

# Rintact

April 19, 2009

---

bait

*A function that obtains the bait protein from the IntAct Interactions*

---

## Description

The `bait` function takes an `intactInteraction` class and returns the proteins which was sampled as a bait in the binary relationship.

## Usage

```
bait(x)
```

## Arguments

`x` An `intactInteraction` class.

## Value

A character. The name of the protein which was sampled as a bait in the tested binary interaction.

## Author(s)

T Chiang

## References

The IntAct Database <http://www.ebi.ac.uk/intact/site/contents/downloads.jsf>.

## See Also

[intactInteraction-class](#)

## Examples

```
# parse complex data
```

---

complexEntry-class *Class "complexEntry"*

---

### Description

A class representing the complex data extracted from an IntAct data file in PSI-MI 2.5 format.

### Objects from the Class

Objects are usually created by `psi25complex`.

### Slots

**releaseDate:** character, release date of the data file.

**interactors:** matrix. Each row represents one interactor. IntAct IDs are used as row names. Each column represents one annotation. Annotations include: UniProt ID, gene symbol, full name, locus name, ORF name, organism name, and NCBI taxonomy ID.

**complexes:** list, a list of `intactComplex` objects, each represents one complex.

### Methods

**show** signature(object = "complexEntry"): a print method

**complexes** signature(object = "complexEntry"): a method to generate and print all the complexes to the complexEntry class

**interactors** signature(object = "complexEntry"): a method to generate and print all the interactors for a complex in the complexEntry class

### Author(s)

Nianhua Li

### References

The IntAct Database: <http://www.ebi.ac.uk/intact/site/contents/downloads.jsf>.

### See Also

[intactComplex-class](#), [psi25complex](#)

---

`complexes`*A function that obtains the complex information from the IntAct Curated Protein Complexes*

---

**Description**

The `complexes` function takes a `complexEntry` class and returns the proteins which were found to be curated in some protein complex.

**Usage**

```
complexes(x)
```

**Arguments**

`x` A `complexEntry` class.

**Value**

A character vectors. The name of the proteins which were curated in some protein complex.

**Author(s)**

T Chiang

**References**

The IntAct Database <http://www.ebi.ac.uk/intact/site/contents/downloads.jsf>.

**See Also**

[complexEntry-class](#)

**Examples**

```
# parse complex data
```

---

`eListHandler`*xmlEventParse handlers for MIF XML documents from EBI intact*

---

**Description**

`xmlEventParse` handlers for MIF XML documents from EBI intact

**Usage**

```
eListHandler()  
iListHandler()
```

**Details**

A dump() method is supplied to deliver the list.

**Value**

list; see examples for structure

**Author(s)**

Vince Carey <stvjc@channing.harvard.edu>

**Examples**

```
fn <- system.file("MIFXML/ebi188.xml", package="Rintact")
xmlEventParse(fn, eListHandler())$dump()
xmlEventParse(fn, iListHandler())$dump()[1:5]
```

---

```
intactComplex-class
      Class "intactComplex"
```

---

**Description**

A class representing complex data.

**Objects from the Class**

Objects are usually created by `psi25complex`.

**Slots**

**intactId:** character, identifier of the complex in IntAct.

**shortLabel:** character, short label of the complex.

**fullName:** character, full name of the complex.

**organismName:** character, the unique organism names of the interactors in this complex.

**taxId:** character, the unique NCBI taxonomy ID(s) of the interactors in this complex.

**interactors:** data.frame. The data.frame contains two columns. The first column is the UniProt ID(s) of the interactors, the second column is the interactor's multiplicity.

**attributes:** named character. The name is the attribute name, the value is the attribute value.

Attribute names include: `curated-complex`, `complex-synonym`, `kd`, `complex-properties`, `disease`.

**Methods**

**show** signature(object = "intactComplex"): a print method

**Author(s)**

Nianhua Li

**See Also**

[psi25complex](#)

---

intactGraph-class *Class "intactGraph"*

---

**Description**

A class representing the IntAct data as a graph. It is a superclass of the graph class.

**Objects from the Class**

Objects are usually created by `intactXML2Graph`.

**Slots**

**interactors:** matrix, Each row represents one interactor. IntAct IDs are used as row names. Each column represents one annotation. Annotations include: UniProt ID, gene symbol, full name, locus name, ORF name, organism name, and NCBI taxonomy ID. Only those interactors which are the nodes of the `intactGraph` is given.

**Methods**

**show** signature(object = "intactGraph"): a print method

**translateIntactID** signature(object = "intactGraph"): a method to translate IntAct codes to the various annotations.

**translateNodeID** signature(object = "intactGraph"): a method that takes an `intactGraph` and translates each node ID from the IntAct accession IDs to any other supported identifier.

**Author(s)**

T Chiang

**See Also**

[intactXML2Graph](#)

---

intactHyperGraph-class  
*Class "intactHyperGraph"*

---

### Description

A class representing the IntAct data as a hypergraph. It is a superclass of the hypergraph class.

### Objects from the Class

Objects are usually created by `intactXML2Graph`.

### Slots

**interactors:** matrix, Each row represents one interactor. IntAct IDs are used as row names. Each column represents one annotation. Annotations include: UniProt ID, gene symbol, full name, locus name, ORF name, organism name, and NCBI taxonomy ID. Only those interactors which are the nodes of the `intactGraph` is given.

### Methods

**show** `signature(object = "intactHyperGraph")`: a print method

**initialize** a method to initialize an instance of an `intactHyperGraph`

**translateIntactID** `signature(object = "intactHyperGraph")`: a method to translate IntAct codes to the various annotations.

### Author(s)

T Chiang

### See Also

[intactXML2Graph](#)

---

intactInteraction-class  
*Class "intactInteraction"*

---

### Description

A class representing interaction data.

### Objects from the Class

Objects are usually created by `psi25interaction`.

## Slots

- intact:** character, IntAct ID of the interaction.
- interactionType:** character, the method used for detecting the interaction, such as pull down.
- expPubMed:** character, PubMed ID of the publication that describes the experiment.
- expIntAct:** character, IntAct ID of the experiment.
- confidenceValue:** character, confidence value of the experimental interaction.
- bait:** character, UniProt ID of the bait.
- prey:** character, UniProt ID(s) of the prey(s).
- inhibitor:** character, UniProt ID of the inhibitor, NA is missing.
- neutralComponent:** character, UniProt ID of the neutral components, NA is missing.

## Methods

- show** signature(object = "intactInteraction"): a print method
- bait** signature(object = "intactInteraction"): a method to find all the baits within the intactInteraction class and print them
- prey** signature(object = "intactInteraction"): a method to find all the prey within the intactInteraction class and print them

## Author(s)

Nianhua Li

## See Also

[psi25interaction](#)

---

intactXML2Graph     *A function that takes a vector of IntAct XML file names and converts them to graph objects.*

---

## Description

The `intactXML2Graph` takes in a vector of related XML files and generates a graph object based on the type of XML file. In general, the entry of this function should be that data from one experiment (this data might be divided into several XML files). If the XML file contains data that should not be combined into one single graph, this function should not be called; the function [separateXMLDataByExpt](#) should be used instead.

## Usage

```
intactXML2Graph(intactFiles, type="interaction", directed=TRUE)
```

**Arguments**

<code>intactFiles</code>	A vector of IntAct XML file names. If the data of a single experiment has been split into multiple XML files (e.g. Ewing et al 2007), then each entry of the vector should contain one of these XML files. NB - different datasets should not be put into the same vector.
<code>type</code>	A character string which is either "interaction" if the XML file contains experimental physical data or "complex" if the XML file contains curated protein complex membership data.
<code>directed</code>	A logical. User defined parameter so that the return value with either be a directed or undirected graph object.

**Value**

If `type = "interaction,"` then a resulting `intactGraph` object is generated on the aggregate of the XML files. If `type = "complex,"` then a resulting `intactHyperGraph` object is generated on the aggregate of the XML files.

**Author(s)**

T Chiang

**References**

Rintact: enabling computational analysis of molecular interaction data from the IntAct repository - Tony Chiang; Nianhua Li; Sandra Orchard; Samuel Kerrien; Henning Hermjakob; Robert Gentleman; Wolfgang Huber *Bioinformatics* 2007; doi: 10.1093/bioinformatics/btm518.

**See Also**

[interactionEntry-class](#), [complexEntry-class](#), [separateXMLDataByExpt](#)

---

`interactionEntry-class`

*Class "interactionEntry"*

---

**Description**

A class representing the interaction data extracted from an IntAct data file in PSI-MI 2.5 format.

**Objects from the Class**

Objects are usually created by `psi25interaction`.

**Slots**

**releaseDate:** character, release date of the data entry, recorded in the `entry` element of the XML file.

**organismName:** character, the unique organism name(s) of the interactors.

**taxId:** character, the unique NCBI taxonomy ID(s) of the interactors.



**interactors:** matrix, Each row represents one interactor. IntAct IDs are used as row names. Each column represents one annotation. Annotations include: UniProt ID, gene symbol, full name, locus name, ORF name, organism name, and NCBI taxonomy ID.

**interactions:** list, a list of `intactInteraction` objects, each represents one interaction.

## Methods

**show** signature(object = "interactionEntry"): a print method

**bait** signature(object = "interactionEntry"): a method to find all the baits within the interactionEntry and print them

**prey** signature(object = "interactionEntry"): a method to find all the prey within the interactionEntry and print them

**interactions** signature(object = "interactionEntry"): a method to find all the interactions within the interactionEntry and print them

**interactors** signature(object = "interactionEntry"): a method to find all the interactors within the interactionEntry and print them

**organismName** signature(object = "interactionEntry"): a method to print all the unique organism names that are of the interactors

**taxId** signature(object = "interactionEntry"): a method to print each taxonomy id for each organism

**releaseDate** signature(object = "interactionEntry"): a method to print the updated release date of the PSI-MI 2.5 XML file; it serves as a proxy for versioning.

## Author(s)

Nianhua Li

## References

The IntAct Database: <http://www.ebi.ac.uk/intact/site/contents/downloads.jsf>.

## See Also

[intactInteraction-class](#), [psi25interaction](#)

---

interactions

*A function that obtains the interaction information from the IntAct repository.*

---

## Description

The `interactions` function takes a `interactionEntry` class and returns all the interactions parsed from the PSI-25MI file upon which the `interactionEntry` class is based.

## Usage

```
interactions(x)
```

**Arguments**

x                    An interactionEntry class.

**Value**

A list of intactInteraction objects. The intactInteraction objects contain information about the experiment, the interaction type, the author's confident value, and the bait and prey proteins.

**Author(s)**

T Chiang

**References**

The IntAct Database <http://www.ebi.ac.uk/intact/site/contents/downloads.jsf>.

**See Also**

[interactionEntry-class](#)

**Examples**

```
# parse complex data
```

---

interactors	<i>A function that obtains the interactors either in a binary interaction or a complex co-membership interaction.</i>
-------------	---

---

**Description**

The `interactors` function either takes a `complexEntry` class or an `interactionEntry` class and returns the proteins which were found to be the interactors of the interaction or the constituent members of the protein complex.

**Usage**

```
interactors(x)
```

**Arguments**

x                    A `complexEntry` or `interactionEntry` class.

**Value**

A character matrix. The name of the proteins which were shown to have been involved in the interaction or protein complex dataset represented by either the `complexEntry` or the `interactionEntry`.

**Author(s)**

T Chiang

## References

The IntAct Database <http://www.ebi.ac.uk/intact/site/contents/downloads.jsf>.

## See Also

[complexEntry-class](#)

## Examples

```
# parse complex data
```

---

list2Matrix	<i>A function that takes a named list representation of a bipartite graph and transforms it into a matrix representation.</i>
-------------	---

---

## Description

The `list2Matrix` takes in a named list (representing a bipartite graph) and transforms into into a (0,1)-matrix representation. The names of the list index the columns and the union of the elements of the list index the row. The (i,j) position of the matrix is 1 if element i is in the list element indexed by name j, and it is 0 otherwise.

## Usage

```
list2Matrix(namedList, type="interaction")
```

## Arguments

<code>namedList</code>	A named list. The list represents a bipartite graph.
<code>type</code>	If type is "interaction", the return value is a bait-prey adjacency matrix with baits indexing the columns and prey indexing rows. If type is "complex", the return value is an incidence matrix with complexes indexing the columns and proteins indexing the rows

## Value

A (0,1)-matrix representation of the bipartite graph.

## Author(s)

T Chiang

## Examples

```
eg = list(first = letters[1:5], second = letters[4:8])
list2Matrix(eg)
```

---

organismName	<i>A method to obtain all the unique organism names from the PSI-MI 2.5 XML files upon which the data is based</i>
--------------	--

---

**Description**

The `organismName` function takes an `intactInteraction` class and returns the unique organism names upon which the data is based.

**Usage**

```
organismName(x)
```

**Arguments**

`x`                    A S4 object of type `intactInteraction`

**Value**

A character vector of the unique organisms

**Author(s)**

T Chiang

**References**

The IntAct Database <http://www.ebi.ac.uk/intact/site/contents/downloads.jsf>.

**See Also**

`\intactInteraction-class`

**Examples**

```
# parse complex data
```

---

prey	<i>A function that obtains the prey protein from the IntAct Interactions</i>
------	--

---

**Description**

The `prey` function takes an `intactInteraction` class and returns the proteins which was sampled as a prey in the binary relationship.

**Usage**

```
prey(x)
```

**Arguments**

x                    An intactInteraction class.

**Value**

A character. The name of the protein which was sampled as a prey in the tested binary interaction.

**Author(s)**

T Chiang

**References**

The IntAct Database <http://www.ebi.ac.uk/intact/site/contents/downloads.jsf>.

**See Also**

[intactInteraction-class](#)

**Examples**

```
# parse complex data
```

---

psi25interaction    *parser for PSI-MI 2.5 XML documents from EBI intact*

---

**Description**

The PSI-MI 2.5 XML format is used by EBI IntAct database to record protein-protein interaction data and protein complex data. `psi25interaction` is the parser for interaction data and `psi25complex` is the parser for complex data.

**Usage**

```
psi25interaction(intactFile)
psi25complex(intactFile)
```

**Arguments**

intactFile    character, file name or URL of the XML document

**Value**

`psi25interaction` returns a list of `interactionEntry` objects, each represents one entry in the XML document. `psi25complex` returns an `complexEntry` object, which represents the data from one XML document.

**Author(s)**

Nianhua Li

## References

The IntAct Database <http://www.ebi.ac.uk/intact/site/contents/downloads.jsf>.

## See Also

[interactionEntry-class](#), [complexEntry-class](#)

## Examples

```
# parse complex data
fn <- system.file("PSI25XML/complexSample.xml", package="Rintact")
complexData <- psi25complex(fn)
complexData

# parse interaction data
fn <- system.file("PSI25XML/interactionSample.xml", package="Rintact")
interactionData <- psi25interaction(fn)
interactionData
```

---

releaseDate

*A method to return the release date of the PSI-MI 2.5 XML file*

---

## Description

The `releaseDate` function takes an `intactInteraction` class and returns the data when the XML file is repacked and released from IntAct. This serves as a proxy for versioning.

## Usage

```
releaseDate(x)
```

## Arguments

`x` A S4 object of type `intactInteraction`

## Value

A character. The release date.

## Author(s)

T Chiang

## References

The IntAct Database <http://www.ebi.ac.uk/intact/site/contents/downloads.jsf>.

## See Also

[\intactInteraction-class](#)

## Examples

```
# parse complex data
```

---

```
separateXMLDataByExpt
```

*A function that takes a vector of IntAct XML file name locations and converts them to graph objects.*

---

## Description

The `separateXMLDataByExpt` function takes in a vector of XML files and generates a list of graph objects dividing the data by separately conducted experiments. If a single XML file contains  $k$  experimental datasets, then  $k$  `intactGraph` objects will be produced.

## Usage

```
separateXMLDataByExpt(intactFiles, type="direct", directed=TRUE)
```

## Arguments

<code>intactFiles</code>	A vector of IntAct XML file name locations, i.e. either where they can be found on the local hard drive or an url to where they can be obtained
<code>type</code>	A character. Either "direct" or "indirect" based on the information that needs to be parsed.
<code>directed</code>	A logical. User defined parameter so that the return value with either be a directed or undirected graph object.

## Value

A list of `intactGraph` objects.

## Author(s)

T Chiang

## References

Rintact: enabling computational analysis of molecular interaction data from the IntAct repository - Tony Chiang; Nianhua Li; Sandra Orchard; Samuel Kerrien; Henning Hermjakob; Robert Gentleman; Wolfgang Huber *Bioinformatics* 2007; doi: 10.1093/bioinformatics/btm518.

## See Also

[interactionEntry-class](#), [complexEntry-class](#), [intactXML2Graph](#)

---

taxId	<i>A method to return all the unique taxonomy IDs indexing the unique organisms upon which the data is based</i>
-------	--

---

**Description**

The `taxId` function takes an `intactInteraction` class and returns the unique `taxId` indexing the organism names upon which the data is based.

**Usage**

```
taxId(x)
```

**Arguments**

`x`                    A S4 object of type `intactInteraction`

**Value**

A character vector of the unique taxonomy Ids

**Author(s)**

T Chiang

**References**

The IntAct Database <http://www.ebi.ac.uk/intact/site/contents/downloads.jsf>.

**See Also**

`\intactInteraction-class`

**Examples**

```
# parse complex data
```

---

translateIntactID	<i>A function that translates the IntAct accession codes for molecules to other common identifiers.</i>
-------------------	---

---

**Description**

The `translateIntactID` takes 3 parameters, either an `intactGraph` or `intactHyperGraph` S-4 Class, a vector of IntAct accession codes, and a vector of other identifiers. Then for the particular graph object, `translateIntactID` will map those IntAct accession codes to the other identifiers if possible.



**Usage**

```
translateIntactID(r, ...)
```

**Arguments**

`r` Either an `intactGraph` or `intactHyperGraph` S-4 Class.

`...` The primary use of this generic are as methods for the `intactGraph` S-4 class and the `intactHyperGraph` S-4 class. In both two more parameters need for the method. The first is "ebiNames" which is a character vector of the IntAct accession codes to be mapped. The second is "toWhat" which is a character vector of identifier names. These identifiers are limited to the following: "uniprotId", "geneName", "fullName", "locusName", and "orfName"

**Value**

A character matrix. The rows are indexed by the IntAct accession codes supplied and the columns are indexed by the identifiers supplied.

**Author(s)**

T Chiang

**References**

Rintact: enabling computational analysis of molecular interaction data from the IntAct repository - Tony Chiang; Nianhua Li; Sandra Orchard; Samuel Kerrien; Henning Hermjakob; Robert Gentleman; Wolfgang Huber *Bioinformatics* 2007; doi: 10.1093/bioinformatics/btm518.

**See Also**

[intactGraph-class](#), [intactHyperGraph-class](#)

**Examples**

```
# parse complex data
```

---

`translateNodeID` *A function that translates the IntAct accession IDs in an intactGraph to other common identifiers.*

---

**Description**

The `translateNodeID` takes 2 parameters, an `intactGraph` and the new ID to which the IntAct accession IDs should be translated. Then for the particular graph object, `translateNodeID` will map those IntAct accession codes to the other identifiers if possible and re-map the node names.

**Usage**

```
translateNodeID(g, ...)
```

**Arguments**

<code>g</code>	An intactGraph S-4 Class.
<code>...</code>	The primary use of this generic are as methods for the in intactGraph S-4 class and the intactHyperGraph S-4 class. In both an extra parameter, <code>newID</code> , is needed to detail to what the IntAct accession ID should be mapped. These identifiers are limited to the following: "uniprotId", "geneName", "fullName", "locusName", and "orfName"

**Value**

An `intactGraph S4`-class with the nodes re-mapped to the user supplied identifiers.

**Author(s)**

T Chiang

**References**

Rintact: enabling computational analysis of molecular interaction data from the IntAct repository - Tony Chiang; Nianhua Li; Sandra Orchard; Samuel Kerrien; Henning Hermjakob; Robert Gentleman; Wolfgang Huber *Bioinformatics* 2007; doi: 10.1093/bioinformatics/btm518.

**See Also**

[intactGraph-class](#), [intactHyperGraph-class](#)

**Examples**

```
# parse complex data
```

# Index

## \*Topic **models**

- bait, 1
  - complexEntry-class, 2
  - complexes, 3
  - eListHandler, 3
  - intactComplex-class, 4
  - intactGraph-class, 5
  - intactHyperGraph-class, 6
  - intactInteraction-class, 6
  - intactXML2Graph, 7
  - interactionEntry-class, 8
  - interactions, 9
  - interactors, 10
  - list2Matrix, 11
  - organismName, 12
  - prey, 12
  - psi25interaction, 13
  - releaseDate, 14
  - separateXMLDataByExpt, 15
  - taxId, 16
  - translateIntactID, 16
  - translateNodeID, 17
- 
- bait, 1
  - bait, intactInteraction-method  
(*intactInteraction-class*),  
6
  - bait, interactionEntry-method  
(*interactionEntry-class*), 8
  
  - complexEntry-class, 3, 8, 11, 14, 15
  - complexEntry-class, 2
  - complexes, 3
  - complexes, complexEntry-method  
(*complexEntry-class*), 2
  
  - eListHandler, 3
  
  - iListHandler (*eListHandler*), 3
  - initialize, intactHyperGraph-method  
(*intactHyperGraph-class*), 6
  - intactComplex-class, 2
  - intactComplex-class, 4
  - intactGraph-class, 17, 18
  
  - intactGraph-class, 5
  - intactHyperGraph-class, 17, 18
  - intactHyperGraph-class, 6
  - intactInteraction-class, 1, 9, 13
  - intactInteraction-class, 6
  - intactXML2Graph, 5, 6, 7, 15
  - interactionEntry-class, 8, 10, 14, 15
  - interactionEntry-class, 8
  - interactions, 9
  - interactions, interactionEntry-method  
(*interactionEntry-class*), 8
  - interactors, 10
  - interactors, complexEntry-method  
(*complexEntry-class*), 2
  - interactors, interactionEntry-method  
(*interactionEntry-class*), 8
  
  - list2Matrix, 11
  
  - organismName, 12
  - organismName, interactionEntry-method  
(*interactionEntry-class*), 8
  
  - prey, 12
  - prey, intactInteraction-method  
(*intactInteraction-class*),  
6
  - prey, interactionEntry-method  
(*interactionEntry-class*), 8
  - psi25complex, 2, 5
  - psi25complex (*psi25interaction*),  
13
  - psi25interaction, 7, 9, 13
  
  - releaseDate, 14
  - releaseDate, interactionEntry-method  
(*interactionEntry-class*), 8
  
  - separateXMLDataByExpt, 7, 8, 15
  - show, complexEntry-method  
(*complexEntry-class*), 2
  - show, intactComplex-method  
(*intactComplex-class*), 4
  - show, intactGraph-method  
(*intactGraph-class*), 5

show, intactHyperGraph-method  
    (*intactHyperGraph-class*), 6

show, intactInteraction-method  
    (*intactInteraction-class*),  
    6

show, interactionEntry-method  
    (*interactionEntry-class*), 8

taxId, 16

taxId, interactionEntry-method  
    (*interactionEntry-class*), 8

translateIntactID, 16

translateIntactID, intactGraph-method  
    (*intactGraph-class*), 5

translateIntactID, intactHyperGraph-method  
    (*intactHyperGraph-class*), 6

translateNodeID, 17

translateNodeID,  
    intactGraph-method  
    (*intactGraph-class*), 5

translateNodeID, intactGraph-method  
    (*translateNodeID*), 17