Path2PPI - Tutorial, example and the algorithm
Prediction of autophagy induction in Podospora anserina

Contents

1 Introduction 1
2 Preparation of the data 2
  2.1 Proteins and interactions of pathways of interest 2
  2.2 Get homology files using NCBI BLAST+ 3
3 Predict PPI in target species 4
  3.1 The Path2PPI object 4
  3.2 Add reference species 4
  3.3 Predict PPI 6
4 Results of the prediction 7
  4.1 Plotting the results 7
  4.2 Get detailed information about each interaction 10
  4.3 Export results 12
5 References 13
6 Session info 14
7 Appendix 14
  7.1 Biological evidence of the predicted PPI network 14
  7.2 The prediction algorithm 15

1 Introduction

Prediction of protein-protein interaction (PPI) networks is an important approach to gain knowledge about protein interactions in model organisms where only a small number of PPI information is available. Current PPI databases, providing predicted interaction data, lack many organisms or contain less reproducible information about the predicted interactions. Currently available prediction approaches are mainly based on biological data (functional annotation, co-expression etc.) which often are not available for many “less established” organisms, for example, where only sequence data is available. In addition, it is of major interest to get knowledge about a certain pathway in such a “less-studied” organism. To overcome these drawbacks Path2PPI can be used to predict proteins and interactions of a certain pathway of interest in a target organism by using and combining the PPIs of other well established model organisms.

To do so, it needs a list of proteins of interest from each reference species and the result files produced by the local NCBI BLAST (Camacho et al., 2009) tool (see next chapter). The relevant interactions based on the users’ protein lists are automatically extracted from the corresponding iRefIndex files (Razick et al., 2008).
2 Preparation of the data

In this tutorial, we make use of the test data set provided with the package. This data set consists of all data files necessary to predict the interactions of the induction step of autophagy in *Podospora anserina* by means of the corresponding PPIs in human and yeast. Hence, we first load the “autophagy induction” test data set:

```r
data(ai) #Load test data set
ls() #"ai" contains six data objects
```

As stated by `ls()` the test data set contains six data objects (three for each of the two reference species human and yeast). First, the algorithm requires a list of proteins which define the corresponding pathway for each reference species, defined in “human.ai.proteins” and “yeast.ai.proteins” (see section 2.1). Second, the algorithm requires the data frames which contain the interactions of each reference species defined in “human.ai.irefindex” and “yeast.ai.irefindex” (described in more detail in section 2.1). Third, the algorithm needs to know the homologous relations between the target species with each reference species. These relations are defined in the data frames “pa2human.ai.homologs” and “pa2yeast.ai.homologs” (we describe this in more detail in section 2.2).

If you want to use Path2PPI for your own demands, you have to generate and prepare the necessary data files.

### 2.1 Proteins and interactions of pathways of interest

We list the proteins which are associated with a specific pathway of interest in a character vector for each reference species. To give you an example for such lists, we take a brief look into the loaded data set. Among others, we found the two named character vectors “human.ai.proteins” and “yeast.ai.proteins” which consist of the corresponding proteins for yeast and human, our two reference species:

#### human.ai.proteins

```r
# [1] "P42345" "075385" "Q8IYT8" "Q6PHR2" "075143"
```

#### yeast.ai.proteins

```r
# [1] "P35169" "P32600" "P53104" "Q06410" "Q12527" "Q06628" "P39968"
```

In this example, the values are the trivial names of the proteins and the names are the actual protein identifiers. Path2PPI also accepts simple character vectors where the values are the protein identifiers, if the trivial names of the proteins are not available. For example, this simple character vector, only consisting of the protein identifiers, would be also a valid protein list:

```r
# [1] "P42345" "075385" "Q8IYT8" "Q6PHR2" "075143"
```
The major advantage of using a named character vector with the trivial names, is that these names will be shown in the plots allowing for a more comfortable interpretation. You can use various accession formats for the protein identifiers which are supported by \textit{iRefIndex} (e.g. UniProt, SwissProt, Ensembl). However, we urgently recommend to use UniProt identifiers, since those are the most established ones.

Use the default R functions to load your own protein lists of interest into R (e.g. \texttt{read.table}).

These proteins of interest are applied to find relevant interactions in the corresponding species \textit{iRefIndex} file. \textit{iRefIndex} tables are available for the seven most established model organisms and can be found here: http://irefindex.org/wiki/index.php?title=iRefIndex. You can also use the corresponding \textit{iRefR}-package to directly archive the \textit{iRefIndex} data frames from this page. Unfortunately, the package is not updated as frequently as the web page and it may be that you do not get the latest release of a corresponding file.

In the “autophagy induction” test data set, only a very small part of the \textit{iRefIndex} files for yeast and human are provided which contain the relevant interactions necessary for this tutorial. The complete files are much larger. The data frames “\texttt{human.ai.irefindex}” and “\texttt{yeast.ai.irefindex}” in the test data set contain these corresponding \textit{iRefIndex} parts. See:

\begin{verbatim}
str(human.ai.irefindex)
str(yeast.ai.irefindex)
\end{verbatim}

\section*{2.2 Get homology files using NCBI BLAST+}

You also need the result files produced by the BLAST+ toolkit provided by the NCBI web page: http://blast.ncbi.nlm.nih.gov/Blast.cgi?PAGE_TYPE=BlastDocs&DOC_TYPE=Download. The test data set already includes the necessary results of the BLAST searches of the proteomes of \textit{P. anserina} against the proteomes of yeast and human (“\texttt{pa2human.ai.homologs}” and “\texttt{pa2yeast.ai.homologs}”):

\begin{verbatim}
head(pa2yeast.ai.homologs)
## V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12
## 2123 B2AX00 Q6PHR2 33.09 269 146 15 11 268 13 258 5e-24 106.0
## 2177 B2AX00 075385 30.30 231 131 9 24 239 22 237 3e-22 103.0
## 2213 B2AX00 Q81YT8 29.24 236 142 8 24 247 15 237 4e-21 99.4
## 4588 B2AXW7 Q6PHR2 32.96 267 159 12 499 754 6 263 9e-30 125.0
## 4649 B2AXW7 Q81YT8 29.17 216 136 6 509 714 14 222 2e-23 107.0
## 4658 B2AXW7 075385 29.95 217 133 9 509 714 21 229 5e-23 106.0
\end{verbatim}

head(pa2human.ai.homologs)
\begin{verbatim}
## V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12
## 534 B2AX00 P53104 25.66 304 165 12 15 268 21 313 1e-15 79.0
## 960 B2AXW7 P53104 28.52 291 154 10 510 751 30 315 7e-24 106.0
## 1278 B2AUR5 P35169 26.74 288 154 10 602 837 2078 2360 2e-11 65.9
## 1279 B2AUR5 P32600 25.09 275 152 9 610 835 2090 2359 1e-09 60.5
## 1555 B2B7B1 P53104 25.99 177 99 7 807 963 149 313 5e-09 59.3
## 2469 B2ASL9 P53104 26.42 352 171 13 20 314 24 344 2e-19 87.8
\end{verbatim}

The second column (V2) contains the protein identifiers of the corresponding reference species to which the protein of the target species (here: \textit{P. anserina}) in the first column (V1) is homologous. Keep in mind that these protein identifiers are equal to those we used in the protein lists described above.

If you are unfamiliar with this toolkit, we refer to the BLAST+ user manual or to the broadly available tutorials in the web.
Nevertheless, we want to give you a very short description on how to use this toolkit. We assume that you already have loaded and installed the NCBI BLAST+ toolkit and you also have each proteome file in FASTA format of each species (target and reference species). You first have to create the databases for each reference species, here, for human and yeast:

```bash
makeblastdb -in human.fa -input_type fasta -dbtype prot -out human_proteins -title human_proteins
makeblastdb -in yeast.fa -input_type fasta -dbtype prot -out yeast_proteins -title yeast_proteins
```

Subsequently, you can start the comprehensive BLAST searches using the FASTA file of your target species (here: *P. anserina*):

```bash
blastp -query panserina.fa -db human_proteins -out human_panserina.out -evalue 0.0001 -outfmt 6
blastp -query panserina.fa -db yeast_proteins -out yeast_panserina.out -evalue 0.0001 -outfmt 6
```

Please, make sure that you use as the output format the tab delimited list indicated by the parameter `-outfmt 6`. The two species-specific homology files which are now generated, can be imported into your R session, using the function `read.table`, and subsequently used as data frames for the Path2PPI-package.

## 3 Predict PPI in target species

After the necessary data sets are generated or loaded, respectively, we can start with the prediction.

### 3.1 The Path2PPI object

An object of the class *Path2PPI* represents the major instance which is responsible for storing and managing of each data set and for each computation and prediction step. Hence, we first have to create a new instance of the class *Path2PPI* with the corresponding information:

```r
ppi <- Path2PPI("Autophagy induction", "Podospora anserina", "5145")
```

The arguments are the title of the pathway we want to predict, the taxonomy name of the target species (“Podospora anserina”) and its corresponding taxonomy id (“5145”).

### 3.2 Add reference species

This new instance does not contain any reference species or a predicted PPI, yet:

```r
ppi
```

```
# Autophagy induction in Podospora anserina (5145)
# -------------------------------------------------
# No reference species yet.
# -------------------------------------------------
# No predicted PPI yet.
```
To add the reference species, for which we have collected the necessary data, we make use of the method `addReference`.

```r
ppi <- addReference(ppi, "Homo sapiens", "9606", human.ai.proteins,
                     human.ai.irefindex, pa2human.ai.homologs)
```

```r
## Search for all relevant interactions:
## 0%--25%--50%--75%--100%
## Remove irrelevant homologs.
```

Besides the taxonomy name and the taxonomy identifier, this method requires the list, containing the proteins of the pathway of interest, the corresponding `iRefIndex`-data frame or the file name of the corresponding `iRefIndex` file, and the species specific homology data set generated by the NCBI BLAST+ toolkit. This method searches for all relevant interactions in the `iRefIndex` data frame. There are different and often ambiguous protein identifiers defined in an `iRefIndex` file and the “major” identifiers are not necessarily those defined in the corresponding “major” columns “uidA” and “uidB”. Furthermore, `iRefIndex` also contains complexes. Hence, this method applies an advanced search algorithm to find automatically relevant interactions associated with the pathway or the proteins of interest, respectively. You do not have to predefined the identifiers’ types (UniProt, Swissprot, Ensembl etc.), since these types are often assigned ambiguously. The algorithm searches for each identifier in 10 columns where any type of identifier or accession number is defined, for example, “uidA”, “altA”, “OriginalReferenceA”, “FinalReferenceA”, “aliasA”, “uidB”, “altB”, “OriginalReferenceB”, “FinalReferenceB” and “aliasB”. Additionally, it searches for each complex to which one or more of the predefined proteins are associated. Subsequently, each homologous relationship which is not relevant for the previously found interactions is declined.

In the same manner we add yeast to our Path2PPI-instance:

```r
ppi <- addReference(ppi, "Saccharomyces cerevisiae (S288c)", "559292",
                     yeast.ai.proteins, yeast.ai.irefindex,
                     pa2yeast.ai.homologs)
```

```r
## Search for all relevant interactions:
## 0%--25%--50%--75%--100%
## Remove irrelevant homologs.
```

In this tutorial, we want to predict the PPIs in *P. anserina* based on these two reference species. You can use other and/or more reference species for your demands.

Now, we can get all processed information about the added reference species using the method `showReferences`:

```r
showReferences(ppi)
```

```r
## Homo sapiens (TaxId: 9606)
## ---------------------------
## 5 proteins (0 not used)
## 894 interactions:
## - 6 interactions have both interactors in protein list.
## - 349 interactions have at least one interactor in protein list.
## - 660 interactions in 102 protein complexes.
##
## SACCHAROMYCES CEREVISIAE (S288C) (TaxId: 559292)
## -----------------------------------------------
```
# 7 proteins (0 not used)
# 2910 interactions:
# - 15 interactions have both interactors in protein list.
# - 834 interactions have at least one interactor in protein list.
# - 2207 interactions in 102 protein complexes.

If we want to know which interactions have been found or which interactions are associated with the proteins of interest in a specific reference species (e.g. human), we can use the method as follows:

```r
interactions <- showReferences(ppi, species="9606",
                            returnValue="interactions")
head(interactions)
```

| ref     | A.db       | A.accession | A.in.prot.list | B.db       |
|---------|------------|-------------|----------------|------------|
| 1       | 287217     | complex     | qx1eWqPyfshUfC/6x17AYjcT/3w | FALSE      |
| 7       | 287217     | complex     | qx1eWqPyfshUfC/6x17AYjcT/3w | UNIPROTKB  |
| 13      | 287217     | complex     | qx1eWqPyfshUfC/6x17AYjcT/3w | FALSE      |
| 19      | 436141     | UNIPROTKB   | A0A090N900     | FALSE      |
| 32      | 5028958    | UNIPROTKB   | P62942         | FALSE      |
| 45      | 408315     | UNIPROTKB   | Q8N122         | FALSE      |

For more information about the method we refer to the corresponding manual page (`?showReferences`).

## 3.3 Predict PPI

After we added all reference species and all necessary data, we can start with the prediction. To predict the PPI network in the target species we use the method `predictPPI`:

```r
ppi <- predictPPI(ppi, h.range=c(1e-60,1e-20))
```

## Begin with Homo sapiens
## 6 interactions processed. These lead to 5 interactions in target species.
## -------------------------------
## Begin with Saccharomyces cerevisiae (S288c)
## 15 interactions processed. These lead to 22 interactions in target species.
## -------------------------------
## Combine results to one single PPI.
## A total of 13 putative interactions were predicted in target species.

This method uses different arguments to influence the prediction approach and to define the output of the PPI network. For a detailed description of the various arguments we refer to the corresponding manual (`?predictPPI`). Here, we only use the argument `h.range` where the first value corresponds to the lower bound and the second value to the upper bound of the homology range. That means that each E-value which is equal or less the lower bound will be scored with 1, and each E-value which is equal or larger than the upper bound will be scored with 0 (see appendix for a detailed description).
According to the reports generated by this method two species specific PPI networks led to a PPI network in the target species with 13 interactions. To achieve further information about the former prediction step, we just type:

```r
ppi #show(ppi)
```

```
## Autophagy induction in Podospora anserina (5145)
## -------------------------------------------------
## 2 reference species: 9606, 559292
## -------------------------------------------------
## Number of predicted proteins: 8
## Number of predicted interactions: 13
## Predicted PPI based on 2 reference species:
## 9606 (2 interactions and 4 homologous relations)
## 559292 (11 interactions and 11 homologous relations)
## -------------------------------------------------
## Settings:
## Homology threshold: 1e-05
## Homology range: [1e-60,1e-20]
## Interactions threshold: 0.7
## Consider complexes: FALSE
```

4 Results of the prediction

After we predicted the PPI network of the “autophagy induction” pathway in *P. anserina* we now want to know how this network looks like. And we want to know which proteins and interactions actually are associated with this pathway in our target species.

4.1 Plotting the results

To get a graphical representation of the predicted PPI network, *Path2PPI* provides three different plotting types. First, to get only the predicted PPI, we use the plot function of the Path2PPI-object, which is based on the *igraph* plotting function (Csardi and Nepusz, 2006):

```r
set.seed(12) #Set random seed
coordinates <- plot(ppi, return.coordinates=TRUE)
```
There are various arguments provided with this method (see `?plot.Path2PPI`). Here, we initially use the `return.coordinates` argument since we want to save the coordinates of the vertices for the next plotting approach.

In the second approach, we want to know from which reference species the different predicted interactions originated. We assign the previously computed coordinates to the plotting function since we want to compare both networks:

```r
plot(ppi, multiple.edges=TRUE, vertices.coordinates=coordinates)
```
Autophagy induction in Podospora anserina (5145) – PPI detailed

The different colors of the edges correspond to the species, see the taxonomy identifiers in the legend: 5154 for *P. anserina*, 9606 for human, and 559292 for yeast) from which the interaction was deduced. For example, we can see that the edge between the proteins “B2AWL” and “B2AE79” in the upper network is thicker than the others. This indicates that the interaction was found in more than one reference species. In the second plot, we see that this interaction is based on six interactions found in yeast and two interactions found in human.

Next, we want to plot the so-called *hybrid* PPI network, where we additionally can see the underlying reference interactions or the underlying reference PPI networks, respectively, and each homologous relationship. We also want to set the vertex labels, since we know the trivial names of the target species proteins. You can set the label for each protein of each species. Additionally, we want to change the species colors:

```r
set.seed(40)
target.labels<-c("B2AE79"="PaTOR","B2AXK6"="PaATG1",
              "B2AUW3"="PaATG17","B2AM44"="PaATG11",
              "B2AQV0"="PaATG13","B2B5M3"="PaVAC8")
species.colors <- c("5145"="red","9606"="blue","559292"="green")
plot(ppi,type="hybrid",species.colors=species.colors,
     protein.labels=target.labels)
```
The dotted edges correspond to an homologous relationship between a protein of the target species and a reference species. Only those proteins and interactions of the reference species are shown which actually led to an interaction in the target species.

### 4.2 Get detailed information about each interaction

After we know how the predicted PPI network of this pathway may look like in the target species we want to know more about the predicted interactions. For this purpose we make use of the method `showInteraction` which requires the protein identifiers of the interaction:

```r
showInteraction(ppi, interaction=c("B2AT71","B2AE79"))
```

```text
## Score: 1.989472
## 8 reference interactions: 559292 (6) 9606 (2)
```

For further details about the underlying reference interactions we can use the additional argument `mode`:

```r
showInteraction(ppi, interaction=c("B2AT71","B2AE79"), mode="detailed", verbose=FALSE)
```

```text
## source.id target.id score h.scoreA h.scoreB r.species r.species.s
##  1  B2AT71   B2AE79 0.855   0.71   1.00    9606        P42345
##  2  B2AT71   B2AE79 0.820   0.64   1.00  559292        P35169
##  3  B2AT71   B2AE79 0.825   0.65   1.00  559292        P32600
##  4  B2AT71   B2AE79 0.820   0.64   1.00  559292        P35169
```
This data frame contains each single predicted interaction of the current interaction and all corresponding reference interactions. For the interaction of the proteins in columns one and two, the third column gives the prediction score. The fourth and the fifth columns show the homology score of the source protein (A) or the target protein (B), respectively, in the target species to its equivalent (column seven and eight) in the reference species (column six). The column “pos.edges” (possible edges) indicates how many interactions could be derived from this interaction in the reference species, in contrast, to the number of interactions in “used.edges” which actually were adopted. The last column gives the identifier of this interaction in the \textit{iRefIndex} data set.

To get the corresponding \textit{iRefIndex} entries for these reference interactions we can use this method as follows:

```
ref.interaction <- showInteraction(ppi, interaction=c("B2AT71","B2AE79"), mode="references.detailed", verbose=FALSE)
```

The data frame now stored in the variable \texttt{ref.interaction} is part of the \textit{iRefIndex} table which contains all information about the current reference interactions. We can use this data frame with default R functionality to search for specific information of the reference interactions. For example, if we want to know from which study (author and publication) and from which database the interaction of the proteins “P42345” and “P42345” with the interaction identifier “742389” (first row) was adopted, we use:

```
ref.interaction[ref.interaction$irigid=="742389", c("author","pmids","sourcedb")]
```
In this manner you are able to search for each entry provided by the *iRefIndex* table.

### 4.3 Export results

Now, we want to work with the predicted PPI network and do further analyses either directly in R or using an advanced network analysis tool like Cytoscape (Cline et al., 2007). To do so, we can use the method `getPPI`, which gives us the edge list of the PPI network:

```r
my.ppi <- getPPI(ppi)
my.ppi
```

```markdown
## source.id target.id score r.species
## 1 B2AT71 B2AE79 1.989472 559292,9606
## 2 B2AE79 B2AWL8 1.969948 559292,9606
## 3 B2AQV0 B2B5M3 1.000000 559292
## 4 B2AQV0 B2AXK6 1.000000 559292
## 5 B2AM44 B2AXK6 1.000000 559292
## 6 B2AM44 B2AUW3 1.000000 559292
## 7 B2AQV0 B2AUW3 1.000000 559292
## 8 B2AE79 B2AQV0 1.000000 559292
## 9 B2AXK6 B2AUW3 1.000000 559292
## 10 B2AXK6 B2B5M3 1.000000 559292
## 11 B2AXK6 B2AE79 0.820000 9606
## 12 B2AT71 B2AQV0 0.820000 559292
## 13 B2AWL8 B2AQV0 0.780000 559292
```

If you want this PPI network with each single (not the combined one) predicted interaction, use the additional argument `raw=TRUE`.

We also want the edge list of the hybrid network which includes the PPIs of the reference species. For this purpose we use the method `getHybridNetwork`:

```r
my.hybrid <- getHybridNetwork(ppi)
my.hybrid
```

```markdown
## source.id target.id t.species.id score type
## 1 B2AT71 P42345 9606 0.710000 homology
## 2 B2AT71 P35169 559292 0.640000 homology
## 3 B2AT71 P32600 559292 0.650000 homology
## 5 B2AE79 P42345 9606 1.000000 homology
## 6 B2AE79 P35169 559292 1.000000 homology
## 7 B2AE79 P32600 559292 1.000000 homology
## 13 B2AWL8 P42345 9606 0.510000 homology
```
If you want to work with an igraph object instead, use the additional argument `igraph=TRUE` in both methods. To export the edge lists, use default R functions like `write.table`.

5 References

Camacho, C. et al. (2009). BLAST+: architecture and applications. BMC Bioinformatics, 10(1), 421.

Cline, M. S. et al. (2007). Integration of biological networks and gene expression data using Cytoscape. Nature Protocols, 2(10), 2366-2382.

Csardi, G. and Nepusz, T. (2006). The igraph software package for complex network research. InterJournal Complex Systems, 1695(5), 1-9

Razick, S. et al. (2008). iRefIndex: a consolidated protein interaction database with provenance. BMC Bioinformatics, 9(1), 405.

Kanehisa, M. et al. (2014). Data, information, knowledge and principle: back to metabolism in KEGG. Nucleic Acids Research, 42(D1), D199-D205.
7 Appendix

7.1 Biological evidence of the predicted PPI network

The example of autophagy induction was chosen, since it is well known and described for many eukaryotic organisms, including yeast, human and even \textit{P. anserina}. Representatively and briefly described for yeast, the induction step works as follows:

Generally, nutrient availability activates the TOR-kinase, which phosphorylates ATG13. Hyperphosphorylated ATG13 cannot interact with ATG1 and ATG17 to build the ATG1 complex (including ATG11 and VAC8), which is important for autophagosome nucleation. Hence, autophagy is inhibited. In contrast, under nutrient-depleted conditions, TOR is deactivated, ATG13 is dephosphorylated and available for complex formation with ATG1 and ATG17. The next step of the autophagy process - autophagosome nucleation - can begin.

As reference proteins we used those provided by the KEGG database (Kanehisa et al., 2014) for yeast (seven) and human (five). We compared the predicted PPI with that provided by the KEGG database for \textit{P. anserina}.

We found that two of the five proteins of human, ULK1 and TOR, and all of the seven proteins of yeast have been taken into account from the algorithm. This is due to the closer evolutionary distance of the both fungi and the corresponding proteins. The algorithm predicts all of the proteins and interactions provided by the KEGG database for the induction step of autophagy in \textit{P. anserina}. Additionally, it also includes two additional proteins, which are not characterized yet. This is probably due to the fact that these two proteins are putative serine/threonine kinases with similar motifs and sequences like the TOR kinase.

Summarizing, this example shows that \textbf{Path2PPI} is able to transfer interaction data amongst organisms. The resulting PPI networks can serve as starting points for further analyses and PPI studies.
7.2 The prediction algorithm

**Aim:** We aim to predict an interaction network in a target organism based on the interaction networks of well established model organisms. The networks are based on a predefined set of proteins which may belong to a certain pathway. We consider the degree of homologies and the number of reference species.

**Requirements:** The major initial requirements for the algorithm are the PPIs for each reference species, the BLAST results of the target species (P. anserina) against each reference species, and an E-value range defined by the upper bound $h_u$ and the lower bound $h_l$. The range is required to map the BLAST E-value interval $[h_l, h_u]$ to a homology score in the interval $[1, 0]$ where 0 is the worst and 1 the best value. We will exemplarily describe the algorithm based on the two reference species, human and yeast.

7.2.1 Computing preliminary reference species-specific PPIs

*Figure 1. Steps of the prediction algorithm* a) It searches for each interaction (blue edges) in each reference species (blue) the corresponding homologs (dotted edges) in the target species (red). Each valid homologous protein from the first set, $H_a$, will now be connected with each homologous protein from the other set, $H_b$. The deduced interactions are scored, using the function described below. b) The linear function to map the BLAST E-values to the interval $[0, 1]$. c) Combining the redundantly predicted interactions to one final PPI network, $PPI_p$, of the target species.

At the beginning, we have the (undirected) PPI graphs of human $PPI_h = (V_h, E_h)$ and yeast $PPI_y = (V_y, E_y)$ (figure 1a, the blue graph corresponds to one reference species), a set of unassigned nodes $V_p$ which represents all P. anserina proteins (figure 1a, red nodes) and an empty preliminary PPI graph of P. anserina $PPI_{pre} = (V_p, E_p) = \emptyset$. The homology comparing approach leads to a directed homology graph

$$H = (V_H, E_H)$$

where

$$V_H \subseteq (V_h \cup V_y \cup V_p)$$

and

$$E_H \subseteq \{(a, b) \mid a \in V_p, b \in (V_y \cup V_h)\}.$$

Each edge in $H$ is weighted by the E-values (EV) of the BLAST search.

Before starting the algorithm we have to set three major parameters. First, the parameter $\epsilon_{thresh}$, which defines a threshold for the E-values. Homologous relations which exceed that threshold are immediately declined and will not be taken into account anymore. The second and third parameters are the upper and the lower bound for an E-value range $[h_l, h_u]$, where $h_l < h_u \leq \epsilon_{thresh}$ which will be mapped to the interval $[1, 0]$ by the scoring function $s$ where 1 is the best score and 0 the worst (figure 1b). Since generally, homology-based network inference has to consider that more homologous proteins exist, we aim to distinguish and weight
these relations by different scores provided by the scoring function. If only one homologous relationship exists, the scoring function will lead to the best score of 1, as well. This exception is due to the unambiguous and desired situation if only one protein in the target species is homologous to the corresponding protein in the reference species.

Each interaction of each reference species is handled one by one and contributes to the prediction and scoring with an equal weight of 1, i.e. we do not distinguish between the underlying experimental methods. Furthermore, it is assumed that interactions and the reference species are stochastically independent, i.e. the occurrence of one interaction does not influence the probability of the occurrence of another interaction. We start with the first reference species, here, $PPI_h$:

For each $e_h = \{a_h, b_h\} \in E_h$ (figure 1a, vertices $a_h$ and $b_h$):

1. Get $H_a \subseteq H$ with each $e_a = \{(a_{ca}, b_{ca}) | a_{ca}, b_{ca} \in V_H, b_{ca} = a_h\}$ and $H_b \subseteq H$ with each $e_b = \{(a_{cb}, b_{cb}) | a_{cb}, b_{cb} \in V_E, b_{cb} = b_h\}$ (figure 1a, dotted edges pointing to set $H_a$ or set $H_b$).

2. Predefinition: Given the graph $H$ and $V(H)$ gives the set of vertices in $H$ and $E(H)$ the set of edges in $H$ then $|V(H)|$ gives the number of vertices in $H$ and $|E(H)|$ the number of edges in $H$.

If $V(H_a) = \emptyset$ or $V(H_b) = \emptyset$, decline $e_h$, otherwise compute the homology score $s_h \rightarrow [0, 1]$ for each edge $e_a \in H_a$ and each edge $e_b \in H_b$ based on the single E-values and the cardinalities of $H_a$ and $H_b$:

If $|E(H_a)| = 1$ then $s_h(e_a) = 1$ else $s_h(e_a) = \frac{\log_{10}(EV(e_a)) + b}{m}$, otherwise compute the homology score $s_h(e_b) = \frac{\log_{10}(EV(e_b)) + b}{m}$ with $m = \frac{1}{\log_{10}(h_a) - \log_{10}(h_b)}$ and $b = -(\log_{10}(h_u))$ (figure 1b).

3. Decline edges which are now scored with 0, i.e. set $H_a = H_a \setminus \{(e_a \in H_a), s_h(e_a) = 0\}$ and $H_b = H_b \setminus \{(e_b \in H_b), s_h(e_b) = 0\}$.

4. Add each remaining vertex $\{v \in V(H_a \cup H_b) \land v \in V_p\}$ from target species to $PPI_{pre}$ and connect each node deduced from set $H_a$ with each node deduced from set $H_b$ (figure 1a, red edges). Compute a score $s_i$ for each of these new predicted edges (interactions) using the arithmetic mean:

\[ s_i = \frac{s_h(e_a) + s_h(e_b)}{2} \]

Decline edges where $s_i \leq s_{thresh}$ with $s_{thresh}$ as a predefined threshold.

Repeat step 1-4 with $PPI_y$ and each other reference species.

### 7.2.2 Combining PPIs deduced from each reference species

Now, the algorithm has to combine all predicted and probably redundant interactions in $PPI_{pre}$ to one combined PPI of $P. anserina$, $PPI_p = (V_p, E_p)$. In particular, it has to consider interactions which were suggested by both reference species PPI networks (figure 1c):

For each $e_i = \{a_{ci}, b_{ci}\} \in E(PPI_{pre})$:

1. Get $I_{a,b} \subseteq PPI_{pre}$ with each $e_j = \{a_{cj}, b_{cj} | a_{cj}, b_{cj} \in E(PPI_{pre}), ((a_{cj} = a_{ci}) \land (b_{cj} = b_{ci})) \lor ((a_{cj} = b_{ci}) \land (b_{cj} = a_{ci}))\}$, i.e. further interactions with the same interacting partners of $e_i$ (each edge X-Y in figure 1c).

2. Divide $I_{a,b}$ into subsets $S_h = I_{a,b}$ and $S_y = I_{a,b}$, depending on the reference species where the corresponding interactions have been found (black bordered areas in figure 1c).
3. Combine each score $s_i(s_h \in S_h)$ to one intra species score $s(S_h)$ and each score $s_i(s_y \in S_y)$ to one intra species score $s(S_y)$. Since a higher number of interactions increases the probability that $I_{a,b}$ exists in the target species, and the predicted interaction should be rated at least with the highest score, we sum up each score of each interaction in $S_h$ or $S_y$, respectively, using a recursive function where $s_k$ is the $k$th score in $s_i(s_y \in S_y)$. Additionally, $s_i(s_y \in S_y)$ has been sorted in a decreasing order:

$$s(s_k) = s_k \text{ if } k = 1,$$

$$f(s_k) = f(s_{k-1}) + (1 - f(s_{k-1})) * s_k \text{ otherwise.}$$

Hence, the intra species score is at least as high as the biggest sub score. For example, if we have three interactions deduced from one reference species, predicting all the same interaction in the target species with the scores 0.7, 0.9 and 0.6. Then, the final species score is:

$$s(0.6, 0.7, 0.9) = 0.9 + (1 - 0.9) * 0.7 + (1 - (0.9 + (1 - 0.9) * 0.7)) * 0.6 = 0.988.$$

4. Combine the two intra species scores to one common inter-species or final-score $i_{a,b}$ (figure 1c lower area):

$$i_{a,b} = \frac{s(S_h) + s(S_y)}{m} + m - 1,$$

where $m$ is the number of reference species for which the interaction $I_{a,b} \subseteq PPI_{pre}$ was found (i.e. $m = 1$, if only in yeast or human and $m = 2$, if in both).

5. Finally, add the new edge (interaction) $I_{a,b}$ to $PPI_p$ with the score $i_{a,b}$. 

17