Package 'yeastExpData'

November 10, 2026

Title Yeast Experimental Data

Version 0.58.0

Author R. Gentleman

Description A collection of different sets of experimental data from yeast.

biocViews ExperimentData, Genome, Saccharomyces_cerevisiae_Data

Maintainer R. Gentleman <rgentlem@fhcrc.org>

License GPL

Depends R (>= 2.4), graph (>= 1.9.26)

Suggests Biobase, GO.db, RBGL, org.Sc.sgd.db

git_url https://git.bioconductor.org/packages/yeastExpData

git_branch RELEASE_3_23

git_last_commit 267316f

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-10

Contents

ccyclered 2
fcabundance 3
gfp 4
litG 6
nPdist 7
proteinProperties 8

Index **10**

`ccyclered` *~~ data name/kind ... ~~*

Description

The data are 2885 yeast genes, common to a number of different experiments. The original data were reported by Cho et al. and were processed to

Usage

```
data(ccyclered)
```

Format

A data frame with 2885 observations on the following 11 variables.

Cluster The cluster number the gene was assigned to.

Distance The distance from the cluster center?

Y.name The name of the gene, using standard yeast nomenclature.

SGDID The Stanford yeast genome database identifier for the gene.

GENE The common name for the gene.

Chromsome The chromosome the gene is located on.

Start The start of the gene, in bases, most likely from the 3' end.

End The end of the gene.

Introns The number of introns.

Exons The location of the exons.

Description A description of the gene.

Details

Cho, et al. discuss the k means clustering of 2885 Saccharomyces genes into 30 clusters with measurements taken over two synchronized cell cycles.

Source

http://arep.med.harvard.edu/network_discovery

References

Cho, et al. (1998) A genome-wide transcriptional analysis of the mitotic cell cycle. Molecular Cell, 2, 65-73.

Tavazoie, et al. (1999) Systematic determination of genetic network architecture. Nature Genetics, 22, 281-285.

Examples

```
data(ccyclered)
```

fcabundance *Yeast protein abundance*

Description

Abundance of yeast proteins measured using flow cytometry and GFP tagging

Usage

```
data(fcabundance)
```

Format

A data frame with 4159 observations on the following 6 variables.

`yORF` a factor denoting yeast Open Reading Frames with levels `YAL001C`, `YAL002W`, `YAL005C`, etc

`gene_name` a factor denoting corresponding gene names, with levels `37164`, `AAC1`, `AAC3`, etc

`YEPD.mean` a numeric vector, giving average abundance in rich (YEPD) media

`YEPD.error` a numeric vector, giving corresponding (standard?) error

`SD.mean` a numeric vector, giving average abundance in minimal (SD) media

`SD.error` a numeric vector, giving corresponding error

Source

http://www.nature.com/nature/journal/v441/n7095/extref/nature04785-s03.xls

References

"Single-cell proteomic analysis of S. cerevisiae reveals the architecture of biological noise" (15 June 2006). John R. S. Newman, Sina Ghaemmaghami, Jan Ihmels, David K. Breslow, Matthew Noble, Joseph L. DeRisi and Jonathan S. Weissman Nature 441, 840-846

Examples

```
data(fcabundance)
plot(YEPD.mean ~ SD.mean, fcabundance, log = "xy")
plot(SD.error ~ SD.mean, fcabundance, log = "x")
```

gfp *Yeast GFP Fusion Data*

Description

This data frame contains data concerning the localization and abundance of various yeast proteins.

Usage

```
data(gfp)
```

Format

A data frame with 6234 observations on the following 33 variables.

`orfid` a numeric vector of identifiers

`yORF` a factor representing yeast ORF names, with levels `YAL001C`, `YAL002W`, etc. These are also the row names of the data frame.

`gene_name` a factor representing corresponding yeast gene names, with levels `AAC1`, `AAC3`, etc.

`GFP_tagged` a factor with levels `not tagged` and `tagged`, indicating whether or not the ORF was GFP tagged

`GFP_visualized` a factor with levels `not visualized` and `visualized`, indicating whether or not GFP fluoresence was visualized

`TAP_visualized` a factor with levels `TAP visualized` and `not TAP visualized`, indicating success of TAP tag

`abundance` a numeric vector, giving estimated abundance in units of molecules per cell

`error` a numeric vector of estimated errors in abundance for a subset of proteins, in the same units as `abundance` (see details below)

`localization_summary` a factor with levels , `ER`, `ER to Golgi`, `ER,ambiguous`, `ER,ambiguous,bud`, etc. Summarizes the information contained in the subsequent columns.

The following columns indicate whether or not the protein was localized in the specific region of the cell. A protein can be localized in more than one region.

`ambiguous` a logical vector

`mitochondrion` a logical vector

`vacuole` a logical vector

`spindle_pole` a logical vector

`cell_periphery` a logical vector

`punctate_composite` a logical vector

`vacuolar_membrane` a logical vector

`ER` a logical vector

`nuclear_periphery` a logical vector

endosome a logical vector

bud_neck a logical vector

microtubule a logical vector

Golgi a logical vector

late_Golgi a logical vector

peroxisome a logical vector

actin a logical vector

nucleolus a logical vector

cytoplasm a logical vector

ER_to_Golgi a logical vector

early_Golgi a logical vector

lipid_particle a logical vector

nucleus a logical vector

bud a logical vector

Explanation for missing abundance values are given by

missingAbundance a factor with levels low signal, not visualized and technical problem

Details

The information on abundance is available in three columns. abundance gives (where available) absolute protein abundances determined by quantitative Western blot analysis of TAP-tagged strains. Abundances that have a non-NA error value were done in triplicate with serial dilutions of purified TAP-tagged standards included in each gel, which substantially reduces the measurement error. In addition, for these strains, the tagged genes were confirmed to rescue the loss of function phenotype of the corresponding deletion strain. For rows where abundance is missing (NA), the missingAbundance column gives the reason. Possible reasons are:

"not visualized" Either the tagging was unsuccessful or no signal was detected.

"low signal" The tagging was successful, but the signal was not sufficiently high above background to permit accurate quantitation (about 50 molecules/cell).

"technical problem" The protein was detectable but could not be quantitated because it did not migrate as a single band or comigrated with the internal standards in the gel.

Replicate analysis for a subset of tagged strains found a linear correlation coefficient of R = 0.94, with the pairs of proteins having a median variation of a factor of 2.0. This error analysis does not account for potential alterations in the endogenous levels of the proteins caused by the the fused tag, which may be particularly disruptive for small proteins.

Source

The data were obtained from http://yeastgfp.ucsf.edu/, which contains a lot more information as well as raw image data. This data frame was specifically generated from http://yeastgfp.ucsf.edu/allOrfData.txt

References

For the Localization data: Huh, et al., Nature 425, 686-691 (2003) – http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=14562095&dopt=Abstract

For the Protein abundance data: Ghaemmaghami, et al., Nature 425, 737-741 (2003) – http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=14562106&dopt=Abstract

Examples

```
data(gfp)
keep <- names(which(table(gfp$localization_summary) > 50))

if (require(lattice)) {
 bwplot(reorder(localization_summary, abundance, median, na.rm = TRUE) ~ log2(abundance), gfp,
        varwidth = TRUE,
        subset = localization_summary %in% keep)
} else {

  opar <- par(las = 2, mar = par("mar") + c(3.5, 0, 0, 0))
  gfp._sub <- subset(gfp, localization_summary %in% keep)
  gfp._sub$localization_summary <- gfp._sub$localization_summary[, drop = TRUE]
 boxplot(log2(abundance) ~ reorder(localization_summary, abundance, median, na.rm = TRUE),
         data = gfp._sub, varwidth = TRUE)
  rm(gfp._sub)
  par(opar)

}
```

---

litG *Literature and Y2H interaction graphs*

Description

These two data objects represent protein/gene interactions as reported in Ge et al.

Usage

```
data(litG)
       data(y2hG)
```

Format

The data are stored as instances of the graphNEL class. Each has 2885 nodes, named using yeast standard names. Interactions either represent literature curated interactions, or Y2H interactions.

Details

The data were reported and used in Correlation between transcriptome and interactome mapping data from Saccharomyces cerevisiae, Nature Genetics, 2001, H. Ge and Z. Liu and G. M. Church and M. Vidal.

See the package vignette for more details.

Examples

```
data(litG)
data(y2hG)
```

`nPdist` *Counts from a node permutation experiment.*

Description

In comparing the yeast cell cycle data to the protein-protein interaction data a node permutation distribution is suggested. The output here are the counts of the number of common edges for 500 permutations.

Usage

```
data(nPdist)
```

Format

Five hundred counts of nodes in the intersection of two graphs.

Details

The seed was `123` and we called the function `nodePerm` with `litG` and `cg1` as arguments. See the vignette in this package for the explicit computations.

Examples

```
data(nPdist)
```

proteinProperties *Properties of Yeast proteins*

Description

A data frame which details 33 properties of proteins in the Yeast Genome

Usage

```
data(proteinProperties)
```

Format

A data frame with 6718 observations on the following 33 variables.

`yORF` a factor representing yeast ORF names, with levels `Q0010`, `Q0017`, etc.

`SGDID` a factor representing SGD IDs

`molwt` a numeric vector giving Molecular Weight in Daltons

`pi` a numeric vector denoting the theoretical isoelectric point(pI), the pH at which the protein carries no net charge

`cai` a numeric vector denoting Codon Adaptation Index

`length` a numeric vector denoting length of the protein (number of amino acids)

`nterm` a factor representing N Term Sequence with levels `MAAACIC` `MAAAPWY`, etc.

`cterm` a factor representing N Term Sequence with levels `AAAAMLL` `AAADKKT`, etc.

`codonBias` a numeric vector denoting Codon Bias

The next set of columns, designated by amino acids, is the number of times that particular residue appears in the protein sequence. For example, if the ALA column is 2, then the protein contains 2 alanines. These columns (should) add up to the `length` column.

`ALA` a numeric vector

`ARG` a numeric vector

`ASN` a numeric vector

`ASP` a numeric vector

`CYS` a numeric vector

`GLN` a numeric vector

`GLU` a numeric vector

`GLY` a numeric vector

`HIS` a numeric vector

`ILE` a numeric vector

`LEU` a numeric vector

`LYS` a numeric vector

`MET` a numeric vector

`PHE` a numeric vector

`PRO` a numeric vector

`SER` a numeric vector

`THR` a numeric vector

`TRP` a numeric vector

`TYR` a numeric vector

`VAL` a numeric vector

The remaining columns are:

`fop` FOP score, a numeric vector, denoting Frequency of Optimal Codons

`gravy` Gravy score, a numeric vector denoting Hydropathicity of Protein

`aromaticity` Aromaticity score, a numeric vector denoting Frequency of aromatic amino acids: Phe, Tyr, Trp

`type` Feature type, a factor with levels `ORF|Dubious` `ORF|Uncharacterized` `ORF|Verified` `ORF|Verified|silenced_gen` `pseudogene` `transposable_element_gene`

Details

This data frame is downloaded directly from SGD. It contains 33 characteristics for 6714 open reading frames (ORFS). From the SGD README:

"Contains basic protein information about each ORF in SGD. This file does not include information on deleted or merged ORFs. Note, however, that it includes ORFs of all other classifications (Verified, Uncharacterized, and Dubious)."

For more details see http://www.yeastgenome.org/help/protein_page.html.

Source

ftp://genome-ftp.stanford.edu/pub/yeast/protein_info/protein_properties.tab. This file is updated weekly (Saturday). The version used here was downloaded on 2009-12-03.

Examples

```
data(proteinProperties)
pairs(proteinProperties[, c("molwt", "pi", "cai", "gravy", "aromaticity")],
      pch = ".", col = as.numeric(proteinProperties$type))
```

Index

∗ **datasets**
ccyclered, 2
fcabundance, 3
gfp, 4
litG, 6
nPdist, 7
proteinProperties, 8

ccyclered, 2

fcabundance, 3

gfp, 4

litG, 6

nPdist, 7

proteinProperties, 8

y2hG *(*litG*)*, 6