

# Package: Hiiragi2013 (via r-universe)

October 10, 2026

**Type** Package

**Title** Cell-to-cell expression variability followed by signal reinforcement progressively segregates early mouse lineages

**Version** 1.48.1

**Description** This package contains the experimental data and a complete executable transcript (vignette) of the statistical analysis presented in the paper ``Cell-to-cell expression variability followed by signal reinforcement progressively segregates early mouse lineages" by Y. Ohnishi, W. Huber, A. Tsumura, M. Kang, P. Xenopoulos, K. Kurimoto, A. K. Oles, M. J. Arauzo-Bravo, M. Saitou, A.-K. Hadjantonakis and T. Hiiragi; Nature Cell Biology (2014) 16(1): 27-37. doi: 10.1038/ncb2881."

**License** Artistic-2.0

**LazyLoad** true

**Depends** R (>= 4.0.0), Biobase

**Imports** MASS, RColorBrewer, cluster, gplots, grid, lattice, latticeExtra, genefilter

**Suggests** affy, BiocStyle, boot, clue, geneplotter, gtools, KEGGREST, mouse4302.db, xtable

**biocViews** ExperimentData, MicroarrayData, qPCRData, ReproducibleResearch

**Config/pak/sysreqs** libjpeg-dev libpng-dev libxml2-dev libssl-dev zlib1g-dev

**Repository** Bioconductor 3.23

**Date/Publication** 2026-06-22 08:30:07 UTC

**RemoteUrl** <https://github.com/bioc/Hiiragi2013>

**RemoteRef** RELEASE\_3\_23

**RemoteSha** 586aff9328ea2696a52c79b2f1cdad786c909622

## Contents

getDifferentialExpressedGenes . . . . . 2

|                          |           |
|--------------------------|-----------|
| MDSplot . . . . .        | 3         |
| myHeatmap . . . . .      | 4         |
| myHeatmap2 . . . . .     | 5         |
| pamCluster . . . . .     | 6         |
| plotProjection . . . . . | 7         |
| x . . . . .              | 8         |
| xq . . . . .             | 10        |
| xql . . . . .            | 11        |
| <b>Index</b>             | <b>14</b> |

---

|                                            |
|--------------------------------------------|
| getDifferentialExpressedGenes              |
| <i>Find Differentially Expressed Genes</i> |

---

**Description**

Returns differentially expressed genes between two conditions.

**Usage**

getDifferentialExpressedGenes(x, groups, g1, g2, theta = 0.5, FDRcutoff = 0.05)

**Arguments**

|           |                                                                                                                               |
|-----------|-------------------------------------------------------------------------------------------------------------------------------|
| x         | an <a href="#">ExpressionSet</a> containing a matrix of expression values with rows representing features and columns samples |
| groups    | list of integer vectors specifying the grouping of samples                                                                    |
| g1        | character string specifying the name of the first tested group from groups                                                    |
| g2        | character string specifying the name of the second tested group from groups                                                   |
| theta     | numeric, probability with values in [0,1] used for quantile filtering of variance                                             |
| FDRcutoff | numeric, p-value cutoff                                                                                                       |

**Details**

Differentially expressed features are selected based on a t-test with the adjusted p-value cutoff specified by FDRcutoff.

The filtering selects only features whose variance is greater than the quantile defined by the probability theta.

**Value**

Integer vector containing indices of differentially expressed features from x.

**Author(s)**

Wolfgang Huber, 2013

**Examples**

```

data(x)
groups = with(pData(x), list(
  "E3.25"      = which(genotype=="WT" & Embryonic.day=="E3.25"),
  "E3.5 (EPI)" = which(genotype=="WT" & Embryonic.day=="E3.5" & lineage=="EPI"),
  "E3.5 (PE)"  = which(genotype=="WT" & Embryonic.day=="E3.5" & lineage=="PE")))

# get a list of differentially expressed genes along the transition from E3.25 to E3.5
de = union(
  getDifferentialExpressedGenes(x, groups, "E3.25", "E3.5 (EPI)"),
  getDifferentialExpressedGenes(x, groups, "E3.25", "E3.5 (PE)"))
fData(x[de,])$symbol

```

MDSplot

*Multidimensional Scaling Plots***Description**

Visualization of data similarity using non-metric multidimensional scaling.

**Usage**

```

MDSplot(
  x,
  mask,
  flip = integer(0),
  rotation = 0,
  cex = 2,
  col = x$sampleColour,
  panellabel,
  pointlabel
)

```

**Arguments**

|            |                                                                                                                               |
|------------|-------------------------------------------------------------------------------------------------------------------------------|
| x          | an <a href="#">ExpressionSet</a> containing a matrix of expression values with rows representing features and columns samples |
| mask       | a logical vector specifying the masking of data points; if missing all points are drawn                                       |
| flip       | an integer vector containing indices of columns of the distance matrix whose sign should be flipped                           |
| rotation   | numeric, angle in radians by which the plot should be rotated                                                                 |
| cex        | numeric, the value giving the amount by which text and symbols should be scaled relative to the current setting               |
| col        | character vector specifying the colors of points                                                                              |
| panellabel | character, the label of the figure panel                                                                                      |
| pointlabel | character vector containing labels corresponding to samples; if specified labels are drawn instead of points                  |

**Value**

The result of the function is a plot.

**Author(s)**

Wolfgang Huber, 2013

**Examples**

```
## data preparation
data(x)
y = x[, with(pData(x), Embryonic.day=="E3.25")]

## define colours for plotting
colours = setNames(RColorBrewer::brewer.pal(10, "Paired")[c(9, 7)], c("WT", "FGF4-KO"))[y$genotype]
stopifnot(!any(is.na(colours)))

## select 100 most variable genes for plotting
selMDS = order(genefilter::rowVars(exprs(y)), decreasing=TRUE)[seq_len(100)]
MDSplot(y[selMDS,], col=colours)
```

---

myHeatmap

*Plot Heatmaps*


---

**Description**

Plots a heatmap with features (rows) and samples (columns) reordered according to hierarchical clustering.

**Usage**

```
myHeatmap(x, collapseDuplicateFeatures = FALSE, haveColDend = FALSE)
```

**Arguments**

**x** an [ExpressionSet](#) containing a matrix of expression values with rows representing features and columns samples

**collapseDuplicateFeatures** logical, if TRUE multiple features per gene will be averaged

**haveColDend** logical, if TRUE displays the grouping of samples using a dendrogram

**Value**

The result of the function is a plot.

**Author(s)**

Wolfgang Huber, 2013

**See Also**[myHeatmap2](#)**Examples**

```

data(x)
groups = with(pData(x), list(
  "E3.25"      = which(genotype=="WT" & Embryonic.day=="E3.25"),
  "E3.5 (EPI)" = which(genotype=="WT" & Embryonic.day=="E3.5" & lineage=="EPI"),
  "E3.5 (PE)"  = which(genotype=="WT" & Embryonic.day=="E3.5" & lineage=="PE")))
samples = unlist(groups)

# heatmap of differentially expressed genes along the transition from E3.25 to E3.5
de = union(
  getDifferentialExpressedGenes(x, groups, "E3.25", "E3.5 (EPI)"),
  getDifferentialExpressedGenes(x, groups, "E3.25", "E3.5 (PE)"))
myHeatmap(x[de, samples], collapseDuplicateFeatures=TRUE)

```

myHeatmap2

*Plot Heatmaps***Description**

Plots a heatmap with clustering of rows and columns specified by the rowGroups and colGroups, respectively.

**Usage**

```

myHeatmap2(
  x,
  rowGroups = factor(rep(1, nrow(x))),
  colGroups = factor(rep(1, ncol(x))),
  keeprownames = TRUE,
  colors = colorRampPalette(brewer.pal(9, "Blues")[-1])(100),
  ...
)

```

**Arguments**

|              |                                                                                                                               |
|--------------|-------------------------------------------------------------------------------------------------------------------------------|
| x            | an <a href="#">ExpressionSet</a> containing a matrix of expression values with rows representing features and columns samples |
| rowGroups    | factor which length aligns with rows of x                                                                                     |
| colGroups    | factor which length aligns with columns of x                                                                                  |
| keeprownames | logical, if TRUE displays row names                                                                                           |
| colors       | color palette used to plot the heatmap                                                                                        |
| ...          | arguments passed to the internal ordermat function                                                                            |

**Value**

The result of the function is a plot.

**Author(s)**

Wolfgang Huber, 2013

**See Also**

[myHeatmap](#)

**Examples**

```
## The function is currently defined as
function (x, rowGroups = factor(rep(1, nrow(x))), colGroups = factor(rep(1, ncol(x))),
  keeprownames = TRUE, colors = colorRampPalette(brewer.pal(9, "Blues")[-1])(100), ...) {
  x = ordermat(x, rowGroups, ...)
  x = ordermat(t(x), colGroups, ...)
  if (!keeprownames)
    colnames(x) = NULL
  print(levelplot(x, aspect = "fill", xlab = "", ylab = "",
    scales = list(x = list(rot = 90), raster = TRUE), col.regions = colors,
    colorkey = list(space = "left", height = 0.15, useRaster = TRUE)))
}
```

---

pamCluster

*Clustering of Most Variable Genes*

---

**Description**

The function `pamCluster` selects the `ngenes` most variable genes and performs their clustering using the partitioning around medoids method [pam](#).

**Usage**

```
pamCluster(ngenes, x, k = 2)
```

**Arguments**

|                     |                                                                 |
|---------------------|-----------------------------------------------------------------|
| <code>ngenes</code> | numeric, the number of most variable genes to select            |
| <code>x</code>      | <a href="#">ExpressionSet</a> containing gene expression values |
| <code>k</code>      | positive integer specifying the number of clusters              |

**Value**

Integer vector specifying the clustering.

**Author(s)**

Wolfgang Huber

**See Also**

pam

**Examples**

```
data("x")
y = x[, x$Embryonic.day=="E3.5"]

## perform the clustering
pc = pamCluster(50, y, k=3)

## display clustering vs. sample lineage
plot(as.factor(pData(y)$lineage), pc, yaxt="n", xlab="lineage", ylab="cluster")
```

---

plotProjection

---

*Overview of the Sample Expression*


---

**Description**

Plots a projection of sample expression profiles on the differential expression signature.

**Usage**

```
plotProjection(projection, label, col, colourMap)
```

**Arguments**

|            |                                                                                                            |
|------------|------------------------------------------------------------------------------------------------------------|
| projection | a vector of scalar products between each sample's expression profile and differential expression signature |
| label      | character vector of sample names                                                                           |
| col        | sample colour palette                                                                                      |
| colourMap  | sample colour map                                                                                          |

**Value**

The result of the function is a plot.

**Author(s)**

Wolfgang Huber, 2013

**Examples**

```
## For illustration of use please see the package vignette
```

x

*Normalized Microarray Data***Description**

An [ExpressionSet](#) object containing the RMA normalized dataset in the assayData and annotation in the phenoData.

**Usage**

```
data(x)
```

**Format**

```
Formal class 'ExpressionSet' [package "Biobase"] with 7 slots
  ..@ experimentData :Formal class 'MIAME' [package "Biobase"] with 13 slots
    .. .. ..@ name      : chr ""
    .. .. ..@ lab       : chr ""
    .. .. ..@ contact    : chr ""
    .. .. ..@ title     : chr ""
    .. .. ..@ abstract   : chr ""
    .. .. ..@ url       : chr ""
    .. .. ..@ pubMedIds  : chr ""
    .. .. ..@ samples    : list()
    .. .. ..@ hybridizations : list()
    .. .. ..@ normControls : list()
    .. .. ..@ preprocessing :List of 2
    .. .. ..$ filenames : chr [1:101] "/tmp/RtmpI5T6yy/1_C32_IN.CEL" "/tmp/RtmpI5T6yy/2_C32_IN.CEL" "
    .. .. ..$ affyversion: chr NA
    .. .. ..@ other      :List of 1
    .. .. ..$ : chr ""
  .. .. ..@ __classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots
    .. .. ..$ :List of 2
    .. .. ..$ : int [1:3] 1 0 0
    .. .. ..$ : int [1:3] 1 1 0
  ..@ assayData      :<environment: 0x167aa270>
  ..@ phenoData       :Formal class 'AnnotatedDataFrame' [package "Biobase"] with 4 slots
    .. .. ..@ varMetadata : 'data.frame': 5 obs. of 1 variable:
    .. .. ..$ labelDescription: chr [1:5] NA NA NA NA ...
    .. .. ..@ data         : 'data.frame': 101 obs. of 5 variables:
    .. .. ..$ File.name     : chr [1:101] "1_C32_IN" "2_C32_IN" "3_C32_IN" "4_C32_IN" ...
    .. .. ..$ Embryonic.day : Factor w/ 3 levels "E3.25", "E3.5", ...: 1 1 1 1 1 1 1 1 1 1 ...
    .. .. ..$ Total.number.of.cells: Factor w/ 11 levels "32", "33", "34", ...: 1 1 1 1 1 1 1 1 1 1 ...
    .. .. ..$ lineage       : chr [1:101] "" "" "" "" "" ...
    .. .. ..$ genotype      : Factor w/ 2 levels "FGF4-KO", "WT": 2 2 2 2 2 2 2 2 2 2 ...
    .. .. ..@ dimLabels     : chr [1:2] "sampleNames" "sampleColumns"
  .. .. ..@ __classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots
```

```

.. .. .. .. ..@ .Data:List of 1
.. .. .. .. ..$ : int [1:3] 1 1 0
..@ featureData :Formal class 'AnnotatedDataFrame' [package "Biobase"] with 4 slots
.. .. ..@ varMetadata : 'data.frame': 3 obs. of 1 variable:
.. .. .. ..$ labelDescription: chr [1:3] NA NA NA
.. .. ..@ data : 'data.frame': 45101 obs. of 3 variables:
.. .. .. ..$ symbol : chr [1:45101] "Copg1" "Atp6v0d1" "Golga7" "Psph" ...
.. .. .. ..$ genename: chr [1:45101] "coatomer protein complex, subunit gamma 1" "ATPase, H+ transporti
.. .. .. ..$ ensembl : chr [1:45101] "ENSMUSG00000030058" "ENSMUSG00000013160" "ENSMUSG00000015341" "E
.. .. ..@ dimLabels : chr [1:2] "featureNames" "featureColumns"
.. .. ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots
.. .. .. .. ..@ .Data:List of 1
.. .. .. .. ..$ : int [1:3] 1 1 0
..@ annotation : chr "mouse4302"
..@ protocolData :Formal class 'AnnotatedDataFrame' [package "Biobase"] with 4 slots
.. .. ..@ varMetadata : 'data.frame': 1 obs. of 1 variable:
.. .. .. ..$ labelDescription: chr NA
.. .. ..@ data : 'data.frame': 101 obs. of 1 variable:
.. .. .. ..$ ScanDate: chr [1:101] "2011-03-16T04:33:05Z" "2011-03-16T04:42:32Z" "2011-03-16T04:51:54
.. .. ..@ dimLabels : chr [1:2] "sampleNames" "sampleColumns"
.. .. ..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots
.. .. .. .. ..@ .Data:List of 1
.. .. .. .. ..$ : int [1:3] 1 1 0
..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots
.. .. ..@ .Data:List of 4
.. .. .. ..$ : int [1:3] 3 0 2
.. .. .. ..$ : int [1:3] 2 22 0
.. .. .. ..$ : int [1:3] 1 3 0
.. .. .. ..$ : int [1:3] 1 0 0

```

## References

Ohnishi et al., 2014

## See Also

[xq](#), [xql](#)

## Examples

```

data(x)
x
pData(x)

```

xq

*qPCR Gene Expression***Description**

An [ExpressionSet](#) object containing single-cell gene expression levels measured by qPCR.

**Usage**

```
data(xq)
```

**Format**

```
Formal class 'ExpressionSet' [package "Biobase"] with 7 slots
  ..@ experimentData :Formal class 'MIAME' [package "Biobase"] with 13 slots
    .. .. ..@ name      : chr ""
    .. .. ..@ lab       : chr ""
    .. .. ..@ contact    : chr ""
    .. .. ..@ title     : chr ""
    .. .. ..@ abstract   : chr ""
    .. .. ..@ url       : chr ""
    .. .. ..@ pubMedIds  : chr ""
    .. .. ..@ samples    : list()
    .. .. ..@ hybridizations : list()
    .. .. ..@ normControls : list()
    .. .. ..@ preprocessing : list()
    .. .. ..@ other      : list()
  .. .. ..@ __classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots
    .. .. .. .. ..@ .Data:List of 2
    .. .. .. .. .. ..$ : int [1:3] 1 0 0
    .. .. .. .. .. ..$ : int [1:3] 1 1 0
  ..@ assayData      :<environment: 0xb12e938>
  ..@ phenoData       :Formal class 'AnnotatedDataFrame' [package "Biobase"] with 4 slots
    .. .. ..@ varMetadata : 'data.frame': 4 obs. of 1 variable:
    .. .. .. ..$ labelDescription: chr [1:4] NA NA NA NA
    .. .. ..@ data        : 'data.frame': 137 obs. of 4 variables:
    .. .. .. ..$ Sample.ID : chr [1:137] "34c_2" "34c_3" "34c_4" "34c_5" ...
    .. .. .. ..$ Embryonic.day: chr [1:137] "E3.25" "E3.25" "E3.25" "E3.25" ...
    .. .. .. ..$ Cell.type : chr [1:137] "ICM" "ICM" "ICM" "ICM" ...
    .. .. .. ..$ sampleGroup : chr [1:137] "E3.25" "E3.25" "E3.25" "E3.25" ...
    .. .. ..@ dimLabels    : chr [1:2] "sampleNames" "sampleColumns"
  .. .. ..@ __classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots
    .. .. .. .. ..@ .Data:List of 1
    .. .. .. .. .. ..$ : int [1:3] 1 1 0
  ..@ featureData     :Formal class 'AnnotatedDataFrame' [package "Biobase"] with 4 slots
    .. .. ..@ varMetadata : 'data.frame': 1 obs. of 1 variable:
    .. .. .. ..$ labelDescription: chr NA
```

```

.. .. ..@ data                : 'data.frame': 38 obs. of  1 variable:
.. .. .. ..$ symbol: chr [1:38] "Fgf4" "Tom1l1" "Tdgf1" "Cldn4" ...
.. .. ..@ dimLabels           : chr [1:2] "featureNames" "featureColumns"
.. .. ..@ __classVersion__: Formal class 'Versions' [package "Biobase"] with 1 slots
.. .. .. ..@ .Data: List of 1
.. .. .. .. ..$ : int [1:3] 1 1 0
.. .. ..@ annotation          : chr "single cell qPCR"
.. .. ..@ protocolData        : Formal class 'AnnotatedDataFrame' [package "Biobase"] with 4 slots
.. .. ..@ varMetadata         : 'data.frame': 0 obs. of  1 variable:
.. .. .. ..$ labelDescription: chr(0)
.. .. ..@ data                : 'data.frame': 137 obs. of  0 variables
.. .. ..@ dimLabels           : chr [1:2] "sampleNames" "sampleColumns"
.. .. ..@ __classVersion__: Formal class 'Versions' [package "Biobase"] with 1 slots
.. .. .. ..@ .Data: List of 1
.. .. .. .. ..$ : int [1:3] 1 1 0
.. .. ..@ __classVersion__: Formal class 'Versions' [package "Biobase"] with 1 slots
.. .. .. ..@ .Data: List of 4
.. .. .. .. ..$ : int [1:3] 2 15 0
.. .. .. .. ..$ : int [1:3] 2 16 0
.. .. .. .. ..$ : int [1:3] 1 3 0
.. .. .. .. ..$ : int [1:3] 1 0 0

```

## References

Ohnishi et al., 2014

## See Also

[x](#), [xql](#)

## Examples

```

data(xq)
xq
pData(xq)

```

---

xql

*Position-dependent Gene Expression*


---

## Description

An [ExpressionSet](#) object containing single-cell gene expression measured by qPCR, with cells facing the blastocyst cavity labelled fluorescently.

## Usage

```
data(xql)
```

**Format**

```

Formal class 'ExpressionSet' [package "Biobase"] with 7 slots
  ..@ experimentData :Formal class 'MIAME' [package "Biobase"] with 13 slots
    .. ..@ name      : chr ""
    .. ..@ lab       : chr ""
    .. ..@ contact   : chr ""
    .. ..@ title     : chr ""
    .. ..@ abstract  : chr ""
    .. ..@ url       : chr ""
    .. ..@ pubMedIds : chr ""
    .. ..@ samples   : list()
    .. ..@ hybridizations : list()
    .. ..@ normControls : list()
    .. ..@ preprocessing : list()
    .. ..@ other     : list()
  .. ..@ __classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots
    .. ..@ .Data:List of 2
    .. ..$ : int [1:3] 1 0 0
    .. ..$ : int [1:3] 1 1 0
  ..@ assayData      :<environment: 0x7fe9c10>
  ..@ phenoData      :Formal class 'AnnotatedDataFrame' [package "Biobase"] with 4 slots
    .. ..@ varMetadata : 'data.frame': 2 obs. of 1 variable:
    .. ..$ labelDescription: chr [1:2] NA NA
    .. ..@ data        : 'data.frame': 43 obs. of 2 variables:
    .. ..$ Embryonic.day: chr [1:43] "E4.5" "E4.5" "E4.5" "E4.5" ...
    .. ..$ Label       : chr [1:43] "High" "High" "Low" "Low" ...
    .. ..@ dimLabels   : chr [1:2] "sampleNames" "sampleColumns"
  .. ..@ __classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots
    .. ..@ .Data:List of 1
    .. ..$ : int [1:3] 1 1 0
  ..@ featureData    :Formal class 'AnnotatedDataFrame' [package "Biobase"] with 4 slots
    .. ..@ varMetadata : 'data.frame': 1 obs. of 1 variable:
    .. ..$ labelDescription: chr NA
    .. ..@ data        : 'data.frame': 10 obs. of 1 variable:
    .. ..$ symbol: chr [1:10] "Fgf4" "Cubn" "Sox17" "Lama1" ...
    .. ..@ dimLabels   : chr [1:2] "featureNames" "featureColumns"
  .. ..@ __classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots
    .. ..@ .Data:List of 1
    .. ..$ : int [1:3] 1 1 0
  ..@ annotation     : chr "single cell qPCR"
  ..@ protocolData    :Formal class 'AnnotatedDataFrame' [package "Biobase"] with 4 slots
    .. ..@ varMetadata : 'data.frame': 0 obs. of 1 variable:
    .. ..$ labelDescription: chr(0)
    .. ..@ data        : 'data.frame': 43 obs. of 0 variables
    .. ..@ dimLabels   : chr [1:2] "sampleNames" "sampleColumns"
  .. ..@ __classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots
    .. ..@ .Data:List of 1
    .. ..$ : int [1:3] 1 1 0

```

```
..@ .__classVersion__:Formal class 'Versions' [package "Biobase"] with 1 slots
.. .. ..@ .Data:List of 4
.. .. .. ..$ : int [1:3] 3 0 1
.. .. .. ..$ : int [1:3] 2 20 1
.. .. .. ..$ : int [1:3] 1 3 0
.. .. .. ..$ : int [1:3] 1 0 0
```

## References

Ohnishi et al., 2014

## See Also

[x](#), [xq](#)

## Examples

```
data(xql)
xql
pData(xql)
```

# Index

## \* **datasets**

x, [8](#)

xq, [10](#)

xql, [11](#)

ExpressionSet, [2–6](#), [8](#), [10](#), [11](#)

getDifferentialExpressedGenes, [2](#)

MDSplot, [3](#)

myHeatmap, [4](#), [6](#)

myHeatmap2, [5](#), [5](#)

pam, [6](#)

pamCluster, [6](#)

plotProjection, [7](#)

x, [8](#), [11](#), [13](#)

xq, [9](#), [10](#), [13](#)

xql, [9](#), [11](#), [11](#)