (CLivingObject * object) [virtual]`
Delete a given object from space (m_objects). The object is erased from map and also delete with delete - method (destructed). This function has to search all objects in m_objects linearly, use `deleteObjectAtPosition()` (p. 25) instead if possible.
Parameters:
  *object* object that should be deleted

### 3.9.4.4 virtual void CDisc3DSpace::deleteObjectAtPosition(CPosition pos) [virtual]

Deletes object at a given position. The object is erased from map and also delete with delete - method (destroyed). This is the fast method, because access to object via its position can be done in O(log(number of objects)).

Parameters:
  *pos* position from where the object should be deleted

### 3.9.4.5 virtual void CDisc3DSpace::doLifeCycle() [protected, pure virtual]

Decrease the life time of all objects and kill the ones with expired life time. Mature cells if needed. Implemented in CCreatENDevEnv (p.20), and CReactEnvNTissue (p.46).

### 3.9.4.6 virtual void CDisc3DSpace::doMovements() [protected, virtual]

Do random movements of all objects in m_objects. Object move to a free neighbour position (if there is one).

### 3.9.4.7 virtual int CDisc3DSpace::getNumFreePlaces() [virtual]

Returns number of free places in the space

Returns:
  number of free places

### 3.9.4.8 virtual CLivingObject* CDisc3DSpace::getObjectAtPosition(const CPosition pos) [virtual]

Get the object at the given position. Make sure to call isPositionFree(p.26) first to test whether the object at the given position exists.

Parameters:
  *pos* 3D Position the object should be got from

Returns:
  object at given position

### 3.9.4.9 virtual CPosition CDisc3DSpace::getRandomFreePosition() [virtual]

Get a random position that is within the bounds of the space and free. Make sure to call get-NumberFreePlaces() first to be sure there is any free position. Otherwise the return value is undefined.

Returns:
  free random Position within the bounds of the space
3.9.4.10 virtual CPosition CDiverse3D::getRandomPosition () [protected, virtual]

Get a random position that is within the bounds of the space.

Returns:
random Position within the bounds of the space

3.9.4.11 virtual bool CDiverse3D::isPositionFree (CPosition pos) [virtual]

Returns true if the position pos is free in the space, else false.

Parameters:
    pos 3D Position that should be checked if free

Returns:
    TRUE if position is free, FALSE otherwise

3.9.4.12 virtual void CDiverse3D::printObjects () [virtual]

Print all objects contained in the space.

3.9.4.13 virtual void CDiverse3D::removeObjectAtPosition (CPosition pos) [virtual]

Removes object at a given position. The object is erased from map but not deleted (destructor).

Parameters:
    pos position from where the object should be deleted

3.9.4.14 virtual void CDiverse3D::run () [pure virtual]

Do one simulation step.

Implemented in CCreateEnv (p.17), CDevEnv (p.22), CReactEnv (p.43), and CTissue (p.55).

3.9.5 Field Documentation

3.9.5.1 int CDiverse3D::freePlaces [protected]

Number of free places in space.

3.9.5.2 Pos2LivObjMap CDiverse3D::m_objects [protected]

Map of all objects of class CLivingObject(p.32) contained in the space.
3.9.5.3 int CDiscr3D::xMax [protected]
Right x bound of space.

3.9.5.4 int CDiscr3D::xMin [protected]
Left x bound of space (including).

3.9.5.5 int CDiscr3D::yMax [protected]
Right y bound of space.

3.9.5.6 int CDiscr3D::yMin [protected]
Left y bound of space (including).

3.9.5.7 int CDiscr3D::zMax [protected]
Right z bound of space.

3.9.5.8 int CDiscr3D::zMin [protected]
Left z bound of space (including).
The documentation for this class was generated from the following file:

- cdiscr3dspace.h
3.10 CDIscrete3DSpace::cmpPos Struct Reference

Comparison method for two CPosition(p. 40) objects needed for STL map.
#include <cdiscrete3dspace.h>

Public Member Functions

- bool operator() (CPosition p1, CPosition p2) const

3.10.1 Detailed Description

Comparison method for two CPosition(p. 40) objects needed for STL map.

A comparison method for ordering objects in map, returns true if key1 (CPosition(p. 40) object) is less than key2 (CPosition(p. 40) object), false otherwise. A Position is less than another if the x-value is less, it’s greater if the x-value is greater. If both x-values are equal the y-values (and z-values) are compared. This is needed for defining an stl map with keytype CPosition(p. 40) because a map is internally handled as B-tree.

Parameters:

- p1 first position that should be compared
- p2 second position that should be compared

Author:

Christin Seifert

3.10.2 Member Function Documentation

3.10.2.1 bool CDIscrete3DSpace::cmpPos::operator() (CPosition p1, CPosition p2) const [inline]

The comparison operator.

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

- cdiscrete3dspace.h
3.11 CImmuneCell Class Reference

#include <cimmuneCell.h>

Inheritance diagram for CImmuneCell:

```
CLivingObject

CImmuneCell

CAPC CBCell CTCell
```

Collaboration diagram for CImmuneCell:

```
CLivingObject

CImmuneCell
```

Public Member Functions

- CImmuneCell ()
- virtual ~CImmuneCell ()
- void decrMaturityLevel ()
- bool getIsMature ()
- int getState ()
- void setState (int state)
- virtual void print ()=0

Protected Attributes

- int maturityLevel
- int currentState

3.11.1 Detailed Description

An abstract class of an immune cell.

Author:
Christin Seifert

3.11.2 Constructor & Destructor Documentation

3.11.2.1 CImmuneCell::CImmuneCell ()

Constructor for CImmuneCell objects
3.11.2.2 virtual CImmuneCell::~CImmuneCell () [virtual]

Destructor for CImmuneCell objects

3.11.3 Member Function Documentation

3.11.3.1 void CImmuneCell::decrMaturityLevel ()

Decreases the value of maturityLevel (zero means that cell is mature).

3.11.3.2 bool CImmuneCell::getIsMature ()

Returns whether the cell is mature.

Returns:
  TRUE if cell is mature, FALSE otherwise

3.11.3.3 int CImmuneCell::getState ()

Get the current state of the cell. For a list of available states see file globals.h(p.??).

Returns:
  the state of the cell

3.11.3.4 virtual void CImmuneCell::print () [pure virtual]

Printing state of the object (values of all attributes).

Implements CLivingObject (p.33).

Implemented in CAPC (p.13), CBCell (p.16), and CTCell (p.52).

3.11.3.5 void CImmuneCell::setState (int state)

Sets the current state of the cell. For a list of all available states see file globals.h(p.??).

Parameters:
  state the state the cell state should be set to

3.11.4 Field Documentation

3.11.4.1 int CImmuneCell::currentState [protected]

Indicates the current state of the T cell. For a list of available states see file globals.h(p.??).

3.11.4.2 int CImmuneCell::maturityLevel [protected]

level of maturity (zero means that cell is mature)

The documentation for this class was generated from the following file:
- cimmuneCell.h
3.12 CLivingObject Class Reference

#include <clivingobject.h>

Inheritance diagram for CLivingObject:

![Inheritance Diagram](image)

Public Member Functions

- CLivingObject()
- virtual ~CLivingObject()
- void setLifetime(int lifetime)
- bool isLifeTimeExpired()
- void incrRemainingLifetime(int value)
- void decrRemainingLifetime()
- virtual void print() = 0

Protected Attributes

- int remainingLifeTime

3.12.1 Detailed Description

Modells a living object. All objects used in the subclasses of CDiskrete3DExpSpace (p. 23) are derived from this class.

Author:

Christin Seifert

3.12.2 Constructor & Destructor Documentation

3.12.2.1 CLivingObject::CLivingObject()

Constructor for CLivingObject objects

3.12.2.2 virtual CLivingObject::~CLivingObject() [virtual]

Destructor for CLivingObject objects
3.12.3 Member Function Documentation

3.12.3.1 void CLivingObject::decRemainingLifetime ()

Decreases the remaining lifetime of a living object by 1.

3.12.3.2 void CLivingObject::incrRemainingLifetime (int value)

Increase the remaining lifetime of the living object. Can be used to model the influence of life-
elongating cytokines.

Parameters:
  value the value the remaining life time should be increased by

3.12.3.3 bool CLivingObject::isLifeTimeExpired ()

isLifeTimeExpired returns whether the lifetime is expired.

Returns:
  TRUE if lifetime is expired, FALSE otherwise

3.12.3.4 virtual void CLivingObject::print () [pure virtual]

Printing state of the object (values of all attributes).
Implemented in CAbObject (p.8), CAgObject (p.10), CAPC (p.13), CBCCell (p.16),
CImmuneCell (p.30), and CTCell (p.52).

3.12.3.5 void CLivingObject::setLifetime (int lifetime)

Set lifetime of the object. Can be used at the beginning of the simulation (initial lifetime), but
also to model the influence of life-elongating cytokines.

Parameters:
  lifetime the value the lifetime of the object should be set to

3.12.4 Field Documentation

3.12.4.1 int CLivingObject::remainingLifeTime [protected]

The remaining lifetime of the living object, which is decreased in each simulation step.
The documentation for this class was generated from the following file:

- clivingobject.h
3.13 CObjectCreator Class Reference

#include <cobjectcreator.h>

Public Member Functions

- CObjectCreator()
- virtual ~CObjectCreator()
- void initCreateSelfSequences(int number, int length)
- void initCreateMHC(int number, int length)
- CAgObject* getSelfAgObject()
- CAgObject* getAgObject()
- CAbObject* getAbObject()
- CBCell* getBCellObject()
- CCell* getTCellObject()
- CAPC* getAPCObject()
- CAPC* getSelfAPCObject()
- void printSelfSequences()
- void printMHCObjects()

3.13.1 Detailed Description

Class for creating new sequences, because lifetime of all objects expires. Has static lists of all existing self sequences and of all existing mhc-molecules.

Author:
  Christin Seifert

3.13.2 Constructor & Destructor Documentation

3.13.2.1 CObjectCreator::CObjectCreator()

Constructor for CObjectCreator objects

3.13.2.2 virtual CObjectCreator::~CObjectCreator() [virtual]

Destructor for CObjectCreator objects

3.13.3 Member Function Documentation

3.13.3.1 CAbObject* CObjectCreator::getAbObject()

Get an antibody object. Used by B cells, that set paratope to their (mutated) BCR.
3.13.3.2 CAgObject* CObjectCreator::getAgObject ()

Get an random (maybe self or nonself) antigen for use in CTissue(p.54). Be sure not to use this method to place antigens in CCreateEnv(p.17) and CDevEnv(p.21), therefore getSelfAgObject() (p.35) should be used.

Returns:
   a ag object

3.13.3.3 CAPC* CObjectCreator::getAPCObject ()

Get a random APC object. Used in CTissue(p.54).

Returns:
   apc object

3.13.3.4 CBCell* CObjectCreator::getBCellObject ()

Get an random bcell object. Used in CCreateEnv(p.17).

Returns:
   bcell object

3.13.3.5 CAgObject* CObjectCreator::getSelfAgObject ()

Gets a self antigene object for use in a CDevEnv(p.21) and CCreateEnv(p.17).

Returns:
   a self ag object

3.13.3.6 CAPC* CObjectCreator::getSelfAPCObject ()

Get a random self APC object. Used in CDevEnv(p.21).

Returns:
   self apc object

3.13.3.7 CTCel* CObjectCreator::getTCellObject ()

Get a random T cell object. Used in CDevEnv(p.21).

Returns:
   tcell object
3.13.3.8  void CObjectCreator::initCreateMHC (int number, int length)

Do the initial creation of (self) MHC molecules. This is done randomly.

Parameters:
   number  number of different MHC molecules to create
   length  length of the molecule sequence

3.13.3.9  void CObjectCreator::initCreateSelfSequences (int number, int length)

Do the initial creation if self sequences. This is done randomly.

Parameters:
   number  number of different self sequences to create
   length  length of the sequence

3.13.3.10 void CObjectCreator::printMHCmolecules ()

Prints all current (self) MHC molecules to stdout. Used for testing purposes.

3.13.3.11 void CObjectCreator::printSelfSequences ()

Prints all self sequences to stdout. Used for testing purposes.

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

   • objectcreator.h
3.14 CPeptide Class Reference

#include <cpeptide.h>

Public Member Functions

- CPeptide (int length)
- CPeptide ()
- CPeptide (CPeptide &peptide)
- CPeptide & operator= (const CPeptide &peptide)
- virtual ~CPeptide ()
- int getLength ()
- vector< bool > getSequence ()
- void setSequence (vector< bool > seq)
- vector< bool > getSequence () const
- void printSequence ()

Protected Member Functions

- void createRandomSequence (int length)

Protected Attributes

- int length
- vector< bool > sequence

3.14.1 Detailed Description

CPeptide is a general class that should model peptides. It consists of a sequence of something, currently something is bit. All receptors use CSequence as members, they don’t care which elements the sequence consists of, so it’s easy to change elements to amino acids, for instance.

Author:
Christin Seifert

3.14.2 Constructor & Destructor Documentation

3.14.2.1 CPeptide::CPeptide (int length)

Constructor for CPeptide objects.

Parameters:

length the length of the peptide

3.14.2.2 CPeptide::CPeptide ()

Constructor for CPeptide objects.
3.14.2.3  CPeptide::CPeptide (CPeptide & peptide)

Copy-Constructor for CPeptide objects.

**Parameters:**
- *peptide* the peptide that should be copied

**Returns:**
- the new peptide

3.14.2.4  virtual CPeptide::~CPeptide () [virtual]

Destructor for CPeptide objects

### 3.14.3  Member Function Documentation

3.14.3.1  void CPeptide::createRandomSequence (int length)  [protected]

Create random sequence of boolean values.

**Parameters:**
- *length* number of bits the sequence should contain

3.14.3.2  int CPeptide::getLength ()

Get the length of the sequence.

**Returns:**
- length the length of the sequence

3.14.3.3  vector<bool> CPeptide::getSequence () const

Get the sequence, as an vector of bool.

**Returns:**
- the sequence of the peptide

3.14.3.4  vector<bool> CPeptide::getSequence ()

Get the sequence, as an vector of bool.

**Returns:**
- the sequence
3.14.3.5  CPeptide& CPeptide::operator= (const CPeptide & peptide)

Assignment operator for CPeptide objects.

Parameters:
    peptide the peptide that should be assigned

Returns:
    the new peptide

3.14.3.6  void CPeptide::printSequence ()

Prints the current sequence to stdout. Used for testing purposes.

3.14.3.7  void CPeptide::setSequence (vector< bool > seq)

Set the sequence, as an vector of bool.

Parameters:
    seq the sequence the peptide’s sequence should be set to

3.14.4  Field Documentation

3.14.4.1  int CPeptide::length  [protected]

The length of the sequence.

3.14.4.2  vector< bool > CPeptide::sequence  [protected]

The sequence of elements - models sequence of proteins - is modeled as sequence of boolean values
(== Bitvector).

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

  * cpeptide.h
3.15 CPosition Class Reference

A class that indicates a position in the 3D space.
#include <cposition.h>

Public Member Functions

- CPosition ()
- virtual ~CPosition ()
- void print ()
- void debugPrint ()

Data Fields

- int x
- int y
- int z

3.15.1 Detailed Description

A class that indicates a position in the 3D space.
A CPosition object knows its position in space and is able to print itself.

Author:
Christin Seifert

3.15.2 Constructor & Destructor Documentation

3.15.2.1 CPosition::CPosition ()

Constructor for CPosition objects.

3.15.2.2 virtual CPosition::~CPosition () [virtual]

Destructor for CPosition objects.

3.15.3 Member Function Documentation

3.15.3.1 void CPosition::debugPrint ()

Printing a Position to log file if DEBUG_ON is defined.

Note:
Only for debugging!

3.15.3.2 void CPosition::print ()

Printing a Position to stdout.
3.15.4 Field Documentation

3.15.4.1 int CPosition::x

x value

3.15.4.2 int CPosition::y

y value

3.15.4.3 int CPosition::z

z value

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

- cposition.h
3.16 CReactEnv Class Reference

Class modelling the peripheral lymphoid organs.

#include <creactenv.h>

Inheritance diagram for CReactEnv:

![Inheritance Diagram]

Public Member Functions

- CReactEnv()
- virtual ~CReactEnv()
- void SetTissueEnv (CDiscrete3DSpace *ts)
- void run()

Friends

- class CStatistics

3.16.1 Detailed Description

Class modelling the peripheral lymphoid organs.

Environment where the antigen - antibody reaction takes place.

Author:

Christin Seifert

3.16.2 Constructor & Destructor Documentation

3.16.2.1 CReactEnv::CReactEnv()

Constructor for CReactEnv objects.

3.16.2.2 virtual CReactEnv::~CReactEnv() [virtual]

Destructor for CReactEnv objects.
3.16.3 Member Function Documentation

3.16.3.1 void CReactEnv::run () [virtual]

Run one simulation step in creation environment, consisting of

- creation of cells – createCells()(p. 24);
- check interactions — checkInteraction()(p. 45): interesting cases are
  - CBCell(p. 14)<->CTCell(p. 51)
  - CAPC(p. 12)<->CTCell(p. 51)
  - CAgObject(p. 9)<->CAbObject(p. 7)
- decrease the life time of all objects and kill the death ones, mature cells – doLife-
  Cycle()(p. 46)
- move active cells away – moveCellsToEnv()(p. 47);
- do random movement of all cells – doMovements()(p. 25);

Implements CDiscrete3D Space (p. 26).

3.16.3.2 void CReactEnv::SetTissueEnv (CDiscrete3D Space * ts)

Set the m_tissue to the only existing CTissue(p. 54) object within the simulator.

Parameters:
  ts the tissue object

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

- creactenv.h
3.17 CReactEnvNTissue Class Reference

#include <creactenvntissue.h>

Inheritance diagram for CReactEnvNTissue:

```
CDiscrete3DSpace

CReactEnvNTissue

CReactEnv  CTissue
```

Collaboration diagram for CReactEnvNTissue:

```
CDiscrete3DSpace

CReactEnvNTissue
```

Public Member Functions

- CReactEnvNTissue()
- ~CReactEnvNTissue()

Protected Member Functions

- void doAgAPCRaction (CAgObject *ag, CAPC *apc)
- bool doAgAbReaction (CAgObject *ag, CAbObject *ab, CPosition agPos, CPosition abPos)
- void doAgBCellReaction (CAgObject *ag, CCell *bcell)
- void doTCellAPCRaction (CTCell *tcell, CAPC *apc, CPosition apcPos)
- void doTCellBCellReaction (CTCell *tcell, CCell *bcell)
- void checkInteraction()
- void doLifeCycle()
- virtual void moveCellsToEnv( )=0

Protected Attributes

- vector< CPosition > m_TCells
- vector< CPosition > m_Ags
- int numB
- int numB_agdetected
- int numB_somhyp
- int numB_active
- int numT
3.17 CReactEnvNTissue Class Reference

- int numT_active
- int numAPC
- int numAPC_presenting
- int numAg
- int numAg_self
- int numAb
- int numAg_death
- int numAb_death

3.17.1 Detailed Description

Superclass of C\text{Tissue}(p. 54) and C\text{ReactEnv}(p. 42). Combines attributes and features of environments where the immune reaction takes place.

\textbf{Author:}
Christin Seifert

3.17.2 Constructor & Destructor Documentation

3.17.2.1 \texttt{CReactEnvNTissue::CReactEnvNTissue ()}

Constructor for CReactEnvNTissue objects.

3.17.2.2 \texttt{CReactEnvNTissue::~CReactEnvNTissue ()}

Destructor for CReactEnvNTissue objects.

3.17.3 Member Function Documentation

3.17.3.1 void \texttt{CReactEnvNTissue::checkInteraction () [protected]}

Checks interaction between all cells in environment. Possible interactions are

- doAgAPCReaction
- doAgAbReaction
- doAgBCellReaction
- doTCellAPCReaction
- doTCellBCellReaction

3.17.3.2 bool \texttt{CReactEnvNTissue::doAgAbReaction (CAgObject* ag, CAbObject* ab, CPosition agPos, CPosition abPos) [protected]}

Do the antigen-antibody reaction. The antigen will be remove if the affinity between ag and ab exceeds the threshold AFFINITY\_REACTION\_THRESHOLD.

\textbf{Parameters:}

\texttt{ag} the antigen object
the antibody object

\textit{agPos} the position of the antigen

\textit{abPos} the position of the antibody

Returns:  
TRUE if reaction took place and the antibody is removed, FALSE otherwise

3.17.3.3 \textbf{void CReactEnvNTissue::doAgAPCReaction (CAgObject \* ag, CAPC \* apc)} [protected]

Do the reaction of neighbouring ag and APC. This is makropinocytosis. APC takes the epitope of the ag and presents in on its surface.

\textbf{Parameters:}

\textit{ag} the antigen object

\textit{apc} the APC

3.17.3.4 \textbf{void CReactEnvNTissue::doAgBCellReaction (CAgObject \* ag, CBCell \* bcell)} [protected]

Do the ag - B cell reaction. The B cell will affiliate the ag if the affinity epitope to BCR is beyond the AFFINITY\_REACTION\_THRESHOLD.

\textbf{Parameters:}

\textit{ag} the antigen object

\textit{bcell} the bcell object

3.17.3.5 \textbf{void CReactEnvNTissue::doLifeCycle ()} [protected, virtual]

Decrease the life time of all objects and kill the ones with expired life time.

Implements \texttt{CDiscrete3DSpace} (p. 25).

3.17.3.6 \textbf{void CReactEnvNTissue::doTCellAPCReaction (CTCell \* tcell, CAPC \* apc, CPosition apcPos)} [protected]

Does T cell - APC reaction. If affinity to t cell receptors (TCR and CD8 receptor) is beyond the thresholds AFFINITY\_REACTION\_THRESHOLD and MHC\_REACTION\_THRESHOLD, a naive T cell would become active and a active T cell kills the apc.

\textbf{Parameters:}

\textit{tcell} the T cell object

\textit{apc} the APC object

\textit{apcPos} the 3D position of the APC
3.17.3.7 void CReactEnvNTissue::doTCellBCellReaction (CTCell * tcell, CBCell * bcell) [protected]

Does T cell - B cell reaction. If affinity to t cell receptors (TCR and CD8 receptor) is beyond the thresholds AFFINITY_REACTION_THRESHOLD and MHC_REACTION_THRESHOLD, a naive T cell would become active and an active T cell can send a costimulating signal to the B cell, that already got first signal.

Parameters:
   tcell the tcell object
   bcell the bcell object

3.17.3.8 virtual void CReactEnvNTissue::moveCellsToEnv () [protected, pure virtual]

Move cells to another environment.

3.17.4 Field Documentation

3.17.4.1 vector<CPosition> CReactEnvNTissue::m_Ags [protected]

Contains all antigene objects.

3.17.4.2 vector<CPosition> CReactEnvNTissue::m_TCells [protected]

Contains all T cells

3.17.4.3 int CReactEnvNTissue::numAb [protected]

Number of Ab for last cycle.

3.17.4.4 int CReactEnvNTissue::numAb_death [protected]

Number of deatch Abs (died by Ag-Ab reaction) for last cycle.

3.17.4.5 int CReactEnvNTissue::numAg [protected]

Number of Agfor last cycle.

3.17.4.6 int CReactEnvNTissue::numAg_death [protected]

Number of death Ags (died by Ag-Ab reaction) for last cycle.

3.17.4.7 int CReactEnvNTissue::numAg_self [protected]

Number of self Agfor last cycle.
3.17.4.8  int CReactEnvNTissue::numAPC  [protected]
Number of AP C for last cycle.

3.17.4.9  int CReactEnvNTissue::numAPC_preseting  [protected]
Number of presenting APC for last cycle.

3.17.4.10 int CReactEnvNTissue::numB  [protected]
Number of B cells for last cycle.

3.17.4.11 int CReactEnvNTissue::numB_active  [protected]
Number of active B cells for last cycle.

3.17.4.12 int CReactEnvNTissue::numB_agdetected  [protected]
Number of ag detected B cells for last cycle.

3.17.4.13 int CReactEnvNTissue::numB_somhyp  [protected]
Number of somhyp B cells for last cycle.

3.17.4.14 int CReactEnvNTissue::numT  [protected]
Number of T cells for last cycle.

3.17.4.15 int CReactEnvNTissue::numT_active  [protected]
Number of active T cells for last cycle.
The documentation for this class was generated from the following file:

* creactenvntissue.h
3.18 CStatistics Class Reference

#include <cstatistics.h>

Collaboration diagram for CStatistics:

Public Member Functions

- CStatistics (CCreateEnv *ce, CDevEnv *de, CReactEnv *re, CTissue *t)
- virtual ~CStatistics ()
- void run ()
- void printVecToDatFile (string vectName)
- void printToDatFiles ()

3.18.1 Detailed Description

Class collecting statistical information, such as the amount of different cell types for each simulation step.

Author:
Christin Seifert

3.18.2 Constructor & Destructor Documentation

3.18.2.1 CStatistics::CStatistics (CCreateEnv *ce, CDevEnv *de, CReactEnv *re, CTissue *t)

Constructor for CStatistics objects.

Parameters:

- ce the CCreateEnv(p.17) object (the only one of this class within the simulator) passed by AIMS2(p.5) class
- de the CDevEnv(p.21) object (the only one of this class within the simulator) passed by AIMS2(p.5) class

Generated on Sat Feb 14 15:59:37 2004 for AIMS2 by Doxygen
re the CReactEnv (p.42) object (the only one of this class within the simulator) passed by AIMS2 (p.5) class

t the CTissue (p.54) object (the only one of this class within the simulator) passed by AIMS2 (p.5) class

3.18.2.2 virtual CStatistics::~CStatistics () [virtual]

Destructor for CStatistics objects.

3.18.3 Member Function Documentation

3.18.3.1 void CStatistics::printToDatFiles ()

Prints all stored data series to files.

3.18.3.2 void CStatistics::printVecToDatFile (string vectName)

Prints locally stored data (series) to file, that can be read with gnuplot.

Parameters:

vectName the name of the time serie that should be printet (filename will be the same).

3.18.3.3 void CStatistics::run ()

Collects all data from the environments and stores them locally.
The documentation for this class was generated from the following file:

- cstatistics.h
3.19 CTCell Class Reference

```
#include <ctcell.h>
```

Inheritance diagram for CTCell:

```
CLivingObject

CImmuneCell

CTCell
```

Collaboration diagram for CTCell:

```
CLivingObject

CImmuneCell

CPeptide

CTCell
```

Public Member Functions

- CTCell()
- CTCell (CTCell &ctcell)
- virtual ~CTCell()
- double getAffinityTCRToPeptide (CPeptide peptide)
- double getAffinityCD8ToPeptide (CPeptide peptide)
- void setIsAbleToBindMHC()
- bool getIsAbleToBindMHC()
- void print()

3.19.1 Detailed Description

T cell, which is responsible for cellular immunity.

**Author:**
Christin Seifert

3.19.2 Constructor & Destructor Documentation

3.19.2.1 CTCell::CTCell ()

Constructor for CTCell objects.
3.19.2.2 CTCell::CTCell (CTCell & tcell)
Copy-Constructor for CBCell(p.14) objects.

Parameters:
  tcell the tcell that should be copied

3.19.2.3 virtual CTCell::~CTCell () [virtual]
Destructor for CTCell objects

3.19.3 Member Function Documentation

3.19.3.1 double CTCell::getAffinityCD8ToPeptide (CPeptide peptide)
Get the affinity value of CD8 receptor to the given peptide. Only if the value extends the threshold MHC_REACTION_THRESHOLD the T cell is able to bind the presented MHC molecule to the receptor.

Parameters:
  peptide the sequence of which the affinity to the CD8 receptor should be calculated

Returns:
  the affinity value in range from 0 to 1

3.19.3.2 double CTCell::getAffinityTCRToPeptide (CPeptide peptide)
Get the affinity value of TCR to the given peptide. Only if the value extends the threshold AFFINITY_REACTION_THRESHOLD the T cell is able to bind the presented Ag to the receptor.

Parameters:
  peptide the sequence of which the affinity to the TCR should be calculated

Returns:
  the affinity value in range from 0 to 1

3.19.3.3 bool CTCell::getIsAbleToBindMHC ()
Get the value of the attribute isAbleToBindMHC.

3.19.3.4 void CTCell::print () [virtual]
Printing state of the object (values of all attributes).
Implements CImmuneCell (p.30).
3.19.3.5 void CTCell::setIsAbleToBindMHC ()

Sets the attribute isAbleToBindMHC to TRUE.

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

- ctcell.h
3.20 CTissue Class Reference

Class modelling the body tissue.
#include <ctissue.h>

Inheritance diagram for CTissue: Collaboration diagram for CTissue:

![Diagram showing inheritance and collaboration relationships between CTissue, CDiscrete3DSpace, CReactEnv, and CObjectCreator]

Public Member Functions

- CTissue (CObjectCreator *oc, CReactEnv *re)
- virtual ~CTissue()
- void run()

Friends

- class CStatistics

3.20.1 Detailed Description

Class modelling the body tissue.

Environment that models the normal tissue of the body. Here the infections will be detected by cells of the innate immune system and these are carried to CReactEnv(p.42), where the ag:ab reaction takes place. Models: normal body tissue.

Author:
Christin Seifert

3.20.2 Constructor & Destructor Documentation

3.20.2.1 CTissue::CTissue (CObjectCreator * oc, CReactEnv * re)

Constructor for CTissue objects. This one is needed to pass the CObjectCreator(p.34) object from AIMS2(p.5) class. CObjectCreator(p.34) object is allowed to exist only once because of the stored self sequen ces.

Parameters:
- oc the CObjectCreator(p.34) object used to create all needed CLivingObjects (as patho genes)
- re the CReactEnv(p.42) environment, where active immune cells are moved to
3.20.2.2 virtual CTissue::~CTissue () [virtual]

Destructor for CTissue objects.

3.20.3 Member Function Documentation

3.20.3.1 void CTissue::run () [virtual]

Run one simulation step in creation environment, consisting of

- creation of cells – createCells()(p. 24);
- check interactions — checkInteraction()(p. 45): interesting cases are
  - CBCCell(p. 14) <-> CTCell(p. 51)
  - CAPC(p. 12) <-> CTCell(p. 51)
  - CAgObject(p. 9) <-> CAbObject(p. 7)
- decrease the life time of all objects and kill the death ones, mature cells – doLifeCycle()(p. 46)
- move active cells away – moveCellsToEnv()(p. 47);
- do random movement of all cells – doMovements()(p. 25);

Implements CDIscrete3DSpace (p. 26).

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

- ctissue.h
Index

~AIMS2
  AIMS2, 6
~CAPC
  CAPC, 12
~CAbObject
  CAbObject, 8
~CAgObject
  CAgObject, 10
~CBCell
  CBCell, 14
~CCreateEnv
  CCreateEnv, 17
~CCreateNDevEnv
  CCreateNDevEnv, 19
~CDevEnv
  CDevEnv, 21
~CDiscrete3DSpace
  CDiscrete3DSpace, 24
~CImmuneCell
  CImmuneCell, 29
~CLivingObject
  CLivingObject, 32
~CObjectCreator
  CObjectCreator, 34
~CPeptide
  CPeptide, 38
~CPosition
  CPosition, 40
~CReactEnv
  CReactEnv, 42
~CReactEnvNTissue
  CReactEnvNTissue, 45
~CStatistics
  CStatistics, 50
~CTCell
  CTCell, 52
~CTissue
  CTissue, 54

addObjectAtPosition
  CDiscrete3DSpace, 24
AIMS2, 5
  ~AIMS2, 6
AIMS2, 6
  getNumberSelfSequences, 6
initSelf, 6
runSimulation, 6

CAbObject, 7
  CAbObject, 7
CAbObject
  ~CAbObject, 8
  CAbObject, 7
getParatope, 8
lengthParatope, 8
m_paratope, 8
print, 8
setParatope, 8

CAgObject, 9
  CAgObject, 10
CAgObject
  ~CAgObject, 10
  CAgObject, 10
getEpitope, 10
isSelfAg, 10
lengthEpitope, 10
m_epitope, 11
print, 10
setEpitope, 10

CAPC, 12
  ~CAPC, 12
  CAPC, 12
eatAndPresent, 13
getMHC, 13
getPresentedPeptide, 13
isPeptidePresented, 13
print, 13
setMHC, 13

CBCell, 14
  ~CBCell, 14
  CBCell, 14
decrRemainingSomHypSteps, 15
eatAndPresent, 15
getAffinityBCRToPeptide, 15
getBCR, 15
getMHC, 15
getPresentedPeptide, 15
getRemainingSomHypSteps, 16
mutateBCR, 16
print, 40
x, 41
y, 41
z, 41
CReactEnv, 42
CReactEnv, 42
CReactEnv
~CReactEnv, 42
CReactEnv, 42
run, 43
SetTissueEnv, 43
CReactEnvNTissue, 44
CReactEnvNTissue, 45
CReactEnvNTissue
~CReactEnvNTissue, 45
checkInteraction, 45
CReactEnvNTissue, 45
doAgAbReaction, 45
doAgAPCReaction, 46
doAgBCellReaction, 46
doLifeCycle, 46
doTCelAPCReaction, 46
doTCelBCellReaction, 46
m_Ags, 47
m_TCells, 47
moveCellsToEnv, 47
numAb, 47
numAb_death, 47
numAg, 47
numAg_death, 47
numAg_self, 47
numAPC, 47
numAPC_presenting, 48
numB, 48
numB_active, 48
numB_agdetected, 48
numB_somhyp, 48
numT, 48
numT_active, 48
createCells
CDiscrete3DSpace, 24
createRandomSequence
CPeptide, 38
CStatistics, 49
~CStatistics, 50
CStatistics, 49
printToDatFiles, 50
printVecToDatFile, 50
run, 50
CTCell, 51
~CTCell, 52
CTCell, 51
getAffinityCD8ToPeptide, 52
getAffinityTCRTToPeptide, 52
getIsAbleToBindMHC, 52
print, 52
setIsAbleToBindMHC, 52
CTissue, 54
~CTissue, 54
CTissue, 54
run, 55
currentState
CImmuneCell, 30
debugPrint
CPosition, 40
decrMaturityLevel
CImmuneCell, 30
decrRemainingLifetime
CLivingObject, 33
decrRemainingSomHypSteps
CBCell, 15
deleteObject
CDiscrete3DSpace, 24
deleteObjectAtPosition
CDiscrete3DSpace, 25
doAgAbReaction
CReactEnvNTissue, 45
doAgAPCReaction
CReactEnvNTissue, 46
doAgBCellReaction
CReactEnvNTissue, 46
doLifeCycle
CCreateNDevEnv, 20
CDiscrete3DSpace, 25
CReactEnvNTissue, 46
doMovements
CDiscrete3DSpace, 25
doTCelAPCReaction
CReactEnvNTissue, 46
doTCelBCellReaction
CReactEnvNTissue, 46
eatAndPresent
CAPC, 13
CBCell, 15
freePlaces
CDiscrete3DSpace, 26
getAbObject
CObjectCreator, 34
getAffinityBCRTToPeptide
CBCell, 15
getAffinityCD8ToPeptide
CTCell, 52
getAffinityTCRTToPeptide
CTCell, 52
INDEX

getAgObject
   CObjectCreator, 34
getAPCObject
   CObjectCreator, 35
getBCObject
   CObjectCreator, 35
getBCR
   CBCell, 15
getEpitope
   CAgObject, 10
getIsAbleToBindMHC
   CTCell, 52
getIsMature
   CImmuneCell, 30
getLength
   CPeptide, 38
getMHCObject
   CAPC, 13
   CBCell, 15
getNumberOfMHCObjects
   AIMS2, 6
getNumberOfSelfSequences
   AIMS2, 6
getNumFreePlaces
   CDisc3DSpace, 25
getObjectAtPosition
   CDisc3DSpace, 25
getParatope
   CAbObject, 8
getPresentedPeptide
   CAPC, 13
   CBCell, 15
getRandomFreePosition
   CDisc3DSpace, 25
getRandomPosition
   CDisc3DSpace, 25
getRemainingSomHypSteps
   CBCell, 16
getSelfAgObject
   CObjectCreator, 35
getSelfAPCObject
   CObjectCreator, 35
getSequence
   CPeptide, 38
getState
   CImmuneCell, 30
getTCObject
   CObjectCreator, 35
incrRemainingLifetime
   CLivingObject, 33
initCreateMHC
   CObjectCreator, 35
initCreateSelfSequences
   CObjectCreator, 36
initSelf
   AIMS2, 6
isLifeTimeExpired
   CLivingObject, 33
isPeptidePresented
   CAPC, 13
isPositionFree
   CDisc3DSpace, 26
isSelfAg
   CAgObject, 10
length
   CPeptide, 39
lengthEpitope
   CAgObject, 10
lengthParatope
   CAbObject, 8
m_Ags
   CRreactEnvNTissue, 47
m_epitope
   CAgObject, 11
m_matureCells
   CCreateNDevEnv, 20
m_objects
   CDisc3DSpace, 26
m_paratope
   CAbObject, 8
m_TCells
   CRreactEnvNTissue, 47
maturityLevel
   CImmuneCell, 30
moveCellsToEnv
   CRreactEnvNTissue, 47
moveMatureCellsToEnv
   CCreateNDevEnv, 20
mutateBCR
   CBCell, 16
numAb
   CRreactEnvNTissue, 47
numAb_death
   CRreactEnvNTissue, 47
numAg
   CRreactEnvNTissue, 47
numAg_death
   CRreactEnvNTissue, 47
numAg_self
   CRreactEnvNTissue, 47
numAPC
   CRreactEnvNTissue, 47
numAPC_presenting
   CRreactEnvNTissue, 47
numB
  CReactEnvNTissue, 48
numB_active
  CReactEnvNTissue, 48
numB_agdetected
  CReactEnvNTissue, 48
numB_somhyp
  CReactEnvNTissue, 48
numT
  CReactEnvNTissue, 48
numT_active
  CReactEnvNTissue, 48
operator()
  CDDiscrete3DSpace::cmpPos, 28
operator=
  CPeptide, 38
Pos2LivObjMap
  CDDiscrete3DSpace, 24
print
  CABObject, 8
  CAgObject, 10
  CAPC, 13
  CBCell, 16
  CLivingCell, 30
  CLivingObject, 33
  CPosition, 40
  CTCell, 52
printMHCmolecules
  CObjectCreator, 36
printObjects
  CDDiscrete3DSpace, 26
printSellSequences
  CObjectCreator, 36
printSequence
  CPeptide, 39
printToDatFiles
  CStatistics, 50
printVecToDatFiles
  CStatistics, 50
remainingLifeTime
  CLivingObject, 33
removObjectAtPosition
  CDDiscrete3DSpace, 26
run
  CCreateEnv, 17
  CDevEnv, 22
  CDDiscrete3DSpace, 26
  CReactEnv, 43
  CStatistics, 50
  CTissue, 55
runSimulation

AIMS2, 6
sequence
  CPeptide, 39
setBCR
  CBCell, 16
setEpitope
  CAgObject, 10
setIsAbleToBindMHC
  CTCell, 52
setLifetime
  CLivingObject, 33
setMHCmolecule
  CAPC, 13
  CBCell, 16
setParatope
  CABObject, 8
setSequence
  CPeptide, 39
setState
  CLivingCell, 30
SetTissueEnv
  CReactEnv, 43
x
  CPosition, 41
xMax
  CDDiscrete3DSpace, 26
xMin
  CDDiscrete3DSpace, 27
y
  CPosition, 41
yMax
  CDDiscrete3DSpace, 27
yMin
  CDDiscrete3DSpace, 27
z
  CPosition, 41
zMax
  CDDiscrete3DSpace, 27
zMin
  CDDiscrete3DSpace, 27