# Orange3 Data Fusion Documentation

***Release***

**Biolab**

**Jan 08, 2018**

# Widgets

CHAPTER 1

# IMDb Actors

Constructs a movies-by-actors or actors-by-actors relation matrix.

## 1.1 Signals

**Inputs**:

- **Filter**

  Data filter.

**Outputs**:

- **Movie Actors**

  A movies-by-actors relation matrix.

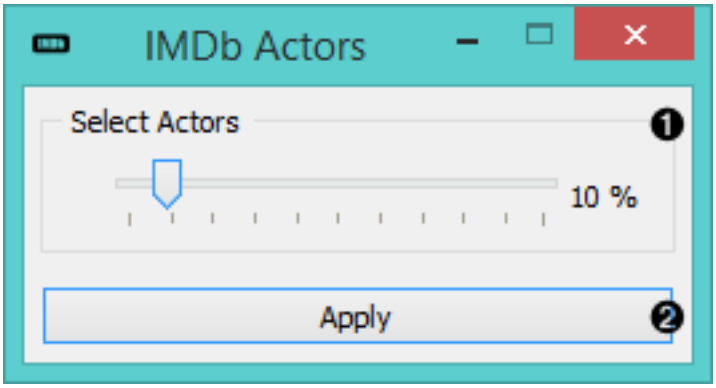
- **Costarring Actors**

  An actors-by-actors relation matrix.

## 1.2 Description

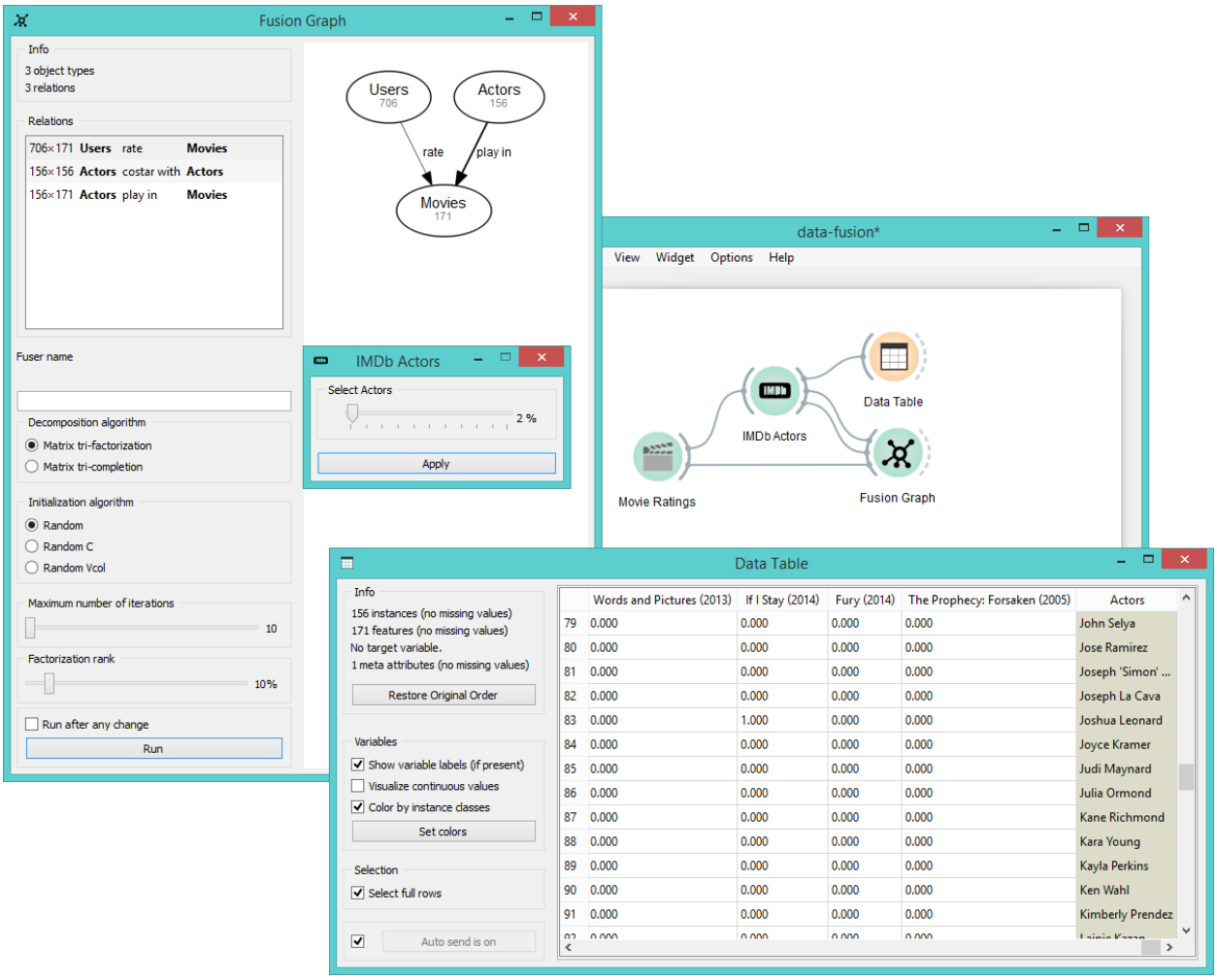
This widget gives you the access to the IMDb data sets on actors and movies. It outputs either a movies-by-actors relation matrix, an actors-by-actors relation matrix or both.

1. Select how many actors from the IMDb database would you like to consider.
2. Click *Apply* to commit your data.




## 1.3 Example

This simple widget is great for learning how data fusion works since it enables immediate access to the IMDb database. To use it, you need to connect it to **Movie Ratings** widget in the input and with **Fusion Graph** in the output. This will add the information on actors in relation to movies. You can view this new data in the **Data Table** widget.


Fusion Graph
Info
3 object types
3 relations
Relations
706×171 Users rate Movies
156×156 Actors costar with Actors
156×171 Actors play in Movies
Users
706
Actors
156
rate
play in
Movies
171
Fuser name
Decomposition algorithm
Matrix tri-factorization
Matrix tri-completion
Initialization algorithm
Random
Random C
Random Vcol
Maximum number of iterations
10
Factorization rank
10%
Run after any change
Run
IMDb Actors
Select Actors
2 %
Apply
data-fusion*
View
Widget
Options
Help
Data Table
IMDb Actors
Movie Ratings
Fusion Graph
Data Table
Info
156 instances (no missing values)
171 features (no missing values)
No target variable.
1 meta attributes (no missing values)
Restore Original Order
Variables
Show variable labels (if present)
Visualize continuous values
Color by instance classes
Set colors
Selection
Select full rows
Auto send is on
Words and Pictures (2013)
If I Stay (2014)
Fury (2014)
The Prophecy: Forsaken (2005)
Actors
79 0.000 0.000 0.000 0.000 John Selya
80 0.000 0.000 0.000 0.000 Jose Ramirez
81 0.000 0.000 0.000 0.000 Joseph 'Simon' ...
82 0.000 0.000 0.000 0.000 Joseph La Cava
83 0.000 1.000 0.000 0.000 Joshua Leonard
84 0.000 0.000 0.000 0.000 Joyce Kramer
85 0.000 0.000 0.000 0.000 Judi Maynard
86 0.000 0.000 0.000 0.000 Julia Ormond
87 0.000 0.000 0.000 0.000 Kane Richmond
88 0.000 0.000 0.000 0.000 Kara Young
89 0.000 0.000 0.000 0.000 Kayla Perkins
90 0.000 0.000 0.000 0.000 Ken Wahl
91 0.000 0.000 0.000 0.000 Kimberly Prendez

CHAPTER 2

# Chaining

Profiles objects of one type in the latent space of another object type through chaining of latent matrices along paths in a data fusion graph.

## 2.1 Signals

**Inputs**:

- **Fitted Fusion Graph**

  Fitted collective latent data model.

**Outputs**:

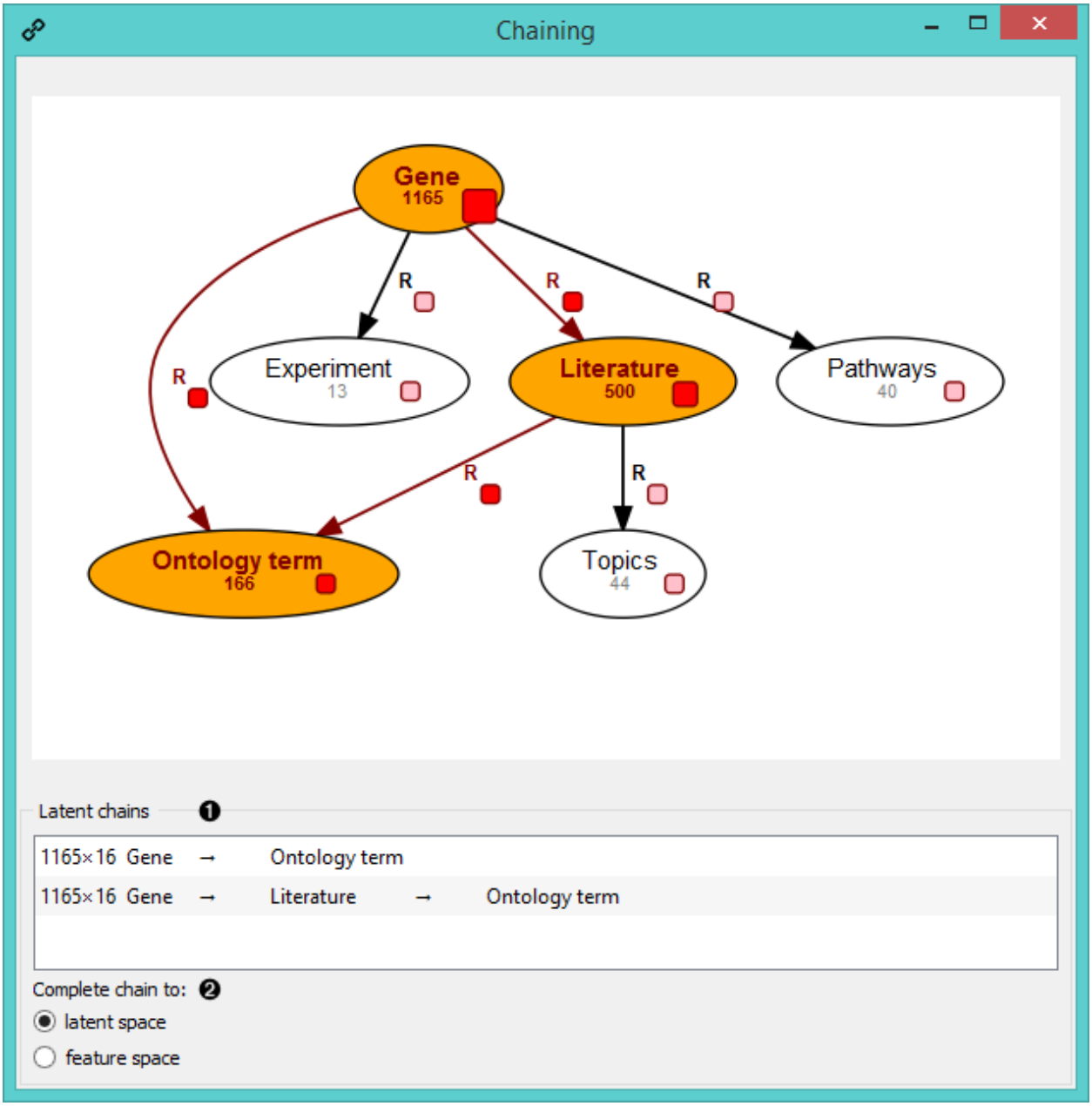
- **Relation**

  Relationships between two groups of objects.

## 2.2 Description

**Chaining** constructs data profiles of objects of one type that are expressed in the latent space of another object type. This is done by appropriately multiplying the latent matrices along paths that connect start and end nodes in the fusion graph. The widget displays a fitted fusion graph on the right, where you can select the start and end node (object type) that are then used in chaining.

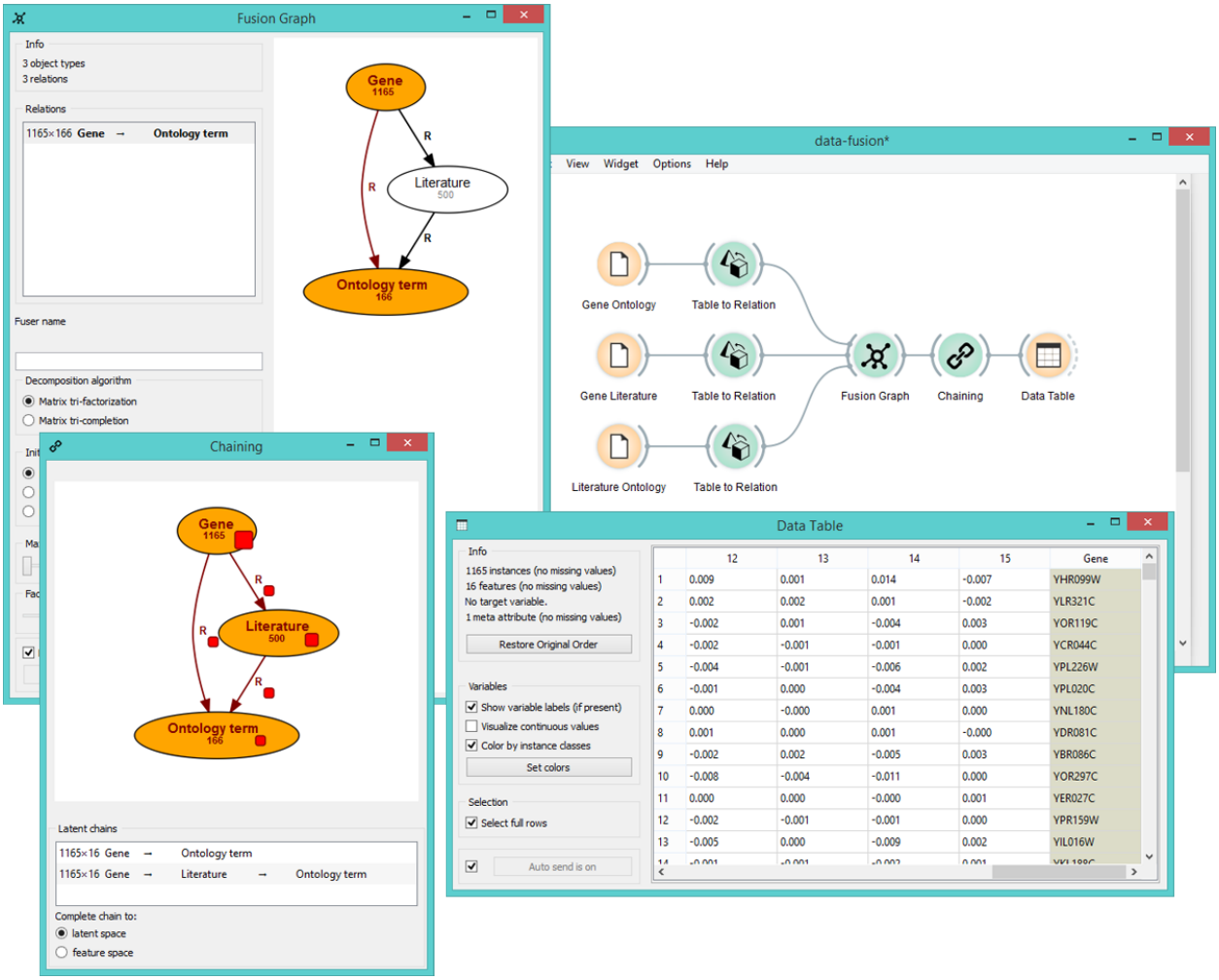
1. The widget displays all chains that connect selected start node with the selected end node (in orange). Click on the chain you wish to output.
2. Select what type of chain you wish to output:


Chaining
Gene
1165
R
R
R
R
Experiment
13
Literature
500
Pathways
40
R
R
R
Ontology term
166
Topics
44
Latent chains
1165×16 Gene → Ontology term
1165×16 Gene → Literature → Ontology term
Complete chain to:
latent space
feature space

- **latent space** (widget outputs data profiles in the latent space)
- **feature space** (widget outputs data profiles in the original domain space)

## 2.3 Example

This widget is great for constructing profiles that relate objects, which are not directly connected in a fusion graph. In the example below we have three data sets: annotations of genes from the Gene Ontology, literature on genes and literature on ontology terms. We use **Chaining** to see how genes relate to ontology terms.



CHAPTER 3

# Completion Scoring

Scores the quality of matrix completion using root mean squared error (RSME) metric.

## 3.1 Signals

**Inputs**:

- **Fitted fusion graph**

  Fitted collective latent data model.

- **Relation**

  Relationships between two groups of objects.

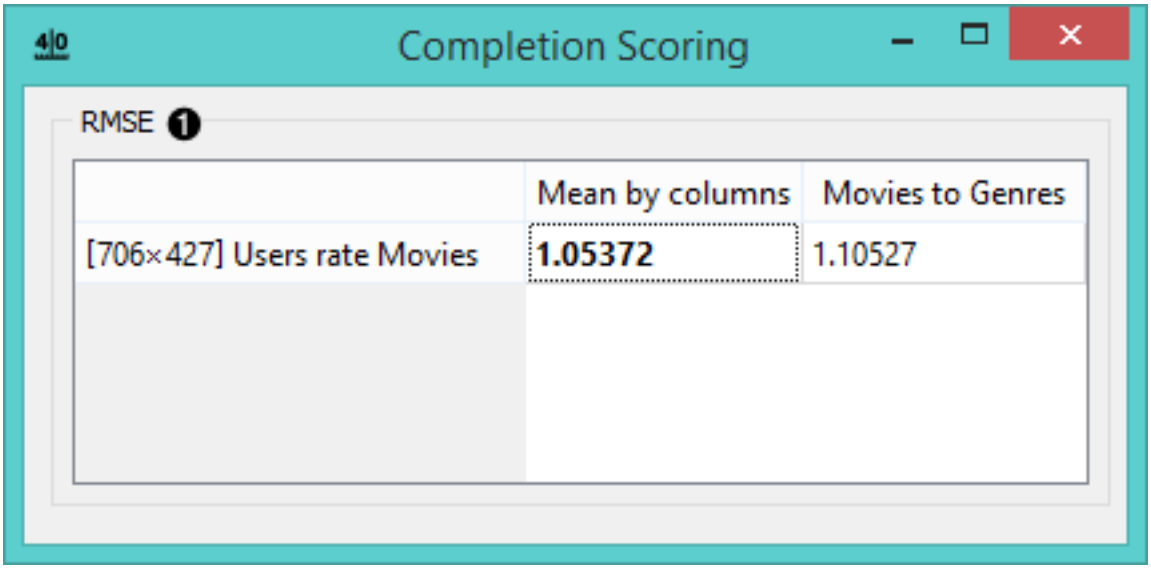
**Outputs**:

- (None)

## 3.2 Description

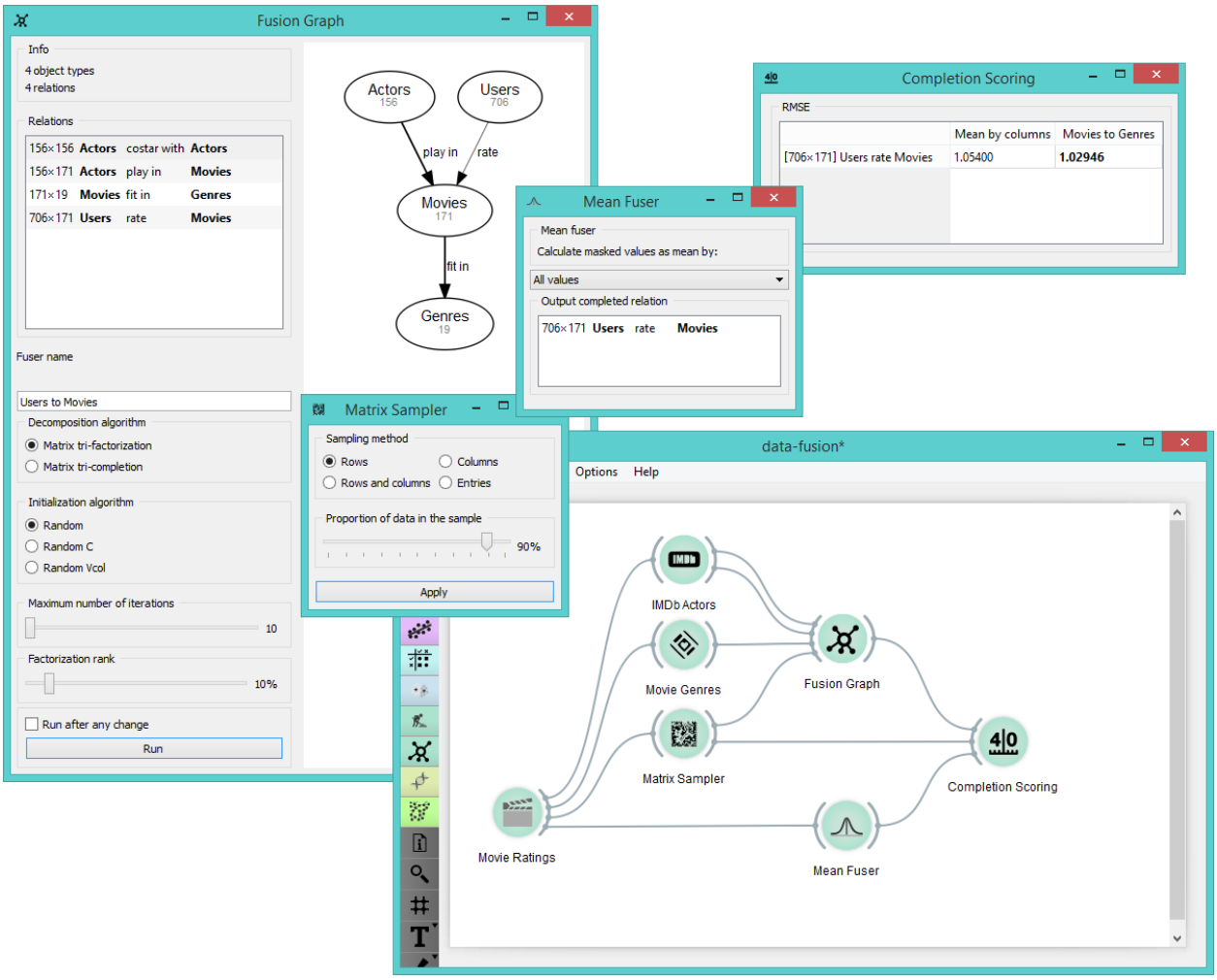
This widget assesses the quality of matrix completion based on root mean squared error metric (RMSE). Each row contains scores representing matrix completion quality of different relations. Results for prediction models are in columns.

1. The RMSE value chart for the input relation matrix.




## 3.3 Example

**Completion Scoring** widget assesses the quality of matrix completion using the RMSE metric. Connect it with **Matrix Sampler** to score prediction models (previously learnt on in-sample data) on out-of-the-sample data. You can also use **Mean Fuser** to get a mean score for latent values.


Fusion Graph
Info
4 object types
4 relations
Relations
156×156 Actors costar with Actors
156×171 Actors play in Movies
171×19 Movies fit in Genres
706×171 Users rate Movies
Fuser name
Users to Movies
Decomposition algorithm
Matrix tri-factorization
Matrix tri-completion
Initialization algorithm
Random
Random C
Random Vcol
Maximum number of iterations
10
Factorization rank
10%
Run after any change
Run
Actors
156
Users
706
play in
rate
Movies
171
fit in
Genres
19
Completion Scoring
RMSE
Mean by columns
Movies to Genres
[706×171] Users rate Movies
1.05400
1.02946
Mean Fuser
Mean fuser
Calculate masked values as mean by:
All values
Output completed relation
706×171 Users rate Movies
Matrix Sampler
Sampling method
Rows
Columns
Rows and columns
Entries
Proportion of data in the sample
90%
Apply
data-fusion*
Options
Help
IMDb Actors
Movie Genres
Matrix Sampler
Movie Ratings
Fusion Graph
Completion Scoring
Mean Fuser

CHAPTER 4

# Fusion Graph

Constructs a data fusion graph and runs collective matrix factorization algorithm.

## 4.1 Signals

**Inputs**:

- **Relation**

  Relationships between two groups of objects.

**Outputs**:

- **Relation**

  Relationships between two groups of objects.

- **Fitted Fusion Graph**

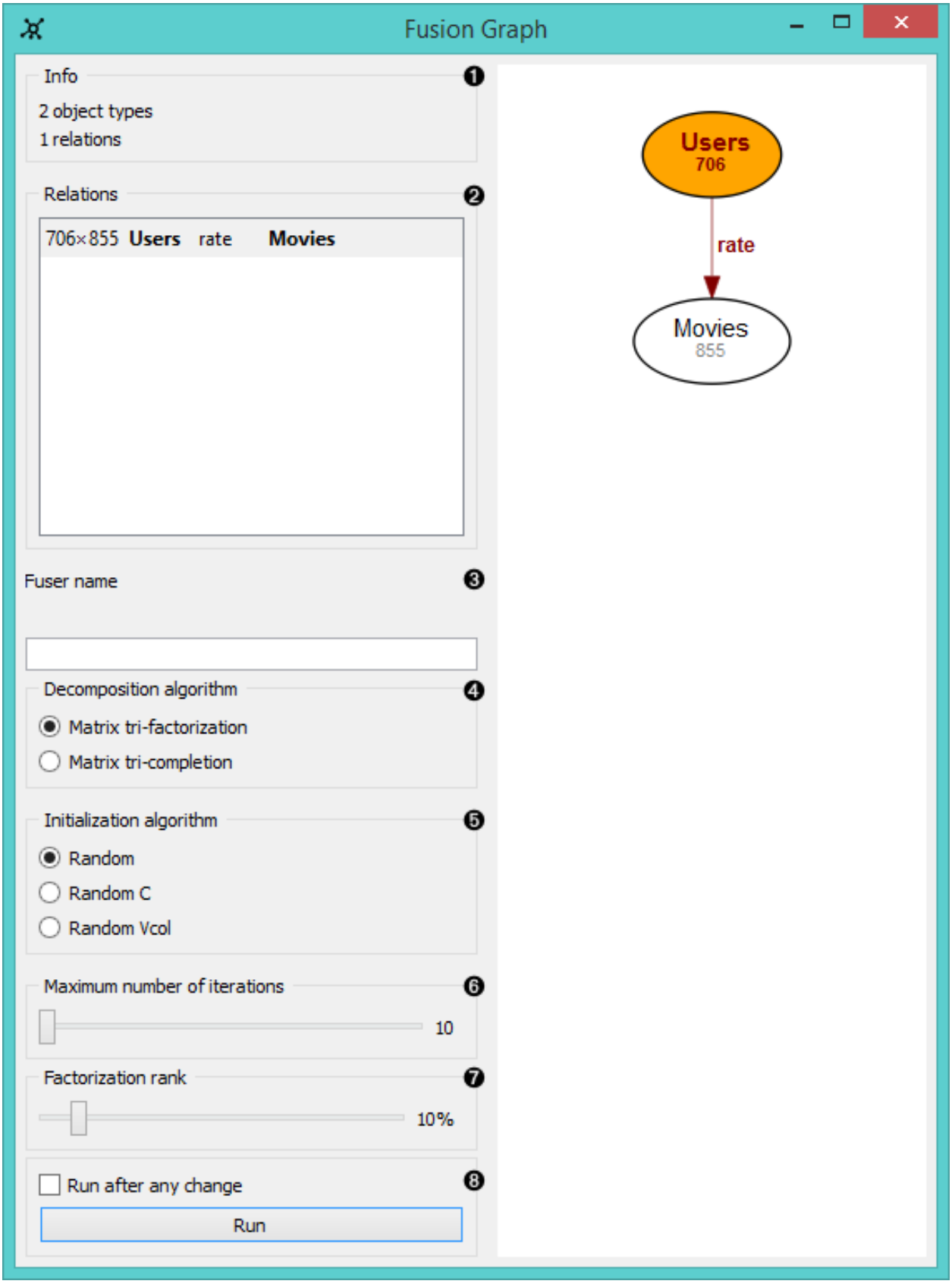
  Fitted collective latent data model.

- **Fusion Graph**

  Input data system.

## 4.2 Description

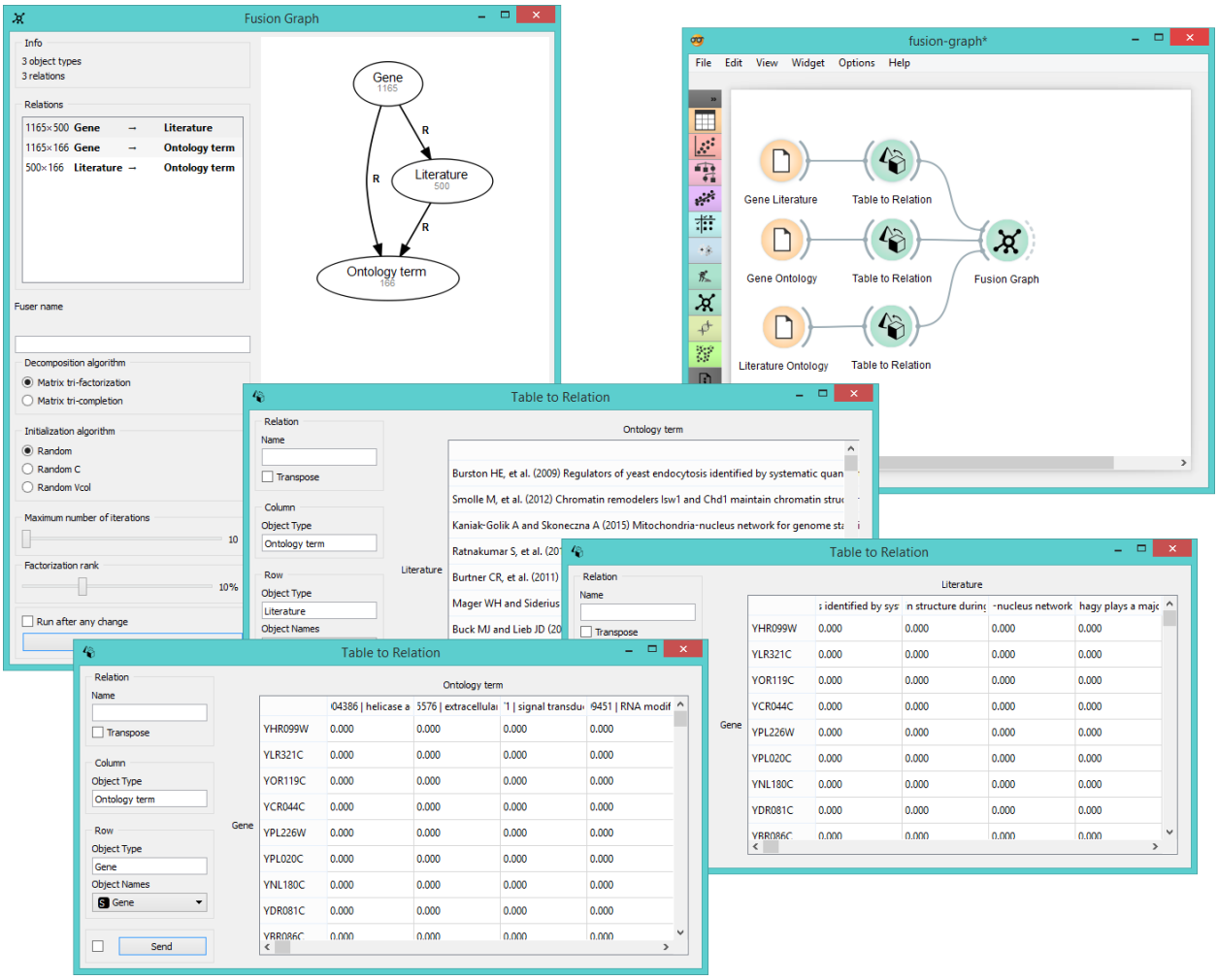
**Fusion Graph** widget performs data fusion by collective matrix factorization. It fuses multiple related data sets into one comprehensive structure. The widget returns a relational structure of the entire data system estimated by a collective latent factor approach.


Fusion Graph
Info
2 object types
1 relations
Relations
706×855 Users rate Movies
Fuser name
Decomposition algorithm
Matrix tri-factorization
Matrix tri-completion
Initialization algorithm
Random
Random C
Random Vcol
Maximum number of iterations
10
Factorization rank
10%
Run after any change
Run
Users
706
rate
Movies
855

1. Information on the input (object types are nodes, relations are links between the nodes).
2. List of identified relations. Click on the relation to output it.
3. Specify a descriptive name for your fusion system.
4. Select the algorithm for factorization:
    - **matrix tri-factorization** decomposes each relation matrix into three latent matrices and shares the latent matrices between related data sets. Unknown values are imputed prior to collective factorization.
    - **matrix tri-completion** works the same as matrix tri-factorization, but does not require relation matrices to be fully observed.
5. Select the *initialization algorithm* for matrix factorization.
6. Set the *maximum number of iterations* used for factorization. Default is 10.
7. Set the *factorization rank* (the ratio of data compression based on the input data). Default is 10%.
8. If *Run after every change* is ticked, the widget will automatically commit changes. Alternatively press *Run*. For large data sets we recommend to commit the changes manually.

## 4.3 Example

The example below shows how to fuse several data sets together. Say we have the data on ontology terms for many genes, literature on ontology terms and literature on genes. To fuse these data together we first use **Table to Relation** widget, where we manually set the object type and relation names. **Fusion Graph** will compile the fusion graph of our three data sets with connections between object types based on previously defined data relations, display the connections and run matrix decomposition algorithm.


Fusion Graph
Info
3 object types
3 relations
Relations
1165×500 Gene → Literature
1165×166 Gene → Ontology term
500×166 Literature → Ontology term
Fuser name
Decomposition algorithm
Matrix tri-factorization
Matrix tri-completion
Initialization algorithm
Random
Random C
Random Vcol
Maximum number of iterations
Factorization rank
Run after any change
Gene
1165
Literature
500
Ontology term
166
R
fusion-graph*
File Edit View Widget Options Help
Gene Literature
Table to Relation
Gene Ontology
Table to Relation
Literature Ontology
Table to Relation
Fusion Graph
Table to Relation
Relation
Name
Transpose
Column
Object Type
Ontology term
Row
Object Type
Literature
Object Names
Ontology term
Burston HE, et al. (2009) Regulators of yeast endocytosis identified by systematic quan
Smolle M, et al. (2012) Chromatin remodelers Isw1 and Chd1 maintain chromatin struc
Kaniak-Golik A and Skoneczna A (2015) Mitochondria-nucleus network for genome st
Ratnakumar S, et al. (20
Burtner CR, et al. (2011)
Mager WH and Siderius
Buck MJ and Lieb JD (20
Table to Relation
Relation
Name
Transpose
Literature
YHR099W 0.000 0.000 0.000 0.000
YLR321C 0.000 0.000 0.000 0.000
YOR119C 0.000 0.000 0.000 0.000
YCR044C 0.000 0.000 0.000 0.000
YPL226W 0.000 0.000 0.000 0.000
YPL020C 0.000 0.000 0.000 0.000
YNL180C 0.000 0.000 0.000 0.000
YDR081C 0.000 0.000 0.000 0.000
Table to Relation
Relation
Name
Transpose
Column
Object Type
Ontology term
Row
Object Type
Gene
Object Names
Gene
Send
Ontology term
Gene
YHR099W 0.000 0.000 0.000 0.000
YLR321C 0.000 0.000 0.000 0.000
YOR119C 0.000 0.000 0.000 0.000
YCR044C 0.000 0.000 0.000 0.000
YPL226W 0.000 0.000 0.000 0.000
YPL020C 0.000 0.000 0.000 0.000
YNL180C 0.000 0.000 0.000 0.000
YDR081C 0.000 0.000 0.000 0.000

CHAPTER 5

# Latent Factors

Draws data fusion graph with the estimated latent factors overlaid. Outputs latent factors for further analysis.

## 5.1 Signals

**Inputs**:

- **Fitted fusion graph**

  Fitted collective latent data model.

**Outputs**:

- **Relation**

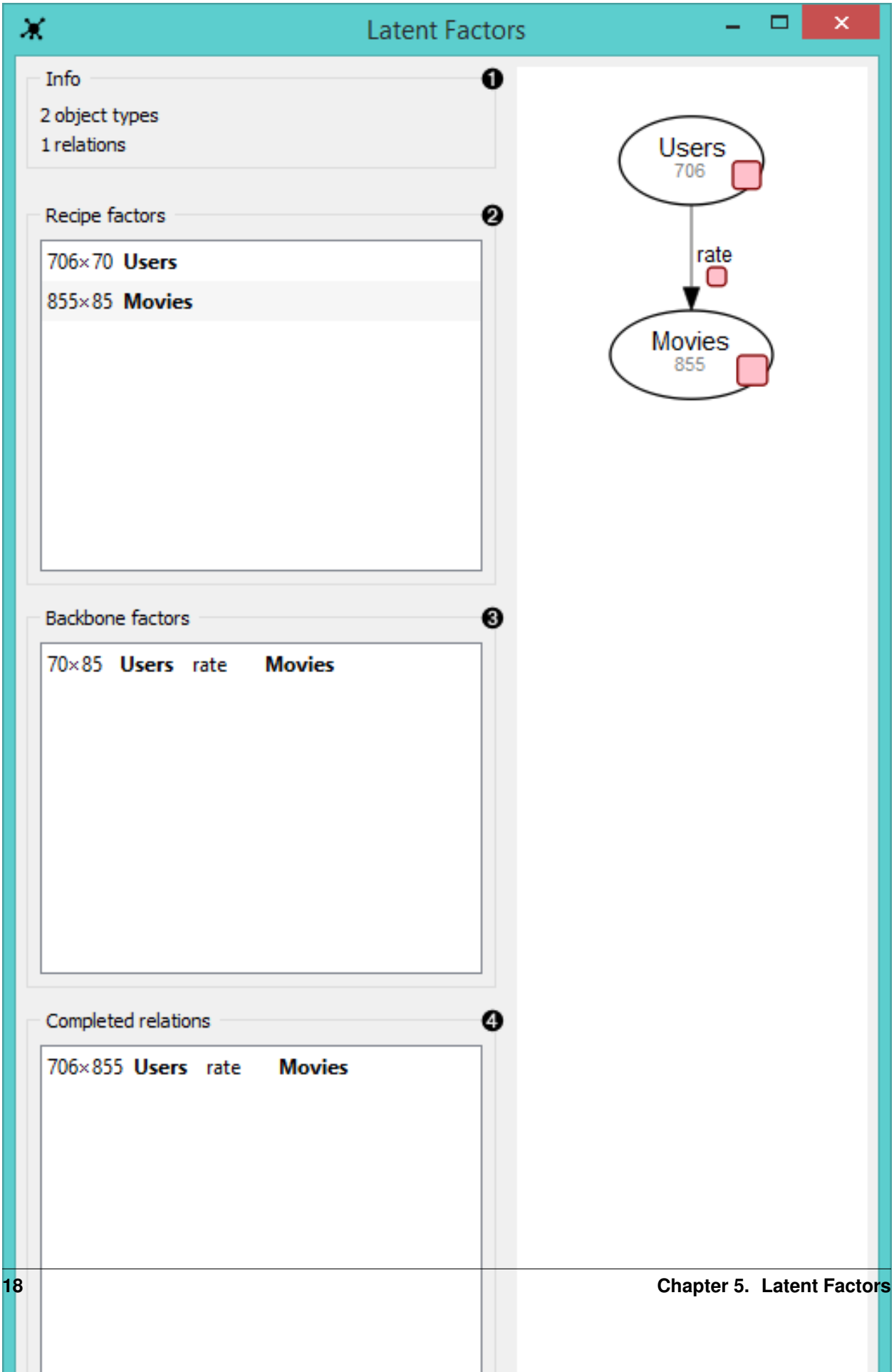
  Selected latent data matrix or a completed relation.

## 5.2 Description

**Latent Factors** widget displays the fusion graph together with the backbone and recipe matrices estimated by collective matrix factorization.

Fused data from the widget input are decomposed into latent factors, which serve as components for subsequent matrix reconstruction. You would normally draw this widget from **Fusion Graph** and feed its output (a backbone matrix, a recipe matrix or a completed relation) into widgets for downstream data analysis, such as **Hierarchial Clustering** or **Heat Map**.

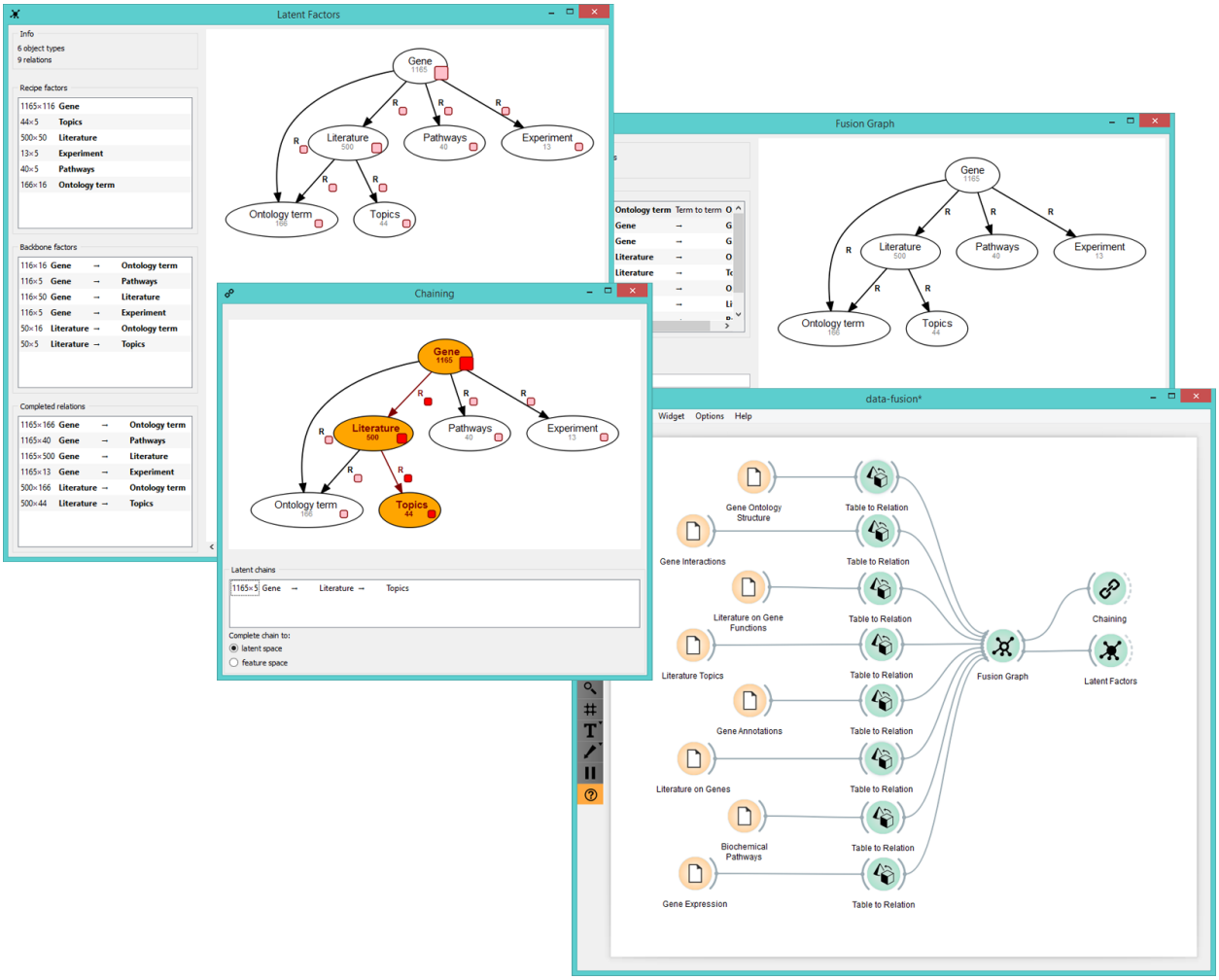
1. Information on the input (object types are nodes, data relations are links between the nodes).


Latent Factors
Info
2 object types
1 relations
Recipe factors
706×70 Users
855×85 Movies
Backbone factors
70×85 Users rate Movies
Completed relations
706×855 Users rate Movies
Users
706
rate
Movies
855

2. A list of **recipe factors** (latent matrices containing compressed representation of object types). Recipe factors encode latent components of respective object types.
3. A list of **backbone factors** (latent matrices containing compressed representation of data relations). Backbone factors encode interactions between the latent components.
4. A list of **completed relations** (completed relation matrices obtained by multiplying the corresponding latent matrices).

## 5.3 Example

In the example below we demonstrate how 8 separate yeast data sets are fused together in **Fusion Graph** and then decomposed into latent factors with **Latent Factors** widget.



CHAPTER 6

# Matrix Sampler

Samples a relation matrix.

## 6.1 Signals

**Inputs**:

- **Data**

Data set.

**Outputs**:

- **In-sample Data**

  Selected data.

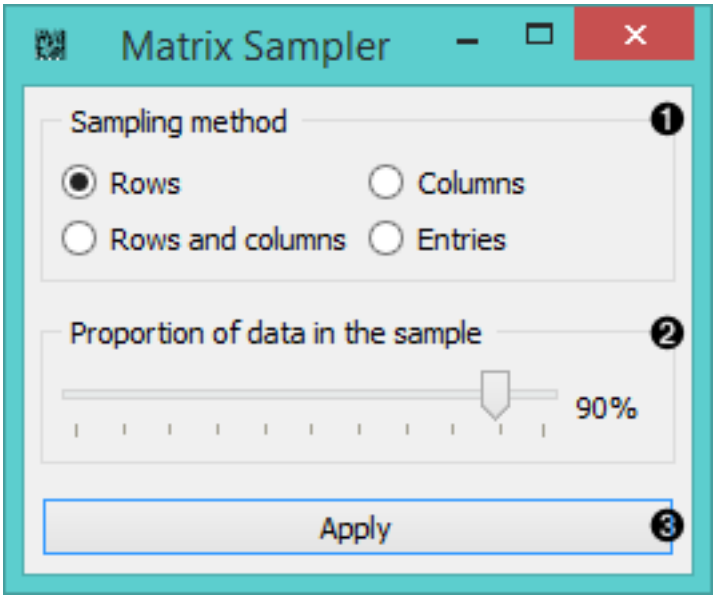
- **Out-of-the-sample Data**

  Remaining data.

## 6.2 Description

This widget samples the input data and sends both the sampled and the remaining data to the output. It is useful for evaluating the performance of recommendation systems.

1. Select the desired *sampling method*:
   - **rows** (randomly samples entire matrix rows)

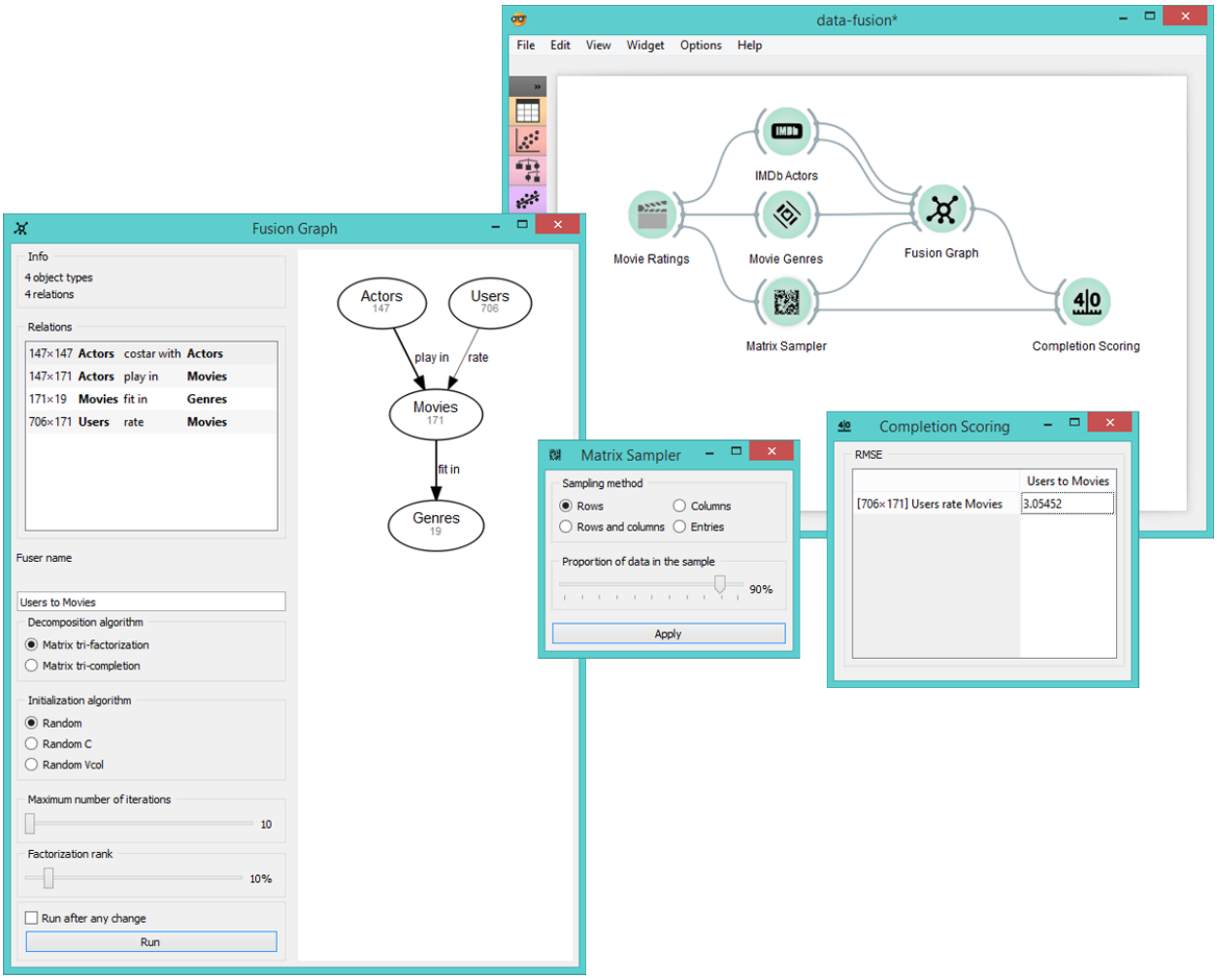



- **columns** (randomly samples entire matrix columns)
- **rows and columns** (samples from the entire matrix)
- **entries** (randomly samples individual matrix elements)

2. Select the proportion of the data you want at the output.
3. Press **Apply** to commit the changes.

## 6.3 Example

**Matrix Sampler** widget samples data into two subsets: in-sample and out-of-the-sample data. This is useful if you want to check the accuracy of matrix reconstruction with **Completion Scoring**. Feed in-sample data into the **Fusion Graph** to reconstruct the matrix and then compare the results with out-of-the-sample data.


data-fusion*
File Edit View Widget Options Help
IMDB
IMDb Actors
Movie Ratings
Movie Genres
Fusion Graph
Matrix Sampler
4|0
Completion Scoring
Fusion Graph
Info
4 object types
4 relations
Relations
147×147 Actors costar with Actors
147×171 Actors play in Movies
171×19 Movies fit in Genres
706×171 Users rate Movies
Fuser name
Users to Movies
Decomposition algorithm
Matrix tri-factorization
Matrix tri-completion
Initialization algorithm
Random
Random C
Random Vcol
Maximum number of iterations
10
Factorization rank
10%
Run after any change
Run
Actors
147
Users
706
play in
rate
Movies
171
fit in
Genres
19
Matrix Sampler
Sampling method
Rows
Columns
Rows and columns
Entries
Proportion of data in the sample
90%
Apply
Completion Scoring
RMSE
Users to Movies
[706×171] Users rate Movies
3.05452

CHAPTER 7

# Mean Fuser

Constructs relation matrices based on the average values of matrix elements.

## 7.1 Signals

**Inputs**:

- **Fusion Graph**

  A relational scheme of a data compendium.

- **Relation**

  Relationships between two groups of objects.

**Outputs**:

- **Mean-fitted fusion graph**

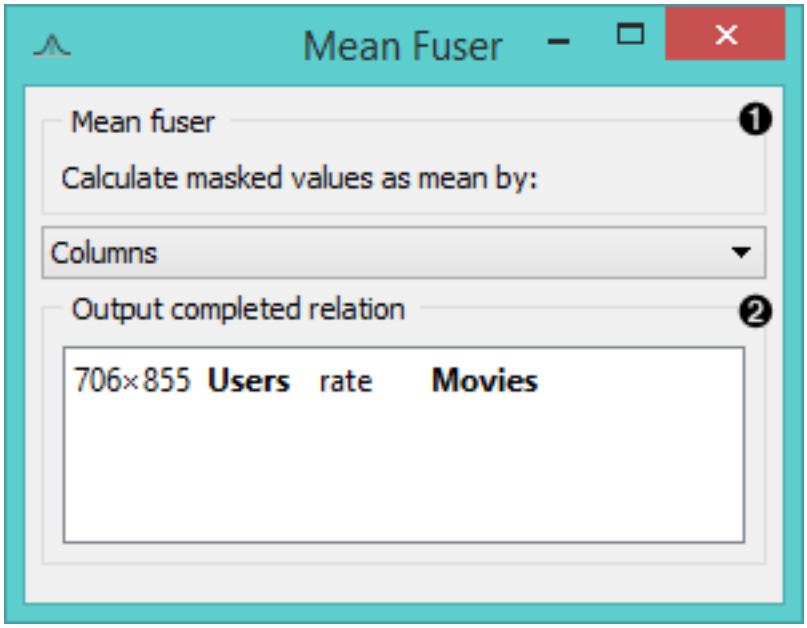
  Mean fuser.

- **Relation**

  Relationships between two groups of objects.

## 7.2 Description

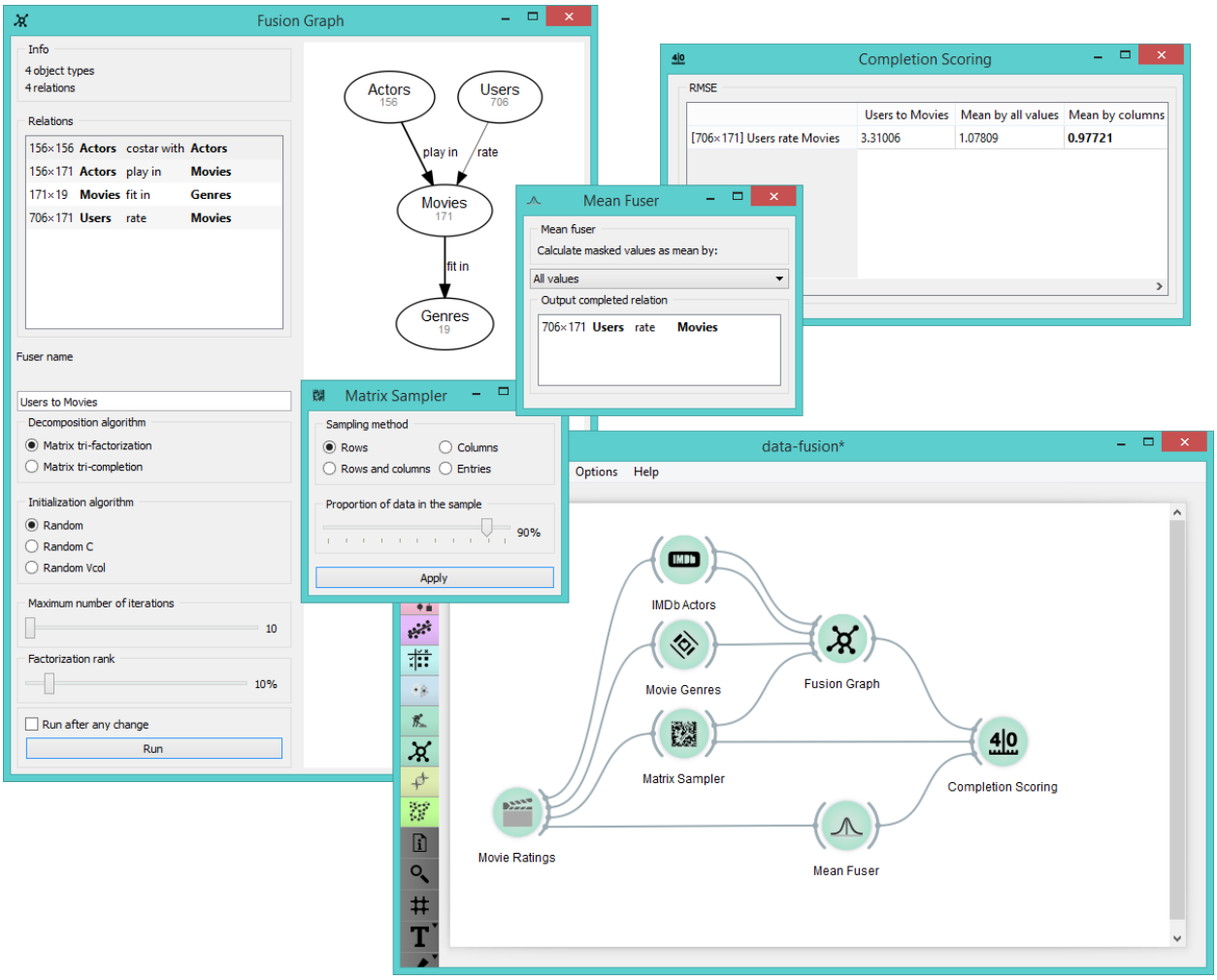
The widget completes each relation matrix at the input based on the available data in the matrix. Unknown values in the matrix can be replaced with the values obtained by averaging matrix rows, matrix columns or the entire data matrix.




1. Select the axis for mean value calculation:
   - **rows**
   - **columns**
   - **all**
2. Output selected relation matrix, where unknown matrix elements are replaced with mean values.

## 7.3 Example

**Mean Fuser** widget is useful for comparing RMSE values in **Completion Scoring** widget for the input data set. In the example below we have sampled movie ratings, fed the in-sample movie ratings data into **Fusion Graph** and from there into **Completion Scoring** for evaluation. We also fed the out-of-sample data from **Matrix Sampler** into **Completion Scoring** widget as out-of-sample movie ratings data is needed to assess how well the predicted values correspond to the true data. Finally, we compare prediction to those obtained by **Mean Fuser**.


Fusion Graph
Info
4 object types
4 relations
Relations
156×156 Actors costar with Actors
156×171 Actors play in Movies
171×19 Movies fit in Genres
706×171 Users rate Movies
Actors
156
Users
706
play in
rate
Movies
171
fit in
Genres
19
Fuser name
Users to Movies
Decomposition algorithm
Matrix tri-factorization
Matrix tri-completion
Initialization algorithm
Random
Random C
Random Vcol
Maximum number of iterations
10
Factorization rank
10%
Run after any change
Run
Completion Scoring
RMSE
Users to Movies
Mean by all values
Mean by columns
[706×171] Users rate Movies
3.31006
1.07809
0.97721
Mean Fuser
Mean fuser
Calculate masked values as mean by:
All values
Output completed relation
706×171 Users rate Movies
Matrix Sampler
Sampling method
Rows
Columns
Rows and columns
Entries
Proportion of data in the sample
90%
Apply
data-fusion*
Options
Help
IMDb Actors
Movie Genres
Matrix Sampler
Movie Ratings
Fusion Graph
Completion Scoring
Mean Fuser

CHAPTER 8

# Movie Genres

Constructs a movies-by-genres or actors-by-genres relation matrix.

## 8.1 Signals

**Inputs**:

- **Row type**

  Instances from the input data.

**Outputs**:

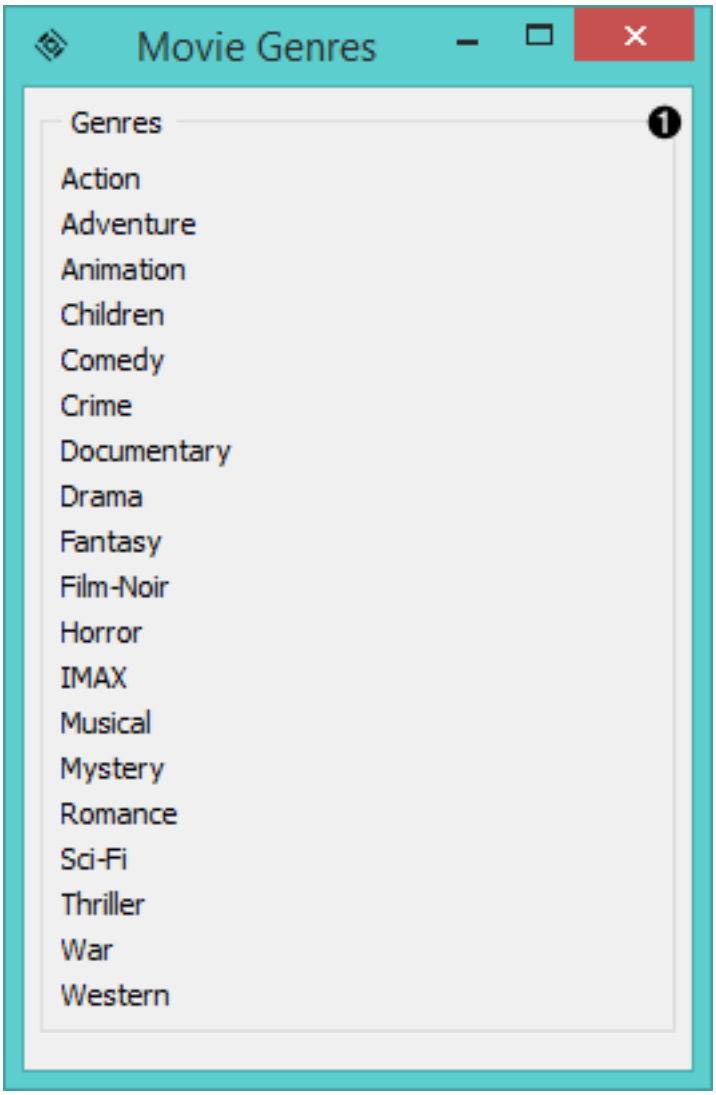
- **Genres**

  Data-by-genres relation matrix.

## 8.2 Description

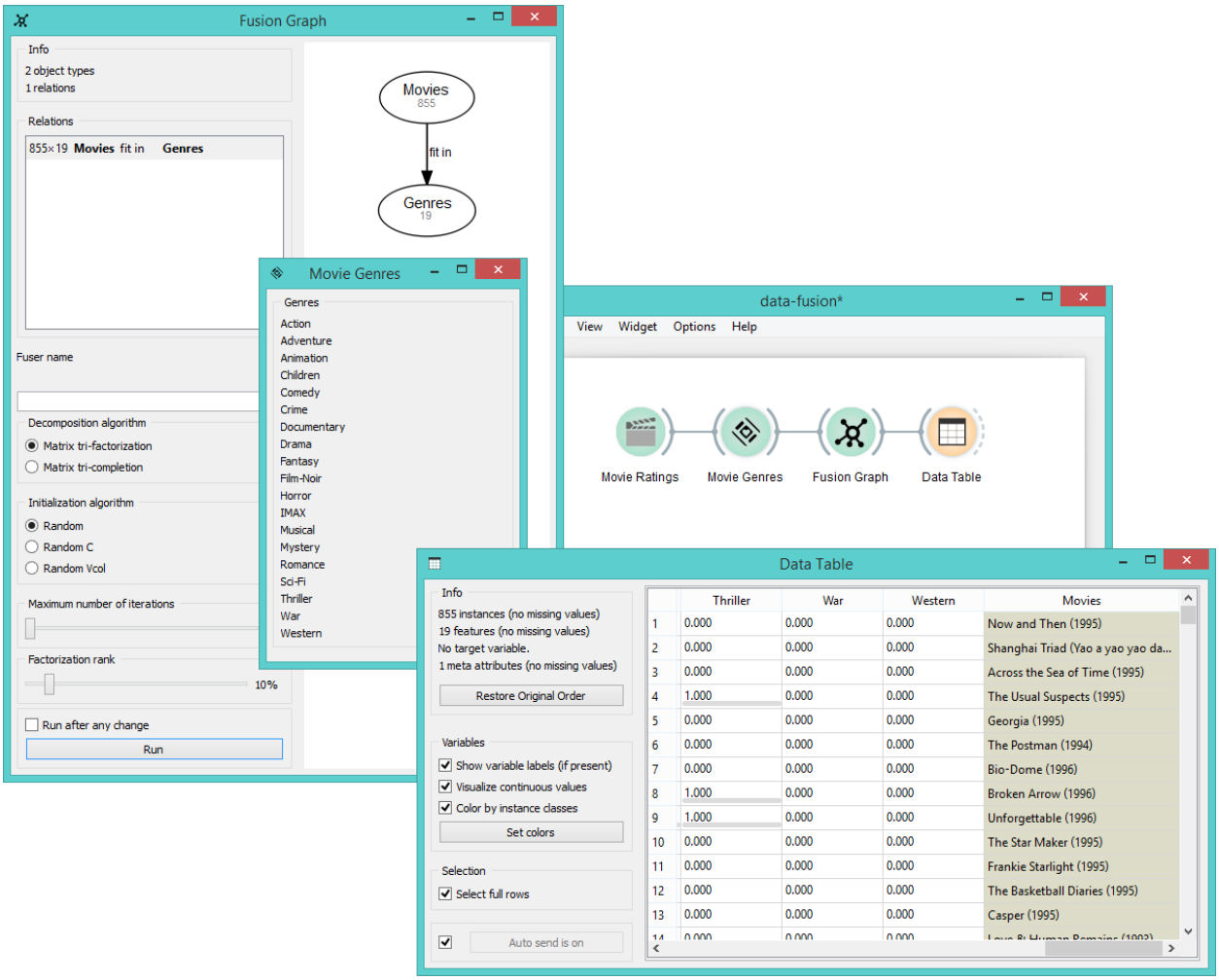
This widget matches movies or actors to movie genres and forms a relation matrix. It is used to obtain information about the genres to which movies in the input belong or about genres that are associated with actors given in the input.

1. A list of movie genres included in the MovieLens database.


Movie Genres
Genres
1
Action
Adventure
Animation
Children
Comedy
Crime
Documentary
Drama
Fantasy
Film-Noir
Horror
IMAX
Musical
Mystery
Romance
Sci-Fi
Thriller
War
Western

## 8.3 Example

Below we constructed a movies-by-genres relation matrix using the **Movie Genres** widget. You can see in the **Data Table** that all movies are matched by their genres.



CHAPTER 9

# Movie Ratings

Constructs a relation matrix of user ratings for movies.

## 9.1 Signals

**Inputs**:

- (None)

**Outputs**:

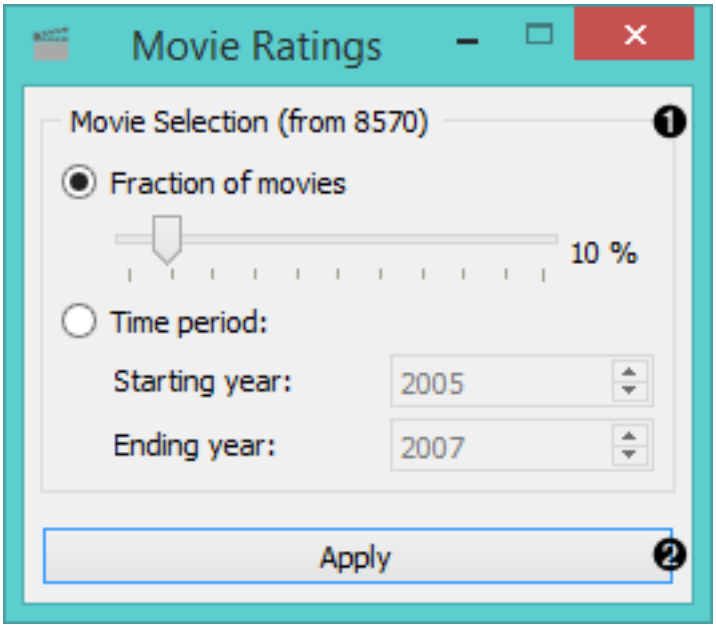
- **Ratings**

  Movie ratings relation matrix.

## 9.2 Description

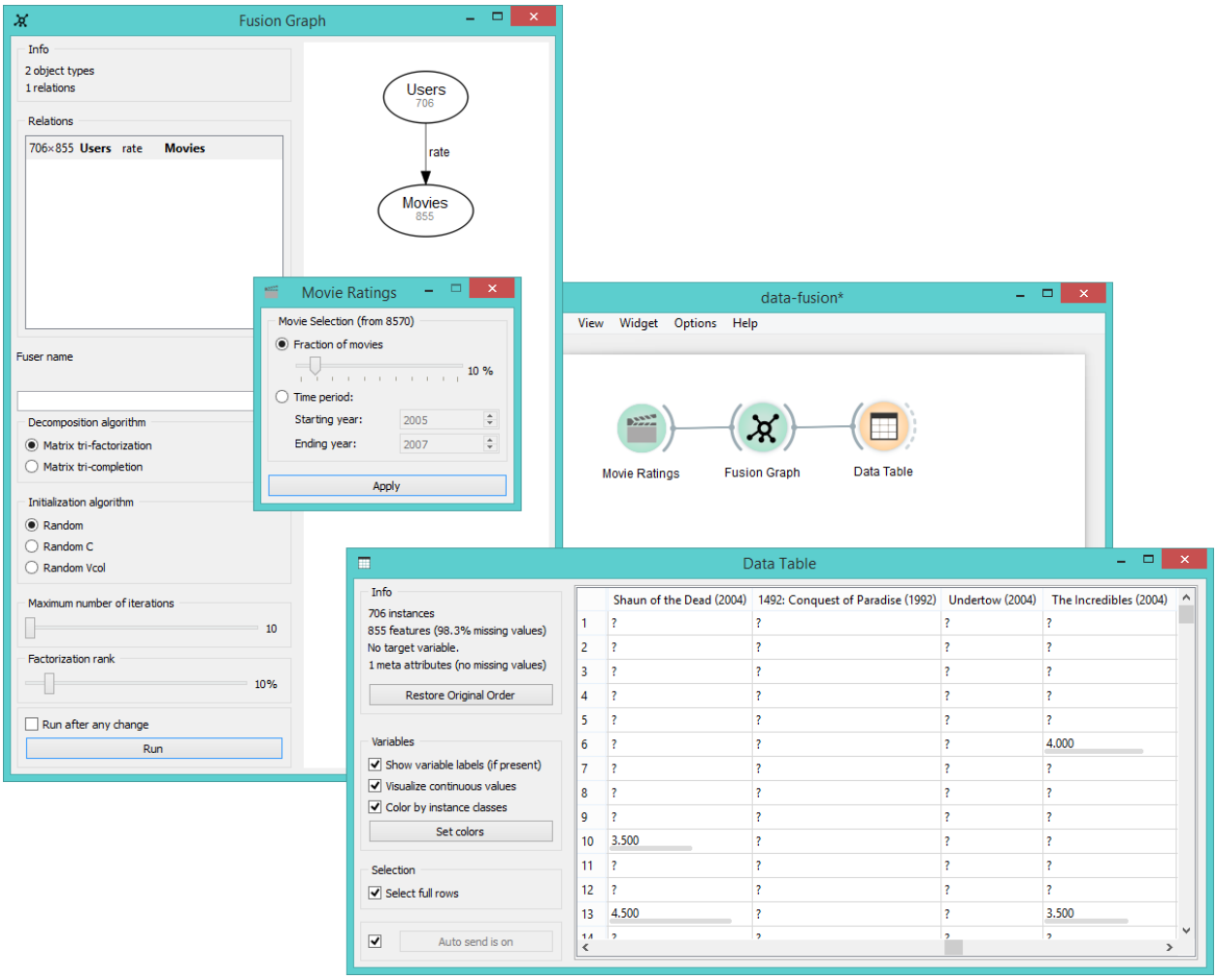
**Movie Ratings** widget gives you access to data on user ratings for more than 8500 movies from the Movielens database. The data set contains 1 to 5-star ratings representing user-movie preferences. This is a good widget to try out data fusion as it gives you instant access to the data.

1. Select a subset of movies for which you would like to obtain user ratings:
   - **fraction of movies** will output a specified fraction of movies selected uniformly at random from the entire database.
   - **time period** will output all the movies released in a specified time period
2. Click *Apply* to commit the changes.




## 9.3 Example

**Movie Ratings** will output users-by-movies data matrix for further analysis. Feed it into the **Fusion Graph** to decompose data matrix into the product of smaller latent data matrices or view the data in a **Data Table**.


Fusion Graph
Info
2 object types
1 relations
Relations
706×855 Users rate Movies
Users
706
rate
Movies
855
Fuser name
Decomposition algorithm
Matrix tri-factorization
Matrix tri-completion
Initialization algorithm
Random
Random C
Random Vcol
Maximum number of iterations
10
Factorization rank
10%
Run after any change
Run
Movie Ratings
Movie Selection (from 8570)
Fraction of movies
10 %
Time period:
Starting year:
2005
Ending year:
2007
Apply
data-fusion*
View
Widget
Options
Help
Movie Ratings
Fusion Graph
Data Table
Data Table
Info
706 instances
855 features (98.3% missing values)
No target variable.
1 meta attributes (no missing values)
Restore Original Order
Variables
Show variable labels (if present)
Visualize continuous values
Color by instance classes
Set colors
Selection
Select full rows
Auto send is on
Shaun of the Dead (2004)
1492: Conquest of Paradise (1992)
Undertow (2004)
The Incredibles (2004)

CHAPTER 10

# Table to Relation

Converts a data table into a relation matrix. Labels objects in rows and columns of a relation matrix.

## 10.1 Signals

**Inputs**:

- **Data**

  Attribute-valued data set.

**Outputs**:

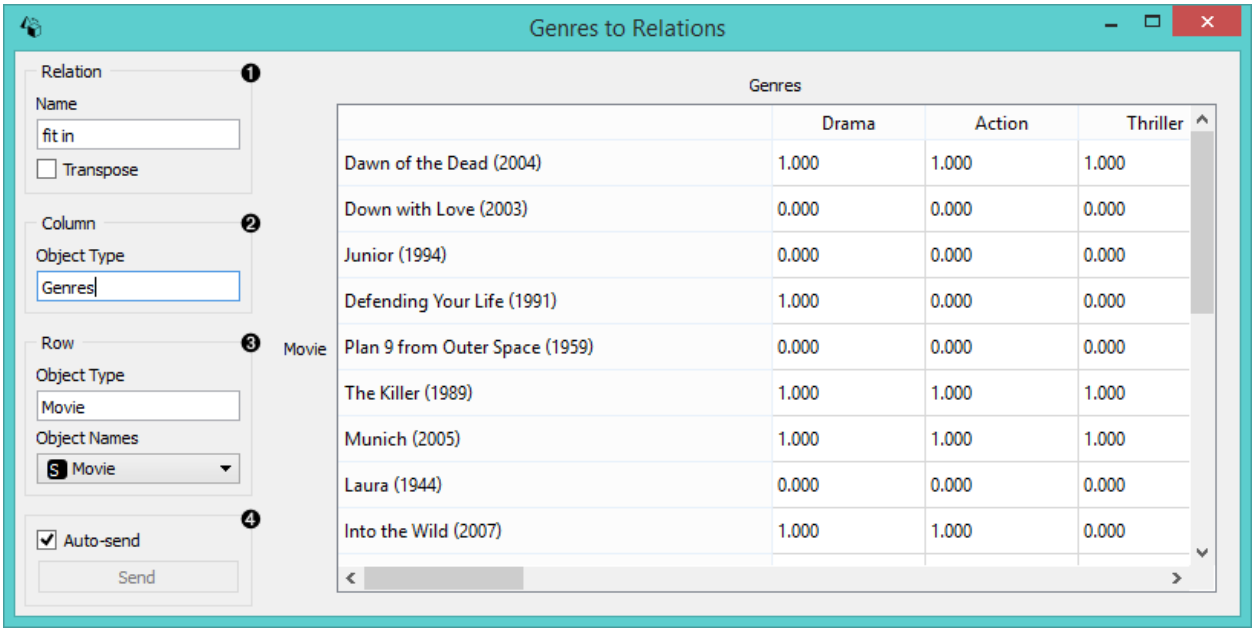
- **Relation**

  Relationships between two groups of objects.

## 10.2 Description

**Table to Relation** widget is probably the most often used widget in the data fusion set. It allows you to define relations just by labeling the axes. Your data set from the **File** widget will be transformed into a relation matrix, which can be later fused together with other relation matrices into a collective latent data model.

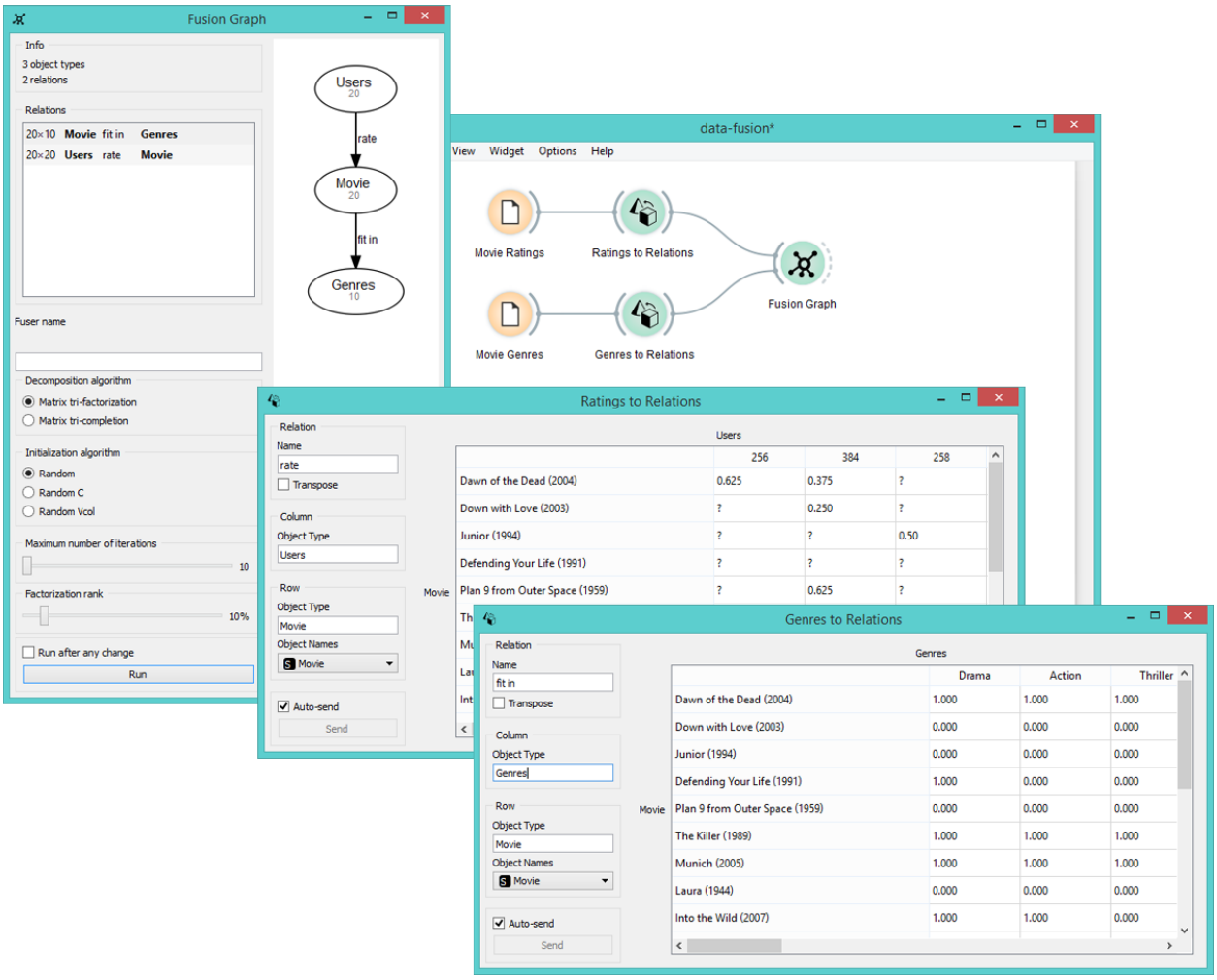
1. Provide a descriptive name for the relation. Option *transpose* will shift the axes.
2. Label the object type in columns. Your entry will be displayed on top of the table. Note that the labels are case-sensitive.
3. Label the object type in rows. If there is a label present in the data, it will be used as default.




4. If *Auto send* is ticked, your changes will be communicated automatically. Alternatively click *Send*.

## 10.3 Example

In the example below we took two regular files with data on movie ratings and movie genres and fed them into separate **Table to Relation** widgets. In these widgets we specified the relations contained in the data and named the axes accordingly. See how **Fusion Graph** is then able to organize data sets into a relational graph, i.e. a data fusion graph, simply on the basis of axes names?


Fusion Graph
Info
3 object types
2 relations
Relations
20×10 Movie fit in Genres
20×20 Users rate Movie
Users
20
rate
Movie
20
fit in
Genres
10
Fuser name
Decomposition algorithm
Matrix tri-factorization
Matrix tri-completion
Initialization algorithm
Random
Random C
Random Vcol
Maximum number of iterations
10
Factorization rank
10%
Run after any change
Run
data-fusion*
View Widget Options Help
Movie Ratings
Ratings to Relations
Movie Genres
Genres to Relations
Fusion Graph
Ratings to Relations
Relation
Name
rate
Transpose
Column
Object Type
Users
Row
Object Type
Movie
Object Names
Movie
Auto-send
Send
Users
256 384 258
Dawn of the Dead (2004) 0.625 0.375 ?
Down with Love (2003) ? 0.250 ?
Junior (1994) ? ? 0.50
Defending Your Life (1991) ? ? ?
Plan 9 from Outer Space (1959) ? 0.625 ?
Movie
Genres to Relations
Relation
Name
fit in
Transpose
Column
Object Type
Genres
Row
Object Type
Movie
Object Names
Movie
Auto-send
Send
Genres
Drama Action Thriller
Dawn of the Dead (2004) 1.000 1.000 1.000
Down with Love (2003) 0.000 0.000 0.000
Junior (1994) 0.000 0.000 0.000
Defending Your Life (1991) 1.000 0.000 0.000
Plan 9 from Outer Space (1959) 0.000 0.000 0.000
The Killer (1989) 1.000 1.000 1.000
Munich (2005) 1.000 1.000 1.000
Laura (1944) 0.000 0.000 0.000
Into the Wild (2007) 1.000 1.000 0.000
Movie

CHAPTER 11

# Indices and tables

- genindex
- modindex
- search