Interaction Analyzer: A short manual

Version 1.0
November 2012

This manual gives a short introduction into the usage of the R package InteractionAnalyzer. InteractionAnalyzer enables the user to predict the activating or inhibiting effect of protein interactions (Suratanee *et al.,* submitted). The package is written for linux machines on which the R package is installed (version 2.15.0 or higher, R can be downloaded at www.r-project.org). Before running the script, please make sure to have the R libraries e1071, ROCR and MASS installed. You also need the GNU Linear Programming Kit (GLPK) which can be downloaded from http://www.ibm.com/developerworks/linux/library/l-glpk1/.

The package InteractionAnalyzer can be downloaded from http://www.ichip.de/software/InteractionAnalyzer.html. The web page provides the tarred and zipped file InteractionAnalyzer.tgz. This file needs to be unpacked on a linux console using the command

```
tar -zxf InteractionAnalyzer.tgz
```

This creates a new folder InteractionAnalyzer in which the main program main.R, this manual and four folders (Scripts, Data, Models, Results) can be found. The folder Scripts contains several programs which are needed by the main program, Data contains the needed input files, Models contains intermediate results and Results contains the computed final results for the user.

InteractionAnalyzer can be started by changing the current directory to the new directory InteractionAnalyzer, starting an R session and running the main program by

```
source("main.R")
```

The program needs several input files comprising features from the images and interaction lists. For illustration, the package contains interactions for training of the Signal Transduction pathways of the Kyoto Encyclopedia of Genes and Genomes (KEGG, http://www.genome.jp/kegg/ of April 2011), a list of non-annotated interactions from the database HIPPIE (Schaefer, et al., 2012) and image features of HeLa cells with knockdown of the corresponding genes. If you want to run your own data, just exchange these input files. The format of the input files will be explained below.

The example is quite CPU intensive and takes approximately 37 hours using one core of an IBM 3755 with sufficient RAM memory. This holds for the default setting in which no parameter optimization of the Support Vector Machines is performed (gamma =1/34, cost=1). If you want to optimize these parameters to improve the results, edit the function F_Classifier.R in the folder Scripts and set optimizeParams = 1 in the first

line (disregarding the documentation lines). Also the range can be set (variables rangeC and rangeG):

```
F_Classifier<-function(act_set=NULL, inh_set=NULL, Features=NULL,
FeaRData=NULL, optimizeParams = 0, rangeC = 2^(seq(-3,3,1)), rangeG =
2^(seq(-3,3,1)), save_model = 1, showROC = 0)
```

Here, the default parameters for gamma and cost will be used. If optimizeParams = 1, a grid search for gamma = { 2^{-3}, 2^{-2}, 2^{-1}, 2^{0}, 2^{1}, 2^{2}, 2^{3} } and cost = { 2^{-3}, 2^{-2}, 2^{-1}, 2^{0}, 2^{1}, 2^{2}, 2^{3} } will be performed testing 49 different combinations of gamma and cost. With this, the whole run takes appr. 52 hours on the same machine.

After successful execution of the program, one gets three output files (in the folder Results):

1. results.training.tab
This file contains the list of interactions from the training set, the class labels of the training set and voting scores of the predictions after cross-validation. This file is the data source for the ROC curve and can be used for own performance estimates.

2. results.prediction.tab
This file contains the list of interactions from the application set, the voting scores of the predictions of all trained classifiers and the predicted classes (activation, inhibition, no_prediction). The default thresholds for the classes are ≥ 920 votes for activation and ≤ 88 votes for inhibition. They can be changed in Scripts/F_Predicting.R in the first line (disregarding the documentation lines)

```
F_Predicting<-function(predict_set=NULL, Features=NULL, FeaRData=NULL,
model = NULL, modelRData = NULL, up_cutoff = NULL, low_cutoff = NULL)
```

by setting up_cutoff and low_cutoff to the wanted thresholds.

3. ROC.pdf
This file contains an ROC curve of the performance of the classier on the validation sets of the cross-validations.

Format of the input data files

If you want to analyze your own data, you need five input files. These input files are stored in three subfolders of the folder Data. The folder Data contains three subfolders: InteractionLists, ImageFeas, and Features, the input files are stored in InteractionLists, ImageFeas and Features. Please use the same, consistent annotation for your genes throughout all input files, so either unique gene symbols, or unique gene ids. The following tab delimited files are needed:

1) A file that contains a list of interactions with known effects (activation or inhibition). The file is located in subdirectory Data/InteractionLists/ and called interaction.training.tab.

   The format for each line is

   ```
   genex <tab> geney <tab> label
   ```

   in which genex and geney are the geneids (or symbols) of the interacting protein pair, label = 1 if the interaction has a known activating and = 2 if the interaction has a known inhibiting effect.

2) A file that contains a list of interactions for which new predictions should be made. The file needs to be located in the folder Data/InteractionLists and is called interaction.prediction.tab. The format for each line is

   ```
   genex <tab> geney
   ```

   in which genex and geney are the geneids (or symbols) of the interacting protein pair.

3) For each gene, at least one file of image features is needed. The images are from cells in which the corresponding gene was knocked down. These files are stored in the folder Data/ImageFeas. The file names are starting with the gene symbols (or gene ids) genex and need the following format:

   genex_set1

   We have the option that two sets of image features for the same genes can be analyzed, which in our case were sets from two different siRNA constructs (for the same genes). In this case, the two files need filenames ending with set1 and set2, respectively:

   genex_set1
   genex_set2

   The format of all these files is

   ```
   ImageId <tab> Feature1 <tab> Feature2 <tab> …
   ```

   ImageId is not needed by the algorithm and can be any annotation of the images, after this the feature values are listed (Feature1, Feature2, …).

4) A file is needed that contains additional features for each gene. We used fraction features (see Suratanee *et al.*, Supplementary Table S1). The file needs to be stored in Data/Features. The file is called fraction.features.tab. Its format is

   `genex <tab> Feature1 <tab> Feature2 <tab>…`

   genex needs to be consistent with the gene names/symbols used before.

5) Another file is needed that contains additional features for each gene. We used the maxima features (see Suratanee *et al.,* Supplementary Table S1). The file needs to be stored in Data/Features. The file is called maxima.features.tab. Its format is

   `genex <tab> Feature1 <tab> Feature2 <tab>…`

   genex needs to be consistent with the gene names/symbols used before.

References

Schaefer, M.H.*, et al.* (2012) HIPPIE: Integrating protein interaction networks with experiment based quality scores, *PloS one*, **7**, e31826.