

# Package ‘ternarynet’

September 24, 2012

**Version** 1.0.0

**Title** Ternary Network Estimation

**Description** A computational Bayesian approach to ternary gene regulatory network estimation from gene perturbation experiments.

**Author** Matthew N. McCall <mccallm@gmail.com>, Anthony Almudevar <Anthony\_Almudevar@urmc.rochester.edu>

**Maintainer** Matthew N. McCall <mccallm@gmail.com>

**Depends** R (>= 2.10.0), methods

**biocViews** Software, CellBiology, GraphsAndNetworks, Bioinformatics

**License** GPL (>= 2)

## R topics documented:

attractorSummary . . . . .	1
graphPosterior . . . . .	2
plotTraces . . . . .	3
ternaryFit-class . . . . .	4
ternaryFitParameters-class . . . . .	5
ternaryPost-class . . . . .	6
tnetfit . . . . .	7
tnetpost . . . . .	8

<b>Index</b>	<b>10</b>
--------------	-----------

---

attractorSummary	<i>Summarize Attractors</i>
------------------	-----------------------------

---

## Description

This function summarizes the posterior probability of possible attractors.

## Usage

```
attractorSummary(tpost, post.prob.limit = 0.01, wildtype = TRUE)
```

**Arguments**

tpost                    a ternaryPost object  
 post.prob.limit  
                           the minimum posterior probability for an attractor to be listed  
 wildtype                if TRUE, the wildtype attractors are summarized; if FALSE, the perturbed attractors are summarized.

**Value**

The function returns a matrix of attractors and posterior probabilities for each perturbation.

**Author(s)**

Matthew N. McCall and Anthony Almudevar

**See Also**

Almudevar A, McCall MN, McMurray H, Land H (2011). Fitting Boolean Networks from Steady State Perturbation Data, *Statistical Applications in Genetics and Molecular Biology*, 10(1): Article 47.

**Examples**

```

ssObj <- matrix(c(1,1,1,0,1,1,0,0,1),nrow=3)
pObj <- matrix(c(1,0,0,0,1,0,0,0,1),nrow=3)
rownames(ssObj) <- rownames(pObj) <- colnames(ssObj) <- colnames(pObj) <- c("Gene1","Gene2","Gene3")
tnfitObj <- tnetfit(ssObj, pObj)
tnpostObj <- tnetpost(tnfitObj, mdelta=10, msample=10)
attractorSummary(tnpostObj)

```

---

graphPosterior

*Network Topology*

---

**Description**

This function summarizes the topology of the ternary network using marginal edge probabilities.

**Usage**

```
graphPosterior(tpost)
```

**Arguments**

tpost                    a ternaryPost object

**Value**

The function returns a matrix of marginal posterior probabilities of each possible network edge – rows are children and columns are parents. The first column represents no parents.

**Author(s)**

Matthew N. McCall and Anthony Almudevar

**See Also**

Almudevar A, McCall MN, McMurray H, Land H (2011). Fitting Boolean Networks from Steady State Perturbation Data, *Statistical Applications in Genetics and Molecular Biology*, 10(1): Article 47.

**Examples**

```
ssObj <- matrix(c(1,1,1,0,1,1,0,0,1),nrow=3)
pObj <- matrix(c(1,0,0,0,1,0,0,0,1),nrow=3)
rownames(ssObj) <- rownames(pObj) <- colnames(ssObj) <- colnames(pObj) <- c("Gene1","Gene2","Gene3")
tnfitObj <- tnetfit(ssObj, pObj)
tnpostObj <- tnetpost(tnfitObj, mdelta=10, msample=10)
graphPosterior(tnpostObj)
```

---

plotTraces

*Network Fit Traces*

---

**Description**

This function plots the trace of four model parameters.

**Usage**

```
plotTraces(tfit)
```

**Arguments**

tfit                    a ternaryFit object

**Value**

The function creates a 2x2 grid of the four trace plots.

**Author(s)**

Matthew N. McCall and Anthony Almudevar

**See Also**

Almudevar A, McCall MN, McMurray H, Land H (2011). Fitting Boolean Networks from Steady State Perturbation Data, *Statistical Applications in Genetics and Molecular Biology*, 10(1): Article 47.

**Examples**

```
ssObj <- matrix(c(1,1,1,0,1,1,0,0,1),nrow=3)
pObj <- matrix(c(1,0,0,0,1,0,0,0,1),nrow=3)
tnfitObj <- tnetfit(ssObj, pObj)
plotTraces(tnfitObj)
```

---

ternaryFit-class      *Ternary Network Fit*

---

### Description

This is a class representation of the output of the ternary network fitting algorithm implemented in the function `tnetfit`.

### Creating Objects

While one can create their own objects using the function `ternaryFit()`, this is highly discouraged. Typically this class is created by running the `tnetfit` function.

### Slots

`perturbationObj`: a matrix of perturbation experiments. Rows are genes and columns are experiments.

`steadyStateObj`: a matrix of steady gene expression observations from a perturbation experiment. Rows are genes and columns are experiments.

`degreeObjMin`: a vector containing the in-degree of each node in the fit achieving the minimum score

`graphObjMin`: a matrix containing the parents of each node in the fit achieving the minimum score

`tableObjMin`: a matrix containing the table in the fit achieving the minimum score

`newScore`: the most recent score

`minScore`: the minimum score

`finalTemperature`: the final value of the temperature parameter

`traces`: a dataframe contain the traces for 4 parameters

`stageCount`: the number of stages

`xSeed`: the random seed.

`inputParams`: the `ternaryFitParameters` object used.

### Methods

All named elements can be accessed and set in the standard way (e.g. `xSeed(object)` and `xSeed(object)<-`).

### Author(s)

Matthew N. McCall and Anthony Almudevar

### See Also

`tnetpost`, `ternaryFitParameters-class`, `ternaryPost-class`. Almudevar A, McCall MN, McMurray H, Land H (2011). Fitting Boolean Networks from Steady State Perturbation Data, *Statistical Applications in Genetics and Molecular Biology*, 10(1): Article 47.

**Examples**

```

ssObj <- matrix(c(1,1,1,0,1,1,0,0,1),nrow=3)
pObj <- matrix(c(1,0,0,0,1,0,0,0,1),nrow=3)
rownames(ssObj) <- rownames(pObj) <- colnames(ssObj) <- colnames(pObj) <- c("Gene1","Gene2","Gene3")
tnfitObj <- tnetfit(ssObj, pObj)
class(tnfitObj)

```

---

ternaryFitParameters-class

*Ternary Network Fitting Parameters*


---

**Description**

This is a class representation of the input parameters for the ternary network fitting algorithm implemented in the function `tnetfit`.

**Creating Objects**

```
ternaryFitParameters()
```

This creates a `ternaryFitParameters` object with the default fitting parameters.

**Slots**

**perturbationType:** this parameter currently can only be set to 1

**scoreType:** the method to score networks. Can be set to either 1 or 2, corresponding to the score types in Almudevar et al. (2011).

**backupStage:** current fit is output periodically according to this parameter

**maxStage:** the maximum number of stages permitted. Ideally, the actual number of stages required until convergence should be much less than this value.

**maxTransition:** This parameter provides an adaptive truncation of the stage sample size. The stage terminates before the specified fixed sample size if the number of transitions resulting in a strict increase of the score reaches this value. If the sampler is in steady state, then this count should be approximately half the number of transitions in which the score changes value.

**epsilon:** Convergence tolerance.

**beta0:** Algorithm terminates when this number of consecutive convergence events have occurred.

**chi0:** The target initial acceptance rate. This should be close to 1, although setting it too close will increase computation time.

**delta:** The increment change in steady state distribution between stages (as variational distance). Larger values tend to decrease computation time, but too large a value will result in spurious convergence.

**ne:** The fixed sample size (number of MCMC transitions) per stage.

**m0:** The sample size (number of transitions) used to determine the initial temperature.

**maxDegree:** Maximum number of parents per node permitted in model topology.

**pAddParent:** This is the probability of adding a parent to a randomly selected node in the proposal function.

**pExchangeParent:** This parameter gives the probability of a parent exchange in the proposal function. The AddParent operation takes precedence, so this probability should be interpreted as being conditional on the rejection of the AddParent operation.

**neighborDegree:** Number of applications of the proposal function.

**pNeighborhood:** Vector of probabilities denoted, which generates the random number of proposal function iterations. The length is one less than neighborDegree. If neighborDegree equals 1 then no iteration is performed, and this vector is ignored.

**rho:** Weight parameter for the exponential smoothing of the variance estimate. For no smoothing set to 1.

**edgePenalty:** This parameter provides a complexity penalty. This number times the number of edges is added to the score. To apply no penalty set this parameter to 0.

## Methods

All named elements can be accessed and set in the standard way (e.g. `scoreType(object)` and `scoreType(object)<-`).

## Author(s)

Matthew N. McCall and Anthony Almudevar

## See Also

`tnetfit`, `ternaryFit-class`, `ternaryPost-class`. Almudevar A, McCall MN, McMurray H, Land H (2011). Fitting Boolean Networks from Steady State Perturbation Data, *Statistical Applications in Genetics and Molecular Biology*, 10(1): Article 47.

## Examples

```
# create an instance
ternaryFitParameters()
```

---

<code>ternaryPost-class</code>	<i>Ternary Network Posterior</i>
--------------------------------	----------------------------------

---

## Description

This is a class representation of the output of the ternary network posterior sampling algorithm implemented in the function `tnetpost`.

## Creating Objects

While one can create their own objects using the function `ternaryPost()`, this is highly discouraged. Typically this class is created by running the `tnetpost` function.

**Slots**

**perturbationObj:** a matrix of perturbation experiments. Rows are genes and columns are experiments.

**steadyStateObj:** a matrix of steady gene expression observations from a perturbation experiment. Rows are genes and columns are experiments.

**scores:** the score of each sample

**degreeObjs:** the in-degree vector for each sample

**graphObjs:** the graph matrix for each sample

**tableObjs:** the table matrix for each sample

**inputParams:** the ternaryFitParameters object used

**Methods**

All named elements can be accessed and set in the standard way (e.g. `scores(object)` and `scores(object)<-`).

**Author(s)**

Matthew N. McCall and Anthony Almudevar

**See Also**

`tnetfit`, `ternaryFitParameters-class`, `ternaryFit-class`. Almudevar A, McCall MN, McMurray H, Land H (2011). Fitting Boolean Networks from Steady State Perturbation Data, *Statistical Applications in Genetics and Molecular Biology*, 10(1): Article 47.

**Examples**

```
ssObj <- matrix(c(1,1,1,0,1,1,0,0,1),nrow=3)
pObj <- matrix(c(1,0,0,0,1,0,0,0,1),nrow=3)
rownames(ssObj) <- rownames(pObj) <- colnames(ssObj) <- colnames(pObj) <- c("Gene1","Gene2","Gene3")
tnfitObj <- tnetfit(ssObj, pObj)
tnpostObj <- tnetpost(tnfitObj, mdelta=10, msample=10)
class(tnpostObj)
```

---

`tnetfit`*Ternary Network Fitting*

---

**Description**

This function fits a ternary network based on perturbation experiments.

**Usage**

```
tnetfit(steadyStateObj, perturbationObj, params=ternaryFitParameters(),
xSeed=NA)
```

**Arguments**

steadyStateObj a matrix of steady gene expression observations from a perturbation experiment. Rows are genes and columns are experiments.

perturbationObj a matrix of perturbation experiments. Rows are genes and columns are experiments.

params a ternaryFitParameters object

xSeed an integer random seed. If NA, a random seed is generated.

**Value**

The function returns a ternaryFit object.

**Author(s)**

Matthew N. McCall and Anthony Almudevar

**See Also**

Almudevar A, McCall MN, McMurray H, Land H (2011). Fitting Boolean Networks from Steady State Perturbation Data, *Statistical Applications in Genetics and Molecular Biology*, 10(1): Article 47.

**Examples**

```
ssObj <- matrix(c(1,1,1,0,1,1,0,0,1),nrow=3)
pObj <- matrix(c(1,0,0,0,1,0,0,0,1),nrow=3)
rownames(ssObj) <- rownames(pObj) <- colnames(ssObj) <- colnames(pObj) <- c("Gene1","Gene2","Gene3")
tnfitObj <- tnetfit(ssObj, pObj)
```

---

tnetpost

*Ternary Network Posterior Sampling*

---

**Description**

This function samples from the posterior density of a ternary network based on perturbation experiments.

**Usage**

```
tnetpost(tfit, mdelta=as.integer(2000), msample=as.integer(2000), temperatureScale=1.0, xSeed=NA)
```

**Arguments**

tfit a ternaryFit object

mdelta number of transitions between samples

msample number of samples

temperatureScale the final temperature is multiplied by this value for sampling

xSeed an integer random seed. If NA, a random seed is generated.



**Value**

The function returns a ternaryPost object.

**Author(s)**

Matthew N. McCall and Anthony Almudevar

**See Also**

Almudevar A, McCall MN, McMurray H, Land H (2011). Fitting Boolean Networks from Steady State Perturbation Data, *Statistical Applications in Genetics and Molecular Biology*, 10(1): Article 47.

**Examples**

```
ssObj <- matrix(c(1,1,1,0,1,1,0,0,1),nrow=3)
pObj <- matrix(c(1,0,0,0,1,0,0,0,1),nrow=3)
rownames(ssObj) <- rownames(pObj) <- colnames(ssObj) <- colnames(pObj) <- c("Gene1","Gene2","Gene3")
tnfitObj <- tnetfit(ssObj, pObj)
tnpostObj <- tnetpost(tnfitObj, mdelta=10, msample=10)
```

# Index

- \*Topic **classes**
  - ternaryFit-class, 4
  - ternaryFitParameters-class, 5
  - ternaryPost-class, 6
- \*Topic **manip**
  - attractorSummary, 1
  - graphPosterior, 2
  - plotTraces, 3
  - tnetfit, 7
  - tnetpost, 8
- attractorSummary, 1
- backupStage
  - (ternaryFitParameters-class), 5
- backupStage, ternaryFitParameters-method
  - (ternaryFitParameters-class), 5
- backupStage<-
  - (ternaryFitParameters-class), 5
- backupStage<-, ternaryFitParameters-method
  - (ternaryFitParameters-class), 5
- beta0 (ternaryFitParameters-class), 5
- beta0, ternaryFitParameters-method
  - (ternaryFitParameters-class), 5
- beta0<- (ternaryFitParameters-class), 5
- beta0<-, ternaryFitParameters-method
  - (ternaryFitParameters-class), 5
- chi0 (ternaryFitParameters-class), 5
- chi0, ternaryFitParameters-method
  - (ternaryFitParameters-class), 5
- chi0<- (ternaryFitParameters-class), 5
- chi0<-, ternaryFitParameters-method
  - (ternaryFitParameters-class), 5
- class:ternaryFit (ternaryFit-class), 4
- class:ternaryFitParameters
  - (ternaryFitParameters-class), 5
- class:ternaryPost (ternaryPost-class), 6
- degreeObjMin (ternaryFit-class), 4
- degreeObjMin, ternaryFit-method
  - (ternaryFit-class), 4
- degreeObjMin<- (ternaryFit-class), 4
- degreeObjMin<-, ternaryFit-method
  - (ternaryFit-class), 4
- degreeObjs (ternaryPost-class), 6
- degreeObjs, ternaryPost-method
  - (ternaryPost-class), 6
- degreeObjs<- (ternaryPost-class), 6
- degreeObjs<-, ternaryPost-method
  - (ternaryPost-class), 6
- delta (ternaryFitParameters-class), 5
- delta, ternaryFitParameters-method
  - (ternaryFitParameters-class), 5
- delta<- (ternaryFitParameters-class), 5
- delta<-, ternaryFitParameters-method
  - (ternaryFitParameters-class), 5
- dim, ternaryFit-method
  - (ternaryFit-class), 4
- dim, ternaryPost-method
  - (ternaryPost-class), 6
- edgePenalty
  - (ternaryFitParameters-class), 5
- edgePenalty, ternaryFitParameters-method
  - (ternaryFitParameters-class), 5
- edgePenalty<-
  - (ternaryFitParameters-class), 5
- edgePenalty<-, ternaryFitParameters-method
  - (ternaryFitParameters-class), 5
- epsilon (ternaryFitParameters-class), 5
- epsilon, ternaryFitParameters-method
  - (ternaryFitParameters-class), 5
- epsilon<- (ternaryFitParameters-class), 5
- epsilon<-, ternaryFitParameters-method
  - (ternaryFitParameters-class), 5
- finalTemperature (ternaryFit-class), 4
- finalTemperature, ternaryFit-method
  - (ternaryFit-class), 4
- finalTemperature<- (ternaryFit-class), 4
- finalTemperature<-, ternaryFit-method
  - (ternaryFit-class), 4
- graphObjMin (ternaryFit-class), 4
- graphObjMin, ternaryFit-method
  - (ternaryFit-class), 4
- graphObjMin<- (ternaryFit-class), 4

- graphObjMin<- , ternaryFit-method  
(ternaryFit-class), 4
- graphObjs (ternaryPost-class), 6
- graphObjs, ternaryPost-method  
(ternaryPost-class), 6
- graphObjs<- (ternaryPost-class), 6
- graphObjs<- , ternaryPost-method  
(ternaryPost-class), 6
- graphPosterior, 2
  
- initialize, ternaryFit-method  
(ternaryFit-class), 4
- initialize, ternaryFitParameters-method  
(ternaryFitParameters-class), 5
- initialize, ternaryPost-method  
(ternaryPost-class), 6
- inputParams (ternaryFit-class), 4
- inputParams, ternaryFit-method  
(ternaryFit-class), 4
- inputParams, ternaryPost-method  
(ternaryPost-class), 6
- inputParams<- (ternaryFit-class), 4
- inputParams<- , ternaryFit-method  
(ternaryFit-class), 4
- inputParams<- , ternaryPost-method  
(ternaryPost-class), 6
  
- m0 (ternaryFitParameters-class), 5
- m0, ternaryFitParameters-method  
(ternaryFitParameters-class), 5
- m0<- (ternaryFitParameters-class), 5
- m0<- , ternaryFitParameters-method  
(ternaryFitParameters-class), 5
- maxDegree (ternaryFitParameters-class),  
5
- maxDegree, ternaryFitParameters-method  
(ternaryFitParameters-class), 5
- maxDegree<-  
(ternaryFitParameters-class), 5
- maxDegree<- , ternaryFitParameters-method  
(ternaryFitParameters-class), 5
- maxStage (ternaryFitParameters-class), 5
- maxStage, ternaryFitParameters-method  
(ternaryFitParameters-class), 5
- maxStage<-  
(ternaryFitParameters-class), 5
- maxStage<- , ternaryFitParameters-method  
(ternaryFitParameters-class), 5
- maxTransition  
(ternaryFitParameters-class), 5
- maxTransition, ternaryFitParameters-method  
(ternaryFitParameters-class), 5
  
- maxTransition<-  
(ternaryFitParameters-class), 5
- maxTransition<- , ternaryFitParameters-method  
(ternaryFitParameters-class), 5
- minScore (ternaryFit-class), 4
- minScore, ternaryFit-method  
(ternaryFit-class), 4
- minScore<- (ternaryFit-class), 4
- minScore<- , ternaryFit-method  
(ternaryFit-class), 4
  
- ne (ternaryFitParameters-class), 5
- ne, ternaryFitParameters-method  
(ternaryFitParameters-class), 5
- ne<- (ternaryFitParameters-class), 5
- ne<- , ternaryFitParameters-method  
(ternaryFitParameters-class), 5
- neighborDegree  
(ternaryFitParameters-class), 5
- neighborDegree, ternaryFitParameters-method  
(ternaryFitParameters-class), 5
- neighborDegree<-  
(ternaryFitParameters-class), 5
- neighborDegree<- , ternaryFitParameters-method  
(ternaryFitParameters-class), 5
- newScore (ternaryFit-class), 4
- newScore, ternaryFit-method  
(ternaryFit-class), 4
- newScore<- (ternaryFit-class), 4
- newScore<- , ternaryFit-method  
(ternaryFit-class), 4
  
- pAddParent  
(ternaryFitParameters-class), 5
- pAddParent, ternaryFitParameters-method  
(ternaryFitParameters-class), 5
- pAddParent<-  
(ternaryFitParameters-class), 5
- pAddParent<- , ternaryFitParameters-method  
(ternaryFitParameters-class), 5
- perturbationObj (ternaryFit-class), 4
- perturbationObj, ternaryFit-method  
(ternaryFit-class), 4
- perturbationObj, ternaryPost-method  
(ternaryPost-class), 6
- perturbationObj<- (ternaryFit-class), 4
- perturbationObj<- , ternaryFit-method  
(ternaryFit-class), 4
- perturbationObj<- , ternaryPost-method  
(ternaryPost-class), 6
- perturbationType  
(ternaryFitParameters-class), 5

- perturbationType, ternaryFitParameters-method  
(ternaryFitParameters-class), 5
- perturbationType<-  
(ternaryFitParameters-class), 5
- perturbationType<- , ternaryFitParameters-method  
(ternaryFitParameters-class), 5
- pExchangeParent  
(ternaryFitParameters-class), 5
- pExchangeParent, ternaryFitParameters-method  
(ternaryFitParameters-class), 5
- pExchangeParent<-  
(ternaryFitParameters-class), 5
- pExchangeParent<- , ternaryFitParameters-method  
(ternaryFitParameters-class), 5
- plotTraces, 3
- pNeighborhood  
(ternaryFitParameters-class), 5
- pNeighborhood, ternaryFitParameters-method  
(ternaryFitParameters-class), 5
- pNeighborhood<-  
(ternaryFitParameters-class), 5
- pNeighborhood<- , ternaryFitParameters-method  
(ternaryFitParameters-class), 5
  
- rho (ternaryFitParameters-class), 5
- rho, ternaryFitParameters-method  
(ternaryFitParameters-class), 5
- rho<- (ternaryFitParameters-class), 5
- rho<- , ternaryFitParameters-method  
(ternaryFitParameters-class), 5
  
- scores (ternaryPost-class), 6
- scores, ternaryPost-method  
(ternaryPost-class), 6
- scores<- (ternaryPost-class), 6
- scores<- , ternaryPost-method  
(ternaryPost-class), 6
- scoreType (ternaryFitParameters-class),  
5
- scoreType, ternaryFitParameters-method  
(ternaryFitParameters-class), 5
- scoreType<-  
(ternaryFitParameters-class), 5
- scoreType<- , ternaryFitParameters-method  
(ternaryFitParameters-class), 5
- show, ternaryFit-method  
(ternaryFit-class), 4
- show, ternaryFitParameters-method  
(ternaryFitParameters-class), 5
- show, ternaryPost-method  
(ternaryPost-class), 6
- stageCount (ternaryFit-class), 4
- stageCount, ternaryFit-method  
(ternaryFit-class), 4
- stageCount<- (ternaryFit-class), 4
- stageCount<- , ternaryFit-method  
(ternaryFit-class), 4
- steadyStateObj (ternaryFit-class), 4
- steadyStateObj, ternaryFit-method  
(ternaryFit-class), 4
- steadyStateObj, ternaryPost-method  
(ternaryPost-class), 6
- steadyStateObj<- (ternaryFit-class), 4
- steadyStateObj<- , ternaryFit-method  
(ternaryFit-class), 4
- steadyStateObj<- , ternaryPost-method  
(ternaryPost-class), 6
  
- tableObjMin (ternaryFit-class), 4
- tableObjMin, ternaryFit-method  
(ternaryFit-class), 4
- tableObjMin<- (ternaryFit-class), 4
- tableObjMin<- , ternaryFit-method  
(ternaryFit-class), 4
- tableObjs (ternaryPost-class), 6
- tableObjs, ternaryPost-method  
(ternaryPost-class), 6
- tableObjs<- (ternaryPost-class), 6
- tableObjs<- , ternaryPost-method  
(ternaryPost-class), 6
- ternaryFit (ternaryFit-class), 4
- ternaryFit-class, 4
- ternaryFit-methods (ternaryFit-class), 4
- ternaryFitParameters  
(ternaryFitParameters-class), 5
- ternaryFitParameters-class, 5
- ternaryFitParameters-methods  
(ternaryFitParameters-class), 5
- ternaryPost (ternaryPost-class), 6
- ternaryPost-class, 6
- ternaryPost-methods  
(ternaryPost-class), 6
- tnetfit, 7
- tnetpost, 8
- traces (ternaryFit-class), 4
- traces, ternaryFit-method  
(ternaryFit-class), 4
- traces<- (ternaryFit-class), 4
- traces<- , ternaryFit-method  
(ternaryFit-class), 4
  
- xSeed (ternaryFit-class), 4
- xSeed, ternaryFit-method  
(ternaryFit-class), 4
- xSeed<- (ternaryFit-class), 4

xSeed<- ,ternaryFit-method  
(ternaryFit-class), 4