

# SQUAD

june 11, 2016

**Type** R package

**Title**

**Version**

**Imports** BoolNet, deSolve

**Requirements**

**Description** Automatically transform Boolean Regulatory Network Models to a continuous one using SQUAD formalism.

**License**

**LazyLoad**

**Encoding**

**Author**

**Maintainer**

**NeedsCompilation**

**Repository**

**Date/Publication**

---

`getTransitionsMatrix`    *return a matrix of the frecuencies of transitions between attractors in a noisy simulation*

---

## Description

Simulate the attractors transitions of a regulatory network system under a noisy updating scheme. The algorithm is implemented according to [1].

## Usage

```
getTransitionsMatrix(net,  
                     attractors,  
                     noiseLevel,  
                     numberOfWalks,  
                     numberOfUpdates)
```

## Arguments

<code>net</code>	A network structure of class <code>BooleanNetwork</code> . These networks can be read from files by <code>loadNetwork</code> or generated by <code>generateRandomNKNetwork</code> .
<code>attractors</code>	An object of the class <code>attractorInfo</code> as returned by <code>getAttractors()</code> .
<code>noiseLevel</code>	A real number in the 0-1 interval that indicates the level of noise in which the logic rules are violated when the system is updated.
<code>numberOfWalks</code>	Number of simulations starting from an attractor.
<code>numberOfUpdates</code>	Number of updating steps in the simulation.

## Details

Noisy dynamic simulations are carried out by updating a node value following the network rules with probability  $1 - p$  but neglecting them with probability  $p$ . Starting from a given attractor the system is updated  $n$  times in the noisy context and record if the system transit to another attractor (it may happen to be the same as the initial state) in this period. This process (*walks*) is repeated  $m$  times and frequencies of transitions are calculated according to previous methods. Frequencies of transitions are calculated by counting the number of transitions and this number is divided by the number of walks.

## Value

Returns a matrix  $A$  of transitions between attractors. The entry  $A_{ij}$  of the matrix  $A$  represent the frequency of transitions from attractor  $i$  to the attractor  $j$ .

## References

[1]

## Example

```
> data(cellCycle)
> attractors <- getAttractors(cellCycle)
> transitions <- getTransitionMatrix(cellCycle, attractors,
  noiseLevel = 0.05,
  numberOfWalks = 300,
  numberOfUpdates = 200)
> transitions
```

---

<code>asContinuous</code>	<i>convert a boolean regulatory model to a continuous model</i>
---------------------------	---

---

## Description

Take a net object as that defined by `loadNetwork()` and return a continuous regulatory network model function. The continuous network can be used as an object function argument for the `deSolve` R package.

## Usage

```
asContinuous(net)
```

## Example

```
> data(cellCycle)
# setting initial state
> state<-generateState(cellCycle,specs=c("CycA"=1,"CycD"=1))
# setting intervals for numerical integration
> times <- seq(min=0,max=1,seq.out=100)

> cellCycle.s <- asContinuous(cellCycle)
> transitions <- ode(y=state, fun=cellCycle.s, times=times,
                    parameters = "default")
> transitions
```

## Details

The continuous network object retrieved by the function represent a system of ordinary equations according to the methodology described in Martínez-Sosa, 2013 [2]. This methodology have the same fixed point attractors of that in the Boolean model but have the advantage to account kinetic parameters.

---

<code>graphToModel</code>	<i>convert a connectivity Matrix to a boolNet regulatory network model</i>
---------------------------	--

---

## Description

Takes a connectivity matrix that represent a static regulatory network graph and transform it to a "BooleanNetwork" object that can be used to simulate network transitions and calculating attractors.

## Usage

```
graphToModel(connectivityMatrix)
```

## Arguments

<code>connectivityMatrix</code>	A square matrix of the nodes of the network with -1, 0, or 1 integer values that represent negative, null or positive regulation.
---------------------------------	---

## Example

```
# generate a sample random connectivity matrix  
> net<-randomMatrix.e(dimensions=20)  
> plotAttractors(getAttractors(net))
```

## Details

Connectivity matrix is defined as that with entries  $A_{ij}$  represent a regulation from the node  $i$  to  $j$ . Positive (negative) regulations are represent

by 1 (-1). No regulatory dependencies are indicated by zero. The model assumes that node  $j$  activation is achieved if and only if there is at least one of their positive regulators is active and all negative regulators inputs are inactive.

# Bibliography

- [1] Floral morphogenesis: stochastic explorations of a gene network epigenetic landscape. *PloS one*, 3(11):e3626, jan 2008.
- [2] P. Martínez-Sosa and L. Mendoza. The regulatory network that controls the differentiation of t lymphocytes. *Biosystems*, 113(2):96–103, 2013.