

# Contents

<b>Executive summary</b>	<b>4</b>
<b>Temporal organization of all the use cases</b>	<b>5</b>
<b>1 Use Case - Data bases (DB)</b>	<b>6</b>
1.1 Attributes . . . . .	7
1.2 Data file reader . . . . .	8
1.3 Data instance . . . . .	9
1.4 Scalable DataInstance management . . . . .	10
1.5 Time tables . . . . .	11
<b>2 Use Case - Basic data structures (BS)</b>	<b>13</b>
2.1 Static variables . . . . .	13
2.2 Dynamic variables . . . . .	14
2.3 Directed acyclic graph (DAG) . . . . .	15
2.4 Distributions . . . . .	15
2.5 Bayesian network . . . . .	16
2.6 Dynamic DAG . . . . .	16
2.7 2T-DBN . . . . .	17
2.8 Time tables . . . . .	18
<b>3 Use Case - Hugin link (HL)</b>	<b>21</b>
3.1 Bayesian networks converter . . . . .	21
3.2 Time tables . . . . .	22
<b>4 Use Case - Maximum likelihood (ML)</b>	<b>23</b>
4.1 Exponential family distributions . . . . .	24
4.2 Time tables . . . . .	24
<b>5 Use Case - Importance sampling (IS)</b>	<b>25</b>
<b>6 Use Case - Variational Message Passing (VMP)</b>	<b>26</b>

<b>7 Use Case - Expectation Propagation (EP)</b>	<b>27</b>
<b>8 Use Case - MAP with Deterministic Approximations (DMAP)</b>	<b>28</b>
<b>9 Use Case - Variational MAP Inference (VMAP)</b>	<b>29</b>
<b>10 Use Case - TAN Classifier (TAN)</b>	<b>30</b>
<b>11 Use Case - Parallel PC (PPC)</b>	<b>31</b>
<b>12 Use Case - Dynamic classifiers (DC)</b>	<b>32</b>
<b>13 Use Case - Feature Selection (FS)</b>	<b>33</b>
<b>Class diagrams</b>	<b>34</b>
13.1 Package core . . . . .	34
13.2 Package core.database . . . . .	35
13.3 Package core.database.dynamics . . . . .	36
13.4 Package core.database.readers . . . . .	36
13.5 Package core.database.filereaders . . . . .	37
13.6 Package core.database.arffFileReader . . . . .	38
13.7 Package core.database.arffWekaReader . . . . .	38
13.8 Package core.distribution . . . . .	39
13.9 Package core.exponentialfamily . . . . .	40
13.10 Package core.exponentialfamily . . . . .	41
13.11 Package core.modelstructure . . . . .	42
13.12 Package core.potential . . . . .	43
13.13 Package core.utils . . . . .	44
13.14 Package core.variables . . . . .	45
<b>A Code Description (Doxygen Document)</b>	<b>47</b>

## Document history

Version	Date	Author (Unit)	Description
v0.3			First draft finished

---

## Executive summary

The aim of this document is to track the development process of the AMIDST toolbox.

The document is structured into sections, such that each section includes the required details for a specific use case, i.e., a general description of the purpose of the use case as well as a categorised list of its associated requirements.

We then associate to each use case a list of so-called *functionalities* (included in independent subsections) that are coded in the toolbox to cover the specific use case and their associated requirements. A *functionality*, or also called a *feature*, is a set of java classes that allows to perform a specific task or that defines a coherent concept within the toolbox. For example, the creation and management of random variables, reading a data set from a file, building and handling dynamic Bayesian networks, the maximum likelihood estimation, etc. The set of *functionalities* identify those key parts of the toolbox that any developer or toolbox user need to understand in order to make a proper use of the AMIDST toolbox.

Another important part of this document is to detail the time line of the toolbox development. Each of the use cases contain an additional subsection named *Time tables* which contain details about the development phase of each *functionality* associated to this use case. Six main phases are identified: design, prototype, code-review, testing, documentation and (first) release. Through this section, we can easily track the evolution of the use case.

In the last section, we detail the class diagram of each package in the AMIDST toolbox.

Finally, the document is concluded with an appendix which contains a pdf document generated by the toolbox Doxygen and which contains all the java docs of all the coded java classes.

We highlight that the contents detailed here can be later modified or re-organized to create the different deliverables and the final user-manual of the AMIDST toolbox.

---

## Temporal organization of all the use cases

Use case ID	Deadline	Start Date	End Date	Status	Priority
DB	M15	M10	-	-	Must
BS	M15	M6	-	-	Must
HL	M15	M11	-	-	Must
VMP	M15	-	-	-	Must
IS	M16	-	-	-	Must
DMAP	M17	-	-	-	Must
PPC	M17	-	-	-	Must
ML	M18	M12	-	-	Must
EP	M18	-	-	-	Must
TAN	M18	-	-	-	Must
FS	M20	-	-	-	Must
VMAP	M28	-	-	-	Should
DC	M29	-	-	-	Could

## 1 Use Case - Data bases (DB)

**Priority:** Must

**Deadline:** M15

**Responsible:** Sigve

### Description of the use case

This use case will cover the managing of the data bases that will be used by the models and learning algorithms implemented in the toolbox.

### Must-Requirements list of the use case

1. Data on memory
2. Data on disk

### Should-Requirements list of the use case

1. Data on stream

### Could-Requirements list of the use case

1. Short Description

## 1.1 Attributes

**Deadline:** M12

**Responsible:** Sigve

**Code-Package:** core.databases

### Description

Attributes serve as an intermediary to build the Static or Dynamic Variables from the dataset that are parsed from the dataset and/or specified by the user (precisely by a class that extends this class).

### Detailed functionality

- List of objects of the type Attribute. This list becomes Unmodifiable after construction.
- There can be two special attributes, namely "TIME\_ID" AND "SEQUENCE\_ID". The former refers to a temporal identifier whereas the second identifies a particular sequence (e.g. a client in CajaMar or a drill in Verdande). They only exist if they explicitly appear in the dataset with that particular names.

## 1.2 Data file reader

**Deadline:** M12

**Responsible:** Sigve

**Code-Package:** core.database.filereaders

### Description

It refers to the different parsers.

### Detailed functionality

Right now we have the following readers:

1. Weka wrapper reader: fully working and tested.
2. AMIDST arff reader.

### 1.3 Data instance

**Deadline:** M12

**Responsible:** Sigve

**Code-Package:** core.database

#### Description

DataInstance is an interface that can be implemented by a StaticDataInstance and a DynamicDataInstance. The former stores one row (DataRow) of the dataset at a time, whereas the second contains two rows of the dataset.

#### Detailed functionality

A DynamicDataInstance always has a TimeID and a SequenceID. If this two attributes, or any of the two, are not in the dataset then they are automatically filled in, incrementally for the TimeID and with a value of 1 for the SequenceID.

- StaticDataInstance: contains one DataRow
  - DynamicDataInstance: contains a past and a present DataRow, a TimeID and a SequenceID. The TimeID attribute in the dataset can be used to represent missing samples (stored as DataRowMissing), in which all attribute values are NaN.
-

## 1.4 Scalable DataInstance management

**Deadline:** M12

**Responsible:** Sigve

**Code-Package:** core.database

### Description

It is used to manage the output of the reader (DataRows) and convert them into DataInstances. We need to distinguish between static and dynamic and whether the data can be load into memory or read from disk.

### Detailed functionality

- Static:

- Data on memory
- Data on disk
- Data on stream.

- Dynamic:

- Data on memory
- Data on disk
- Data on stream.

## 1.5 Time tables

### Attributes

<b>Version</b>	<b>Phase</b>	<b>Author(s)</b>	<b>Deadline</b>	<b>Start Date</b>	<b>End Date</b>
0.1	Design	Andres,Sigve	M15	M10	M10
0.2	Prototype	Ana, Sigve	M15	M10	M10
0.3	Code Review	Ana	M15	M11	M10
0.4	Testing	Ana	M15	M11	M11
0.5	Java-Doc	Sigve	M15		
0.6	First Release	Post-docs	M15		

### Data file reader

<b>Version</b>	<b>Phase</b>	<b>Author(s)</b>	<b>Deadline</b>	<b>Start Date</b>	<b>End Date</b>
0.1	Design	Ana,Andres	M15	M10	M11
0.2	1st Prototype (AMIDSTarffParser)	Sigve	M15	M10	Unfinished
0.2	Prototype (WekaArffParser)	Ana	M15	M11	M11
0.2	Testing (WekaArffParser)	Ana	M15	M11	M11
0.3	Code Review	Post-docs	M15		
0.4	Testing	Sigve	M15		
0.5	Java-Doc	Sigve	M15		
0.6	First Release	Post-docs	M15		

### Data instance

<b>Version</b>	<b>Phase</b>	<b>Author(s)</b>	<b>Deadline</b>	<b>Start Date</b>	<b>End Date</b>
0.1	Design	Ana,Andres	M15	M11	M11
0.2	Prototype	Ana	M15	M11	M11
0.2	Prototype testing	Ana	M15	M11	M11
0.3	Code Review	Sigve	M15		
0.4	Testing	Sigve	M15		
0.5	Java-Doc	Sigve	M15		
0.6	First Release	Post-docs	M15		

## Scalable DataInstance management

<b>Version</b>	<b>Phase</b>	<b>Author(s)</b>	<b>Deadline</b>	<b>Start Date</b>	<b>End Date</b>
0.1	Design	Ana,Andres	M15	M11	M11
0.2	Prototype	Ana	M15	M11	M11
0.2	Prototye Testing	Ana	M15	M11	M11
0.3	Code Review	Sigve	M15		
0.4	Testing	Sigve	M15		
0.5	Java-Doc	Sigve	M15		
0.6	First Release	Post-docs	M15		

## 2 Use Case - Basic data structures (BS)

**Priority:** Must

**Deadline:** M15

**Responsible:**

### Description of the use case

This use case will cover the basic data structures to handle the probabilistic graphical models belonging to the AMIDST model class, and to be included in the toolbox (a tentative list is given below).

On the other hand, information about possible models to be plug-in into the toolbox beyond the project is also desirable to be indicated.

### Must-Requirements list of the use case

1. Static Bayesian network (BN)
2. Two-time slice dynamic Bayesian network (2T-DBN)
3. Bounded dynamic Bayesian network

### Should-Requirements list of the use case

1. Additional operations for learning Bayesian networks

### Could-Requirements list of the use case

1. Factor graphs

## 2.1 Static variables

**Deadline:** M12

**Responsible:**

**Code-Package:** eu.amidst.core.variables

### Description

Static variables define the list of static variables to be used in the static BN models.

---

## Detailed functionality

- List of objects of the type Variable.
- A static variable is characterised by its name, ID, the number of states, the state space type, the distribution type, as well as if it is observable or not.
- The state space type could be either Multinomial or Real.
- The distribution type could be either Multinomial or Gaussian.
- The list of observable static variables is initialised using the list of Attributes (that are already parsed from the dataset or specified by the user), then hidden variables can be also added.

## 2.2 Dynamic variables

**Deadline:** M12

**Responsible:**

**Code-Package:** eu.amidst.core.variables

### Description

Dynamic variables define the list of dynamic variables to be used in the dynamic BN models.

## Detailed functionality

- List of objects named allVariables and temporalClones of the type Variable.
- A dynamic variable is characterised by its name, ID, the number of states, the state space type, the distribution type, if it is observable or not, and if it is temporal clone or not.
- The state space type could be either Multinomial or Real.
- The distribution type could be either Multinomial or Gaussian.
- The list of observable dynamic variables and their temporal clones is initialised using the list of Attributes (that are already parsed from the dataset or specified by the user), then hidden variables and their temporal clones can be also added.

## 2.3 Directed acyclic graph (DAG)

**Deadline:** M12

**Responsible:**

**Code-Package:** eu.amidst.core.models

### Description

The class Directed acyclic graph (DAG) defines the Bayesian network graphical structure over a list of static variables.

### Detailed functionality

- It defines the parent set for each variable.
- It test and detect if a DAG contains cycles or not.

## 2.4 Distributions

**Deadline:** M15

**Responsible:** Antonio Fernández

**Code-Package:** eu.amidst.core.distributions

### Description

This functionality addresses the set of conditional probability distributions considered to be included in the toolbox. Variables with Gaussian and multinomial distributions are modeled. The variables arrangement in the model structure gives rise to the different types of probability distributions, one for each variable in the network.

This functionality is tightly connected to functionality **Variable** and **DAG** to know both the type and the set of parents of each variable.

### Detailed functionality

The type and the set of parents of each variable determine the different defined probability distributions as follows:

- Multinomial variable with no parents
-

- Multinomial variable with multinomial parents.
- Gaussian variable with no parents.
- Gaussian variable with multinomial parents.
- Gaussian variable with Gaussian parents.
- Gaussian variable with a mixture of multinomial and Gaussian parents.

Note that a multinomial variable is not allowed to have Gaussian parents and therefore it has not been included in the list above.

Multinomial parents are only used for indexing the set of possible distributions of the variable, so the functionality when no multinomial parents reduces to the general case.

## 2.5 Bayesian network

**Deadline:** M12

**Responsible:**

**Code-Package:** eu.amidst.core.models

### Description

This class defines a static Bayesian network using the already specified graphical structure (DAG) along with the conditional probability distribution of each variable given the set of its parents.

### Detailed functionality

- The distribution of each variable in the Bayesian network is initialised and specified according to its type and the type of its potential parent set.
- After this step, the set of parents of each variable becomes unmodifiable.

## 2.6 Dynamic DAG

**Deadline:** M12

**Responsible:**

**Code-Package:** eu.amidst.core.models

---

## Description

It defines the dynamic Bayesian network graphical structure over a list of dynamic variables by specifying their parent sets at time 0 and time T.

## Detailed functionality

- It defines the parent set at time 0 and time T for each dynamic variable.
- It test and detect if a dynamic DAG contains cycles or not.

## 2.7 2T-DBN

**Deadline:** M12

**Responsible:**

**Code-Package:** core.models

## Description

This functionality is similar to the one defined for Bayesian network (see 2.5), but it aims here to represent instead a two slice-time dynamic Bayesian network (2T-DBN). It handles its structure (dynamic DAG defined over a set of dynamic variables) and the set of its distributions at time 0 and time T.

## Detailed functionality

- The distributions of each dynamic variable at both time 0 and time T are initialised and specified according to the variable type and the type of its potentiel parent set.
- After this step, the set of parents of each dynamic variable becomes unmodifiable.

## 2.8 Time tables

### Static variables

Version	Phase	Author(s)	Deadline	Start Date	End Date
0.1	Design	Ana, Andres, Antonio	M11	M6	M11
0.2	Prototype	Andres, Hanen	M11	M11	M11
0.3	Code Review	Ana	M11	M11	M11
0.4	Testing	Ana, Antonio, Hanen	M11	M11	M11
0.5	Java-Doc	Post-docs			
0.6	First Release	Post-docs	M15		

### Dynamic variables

Version	Phase	Author(s)	Deadline	Start Date	End Date
0.1	Design	Ana, Andres, Antonio	M11	M6	M11
0.2	Prototype	Ana	M11	M11	M11
0.3	Code Review	Andres	M11	M11	M11
0.4	Testing	Ana	M11	M11	M11
0.5	Java-Doc	Post-docs			
0.6	First Release	Post-docs	M15		

### DAG

Version	Phase	Author(s)	Deadline	Start Date	End Date
0.1	Design	Andres, Hanen	M11	M6	M11
0.2	Prototype	Hanen	M11	M11	M11
0.3	Code Review	Ana, Andres, Antonio	M11	M11	M11
0.4	Testing	Hanen	M11	M11	M11
0.5	Java-Doc	Post-docs			
0.6	First Release	Post-docs	M15		

## Distributions

<b>Version</b>	<b>Phase</b>	<b>Author(s)</b>	<b>Deadline</b>	<b>Start Date</b>	<b>End Date</b>
0.1	Design	Andres, Antonio	M11	M6	M11
0.2	Prototype	Antonio	M11	M11	M11
0.3	Code Review	Ana, Andres	M11	M11	M11
0.4	Testing	Antonio	M11	M11	M11
0.5	Java-Doc	Post-docs			
0.6	First Release	Post-docs	M15		

## Bayesian network

<b>Version</b>	<b>Phase</b>	<b>Author(s)</b>	<b>Deadline</b>	<b>Start Date</b>	<b>End Date</b>
0.1	Design	Andres, Antonio, Hanen	M11	M6	M11
0.2	Prototype	Hanen	M11	M11	M11
0.3	Code Review	Ana, Andres, Antonio	M11	M11	M11
0.4	Testing	Post-docs	M11		
0.5	Java-Doc	Post-docs			
0.6	First Release	Post-docs	M15		

## Dynamic DAG

<b>Version</b>	<b>Phase</b>	<b>Author(s)</b>	<b>Deadline</b>	<b>Start Date</b>	<b>End Date</b>
0.1	Design	Ana, Andres,	M12	M11	M11
0.2	Prototype	Ana, Andres	M12	M11	M11
0.3	Code Review	Post-docs	M12		
0.4	Testing	Post-docs	M12		
0.5	Java-Doc	Post-docs			
0.6	First Release	Post-docs	M15		

**2T-DBN**

<b>Version</b>	<b>Phase</b>	<b>Author(s)</b>	<b>Deadline</b>	<b>Start Date</b>	<b>End Date</b>
0.1	Design	Ana, Andres, Antonio	M12	M6	M11
0.2	Prototype	Ana, Antonio	M12	M11	M11
0.3	Code Review	Andres	M12	M11	
0.4	Testing	Post-docs	M12		
0.5	Java-Doc	Post-docs			
0.6	First Release	Post-docs	M15		

### 3 Use Case - Hugin link (HL)

**Priority:** Must

**Deadline:** M15

**Responsible:** A. Fernández

#### Description of the use case

This use case contains all the functionality needed to link the AMIDST toolbox with the HUGIN software. This linkage is addressed by converting Hugin models into AMIDST models, and vice versa. This feature is extremely useful as it allows expanding the testing possibilities of the AMIDST models within a well-established platform as Hugin. Also, the link to Hugin can be used for providing some extra functionality to AMIDST that will not be implemented. Finally, the linkage is useful for comparison purposes, i.e., a new inference algorithm implemented in AMIDST could be compared with some state-of-the-art algorithm included in Hugin.

#### Must-Requirements list of the use case

1. Bayesian network converter from AMIDST to Hugin format.
2. Bayesian network converter from Hugin to AMIDST format.
3. Possibility of saving the converted Hugin network in a .net file.

#### Should-Requirements list of the use case

- 1.

#### Could-Requirements list of the use case

1. Converter from AMIDST to HUGIN and vice versa of some functionality that is not relevant for model representation.

#### 3.1 Bayesian networks converter

**Deadline:** M12

**Responsible:** A. Fernández

**Code-Package:** core.huginlink

---

## Description

This functionality addresses the conversion of a static Bayesian network model from AMIDST to HUGIN and vice-versa. This conversion is done at “object-level”, which is far more efficient than if done by converting the models to data files and, then, parsing them.

## Detailed functionality

- Short Description

### 3.2 Time tables

#### Bayesian networks converter

Version	Phase	Author(s)	Deadline	Start Date	End Date
0.1	Design	Antonio	M12	M11	M11
0.2	Prototype	Antonio	M12	M11	M11
0.3	Code Review	Andres, Ana	M12	M11	M11
0.4	Testing	Post-docs	M12	M11	M11
0.5	Java-Doc	Post-docs	M15		
0.6	First Release	Post-docs	M15		

## 4 Use Case - Maximum likelihood (ML)

**Priority:** Must

**Deadline:** M12

**Responsible:**

### Description of the use case

Enter a textual description of the use case. Link to other use cases or functionalities if needed.

### Must-Requirements list of the use case

1. Short Description

### Should-Requirements list of the use case

1. Short Description

### Could-Requirements list of the use case

1. Short Description

## 4.1 Exponential family distributions

**Deadline:** M12

**Responsible:** Andres

**Code-Package:** core.exponentialfamily

### Description

This functionality addresses the representation of the probability distributions in an exponential family form. I.e. they are encoded by a vector of natural and moment parameters. Functionality for computing the sufficient statistics for a data instance is also provided. These set of classes are going to be used across several use cases: ML, VMP, EP, VMAP, etc.

### Detailed functionality

The type of each variable and its parents determine the different exponential family distributions detailed next:

- Multinomial variable with no parents
- Multinomial variable with multinomial parents.
- Gaussian variable with no parents.
- Gaussian variable with multinomial parents.
- Gaussian variable with Gaussian parents.
- Gaussian variable with multinomial and Gaussian parents.

## 4.2 Time tables

### Exponential family

Version	Phase	Author(s)	Deadline	Start Date	End Date
0.1	Design	Andres	M12	M11	M11
0.2	Prototype	Andres	M12	M11	
0.3	Code Review	Post-docs	M12		
0.4	Testing	Post-docs	M12		
0.5	Java-Doc	Post-docs	M15		
0.6	First Release	Post-docs	M15		

## 5 Use Case - Importance sampling (IS)

**Priority:** Must

**Deadline:** M16

**Responsible:**

### Description of the use case

Enter a textual description of the use case. Link to other use cases or functionalities if needed.

### Must-Requirements list of the use case

1. Short Description

### Should-Requirements list of the use case

1. Short Description

### Could-Requirements list of the use case

1. Short Description

## 6 Use Case - Variational Message Passing (VMP)

**Priority:** Must

**Deadline:** M15

**Responsible:**

### Description of the Use Case

Enter a textual description of the use case. Link to other use cases or functionalities if needed.

### Must-Requirements List of the Use Case

1. Short Description

### Should-Requirements List of the Use Case

1. Short Description

### Could-Requirements List of the Use Case

1. Short Description

## 7 Use Case - Expectation Propagation (EP)

**Priority:** Must

**Deadline:** M18

**Responsible:**

### Description of the Use Case

Enter a textual description of the use case. Link to other use cases or functionalities if needed.

### Must-Requirements List of the Use Case

1. Short Description

### Should-Requirements List of the Use Case

1. Short Description

### Could-Requirements List of the Use Case

1. Short Description

## 8 Use Case - MAP with Deterministic Approximations (DMAP)

**Priority:** Must

**Deadline:** M17

**Responsible:**

### Description of the Use Case

Enter a textual description of the use case. Link to other use cases or functionalities if needed.

### Must-Requirements List of the Use Case

1. Short Description

### Should-Requirements List of the Use Case

1. Short Description

### Could-Requirements List of the Use Case

1. Short Description

## 9 Use Case - Variational MAP Inference (VMAP)

**Priority:** Should

**Deadline:** M28

**Responsible:**

### Description of the Use Case

Enter a textual description of the use case. Link to other use cases or functionalities if needed.

### Must-Requirements List of the Use Case

1. Short Description

### Should-Requirements List of the Use Case

1. Short Description

### Could-Requirements List of the Use Case

1. Short Description

## 10 Use Case - TAN Classifier (TAN)

**Priority:** Must

**Deadline:** M18

**Responsible:**

### Description of the Use Case

Enter a textual description of the use case. Link to other use cases or functionalities if needed.

### Must-Requirements List of the Use Case

1. Short Description

### Should-Requirements List of the Use Case

1. Short Description

### Could-Requirements List of the Use Case

1. Short Description

## 11 Use Case - Parallel PC (PPC)

**Priority:** Must

**Deadline:** M17

**Responsible:**

### Description of the Use Case

Enter a textual description of the use case. Link to other use cases or functionalities if needed.

### Must-Requirements List of the Use Case

1. Short Description

### Should-Requirements List of the Use Case

1. Short Description

### Could-Requirements List of the Use Case

1. Short Description

## 12 Use Case - Dynamic classifiers (DC)

**Priority:** Could

**Deadline:** M29

**Responsible:**

### Description of the Use Case

Enter a textual description of the use case. Link to other use cases or functionalities if needed.

### Must-Requirements List of the Use Case

1. Short Description

### Should-Requirements List of the Use Case

1. Short Description

### Could-Requirements List of the Use Case

1. Short Description

## 13 Use Case - Feature Selection (FS)

**Priority:** Must

**Deadline:** M20

**Responsible:**

### Description of the Use Case

Enter a textual description of the use case. Link to other use cases or functionalities if needed.

### Must-Requirements List of the Use Case

1. Short Description

### Should-Requirements List of the Use Case

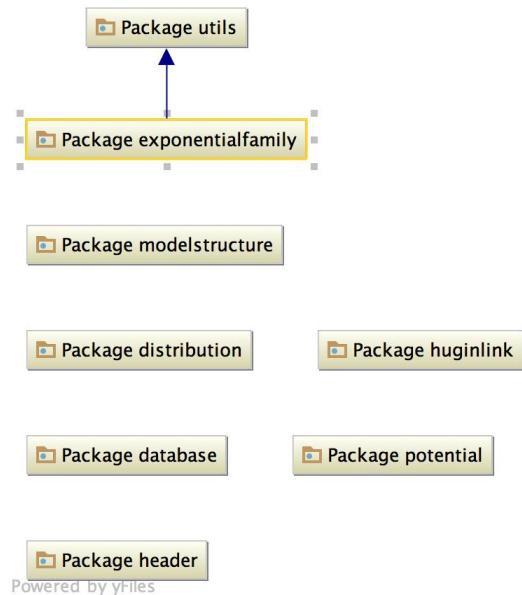
1. Short Description

### Could-Requirements List of the Use Case

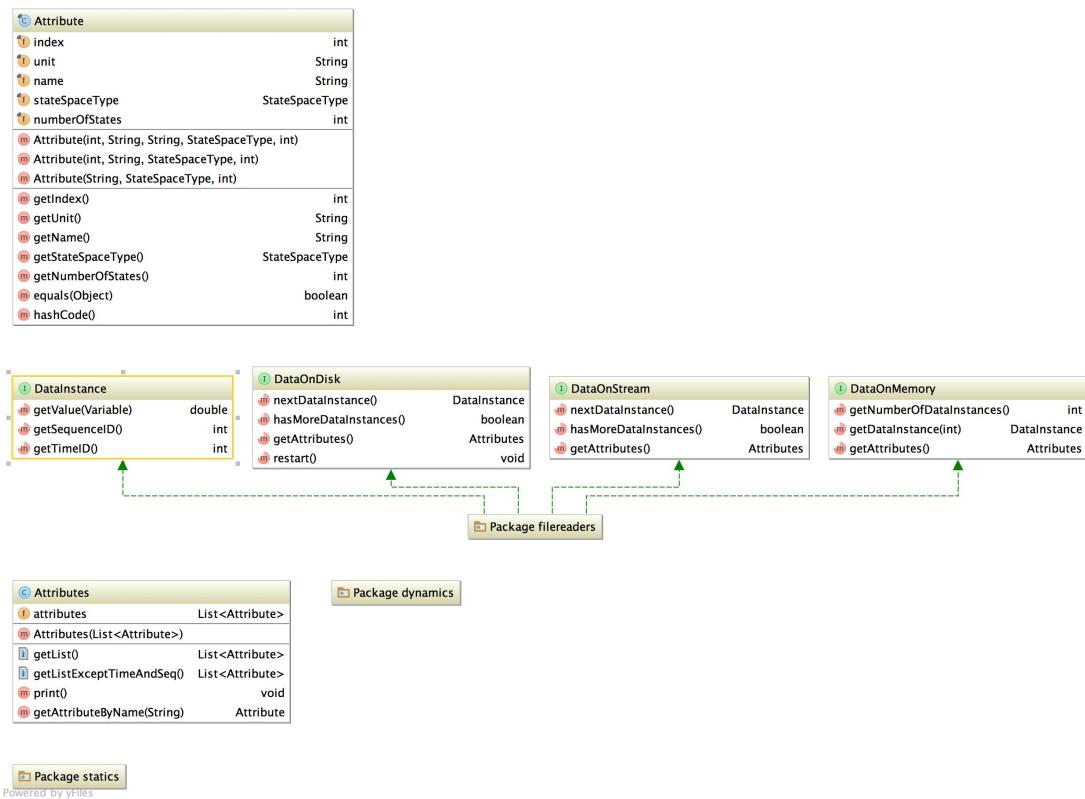
1. Short Description

## Class diagrams

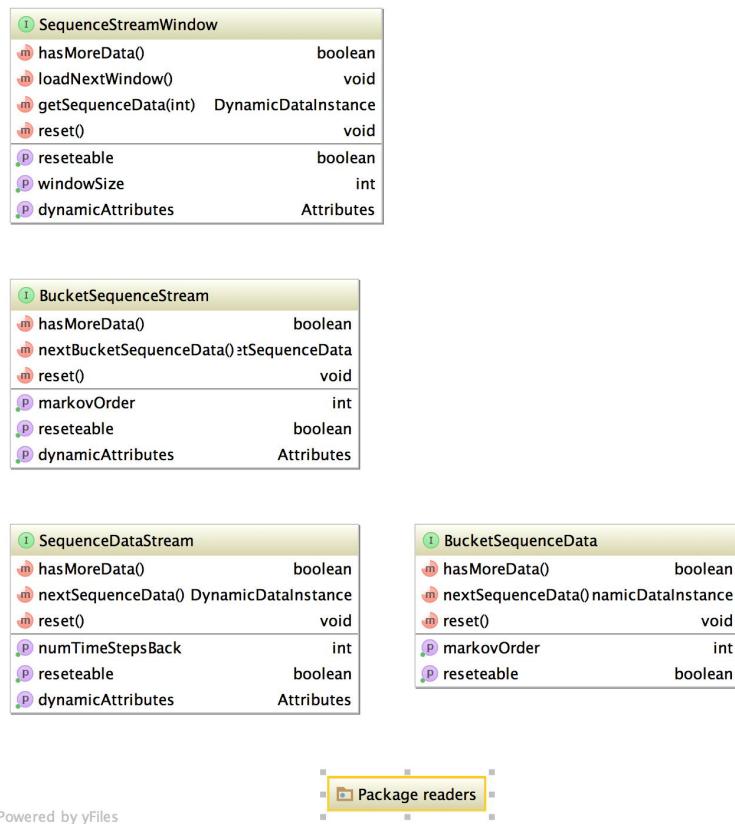
### 13.1 Package core



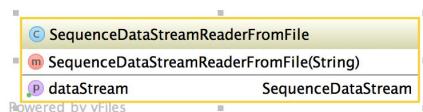
## 13.2 Package core.database



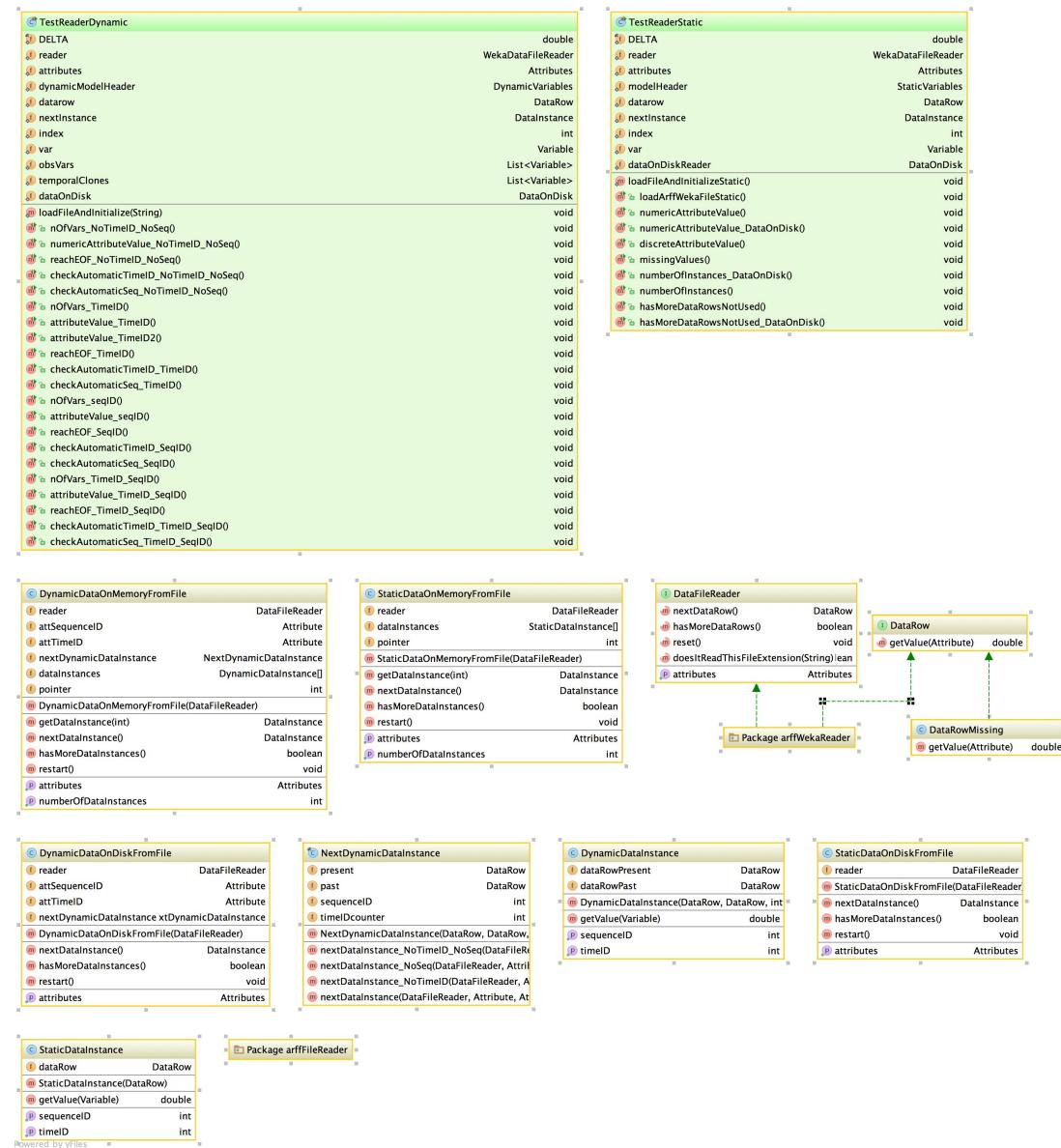
### 13.3 Package core.database.dynamics



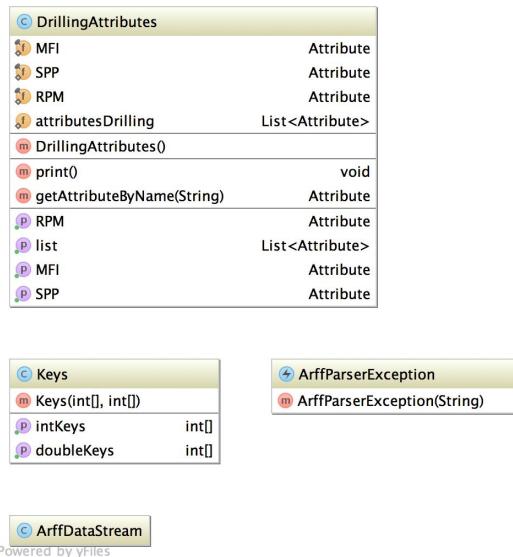
### 13.4 Package core.database.readers



## 13.5 Package core.database.filereaders



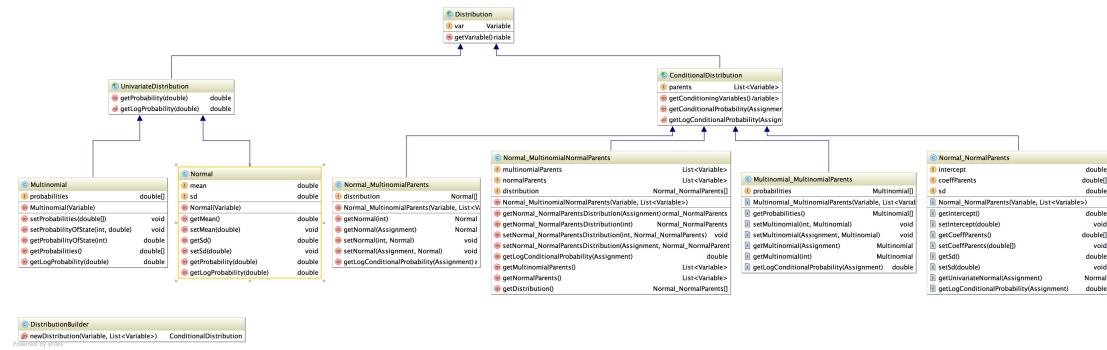
## 13.6 Package core.database.arffFileReader



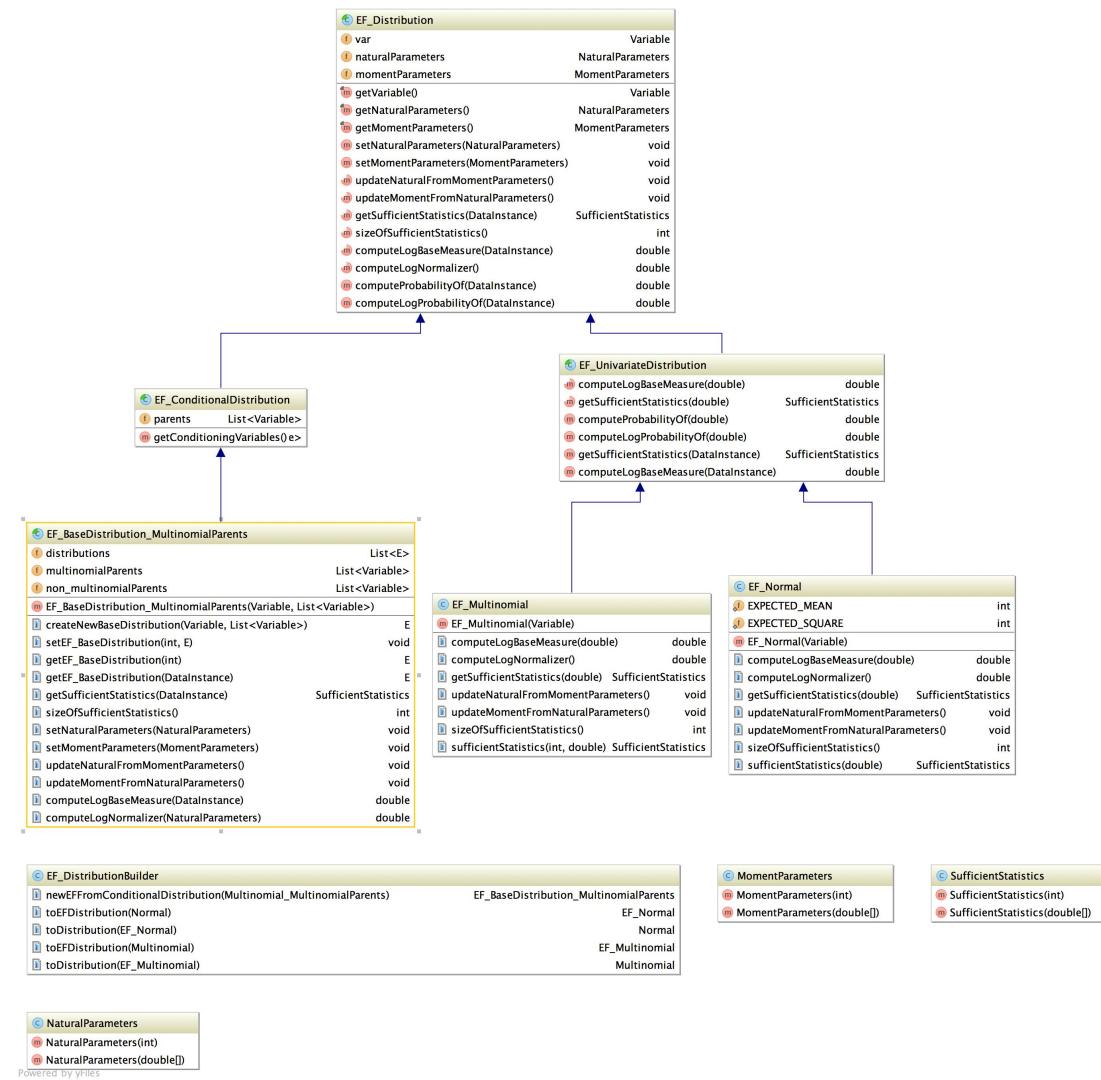
## 13.7 Package core.database.arffWekaReader



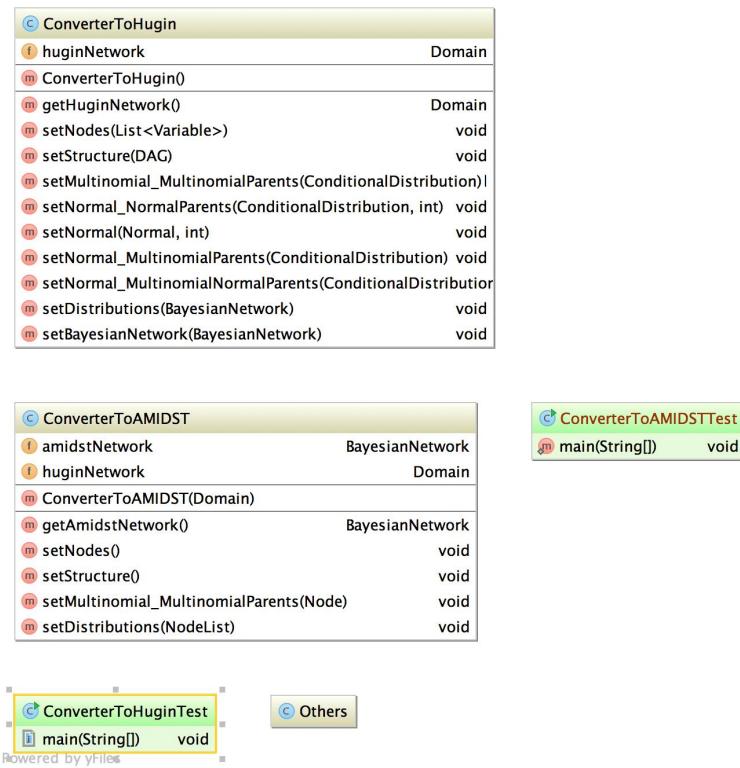
## 13.8 Package core.distribution



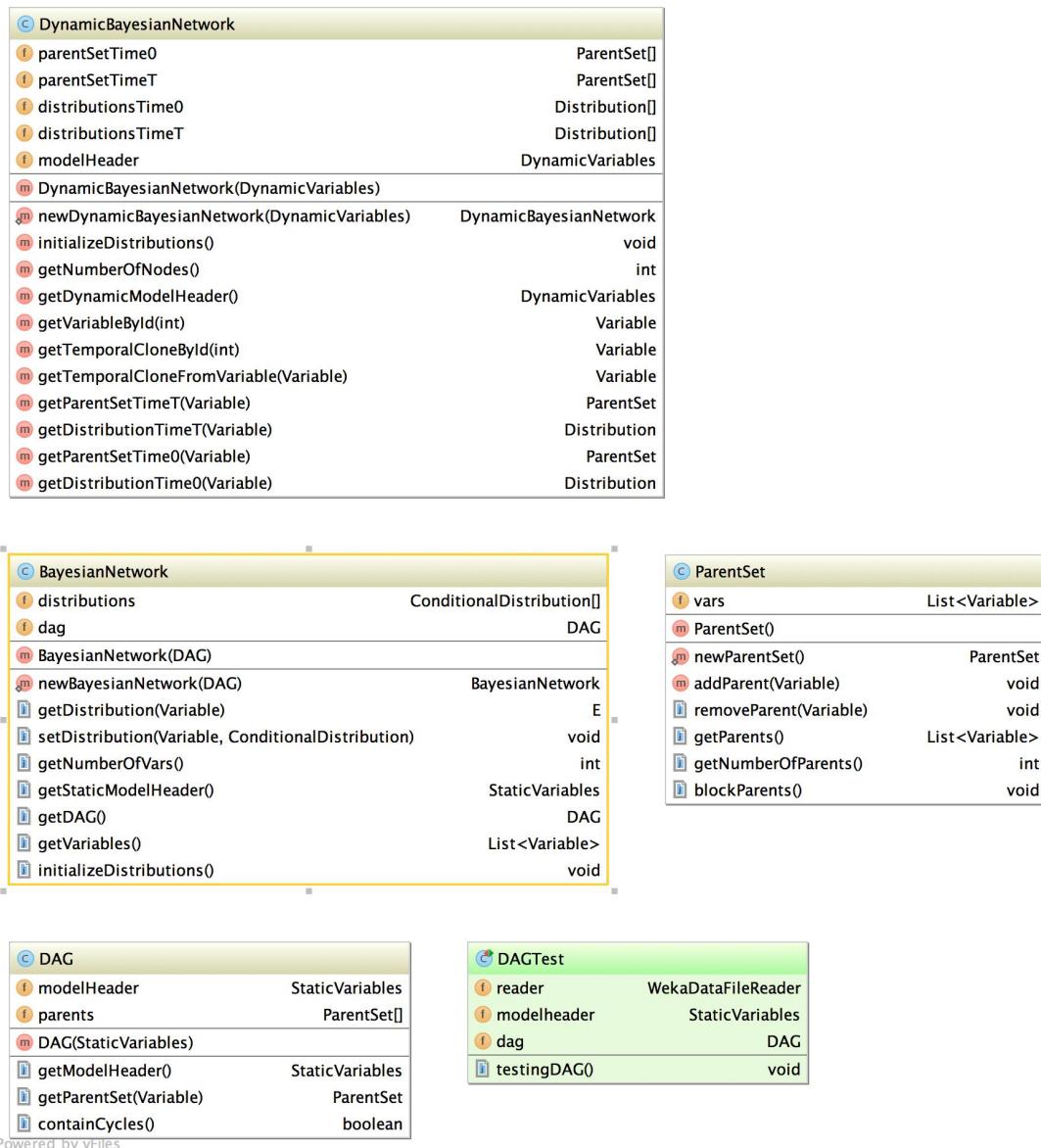
### 13.9 Package core.exponentialfamily



### 13.10 Package core.exponentialfamily

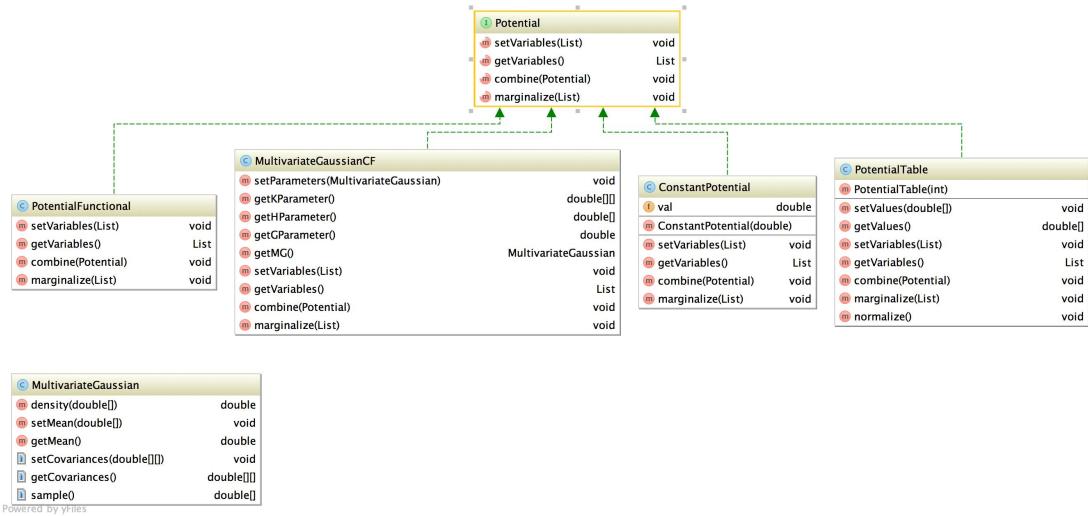


### 13.11 Package core.modelstructure

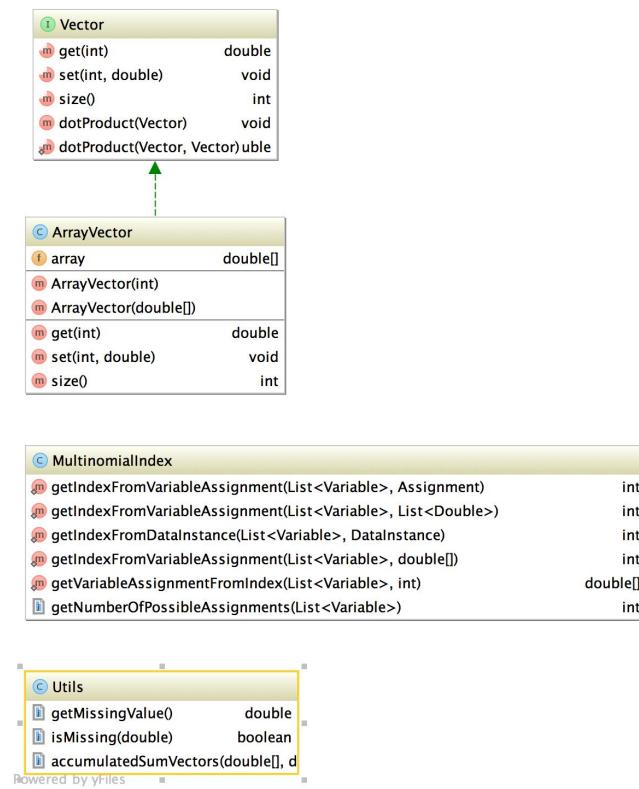


## 13.12 Package core.potential

Figure 1: Class diagram of the package: core.potential

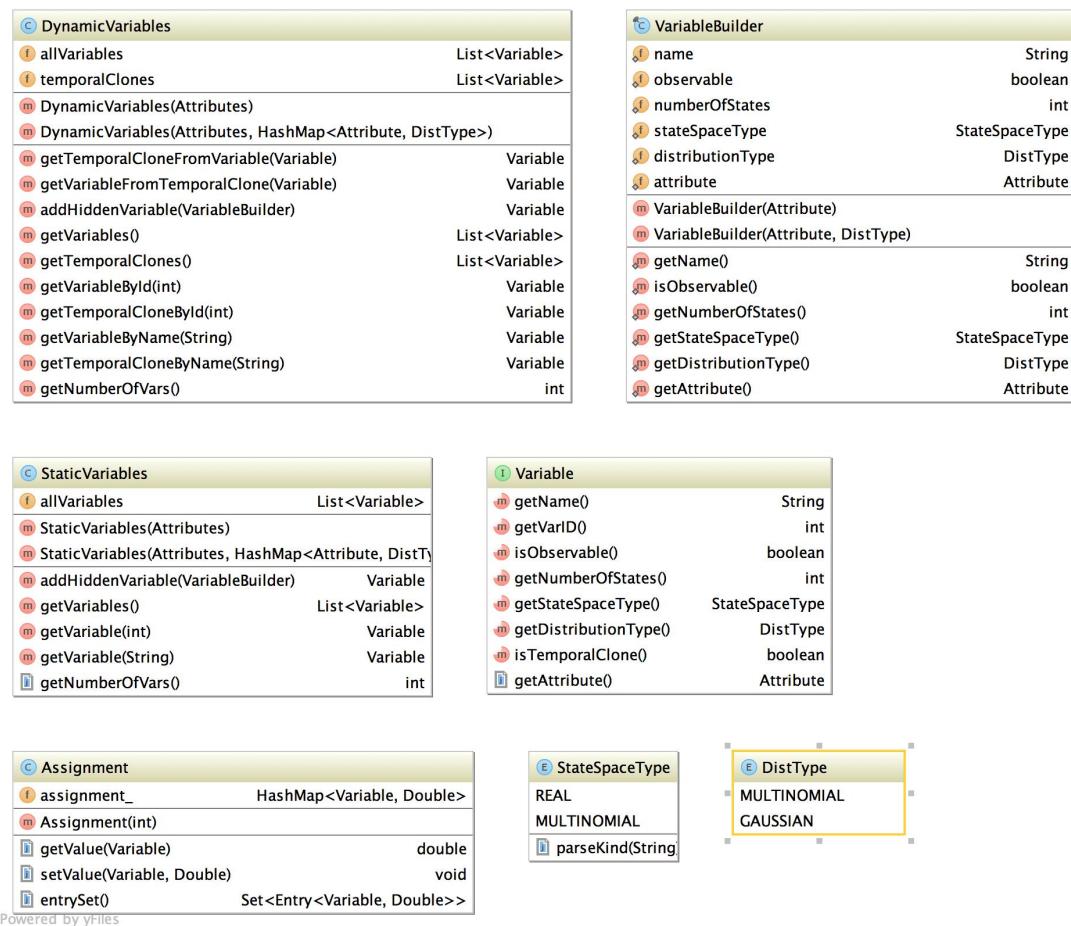


### 13.13 Package core.utils



### 13.14 Package core.variables

Figure 2: Class diagram of the package: `core.variables`



# AMIDSTtoolbox

Generated by Doxygen 1.8.8

Thu Nov 20 2014 15:24:30

## A Code Description (Doxygen Document)

---

## AMIDSTtoolbox

Generated by Doxygen 1.8.8

Thu Nov 20 2014 15:24:30

---



# Contents

<b>1 Namespace Index</b>	<b>1</b>
1.1 Packages . . . . .	1
<b>2 Hierarchical Index</b>	<b>3</b>
2.1 Class Hierarchy . . . . .	3
<b>3 Class Index</b>	<b>5</b>
3.1 Class List . . . . .	5
<b>4 Namespace Documentation</b>	<b>7</b>
4.1 Package eu.amidst.core.database . . . . .	7
4.1.1 Detailed Description . . . . .	7
4.2 Package eu.amidst.core.database.filereaders . . . . .	7
4.2.1 Detailed Description . . . . .	8
4.3 Package eu.amidst.core.distribution . . . . .	8
4.3.1 Detailed Description . . . . .	8
4.4 Package eu.amidst.core.exponentialfamily . . . . .	9
4.4.1 Detailed Description . . . . .	9
4.5 Package eu.amidst.core.variables . . . . .	9
4.5.1 Detailed Description . . . . .	10
<b>5 Class Documentation</b>	<b>11</b>
5.1 eu.amidst.core.database.filereaders.arffFileReader.ArffDataStream Class Reference . . . . .	11
5.1.1 Detailed Description . . . . .	11
5.2 eu.amidst.core.database.filereaders.arffFileReader.ArffParserException Class Reference . . . . .	11
5.2.1 Detailed Description . . . . .	11
5.3 eu.amidst.core.utils.ArrayVector Class Reference . . . . .	12
5.3.1 Detailed Description . . . . .	12
5.4 eu.amidst.core.variables.Assignment Class Reference . . . . .	12
5.4.1 Detailed Description . . . . .	12
5.5 eu.amidst.core.database.Attribute Class Reference . . . . .	12
5.5.1 Detailed Description . . . . .	13
5.6 eu.amidst.core.database.Attributes Class Reference . . . . .	13

---

5.6.1	Detailed Description	13
5.7	eu.amidst.core.modelstructure.BayesianNetwork Class Reference	13
5.7.1	Detailed Description	14
5.8	eu.amidst.core.database.dynamics.BucketSequenceData Interface Reference	14
5.8.1	Detailed Description	14
5.9	eu.amidst.core.database.dynamics.BucketSequenceStream Interface Reference	14
5.9.1	Detailed Description	14
5.10	eu.amidst.staticmodelling.models.Classifier Interface Reference	15
5.10.1	Detailed Description	15
5.11	eu.amidst.core.distribution.ConditionalDistribution Class Reference	15
5.11.1	Detailed Description	15
5.11.2	Member Function Documentation	16
5.11.2.1	getConditionalProbability	16
5.11.2.2	getConditioningVariables	17
5.11.2.3	getLogConditionalProbability	17
5.11.3	Member Data Documentation	17
5.11.3.1	parents	17
5.12	eu.amidst.core.potential.ConstantPotential Class Reference	17
5.12.1	Detailed Description	18
5.13	eu.amidst.core.huginlink.ConverterToAMIDST Class Reference	18
5.13.1	Detailed Description	18
5.14	eu.amidst.core.huginlink.ConverterToHugin Class Reference	18
5.15	eu.amidst.core.modelstructure.DAG Class Reference	18
5.15.1	Detailed Description	19
5.16	eu.amidst.core.database.filereaders.DataFileReader Interface Reference	19
5.16.1	Detailed Description	19
5.17	eu.amidst.core.database.DataInstance Interface Reference	19
5.17.1	Detailed Description	20
5.18	eu.amidst.core.database.DataOnDisk Interface Reference	20
5.18.1	Detailed Description	20
5.19	eu.amidst.core.database.DataOnMemory Interface Reference	20
5.19.1	Detailed Description	20
5.20	eu.amidst.core.database.DataOnStream Interface Reference	21
5.20.1	Detailed Description	21
5.21	eu.amidst.core.database.filereaders.DataRow Interface Reference	21
5.21.1	Detailed Description	21
5.22	eu.amidst.core.database.filereaders.DataRowMissing Class Reference	21
5.22.1	Detailed Description	22
5.23	eu.amidst.core.database.filereaders.arffWekaReader.DataRowWeka Class Reference	22
5.23.1	Detailed Description	22

5.24 eu.amidst.core.distribution.Distribution Class Reference . . . . .	22
5.24.1 Detailed Description . . . . .	23
5.24.2 Member Function Documentation . . . . .	23
5.24.2.1 getVariable . . . . .	23
5.24.3 Member Data Documentation . . . . .	23
5.24.3.1 var . . . . .	23
5.25 eu.amidst.core.distribution.DistributionBuilder Class Reference . . . . .	23
5.25.1 Detailed Description . . . . .	23
5.26 eu.amidst.core.variables.DistType Enum Reference . . . . .	23
5.26.1 Detailed Description . . . . .	24
5.27 eu.amidst.core.database.filereaders.arffFileReader.DrillingAttributes Class Reference . . . . .	24
5.27.1 Detailed Description . . . . .	24
5.28 eu.amidst.core.modelstructure.DynamicBayesianNetwork Class Reference . . . . .	24
5.28.1 Detailed Description . . . . .	25
5.28.2 Member Function Documentation . . . . .	25
5.28.2.1 initializeDistributions . . . . .	25
5.28.2.2 newDynamicBayesianNetwork . . . . .	25
5.29 eu.amidst.core.database.filereaders.DynamicDataInstance Class Reference . . . . .	25
5.29.1 Detailed Description . . . . .	26
5.30 eu.amidst.core.database.filereaders.DynamicDataOnDiskFromFile Class Reference . . . . .	26
5.30.1 Detailed Description . . . . .	26
5.30.2 Constructor & Destructor Documentation . . . . .	26
5.30.2.1 DynamicDataOnDiskFromFile . . . . .	26
5.31 eu.amidst.core.database.filereaders.DynamicDataOnMemoryFromFile Class Reference . . . . .	27
5.31.1 Detailed Description . . . . .	27
5.32 eu.amidst.core.variables.DynamicVariables Class Reference . . . . .	27
5.32.1 Detailed Description . . . . .	27
5.32.2 Constructor & Destructor Documentation . . . . .	28
5.32.2.1 DynamicVariables . . . . .	28
5.33 eu.amidst.core.exponentialfamily.EF_BaseDistribution_MultinomialParents< E extends EF_<> Distribution > Class Reference . . . . .	28
5.34 eu.amidst.core.exponentialfamily.EF_ConditionalDistribution Class Reference . . . . .	28
5.35 eu.amidst.core.exponentialfamily.EF_Distribution Class Reference . . . . .	29
5.35.1 Detailed Description . . . . .	30
5.35.2 Member Function Documentation . . . . .	30
5.35.2.1 getVariable . . . . .	30
5.35.3 Member Data Documentation . . . . .	30
5.35.3.1 var . . . . .	30
5.36 eu.amidst.core.exponentialfamily.EF_DistributionBuilder Class Reference . . . . .	30
5.36.1 Detailed Description . . . . .	30

5.37 eu.amidst.core.exponentialfamily.EF_Multinomial Class Reference . . . . .	30
5.37.1 Detailed Description . . . . .	31
5.37.2 Constructor & Destructor Documentation . . . . .	31
5.37.2.1 EF_Multinomial . . . . .	31
5.38 eu.amidst.core.exponentialfamily.EF_Normal Class Reference . . . . .	31
5.38.1 Detailed Description . . . . .	32
5.39 eu.amidst.core.exponentialfamily.EF_UnivariateDistribution Class Reference . . . . .	32
5.39.1 Detailed Description . . . . .	33
5.40 eu.amidst.core.database.filereaders.arffFileReader.Keys Class Reference . . . . .	33
5.40.1 Detailed Description . . . . .	33
5.41 eu.amidst.Main Class Reference . . . . .	33
5.42 eu.amidst.core.exponentialfamily.MomentParameters Class Reference . . . . .	33
5.42.1 Detailed Description . . . . .	34
5.43 eu.amidst.core.distribution.Multinomial Class Reference . . . . .	34
5.43.1 Detailed Description . . . . .	34
5.43.2 Constructor & Destructor Documentation . . . . .	34
5.43.2.1 Multinomial . . . . .	34
5.43.3 Member Function Documentation . . . . .	35
5.43.3.1 getLogProbability . . . . .	35
5.43.3.2 getProbabilities . . . . .	35
5.43.3.3 getProbabilityOfState . . . . .	35
5.43.3.4 setProbabilities . . . . .	35
5.43.3.5 setProbabilityOfState . . . . .	35
5.44 eu.amidst.core.distribution.Multinomial_MultinomialParents Class Reference . . . . .	36
5.44.1 Detailed Description . . . . .	36
5.44.2 Constructor & Destructor Documentation . . . . .	36
5.44.2.1 Multinomial_MultinomialParents . . . . .	36
5.44.3 Member Function Documentation . . . . .	37
5.44.3.1 getLogConditionalProbability . . . . .	37
5.44.3.2 getMultinomial . . . . .	37
5.44.3.3 setMultinomial . . . . .	37
5.44.3.4 setMultinomial . . . . .	37
5.45 eu.amidst.core.utils.MultinomialIndex Class Reference . . . . .	38
5.45.1 Detailed Description . . . . .	38
5.45.2 Member Function Documentation . . . . .	38
5.45.2.1 getIndexFromVariableAssignment . . . . .	38
5.45.2.2 getIndexFromVariableAssignment . . . . .	40
5.45.2.3 getIndexFromVariableAssignment . . . . .	40
5.45.2.4 getNumberOfPossibleAssignments . . . . .	40
5.45.2.5 getVariableAssignmentFromIndex . . . . .	40

5.46 eu.amidst.core.potential.MultivariateGaussian Class Reference . . . . .	41
5.46.1 Detailed Description . . . . .	41
5.47 eu.amidst.core.potential.MultivariateGaussianCF Class Reference . . . . .	41
5.47.1 Detailed Description . . . . .	42
5.48 eu.amidst.core.exponentialfamily.NaturalParameters Class Reference . . . . .	42
5.48.1 Detailed Description . . . . .	42
5.49 eu.amidst.core.database.filereaders.NextDynamicDataInstance Class Reference . . . . .	42
5.49.1 Detailed Description . . . . .	42
5.50 eu.amidst.core.distribution.Normal Class Reference . . . . .	43
5.50.1 Detailed Description . . . . .	43
5.50.2 Constructor & Destructor Documentation . . . . .	43
5.50.2.1 Normal . . . . .	43
5.50.3 Member Function Documentation . . . . .	44
5.50.3.1 getLogProbability . . . . .	44
5.50.3.2 getMean . . . . .	44
5.50.3.3 getProbability . . . . .	44
5.50.3.4 getSd . . . . .	44
5.50.3.5 setMean . . . . .	44
5.50.3.6 setSd . . . . .	45
5.51 eu.amidst.core.distribution.Normal_MultinomialNormalParents Class Reference . . . . .	45
5.51.1 Detailed Description . . . . .	45
5.51.2 Constructor & Destructor Documentation . . . . .	46
5.51.2.1 Normal_MultinomialNormalParents . . . . .	46
5.51.3 Member Function Documentation . . . . .	46
5.51.3.1 getLogConditionalProbability . . . . .	46
5.51.3.2 getNormal_NormalParentsDistribution . . . . .	46
5.51.3.3 setNormal_NormalParentsDistribution . . . . .	46
5.51.3.4 setNormal_NormalParentsDistribution . . . . .	47
5.52 eu.amidst.core.distribution.Normal_MultinomialParents Class Reference . . . . .	47
5.52.1 Detailed Description . . . . .	47
5.52.2 Constructor & Destructor Documentation . . . . .	48
5.52.2.1 Normal_MultinomialParents . . . . .	48
5.52.3 Member Function Documentation . . . . .	48
5.52.3.1 getLogConditionalProbability . . . . .	48
5.52.3.2 getNormal . . . . .	48
5.52.3.3 setNormal . . . . .	48
5.52.3.4 setNormal . . . . .	49
5.53 eu.amidst.core.distribution.Normal_NormalParents Class Reference . . . . .	49
5.53.1 Detailed Description . . . . .	49
5.53.2 Constructor & Destructor Documentation . . . . .	50

5.53.2.1	Normal_NormalParents	50
5.53.3	Member Function Documentation	50
5.53.3.1	getCoeffParents	50
5.53.3.2	getIntercept	50
5.53.3.3	getLogConditionalProbability	50
5.53.3.4	getSd	51
5.53.3.5	getUnivariateNormal	51
5.53.3.6	setCoeffParents	51
5.53.3.7	setIntercept	51
5.53.3.8	setSd	51
5.54	eu.amidst.core.huginlink.Others Class Reference	51
5.54.1	Detailed Description	51
5.55	eu.amidst.core.modelstructure.ParentSet Class Reference	52
5.55.1	Detailed Description	52
5.55.2	Member Function Documentation	52
5.55.2.1	blockParents	52
5.56	eu.amidst.core.potential.Potential Interface Reference	52
5.56.1	Detailed Description	52
5.57	eu.amidst.core.potential.PotentialFunctional Class Reference	53
5.57.1	Detailed Description	53
5.58	eu.amidst.core.potential.PotentialTable Class Reference	53
5.58.1	Detailed Description	53
5.59	eu.amidst.core.database.dynamics.SequenceDataStream Interface Reference	54
5.59.1	Detailed Description	54
5.60	eu.amidst.core.database.dynamics.readers.SequenceDataStreamReaderFromFile Class Reference	54
5.60.1	Detailed Description	54
5.61	eu.amidst.core.database.dynamics.SequenceStreamWindow Interface Reference	54
5.61.1	Detailed Description	55
5.62	eu.amidst.core.variables.StateSpaceType Enum Reference	55
5.62.1	Detailed Description	55
5.63	eu.amidst.core.database.filereaders.StaticDataInstance Class Reference	55
5.63.1	Detailed Description	55
5.64	eu.amidst.core.database.filereaders.StaticDataOnDiskFromFile Class Reference	56
5.64.1	Detailed Description	56
5.65	eu.amidst.core.database.filereaders.StaticDataOnMemoryFromFile Class Reference	56
5.65.1	Detailed Description	56
5.66	eu.amidst.core.variables.StaticVariables Class Reference	57
5.66.1	Detailed Description	57
5.66.2	Constructor & Destructor Documentation	57
5.66.2.1	StaticVariables	57

5.66.2.2	StaticVariables	57
5.67	eu.amidst.core.exponentialfamily.SufficientStatistics Class Reference	57
5.67.1	Detailed Description	58
5.68	eu.amidst.core.distribution.UnivariateDistribution Class Reference	58
5.68.1	Detailed Description	58
5.68.2	Member Function Documentation	58
5.68.2.1	getLogProbability	58
5.68.2.2	getProbability	59
5.69	eu.amidst.core.utils.Utils Class Reference	59
5.69.1	Detailed Description	59
5.70	eu.amidst.core.variables.Variable Interface Reference	59
5.70.1	Detailed Description	60
5.71	eu.amidst.core.variables.VariableBuilder Class Reference	60
5.71.1	Detailed Description	60
5.72	eu.amidst.core.utils.Vector Interface Reference	60
5.72.1	Detailed Description	61
5.73	eu.amidst.core.database.filereaders.arffWekaReader.WekaDataFileReader Class Reference	61
5.73.1	Detailed Description	61



## Chapter 1

# Namespace Index

### 1.1 Packages

Here are the packages with brief descriptions (if available):

<a href="#">eu.amidst.core.database</a>	7
<a href="#">eu.amidst.core.database.filereaders</a>	7
<a href="#">eu.amidst.core.distribution</a>	8
<a href="#">eu.amidst.core.exponentialfamily</a>	9
<a href="#">eu.amidst.core.variables</a>	9



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

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

eu.amidst.core.database.filereaders.arffFileReader.ArffDataStream . . . . .	11
eu.amidst.core.variables.Assignment . . . . .	12
eu.amidst.core.database.Attribute . . . . .	12
eu.amidst.core.database.Attributes . . . . .	13
eu.amidst.core.database.filereaders.arffFileReader.DrillingAttributes . . . . .	24
eu.amidst.core.modelstructure.BayesianNetwork . . . . .	13
eu.amidst.core.database.dynamics.BucketSequenceData . . . . .	14
eu.amidst.core.database.dynamics.BucketSequenceStream . . . . .	14
eu.amidst.staticmodelling.models.Classifier . . . . .	15
eu.amidst.core.huginlink.ConverterToAMIDST . . . . .	18
eu.amidst.core.huginlink.ConverterToHugin . . . . .	18
eu.amidst.core.modelstructure.DAG . . . . .	18
eu.amidst.core.database.filereaders.DataFileReader . . . . .	19
eu.amidst.core.database.filereaders.arffWekaReader.WekaDataFileReader . . . . .	61
eu.amidst.core.database.DataInstance . . . . .	19
eu.amidst.core.database.filereaders.DynamicDataInstance . . . . .	25
eu.amidst.core.database.filereaders.StaticDataInstance . . . . .	55
eu.amidst.core.database.DataOnDisk . . . . .	20
eu.amidst.core.database.filereaders.DynamicDataOnDiskFromFile . . . . .	26
eu.amidst.core.database.filereaders.DynamicDataOnMemoryFromFile . . . . .	27
eu.amidst.core.database.filereaders.StaticDataOnDiskFromFile . . . . .	56
eu.amidst.core.database.filereaders.StaticDataOnMemoryFromFile . . . . .	56
eu.amidst.core.database.DataOnMemory . . . . .	20
eu.amidst.core.database.filereaders.DynamicDataOnMemoryFromFile . . . . .	27
eu.amidst.core.database.filereaders.StaticDataOnMemoryFromFile . . . . .	56
eu.amidst.core.database.DataOnStream . . . . .	21
eu.amidst.core.database.filereaders.DynamicDataOnDiskFromFile . . . . .	26
eu.amidst.core.database.filereaders.DynamicDataOnMemoryFromFile . . . . .	27
eu.amidst.core.database.filereaders.StaticDataOnDiskFromFile . . . . .	56
eu.amidst.core.database.filereaders.StaticDataOnMemoryFromFile . . . . .	56
eu.amidst.core.database.filereaders.DataRow . . . . .	21
eu.amidst.core.database.filereaders.arffWekaReader.DataRowWeka . . . . .	22
eu.amidst.core.database.filereaders.DataRowMissing . . . . .	21
eu.amidst.core.distribution.Distribution . . . . .	22
eu.amidst.core.distribution.ConditionalDistribution . . . . .	15
eu.amidst.core.distribution.Multinomial_MultinomialParents . . . . .	36

eu.amidst.core.distribution.Normal_MultinomialNormalParents . . . . .	45
eu.amidst.core.distribution.Normal_MultinomialParents . . . . .	47
eu.amidst.core.distribution.Normal_NormalParents . . . . .	49
eu.amidst.core.distribution.UnivariateDistribution . . . . .	58
eu.amidst.core.distribution.Multinomial . . . . .	34
eu.amidst.core.distribution.Normal . . . . .	43
eu.amidst.core.distribution.DistributionBuilder . . . . .	23
eu.amidst.core.variables.DistType . . . . .	23
eu.amidst.core.modelstructure.DynamicBayesianNetwork . . . . .	24
eu.amidst.core.variables.DynamicVariables . . . . .	27
eu.amidst.core.exponentialfamily.EF_Distribution . . . . .	29
eu.amidst.core.exponentialfamily.EF_ConditionalDistribution . . . . .	28
eu.amidst.core.exponentialfamily.EF_BaseDistribution_MultinomialParents< E extends EF_< Distribution > . . . . .	28
eu.amidst.core.exponentialfamily.EF_UnivariateDistribution . . . . .	32
eu.amidst.core.exponentialfamily.EF_Multinomial . . . . .	30
eu.amidst.core.exponentialfamily.EF_Normal . . . . .	31
eu.amidst.core.exponentialfamily.EF_DistributionBuilder . . . . .	30
eu.amidst.core.database.filereaders.arffFileReader.Keys . . . . .	33
eu.amidst.Main . . . . .	33
eu.amidst.core.utils.MultinomialIndex . . . . .	38
eu.amidst.core.potential.MultivariateGaussian . . . . .	41
eu.amidst.core.database.filereaders.NextDynamicDataInstance . . . . .	42
eu.amidst.core.huginlink.Others . . . . .	51
eu.amidst.core.modelstructure.ParentSet . . . . .	52
eu.amidst.core.potential.Potential . . . . .	52
eu.amidst.core.potential.ConstantPotential . . . . .	17
eu.amidst.core.potential.MultivariateGaussianCF . . . . .	41
eu.amidst.core.potential.PotentialFunctional . . . . .	53
eu.amidst.core.potential.PotentialTable . . . . .	53
eu.amidst.core.database.dynamics.SequenceDataStream . . . . .	54
eu.amidst.core.database.dynamics.readers.SequenceDataStreamReaderFromFile . . . . .	54
eu.amidst.core.database.dynamics.SequenceStreamWindow . . . . .	54
eu.amidst.core.variables.StateSpaceType . . . . .	55
eu.amidst.core.variables.StaticVariables . . . . .	57
eu.amidst.core.utils.Utils . . . . .	59
eu.amidst.core.variables.Variable . . . . .	59
eu.amidst.core.variables.VariableBuilder . . . . .	60
eu.amidst.core.utils.Vector . . . . .	60
eu.amidst.core.utils.ArrayVector . . . . .	12
eu.amidst.core.exponentialfamily.MomentParameters . . . . .	33
eu.amidst.core.exponentialfamily.NaturalParameters . . . . .	42
eu.amidst.core.exponentialfamily.SufficientStatistics . . . . .	57
IOException . . . . .	
eu.amidst.core.database.filereaders.arffFileReader.ArffParserException . . . . .	11

# Chapter 3

## Class Index

### 3.1 Class List

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

eu.amidst.core.database.filereaders.arffFileReader.ArffDataStream . . . . .	11
eu.amidst.core.database.filereaders.arffFileReader.ArffParserException . . . . .	11
eu.amidst.core.utils.ArrayVector . . . . .	12
eu.amidst.core.variables.Assignment . . . . .	12
eu.amidst.core.database.Attribute . . . . .	12
eu.amidst.core.database.Attributes . . . . .	13
eu.amidst.core.modelstructure.BayesianNetwork . . . . .	13
eu.amidst.core.database.dynamics.BucketSequenceData . . . . .	14
eu.amidst.core.database.dynamics.BucketSequenceStream . . . . .	14
eu.amidst.staticmodelling.models.Classifier . . . . .	15
eu.amidst.core.distribution.ConditionalDistribution . . . . .	15
eu.amidst.core.potential.ConstantPotential . . . . .	17
eu.amidst.core.huginlink.ConverterToAMIDST . . . . .	18
eu.amidst.core.huginlink.ConverterToHugin . . . . .	18
eu.amidst.core.modelstructure.DAG . . . . .	18
eu.amidst.core.database.filereaders.DataFileReader . . . . .	19
eu.amidst.core.database.DataInstance . . . . .	19
eu.amidst.core.database.DataOnDisk . . . . .	20
eu.amidst.core.database.DataOnMemory . . . . .	20
eu.amidst.core.database.DataOnStream . . . . .	21
eu.amidst.core.database.filereaders.DataRow . . . . .	21
eu.amidst.core.database.filereaders.DataRowMissing . . . . .	21
eu.amidst.core.database.filereaders.arffWekaReader.DataRowWeka . . . . .	22
eu.amidst.core.distribution.Distribution . . . . .	22
eu.amidst.core.distribution.DistributionBuilder . . . . .	23
eu.amidst.core.variables.DistType . . . . .	23
eu.amidst.core.database.filereaders.arffFileReader.DrillingAttributes . . . . .	24
eu.amidst.core.modelstructure.DynamicBayesianNetwork . . . . .	24
eu.amidst.core.database.filereaders.DynamicDataInstance . . . . .	25
eu.amidst.core.database.filereaders.DynamicDataOnDiskFromFile . . . . .	26
eu.amidst.core.database.filereaders.DynamicDataOnMemoryFromFile . . . . .	27
eu.amidst.core.variables.DynamicVariables . . . . .	27
eu.amidst.core.exponentialfamily.EF_BaseDistribution_MultinomialParents< E extends EF_Distribution > . . . . .	28
eu.amidst.core.exponentialfamily.EF_ConditionalDistribution . . . . .	28
eu.amidst.core.exponentialfamily.EF_Distribution . . . . .	29
eu.amidst.core.exponentialfamily.EF_DistributionBuilder . . . . .	30
eu.amidst.core.exponentialfamily.EF_Multinomial . . . . .	30
eu.amidst.core.exponentialfamily.EF_Normal . . . . .	31

eu.amidst.core.exponentialfamily.EF_UnivariateDistribution . . . . .	32
eu.amidst.core.database.filereaders.arffFileReader.Keys . . . . .	33
eu.amidst.Main . . . . .	33
eu.amidst.core.exponentialfamily.MomentParameters . . . . .	33
eu.amidst.core.distribution.Multinomial . . . . .	34
eu.amidst.core.distribution.Multinomial_MultinomialParents . . . . .	36
eu.amidst.core.utils.MultinomialIndex . . . . .	38
eu.amidst.core.potential.MultivariateGaussian . . . . .	41
eu.amidst.core.potential.MultivariateGaussianCF . . . . .	41
eu.amidst.core.exponentialfamily.NaturalParameters . . . . .	42
eu.amidst.core.database.filereaders.NextDynamicDataInstance . . . . .	42
eu.amidst.core.distribution.Normal . . . . .	43
eu.amidst.core.distribution.Normal_MultinomialNormalParents . . . . .	45
eu.amidst.core.distribution.Normal_MultinomialParents . . . . .	47
eu.amidst.core.distribution.Normal_NormalParents . . . . .	49
eu.amidst.core.huginlink.Others . . . . .	51
eu.amidst.core.modelstructure.ParentSet . . . . .	52
eu.amidst.core.potential.Potential . . . . .	52
eu.amidst.core.potential.PotentialFunctional . . . . .	53
eu.amidst.core.potential.PotentialTable . . . . .	53
eu.amidst.core.database.dynamics.SequenceDataStream . . . . .	54
eu.amidst.core.database.dynamics.readers.SequenceDataStreamReaderFromFile . . . . .	54
eu.amidst.core.database.dynamics.SequenceStreamWindow . . . . .	54
eu.amidst.core.variables.StateSpaceType . . . . .	55
eu.amidst.core.database.filereaders.StaticDataInstance . . . . .	55
eu.amidst.core.database.filereaders.StaticDataOnDiskFromFile . . . . .	56
eu.amidst.core.database.filereaders.StaticDataOnMemoryFromFile . . . . .	56
eu.amidst.core.variables.StaticVariables . . . . .	57
eu.amidst.core.exponentialfamily.SufficientStatistics . . . . .	57
eu.amidst.core.distribution.UnivariateDistribution . . . . .	58
eu.amidst.core.utils.Utils . . . . .	59
eu.amidst.core.variables.Variable . . . . .	59
eu.amidst.core.variables.VariableBuilder . . . . .	60
eu.amidst.core.utils.Vector . . . . .	60
eu.amidst.core.database.filereaders.arffWekaReader.WekaDataFileReader . . . . .	61

## Chapter 4

# Namespace Documentation

### 4.1 Package eu.amidst.core.database

#### Packages

- package [filereaders](#)

#### Classes

- class [Attribute](#)
- class [Attributes](#)
- interface [DataInstance](#)
- interface [DataOnDisk](#)
- interface [DataOnMemory](#)
- interface [DataOnStream](#)

#### 4.1.1 Detailed Description

ISSUE LIST \*\*\*\*\*

1. The number of states should be parsed and stored.

\*\*\*\*\* ISSUE LIST \*\*\*\*\*

1. (Andres) Add a "close" method to close the possible linked file or whatever.

### 4.2 Package eu.amidst.core.database.filereaders

#### Classes

- interface [DataFileReader](#)
  - interface [DataRow](#)
  - class [DataRowMissing](#)
  - class [DynamicDataInstance](#)
  - class [DynamicDataOnDiskFromFile](#)
  - class [DynamicDataOnMemoryFromFile](#)
  - class [NextDynamicDataInstance](#)
-

- class [StaticDataInstance](#)
- class [StaticDataOnDiskFromFile](#)
- class [StaticDataOnMemoryFromFile](#)

#### 4.2.1 Detailed Description

ISSUE LIST \*\*\*\*\*

1. We could eliminate the if(timerIDcounter == 1) in nextDataInstance\_NoTimeID\_NoSeq if we maintain a future [DataRow](#) (we read an extra row in advance). Then we would need the method public boolean isNull(){ return (present==null || past==null); }

### 4.3 Package eu.amidst.core.distribution

#### Classes

- class [ConditionalDistribution](#)
- class [Distribution](#)
- class [DistributionBuilder](#)
- class [Multinomial](#)
- class [Multinomial\\_MultinomialParents](#)
- class [Normal](#)
- class [Normal\\_MultinomialNormalParents](#)
- class [Normal\\_MultinomialParents](#)
- class [Normal\\_NormalParents](#)
- class [UnivariateDistribution](#)

#### 4.3.1 Detailed Description

ISSUE LIST \*\*\*\*\*

1. In general, should we clone attributes in the constructor to avoid bad uses of input variables later on?
2. How are we going to update the probabilities? Value by value? Or directly with the whole set of probabilities? or both? Two methods are included: `setProbabilities(double[] probabilities)` and `setProbabilityOfState(int index, double value)`
3. Is needed the method `setProbabilityOfState` ?

\*\*\*\*\* ISSUE LIST \*\*\*\*\*

1. `getConditioningVariables` change to `getParentsVariables()`

ISSUE LIST \*\*\*\*\*

1. Do we need here the min and max of the variable, for instance, to check that the input value in `computeProbabilityOf(value)` is in the range [min,max]?

ISSUES \*\*\*\*\*

1. CODING: - `this.multinomialParents` or `multinomialParents`? Common criteria.  
- methods are ordered? alphabetically?

## ISSUE LIST \*\*\*\*\*

1. In the constructor, should we initialize the CLG attributes in this way?
2. The name of the method computeProbabilityOf(..) is a bit confusing for continuous domains. It does not compute probabilities but the value for the density function which is not a probability. However as this class implements this method of [ConditionalDistribution](#), we could leave like this.
3. QAPlug gives a warning when using the same name for a attribute and a given argument, e.g. this.var = var

## 4.4 Package eu.amidst.core.exponentialfamily

### Classes

- class [EF\\_BaseDistribution\\_MultinomialParents](#)< E extends EF\_Distribution >
- class [EF\\_ConditionalDistribution](#)
- class [EF\\_Distribution](#)
- class [EF\\_DistributionBuilder](#)
- class [EF\\_Multinomial](#)
- class [EF\\_Normal](#)
- class [EF\\_UnivariateDistribution](#)
- class [MomentParameters](#)
- class [NaturalParameters](#)
- class [SufficientStatistics](#)

### 4.4.1 Detailed Description

## \*\*\*\*\* ISSUE LIST \*\*\*\*\*

1. getConditioningVariables change to getParentsVariables()

## ISSUE LIST \*\*\*\*\*

1. Make SufficientStatics an static class to avoid the creation of an object in each call to getSuffStatistics();
2. Make naturalParameters and momentParameters statics?

## 4.5 Package eu.amidst.core.variables

### Classes

- class [Assignment](#)
- enum [DistType](#)
- class [DynamicVariables](#)
- enum [StateSpaceType](#)
- class [StaticVariables](#)
- interface [Variable](#)
- class [VariableBuilder](#)

#### 4.5.1 Detailed Description

ISSUE LIST \*\*\*\*\*

1. Rename to [DynamicVariables](#)
2. We can/should remove all setters from VariableImplementation right?
3. Is there any need for the field attrs? It is only used in the constructor.
4. If the fields in VariableImplementation are all objects then the TemporalClone only contains pointers, which would ensure consistency, although we are not planing to modify these values.

## Chapter 5

# Class Documentation

## 5.1 eu.amidst.core.database.filereaders.arffFileReader.ArffDataStream Class Reference

### 5.1.1 Detailed Description

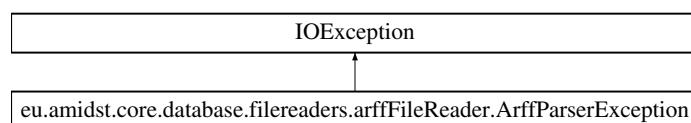
Created by sigveh on 10/7/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/database/filereaders/arffFileReader/ArffDataStream.java

## 5.2 eu.amidst.core.database.filereaders.arffFileReader.ArffParserException Class Reference

Inheritance diagram for eu.amidst.core.database.filereaders.arffFileReader.ArffParserException:



### Public Member Functions

- **ArffParserException** (String message)

### 5.2.1 Detailed Description

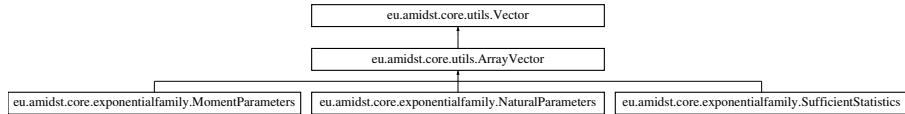
Created by sigveh on 10/8/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/database/filereaders/arffFileReader/ArffParserException.java

### 5.3 eu.amidst.core.utils.ArrayVector Class Reference

Inheritance diagram for eu.amidst.core.utils.ArrayVector:



#### Public Member Functions

- **ArrayVector** (int size)
- **ArrayVector** (double[] vec)
- double **get** (int i)
- void **set** (int i, double val)
- int **size** ()

#### Additional Inherited Members

##### 5.3.1 Detailed Description

Created by andresmasegosa on 12/11/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/utils/ArrayVector.java

### 5.4 eu.amidst.core.variables.Assignment Class Reference

#### Public Member Functions

- **Assignment** (int nOfVars)
- double **getValue** ([Variable](#) key)
- void **setValue** ([Variable](#) var, Double value)
- Set< Map.Entry< [Variable](#), Double > > **entrySet** ()

##### 5.4.1 Detailed Description

Created by [ana@cs.aau.dk](mailto:ana@cs.aau.dk) on 03/11/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/variables/Assignment.java

### 5.5 eu.amidst.core.database.Attribute Class Reference

#### Public Member Functions

- **Attribute** (int index, String name, String unit, [StateSpaceType](#) stateSpaceType, int numberofStates)

- **Attribute** (int index, String name, [StateSpaceType](#) stateSpaceType, int numberOfStates)
- **Attribute** (String name, [StateSpaceType](#) stateSpaceType, int numberOfStates)
- int **getIndex** ()
- String **getUnit** ()
- String **getName** ()
- [StateSpaceType](#) **getStateSpaceType** ()
- int **getNumberOfStates** ()
- boolean **equals** (Object o)
- int **hashCode** ()

### 5.5.1 Detailed Description

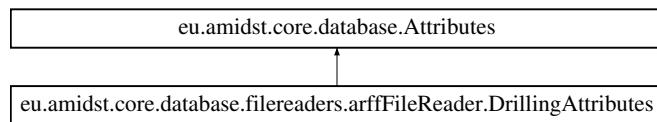
Created by sigveh on 10/20/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/database/Attribute.java

## 5.6 eu.amidst.core.database.Attributes Class Reference

Inheritance diagram for eu.amidst.core.database.Attributes:



### Public Member Functions

- **Attributes** (List<> [Attribute](#) attributes)
- List<> [Attribute](#) > **getList** ()
- List<> [Attribute](#) > **getListExceptTimeAndSeq** ()
- void **print** ()
- [Attribute](#) **getAttributeByName** (String name)

### 5.6.1 Detailed Description

Created by sigveh on 10/16/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/database/Attributes.java

## 5.7 eu.amidst.core.modelstructure.BayesianNetwork Class Reference

### Public Member Functions

- void **setDistribution** ([Variable](#) var, [ConditionalDistribution](#) distribution)
- int **getNumberOfVars** ()
- [StaticVariables](#) **getStaticVariables** ()
- [DAG](#) **getDAG** ()
- List<> [Variable](#) > **getVariables** ()

## Static Public Member Functions

- static [BayesianNetwork](#) **newBayesianNetwork** ([DAG](#) dag)

### 5.7.1 Detailed Description

Created by afa on 02/07/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/modelstructure/BayesianNetwork.java

## 5.8 eu.amidst.core.database.dynamics.BucketSequenceData Interface Reference

### Public Member Functions

- int **getMarkovOrder** ()
- boolean **hasMoreData** ()
- [DynamicDataInstance](#) **nextSequenceData** ()
- boolean **isResetable** ()
- void **reset** ()

### 5.8.1 Detailed Description

Created by afa on 03/07/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/database/dynamics/BucketSequenceData.java

## 5.9 eu.amidst.core.database.dynamics.BucketSequenceStream Interface Reference

### Public Member Functions

- [Attributes](#) **getDynamicAttributes** ()
- int **getMarkovOrder** ()
- boolean **hasMoreData** ()
- [BucketSequenceData](#) **nextBucketSequenceData** ()
- boolean **isResetable** ()
- void **reset** ()

### 5.9.1 Detailed Description

Created by afa on 03/07/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/database/dynamics/BucketSequenceStream.java

## 5.10 eu.amidst.staticmodelling.models.Classifier Interface Reference

### Public Member Functions

- `double[] predict (DataInstance instance)`
- `int getClassVarID ()`
- `void setClassVarID (int varID)`

#### 5.10.1 Detailed Description

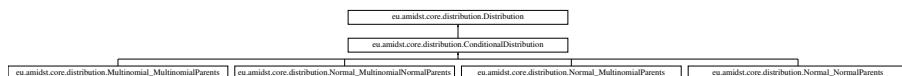
Created by afa on 02/07/14.

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

- `/Users/ana/Documents/core/src/main/java/eu/amidst/staticmodelling/models/Classifier.java`

## 5.11 eu.amidst.core.distribution.ConditionalDistribution Class Reference

Inheritance diagram for eu.amidst.core.distribution.ConditionalDistribution:



### Public Member Functions

- `List< Variable > getConditioningVariables ()`
- `double getConditionalProbability (Assignment assignment)`
- `abstract double getLogConditionalProbability (Assignment assignment)`

### Protected Attributes

- `List< Variable > parents`

#### 5.11.1 Detailed Description

This interface generalizes the set of possible conditional distributions.

#### Author

Antonio Fernández

#### Version

1.0

#### Since

2014-11-3

## 5.11.2 Member Function Documentation

### 5.11.2.1 double eu.amidst.core.distribution.ConditionalDistribution.getConditionalProbability ( *Assignment assignment* )

Evaluates the conditional distribution given a value of the variable and an assignment of the parents.

**Parameters**

<code>assignment</code>	An Assignment for the parents.
-------------------------	--------------------------------

**Returns**

A double value with the evaluated distribution.

**5.11.2.2 List<Variable> eu.amidst.core.distribution.ConditionalDistribution.getConditioningVariables( )**

Gets the set of conditioning variables

**Returns**

An unmodifiable List object with the set of conditioning variables.

**5.11.2.3 abstract double eu.amidst.core.distribution.ConditionalDistribution.getLogConditionalProbability( Assignment assignment ) [abstract]**

Evaluates the conditional distribution given a value of the variable and an assignment of the parents.

**Parameters**

<code>assignment</code>	An Assignment for the parents.
-------------------------	--------------------------------

**Returns**

A double value with the logarithm of the evaluated distribution.

**5.11.3 Member Data Documentation****5.11.3.1 List<Variable> eu.amidst.core.distribution.ConditionalDistribution.parents [protected]**

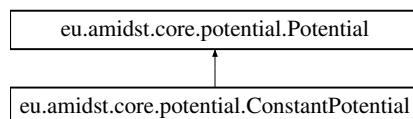
The list of parents of the variable

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/distribution/ConditionalDistribution.java

**5.12 eu.amidst.core.potential.ConstantPotential Class Reference**

Inheritance diagram for eu.amidst.core.potential.ConstantPotential:

**Public Member Functions**

- **ConstantPotential** (double val)
- void **setVariables** (List variables)
- List **getVariables** ()
- void **combine** (Potential pot)
- void **marginalize** (List variables)

### 5.12.1 Detailed Description

Created by andresmasegosa on 28/08/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/potential/ConstantPotential.java

## 5.13 eu.amidst.core.huginlink.ConverterToAMIDST Class Reference

### Public Member Functions

- **ConverterToAMIDST** (Domain huginNetwork)
- **BayesianNetwork getAmidstNetwork ()**
- void **setNodes ()**
- void **setStructure ()**
- void **setMultinomial\_MultinomialParents** (Node huginVar)
- void **setDistributions** (NodeList huginNodes)

### 5.13.1 Detailed Description

Created by afa on 14/11/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/huginlink/ConverterToAMIDST.java

## 5.14 eu.amidst.core.huginlink.ConverterToHugin Class Reference

### Public Member Functions

- Domain **getHuginNetwork ()**
- void **setNodes** (List< Variable > amidstVars)
- void **setStructure** (DAG dag)
- void **setMultinomial\_MultinomialParents** (ConditionalDistribution dist)
- void **setNormal\_NormalParents** (ConditionalDistribution dist, int assign\_i)
- void **setNormal** (Normal dist, int i)
- void **setNormal\_MultinomialParents** (ConditionalDistribution dist)
- void **setNormal\_MultinomialNormalParents** (ConditionalDistribution dist)
- void **setDistributions** (BayesianNetwork bn)
- void **setBayesianNetwork** (BayesianNetwork bn)

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/huginlink/ConverterToHugin.java

## 5.15 eu.amidst.core.modelstructure.DAG Class Reference

### Public Member Functions

- **DAG** (**StaticVariables** variables)
- **StaticVariables getVariables ()**
- **ParentSet getParentSet** (Variable var)
- boolean **containCycles ()**

### 5.15.1 Detailed Description

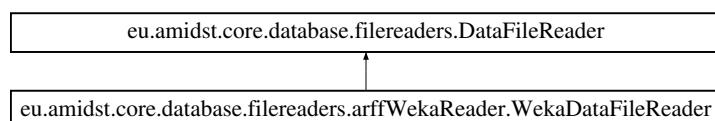
Created by Hanen on 13/11/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/modelstructure/DAG.java

## 5.16 eu.amidst.core.database.filereaders.DataFileReader Interface Reference

Inheritance diagram for eu.amidst.core.database.filereaders.DataFileReader:



### Public Member Functions

- **Attributes** `getAttributes ()`
- **DataRow** `nextDataRow ()`
- boolean `hasMoreDataRows ()`
- void `reset ()`
- boolean `doesItReadThisFileExtension` (String fileExtension)

### 5.16.1 Detailed Description

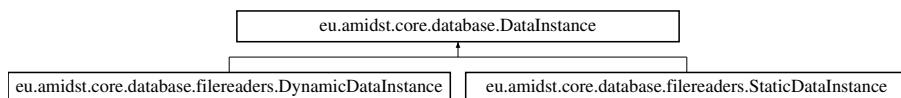
Created by andresmasegosa on 11/11/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/database/filereaders/DataFileReader.java

## 5.17 eu.amidst.core.database.DataInstance Interface Reference

Inheritance diagram for eu.amidst.core.database.DataInstance:



### Public Member Functions

- double `getValue` (**Variable** var)
- int `getSequenceID ()`
- int `getTimeID ()`

### 5.17.1 Detailed Description

Created by [ana@cs.aau.dk](mailto:ana@cs.aau.dk) on 10/11/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/database/DataInstance.java

## 5.18 eu.amidst.core.database.DataOnDisk Interface Reference

Inheritance diagram for eu.amidst.core.database.DataOnDisk:



### Public Member Functions

- **DataInstance** `nextDataInstance ()`
- boolean `hasMoreDataInstances ()`
- **Attributes** `getAttributes ()`
- void `restart ()`

### 5.18.1 Detailed Description

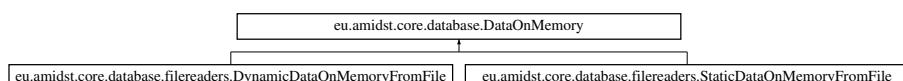
Created by afa on 02/07/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/database/DataOnDisk.java

## 5.19 eu.amidst.core.database.DataOnMemory Interface Reference

Inheritance diagram for eu.amidst.core.database.DataOnMemory:



### Public Member Functions

- int `getNumberOfDataInstances ()`
- **DataInstance** `getDataInstance (int i)`
- **Attributes** `getAttributes ()`

### 5.19.1 Detailed Description

Created by andresmasegosa on 11/11/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/database/DataOnMemory.java

## 5.20 eu.amidst.core.database.DataOnStream Interface Reference

Inheritance diagram for eu.amidst.core.database.DataOnStream:



### Public Member Functions

- **DataInstance** `nextDataInstance ()`
- boolean `hasMoreDataInstances ()`
- **Attributes** `getAttributes ()`

#### 5.20.1 Detailed Description

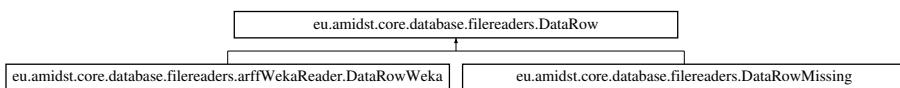
Created by afa on 02/07/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/database/DataOnStream.java

## 5.21 eu.amidst.core.database.filereaders.DataRow Interface Reference

Inheritance diagram for eu.amidst.core.database.filereaders.DataRow:



### Public Member Functions

- double `getValue (Attribute att)`

#### 5.21.1 Detailed Description

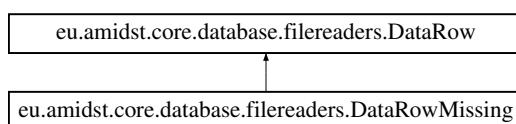
Created by andresmasegosa on 11/11/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/database/filereaders/DataRow.java

## 5.22 eu.amidst.core.database.filereaders.DataRowMissing Class Reference

Inheritance diagram for eu.amidst.core.database.filereaders.DataRowMissing:



## Public Member Functions

- double **getValue** ([Attribute](#) att)

### 5.22.1 Detailed Description

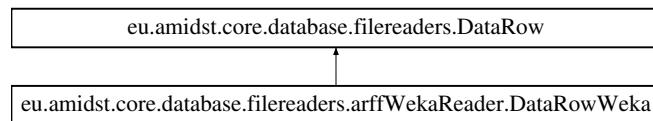
Created by [ana@cs.aau.dk](#) on 13/11/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/database/filereaders/DataRowMissing.java

## 5.23 eu.amidst.core.database.filereaders.arffWekaReader.DataRowWeka Class Reference

Inheritance diagram for eu.amidst.core.database.filereaders.arffWekaReader.DataRowWeka:



## Public Member Functions

- **DataRowWeka** (Instance `dataRow`)
- double **getValue** ([Attribute](#) att)

### 5.23.1 Detailed Description

Created by [ana@cs.aau.dk](#) on 14/11/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/database/filereaders/arffWekaReader/DataRowWeka.java

## 5.24 eu.amidst.core.distribution.Distribution Class Reference

Inheritance diagram for eu.amidst.core.distribution.Distribution:



## Public Member Functions

- **Variable getVariable ()**

### Protected Attributes

- [Variable var](#)

#### 5.24.1 Detailed Description

Created by afa on 12/11/14.

#### 5.24.2 Member Function Documentation

##### 5.24.2.1 Variable eu.amidst.core.distribution.Distribution.getVariable( )

Gets the variable of the distribution

Returns

A [Variable](#) object.

#### 5.24.3 Member Data Documentation

##### 5.24.3.1 Variable eu.amidst.core.distribution.Distribution.var [protected]

The variable of the distribution

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/distribution/Distribution.java

## 5.25 eu.amidst.core.distribution.DistributionBuilder Class Reference

### Static Public Member Functions

- static [ConditionalDistribution newDistribution](#) ([Variable](#) mainVar, List<[Variable](#)> conditioningVars)

#### 5.25.1 Detailed Description

Created by andresmasegosa on 11/11/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/distribution/DistributionBuilder.java

## 5.26 eu.amidst.core.variables.DistType Enum Reference

### Public Attributes

- **MULTINOMIAL**

### 5.26.1 Detailed Description

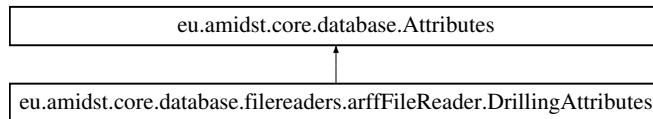
Created by Hanen on 05/11/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/variables/DistType.java

## 5.27 eu.amidst.core.database.filereaders.arffFileReader.DrillingAttributes Class Reference

Inheritance diagram for eu.amidst.core.database.filereaders.arffFileReader.DrillingAttributes:



### Public Member Functions

- Attribute **getMFI** ()
- Attribute **getRPM** ()
- Attribute **getSPP** ()
- List< Attribute > **getList** ()
- void **print** ()
- Attribute **getAttributeByName** (String name)

### 5.27.1 Detailed Description

Created by sigveh on 10/16/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/database/filereaders/arffFileReader/DrillingAttributes.java

## 5.28 eu.amidst.core.modelstructure.DynamicBayesianNetwork Class Reference

### Public Member Functions

- void **initializeDistributions** ()
- int **getNumberOfNodes** ()
- DynamicVariables **getDynamicVariables** ()
- Variable **getVariableById** (int varID)
- Variable **getTemporalCloneById** (int varID)
- Variable **getTemporalCloneFromVariable** (Variable variable)
- ParentSet **getParentSetTimeT** (Variable var)
- Distribution **getDistributionTimeT** (Variable var)
- ParentSet **getParentSetTime0** (Variable var)
- Distribution **getDistributionTime0** (Variable var)

**Static Public Member Functions**

- static [DynamicBayesianNetwork newDynamicBayesianNetwork \(DynamicVariables variables\)](#)

**5.28.1 Detailed Description**

This class implements a dynamic Bayesian network.

**Author**

[a.alvarez@ual.es](mailto:a.alvarez@ual.es), [andres@cs.aau.dk](mailto:andres@cs.aau.dk) & [ana@cs.aau.dk](mailto:ana@cs.aau.dk)

**Version**

1.0

**Since**

2014-07-3

**5.28.2 Member Function Documentation****5.28.2.1 void eu.amidst.core.modelstructure.DynamicBayesianNetwork.initializeDistributions ( )**

Initialize the Distributions of the variables based on their StateSpaceType

**5.28.2.2 static DynamicBayesianNetwork eu.amidst.core.modelstructure.DynamicBayesianNetwork.newDynamicBayesianNetwork ( DynamicVariables variables ) [static]**

The class public constructor, as a factory pattern

**Parameters**

<code>variables</code>	The variables or list of variables
------------------------	------------------------------------

**Returns**

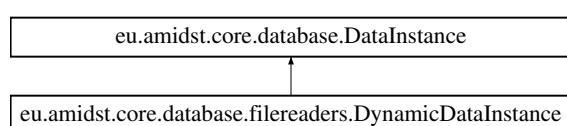
A [DynamicBayesianNetwork](#) with the given list of variables

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/modelstructure/DynamicBayesianNetwork.java

**5.29 eu.amidst.core.database.filereaders.DynamicDataInstance Class Reference**

Inheritance diagram for eu.amidst.core.database.filereaders.DynamicDataInstance:



## Public Member Functions

- **DynamicDataInstance** (`DataRow` `dataRowPast_`, `DataRow` `dataRowPresent_`, int `sequenceID_`, int `timeID_`)
- double **getValue** (`Variable` `var`)
- int **getSequenceID** ()
- int **getTimeID** ()

### 5.29.1 Detailed Description

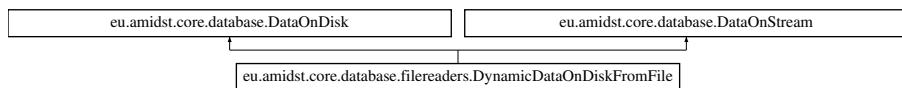
Created by andresmasegosa on 11/11/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/database/filereaders/DynamicDataInstance.java

## 5.30 eu.amidst.core.database.filereaders.DynamicDataOnDiskFromFile Class Reference

Inheritance diagram for eu.amidst.core.database.filereaders.DynamicDataOnDiskFromFile:



## Public Member Functions

- **DynamicDataOnDiskFromFile** (`DataFileReader` `reader`)
- **DataInstance** **nextDataInstance** ()
- boolean **hasMoreDataInstances** ()
- **Attributes** **getAttributes** ()
- void **restart** ()

### 5.30.1 Detailed Description

Created by `ana@cs.aau.dk` on 12/11/14.

### 5.30.2 Constructor & Destructor Documentation

#### 5.30.2.1 eu.amidst.core.database.filereaders.DynamicDataOnDiskFromFile.DynamicDataOnDiskFromFile ( `DataFileReader reader` )

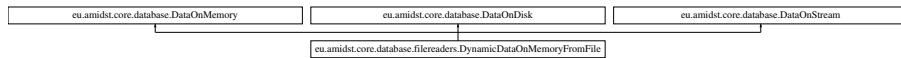
We read the two first rows now, to create the first couple in nextDataInstance

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/database/filereaders/DynamicDataOnDiskFromFile.java

## 5.31 eu.amidst.core.database.filereaders.DynamicDataOnMemoryFromFile Class Reference

Inheritance diagram for eu.amidst.core.database.filereaders.DynamicDataOnMemoryFromFile:



### Public Member Functions

- **DynamicDataOnMemoryFromFile** ([DataFileReader](#) reader)
- int **getNumberOfDataInstances** ()
- **DataInstance** **getDataInstance** (int i)
- **Attributes** **getAttributes** ()
- **DataInstance** **nextDataInstance** ()
- boolean **hasMoreDataInstances** ()
- void **restart** ()

#### 5.31.1 Detailed Description

Created by [ana@cs.aau.dk](#) on 12/11/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/database/filereaders/DynamicDataOnMemoryFromFile.java

## 5.32 eu.amidst.core.variables.DynamicVariables Class Reference

### Public Member Functions

- **DynamicVariables** ([Attributes](#) atts)
- **DynamicVariables** ([Attributes](#) atts, [HashMap](#)< [Attribute](#), [DistType](#) > typeDists)
- **Variable** **getTemporalCloneFromVariable** ([Variable](#) var)
- **Variable** **getVariableFromTemporalClone** ([Variable](#) var)
- **Variable** **addHiddenVariable** ([VariableBuilder](#) builder)
- [List](#)< [Variable](#) > **getVariables** ()
- [List](#)< [Variable](#) > **getTemporalClones** ()
- **Variable** **getVariableById** (int varID)
- **Variable** **getTemporalCloneById** (int varID)
- **Variable** **getVariableByName** (String name)
- **Variable** **getTemporalCloneByName** (String name)
- int **getNumberOfVars** ()

#### 5.32.1 Detailed Description

Created by afa on 02/07/14.

### 5.32.2 Constructor & Destructor Documentation

5.32.2.1 `eu.amidst.core.variables.DynamicVariables.DynamicVariables ( Attributes atts, HashMap< Attribute, DistType > typeDists )`

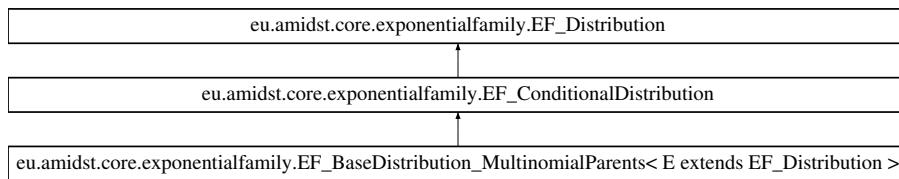
Constructor where the distribution type of random variables is provided as an argument.

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

- `/Users/ana/Documents/core/src/main/java/eu/amidst/core/variables/DynamicVariables.java`

### 5.33 `eu.amidst.core.exponentialfamily.EF_BaseDistribution_MultinomialParents< E extends EF_Distribution >` Class Reference

Inheritance diagram for `eu.amidst.core.exponentialfamily.EF_BaseDistribution_MultinomialParents< E extends EF_Distribution >:`:



#### Public Member Functions

- `EF_BaseDistribution_MultinomialParents (Variable var, List< Variable > parents)`
- abstract `E createNewBaseDistribution (Variable var, List< Variable > non_multinomialParents)`
- `void setEF_BaseDistribution (int indexMultinomial, E baseDist)`
- `E getEF_BaseDistribution (int indexMultinomial)`
- `E getEF_BaseDistribution (DataInstance dataInstance)`
- `SufficientStatistics getSufficientStatistics (DataInstance instance)`
- `int sizeOfSufficientStatistics ()`
- `void setNaturalParameters (NaturalParameters parameters)`
- `void setMomentParameters (MomentParameters parameters)`
- `void updateNaturalFromMomentParameters ()`
- `void updateMomentFromNaturalParameters ()`
- `double computeLogBaseMeasure (DataInstance dataInstance)`
- `double computeLogNormalizer (NaturalParameters parameters)`

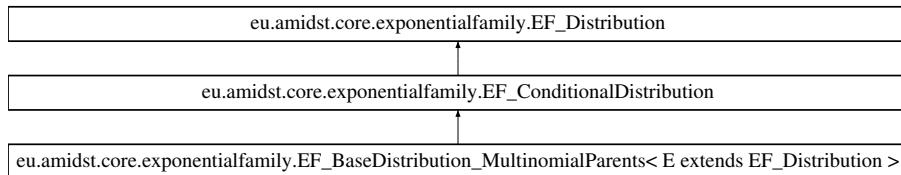
#### Additional Inherited Members

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

- `/Users/ana/Documents/core/src/main/java/eu/amidst/core/exponentialfamily/EF_BaseDistribution_MultinomialParents.java`

### 5.34 `eu.amidst.core.exponentialfamily.EF_ConditionalDistribution` Class Reference

Inheritance diagram for `eu.amidst.core.exponentialfamily.EF_ConditionalDistribution`:



### Public Member Functions

- List< [Variable](#) > **getConditioningVariables** ()

### Protected Attributes

- List< [Variable](#) > **parents**

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/exponentialfamily/EF\_ConditionalDistribution.java

## 5.35 eu.amidst.core.exponentialfamily.EF\_Distribution Class Reference

Inheritance diagram for eu.amidst.core.exponentialfamily.EF\_Distribution:



### Public Member Functions

- final [Variable](#) **getVariable** ()
- final [NaturalParameters](#) **getNaturalParameters** ()
- final [MomentParameters](#) **getMomentParameters** ()
- void **setNaturalParameters** ([NaturalParameters](#) parameters)
- void **setMomentParameters** ([MomentParameters](#) parameters)
- abstract void **updateNaturalFromMomentParameters** ()
- abstract void **updateMomentFromNaturalParameters** ()
- abstract [SufficientStatistics](#) **getSufficientStatistics** ([DataInstance](#) data)
- abstract int **sizeOfSufficientStatistics** ()
- abstract double **computeLogBaseMeasure** ([DataInstance](#) datainstance)
- abstract double **computeLogNormalizer** ()
- double **computeProbabilityOf** ([DataInstance](#) datainstance)
- double **computeLogProbabilityOf** ([DataInstance](#) datainstance)

### Protected Attributes

- [Variable](#) var
- [NaturalParameters](#) **naturalParameters**
- [MomentParameters](#) **momentParameters**

### 5.35.1 Detailed Description

Created by andresmasegosa on 13/11/14.

### 5.35.2 Member Function Documentation

#### 5.35.2.1 final Variable eu.amidst.core.exponentialfamily.EF\_Distribution.getVariable( )

Gets the variable of the distribution

Returns

A `Variable` object.

### 5.35.3 Member Data Documentation

#### 5.35.3.1 Variable eu.amidst.core.exponentialfamily.EF\_Distribution.var [protected]

The variable of the distribution

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/exponentialfamily/EF\_Distribution.java

## 5.36 eu.amidst.core.exponentialfamily.EF\_DistributionBuilder Class Reference

### Static Public Member Functions

- static `EF_BaseDistribution_MultinomialParents newEFFromConditionalDistribution (Multinomial_MultinomialParents dist)`
- static `EF_Normal toEFDistribution (Normal dist)`
- static `Normal toDistribution (EF_Normal ef_normal)`
- static `EF_Multinomial toEFDistribution (Multinomial dist)`
- static `Multinomial toDistribution (EF_Multinomial ef_multinomial)`

### 5.36.1 Detailed Description

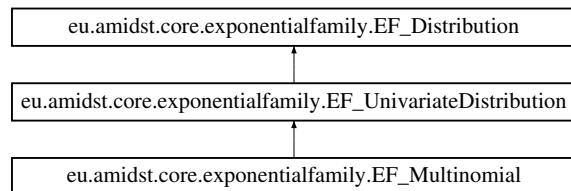
Created by andresmasegosa on 12/11/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/exponentialfamily/EF\_DistributionBuilder.java

## 5.37 eu.amidst.core.exponentialfamily.EF\_Multinomial Class Reference

Inheritance diagram for eu.amidst.core.exponentialfamily.EF\_Multinomial:



## Public Member Functions

- **EF\_Multinomial (Variable var\_)**
- double **computeLogBaseMeasure** (double val)
- double **computeLogNormalizer** ()
- **SufficientStatistics getSufficientStatistics** (double val)
- void **updateNaturalFromMomentParameters** ()
- void **updateMomentFromNaturalParameters** ()
- int **sizeOfSufficientStatistics** ()

## Static Public Member Functions

- static **SufficientStatistics sufficientStatistics** (int nstates, double val)

## Additional Inherited Members

### 5.37.1 Detailed Description

Created by andresmasegosa on 13/11/14.

### 5.37.2 Constructor & Destructor Documentation

#### 5.37.2.1 eu.amidst.core.exponentialfamily.EF\_Multinomial.EF\_Multinomial ( Variable var\_ )

The class constructor.

##### Parameters

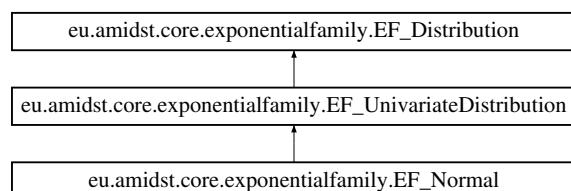
<i>var_</i>	The variable of the distribution.
-------------	-----------------------------------

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/exponentialfamily/EF\_Multinomial.java

## 5.38 eu.amidst.core.exponentialfamily.EF\_Normal Class Reference

Inheritance diagram for eu.amidst.core.exponentialfamily.EF\_Normal:



## Public Member Functions

- **EF\_Normal (Variable var\_)**
- double **computeLogBaseMeasure** (double val)
- double **computeLogNormalizer** ()
- **SufficientStatistics getSufficientStatistics** (double val)
- void **updateNaturalFromMomentParameters** ()
- void **updateMomentFromNaturalParameters** ()
- int **sizeOfSufficientStatistics** ()

## Static Public Member Functions

- static **SufficientStatistics sufficientStatistics** (double val)

## Static Public Attributes

- static int **EXPECTED\_MEAN** = 0
- static int **EXPECTED\_SQUARE** = 1

## Additional Inherited Members

### 5.38.1 Detailed Description

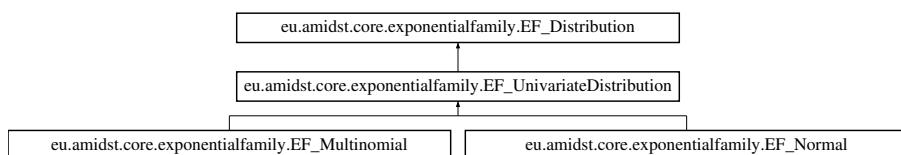
Created by andresmasegosa on 13/11/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/exponentialfamily/EF\_Normal.java

## 5.39 eu.amidst.core.exponentialfamily.EF\_UnivariateDistribution Class Reference

Inheritance diagram for eu.amidst.core.exponentialfamily.EF\_UnivariateDistribution:



## Public Member Functions

- abstract double **computeLogBaseMeasure** (double val)
- abstract **SufficientStatistics getSufficientStatistics** (double val)
- double **computeProbabilityOf** (double val)
- double **computeLogProbabilityOf** (double val)
- **SufficientStatistics getSufficientStatistics** (**DataInstance** data)
- double **computeLogBaseMeasure** (**DataInstance** dataInstance)

## Additional Inherited Members

### 5.39.1 Detailed Description

Created by andresmasegosa on 12/11/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/exponentialfamily/EF\_UnivariateDistribution.java

## 5.40 eu.amidst.core.database.filereaders.arffFileReader.Keys Class Reference

### Public Member Functions

- **Keys** (int[] doubleKeys, int[] intKeys)
- int[] **getDoubleKeys** ()
- int[] **getIntKeys** ()

### 5.40.1 Detailed Description

Created by sigveh on 10/20/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/database/filereaders/arffFileReader/Keys.java

## 5.41 eu.amidst.Main Class Reference

### Static Public Member Functions

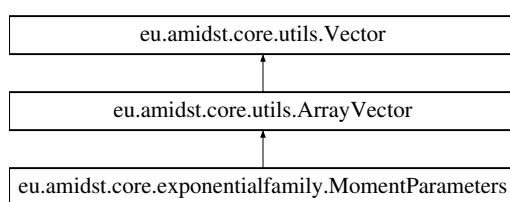
- static void **main** (String[] args)

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/Main.java

## 5.42 eu.amidst.core.exponentialfamily.MomentParameters Class Reference

Inheritance diagram for eu.amidst.core.exponentialfamily.MomentParameters:



### Public Member Functions

- **MomentParameters** (int size)
- **MomentParameters** (double[] vec)

### Additional Inherited Members

#### 5.42.1 Detailed Description

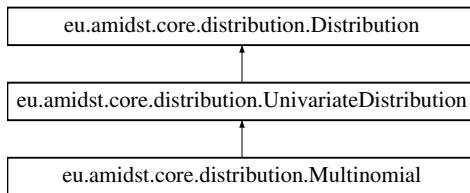
Created by andresmasegosa on 12/11/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/exponentialfamily/MomentParameters.java

## 5.43 eu.amidst.core.distribution.Multinomial Class Reference

Inheritance diagram for eu.amidst.core.distribution.Multinomial:



### Public Member Functions

- **Multinomial (Variable var)**
- void **setProbabilities** (double[] probabilities)
- void **setProbabilityOfState** (int state, double prob)
- double **getProbabilityOfState** (int state)
- double[] **getProbabilities** ()
- double **getLogProbability** (double value)

### Additional Inherited Members

#### 5.43.1 Detailed Description

This class implements a univariate multinomial distribution.

##### Author

Antonio Fernández

##### Version

1.0

##### Since

2014-11-3

## 5.43.2 Constructor & Destructor Documentation

### 5.43.2.1 eu.amidst.core.distribution.Multinomial.Multinomial ( Variable var )

The class constructor.

**Parameters**

<i>var</i>	The variable of the distribution.
------------	-----------------------------------

**5.43.3 Member Function Documentation****5.43.3.1 double eu.amidst.core.distribution.Multinomial.getLogProbability ( double *value* )**

Computes the logarithm of the probability for a given variable state.

**Parameters**

<i>value</i>	The position of the variable state in the array of probabilities (represented as a double for generality reasons).
--------------	--------------------------------------------------------------------------------------------------------------------

**Returns**

A double value with the logarithm of the probability.

**5.43.3.2 double [] eu.amidst.core.distribution.Multinomial.getProbabilities ( )**

Gets the array of probabilities for the different states of the variable.

**Returns**

An array of double with the probabilities.

**5.43.3.3 double eu.amidst.core.distribution.Multinomial.getProbabilityOfState ( int *state* )****Parameters**

<i>state</i>
--------------

**Returns****5.43.3.4 void eu.amidst.core.distribution.Multinomial.setProbabilities ( double[] *probabilities* )**

Sets the probability values to the distribution.

**Parameters**

<i>probabilities</i>	An array of probabilities in the same order as the variable states.
----------------------	---------------------------------------------------------------------

**5.43.3.5 void eu.amidst.core.distribution.Multinomial.setProbabilityOfState ( int *state*, double *prob* )**

Set a probability value in a given position in the array of probabilities.

**Parameters**

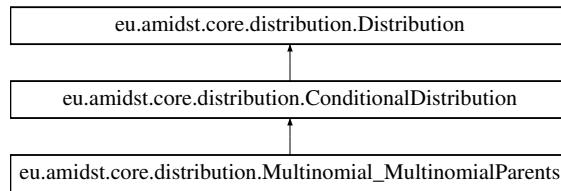
<i>state</i>	The position in which the probability is set.
<i>prob</i>	A probability value.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/distribution/Multinomial.java

## 5.44 eu.amidst.core.distribution.Multinomial\_MultinomialParents Class Reference

Inheritance diagram for eu.amidst.core.distribution.Multinomial\_MultinomialParents:



### Public Member Functions

- [Multinomial\\_MultinomialParents \(Variable var, List< Variable > parents\)](#)
- [Multinomial\[\] getProbabilities \(\)](#)
- void [setMultinomial \(int position, Multinomial multinomialDistribution\)](#)
- void [setMultinomial \(Assignment parentAssignment, Multinomial multinomialDistribution\)](#)
- [Multinomial getMultinomial \(Assignment parentAssignment\)](#)
- [Multinomial getMultinomial \(int position\)](#)
- double [getLogConditionalProbability \(Assignment parentAssignment\)](#)

### Additional Inherited Members

#### 5.44.1 Detailed Description

This class implements a conditional distribution of a multinomial variable given a set of multinomial parents.

#### Author

Antonio Fernández

#### Version

1.0

#### Since

2014-11-4

#### 5.44.2 Constructor & Destructor Documentation

##### 5.44.2.1 eu.amidst.core.distribution.Multinomial\_MultinomialParents ( Variable var, List< Variable > parents )

The class constructor.

**Parameters**

<i>var</i>	The variable of the distribution.
<i>parents</i>	The set of parents of the variable.

**5.44.3 Member Function Documentation**

5.44.3.1 double eu.amidst.core.distribution.Multinomial\_MultinomialParents.getLogConditionalProbability ( Assignment *parentAssignment* )

Computes the logarithm of the probability of the variable for a given state and a parent assignment.

**Parameters**

<i>parentAssignment</i>	An Assignment for the parents.
-------------------------	--------------------------------

**Returns**

A double value with the logarithm of the probability.

5.44.3.2 Multinomial eu.amidst.core.distribution.Multinomial\_MultinomialParents.getMultinomial ( Assignment *parentAssignment* )

Gets the [Multinomial](#) distribution for given a parents assignment.

**Parameters**

<i>parentAssignment</i>	An Assignment for the parents.
-------------------------	--------------------------------

**Returns**

A [Multinomial](#) object.

5.44.3.3 void eu.amidst.core.distribution.Multinomial\_MultinomialParents.setMultinomial ( int *position*, [Multinomial](#) *multinomialDistribution* )

Sets a [Multinomial](#) distribution in a given position in the array of probabilities.

**Parameters**

<i>position</i>	The position in which the distribution is set.
<i>multinomialDistribution</i>	A <a href="#">Multinomial</a> object.

5.44.3.4 void eu.amidst.core.distribution.Multinomial\_MultinomialParents.setMultinomial ( Assignment *parentAssignment*, [Multinomial](#) *multinomialDistribution* )

Sets a [Multinomial](#) distribution in a position in the array of probabilities determined by a given parents assignment.

**Parameters**

<i>parent</i> <i>Assignment</i>	An <a href="#">Assignment</a> for the parents.
<i>multinomial</i> <i>Distribution</i>	A <a href="#">Multinomial</a> object.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/distribution/Multinomial\_MultinomialParents.java

## 5.45 eu.amidst.core.utils.MultinomialIndex Class Reference

### Static Public Member Functions

- static int [getIndexFromVariableAssignment](#) (List< Variable > vars, [Assignment](#) assignment)
- static int [getIndexFromVariableAssignment](#) (List< Variable > vars, List< Double > assignment)
- static int [getIndexFromDataInstance](#) (List< Variable > vars, [DataInstance](#) dataInstance)
- static int [getIndexFromVariableAssignment](#) (List< Variable > vars, double[] assignment)
- static double[] [getVariableAssignmentFromIndex](#) (List< Variable > vars, int index)
- static int [getNumberOfPossibleAssignments](#) (List< Variable > vars)

#### 5.45.1 Detailed Description

This class implements various static methods useful when indexing arrays of distributions involving multinomial variables.

##### Author

Antonio Fernández

##### Version

1.0

##### Since

2014-11-4

#### 5.45.2 Member Function Documentation

##### 5.45.2.1 static int eu.amidst.core.utils.MultinomialIndex.getIndexFromVariableAssignment ( List< Variable > vars, Assignment assignment ) [static]

Computes the order of an assignment when indexing the set of possible values for a set of multinomial variables.

Example: Let X, Y and Z three multinomial variables with states {0,1}, {0,1} and {0,1,2} respectively. Then, they are indexed as:

```
X Y Z Index
0 0 0 0
1 0 0 1
0 1 0 2
1 1 0 3
0 0 1 4
1 0 1 5
0 1 1 6
1 1 1 7
0 0 2 8
1 0 2 9
0 1 2 10
1 1 2 11
```

So, for instance  $\text{Index}(0,0,2) = 8$ .

**Parameters**

<i>vars</i>	A List of variables.
<i>assignment</i>	A Assignment for a set of variables.

**Returns**

The index of the corresponding assignment among the possible ones.

5.45.2.2 static int eu.amidst.core.utils.MultinomialIndex.getIndexFromVariableAssignment ( List< Variable > *vars*, List< Double > *assignment* ) [static]

Computes the order of an assignment when indexing the set of possible values for a set of multinomial variables.

**Parameters**

<i>vars</i>	A List of variables.
<i>assignment</i>	A List of double values for the variables in the same order.

**Returns**

The index of the corresponding assignment among the possible ones.

5.45.2.3 static int eu.amidst.core.utils.MultinomialIndex.getIndexFromVariableAssignment ( List< Variable > *vars*, double[] *assignment* ) [static]

Computes the order of an assignment when indexing the set of possible values for a set of multinomial variables.

**Parameters**

<i>vars</i>	A List of variables.
<i>assignment</i>	An array of double with the values of variables in the same order.

**Returns**

The index of the corresponding assignment among the possible ones.

5.45.2.4 static int eu.amidst.core.utils.MultinomialIndex.getNumberOfPossibleAssignments ( List< Variable > *vars* ) [static]

Computes the number of possible assignments for a list of variables

**Parameters**

<i>vars</i>	The List of variables.
-------------	------------------------

**Returns**

A integer indicating the number of possible assignments.

5.45.2.5 static double [] eu.amidst.core.utils.MultinomialIndex.getVariableAssignmentFromIndex ( List< Variable > *vars*, int *index* ) [static]

Computes the variable assignment located in a given position.

**Parameters**

<i>vars</i>	A List of variables.
<i>index</i>	The position of the Assignment.

**Returns**

An array of double with the values of the variables representing the assignment.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/utils/MultinomialIndex.java

## 5.46 eu.amidst.core.potential.MultivariateGaussian Class Reference

### Public Member Functions

- double **density** (double[] values)
- void **setMean** (double[] values)
- double **getMean** ()
- void **setCovariances** (double[][] values)
- double[][] **getCovariances** ()
- double[] **sample** ()

#### 5.46.1 Detailed Description

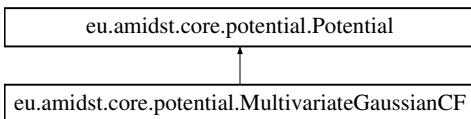
Created by afa on 03/07/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/potential/MultivariateGaussian.java

## 5.47 eu.amidst.core.potential.MultivariateGaussianCF Class Reference

Inheritance diagram for eu.amidst.core.potential.MultivariateGaussianCF:



### Public Member Functions

- void **setParameters** (MultivariateGaussian prob)
- double[][] **getKParameter** ()
- double[] **getHParameter** ()
- double **getGParameter** ()
- MultivariateGaussian **getMG** ()
- void **setVariables** (List variables)
- List **getVariables** ()
- void **combine** (Potential pot)
- void **marginalize** (List variables)

### 5.47.1 Detailed Description

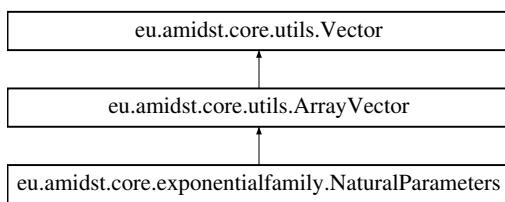
Created by afa on 03/07/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/potential/MultivariateGaussianCF.java

## 5.48 eu.amidst.core.exponentialfamily.NaturalParameters Class Reference

Inheritance diagram for eu.amidst.core.exponentialfamily.NaturalParameters:



### Public Member Functions

- **NaturalParameters** (int size)
- **NaturalParameters** (double[] vec)

### Additional Inherited Members

#### 5.48.1 Detailed Description

Created by andresmasegosa on 12/11/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/exponentialfamily/NaturalParameters.java

## 5.49 eu.amidst.core.database.filereaders.NextDynamicDataInstance Class Reference

### Public Member Functions

- **NextDynamicDataInstance** (DataRow past, DataRow present, int sequenceID, int timeIDcounter)
- **DynamicDataInstance nextDataInstance\_NoTimeID\_NoSeq** (DataFileReader reader)
- **DynamicDataInstance nextDataInstance\_NoSeq** (DataFileReader reader, Attribute attTimeID)
- **DynamicDataInstance nextDataInstance\_NoTimeID** (DataFileReader reader, Attribute attSequenceID)
- **DynamicDataInstance nextDataInstance** (DataFileReader reader, Attribute attSequenceID, Attribute attTimeID)

#### 5.49.1 Detailed Description

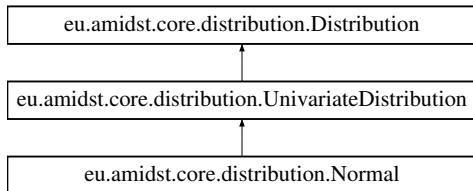
Created by ana@cs.aau.dk on 13/11/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/database/filereaders/NextDynamicDataInstance.java

## 5.50 eu.amidst.core.distribution.Normal Class Reference

Inheritance diagram for eu.amidst.core.distribution.Normal:



### Public Member Functions

- [Normal \(Variable var\)](#)
- double [getMean \(\)](#)
- void [setMean \(double mean\)](#)
- double [getSd \(\)](#)
- void [setSd \(double sd\)](#)
- double [getProbability \(double value\)](#)
- double [getLogProbability \(double value\)](#)

### Additional Inherited Members

#### 5.50.1 Detailed Description

This class implements a univariate [Normal](#) distribution.

#### Author

Antonio Fernández

#### Version

1.0

#### Since

2014-11-3

#### 5.50.2 Constructor & Destructor Documentation

##### 5.50.2.1 eu.amidst.core.distribution.Normal.Normal ( Variable var )

The class constructor.

**Parameters**

<i>var</i>	The variable of the distribution.
------------	-----------------------------------

**5.50.3 Member Function Documentation****5.50.3.1 double eu.amidst.core.distribution.Normal.getLogProbability ( double *value* )**

Computes the logarithm of the density function in a given point.

**Parameters**

<i>value</i>	An value for the variable.
--------------	----------------------------

**Returns**

A double with the logarithm of the density value.

**5.50.3.2 double eu.amidst.core.distribution.Normal.getMean ( )**

Gets the mean of the distribution.

**Returns**

A double value with the mean.

**5.50.3.3 double eu.amidst.core.distribution.Normal.getProbability ( double *value* )**

Evaluates the density function in a given point.

**Parameters**

<i>value</i>	An value for the variable.
--------------	----------------------------

**Returns**

A double with the value of the density.

**5.50.3.4 double eu.amidst.core.distribution.Normal.getSd ( )**

Gets the standard deviation of the distribution.

**Returns**

A double value with the standar deviation.

**5.50.3.5 void eu.amidst.core.distribution.Normal.setMean ( double *mean* )**

Sets the mean of the distribution.

## Parameters

<i>mean</i>	A value for the mean.
-------------	-----------------------

#### 5.50.3.6 void eu.midst.core.distribution.Normal.setSd ( double sd )

Sets the standard deviation of the distribution.

## Parameters

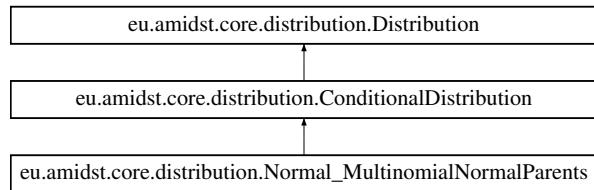
**sd** | A value for the standard deviation.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/distribution/Normal.java

5.51 eu.amidst.core.distribution.Normal\_MultinomialNormalParents Class Reference

Inheritance diagram for eu.amidst.core.distribution.Normal\_MultinomialNormalParents:



## Public Member Functions

- `Normal_MultinomialNormalParents (Variable var, List< Variable > parents)`
  - `Normal_NormalParents getNormal_NormalParentsDistribution (Assignment assignment)`
  - `Normal_NormalParents getNormal_NormalParentsDistribution (int i)`
  - `void setNormal_NormalParentsDistribution (int position, Normal_NormalParents distribution)`
  - `void setNormal_NormalParentsDistribution (Assignment assignment, Normal_NormalParents distribution)`
  - `double getLogConditionalProbability (Assignment assignment)`
  - `List< Variable > getMultinomialParents ()`
  - `List< Variable > getNormalParents ()`
  - `Normal_NormalParents[] getDistribution ()`

## Additional Inherited Members

### **5.51.1 Detailed Description**

This class implements a conditional distribution of a normal variable given a set of multinomial and normal parents.

## Author

Antonio Fernández

## Version

1.0

Since

2014-11-4

### 5.51.2 Constructor & Destructor Documentation

5.51.2.1 `eu.amidst.core.distribution.Normal_MultinomialNormalParents.Normal_MultinomialNormalParents ( Variable var, List< Variable > parents )`

The class constructor.

Parameters

<code>var</code>	The variable of the distribution.
<code>parents</code>	The set of parent variables.

### 5.51.3 Member Function Documentation

5.51.3.1 `double eu.amidst.core.distribution.Normal_MultinomialNormalParents.getLogConditionalProbability ( Assignment assignment )`

Computes the logarithm of the evaluated density function in a point after restricting the distribution to a given parent Assignment.

Parameters

<code>assignment</code>	An Assignment
-------------------------	---------------

Returns

A double with the logarithm of the corresponding density value.

5.51.3.2 `Normal_NormalParents eu.amidst.core.distribution.Normal_MultinomialNormalParents.getNormal_NormalParentsDistribution ( Assignment assignment )`

Gets a `Normal_NormalParentsDistribution` distribution conditioned to an assignment over a set of `Multinomial` parents. Let X and Y two sets of `Normal` variables, and Z a set of `Multinomial`. Then this method computes  $f(X|Y, Z=z)$ .

Parameters

<code>assignment</code>	An assignment over a set of parents. For generality reasons, apart from the <code>Multinomial</code> parents, the assignment contains values for the <code>Normal</code> parents as well (although they are not used in this case).
-------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Returns

a `Normal_NormalParentsDistribution` distribution conditioned to the assignment given as argument.

5.51.3.3 `void eu.amidst.core.distribution.Normal_MultinomialNormalParents.setNormal_NormalParentsDistribution ( int position, Normal_NormalParents distribution )`

Sets a `Normal_NormalParents` distribution to a given position in the array of distributions.

**Parameters**

<i>position</i>	The position in which the distribution is set.
<i>distribution</i>	A <a href="#">Normal_NormalParents</a> distribution.

5.51.3.4 void eu.amidst.core.distribution.Normal\_MultinomialNormalParents.setNormal\_NormalParentsDistribution (  
**Assignment** *assignment*, **Normal\_NormalParents** *distribution* )

Sets a [Normal\\_NormalParents](#) distribution to the array of distributions in a position determined by an given Assignment. Note that this assignment contains values for the [Normal](#) parents as well (although they are not used in this case).

**Parameters**

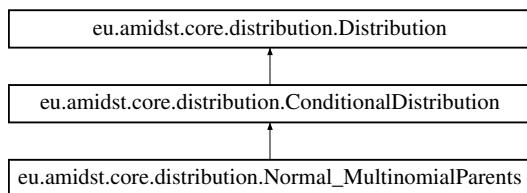
<i>assignment</i>	An Assignment for the parents variables.
<i>distribution</i>	A <a href="#">Normal_NormalParents</a> distribution.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/distribution/Normal\_MultinomialNormalParents.java

## 5.52 eu.amidst.core.distribution.Normal\_MultinomialParents Class Reference

Inheritance diagram for eu.amidst.core.distribution.Normal\_MultinomialParents:



### Public Member Functions

- **Normal\_MultinomialParents** ([Variable](#) *var*, [List< Variable >](#) *parents*)
- **Normal** [getNormal](#) (int *position*)
- **Normal** [getNormal](#) ([Assignment](#) *parentsAssignment*)
- void [setNormal](#) (int *position*, [Normal](#) *normalDistribution*)
- void [setNormal](#) ([Assignment](#) *parentsAssignment*, [Normal](#) *normalDistribution*)
- double [getLogConditionalProbability](#) ([Assignment](#) *assignment*)

### Additional Inherited Members

#### 5.52.1 Detailed Description

This class implements a conditional distribution of a normal variable given a set of multinomial parents.

#### Author

Antonio Fernández

**Version**

1.0

**Since**

2014-11-4

**5.52.2 Constructor & Destructor Documentation**

5.52.2.1 `eu.amidst.core.distribution.Normal_MultinomialParents.Normal_MultinomialParents ( Variable var, List< Variable > parents )`

The class constructor.

**Parameters**

<code>var</code>	The variable of the distribution.
<code>parents</code>	The set of parent variables.

**5.52.3 Member Function Documentation**

5.52.3.1 `double eu.amidst.core.distribution.Normal_MultinomialParents.getLogConditionalProbability ( Assignment assignment )`

Computes the logarithm of the evaluated density function in a point after conditioning the distribution to a given parent Assignment.

**Parameters**

<code>assignment</code>	An Assignment for the parents.
-------------------------	--------------------------------

**Returns**

A double with the logarithm of the corresponding density value.

5.52.3.2 `Normal eu.amidst.core.distribution.Normal_MultinomialParents.getNormal ( Assignment parentsAssignment )`

Gets the corresponding univariate normal distribution after conditioning the distribution to a multinomial parent assignment.

**Parameters**

<code>parentsAssignment</code>	An Assignment for the parents.
--------------------------------	--------------------------------

**Returns**

A `Normal` object with the univariate distribution.

5.52.3.3 `void eu.amidst.core.distribution.Normal_MultinomialParents.setNormal ( int position, Normal normalDistribution )`

Sets a `Normal` distribution in a given position in the array of distributions.

**Parameters**

<i>position</i>	The position in which the distribution is set.
<i>normal</i> <i>Distribution</i>	The <a href="#">Normal</a> distribution to be set.

5.52.3.4 void eu.amidst.core.distribution.Normal\_MultinomialParents.setNormal ( [Assignment parentsAssignment](#), [Normal normalDistribution](#) )

Sets a [Multinomial](#) distribution in a position in the array of distributions determined by a given parents assignment.

**Parameters**

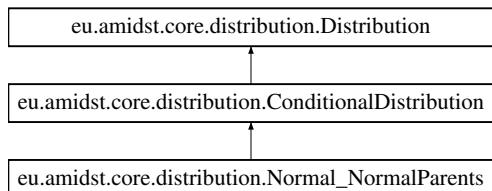
<i>parents</i> <i>Assignment</i>	An <a href="#">Assignment</a> for the parents.
<i>normal</i> <i>Distribution</i>	The <a href="#">Normal</a> distribution to be set.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/distribution/Normal\_MultinomialParents.java

## 5.53 eu.amidst.core.distribution.Normal\_NormalParents Class Reference

Inheritance diagram for eu.amidst.core.distribution.Normal\_NormalParents:



### Public Member Functions

- [Normal\\_NormalParents \(Variable var, List< Variable > parents\)](#)
- double [getIntercept \(\)](#)
- void [setIntercept \(double intercept\)](#)
- double[] [getCoeffParents \(\)](#)
- void [setCoeffParents \(double\[\] coeffParents\)](#)
- double [getSd \(\)](#)
- void [setSd \(double sd\)](#)
- [Normal getUnivariateNormal \(Assignment parentsAssignment\)](#)
- double [getLogConditionalProbability \(Assignment assignment\)](#)

### Additional Inherited Members

#### 5.53.1 Detailed Description

This class implements a Conditional Linear Gaussian distribution, i.e. a distribution of a normal variable with continuous normal parents.

**Author**

Antonio Fernández

**Version**

1.0

**Since**

2014-11-4

### 5.53.2 Constructor & Destructor Documentation

5.53.2.1 `eu.amidst.core.distribution.Normal_NormalParents.Normal_NormalParents ( Variable var, List<Variable> parents )`

The class constructor.

**Parameters**

<code>var</code>	The variable of the distribution.
<code>parents</code>	The set of parents of the variable.

### 5.53.3 Member Function Documentation

5.53.3.1 `double [] eu.amidst.core.distribution.Normal_NormalParents.getCoeffParents ( )`

Gets the coefficients for the parent variables.

**Returns**

An array of `double` with the coefficients.

5.53.3.2 `double eu.amidst.core.distribution.Normal_NormalParents.getIntercept ( )`

Gets the intercept of the distribution.

**Returns**

A `double` value with the intercept.

5.53.3.3 `double eu.amidst.core.distribution.Normal_NormalParents.getLogConditionalProbability ( Assignment assignment )`

Computes the logarithm of the evaluated density function in a point after conditioning the distribution to a given parent Assignment.

**Parameters**

<code>assignment</code>	An Assignment
-------------------------	---------------

**Returns**

A `double` with the logarithm of the corresponding density value.

5.53.3.4 double eu.amidst.core.distribution.Normal\_NormalParents.getSd( )

Gets the standard deviation of the variable.

**Returns**

A double value with the standard deviation.

5.53.3.5 Normal eu.amidst.core.distribution.Normal\_NormalParents.getUnivariateNormal( Assignment parentsAssignment )

Gets the corresponding univariate normal distribution after conditioning the distribution to a parent assignment.

**Parameters**

<i>parents</i> ↪ Assignment	An Assignment for the parents.
-----------------------------	--------------------------------

**Returns**

A [Normal](#) object with the univariate distribution.

5.53.3.6 void eu.amidst.core.distribution.Normal\_NormalParents.setCoeffParents( double[] coeffParents )

Sets the coefficients of the distribution

**Parameters**

<i>coeffParents</i>	An array of double with the coefficients, one for each parent.
---------------------	----------------------------------------------------------------

5.53.3.7 void eu.amidst.core.distribution.Normal\_NormalParents.setIntercept( double intercept )

Sets the intercept of the distribution.

**Parameters**

<i>intercept</i>	A double value with the intercept.
------------------	------------------------------------

5.53.3.8 void eu.amidst.core.distribution.Normal\_NormalParents.setSd( double sd )

Sets the standard deviation of the variable.

**Parameters**

<i>sd</i>	A double value with the standard deviation.
-----------	---------------------------------------------

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/distribution/Normal\_NormalParents.java

## 5.54 eu.amidst.core.huginlink.Others Class Reference

### 5.54.1 Detailed Description

Created by afa on 18/11/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/huginlink/Others.java

## 5.55 eu.amidst.core.modelstructure.ParentSet Class Reference

### Public Member Functions

- void **addParent** ([Variable](#) var)
- void **removeParent** ([Variable](#) var)
- List< [Variable](#) > **getParents** ()
- int **getNumberOfParents** ()
- void **blockParents** ()

### Static Public Member Functions

- static [ParentSet](#) **newParentSet** ()

#### 5.55.1 Detailed Description

Created by afa on 02/07/14.

#### 5.55.2 Member Function Documentation

##### 5.55.2.1 void eu.amidst.core.modelstructure.ParentSet.blockParents ( )

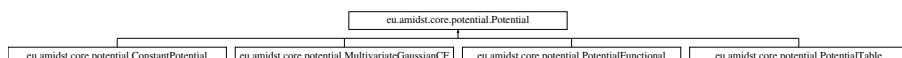
Is an ArrayList pointer to an ArrayList unmodifiable object still unmodifiable? I guess so right?

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/modelstructure/ParentSet.java

## 5.56 eu.amidst.core.potential.Potential Interface Reference

Inheritance diagram for eu.amidst.core.potential.Potential:



### Public Member Functions

- void **setVariables** (List variables)
- List **getVariables** ()
- void **combine** ([Potential](#) pot)
- void **marginalize** (List variables)

#### 5.56.1 Detailed Description

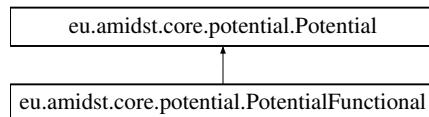
Created by afa on 03/07/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/potential/Potential.java

## 5.57 eu.amidst.core.potential.PotentialFunctional Class Reference

Inheritance diagram for eu.amidst.core.potential.PotentialFunctional:



### Public Member Functions

- void **setVariables** (List variables)
- List **getVariables** ()
- void **combine** (Potential pot)
- void **marginalize** (List variables)

#### 5.57.1 Detailed Description

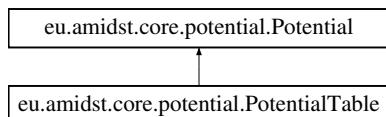
Created by afa on 03/07/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/potential/PotentialFunctional.java

## 5.58 eu.amidst.core.potential.PotentialTable Class Reference

Inheritance diagram for eu.amidst.core.potential.PotentialTable:



### Public Member Functions

- PotentialTable (int nstates)
- void **setValues** (double[] values)
- double[] **getValues** ()
- void **setVariables** (List variables)
- List **getVariables** ()
- void **combine** (Potential pot)
- void **marginalize** (List variables)
- void **normalize** ()

#### 5.58.1 Detailed Description

Created by afa on 03/07/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/potential/PotentialTable.java

## 5.59 eu.amidst.core.database.dynamics.SequenceDataStream Interface Reference

### Public Member Functions

- [Attributes \*\*getDynamicAttributes\*\* \(\)](#)
- [int \*\*getNumTimeStepsBack\*\* \(\)](#)
- [boolean \*\*hasMoreData\*\* \(\)](#)
- [DynamicDataInstance \*\*nextSequenceData\*\* \(\)](#)
- [void \*\*reset\*\* \(\)](#)
- [boolean \*\*isResetable\*\* \(\)](#)

### 5.59.1 Detailed Description

Created by afa on 03/07/14.

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

- [/Users/ana/Documents/core/src/main/java/eu/amidst/core/database/dynamics/SequenceDataStream.java](#)

## 5.60 eu.amidst.core.database.dynamics.readers.SequenceDataStreamReaderFromFile Class Reference

### Public Member Functions

- [SequenceDataStreamReaderFromFile \(String fileName\)](#)
- [SequenceDataStream \*\*getDataStream\*\* \(\)](#)

### 5.60.1 Detailed Description

Created by andresmasegosa on 27/08/14.

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

- [/Users/ana/Documents/core/src/main/java/eu/amidst/core/database/dynamics/readers/SequenceDataStreamReaderFromFile.java](#)

## 5.61 eu.amidst.core.database.dynamics.SequenceStreamWindow Interface Reference

### Public Member Functions

- [Attributes \*\*getDynamicAttributes\*\* \(\)](#)
- [int \*\*getWindowSize\*\* \(\)](#)
- [boolean \*\*hasMoreData\*\* \(\)](#)
- [void \*\*loadNextWindow\*\* \(\)](#)
- [DynamicDataInstance \*\*getSequenceData\*\* \(int indexInWindow\)](#)
- [boolean \*\*isResetable\*\* \(\)](#)
- [void \*\*reset\*\* \(\)](#)

### 5.61.1 Detailed Description

Created by afa on 03/07/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/database/dynamics/SequenceStreamWindow.java

## 5.62 eu.amidst.core.variables.StateSpaceType Enum Reference

### Static Public Member Functions

- static [StateSpaceType parseKind](#) (String s)

### Public Attributes

- **REAL**

### 5.62.1 Detailed Description

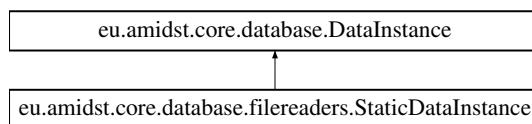
Created by sigveh on 10/20/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/variables/StateSpaceType.java

## 5.63 eu.amidst.core.database.filereaders.StaticDataInstance Class Reference

Inheritance diagram for eu.amidst.core.database.filereaders.StaticDataInstance:



### Public Member Functions

- **StaticDataInstance** ([DataRow](#) dataRow\_)
- double **getValue** ([Variable](#) var)
- int **getSequenceID** ()
- int **getTimeID** ()

### 5.63.1 Detailed Description

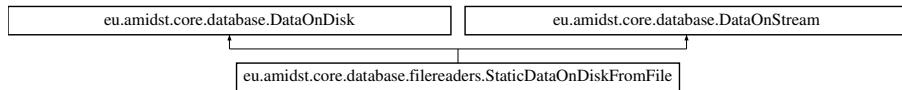
Created by andresmasegosa on 11/11/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/database/filereaders/StaticDataInstance.java

## 5.64 eu.amidst.core.database.filereaders.StaticDataOnDiskFromFile Class Reference

Inheritance diagram for eu.amidst.core.database.filereaders.StaticDataOnDiskFromFile:



### Public Member Functions

- **StaticDataOnDiskFromFile** ([DataFileReader](#) reader)
- **DataInstance** [nextDataInstance](#) ()
- boolean [hasMoreDataInstances](#) ()
- **Attributes** [getAttributes](#) ()
- void [restart](#) ()

#### 5.64.1 Detailed Description

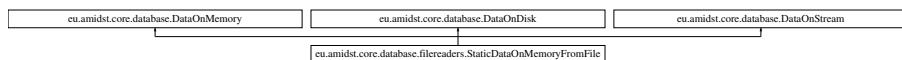
Created by andresmasegosa on 11/11/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/database/filereaders/StaticDataOnDiskFromFile.java

## 5.65 eu.amidst.core.database.filereaders.StaticDataOnMemoryFromFile Class Reference

Inheritance diagram for eu.amidst.core.database.filereaders.StaticDataOnMemoryFromFile:



### Public Member Functions

- **StaticDataOnMemoryFromFile** ([DataFileReader](#) reader)
- int [getNumberOfDataInstances](#) ()
- **DataInstance** [getDataInstance](#) (int i)
- **DataInstance** [nextDataInstance](#) ()
- boolean [hasMoreDataInstances](#) ()
- void [restart](#) ()
- **Attributes** [getAttributes](#) ()

#### 5.65.1 Detailed Description

Created by andresmasegosa on 11/11/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/database/filereaders/StaticDataOnMemoryFromfile.java

## 5.66 eu.amidst.core.variables.StaticVariables Class Reference

### Public Member Functions

- **StaticVariables** (**Attributes** atts)
- **StaticVariables** (**Attributes** atts, **HashMap< Attribute, DistType >** typeDists)
- **Variable addHiddenVariable** (**VariableBuilder** builder)
- **List< Variable > getVariables** ()
- **Variable getVariable** (int varID)
- **Variable getVariable** (String name)
- **int getNumberOfVars** ()

### 5.66.1 Detailed Description

Created by afa on 02/07/14.

### 5.66.2 Constructor & Destructor Documentation

#### 5.66.2.1 eu.amidst.core.variables.StaticVariables ( Attributes atts )

Constructor where the distribution type of random variables is initialized by default.

#### 5.66.2.2 eu.amidst.core.variables.StaticVariables ( Attributes atts, **HashMap< Attribute, DistType >** typeDists )

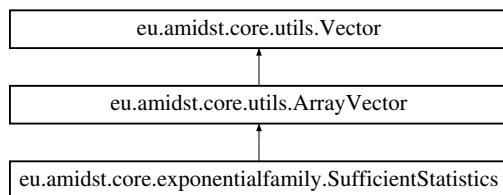
Constructor where the distribution type of random variables is provided as an argument.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/variables/StaticVariables.java

## 5.67 eu.amidst.core.exponentialfamily.SufficientStatistics Class Reference

Inheritance diagram for eu.amidst.core.exponentialfamily.SufficientStatistics:



### Public Member Functions

- **SufficientStatistics** (int size)
- **SufficientStatistics** (double[] vec)

## Additional Inherited Members

### 5.67.1 Detailed Description

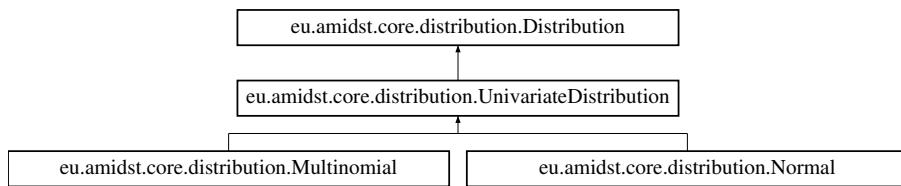
Created by andresmasegosa on 12/11/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/exponentialfamily/SufficientStatistics.java

## 5.68 eu.amidst.core.distribution.UnivariateDistribution Class Reference

Inheritance diagram for eu.amidst.core.distribution.UnivariateDistribution:



## Public Member Functions

- double `getProbability` (double value)
- abstract double `getLogProbability` (double value)

## Additional Inherited Members

### 5.68.1 Detailed Description

This interface generalizes the set of univariate distributions.

#### Author

Antonio Fernández

#### Version

1.0

#### Since

2014-11-3

## 5.68.2 Member Function Documentation

### 5.68.2.1 abstract double eu.amidst.core.distribution.UnivariateDistribution.getLogProbability ( double value ) [abstract]

Evaluates the distribution in a given point.

**Parameters**

<code>value</code>	The point to be evaluated.
--------------------	----------------------------

**Returns**

A double value with the logarithm of the evaluated distribution.

**5.68.2.2 double eu.amidst.core.distribution.UnivariateDistribution.getProbability ( double value )**

Evaluates the distribution in a given point.

**Parameters**

<code>value</code>	The point to be evaluated.
--------------------	----------------------------

**Returns**

A double value with the evaluated distribution.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/distribution/UnivariateDistribution.java

**5.69 eu.amidst.core.utils.Utils Class Reference****Static Public Member Functions**

- static double **getMissingValue** ()
- static boolean **isMissing** (double val)
- static void **accumulatedSumVectors** (double[] a, double[] b)

**5.69.1 Detailed Description**

Created by andresmasegosa on 28/08/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/utils/Utils.java

**5.70 eu.amidst.core.variables.Variable Interface Reference****Public Member Functions**

- String **getName** ()
- int **getVarID** ()
- boolean **isObservable** ()
- int **getNumberOfStates** ()
- **StateSpaceType** **getStateSpaceType** ()
- **DistType** **getDistributionType** ()
- boolean **isTemporalClone** ()
- **Attribute** **getAttribute** ()

### 5.70.1 Detailed Description

Created by afa on 02/07/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/variables/Variable.java

## 5.71 eu.amidst.core.variables.VariableBuilder Class Reference

### Public Member Functions

- **VariableBuilder** ([Attribute](#) att)
- **VariableBuilder** ([Attribute](#) att, [DistType](#) typeDist)

### Static Public Member Functions

- static String **getName** ()
- static boolean **isObservable** ()
- static int **getNumberOfStates** ()
- static [StateSpaceType](#) **getStateSpaceType** ()
- static [DistType](#) **getDistributionType** ()
- static [Attribute](#) **getAttribute** ()

### 5.71.1 Detailed Description

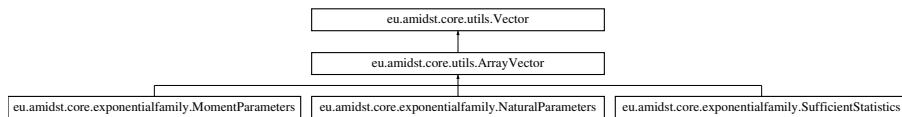
Created by andresmasegosa on 04/11/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/variables/VariableBuilder.java

## 5.72 eu.amidst.core.utils.Vector Interface Reference

Inheritance diagram for eu.amidst.core.utils.Vector:



### Public Member Functions

- double **get** (int i)
- void **set** (int i, double val)
- int **size** ()
- default void **dotProduct** ([Vector](#) vec)

### Static Public Member Functions

- static double **dotProduct** ([Vector](#) vec1, [Vector](#) vec2)

### 5.72.1 Detailed Description

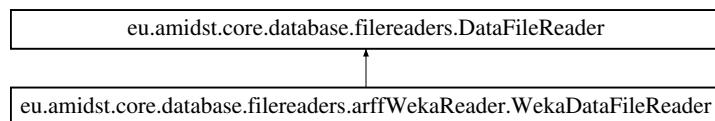
Created by andresmasegosa on 12/11/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/utils/Vector.java

## 5.73 eu.amidst.core.database.filereaders.arffWekaReader.WekaDataFileReader Class Reference

Inheritance diagram for eu.amidst.core.database.filereaders.arffWekaReader.WekaDataFileReader:



### Public Member Functions

- **WekaDataFileReader** (String s)
- **Attributes** **getAttributes** ()
- **DataRow** **nextDataRow** ()
- boolean **hasMoreDataRows** ()
- void **reset** ()
- boolean **doesItReadThisFileExtension** (String fileExtension)

### 5.73.1 Detailed Description

Created by [ana@cs.aau.dk](mailto:ana@cs.aau.dk) on 14/11/14.

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

- /Users/ana/Documents/core/src/main/java/eu/amidst/core/database/filereaders/arffWekaReader/WekaDataFileReader.java