ppiData

October 5, 2010

`Cagney2001BPGraph` *A directed Graph for the Y2H Bait to Prey Interaction data detected by Cagney et al. 2001.*

Description

An instance of class graph, Cagney2001BPGraph is a graphNEL object. The nodes are the union of viable baits (VB) and viable prey (VP) of the experiment conducted by Cagney et al. 2001. A viable bait is a node that has at least one directed edge for which this node serves as the source. A viable prey is a node that has at least one directed edge for which this node serves as a sink.

The data from Cagney et al. 2001 were obtained via the Intact repository. The Intact repository assigns an Intact specific code for each VB and each VP. Each Intact ID is mapped to the a SGD ID (if available) and then mapped to the gene systematic name via the org.Sc.sgdCOMMON2ORF environment of the org.Sc.sgd R-data package. If the mapping is one to many at any point, the first entry of the mapping is selected.

For example: x –> (a,b); a –> (c,d); b –> (e,f), in our algorithm, x ——-> c

While this selection is arbitrary, there is no definitive way to select from the mappings without more information. If more information concerning the experiment is given, the non-arbitrary choice can be made.

If no proper mapping could be achieved in this manner, the Intact ID's were kept so that other methods can be employed for translation in the future. In all 0 VBs and 0 VPs could not be mapped to the gene systematic names, and so retain the Intact ID's for Cagney 2001.

These graphs are not simple. While we chose not to present data with multiple edges between nodes (i.e. if bait b found prey p with multiplicity k, we do not assign k directed edges from b to p, only a single edge). We do, however, allow self loops to detail homodimer relationships.

Usage

```
data(Cagney2001BPGraph)
```

Format

The format is: graphNEL "Cagney2001BPGraph"

Source

The sparse matrix adjacency matrix for this graph can be found in the bioconductor R-package ppiStats.

References

Two-hybrid analysis of the Saccharomyces cerevisiae 26S proteasome. Physiol Genomics. 2001 Oct 10;7(1):27-34.

Examples

```
data(Cagney2001BPGraph)
```

`Gavin2002BPGraph` *A directed Graph for the AP-MS Bait to Prey Interaction data detected by Gavin et al. 2002.*

Description

An instance of class graph, Gavin2002BPGraph is a graphNEL object. The nodes are the union of viable baits (VB) and viable prey (VP) of the experiment conducted by Gavin et al. 2002. A viable bait is a node that has at least one directed edge for which this node serves as the source. A viable prey is a node that has at least one directed edge for which this node serves as a sink.

One key fact is that Gavin et al. Used the protein common names when they recorded the data. As we have chosen to use the gene systematic names, we had to map from these common names to systematic names. We used the org.Sc.sgdCOMMON2ORF environment of the org.Sc.sgd R-data package to translate from common protein names to systematic gene names.

We encountered two mappings which were not one to one. The VPs "Osh1" and "Swh1" both mapped to the Open Reading Frame (ORF) "YAR042W". The VPs "Blm3" and the alias "YFL006W" both mapped to the ORF "YFL007W". When mappings are not one to one, it is difficult to reproduce all the information.

The VPs "Osh1" and "Swh1" were found by the same VB "Scs2" while "Blm3" and the alias "YFL006W" were both detected by the VB "Scl1". In creating this graphNEL, we deleted the VPs "Swh1" and "YFL006W".

If, on the other hand, one protein mapped to several different ORFs we simple selected the first ORF in the list since there is no definitive process to make the choice un-arbitrary.

For example: x –> (a,b), then x —> a in our algorithm.

These graphs are not simple. While we chose not to present data with multiple edges between nodes (i.e. if bait b found prey p with multiplicity k, we do not assign k directed edges from b to p, only a single edge). We do, however, allow self loops to detail homodimer relationships.

Usage

```
data(Gavin2002BPGraph)
```

Format

The format is: graphNEL "Gavin2002BPGraph"

Source

The adjacency matrix for this graph can be found in the bioconductor R-package apComplex.

References

Functional organization of the yeast proteome by systematic analysis of protein complexes. Nature. 2002 Jan 10;415(6868):141-7

Examples

```
data(Gavin2002BPGraph)
```

`Gavin2006BPGraph` *A directed Graph for the AP-MS Bait to Prey Interaction data detected by Gavin et al. 2006.*

Description

An instance of class graph, Gavin2006BPGraph is a graphNEL object. The nodes are the union of viable baits (VB) and viable prey (VP) of the experiment conducted by Gavin et al. 2006. A viable bait is a node that has at least one directed edge for which this node serves as the source. A viable prey is a node that has at least one directed edge for which this node serves as a sink.

The data from Gavin et al. 2006 were obtained via the Intact repository. The Intact repository assigns an Intact specific code for each VB and each VP. Each Intact ID is mapped to the a SGD ID (if available) and then mapped to the gene systematic name via the org.Sc.sgdCOMMON2ORF environment of the org.Sc.sgd R-data package. If the mapping is one to many at any point, the first entry of the mapping is selected. While this selection is arbitrary, there is no definitive way to select from the mappings without more information. If more information concerning the experiment is given, the non-arbitrary choice can be made.

For example: x –> (a,b); a –> (c,d); b –> (e,f), in our algorithm, x ——-> c

If no proper mapping could be achieved in this manner, the Intact ID's were kept so that other methods can be employed for translation in the future. In all 3 VBs and 8 VPs could not be mapped to the gene systematic names.

These graphs are not simple. While we chose not to present data with multiple edges between nodes (i.e. if bait b found prey p with multiplicity k, we do not assign k directed edges from b to p, only a single edge). We do, however, allow self loops to detail homodimer relationships.

Usage

```
data(Gavin2006BPGraph)
```

Format

The format is: graphNEL "Gavin2006BPGraph"

Source

The adjacency matrix for this graph can be found in the bioconductor R-package apComplex.

References

Proteome survey reveals modularity of the yeast cell machinery. Nature. 2006 Mar 30;440(7084):631-6. Epub 2006 Jan 22.

Examples

```
data(Gavin2006BPGraph)
```

`Hazbun2003BPGraph` *A directed Graph for the Y2H Bait to Prey Interaction data detected by Hazbun et al. 2003.*

Description

An instance of class graph, Hazbun2003BPGraph is a graphNEL object. The nodes are the union of viable baits (VB) and viable prey (VP) of the experiment conducted by Hazbun et al. 2003. A viable bait is a node that has at least one directed edge for which this node serves as the source. A viable prey is a node that has at least one directed edge for which this node serves as a sink.

The data from Hazbun et al. 2003 were obtained via the Intact repository. The Intact repository assigns an Intact specific code for each VB and each VP. Each Intact ID is mapped to the a SGD ID (if available) and then mapped to the gene systematic name via the org.Sc.sgdCOMMON2ORF environment of the org.Sc.sgd R-data package. If the mapping is one to many at any point, the first entry of the mapping is selected. While this selection is arbitrary, there is no definitive way to select from the mappings without more information. If more information concerning the experiment is given, the non-arbitrary choice can be made.

For example: x –> (a,b); a –> (c,d); b –> (e,f), in our algorithm, x ——-> c

If no proper mapping could be achieved in this manner, the Intact ID's were kept so that other methods can be employed for translation in the future. In all 0 VBs and 12 VPs could not be mapped to the gene systematic names, and so retain the Intact ID's.

These graphs are not simple. While we chose not to present data with multiple edges between nodes (i.e. if bait b found prey p with multiplicity k, we do not assign k directed edges from b to p, only a single edge). We do, however, allow self loops to detail homodimer relationships.

Usage

```
data(Hazbun2003BPGraph)
```

Format

The format is: graphNEL "Hazbun2003BPGraph"

Source

The sparse matrix adjacency matrix for this graph can be found in the bioconductor R-package ppiStats.

References

Assigning function to yeast proteins by integration of technologies. Mol Cell. 2003 Dec;12(6):1353-65.

Examples

```
data(Hazbun2003BPGraph)
```

`Ho2002BPGraph` *A directed Graph for the AP-MS Bait to Prey Interaction data detected by Ho et al. 2002.*

Description

An instance of class graph, Ho2002BPGraph is a graphNEL object. The nodes are the union of viable baits (VB) and viable prey (VP) of the experiment conducted by Ho et al. 2002. A viable bait is a node that has at least one directed edge for which this node serves as the source. A viable prey is a node that has at least one directed edge for which this node serves as a sink.

All nodes are indexed by the gene systematic names as given by the primary source.

These graphs are not simple. While we chose not to present data with multiple edges between nodes (i.e. if bait b found prey p with multiplicity k, we do not assign k directed edges from b to p, only a single edge). We do, however, allow self loops to detail homodimer relationships.

Usage

```
data(Ho2002BPGraph)
```

Format

The format is: graphNEL "Ho2002BPGraph"

Source

The adjacency matrix for this graph can be found in the bioconductor R-package apComplex.

References

Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry. Nature. 2002 Jan 10;415(6868):180-3.

Examples

```
data(Ho2002BPGraph)
```

`Ito2001BPGraph` *A directed Graph for the Y2H Bait to Prey Interaction data detected by Ito et al. 2001.*

Description

An instance of class graph, Ito2001BPGraph is a graphNEL object. The nodes are the union of viable baits (VB) and viable prey (VP) of the experiment conducted by Ito et al. 2001. A viable bait is a node that has at least one directed edge for which this node serves as the source. A viable prey is a node that has at least one directed edge for which this node serves as a sink.

The data from Ito et al. 2001 were obtained via the Intact repository. The Intact repository assigns an Intact specific code for each VB and each VP. Each Intact ID is mapped to the a SGD ID (if

available) and then mapped to the gene systematic name via the org.Sc.sgdCOMMON2ORF environment of the org.Sc.sgd R-data package. If the mapping is one to many at any point, the first entry of the mapping is selected. While this selection is arbitrary, there is no definitive way to select from the mappings without more information. If more information concerning the experiment is given, the non-arbitrary choice can be made.

For example: x –> (a,b); a –> (c,d); b –> (e,f), in our algorithm, x ——-> c

If no proper mapping could be achieved in this manner, the Intact ID's were kept so that other methods can be employed for translation in the future. In all 7 VBs and 12 VPs could not be mapped to the gene systematic names, and so retain the Intact ID's.

These graphs are not simple. While we chose not to present data with multiple edges between nodes (i.e. if bait b found prey p with multiplicity k, we do not assign k directed edges from b to p, only a single edge). We do, however, allow self loops to detail homodimer relationships.

Usage

```
data(Ito2001BPGraph)
```

Format

The format is: graphNEL "Ito2001BPGraph"

Source

The sparse matrix adjacency matrix for this graph can be found in the bioconductor R-package ppiStats.

References

A comprehensive two-hybrid analysis to explore the yeast protein interactome. Proc Natl Acad Sci U S A. 2001 Apr 10;98(8):4569-74. Epub 2001 Mar 13.

Examples

```
data(Ito2001BPGraph)
```

ItoCore2001BPGraph *A directed Graph for the Y2H Bait to Prey Interaction data detected by Ito et al. 2001.*

Description

An instance of class graph, ItoCore2001BPGraph is a graphNEL object. The nodes are the union of viable baits (VB) and viable prey (VP) of the experiment conducted by Ito et al. 2001 which they annotated as the core dataset by having verified the bait-prey interaction at least 3 times. A viable bait is a node that has at least one directed edge for which this node serves as the source. A viable prey is a node that has at least one directed edge for which this node serves as a sink.

The data from Ito et al. 2001 were obtained via the Intact repository. The Intact repository assigns an Intact specific code for each VB and each VP. Each Intact ID is mapped to the a SGD ID (if available) and then mapped to the gene systematic name via the org.Sc.sgdCOMMON2ORF environment of the org.Sc.sgd R-data package. If the mapping is one to many at any point, the first entry of the mapping is selected. While this selection is arbitrary, there is no definitive way to select from

the mappings without more information. If more information concerning the experiment is given, the non-arbitrary choice can be made.

For example: x –> (a,b); a –> (c,d); b –> (e,f), in our algorithm, x ——-> c

If no proper mapping could be achieved in this manner, the Intact ID's were kept so that other methods can be employed for translation in the future. In all 7 VBs and 12 VPs could not be mapped to the gene systematic names, and so retain the Intact ID's.

These graphs are not simple. While we chose not to present data with multiple edges between nodes (i.e. if bait b found prey p with multiplicity k, we do not assign k directed edges from b to p, only a single edge). We do, however, allow self loops to detail homodimer relationships.

Usage

```
data(ItoCore2001BPGraph)
```

Format

The format is: graphNEL "ItoCore2001BPGraph"

Source

The sparse matrix adjacency matrix for this graph can be found in the bioconductor R-package ppiStats.

References

A comprehensive two-hybrid analysis to explore the yeast protein interactome. Proc Natl Acad Sci U S A. 2001 Apr 10;98(8):4569-74. Epub 2001 Mar 13.

Examples

```
data(ItoCore2001BPGraph)
```

ItoFull2001BPGraph *A directed Graph for the Y2H Bait to Prey Interaction data detected by Ito et al. 2001.*

Description

An instance of class graph, ItoFull2001BPGraph is a graphNEL object. The nodes are the union of viable baits (VB) and viable prey (VP) of the experiment conducted by Ito et al. 2001. A viable bait is a node that has at least one directed edge for which this node serves as the source. A viable prey is a node that has at least one directed edge for which this node serves as a sink.

The data from Ito et al. 2001 were obtained via the Intact repository. The Intact repository assigns an Intact specific code for each VB and each VP. Each Intact ID is mapped to the a SGD ID (if available) and then mapped to the gene systematic name via the org.Sc.sgdCOMMON2ORF environment of the org.Sc.sgd R-data package. If the mapping is one to many at any point, the first entry of the mapping is selected. While this selection is arbitrary, there is no definitive way to select from the mappings without more information. If more information concerning the experiment is given, the non-arbitrary choice can be made.

For example: x –> (a,b); a –> (c,d); b –> (e,f), in our algorithm, x ——-> c

If no proper mapping could be achieved in this manner, the Intact ID's were kept so that other methods can be employed for translation in the future. In all 7 VBs and 12 VPs could not be mapped to the gene systematic names, and so retain the Intact ID's.

These graphs are not simple. While we chose not to present data with multiple edges between nodes (i.e. if bait b found prey p with multiplicity k, we do not assign k directed edges from b to p, only a single edge). We do, however, allow self loops to detail homodimer relationships.

Usage

```
data(ItoFull2001BPGraph)
```

Format

The format is: graphNEL "ItoFull2001BPGraph"

Source

The sparse matrix adjacency matrix for this graph can be found in the bioconductor R-package ppiStats.

References

A comprehensive two-hybrid analysis to explore the yeast protein interactome. Proc Natl Acad Sci U S A. 2001 Apr 10;98(8):4569-74. Epub 2001 Mar 13.

Examples

```
data(ItoFull2001BPGraph)
```

Krogan2004BPGraph *A directed Graph for the AP-MS Bait to Prey Interaction data detected by Krogan et al. 2004.*

Description

An instance of class graph, Krogan2004BPGraph is a graphNEL object. The nodes are the union of viable baits (VB) and viable prey (VP) of the experiment conducted by Krogan et al. 2004. A viable bait is a node that has at least one directed edge for which this node serves as the source. A viable prey is a node that has at least one directed edge for which this node serves as a sink.

All nodes are indexed by the gene systematic names as given in the primary source.

These graphs are not simple. While we chose not to present data with multiple edges between nodes (i.e. if bait b found prey p with multiplicity k, we do not assign k directed edges from b to p, only a single edge). We do, however, allow self loops to detail homodimer relationships.

Usage

```
data(Krogan2004BPGraph)
```

Format

The format is: graphNEL "Krogan2004BPGraph"

Source

The adjacency matrix for this graph can be found in the bioconductor R-package apComplex.

References

High-definition macromolecular composition of yeast RNA-processing complexes. Mol Cell. 2004 Jan 30;13(2):225-39.

Examples

```
data(Krogan2004BPGraph)
```

Krogan2006BPGraph *A directed Graph for the AP-MS Bait to Prey Interaction data detected by Krogan et al. 2006.*

Description

An instance of class graph, Krogan2006BPGraph is a graphNEL object. The nodes are the union of viable baits (VB) and viable prey (VP) of the experiment conducted by Krogan et al. 2006. A viable bait is a node that has at least one directed edge for which this node serves as the source. A viable prey is a node that has at least one directed edge for which this node serves as a sink.

All nodes are indexed by the gene systematic names as given by the primary source.

These graphs are not simple. While we chose not to present data with multiple edges between nodes (i.e. if bait b found prey p with multiplicity k, we do not assign k directed edges from b to p, only a single edge). We do, however, allow self loops to detail homodimer relationships.

Usage

```
data(Krogan2006BPGraph)
```

Format

The format is: graphNEL "Krogan2006BPGraph"

Source

The adjacency matrix for this graph can be found in the bioconductor R-package apComplex.

References

Global landscape of protein complexes in the yeast Saccharomyces cerevisiae. Nature 440, 637-643 (30 March 2006).

Examples

```
data(Krogan2006BPGraph)
```

`Tong2002BPGraph` *A directed Graph for the Y2H Bait to Prey Interaction data detected by Tong et al. 2002.*

Description

An instance of class graph, Tong2002BPGraph is a graphNEL object. The nodes are the union of viable baits (VB) and viable prey (VP) of the experiment conducted by Tong et al. 2002. A viable bait is a node that has at least one directed edge for which this node serves as the source. A viable prey is a node that has at least one directed edge for which this node serves as a sink.

The data from Tong et al. 2002 were obtained via the Intact repository. The Intact repository assigns an Intact specific code for each VB and each VP. Each Intact ID is mapped to the a SGD ID (if available) and then mapped to the gene systematic name via the org.Sc.sgdCOMMON2ORF environment of the org.Sc.sgd R-data package. If the mapping is one to many at any point, the first entry of the mapping is selected. While this selection is arbitrary, there is no definitive way to select from the mappings without more information. If more information concerning the experiment is given, the non-arbitrary choice can be made.

For example: x –> (a,b); a –> (c,d); b –> (e,f), in our algorithm, x ——-> c

If no proper mapping could be achieved in this manner, the Intact ID's were kept so that other methods can be employed for translation in the future. In all 0 VBs and 1 VPs could not be mapped to the gene systematic names, and so retain the Intact ID's.

These graphs are not simple. While we chose not to present data with multiple edges between nodes (i.e. if bait b found prey p with multiplicity k, we do not assign k directed edges from b to p, only a single edge). We do, however, allow self loops to detail homodimer relationships.

Usage

```
data(Tong2002BPGraph)
```

Format

The format is: graphNEL "Tong2002BPGraph"

Source

The sparse matrix adjacency matrix for this graph can be found in the bioconductor R-package ppiStats.

References

A combined experimental and computational strategy to define protein interaction networks for peptide recognition modules. Science. 2002 Jan 11;295(5553):321-4. Epub 2001 Dec 13.

Examples

```
data(Tong2002BPGraph)
```

`Uetz2000BPGraph1` *A directed Graph for the Y2H Bait to Prey Interaction data detected by Uetz et al. 2000.*

Description

An instance of class graph, Uetz2000BPGraph is a graphNEL object. The nodes are the union of viable baits (VB) and viable prey (VP) of the experiment conducted by Uetz et al. 2000. A viable bait is a node that has at least one directed edge for which this node serves as the source. A viable prey is a node that has at least one directed edge for which this node serves as a sink.

The data from Uetz et al. 2000 were obtained via the Intact repository. The Intact repository assigns an Intact specific code for each VB and each VP. Each Intact ID is mapped to the a SGD ID (if available) and then mapped to the gene systematic name via the org.Sc.sgdCOMMON2ORF environment of the org.Sc.sgd R-data package. If the mapping is one to many at any point, the first entry of the mapping is selected. While this selection is arbitrary, there is no definitive way to select from the mappings without more information. If more information concerning the experiment is given, the non-arbitrary choice can be made.

For example: x –> (a,b); a –> (c,d); b –> (e,f), in our algorithm, x ——-> c

If no proper mapping could be achieved in this manner, the Intact ID's were kept so that other methods can be employed for translation in the future. In all 1 VBs and 2 VPs could not be mapped to the gene systematic names, and so retain the Intact ID's.

These graphs are not simple. While we chose not to present data with multiple edges between nodes (i.e. if bait b found prey p with multiplicity k, we do not assign k directed edges from b to p, only a single edge). We do, however, allow self loops to detail homodimer relationships.

Usage

```
data(Uetz2000BPGraph1)
```

Format

The format is: graphNEL "Uetz2000BPGraph1"

Source

The sparse matrix adjacency matrix for this graph can be found in the bioconductor R-package ppiStats.

References

A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. Nature. 2000 Feb 10;403(6770):623-7.

Examples

```
data(Uetz2000BPGraph1)
```

`Uetz2000BPGraph2` *A directed Graph for the Y2H Bait to Prey Interaction data detected by Uetz et al. 2000.*

Description

An instance of class graph, Uetz2000BPGraph2 is a graphNEL object. The nodes are the union of viable baits (VB) and viable prey (VP) of the experiment conducted by Uetz et al. 2000. A viable bait is a node that has at least one directed edge for which this node serves as the source. A viable prey is a node that has at least one directed edge for which this node serves as a sink.

The data from Uetz et al. 2000 were obtained via the Intact repository. The Intact repository assigns an Intact specific code for each VB and each VP. Each Intact ID is mapped to the a SGD ID (if available) and then mapped to the gene systematic name via the org.Sc.sgdCOMMON2ORF environment of the org.Sc.sgd R-data package. If the mapping is one to many at any point, the first entry of the mapping is selected. While this selection is arbitrary, there is no definitive way to select from the mappings without more information. If more information concerning the experiment is given, the non-arbitrary choice can be made.

For example: x –> (a,b); a –> (c,d); b –> (e,f), in our algorithm, x ——-> c

If no proper mapping could be achieved in this manner, the Intact ID's were kept so that other methods can be employed for translation in the future. In all 0 VBs and 2 VPs could not be mapped to the gene systematic names, and so retain the Intact ID's.

These graphs are not simple. While we chose not to present data with multiple edges between nodes (i.e. if bait b found prey p with multiplicity k, we do not assign k directed edges from b to p, only a single edge). We do, however, allow self loops to detail homodimer relationships.

Usage

```
data(Uetz2000BPGraph2)
```

Format

The format is: graphNEL "Uetz2000BPGraph2"

Source

The sparse matrix adjacency matrix for this graph can be found in the bioconductor R-package ppiStats.

References

A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. Nature. 2000 Feb 10;403(6770):623-7.

Examples

```
data(Uetz2000BPGraph2)
```

`Zhao2005BPGraph` *A directed Graph for the Y2H Bait to Prey Interaction data detected by Zhao et al. 2005.*

Description

An instance of class graph, Zhao2005BPGraph is a graphNEL object. The nodes are the union of viable baits (VB) and viable prey (VP) of the experiment conducted by Zhao et al. 2005. A viable bait is a node that has at least one directed edge for which this node serves as the source. A viable prey is a node that has at least one directed edge for which this node serves as a sink.

The data from Zhao et al. 2005 were obtained via the Intact repository. The Intact repository assigns an Intact specific code for each VB and each VP. Each Intact ID is mapped to the a SGD ID (if available) and then mapped to the gene systematic name via the org.Sc.sgdCOMMON2ORF environment of the org.Sc.sgd R-data package. If the mapping is one to many at any point, the first entry of the mapping is selected. While this selection is arbitrary, there is no definitive way to select from the mappings without more information. If more information concerning the experiment is given, the non-arbitrary choice can be made.

For example: x –> (a,b); a –> (c,d); b –> (e,f), in our algorithm, x ——-> c

If no proper mapping could be achieved in this manner, the Intact ID's were kept so that other methods can be employed for translation in the future. In all 0 VBs and 0 VPs could not be mapped to the gene systematic names, and so retain the Intact ID's.

These graphs are not simple. While we chose not to present data with multiple edges between nodes (i.e. if bait b found prey p with multiplicity k, we do not assign k directed edges from b to p, only a single edge). We do, however, allow self loops to detail homodimer relationships.

Usage

```
data(Zhao2005BPGraph)
```

Format

The format is: graphNEL "Zhao2005BPGraph"

Source

The sparse matrix adjacency matrix for this graph can be found in the bioconductor R-package ppiStats.

References

Navigating the chaperone network: an integrative map of physical and genetic interactions mediated by the hsp90 chaperone. Cell. 2005 Mar 11;120(5):715-27.

Examples

```
data(Zhao2005BPGraph)
```

bpExperimentNames *A Character Vector Containing the Names of the 13 Graph Objects Stored in this Data Package*

Description

This is a character vector of the names of the graph objects stored in this data directory.

Usage

```
data(bpExperimentNames)
```

Format

The format is: chr "bpExperimentNames"

Examples

```
data(bpExperimentNames)
bpExperimentNames
```

collectIntactPPIData *A function that parses the Intact repository for bait to prey affiliation*

Description

This function is the parsing function for the Intact downloaded data-sets.

Usage

```
collectIntactPPIData(intactID)
```

Arguments

intactID	A character vector of the Intact ID's that reference the particular experiment needed to be parsed.

Details

The collectIntactPPIData is a general function that reads the R data file tableList.rda and gathers information for the set of desired experiments.

The tableList.rda is generated by running the script parseIntAct.R to download and generate a data file will the latest data from IntAct.

Value

allBaits	A character vector of all distince baits over all experiments
allPreys	A character vector of all distinct preys over all experiments
indexSetAll	A list of lists. Each element of the top list corresponds to a particular experiment; the elements of these sub-lists are character vectors. The character vectors will always carry the bait protein in the first element and all of the preys that particular bait has found as the rest of the elements to the vector. Note that the multiplicity is not guarenteed to be unity.
baitsSystematic	A character vector with a number of baits mapped to their systematic names
preysSystematic	A character vector with a number of preys mapped to their systematic names
shortLabel	A character vector listing the first author of each experiment.

Author(s)

T Chiang

References

http://www.ebi.ac.uk/intact

Examples

```
inTactData <- collectIntactPPIData(c("EBI-375746", "EBI-531419",
                                     "EBI-295760", "EBI-698096"))
```

---

createBPList *A function to create the Bait to Prey association list.*

Description

This function takes the indexSetAll, baitsSystematic, and preysSystematic entries respectively from the output list of the R function collectIntactPPIData, and it generates a three tier-ed list which ultimately records the bait to prey association from each protein-protein interaction (ppi) experiment obtained from the IntAct repository.

Usage

```
createBPList(indexSet, baitsSystematic, preysSystematic)
```

Arguments

indexSet	A list of named lists. The names correspond to each experiment while the entries to the lower level lists are character vectors with two elements: the first element is the id for the bait protein and the second an id for the prey. This is an entry of the output of the collectIntactData function called indexSetAll

`baitsSystematic`
: A named list. The names correspond to the IntAct acension codes for any proteins used as baits, and the entries of the list are character vectors consisting of Systematic gene names with respect to the IntAct ID. This is also an entry from the output of the function collectIntactData.

`preysSystematic`
: A named list. The names correspond to the IntAct acension codes for any proteins used as preys, and the entries of the list are character vectors consisting of Systematic gene names with respect to the IntAct ID. This is also an entry from the output of the function collectIntactData.

Value

This function generates a three tiered list. The top tier is the main list. The second tier-ed set of lists should be named or referenced by experimentors. The third tier is a list of bait to prey associations. The names in the third tier of the list are those proteins sampled as baits, and the entries are character vectors of proteins detected as prey.

Author(s)

T Chiang

Examples

```
data(y2h)
eg = y2h$indexSetAll[3]
eg1 = y2h$baitsSystematic[3]
eg2 = y2h$preysSystematic[3]
createBPList(eg, eg1, eg2)
```

---

intAct2Sys *A function that maps the Intact ID's to the first org.Sc.sgd systematic name using the org.Sc.sgd package*

Description

This function takse the Intact ID's maps to the yeast common names via intact repository and uses the org.Sc.sgd package to map the common names to the org.Sc.sgd Systematic names. There is problems if more than one systematic name corresponds to the common name. This function will simply always return the first name given by each mapping: from Intact to common names and also from common names to systematic names.

Usage

```
intAct2Sys(prot2Sys, bpSysL)
```

Arguments

`prot2Sys`
: An Intact accession number for some protein which is needed to be mapped to its systematic name

`bpSysL`
: A named list with the IntAct accession numbers as the names and the corresponding systematic names as the entries

Value

The return value is a mapping of each Intact ID to it's respecting systematic name.

Author(s)

T Chiang

---

intActPPIData *A data set documenting the output of collectIntactPPIData*

Description

This data set is the output for the function collectIntactPPIData. Please refer to that man page of that function for a full description of the data set.

Usage

```
data(intActPPIData)
```

Format

The format is: chr "y2h"

Examples

```
data(intActData)
```

---

map2Systematic *A function that maps the Intact ID's to Yeast Systematic Names if possible*

Description

This function takse the IntAct accension numbers and maps to the yeast common names via intact repository and uses the org.Sc.sgd package to map the common names to the org.Sc.sgd Systematic names.

Usage

```
map2Systematic(allProt, tableList, sWAC)
```

Arguments

allProt	A character vector of proteins to be mapped
tableList	The tableList element of the output from collectIntactData
sWAC	A named character vector; the names are SwissProt Accesion Codes and the elemnets are the Yeast Systematic names

Value

The return value is a mapping of each Intact ID to it's respecting systematic name.

Author(s)

T Chiang

Examples

```
data(sWAC2Sys)
dataL = collectIntactData("EBI-375746")
sysN = map2Systematic(dataL$allBaits, dataL$tableList, sWAC2Sys)
```

`map2Systematic2` *A secondary function that maps the Intact ID's to Yeast Systematic Names if possible*

Description

This function takse the Intact accession numbers and maps to the SGD ID's via intact repository and truncates the SGD ID's to find the yeast systematic names.

Usage

```
map2Systematic2(allProt, tableList, sWAC)
```

Arguments

`allProt`	A character vector of protein ID's
`tableList`	A List of tables. The parsed XLM files of the IntAct database
`sWAC`	The data file sWAC2Sys

Value

A character vector - The return value is a mapping of each Intact ID to it's respecting systematic name.

Author(s)

T Chiang

ppipred *Protein-protein predictions*

Description

The data in ppipred are those provided by Liu et al for S. cervisiae protein protein interactions.

Usage

```
data(ppipred)
```

Format

A data frame with 20088 observations on the following 3 variables.

Gene1 A vector of the gene names for one interactor.

Gene2 A vector of the gene names for the other interactor.

Probability A vectror containing the probability they give to the interaction.

Details

Liu et al discuss a method for estimating protein-protein interactions from existing data, across species. They have provided the data with the following description: "We compute domain-domain interaction probabilities from Y2H protein-protein interactions, and then use these domain-domain interaction probabilities to compute the interaction probability between every pair of proteins. The prediction results with a false positive rate fp=3E-4 and a false negative rate fn=0.85 are listed blow."

Source

http://bioinformatics.med.yale.edu/interaction/

References

Inferring protein-protein interactions through high-throughput interaction data from diverse organisms, Y. Liu, N. Liu and H. Zhao, Bioinformatics, 2005, 21, 3279-3285.

Examples

```
data(ppipred)
```

proteinProperty *A data frame which details 33 properties of the Yeast Genome*

Description

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). This file is updated weekly (Saturday).

The columns are below; only the first two columns are mandatory. The column designated by an amino acid is the number of that particular residue in the protein sequence. For example, if the ALA column is 2, then the protein contains 2 alanines.

The columns of the dataframe are:

FEATURE (ORF name) SGDID MOLECULAR WEIGHT (in Daltons) PI CAI (Codon Adaptation Index) PROTEIN LENGTH N TERM SEQ C TERM SEQ CODON BIAS ALA ARG ASN ASP CYS GLN GLU GLY HIS ILE LEU LYS MET PHE PRO SER THR TRP TYR VAL FOP SCORE (Frequency of Optimal Codons) GRAVY SCORE (Hydropathicity of Protein) AROMATICITY SCORE (Frequency of aromatic amino acids: Phe, Tyr, Trp) Feature type (ORF classification: Verified, Uncharacterized, Dubious)

For more details, please visit the SGD website.

Usage

```
data(proteinProperty)
```

Format

The format is: data.frame "proteinProperty"

Source

ftp://genome-ftp.stanford.edu/pub/yeast/protein_info/protein_properties.tab

Examples

```
data(proteinProperty)
```

sWAC2Sys *A data set that contains the translation between Swiss accension codes to Gene Systematic Names*

Description

This is a named character vector. The names are the Swiss Acension Codes and the elements are the corresponding gene systematic names

Usage

```
data(sWAC2Sys)
```

Format

The format is: chr "sWAC2Sys"

Examples

```
data(sWAC2Sys)
```

---

stochasticBaits *Lists of viable baits in each bait to prey dataset which are conjectured not to be systematically biased.*

Description

These are named lists consisting of viable baits corresponding to each dataset which are not believed to suffer from systematic bias due to the bait to prey systems.

The names should be listed by the systematic gene names. If the systematic gene names could not be found, then a common name is used. If the data was obtained from IntAct, and neither a common nor a systematic name could be found, the IntAct Accension Identification Code ("EBI-*****") is retained. The IntAct Codes will always be listed first if they exist.

Because we have chosen to supress the homodimer relationships within the respective bait to prey directed graphs, there will be proteins listed as viable prey which index isolated nodes in the graphs. These isolated nodes simply mean that these proteins were only seen to interact within homodimer relationships.

Usage

```
data(stochasticBaits)
```

Format

A named list. Each element is a character vector of the viable baits for the corresponding dataset.

Examples

```
data(stochasticBaits)
stochasticBaits[[1]][1:5]
```

`tableList` *A list of tables for the downloaded and parsed xml files from the IntAct database*

Description

This is a list of tables generated from the a script used to parse the xml files from the IntAct data repository.

tableList is a list that contains six data-frames (tables) that records all the various data given by IntAct. The names of these six tables are:

1. "acInfo" 2. "experimentInfo" 3. "ac2xref" 4. "interactionInfo" 5. "experiment2interaction" 6. "interaction2inteactor"

The "acInfo" data frame contains 4 columns: "ac", "type", "shortLabel", "fullName". They are IntAct accession numbers, types (experiment, interaction, or interactor), and short/long names, respectively. The reason to have the second column is because IntAct assigns an accession number to each experiment, interaction, and interactor. The second column is an indicator between these three. The accession numbers in the first column of this data frame are those accession numbers in the yeast xml files from the IntAct repository.

There are also 4 columns in the "experimentInfo" data frame: "ac","hostOrganism","interactionDetection", and "participantDetection". The IntAct accession numbers of experiments will be in the first column. The "hostOrganism" column is the tax id of the host organism (yeast in this case). The next two columns are the detection methods of the interaction and of the participant, respectively. The values in the 3rd and 4th columns are Proteomics Standards Initiative (PSI) Molecular Interaction (MI) identification codes. IntAct documents both the PSI MI 1.0 as well as PSI MI 2.5 codes.

The "ac2xref" data frame contains 4 columns: "ac", "db", "id", and "secondary". This data frame provides the cross references (among several databases) for each IntAct accession number. The column "ac" is the IntAct accession numbers, and the columns "db" and "id" are the reference database names and the corresponding identification code for that database. The secondary column provides secondary id if provided.

There are also 4 columns in the "interactionInfo" data frame: "ac", "interactionType", "confidenceUnit", and "confidenceValue". The first column contains all the interaction accession numbers among these xml files. The second column is the type of interaction. The values here are also the PSI MI codes. The third and forth columns are the confidence information for this interaction.

There are 2 columns in the "experiment2interaction" data frame: "experiment", and "interaction". This data frame tells us the what types of interactions were derived from the experiment. Both the experiments and the interactions are given by the IntAct accession numbers.

There are 3 columns in the "interaction2inteactor" data frame: "interaction", "interactor", and "role". The first column provide the type of interaction while the second column provides the interactors within the interaction. Both the "interaction" and the "interactor" are given by the IntAct accession numbers. The third column is the role (bait or prey) of the interactors.

Usage

```
data(tableList)
```

Format

The format is: chr "tableList"

Examples

```
data(tableList)
```

`twTableAPMS` *A table to look at the permutation statistics of pairs of Y2H experiment.*

Description

For each pair of APMS experiment, let X be the restriction of common viable proteins. Restricted to the set X, entry n11 of the table details how many times the protein is found in both experiments; n12 details how many times the protein was found in experiment 1 but not experiment 2; entry n21 details how many times the protein was found in experiment 2 and not experiment 1; and n22 details the number of times the proteins were found in neither experiments.

Usage

```
data(twTableAPMS)
```

Format

A list of matrices. Each element of the list contains a table for a particular pair of experimental datasets. The rows are indexed by (n11,n12,n21,n22), and the columns are indexed by the elements of X.

Examples

```
data(twTableAPMS)
```

`twTableY2H` *A table to look at the permutation statistics of pairs of Y2H experiment.*

Description

For each pair of Y2H experiment, let X be the restriction of common viable proteins. Restricted to the set X, entry n11 of the table details how many times the protein is found in both experiments; n12 details how many times the protein was found in experiment 1 but not experiment 2; entry n21 details how many times the protein was found in experiment 2 and not experiment 1; and n22 details the number of times the proteins were found in neither experiments.

Usage

```
data(twTableY2H)
```

Format

A list of matrices. Each element of the list contains a table for a particular pair of experimental datasets. The rows are indexed by (n11,n12,n21,n22), and the columns are indexed by the elements of X.

Examples

```
data(twTableY2H)
```

`viablePrey` *Lists of viable prey or baits in each bait to prey dataset*

Description

These are named lists consisting of either the viable prey or the viable baits corresponding to each dataset.

The names should be listed by the systematic gene names. If the systematic gene names could not be found, then a common name is used. If the data was obtained from IntAct, and neither a common nor a systematic name could be found, the IntAct Accension Identification Code ("EBI-*****") is retained. The IntAct Codes will always be listed first if they exist.

Because we have chosen to supress the homodimer relationships within the respective bait to bait or viable prey directed graphs, there will be proteins listed as viable prey which index isolated nodes in the graphs. These isolated nodes simply mean that these proteins were only seen to interact within homodimer relationships.

Usage

```
data(viableBaits)
data(viablePrey)
```

Format

A named list. Each element is a character vector of the viable prey (or baits) for the corresponding dataset.

Examples

```
data(viablePrey)
data(viableBaits)
names(viablePrey)
names(viableBaits)
viableBaits[[1]][1:5]
```

`y2h` *A data set documenting the bait to prey associations for the y2h empirical data-sets as found within the IntAct repository.*

Description

This data set is the bait to prey assocation after the function createBPList has been called on three entries of intActPPIData: indexSetAll, baitsSystematic, preysSystematic.

The structure of this R data object is a three-tiered list. The first (or top) tier of the list has 42 sub-lists corresponding to the 42 y2h data-sets found within IntAct. Each sub-list is named by the corresponding experimentor and the date of publication. Each one of these sub-lists contains a set of lists; each of these lists is named by an individual viable bait of that experiment. Each viable bait list contains a character vector of the viable preys dectect by the corresponding viable bait.

For a more in depth summary, please see the intactPPIData.Rnw file.

Usage

```
data(y2h)
```

Format

The format is: chr "y2h"

Examples

```
data(y2h)
```

`y2hSysGW` *A data object documenting the bait to prey associations for the y2h empirical data-sets with two restrictions.*

Description

This list is a sublist of the file y2h.rda of bait to prey interactions which have 2 restrictions: Genome Wide prey population and the use of GAL4 transcription factor. There are currently 7 experiments that fall within these restrictions. The protein names in this list have been mapped to the systematic names from the IntAct codes.

Usage

```
data(y2hSysGW)
```

Format

The format is: chr "y2hSysGW"

Details

The format of this data object is the same as that of y2h.rda. Please see the help page for that object.

Examples

```
data(y2hSysGW)
y2hSysGW[[2]]
```

Index