

# Package ‘geneARMA’

January 2, 2012

**Type** Package

**Title** Simulate, model, and display data from a time-course microarray experiment with periodic gene expression

**Version** 1.0

**Date** 2009-10-06

**Author** Timothy McMurry <tmcmurry@depaul.edu>, Arthur Berg <berg@psu.edu>

**Maintainer** Timothy McMurry <tmcmurry@depaul.edu>

**Depends** mvtnorm

**Description** Fit models for periodic gene expression to time-course microarray data in a normal mixture model framework with mean approximated by a truncated Fourier series and covariance structure modeled by an ARMA(p,q) process. Estimation is performed with the EM algorithm.

**License** GPL-3

**LazyLoad** yes

**Repository** CRAN

**Date/Publication** 2009-10-19 19:43:48

## R topics documented:

geneARMA-package . . . . .	2
geneARMAfit . . . . .	2
geneARMAsim . . . . .	4
plot.geneARMA . . . . .	5

<b>Index</b>	<b>7</b>
--------------	----------

geneARMA-package

*Simulate, model, and display data from a time-course microarray experiment with periodic gene expression*

---

**Description**

Fit models for periodic gene expression to time-course microarray data in a normal mixture model framework with mean approximated by a truncated Fourier series and covariance structure modeled by an ARMA(p,q) process. Estimation is performed with the EM algorithm.

**Details**

Package: geneARMA  
Type: Package  
Version: 1.0  
Date: 2009-10-07  
License: GPL-3  
LazyLoad: yes

**Author(s)**

Timothy McMurry and Arthur Berg

Maintainer: Timothy McMurry <tmcmurry@depaul.edu>

**References**

Ning Li, et al. Functional clustering of periodic transcriptional profiles through ARMA(p,q)

**Examples**

```
set.seed(100)
Data <- geneARMAsim(400, ars=c(.5, .1))
f1 <- geneARMAfit(Data$Y, Data$tm, 2, 2, 2, 0, eps.conv = .001, max.iter = 15, tau.init=c(.25, .45))
plot(f1, y=NULL, "all.means")
plot(f1, y=NULL, "single.cluster", j=2)
```

---

geneARMAfit*Fit a periodic model to time-course gene expression data.*

---

**Description**

Fits a Fourier series model to time-course gene expression data. Data are classified into one of  $J$  groups, each with different mean parameters for the gene expression profiles. The signal mean is modeled by an order  $K$  Fourier series. The residuals are modeled as a Gaussian ARMA( $p,q$ ) process. Fitting is done by a modified EM algorithm, where some updates are estimated by numerical approximation.

**Usage**

```
geneARMAfit(Y, times, J = 4, K = 2, p = 2, q = 0, arma.skip = 10, eps.conv = 0.01, max.iter = Inf, print.up
```

**Arguments**

<code>Y</code>	A matrix with the normalized gene expression data. The rows should be genes and the columns observation times.
<code>times</code>	A vector containing the times at which gene expression was measured. The order must correspond to the order of the data in <code>Y</code> .
<code>J</code>	The number of clusters to fit.
<code>K</code>	The number of terms in the Fourier sum.
<code>p</code>	The autoregressive order.
<code>q</code>	The moving-average order.
<code>arma.skip</code>	The number of iterations the algorithm should run before attempting to fit the time series model. It is often helpful to let the algorithm do approximate clustering first, and then refine the time series fit.
<code>eps.conv</code>	The convergence criterion. If the change in all estimated terms is less than <code>eps.conv</code> , the program stops.
<code>max.iter</code>	The maximum number of iterations to run before stopping.
<code>print.updates</code>	Turn on/off output produced as the program runs.
<code>omega.init</code>	Initial cluster probabilities. Uniform if left unspecified.
<code>fourier.init</code>	A $J$ by $2K+1$ matrix with initial values for the Fourier coefficients. The columns are constant term, cosine term, sine term, cosine term, sine term,... where lower frequency terms come first. Filled with iid $N(0,1)$ if left unspecified.
<code>tau.init</code>	Initial values for the period of the signal. Defaults to a range of values covering some fraction of the overall length of time measured.
<code>sigsq.init</code>	Initial value for the time series innovation variance. The initial value should be very large, probably hundreds of times the actual, in order for the algorithm to find good clusters.

**Details**

An EM algorithm is used to fit the model and cluster the data. The results are somewhat sensitive to initial conditions. While the program attempts to assign reasonable values, much better results can often be obtained by setting the starting values by hand.

Some of the parameters do not have closed form maximum likelihood estimates. On each iteration these parameters are updated by a 1-step Newton-Raphson estimate. This may cause slightly unsteady estimates near convergence.

**Value**

Returns a list with two components.

Theta is a list of all the model parameters estimated by the EM algorithm.

Data is a list containing the original data and observation times.

**Author(s)**

Timothy McMurry and Arthur Berg

**References**

Ning Li, et al. Functional clustering of periodic transcriptional profiles through ARMA(p,q)

**Examples**

```
set.seed(100)
Data <- geneARMAsim(400, ars=c(.5, .1))
f1 <- geneARMAfit(Data$Y, Data$tm, 2, 2, 2, 0, eps.conv = .001, max.iter = 15, tau.init=c(.25, .45))
plot(f1, y=NULL, "all.means")
plot(f1, y=NULL, "single.cluster", j=2)
```

---

geneARMAsim

*Simulate a periodic model of time-course gene expression data.*

---

**Description**

A function which generates data to test the geneARMAfit results.

**Usage**

```
geneARMAsim(n = 100, J = 2, K = 2, sigsq = 0.01, ars = numeric(0), mas = numeric(0), length.out = 21, four
```

**Arguments**

n	simulated sample size.
J	number of clusters.
K	number of Fourier terms.
sigsq	innovation variance.
ars	a vector of AR coefficients, where the time series is parameterized as $X[t] = ars[1] * X[t-1] + ars[2]*X[t-2] + \dots + e[t] + mas[1]*e[t-1] + \dots$
mas	a vector of MA coefficients, where the time series is parameterized as $X[t] = ars[1] * X[t-1] + ars[2]*X[t-2] + \dots + e[t] + mas[1]*e[t-1] + \dots$
length.out	number of equally spaced time points at which each gene is obsered.
fourier.coefs	A J by 2K+1 matrix with fourier coefficients. The rows are clusters, and the columns are constant term, cosine term, sine term, cosine term, sine term,... where lower frequency terms come first.
tau	A vector of length J specifying the frequencies for the different clusters.

**Value**

Returns a list with two components

Y	The simulated data
tm	The corresponding time points

**Author(s)**

Timothy McMurry and Arthur Berg

**References**

Ning Li, et al. Functional clustering of periodic transcriptional profiles through ARMA(p,q)

**Examples**

```
set.seed(100)
Data <- geneARMAsim(400, ars=c(.5, .1))
f1 <- geneARMAfit(Data$Y, Data$tm, 2, 2, 2, 0, eps.conv = .001, max.iter = 15, tau.init=c(.25, .45))
plot(f1, y=NULL, "all.means")
plot(f1, y=NULL, "single.cluster", j=2)
```

---

plot.geneARMA	<i>Plot method for "geneARMA" objects. Displays fitted models and gene expression by group and time.</i>
---------------	--

---

**Description**

Produces two types of plots depending on "type" variable. The first, "all.means," shows all periodic signals fitted to each of the clusters, with the last cluster, which is typically used for the group of genes with no abnormal expression, colored black. The second type of plot shows the gene expression values by time with the fitted mean overlaid.

**Usage**

```
## S3 method for class 'geneARMA'
plot(x, y=NULL, type = c("all.means", "single.cluster"), j, ...)
```

**Arguments**

x	A fitted model object of class "geneARMA" produced by geneARMAfit(...).
y	Unused. Supplied for compatibility with plot method.
type	all.means produces a plot of all cluster means. single.cluster plots the data for a single cluster with the mean overlaid.
j	is used to indicate which cluster to plot for single.cluster plots only.
...	

**Author(s)**

Timothy McMurry and Arthur Berg

**References**

Ning Li, et al. Functional clustering of periodic transcriptional profiles through ARMA(p,q)

**Examples**

```
set.seed(100)
Data <- geneARMAsim(400, ars=c(.5, .1))
f1 <- geneARMAfit(Data$Y, Data$tm, 2, 2, 2, 0, eps.conv = .001, max.iter = 15, tau.init=c(.25, .45))
plot(f1, y=NULL, "all.means")
plot(f1, y=NULL, "single.cluster", j=2)
```

# Index

`geneARMA (geneARMA-package)`, [2](#)

`geneARMA-package`, [2](#)

`geneARMAfit`, [2](#)

`geneARMAsim`, [4](#)

`plot.geneARMA`, [5](#)