pint

October 5, 2010

`ChromosomeArmModels-class`
Class "ChromosomeArmModels", for dependency models in chromosomal arm

---

Description

Collection of dependency models fitting two data sets in particular chromosome arm.

Objects from the Class

Function `screen.cgh.mrna` and `screen.cgh.mir` returns an object of this class.

Slots

models a list of DependencyModels

chromosome factor of chromosome

arm factor of arm of the chromosome

method a string with name of the method used in dependency models

params a list of parameters of the method

windowSize number of genes in dependecy models windows

Methods

[[`signature(x = "ChromosomeArmModels")`: Returns the a model from the list

[[<- `signature(x = "ChromosomeArmModels")`: Attaches the a model to the list

getModelMethod `signature(model = "ChromosomeArmModels")`: Returns the name of the used method

getParams `signature(model = "ChromosomeArmModels")`: Returns a list of used parameters for the method

getModelNumbers `signature(model = "ChromosomeArmModels")`: Returns the number of the dependency models

getLoc `signature(model = "ChromosomeArmModels")`: Returns a vector of gene locations in the dependency models

getGeneName `signature(model = "ChromosomeArmModels")`: Returns a vector of gene names in the dependency models

getScore `signature(model = "ChromosomeArmModels")`: Returns a vector of dependency scores of the dependency models

getChromosome `signature(model = "ChromosomeArmModels")`: Returns the chromosome

getArm `signature(model = "ChromosomeArmModels")`: Returns the arm of the chromosome

getWindowSize `signature(model = "ChromosomeArmModels")`: Returns the size of the window used in the dependencymodels.

topGenes `signature(model = "ChromosomeArmModels", num = "numeric")`: Returns a vector of given number of names of the genes which have the highest dependency score

topModels `signature(model = "ChromosomeArmModels", num = "numeric")`: Returns a list with given number of dependency models which have the highest dependency score

isEmpty `signature(model = "ChromosomeArmModels")`: Returns TRUE if `model` has no dependency models

orderGenes `signature(model = "ChromosomeArmModels")`: Returns a data frame with gene names and their model scores sorted

findModel `signature(model = "ChromosomeArmModels")`: Finds a dependency model by gene name and returns it.

Author(s)

Olli-Pekka Huovilainen <ohuovila@gmail.com>

See Also

To calculate dependency models for chromosomal arm: `screen.cgh.mrna`. This class holds a number of DependencyModels. To plot dependency scores see `dependency score plotting`. Dependency models for whole chromosomal arm: ChromosomeModels. Dependency models for whole genome: GenomeModels.

Examples

```
data(chromosome17)

## Calculation of dependency models for chromosomal arm
model17p <- screen.cgh.mrna(geneExp, geneCopyNum, windowSize = 10, chr
= 17, arm = 'p')

model17p

## Information of the dependency model which has the highesst dependency score
topGenes(model17p, 1)

## Finding a dependency model by its name
findModel(model17p, "ENSG00000129250")

## Information of the first dependency model
model17p[[1]]

#Plotting
plot(model17p)
```

`ChromosomeModels-class` *Class "ChromosomeModels"*

Description

Collection of dependency models fitting two data sets in particular chromosome. The dependency models are in two ChromosomeArmModels objects which represents q and p arms.

Objects from the Class

Function `screen.cgh.mrna` and `screen.cgh.mir` returns an object of this class.

Slots

pArmModels, qArmModels an ChromosomeArmModels object, dependency models in p or q arm

chromosome a factor of chromosome

method a string with name of the method used in dependency models

params a list of parameters of the used method

Methods

getChromosome `signature(model = "ChromosomeModels")`: Returns the chromosome

getPArm `signature(model = "ChromosomeModels")`: Returns the dependency models of the p arm which is of class ChromosomeArmModels

getQArm `signature(model = "ChromosomeModels")`: Returns the dependency models of the q arm which is of class ChromosomeArmModels

getModelMethod `signature(model = "ChromosomeModels")`: Returns the name of the used method

getParams `signature(model = "ChromosomeModels")`: Returns a list of used parameters for the method

getChr `signature(model = "ChromosomeModels")`: Returns the chromosome

getWindowSize `signature(model = "ChromosomeModels")`: Returns the size of the window used in the dependency models.

topGenes `signature(model = "ChromosomeModels", num = "numeric")`: Returns a vector of given number of names of the genes which have the highest dependency score

topModels `signature(model = "ChromosomeModels", num = "numeric")`: Returns a list with given number of dependency models which have the highest dependency score

isEmpty `signature(model = "ChromosomeModels")`: Returns TRUE if `model` has no dependency models

orderGenes `signature(model = "ChromosomeModels")`: Returns a data frame with gene names and their model scores sorted

findModel `signature(model = "ChromosomeArmModels")`: Finds a dependency model by gene name and returns it.

Author(s)

Olli-Pekka Huovilainen <ohuovila@gmail.com>

See Also

For calculation of dependency models for chromosomal arm: `screen.cgh.mrna`. This class holds a number of DependencyModel in two ChromosomeArmModels objects. For plotting dependency scores see dependency score plotting. Dependency models for whole genome: GenomeModels.

Examples

```
data(chromosome17)

## calculate dependency models over chromosome 17
model17 <- screen.cgh.mrna(geneExp, geneCopyNum, windowSize = 10, chr
= 17)

model17

# genes in p arm with the highest dependency scores
topGenes(getPArm(model17), 5)

plot(model17)
```

DependencyModel-class

Class "DependencyModel"

Description

A Dependency model for one or two data sets

Objects from the Class

Returned by `fit.dependency.model`, `ppca`, `pfa`, `pcca` and `pcca.isotropic` functions.

Slots

W a list of X, Y and total components containing the relationship between two data sets; for dependency model for one dataset, only total is given

phi a list of X, Y and total components containing the data set specific covariances; for dependency model for one dataset, only total is given

score score for fitness of model

loc middle location of the window in base pairs

geneName name of the gene in the middle of the window

windowSize size of the window

method name of the used method

params list of parameters used in dependency model

chromosome Chromosome where the dependency model is calculated

arm Chromosome arm where the dependency model is calculated

Methods

setLoc<- `signature(model = "DependencyModel")`: sets models location

setGeneName<- `signature(model = "DependencyModel")`: sets models gene name

setChromosome<- `signature(model = "DependencyModel")`: sets models chromosome

setArm<- `signature(model = "DependencyModel")`: sets models chromosome arm

getW `signature(model = "DependencyModel")`: Returns a list of model variable `Ws` `X` , `Y` and `total` component

getPhi `signature(model = "DependencyModel")`: Returns a list of model variable `phis` `X` and `Y` and `total` component

getScore `signature(model = "DependencyModel")`: Returns the dependency score of model

getLoc `signature(model = "DependencyModel")`: Returns the middle location of the window

getGeneName `signature(model = "DependencyModel")`: Returns the name of the gene in the middle of window

getChromosome `signature(model = "DependencyModel")`: Returns the chromosome

getArm `signature(model = "DependencyModel")`: Returns the chromosome arm

getParams `signature(model = "DependencyModel")`: Returns a list of used parameters for the method

getModelMethod `signature(model = "DependencyModel")`: Returns the name of the used method

getWindowSize `signature(model = "DependencyModel")`: Returns the size of window

Author(s)

Olli-Pekka Huovilainen `<ohuovila@gmail.com>`

See Also

Calculation of latent variable z: `z.expectation`. For calculation of dependency models for chromosomal arm, chromosome or genome: `screen.cgh.mrna`. Dependency models for whole chromosomal arm: ChromosomeArmModels. Dependency models for whole chromosome: ChromosomeModels. Dependency models for whole genome: GenomeModels. For plotting dependency scores see dependency score plotting.

Examples

```
data(chromosome17)
window <- fixed.window(geneExp, geneCopyNum, 10, 10)
model <- fit.dependency.model(window$X, window$Y)
model

# Contributions of samples and variables to model
plot(model,geneExp,geneCopyNum)
```

`GenomeModels-class` *Class "GenomeModels"*

Description

Collection of dependency models fitting two data sets in whole genome. The dependency models are in a list of ChromosomeModelss (which represents each chromosome) that have two ChromosomeArmModels objects (which represents q and p arms) that have a list of dependency models in that chromosomal arm.

Objects from the Class

Function `screen.cgh.mrna` and `screen.cgh.mir` returns an object of this class.

Slots

chromosomeModels a list of ChromosomeModels of all chromosomes

method a string with name of the method used in dependency model

params a list of parameters of the method

Methods

[[`signature(x = "GenomeModels")`: Returns a ChromosomeModels from the list. X and Y chromosomes can be accessed with `23` and `24` or `'X'` and `'Y'`

[[<- `signature(x = "GenomeModels")`: Attaches a ChromosomeModels to the list. X and Y chromosomes can be accessed with `23` and `24` or `'X'` and `'Y'`

getModelMethod `signature(model = "GenomeModels")`: Returns the name of the used method

getParams `signature(model = "GenomeModels")`: Returns a list of used parameters for the method

getChr `signature(model = "GenomeModels")`: Returns the chromosome

getWindowSize `signature(model = "GenomeModels")`: Returns the size of the window used in the dependency models.

topGenes `signature(model = "GenomeModels", num = "numeric")`: Returns a vector of given number of names of the genes which have the highest dependency score

topModels `signature(model = "GenomeModels", num = "numeric")`: Returns a list with given number of dependency models which have the highest dependency score

orderGenes `signature(model = "GenomeModels")`: Returns a data frame with gene names and their model scores sorted

findModel `signature(model = "ChromosomeArmModels")`: Finds a dependency model by gene name and returns it.

Author(s)

Olli-Pekka Huovilainen

See Also

For calculation of dependency models for chromosomal arm: `screen.cgh.mrna`. This class holds a number of DependencyModel in two ChromosomeModels objects in each ChromosomeArmModels. For plotting dependency scores see `dependency score plotting`.

`fit.byname` *Fit dependency model around one gene between two data sets.*

Description

Takes a window from two datasets around chosen gene and fits a selected dependency model between windows.

Usage

```
fit.cgh.mir.byname(X, Y, geneName, windowSize, ...)

fit.cgh.mrna.byname(X, Y, geneName, windowSize, ...)
```

Arguments

`X,Y`	Data sets. Lists containing the following items: `data` Data in a matrix form. Genes are in columns and samples in rows. e.g. gene copy number. `info` Data frame which contains following information about genes in data matrix. `chr` Factor indicating the chrosome for the gene: (1 to 23, or X or Y `arm` Factor indicating the chromosomal arm for the gene ('p' or 'q') `loc` Location of the gene in base pairs. `pint.data` can be used to create data sets in this format.
`geneName`	The dependency model is calculated around this gene.
`windowSize`	Size of the data window.
`...`	Arguments to be passed to function `fit.dependency.model`

Details

See `fit.dependency.model` for details about dependency models and parameters.

Value

DependencyModel

Author(s)

Olli-Pekka Huovilainen `<ohuovila@gmail.com>` and Leo Lahti `<leo.lahti@iki.fi>`

References

Dependency Detection with Similarity Constraints, Lahti et al., 2009 Proc. MLSP'09 IEEE International Workshop on Machine Learning for Signal Processing, http://www.cis.hut.fi/lmlahti/publications/mlsp09_preprint.pdf

A Probabilistic Interpretation of Canonical Correlation Analysis, Bach Francis R. and Jordan Michael I. 2005 Technical Report 688. Department of Statistics, University of California, Berkley. http://www.di.ens.fr/~fbach/probacca.pdf

Probabilistic Principal Component Analysis, Tipping Michael E. and Bishop Christopher M. 1999. *Journal of the Royal Statistical Society*, Series B, **61**, Part 3, pp. 611–622. http://research.microsoft.com/en-us/um/people/cmbishop/downloads/Bishop-PPCA-JRSS.pdf

EM Algorithms for ML Factorial Analysis, Rubin D. and Thayer D. 1982. *Psychometrika*, **vol. 47**, no. 1.

See Also

Reults from this function: DependencyModel. `fit.dependency.model`. Calculating dependency models to chromosomal arm, chromosome or genome `screen.cgh.mrna`. For calculation of latent variable z: `link{z.expectation}`.

Examples

```
data(chromosome17)

model <- fit.cgh.mrna.byname(geneExp,geneCopyNum,"ENSG00000132361",10)
## With different model parameters (pCCA)
model2 <- fit.cgh.mrna.byname(geneExp,geneCopyNum,"ENSG00000132361",10,zDimension=5,H=NA)
```

`fit.dependency.model`

Fit dependency model between two data sets.

Description

Fits a selected dependency model between two data sets. The function can fit probabilistic canonical correlation analysis (pCCA; *Bach & Jordan 2005*), probabilistic principal component (pPCA; *Tipping & Bishop 1999*) analysis, probabilistic factorial analysis (pFA; *Rubin & Thayer 1982*) or similarity constrained canonical correlation analysis (pSimCCA; *Lahti et al. 2009*). These correspond to `ppca`, `pcca`, `pcca.isotropic` and `pfa` as well as different choices of the model structure and parameters in `fit.dependency.model`.

Usage

```
fit.dependency.model(X, Y, zDimension = 1, marginalCovariances = "full",
H = 1, sigmas = 0, covLimit = 0, mySeed = 123)

ppca(X, Y, zDimension = 1)
pcca(X, Y, zDimension = 1)
pcca.isotropic(X, Y, zDimension = 1, covLimit = 1e-6)
pfa(X, Y = NULL, zDimension = 1)
```

Arguments

X, Y	The data sets. 'Variables x samples'. If NULL is given, model is calculated for only one data set.
zDimension	Dimensionality of the shared latent variable.
marginalCovariances	Type of marginal covariances. Options: "identical isotropic", "isotropic", "diagonal" and "full"
H	Mean of the matrix normal prior distribution for the transformation matrix T. Must be a matrix of size (variables in first data set) x (variables in second data set). If value is 1, H will be made identity matrix of appropriate size.
sigmas	Variance parameter for the matrix normal prior distribution of the transformation matrix T. Described the allowed deviation scale of the transformation matrix T from the mean matrix H.
covLimit	Convergence limit. default value depends on chosen model type.
mySeed	Random seed

Details

The dependency models considered in *Lahti et al. 2009* are obtained as follows:

pPCA `H = NA, marginalCovariances = "identical isotropic"` (*Tipping & Bishop 1999*)

pFA `H = NA, marginalCovariances = "diagonal"` (*Rubin & Thayer 1982*)

pCCA `H = NA, marginalCovariances = "full"` or `"isotropic"` (*Bach & Jordan 2005*)

pSimCCA `H = I, sigmas = 0, marginaCovariances = "full"`. This is the default method. (*Lahti et al. 2009*)

pSimCCA with T prior `H = I, marginalCovariances = "isotropic"` (*Lahti et al. 2009*)

Resulting DependencyModel object does not have location or z variable. Location can be set with `setLoc` method (see examples) and expectation of the latent variable z can be calculated with `link{z.expectation}`.

To avoid computational singularities, the covariance matrix phi is regularised by adding a small constant to the diagonal

Value

DependencyModel

Author(s)

Olli-Pekka Huovilainen <ohuovila@gmail.com> and Leo Lahti <leo.lahti@iki.fi>

References

Dependency Detection with Similarity Constraints, Lahti et al., 2009 Proc. MLSP'09 IEEE International Workshop on Machine Learning for Signal Processing, http://www.cis.hut.fi/lmlahti/publications/mlsp09_preprint.pdf

A Probabilistic Interpretation of Canonical Correlation Analysis, Bach Francis R. and Jordan Michael I. 2005 Technical Report 688. Department of Statistics, University of California, Berkley. http://www.di.ens.fr/~fbach/probacca.pdf

Probabilistic Principal Component Analysis, Tipping Michael E. and Bishop Christopher M. 1999. *Journal of the Royal Statistical Society*, Series B, **61**, Part 3, pp. 611–622. http://research.microsoft.com/en-us/um/people/cmbishop/downloads/Bishop-PPCA-JRSS.pdf

EM Algorithms for ML Factorial Analysis, Rubin D. and Thayer D. 1982. *Psychometrika*, **vol. 47**, no. 1.

See Also

For windowing data: `fixed.window`. Reults from this function: DependencyModel. Calculating dependency models to chromosomal arm, chromosome or genome `screen.cgh.mrna`. For calculation of latent variable z: `link{z.expectation}`.

Examples

```
data(chromosome17)

# pSimCCA
window <- fixed.window(geneExp, geneCopyNum, 10, 10)

model <- fit.dependency.model(window$X, window$Y, zDimension = 1)
setLoc(model) <- window$loc
model

# Contributions of samples and variables to model
plot(model, geneExp, geneCopyNum)
```

---

geneCopyNum *Gene copy number data in chromosome 17*

Description

Preprocessed gene copy number (aCGH) data for 51 patients in chromosome 17.

Usage

```
data(chromosome17)
```

Format

A list which contain the following data:

data gene copy number data in matrix form. Genes are in columns and samples in rows

info Data frame which contains following information about genes in data matrix.

chr Factor indicating the chrosome for the gene (1 to 23, or X or Y

arm Factor indicating the chromosomal arm for the gene ('p' or 'q')

loc Location of the gene in base pairs.

Source

Integrated gene copy number and expression microarray analysis of gastric cancer highlights potential target genes. Myllykangas et al., *International Journal of Cancer*, vol. **123**, **no. 4**, pp. 817–25, 2008.

---

geneExp *Gene expression data in chromosome 17*

Description

Preprocessed gene expression levels of 51 patients in chromosome 17.

Usage

```
data(chromosome17)
```

Format

A list which contain the following data:

data gene expression data in matrix form. Genes are in columns and samples in rows

info Data frame which contains following information about genes in data matrix.

chr Factor of chrosome where the gene is. (1 to 23 or X or Y

arm Factor of arm of the chromosome arm where the gene is. ('p' or 'q')

loc Location of the gene from centromere in base pairs.

Source

Integrated gene copy number and expression microarray analysis of gastric cancer highlights potential target genes. Myllykangas et al., *International Journal of Cancer*, vol. **123**, **no. 4**, pp. 817–25, 2008.

pint.data *Forms a data set and pairs samples in two data sets.*

Description

Forms a data set for use in functions in 'pint' package (e.g. screen.cgh.mrna). Pairs samples in two data sets.

Usage

```
pint.data(data, info)
pint.match(X, Y, max.dist)
```

Arguments

data	Probe-level data in a matrix or data frame.
info	Location, chromosome, and chromosome arm. Information of the probes as data frame. Location can be given either as loc or bp, which is middle location of probe, or as start and end. Chromosome arm is given as arm and chromosome as chr.
X, Y	Data sets to be paired.
max.dist	maximum distance between paired genes in base pairs.

Details

Function pint.match goes through every sample in X and finds the nearest sample in Y which is in the same chromosome arm. If more than one sample in X has same nearest sample in Y, all but one is discarded. Samples with longer distance than max.dist are discarded.

Value

pint.data returns a list with a matrix with sample data and a data frame with chr (chromosome), arm (chromosome arm) and loc (location).

pint.match return a list with two data sets. These can be used in screen.cgh.mrna function.

Author(s)

Olli-Pekka Huovilainen <ohuovila@gmail.com>

See Also

screen.cgh.mrna, screen.cgh.mir, fit.cgh.mir.byname

Examples

```
data(chromosome17)

newData <- pint.match(geneExp,geneCopyNum,max.dist=1000)
```

plot *Dependency score plotting*

Description

Plot the contribution of the samples and variables to the dependency model or dependency model fitting scores of chromosomal arm, chromosome or genome.

Usage

```
plot.DependencyModel(x, X, Y, ann.types = NULL, ann.cols = NULL, legend.x = 0,
        legend.y = 1, legend.xjust = 0, legend.yjust = 1, order = FALSE,
        cex.z = 0.6, cex.WX = 0.6, cex.WY = 0.6, ...)

plot.ChromosomeArmModels(x, hilightGenes = NULL, showDensity = FALSE, showTop =
topName = FALSE, type = 'l', xlab = 'gene location (Mbp)',
ylab = 'dependency score', main = paste('Dependency score for chromosome ',
chr, arm, sep = ''), pch = 20, cex = 0.75, tpch = 3, tcex = 1, ylim = NA, ...)

plot.ChromosomeModels(x, hilightGenes = NULL, showDensity = FALSE, showTop = 0,
topName = FALSE, type = 'l', xlab = 'gene location (Mbp)', ylab = 'dependency sc
main = paste('Dependency score for chromosome ', chr, sep = ''),
pch = 20, cex = 0.75, tpch = 3, tcex = 1, xlim = NA, ylim = NA,...)

plot.GenomeModels(x, hilightGenes = NULL, showDensity = FALSE, showTop = 0,
topName = FALSE, onePlot = FALSE, type = 'l', ylab = "Dependency Scores",
xlab = "Gene location (chromosome)", main = "Dependency Scores in All Chromosome
pch = 20, cex = 0.75, tpch = 20, tcex = 0.7, mfrow = c(5,5), mar = c(3,2.5,1.3,0
ps = 5, mgp = c(1.5,0.5,0),ylim=NA,...)
```

Arguments

x	DependencyModel-class, ChromosomeArmModels-class, ChromosomeModels-class, GenomeModels-class; models to be plotted.
X, Y	data sets used in dependency modeling.
ann.types	a factor for annotation types for samples. Each value corresponds one sample in datasets. Colors are used to indicate different types.
ann.cols	colors used to indicate different annotation types. Gray scale is used if 'NULL' given.
legend.x, legend.y	the x and y co-ordinates to be used to position the legend for annotation types.
legend.xjust, legend.yjust	how the legend is to be justified relative to the legend x and y location. A value of 0 means left or top justified, 0.5 means centered and 1 means right or bottom justified.
order	logical; if 'TRUE', values for sample contributions are ordered according to their values.
cex.z, cex.WX, cex.WY	Text size for variable names.

hilightGenes	vector of strings; Name of genes to be hilighted with dots.
showDensity	logical; if 'TRUE' small vertical lines are drwan in the bottom of the plot under each gene.
showTop	numeric; Number of models with highest dependencies to be hilighted. A horizontal dashed line is drawn to show threshold value. With 0 no line is drawn.
topName	logical; If TRUE, gene names are printed to hilighted models with highest dependecies. Otherwise hilighted models are numbered according to their rank in dependency score.
type, xlab, ylab, main	plot type and labels. See plot for details. In plot.GenomeModels these affet only if onePlot is TRUE.
onePlot	If TRUE, all dependency scores are plotted in one plot window. Otherwise one plot window is used for each chromosome.
pch, cex	symbol type and size for hilightGenes. See points for details.
tpch, tcex	symbol type and size for genes with highest scores. See points for details.
ylim, xlim	axis limits. Default values are calculated from data. Lower limit for y is 0 and upper limit is either 1 or maximum score value. X limits are gene location range. See plot for details.
mfrow, mar, ps, mgp	chromosome plots' layout, marginals, text size and margin line. See par for details.
...	optional plotting parameters

Details

Function plots scores of each dependency model of a gene for the whole chromosomal arm, chromosome or genome according to used method. `plot(x, cancerGenes = NULL, showDensity = FALSE, ...)` is also usable and chosen according to class of `models`.

Author(s)

Olli-Pekka Huovilainen <ohuovila@gmail.com>

References

Dependency Detection with Similarity Constraints Lahti et al., MLSP'09. See http://www.cis.hut.fi/lmlahti/publications/mlsp09_preprint.pdf

See Also

DependencyModel-class, ChromosomeArmModels-class, ChromosomeModels-class, GenomeModels-class, screen.cgh.mrna, screen.cgh.mir

Examples

```
data(chromosome17)

## Dependency model around 150th gene
window <- fixed.window(geneExp, geneCopyNum, 150, 10)
model <- fit.dependency.model(window$X, window$Y)
```

```
## example annnotation types
ann.types <- factor(c(rep("Samples 1 - 10", 10), rep("Samples 11 - 51", 41)))
plot(model, geneExp, geneCopyNum, ann.types, legend.x = 40, legend.y = -4,
     order = TRUE)

## pSimCCA model on chromosome 17p
models17ppSimCCA <- screen.cgh.mrna(geneExp, geneCopyNum, 10, 17, 'p')
plot(models17ppSimCCA,
     hilightGenes=c("ENSG00000108342", "ENSG00000108298"), showDensity = TRUE)
```

`calculate` *Fits dependency models to chromosomal arm, chromosome or the whole genome.*

Description

Fits dependency models for whole chromosomal arm, chromosome or genome depending on arguments with chosen window size between two data sets.

Usage

```
screen.cgh.mrna(X, Y, windowSize, chromosome, arm, method = "", params = list())
screen.cgh.mir(X, Y, windowSize, chromosome, arm, method = "", params = list())
```

Arguments

`X,Y`	Data sets. Lists containing the following items: `data` Data in a matrix form. Genes are in columns and samples in rows. e.g. gene copy number. `info` Data frame which contains following information about genes in data matrix. `chr` Factor indicating the chrosome for the gene: (1 to 23, or X or Y `arm` Factor indicating the chromosomal arm for the gene ('p' or 'q') `loc` Location of the gene in base pairs. `pint.data` can be used to create data sets in this format.
`chromosome`	Specify the chromosome for model fitting. If missing, whole genome is screened.
`arm`	Specify chromosomal arm for model fitting. If missing, both arms are modeled.
`windowSize`	Determine the window size. This specifies the number of nearest genes to be included in the chromosomal window of the model, and therefore the scale of the investigated chromosomal region.
`method`	Specify the dependency model: **'pCCA'** probabilistic canonical correlation analysis *Bach & Jordan 2005* **'pPCA'** probabilistic principal component analysis *Tipping & Bishop 1999* **'pFA'** probabilistic factor analysis *Rubin & Thayer 1982* **'pSimCCA'** probabilistic similarity constrained canonical correlation analysis *Lahti et al. 2009*

'TPriorpSimCCA' probabilistic similarity constarined canonical correlation analysis with possibility to tune T prior (Lahti et al. 2009)

If anything else, the model is specified by the given parameters.

`params` List of parameters for the dependency model.

sigmas Variance parameter for the matrix normal prior distribution of the transformation matrix T. This describes the deviation of T from H

H Mean parameter for the matrix normal prior distribution prior of transformation matrix T

zDimension Dimensionality of the latent variable

mySeed Random seed.

covLimit Convergence limit. Default depends on the selected method: 1e-3 for pSimCCA with full marginal covariances and 1e-6 for pSimCCA in other cases.

Details

Function `screen.cgh.mrna` assumes that data is already paired. This can be done with `pint.match`. It takes sliding gene windows with `fixed.window` and fits dependency models to each window with `fit.dependency.model` function.

Function `screen.cgh.mir` calculates dependencies around a chromosomal window in each sample in `X`; only one sample from `X` will be used. Data sets do not have to be of the same size and`X` can be considerably smaller. This is used with e.g. miRNA data.

If method name is specified, this overrides the corresponding model parameters, corresponding to the modeling assumptions of the specified model. Otherwise method for dependency models is determined by parameters.

Dependency scores are plotted with dependency score plotting.

Value

Depending on the arguments, returns a ChromosomeArmModels which contains all the models for dependencies in chromosomal arm, a ChromosomeModels which contains all the models for dependencies in chromosome or a GenomeModels which contains all the models for dependencies in genome.

Author(s)

Olli-Pekka Huovilainen `<ohuovila@gmail.com>` and Leo Lahti `<leo.lahti@iki.fi>`

References

Dependency Detection with Similarity Constraints, Lahti et al., 2009 Proc. MLSP'09 IEEE International Workshop on Machine Learning for Signal Processing, See http://www.cis.hut.fi/lmlahti/publications/mlsp09_preprint.pdf

A Probabilistic Interpretation of Canonical Correlation Analysis, Bach Francis R. and Jordan Michael I. 2005 Technical Report 688. Department of Statistics, University of California, Berkley. http://www.di.ens.fr/~fbach/probacca.pdf

Probabilistic Principal Component Analysis, Tipping Michael E. and Bishop Christopher M. 1999. *Journal of the Royal Statistical Society*, Series B, **61**, Part 3, pp. 611–622. http://research.microsoft.com/en-us/um/people/cmbishop/downloads/Bishop-PPCA-JRSS.pdf

EM Algorithms for ML Factoral Analysis, Rubin D. and Thayer D. 1982. *Psychometrika*, **vol. 47**, no. 1.

See Also

To fit a dependency model: `fit.dependency.model`. ChromosomeArmModels holds dependency models for chromosomal arm, ChromosomeModels holds dependency models for chromosome, GenomeModels holds dependency models for genome. For plotting, see: dependency score plotting

Examples

```
data(chromosome17)

## pSimCCA model on chromosome 17

models17pSimCCA <- screen.cgh.mrna(geneExp, geneCopyNum,
                                   windowSize = 10, chr = 17)

plot(models17pSimCCA)

## pCCA model on chromosome 17q with 3-dimensional latent variable z
models17ppCCA <- screen.cgh.mrna(geneExp, geneCopyNum,
                                 windowSize = 10,
                                 chromosome = 17, arm = 'p',method="pCCA",
                    params = list(zDimension = 3))
plot(models17ppCCA)

## pFA on chromosome 17p. method is determined by the parameters
models17ppFA <- screen.cgh.mrna(geneExp, geneCopyNum,
                                windowSize = 10,
                                chromosome = 17, arm = 'p',
                                params = list(marginalCovariances = "diagonal", H = NA))
plot(models17ppFA)
```

---

window *Form data with a selected window size for the model fitting*

Description

Forms a chosen window of two data matrices to use for `fit.dependency.model` either iteratively picking nearest genes or picking same number of genes from both directions. `sparse.window` forms a window around one sample in the first data set with a number of samples from the second data set.

Usage

```
fixed.window(X, Y, middleIndex, windowSize)
iterative.window(X, Y, middleIndex, windowSize)
sparse.window(X, Y, xIndex, windowSize)
```

Arguments

X	First data set. In sparse.window windows will be formed around each sample in this data set.
Y	Second data set.
middleIndex	Index of middle position for window.
xIndex	Index of middle position in X for window.
windowSize	Number of genes in window. In sparse.window X has always one sample in window.

Details

Window contains windowSize nearest genes. Warning is given if windowSize genes is not found in the same chromosomal arm. Data of both data sets is normalised so that each genes data has zero mean.

Value

List of window data:

X	window of the first data set
Y	window of the second data set
loc	location of gene
geneName	name of the gene
edge	logical; TRUE if iteration to one direction has stopped because edge of data in chromosomal arm has been found.
fail	logical; TRUE if chromosomal arm contains less than windowSize genes.

Author(s)

Olli-Pekka Huovilainen <ohuovila@gmail.com>

See Also

Dependency model fitting: fit.dependency.model

Examples

```
data(chromosome17)
window <- iterative.window(geneExp, geneCopyNum, 30, 10)
model <- fit.dependency.model(window$X, window$Y)
setGeneName(model) <- window$geneName
setLoc(model) <- window$loc
model

window <- fixed.window(geneExp, geneCopyNum, 10, 10)
model <- fit.dependency.model(window$X, window$Y, H = NA)
model
```

z.expectation *The model parameters z and W*

Description

Expectation of the latent variable z, contribution of each sample to a dependency model, and contribution of each variable.

Usage

```
z.expectation(model, X, Y = NULL)
z.effects(model, X, Y = NULL)
W.effects(model, X, Y = NULL)
```

Arguments

model	The fitted dependency model.
X, Y	Data sets used in fitting the dependency modeling functions (screen.cgh.mrna or link{fit.dependency.model}). Note: Arguments must be given in the same order as in fit.dependency.model or screen.cgh.mrna. Only X is needed for dependency model for one data set.

Details

z.expectation gives ML estimate of the shared latent variable Z, given data X, Y and the model parameters in model.

z.effects gives the contribution of each sample to the dependency score. This is approximated by projecting original data to first principal component of Wz.

W.effects gives the contribution of each variable to the observed dependency. This is approximated with the loadings of the first principal component of Wz

Original data can be retrieved by locating the row in X (or Y) which has the same variable (gene) name than model.

Value

z.expectation gives the matrix z. z.effects gives a projection vector over the samples and W.effects gives a projection vector over the variables.

Author(s)

Olli-Pekka Huovilainen <ohuovila@gmail.com> and Leo Lahti <leo.lahti@iki.fi>

References

Dependency Detection with Similarity Constraints, Lahti et al., 2009 Proc. MLSP'09 IEEE International Workshop on Machine Learning for Signal Processing, See http://www.cis.hut.fi/lmlahti/publications/mlsp09_preprint.pdf

A Probabilistic Interpretation of Canonical Correlation Analysis, Bach Francis R. and Jordan Michael I. 2005 Technical Report 688. Department of Statistics, University of California, Berkley. http://www.di.ens.fr/~fbach/probacca.pdf

Probabilistic Principal Component Analysis, Tipping Michael E. and Bishop Christopher M. 1999. *Journal of the Royal Statistical Society*, Series B, **61**, Part 3, pp. 611–622. http://research.microsoft.com/en-us/um/people/cmbishop/downloads/Bishop-PPCA-JRSS.pdf

See Also

DependencyModel-class, screen.cgh.mrna

Examples

```
data(chromosome17)
window <- fixed.window(geneExp, geneCopyNum, 150, 10)

## pSimCCA model around one gene
depmodel2 <- fit.dependency.model(window$X, window$Y)
barplot(z.effects(depmodel2, geneExp, geneCopyNum))

## Plot the contribution of each genes to the model. Only the X component is plotted
## here since Wx = Wy (in SimCCA)
barplot(W.effects(depmodel2, geneExp, geneCopyNum)$X)

## Fit pCCA around one gene
depmodel <- fit.dependency.model(window$X, window$Y, zDimension = 1,
                         marginalCovariances = "full", H = NA)

## Retrieve the shared latent variable Z between data sets
z.expectation(depmodel, geneExp, geneCopyNum)

## plot.DpenendencyModel shows also sample and variable effects
plot(depmodel,geneExp,geneCopyNum)
```

Index

∗Topic **classes**
ChromosomeArmModels-class, 1
ChromosomeModels-class, 3
DependencyModel-class, 4
GenomeModels-class, 6
∗Topic **datasets**
geneCopyNum, 10
geneExp, 11
∗Topic **hplot**
plot, 13
∗Topic **iteration**
calculate, 15
fit.byname, 7
fit.dependency.model, 8
∗Topic **math**
calculate, 15
fit.byname, 7
fit.dependency.model, 8
z.expectation, 19
[[*(ChromosomeArmModels-class)*, 1
[[,ChromosomeArmModels-method *(ChromosomeArmModels-class)*, 1
[[,GenomeModels-method *(GenomeModels-class)*, 6
[[<- *(ChromosomeArmModels-class)*, 1
[[<-,ChromosomeArmModels-method *(ChromosomeArmModels-class)*, 1
[[<-,GenomeModels-method *(GenomeModels-class)*, 6

calculate, 15
ChromosomeArmModels, *3–7*, *16, 17*
ChromosomeArmModels-class, *13, 14*
ChromosomeArmModels-class, 1
ChromosomeModels, *2*, *5–7*, *16, 17*
ChromosomeModels-class, *13, 14*
ChromosomeModels-class, 3

dependency score plotting, *2*, *4, 5*, *7*, *16, 17*
dependency score plotting *(plot)*, 13
DependencyModel, *1, 2*, *4*, *7–10*
DependencyModel-class, *13, 14*, *20*
DependencyModel-class, 4

findModel *(ChromosomeArmModels-class)*, 1
findModel,ChromosomeArmModels-method *(ChromosomeArmModels-class)*, 1
findModel,ChromosomeModels-method *(ChromosomeModels-class)*, 3
findModel,GenomeModels-method *(GenomeModels-class)*, 6
fit.byname, 7
fit.cgh.mir.byname, *12*
fit.cgh.mir.byname *(fit.byname)*, 7
fit.cgh.mrna.byname *(fit.byname)*, 7
fit.dependency.model, *4*, *7*, 8, *8*, *16–19*
fixed.window, *10*, *16*
fixed.window *(window)*, 17

geneCopyNum, 10
geneExp, 11
GenomeModels, *2*, *4, 5*, *16, 17*
GenomeModels-class, *13, 14*
GenomeModels-class, 6
getArm *(ChromosomeArmModels-class)*, 1
getArm,ChromosomeArmModels-method *(ChromosomeArmModels-class)*, 1
getArm,DependencyModel-method *(DependencyModel-class)*, 4
getChromosome *(ChromosomeArmModels-class)*, 1
getChromosome,ChromosomeArmModels-method *(ChromosomeArmModels-class)*,

1
getChromosome,ChromosomeModels-method (*ChromosomeModels-class*), 3
getChromosome,DependencyModel-method (*DependencyModel-class*), 4
getGeneName (*DependencyModel-class*), 4
getGeneName,ChromosomeArmModels-method (*ChromosomeArmModels-class*), 1
getGeneName,DependencyModel-method (*DependencyModel-class*), 4
getLoc (*DependencyModel-class*), 4
getLoc,ChromosomeArmModels-method (*ChromosomeArmModels-class*), 1
getLoc,DependencyModel-method (*DependencyModel-class*), 4
getModelMethod (*ChromosomeArmModels-class*), 1
getModelMethod,ChromosomeArmModels-method (*ChromosomeArmModels-class*), 1
getModelMethod,ChromosomeModels-method (*ChromosomeModels-class*), 3
getModelMethod,DependencyModel-method (*DependencyModel-class*), 4
getModelMethod,GenomeModels-method (*GenomeModels-class*), 6
getModelNumbers (*ChromosomeArmModels-class*), 1
getModelNumbers,ChromosomeArmModels-method (*ChromosomeArmModels-class*), 1
getParams (*ChromosomeArmModels-class*), 1
getParams,ChromosomeArmModels-method (*ChromosomeArmModels-class*), 1
getParams,ChromosomeModels-method (*ChromosomeModels-class*), 3
getParams,DependencyModel-method (*DependencyModel-class*), 4
getParams,GenomeModels-method (*GenomeModels-class*), 6
getPArm (*ChromosomeModels-class*), 3
getPArm,ChromosomeModels-method (*ChromosomeModels-class*), 3
getPhi (*DependencyModel-class*), 4
getPhi,DependencyModel-method (*DependencyModel-class*), 4
getQArm (*ChromosomeModels-class*), 3
getQArm,ChromosomeModels-method (*ChromosomeModels-class*), 3
getScore (*DependencyModel-class*), 4
getScore,ChromosomeArmModels-method (*ChromosomeArmModels-class*), 1
getScore,DependencyModel-method (*DependencyModel-class*), 4
getW (*DependencyModel-class*), 4
getW,DependencyModel-method (*DependencyModel-class*), 4
getWindowSize (*ChromosomeArmModels-class*), 1
getWindowSize,ChromosomeArmModels-method (*ChromosomeArmModels-class*), 1
getWindowSize,ChromosomeModels-method (*ChromosomeModels-class*), 3
getWindowSize,DependencyModel-method (*DependencyModel-class*), 4
getWindowSize,GenomeModels-method (*GenomeModels-class*), 6

isEmpty (*ChromosomeArmModels-class*), 1
isEmpty,ChromosomeArmModels-method (*ChromosomeArmModels-class*), 1
isEmpty,ChromosomeModels-method (*ChromosomeModels-class*), 3
iterative.window (*window*), 17

orderGenes (*ChromosomeArmModels-class*), 1
orderGenes,ChromosomeArmModels-method (*ChromosomeArmModels-class*), 1
orderGenes,ChromosomeModels-method (*ChromosomeModels-class*), 3
orderGenes,GenomeModels-method (*GenomeModels-class*), 6

par, *14*
pcca, *4*

pcca (fit.dependency.model), 8
pcca.isotropic, 4
pfa, 4
pfa (fit.dependency.model), 8
pint.data, 7, 12, 15
pint.match, 16
pint.match (pint.data), 12
plot, 13, 14
plot.ChromosomeArmModels (plot), 13
plot.ChromosomeModels (plot), 13
plot.DependencyModel (plot), 13
plot.GenomeModels (plot), 13
points, 14
ppca, 4
ppca (fit.dependency.model), 8

screen.cgh.mir, 1, 3, 6, 12, 14
screen.cgh.mir (calculate), 15
screen.cgh.mrna, 1–8, 10, 12, 14, 19, 20
screen.cgh.mrna (calculate), 15
setArm<- (DependencyModel-class), 4
setArm<-,DependencyModel-method (DependencyModel-class), 4
setChromosome<- (DependencyModel-class), 4
setChromosome<-,DependencyModel-method (DependencyModel-class), 4
setGeneName<- (DependencyModel-class), 4
setGeneName<-,DependencyModel-method (DependencyModel-class), 4
setLoc<- (DependencyModel-class), 4
setLoc<-,DependencyModel-method (DependencyModel-class), 4
sparse.window (window), 17

topGenes (ChromosomeArmModels-class), 1
topGenes,ChromosomeArmModels-method (ChromosomeArmModels-class), 1
topGenes,ChromosomeModels-method (ChromosomeModels-class), 3
topGenes,GenomeModels-method (GenomeModels-class), 6
topModels (ChromosomeArmModels-class), 1
topModels,ChromosomeArmModels-method (ChromosomeArmModels-class), 1
topModels,ChromosomeModels-method (ChromosomeModels-class), 3
topModels,GenomeModels-method (GenomeModels-class), 6

W.effects (z.expectation), 19
window, 17

z.effects (z.expectation), 19
z.expectation, 5, 19