This document is served as the manual to build a new version of FI network. In this version, a random forest will be used to predict functional interactions between any pair of proteins/genes.

**Set up Eclipse and Load FINetworkBuild Project**

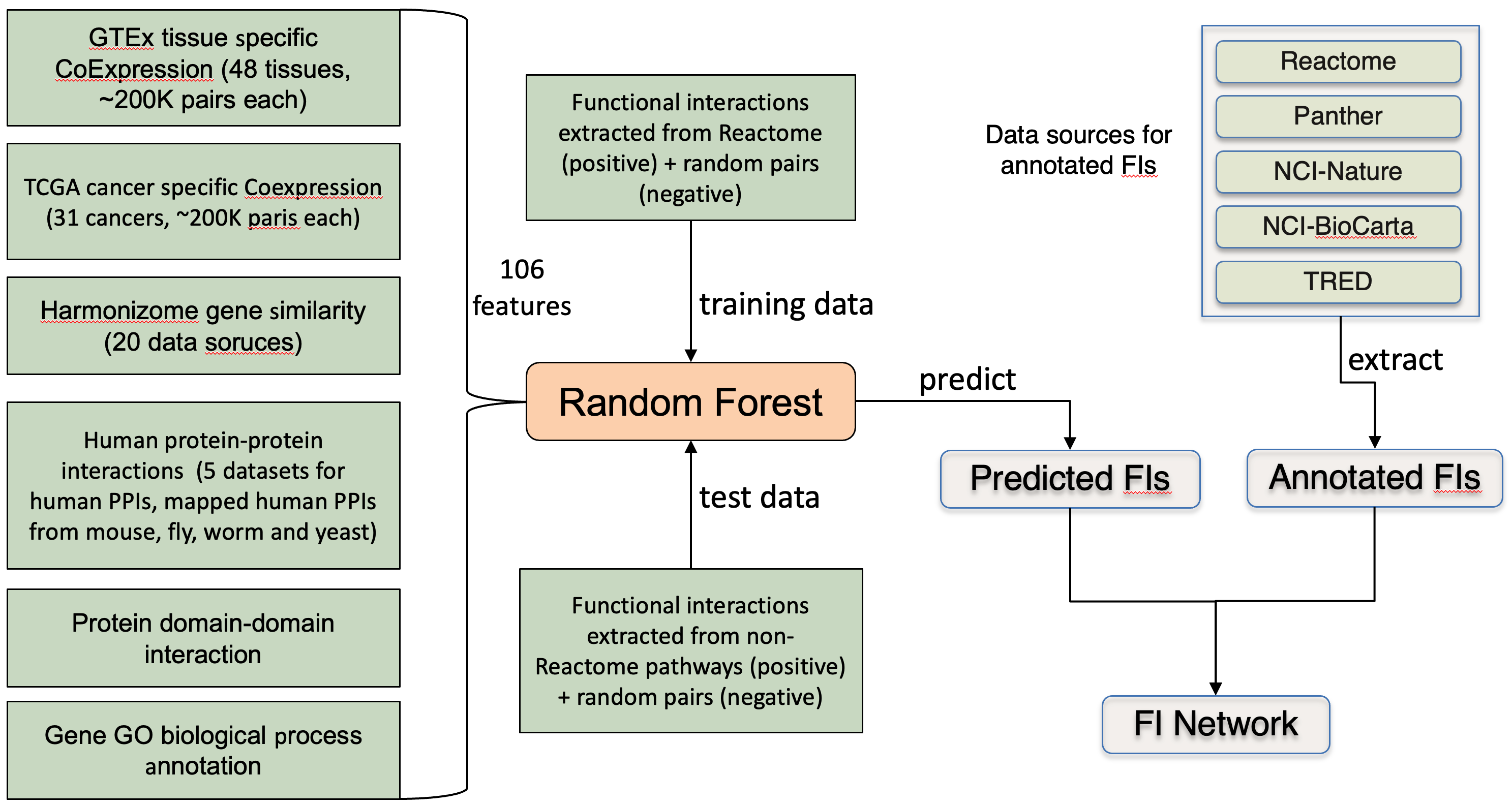
The project used to build the FI network is mainly programmed in Java. But the code for random forest training and the FI prediction is in Python. Eclipse has been used for all Java related programming. For this project, we have not built a project that can run independent of Eclipse. So you need to download Eclipse from http://www.eclipse.org, set it up and download the FINetworkBuild\_RF project from <https://github.com/reactome-fi/fi_network_build_rf>. The Python code used to train RF and predict FIs is hosted at <https://github.com/reactome-idg/fi-network-ml/tree/master/scripts/ml>. It is planned to move the related code to the fi\_network\_build\_rf project.

**Notes:**

1. The configuration file (file name: configuration.prop) in the resources folder is used to configure files and some parameters used by the project. Make sure these three values in the file are correct: YEAR, RESULT\_DIR, and DATA\_SET\_DIR.
2. YEAR is the year when the FI network is built.
3. RESULT\_DIR is used to hold results generated from the project.
4. DATA\_SET\_DIR is used to hold the downloaded files and other source files for the project. CHECK\_ANY\_NEW\_DATASET

**Overview of Workflow**

The workflow to build a functional interaction (FI) network is described in Figure 1. Some changes may be made version by version. So the following Figure may be updated periodically.



*Figure 1. Workflow to build the Reactome FI network.*

Two types of data sources are used: Data sources for predicted FIs and data sources for annotated FIs. The data sources for predicted FIs are based on the original Reactome IDG project (see details for more details, <https://www.biorxiv.org/content/10.1101/2023.06.05.543335v1>). The data sources for annotated FIs are several human curated pathway databases.

**Data Sources for id mapping**

**UniProt**

UniProt is used as the protein reference database in the FI network construction. We need to download the latest version of UniProt so other data sets can be mapped to it. The download site for UniProt is: [ftp.uniprot.org](ftp://ftp.uniprot.org). You don’t need any registration. We only need to download the human data set.

1. After log into [ftp.uniprot.org](ftp://ftp.uniprot.org), download two files from current\_release/knowledgebase/taxonomic\_divisions/: uniprot\_sprot\_human.data.gz and uniprot\_trembl\_huma.dat.gz. You may also download relnotes.txt or README files for future reference. After download these files, unzip them, and place them into a sub-directory in the dataset folder. Change the configuration value in the configuration file for UNIPROT\_DIR to point to this uniprot directory.

# Used to control the UniProt version

UNIPROT\_DIR=${DATA\_SET\_DIR}/UniProt/release\_2024\_02/

1. Besides these files in step 1, you also need to download the fasta file for Isoform sequences to be used in normalizing features for NBC training and predictions from this web site: <https://www.uniprot.org/downloads>: Isoform sequences: the file name should be called “uniprot\_sprot\_varsplic.fasta” after downloaded.

**IPRO\_CLASS**

IPROCLASS\_HUMAN\_FILE

A mapping file is needed to map entrez gene id to UniProt accession number. We need to download an original mapping file from the PIR web site, and will do some pre-process using a Java method later on.

1. Download the mapping file from this ftp site: [ftp.proteininformationresource.org/databases/](ftp://ftp.proteininformationresource.org/databases/). Go to idmapping/mapping\_by\_sp, and download file h\_sapiens.tb. Remember where you place your file.
   1. The ftp address may change. Check <https://proteininformationresource.org/pirwww/download/> to find out the address. Under Mac OS, don’t enter any password after enter ‘anonymous’.
2. Modify the following two values in the configuration file:

# This file is used to map Entrez id to UniProt accession number

IPROCLASS\_HUMAN\_FILE = ${DATA\_SET\_DIR}/iproclass/${DATE}/h\_sapiens.tb

ENTREZ\_TO\_UNIPROT\_MAP\_FILE\_NAME=${DATA\_SET\_DIR}/iproclass/${DATE}/EntrezToUniProt.txt (Note: This file will be produced by code)

**Data Sources for Annotated FIs**

Annotated FIs are FIs extracted from human curated pathways. Multiple pathway databases have been used. Pathways in other non-Reactome databases are converted to curator tool project first, and then dumped into a modified gk\_central Reactome databases. Different file formats are used for different databases.

**Reactome:**

Reactome is used as the foundation for the FI network build. The publicly released mysql database is used so that both human and mouse pathways are coverted.

1. Download the mysql dump file from Reactome’s download site, <https://reactome.org/download-data> (Search for main). The download link usually should be, <https://reactome.org/download/current/databases/gk_current.sql.gz>.
2. Create a Reactome database: Copy the above dump file into your build computer (e.g. your desktop machine), unzip it using the following command if it is not unzipped automatically:

*jar –xvf gk\_current.sql.zip*

Log into your mysql database using a user account that can create a database. Create a database as following:

*mysql> create database reactome\_{release\_number}\_plus\_i;*

Here release\_number should be replaced by actual release number (e.g. 79). The release number should be the number in the slice database. Note: Reactome has a quarterly release schedule.

After the empty database is created, run the following two commands to load the generated slice database dump:

*mysql> use reactome\_{release\_number}\_plus\_i;*

*mysql> source test\_slice\_{release\_number}.sql*

Note: In order to make sure mysql can find the dump file, start mysql from the directory containing the unzipped mysqldump file. Or you have to use absolute path to your dump file (no quotation marks)

1. Make changes to the configuration file in the source folder for these three properties: REACTOME\_SOURCE\_DB\_NAME, DB\_USER and DB\_PWD, so that the expanded Reactome database can be used in the future as follows:

# local database that is used as the data source

REACTOME\_SOURCE\_DB\_NAME=reactome\_88\_plus\_i // This should be updated for each build

DB\_USER={change\_to\_your\_own\_user\_name}

DB\_PWD={change\_to\_your\_own\_mysql\_pwd}

1. The FI network may need the whole set of human SwissProt records. However, the slice database contains a subset of them only. Following steps 1 and 2, get a copy of gk\_central at curator.reactome.org into your local desktop. The database name has been assumed as following in the configuration file:

REACTOME\_GK\_CENTRAL\_DB\_NAME=gk\_central\_120413

1. Modify the loaded Reactome database (named by REACTOME\_SOURCE\_DB\_NAME) by running JUnit methods in class org.reactome.data. ReactomeDatabaseModifier:
   1. Before running any Java method, create an empty text file for keeping all logging output in the RESTULT directory. The file name should be called “Combined\_Logging.txt”.
   2. For transaction protection, we need to use InnoDB. Run JUnit method changeMyISAMToInnodb() in the class. You should see logging output as following:

2013-12-05 10:53:37,084 [main] INFO org.reactome.data.ReactomeDatabaseModifier - Drop full text index: Affiliation.address

2013-12-05 10:53:37,318 [main] INFO org.reactome.data.ReactomeDatabaseModifier - Drop full text index: Affiliation\_2\_name.name

2013-12-05 10:53:37,442 [main] INFO org.reactome.data.ReactomeDatabaseModifier - Drop full text index: Book.ISBN

……

2013-12-05 10:55:35,218 [main] INFO org.reactome.data.ReactomeDatabaseModifier - Alter Table to InnoDB: \_InstanceBeforeChange

2013-12-05 10:55:35,424 [main] INFO org.reactome.data.ReactomeDatabaseModifier - Alter Table to InnoDB: \_InstanceBeforeChange\_2\_attributeValuesBeforeChange

2013-12-05 10:55:35,574 [main] INFO org.reactome.data.ReactomeDatabaseModifier - Alter Table to InnoDB: \_Release

* 1. **(Don't’ forget to run this)** Copy human ReferenceGeneProducts that are not in the slice database into the generated Reactome\_plus\_i database by running method, copyHumanReferenceGeneProducts(). (Note: assign 12G as –Xmx). Some of outputs are copied below:

2015-12-07 15:49:41,602 [main] INFO org.reactome.data.ReactomeDatabaseModifier - Total human ReferenceGeneProduct in the source database: 41585

2015-12-07 15:49:41,937 [main] INFO org.reactome.data.ReactomeDatabaseModifier - Copy [ReferenceGeneProduct:244824] UniProt:Q8NDT2 RBM15B

2015-12-07 15:49:42,542 [main] INFO org.reactome.data.ReactomeDatabaseModifier - Copy [ReferenceIsoform:797706] UniProt:B6A8C7-1 TARM1

2015-12-07 15:49:42,739 [main] INFO org.reactome.data.ReactomeDatabaseModifier - Copy [ReferenceIsoform:410879] UniProt:Q8IX30-1 SCUBE3

……

2015-12-07 16:07:56,382 [main] INFO org.reactome.data.ReactomeDatabaseModifier - Copy [ReferenceGeneProduct:385012] UniProt:Q3ZLR7 SUPT20HL1

2015-12-07 16:07:56,396 [main] INFO org.reactome.data.ReactomeDatabaseModifier - Copy [ReferenceIsoform:402121] UniProt:O95263-1 PDE8B

2015-12-07 16:07:56,416 [main] INFO org.reactome.data.ReactomeDatabaseModifier - Copy [ReferenceIsoform:246797] UniProt:Q59EK9-4 RUNDC3A

2015-12-07 16:07:56,434 [main] INFO org.reactome.data.ReactomeDatabaseModifier - Copy [ReferenceIsoform:356555] UniProt:Q2Y0W8-6 SLC4A8

2015-12-07 16:07:56,452 [main] INFO org.reactome.data.ReactomeDatabaseModifier - Copy [ReferenceIsoform:355421] UniProt:Q2KHT4-4 GSG1

2015-12-07 16:08:07,491 [main] INFO org.reactome.data.ReactomeDatabaseModifier - Total copied: 32655

1. Modify Reactome database schema: the regular Reactome database schema needs to be expanded for the FI network. A new attribute called “dataSource” should be added to the top-level class, “DatabaseObject”. Two new classes, Interaction and TargetedInteraction should be created.

*mysql> source {absolute\_path\_to}/SchemaModification.sql*

Note: You should find a copy of SchemaModifcation.sql in resources folder.

After a successful run of the above sql, check if two new classes (Interaction and TargetedInteraction), and new attribute dataSource for DatabaseObject have been created by connecting your curator tool to your newly created database. Check the database schema view to make sure these three items are there.

In order to use this new expanded Reactome schema for future procedures, we need to export the new schema from the database into a local computer. In the curator tool, after connecting to the modified database, do the following:

In the database schema view, choose “Export Schema” in the File menu, and save the schema into the resource folder in your project folder. This schema file will be needed by the Reactome Java API, which is used extensively in the following procedures.

Note: the file name has to be “schema” without any extension. Don’t change it!

**KEGG: As of the 2024 version, KEGG is not used anymore because of the license issue!**

**NCI-PID**

There are two kinds of pathways available in this database: pathways curated by NCI-PID curators and pathways imported from BioCarta and Reactome. Indeed, we don’t need to import Reactome pathways here. BioCarta pathways have not been updated since they were imported in this database. But because the changes of the Reactome schema, these pathways should be re-imported each time if we want to use them.

1. Download pathways from NCI-PID: The data format we are going to use is BioPAX level 2. Download these two BP2 files in its download site, <http://pid.nci.nih.gov/download.shtml>: NCI-Nature Curated Data (BioPAX Level 2), and BioCarta data (BioPAX Level 2). After unzipping, the file names should be:
   1. NCI-Nature\_Curated.bp2.owl
   2. BioCarta.bp2.owl

Note: Since June, 2012, the BP2 file for curated pathways include pathways imported from both Reactome and BioCarta. So this file cannot be used any more. Since there are not many new pathways added, the file downloaded in January, 2012 is used (If you don’t have these files, ask Guanming). This issue needs to be monitored. However, an email using the feedback link was bounced back!!!

Note: On December 15, 2014, the last update is still in December, 2012. So the old files were still used for the 2014 version and afterwards of the FI network.

1. Modified the following configurations in the configuration file:

# Used for the Nature-PID database files

NATURE\_PID\_DIR=${DATA\_SET\_DIR}/NCI-Pathways/011612/

NATURE\_PID\_CURATED=${NATURE\_PID\_DIR}/NCI-Nature\_Curated.bp2.owl

NATURE\_PID\_CURATED\_CONVERTED=${NATURE\_PID\_DIR}/NCI-Nature\_Curated.bp2.rtpj

NATURE\_PID\_BIOCARTA=${NATURE\_PID\_DIR}/BioCarta.bp2.owl

NATURE\_PID\_BIOCARTA\_CONVERTED=${NATURE\_PID\_DIR}/BioCarta.bp2.rtpj

1. Find the BioPAX to Reactome mapping file: resources/BioPAXToReactomeMappers.xml. Make sure the following XML element is **NOT** commented out so that we can use NciPIDBToRPostProcessor class in package org.reactome.b2rPostProcessor during mapping:

<!-- This PostProcessor is for NCI Pathways -->

<postProcessor class=*"org.reactome.b2rPostProcessor.NciPIDBToRPostProcessor"* />

**Panther**

Pathways in the Panther database have not been updated a lot. However, because our Reactome schema has been evolved continually, we still need to re-convert pathways from Panther into the Curator Tool project in order to merge with other pathways.

1. Download Pathways: download pathways from Panther’s ftp site, [ftp.pantherdb.org/pathway/current\_release](ftp://ftp.pantherdb.org/pathway/current_release). Please download these two files: SBML\_{version}.zip and SequenceAssociationPathway3.0.1.txt. We use the SBML files, which are the file format used by the Panther annotation. The second file is used to map pathway component ids used in pathways into UniProt accession numbers.

Notes: 1). Annotation at Panther is very slow. However, some pathways have been added into their pathway databases. As of December, 2013, the latest release is 3.2.1 containing 176 pathways (2520 pathway components). The 3.0.1 has 165 pathways (2408 pathway components). It will be nicer to check the latest version of panther pathways, if significant new pathways have been added, probably a new download should be used.

2). One pathway component used in panther pathways can be mapped to multiple UniProt identifiers because these components are actually based on HMM families constructed by Panther. In order to control the quality of imported pathways, we use mappings based on the following reliable confidence only:

"IGI",

"IPI",

"IDA",

"IEP",

"TAS",

"IC",

"IMP",

"RCA"

Notes: 1). As of April 15, 2024, the latest release is 3.6.7, released in September, 2023. However, only BioPAX files are available. We need to update the code to handle Panther BioPAX dump.

1. Modify the following values in the configuration file related to Panther:

# Used for the panther database files

# Download on Jan 18, 2011

PANTHER\_DIR=${DATA\_SET\_DIR}/Panther/Version3.0.1/

PANTHER\_FILES\_DIR=${PANTHER\_DIR}/SBML/

PANTHER\_MAPPING\_FILE=${PANTHER\_DIR}/SequenceAssociationPathway3.01.txt

PANTHER\_CONVERTED\_FILE=${PANTHER\_DIR}/Panther\_3\_0\_1\_120513.rtpj

Note: A file called “SpeciesToChEBIId.txt” in the resources folder has been used for converting. This file is old and has not been updated. This may be fine considering all Panther pathways have not been updated for quite a while.

**TRED**

TRED provides us transcription factors (TF) and their targets interactions. Many of interactions in this database are computationally predicted. We extract only manual curated interactions, which have been supported by literatures if not considering curation errors. The content in the TRED database should not be updated any more. What we need to do is just convert the data in the TRED database into a Curator tool project using the latest Reactome data model. To fetch the data from the TRED database, we use a customized hibernate based API. So have to make sure you have set up your class path correctly in order to use this hibernate API.

1. Install the TRED database into your local computer using the TRED.sql dump file.

Note: There is a back-up file in data\_archive/tred/TRED.sql.zip, which can be used to load the database.

1. Configure the hibernate configuration file: Find file TREDHibernate.cfg.xml in the resources folder. Make changes to these three properties so that they point to the correct values:

<property name=*"connection.url"*>jdbc:mysql://localhost:3306/TRED</property>

<property name=*"connection.username"*>root</property>

<property name=*"connection.password"*>macmysql01</property>

1. Make changes to these two values in the configuration file:

# For TRED files

TRED\_DIR=${DATA\_SET\_DIR}/TRED/

TRED\_CONVERTED\_FILE=${TRED\_DIR}/TRED\_${DATE}.rtpj

**ENCODE**

Encode TF/target interactions were generated by the Geistein’s group in Yale. The interaction files we have used are based on a Nature publication (TODO: add a link to the nature paper), and downloaded from http://archive.gersteinlab.org/proj/encodenets/ (This link is not accessible right now). For our network built, a pre-processed file by combing both proximity and distal interaction together has been generated and stored in our Git repository: data\_archive/encode/tf-targets.txt.

1. Synchronize the above interaction file into your local project.
2. Created a new directory called “encode” in the {DATA\_SET\_DIR} directory, and create a link to the above tf-targets.txt file.
3. Make changes (or Add) the following configurations related to ENCODE:

# ENCODE data

ENCODE\_DIR=${DATA\_SET\_DIR}/encode

ENCODE\_TFF\_FILE=${ENCODE\_DIR}/tf-targets.txt

ENCODE\_TFF\_CONVERTED\_FILE=${ENCODE\_DIR}/tf-targets\_${DATE}.rtpj

Note: There are many TF/Target interactions in the original ENCODE project release. However, many of them may not be related to actual biological functions since they are physical interactions basically. We use a simple filter to pick up TF/Target interactions that are supported by gene co-expression and/or GO BP annotation sharing.

**Data sources for predicted FIs**

The following datasets are used to train a random forest and then use the trained random forest (RF) to predict functional interactions for all human protein pairs. The workflow is ported from the one used to build the Reactome IDG web portal.

**Panther Orthologous Mapping**

The files downloaded from panther orthology are used to map proteins from non-human model organisms (MODs) to humans for PPIs datasets.

1. URL: <http://data.pantherdb.org/ftp/ortholog/>
2. <http://data.pantherdb.org/ftp/ortholog/current_release/RefGenomeOrthologs.tar.gz> to download the file
3. Run the following command at termina to extract human genes related mapping file:

grep '^HUMAN' RefGenomeOrthologs > HUMAN\_RefGenomeOrthologs

1. Note: We use this mapping file from all species to human. This is different from the original Reactome IDG workflow, which used ensemble mapping file for mouse PPIs to increase the coverage. Here we’d like to emphasize the consistence. Therefore, use the same mapping files.

**Protein-Protein Interactions (PPIs)**

PPIs data are download from the three data sources, StringDB, BioGrid, and BioPlex (human PPIs only).

1. Human protein-protein interactions
   1. StringDB:
      1. URL: <https://string-db.org/cgi/download>
      2. Files:
         1. 9609.protein.links.full.v12.0.txt (Note: the version may be different)
         2. 9606.protein.info.v12.0.txt (Note: Supposed to use file human.name\_2\_string.tsv. But this file is missing for release v12.0. Therefore, we will generate this file from this info text file. The code needs to be modified in order to use this file.)
      3. Since we will use gene names only for human, therefore, we use the info, instead of aliases, file for mapping. The code to load this file is different from others.
   2. BioGrid:
      1. URL: <https://downloads.thebiogrid.org/BioGRID/Release-Archive/BIOGRID-4.4.232/>
      2. Files:
         1. BIOGRID-ORGANISM-4.4.232.tab2.zip (Need to unzip the file to get the file for homo sapiens)
      3. Note: We are using the tab2 format from BioGrid!
   3. BioPlex:
      1. URL: <https://bioplex.hms.harvard.edu/interactions.php#datasets>
      2. Files:
         1. BioPlex\_293T\_Network\_10K\_Dec\_2019.tsv
         2. BioPlex\_HCT116\_Network\_5.5K\_Dec\_2019.tsv
         3. Note: Most likely these two files will not change. Check to make sure!
2. Yeast PPIs
   1. StringDB:
      1. URL: <https://string-db.org/cgi/download?sessionId=bytaIMHiqR2a&species_text=Saccharomyces+cerevisiae>
      2. Files:
         1. 4932.protein.links.full.v12.0.txt
         2. 4932.protein.aliases.v12.0.txt.gz
   2. BioGrid:
      1. URL: Same as the human (above)
      2. File: BIOGRID-ORGANISM-Saccharomyces\_cerevisiae\_S288c-4.4.232.tab2.txt
3. Fly PPIs:
   1. StringDB:
      1. URL: <https://string-db.org/cgi/download?sessionId=bytaIMHiqR2a&species_text=Drosophila+melanogaster>
      2. Files:
         1. 7227.protein.aliases.v12.0.txt.gz
         2. 7227.protein.links.full.v12.0.txt.gz
   2. BioGrid:
      1. File: BIOGRID-ORGANISM-Drosophila\_melanogaster-4.4.232.tab2.txt
4. Worm PPIs:
   1. StringDB:
      1. URL: <https://string-db.org/cgi/download?sessionId=bytaIMHiqR2a&species_text=Caenorhabditis+elegans>
      2. Files:
         1. 6239.protein.links.full.v12.0.txt
         2. 6239.protein.aliases.v12.0.txt
   2. BioGrid:
      1. File: BIOGRID-ORGANISM-Caenorhabditis\_elegans-4.4.232.tab2.txt
5. Mouse PPIs:
   1. StringDB:
      1. URL: <https://string-db.org/cgi/download?sessionId=bytaIMHiqR2a&species_text=Mus+musculus>
      2. Files:
         1. 10090.protein.links.full.v12.0.txt
         2. 10090.protein.aliases.v12.0.txt
   2. BioGrid:
      1. File: BIOGRID-ORGANISM-Mus\_musculus-4.4.232.tab2.txt
6. To map PPIs from other species to human, we also need the biogrid id file:
   1. URL: <https://downloads.thebiogrid.org/BioGRID/Release-Archive/BIOGRID-4.4.232/>
   2. File: BIOGRID-IDENTIFIERS-4.4.232.tab.zip
   3. To make the file smaller and easy to use, run the following Java method in the idg-fi-network-ml after updating the following two keys in its application.properties, org.reactome.idg.ppi.BioGridHandler.processIdentifierFiles()

biogrid.id.file=/Users/wug/datasets/BioGrid/BIOGRID-IDENTIFIERS-3.5.181.tab.txt

biogrid.id.file.selected=/Users/wug/datasets/BioGrid/BIOGRID-IDENTIFIERS-3.5.181.selected.tab.txt

Note: There is no necessary we have to run this method. If not, make sure the above two keys point to the same file.

**Gene-Expression Correlations**

Gene-coexpression features are based on TCGA and GEO datasets. These datasets are supposed to be stable. Therefore, no need to repeat the correlation calculation. For archiving purpose, the procedures to build these features are copied below from the IDG bioRxiv paper:

------

We collected a dataset of 106 protein/gene pairwise relationship features to train a random forest model for predicting functional interactions of proteins. The majority of these features were obtained from gene coexpressions derived from bulk RNA-seq data obtained from the GTEx and TCGA projects. Tissue-specific gene expression data was downloaded from the GTEx portal, <https://www.gtexportal.org/home/datasets>. The file we used was the gene read counts file, GTEx\_Analysis\_2017-06-05\_v8\_RNASeQCv1.1.9\_gene\_reads.gct.gz. Samples with RIN values less than or equal to 6.0 and tissues with fewer than 30 samples were filtered out. Cancer-specific RNA-seq gene expression data was acquired in November, 2019 using a customized Python script that utilized the GDC API as described in this document, <https://docs.gdc.cancer.gov/API/Users_Guide/Getting_Started/>. The script is available at <https://github.com/reactome-idg/gather-app/blob/master/python/switch/tcga_gather.py>. Outlier analysis based on PCA's loading matrix (<https://github.com/reactome-idg/gather-app/blob/master/python/scripts/R/functions.R>) was conducted, and samples with z-scores of the first PC (principal component) greater than or equal to 3 were marked as outliers. The percentage of outliers in each tissue or cancer was less than 5%. The gene counts were normalized to cpm (count per million) values and then subjected to pairwise Spearman correlation analysis after removing outliers. A total of 50 and 32 correlation matrices were generated for GTEx and TCGA, respectively. Additionally, we created a new skin dataset by merging two skin datasets together, Skin-NotSunExposed-Suprapubic and Skin-SunExposed-Lowerleg, as an internal control. The code used to download and process these two datasets is hosted at <https://github.com/reactome-idg/gather-app>.

------

**Gene Expression Data for Attaching Evidence to ENCODE TF/Target interactions**

The following two old microarray-based gene expression datasets are still used to attach some evidence to TF/Target interactions from ENCODE. Most likely we will prepare to use better way to collect TF/Target interactions in the future! These two datasets are: Lee’s Gene Expression and Prieto’s Gene Expression. The original data files were downloaded from their paper’s web sites, and should not be changed any more. However, we need to update protein-pairs to latest version of UniProt.

Note: The original downloaded data files contain gene pairs only. In order to be used as features in NBC, we have to map gene names to UniProt accession numbers. We use downloaded UniProt data file to do this mapping. We use the swissprot part of UniProt data for doing the mapping for the Prieto data file. For Lee’s data file, we use the original downloaded mapping file and normalize with the latest UniProt data.

1. Get the archived source files: data\_archive/data\_archive.zip. Unzip this zipped file. You should find two files: GeneExpWith3FromPavlidis.txt under the LeeGeneExp folder, and union60.txt in the PrietoGeneExp folder.
2. Make sure these two constants in the configuration file have been set correctly (usually you don’t need to make a change):

# Two Gene expression data files

LEE\_GENE\_EXP\_FILE\_SOURCE=${DATA\_SET\_DIR}/microarray/Pavlidis/GeneExpWith3FromPavlidis.txt;

LEE\_GENE\_EXP\_FILE=${RESULT\_DIR}/LeeGeneExp.txt

PRIETO\_PAIRS\_FILE=${DATA\_SET\_DIR}/microarray/PrietoCarlos/union60.txt

PRIETO\_GENE\_EXP\_FILE=${RESULT\_DIR}/PrietoGeneExp.txt

**Gene Similarity Data**

These data were downloaded from Harmonizome. Apparently the datasets are added gradually there. The current release is 3. However, no release history is provided there. For the time being, we will use files downloaded in 2020, which were used to build the idg.reactome.org portal. We may update these similarity files for the future FI network construction. The procedures used to build these features are copied below from the IDG preprint:

------

Gene similarity data was downloaded from Harmonizome, <https://maayanlab.cloud/Harmonizome/download>. To integrate this data into our workflow, we ported the Python script at its download website into Java. We manually selected a subset of datasets that were likely to provide pathway-related information, while excluding datasets related to gene expressions or non-human data. For details on the selected datasets, please refer to the “**harmonizome\_datasets\_annotations\_062819.xlsx**” file in **Supplemental Results**.

------

**Gene Ontology Annotation**

The sharing of GO annotation (Biological Process only) is used as another feature in our ML training and prediction. In our usage, we check a simple sharing only without considering any parent-child relationships among GO terms.

1. URL: <http://current.geneontology.org/products/pages/downloads.html>
   1. File: <https://current.geneontology.org/annotations/goa_human.gaf.gz>
   2. Homo sapiens (EBI Gene Ontology Annotation Database, protein)
2. A GO term and id mapping is needed in converting BioPAX for BioCarta pathways, and be used for GO enrichment analysis for ReactomeFIViz. Download go.obo file via <https://purl.obolibrary.org/obo/go.obo>.
   1. Note: Based on this file, another file, GO.terms\_and\_ids will be generated by the workflow.
3. Modify the following value in the configuration file to point to the correct directory that holds the above downloaded file:

# For GO related files

GO\_DIR=${DATA\_SET\_DIR}/GO/${DATE}/

GOA\_FILE\_NAME=${GO\_DIR}/goa\_human.gaf

**Domain-Domain Interactions**

We use domain-domain interactions from the pFam database.

1. Download domain interaction file from pFam: go to pFam release ftp site, f <https://ftp.ebi.ac.uk/pub/databases/Pfam/releases/>, go to the latest release folder, and then go to database\_files. Download three files: pfamA.txt.gz and pfamA\_interactions.txt.gz, and version.txt.gz, unzip them.
2. Specify the following in the configuration file as being used in step 2:

# Directory for pFam

PFAM\_DIR\_NAME=${DATA\_SET\_DIR}/Pfam/33.1/

Note: The latest version we can find and has the pfamA\_interactions.txt.gz file is release 33.1 (released on May, 2020). This needs to be investigated more. For the time being, this release is used (FI version 2024).

1. Make sure to copy pfamA\_interactions.txt by running org.reactome.data.PfamAnalyzer.convertIntToPfamIDs().

**Functional Interaction Network Build**

Run the following methods in order in class org.reactome.fi.FINetworkBuilder

Notes: 1). It is suggested to copy all logging output from Eclipse in a file for future reference. A file Combined\_Logging.txt is created for this purpose.

2). Make sure all directories configured should be there in the dataset directory. Otherwise, exceptions may be thrown.

1. **prepareMappingFile()**

Example logging from running this method like this:

2024-04-16 13:56:39,418 [main] INFO org.reactome.fi.FINetworkBuilder - Running UniProtAnalyzer.generateUniProtIDsMap()...

total entries in uniprot: 275898

total entries in uniprot: 90513

2024-04-16 13:56:41,811 [main] INFO org.reactome.fi.FINetworkBuilder - Running UniProtAnalyzer.generateEntrezGeneToUniProt()...

Counter: 94968

2024-04-16 13:56:42,208 [main] INFO org.reactome.fi.FINetworkBuilder - Running UniProtAnalyzer.generateUniToPfamMap()...

2024-04-16 13:56:43,467 [main] INFO org.reactome.fi.FINetworkBuilder - Generating GO.terms\_and\_ids.txt from go.obo...

After this method running, 5 files should be generated as following:

Uni2Pfam.txt in the UniProt directory

SwissProtACIDMap.txt in the UniProt directory

ACIDMap.txt in the UniProt directory

ENTREZ\_TO\_UNIPROT\_MAP\_FILE\_NAME as configured (e.g. ${DATA\_SET\_DIR}/iproclass/122816/EntrezToUniProt.txt)

GO.terms\_and\_ids.txt in the configured GO directory

1. **convertPathwayDBs()**

Convert Pathways in NCI-PID, and Panther, and TF/Target interactions into individual curator tool project files. You may need to open these converted project files in the curator tool to see how they look like and make sure they are correct.

Notes: 0). Make use to change the URL for BioPAX level 2 from http to https for the two old BioPAX files from NCI-PID. The two files used for FI\_2024 have been edited and should be good.

1). Before running this method, make sure all jar files are in the local maven repo. Read READE.md to install jar files used by protégé. Otherwise, it will not work!

2). In order to keep the logging file, make sure log4j.prop is configured to write output to a file.

3). Make sure to assign enough memory for running this method (e.g. –Xmx10G).

4). ReferenceIsoform instances may have been fetched out from the database during converting. However, our converting cannot take use of isoforms yet. So the Isoform instances should be treated as the top-level ReferenceGeneProduct instance!

5). For converting the BioCarta pathways in the NCI-PID pathway, a file called GO.terms\_and\_ids.txt in the resource folder has been used, which is required by BioPAXToReactomeConverter in another project, PathwayExchange. This file will be copied from the GO directory to the resources directory so that an updated version can be used by the method automatically (Note: This step is handled by the program, there is no need to copy it manually. Check to make sure!).

6). For converting pathways in the Panther database, a file called “SpeciesToChEBIId.txt” in the resources folder is used. This file is old and has not been updated. This may be fine considering all Panther pathways have not been updated for quite a while.

7). Starting with Version 2020, literature references used by Panther are not processed any more since the NCBI EUtil API used by auto-fetching detailed pubmed information is just too slow. Also the detailed information for imported panther pathways are not really investigated.

8). For TRED database, we use gene names, which are used in the TRED database, in our target Reactome database directly to map to UniProt. Some of gene names in TRED cannot be mapped. See below (partial list):

……

2013-12-06 15:01:15,278 [main] WARN org.reactome.convert.common.PostProcessTemplate - PWCR1 for EWAS -4791 cannot be mapped to UniProt!

2013-12-06 15:01:16,685 [main] WARN org.reactome.convert.common.PostProcessTemplate - NYD-SP28 for EWAS -5071 cannot be mapped to UniProt!

2013-12-06 15:01:18,931 [main] WARN org.reactome.convert.common.PostProcessTemplate - HMG-I for EWAS -5546 cannot be mapped to UniProt!

2013-12-06 15:01:21,482 [main] WARN org.reactome.convert.common.PostProcessTemplate - MIG-6 for EWAS -6256 cannot be mapped to UniProt!

2013-12-06 15:01:21,648 [main] WARN org.reactome.convert.common.PostProcessTemplate - LOC339524 for EWAS -6278 cannot be mapped to UniProt!

2013-12-06 15:01:25,102 [main] WARN org.reactome.convert.common.PostProcessTemplate - FLJ23356 for EWAS -7005 cannot be mapped to UniProt!

2013-12-06 15:01:26,128 [main] WARN org.reactome.convert.common.PostProcessTemplate - col4a1 for EWAS -7393 cannot be mapped to UniProt!

2013-12-06 15:01:26,128 [main] WARN org.reactome.convert.common.PostProcessTemplate - col4a2 for EWAS -7397 cannot be mapped to UniProt!

2013-12-06 15:01:26,882 [main] WARN org.reactome.convert.common.PostProcessTemplate - RPS3a for EWAS -7709 cannot be mapped to UniProt!

2013-12-06 15:01:30,853 [main] WARN org.reactome.convert.common.PostProcessTemplate - LOC285847 for EWAS -8409 cannot be mapped to UniProt!

2013-12-06 15:01:31,805 [main] WARN org.reactome.convert.common.PostProcessTemplate - TRD@ for EWAS -8742 cannot be mapped to UniProt!

2013-12-06 15:01:31,944 [main] INFO org.reactome.convert.common.PostProcessTemplate - Total unmapped names: 48

9). For the Encode interaction, we got much more unmapped gene names:

……

2013-12-06 15:09:17,791 [main] WARN org.reactome.convert.common.PostProcessTemplate - LOC100289673 for EWAS -52795 cannot be mapped to UniProt!

2013-12-06 15:09:18,127 [main] WARN org.reactome.convert.common.PostProcessTemplate - LOC283314 for EWAS -52847 cannot be mapped to UniProt!

2013-12-06 15:09:19,042 [main] WARN org.reactome.convert.common.PostProcessTemplate - SAA2 for EWAS -52939 cannot be mapped to UniProt!

2013-12-06 15:09:20,498 [main] WARN org.reactome.convert.common.PostProcessTemplate - SNORD59A for EWAS -53527 cannot be mapped to UniProt!

2013-12-06 15:09:22,622 [main] WARN org.reactome.convert.common.PostProcessTemplate - SNORD59B for EWAS -54534 cannot be mapped to UniProt!

2013-12-06 15:09:25,656 [main] INFO org.reactome.convert.common.PostProcessTemplate - Total unmapped names: 602

10). We may get a rough idea how many proteins can be merged into the original Reactome from the output in the last section like the following (ids for UniProt accession numbers):

2024-04-16 19:55:42,856 [main] INFO org.reactome.fi.FINetworkBuilder - Count proteins...

Total ids from Reactome: 12937

Total ids from KEGG: 0 # Note: KEGG is not used any more as of FI\_2024

Total ids after merging: 12937 (NaN, 0)

Total SwissProt ids after merging: 11193 (0.5477635313692865)

Total ids from Nature-PID: 2573

Total ids after merging: 13218 (0.10921103769918383, 281)

Total SwissProt ids after merging: 11460 (0.5608299892336303)

Total ids from Biocarta-PID: 1550

Total ids after merging: 14108 (0.5741935483870968, 890)

Total SwissProt ids after merging: 11534 (0.5644514045218753)

Total ids from TRED: 1136

Total ids after merging: 14198 (0.07922535211267606, 90)

Total SwissProt ids after merging: 11624 (0.5688558285210923)

Total ids from ENCODE: 9400

Total ids after merging: 17565 (0.35819148936170214, 3367)

Total SwissProt ids after merging: 14990 (0.7335812860918077)

Total ids from Panther: 1176

Total ids after merging: 17656 (0.07738095238095238, 91)

Total SwissProt ids after merging: 15081 (0.7380346481354605)

1. **dumpPathwayDBs()**

Dump the converted curator tool projects into the extended Reactome database created before, and generate FI files from these converted projects.

Notes: 1). It is prudent to dump the reactome\_plus\_i database first before running this method just in case there is anything wrong to save time since the database has been modified from the original one. Do the following at the results dir:

mysqldump -u{} -p reactome\_{release}\_plus\_i >reactome\_{release}\_plus\_i\_before\_dump.sql

2). You are encouraged to compare the numbers from different pathways to ones in the previous reactome\_plus\_i database to make sure there is no big surprise there. You can use the dataSource attribute to get a list of converted Event instances for a specific database (e.g. KEGG, Panther) with the following data sources: pantherdb (2019: 3066; 2024: 3297), Pathway Interaction Database (2019: 7616; 2024: 7616), TRED (2019: 3096; 2024: 3096), KEGG (2019: 9165; 2024: Not used any more), BioCarta – Imported by PID (2907), ENCODE (45328).

3). Some of UniProt identifiers that cannot be mapped to the original ReferenceGeneProducts from different databases may be duplicated in the database. Just leave them as they are. But have to remember to merge them when counting how many UniProt accessions in the database. Avoid checking the numbers based on ReferenceGeneProduct instances only.

1. **dumpPathwayFIs()**

Notes: 1). For the 2014 version of the FI network, hasCandidate values are not considered any more, which reduces the number of FIs extracted from Reactome about 11% from the 2013 version (aka from 144733 to 127382).

2). The current implementation will not extract TF/Target interactions annotated in Reactome using BlackboxEvent (e.g. DB\_ID = 452894) because the extraction checks only ReferenceGeneProducts. For the time being, this should be fine since Reactome doesn’t have many TF/Target interactions. We should get enough from TRED and ENCODE. Most likely these interactions will be extracted later on. Here are numbers for the 2014 FI network: if TF/target interactions are considered: 128,020; 127,382 if not considered (0.5%).

3). The 2017 version of the FI network got less FIs from complexes than 2016 from Reactome directly. This is a little bit surprising though more FIs were extracted from reactions.

The dumped pathway FIs should have similar numbers or more as follows (FI\_2024):

Total interactions from reactions: 232188

Time for looping: 1862

Total interactions from Reactome: 251784

After filtering: 155500

Before sequence consolidating: 155500

After sequence consolidating: 154898

Done data source: Reactome

Total interactions from reactions: 12465

Time for looping: 140

Total interactions from Reactome: 13878

After filtering: 13878

Before sequence consolidating: 13878

After sequence consolidating: 13872

Done data source: pantherdb

Total interactions from reactions: 15101

Time for looping: 172

Total interactions from Reactome: 15907

Total interactions from Interaction Event: 0

After filtering: 15826

Before sequence consolidating: 15804

After sequence consolidating: 15804

Done data source: Pathway Interaction Database

Total interactions from reactions: 5639

Time for looping: 105

Total interactions from Reactome: 6709

Total interactions from Interaction Event: 0

After filtering: 6709

Before sequence consolidating: 6709

After sequence consolidating: 6709

Done data source: BioCarta - Imported by PID

Total interactions from reactions: 0

Time for looping: 0

Total interactions from Reactome: 0

Total interactions from Interaction Event: 2849

After filtering: 2849

Before sequence consolidating: 2849

After sequence consolidating: 2849

Done data source: TRED

Total interactions from reactions: 0

Time for looping: 0

Total interactions from Reactome: 0

Total interactions from Interaction Event: 8354

After filtering: 8354

Before sequence consolidating: 8354

After sequence consolidating: 8351

Done data source: ENCODE

4). You may run org.reactome.fi.RFPredictionResultAnalyser.checkAnnotaedFIs() to the the total number of annotated FIs, proteins and genes:

2024-04-19 16:47:33,528 [main] INFO org.reactome.fi.RFPredictionResultAnalyzer - Total SwissProt ids: 20434

2024-04-19 16:47:33,530 [main] INFO org.reactome.fi.RFPredictionResultAnalyzer - Annotated FIs:

FI: 195006

Ids: 9616

2024-04-19 16:47:35,724 [main] INFO org.reactome.fi.RFPredictionResultAnalyzer - Size of uni2gene: 20490

2024-04-19 16:47:35,846 [main] INFO org.reactome.fi.RFPredictionResultAnalyzer - Annotated FIs in Genes:

FI: 188065

Ids: 9613

1. **Prepare feature files for random forest training, test, and prediction**

These steps are handled by project idg-fi-network-ml (<https://github.com/reactome-idg/fi-network-ml>). The runnable jar file has been generated and deployed at the OHSU linux machine (RWS00061). Because of the large memory requirement and feature files used, this step should be run at that machine.

1. The jar file is: FeatureFileGenerator.jar. Before running anything, make sure the following two configuration files have been configured correctly: application.properties, and configuration.prop. The configuration.prop should be the same as the configuration.prop used in the fi\_network\_build\_rf project. The jar file is created by running the following command at terminal:

mvn clean package -P build-runnable-jar -DskipTests

mv target/fi-network-ml-jar-with-dependencies.jar target/FeatureFileGenerator.jar

1. Before running the following files, make sure all data files have been copied by checking the settings in application.properties.
2. Generate the three feature files in one call since the majority of time will be spent on loading pairwise features:

java -Xmx48G -jar FeatureFileGenerator.jar generate\_three\_files results/train\_feature\_file\_04152024.csv results/test\_feature\_file\_04152024.csv results/prediction\_feature\_file\_04152024.csv >generate\_features\_04152024.log 2>&1 &

Note: At present, the above process is run at the OHSU Linux box (/ssd/d0/ml/fi\_build/2024 for FI\_2024). The total running time should be around 7 or 8 hours.

1. The logging file is kept separated and copied over to the same results folder: generate\_features\_04152024.log.
2. **Train random forest**
3. Python is used for training random forest: rf\_train\_predict.py in scripts/ml of the idg\_fi\_network\_ml project
4. The current env used to run this Python script is flair at MacPro (Python version = 3.8.12). The folder is /Users/wug/programs/fi\_build. The results are in 2024 (FI\_2024). The command used to run is:

python rf\_train\_predict {training\_file} {test\_file} {prediction\_file} {working\_dir} {postfix\_name} {down\_sample (optional for test)}

To check if the code runs fine using down-sampling for FI\_2024

(flair) RWS07890:fi\_build wug$ python rf\_train\_predict.py 2024/train\_feature\_file\_04152024.csv 2024/test\_feature\_file\_04

152024.csv 2024/prediction\_feature\_file\_04152024.csv 2024 04202024 0.10 0.10 >2024/rf\_train\_predict\_04202024.log 2>&1 &

To run actually without down-sampling for FI\_2024

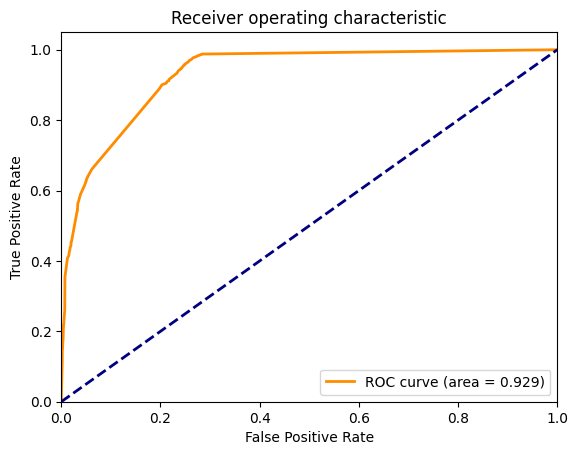
(flair) RWS07890:fi\_build wug$ python rf\_train\_predict.py 2024/train\_feature\_file\_04152024.csv 2024/test\_feature\_file\_04152024.csv 2024/prediction\_feature\_file\_04152024.csv 2024 04202024 none 0.10 >2024/rf\_train\_predict\_04202024.log 2>&1 &

1. Since we cannot get the performance curve from MacPro since it runs at terminal, we need to run the following Python script locally to get the plot:

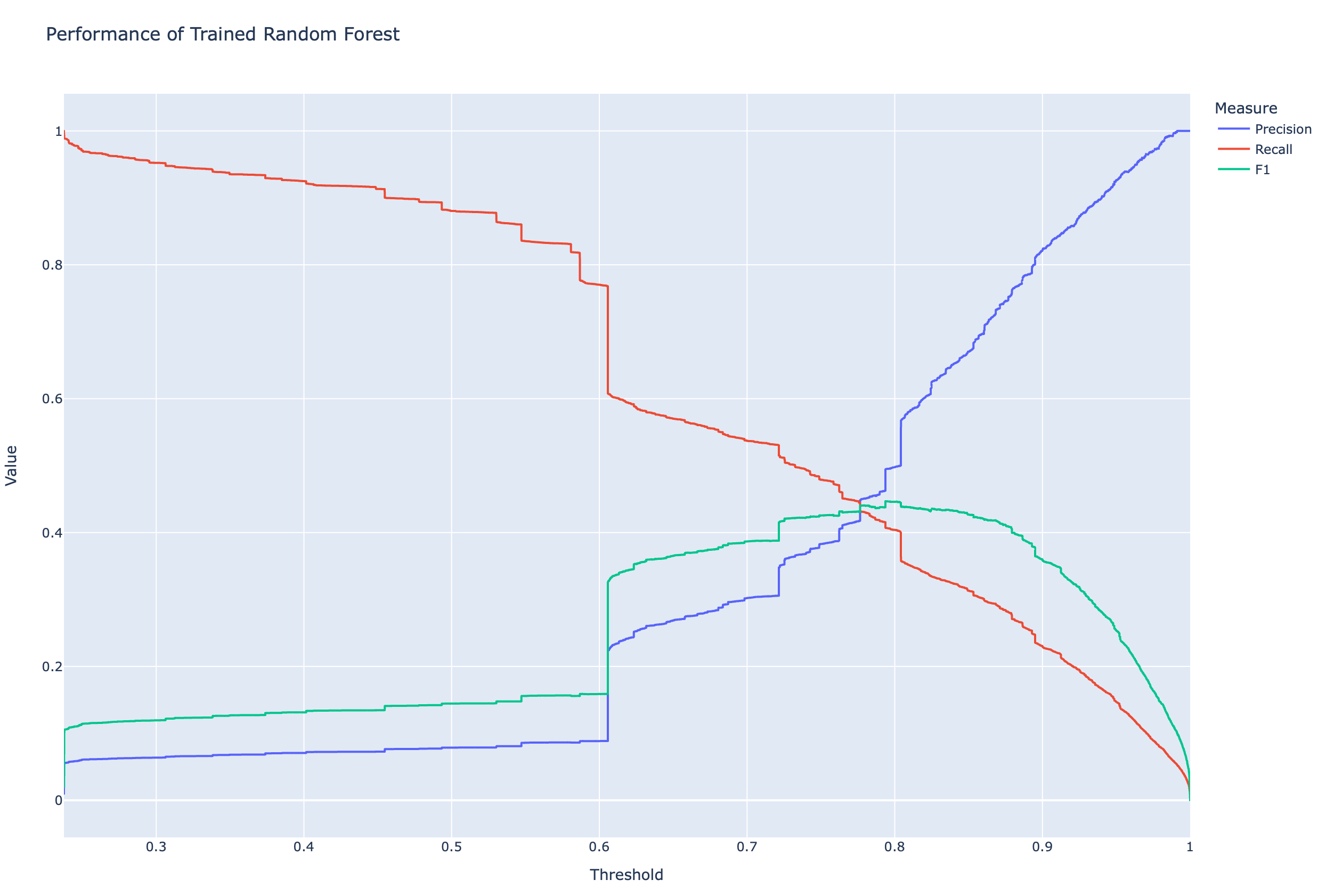
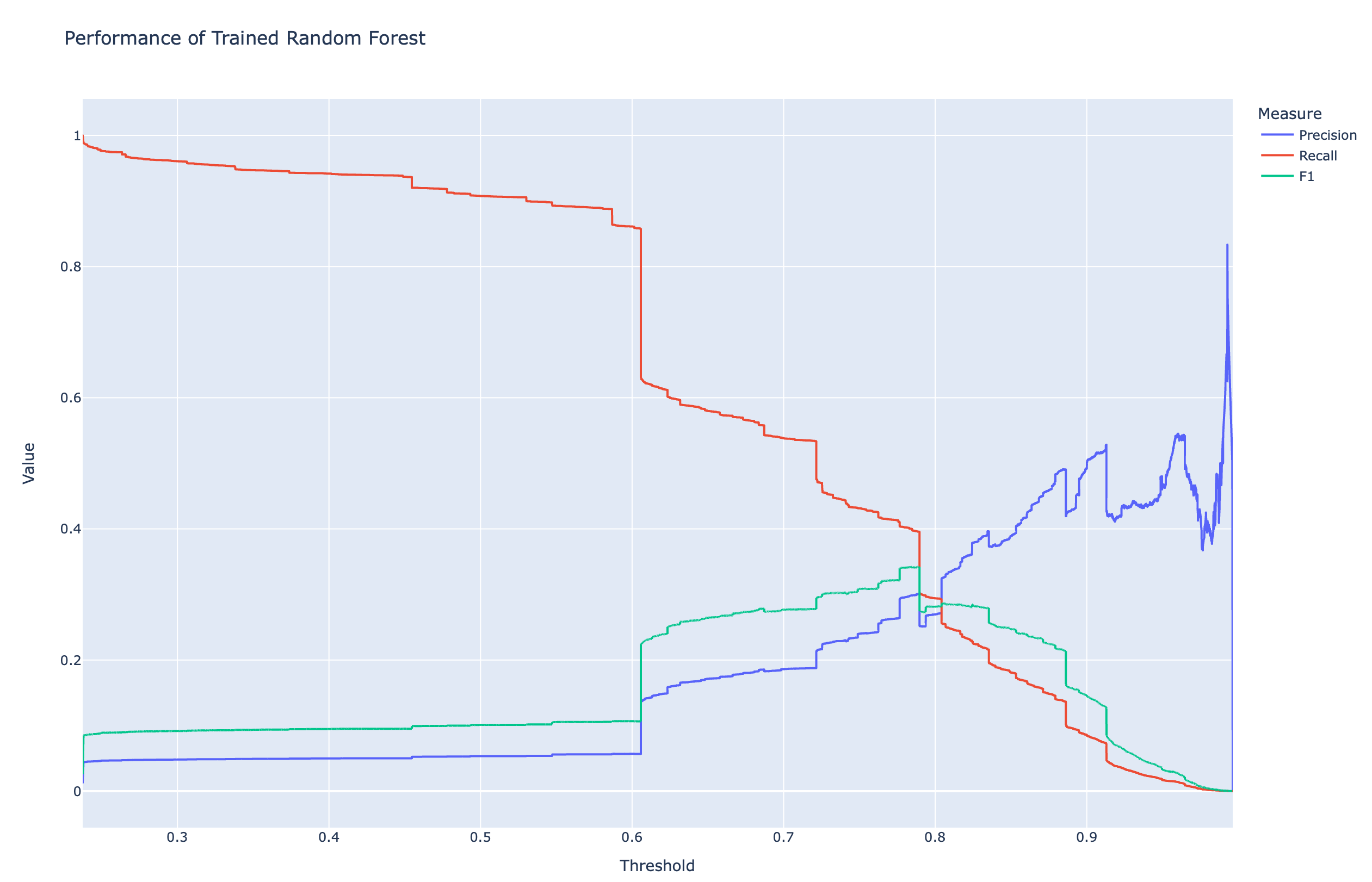
(scanpy\_3\_10) RNB13382:ml wug$ python F1Analyzer.py /Volumes/ssd/results/reactome-fi/fi\_network\_build\_rf/2024/train\_test\_precision\_recall\_04202024.csv

Note: Make sure the working directory and the input file are correct and use the correct python ENV.

1. The results from the above script should be saved in the same folder as the input files. Two files should be exported, one for FI plot and another for ROC. Screenshot copied below (results for FI\_2024):



1. Starting from FI\_2024, we take 10% of Reactome FIs aside as another test datasets. The above check is repeated for this subset of Reactome FIs. The final trained RF used in the following steps is based on 90% of Reactome FIs. By doing this, we have a better understanding about the performance of the training RF since FIs from other pathway databases without KEGG may have quite lower quality. Here is an example of the performance plot with 10% set aside Reactome FIs (the right is the non Reactome FI results for comparison). Both of plots are based on 90% of Reactome FIs.



1. **Choose predicted FIs based on score from the trained random forest**
   1. To determine the threshold that should be used to choose predicted FIs, run checkThreshold() in class org.reactome.fi.RFPredictedResultAnalyzer:

Partial outputs at terminal:

2024-04-19 16:54:20,086 [main] INFO org.reactome.fi.RFPredictionResultAnalyzer - Total predicted pairs: 22972504

2024-04-19 16:54:23,133 [main] INFO org.reactome.fi.RFPredictionResultAnalyzer - Size of uni2gene: 20490

2024-04-19 16:54:23,315 [main] INFO org.reactome.fi.RFPredictionResultAnalyzer - Total predicted pairs after removing annotated FIs: 22838640

SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".

SLF4J: Defaulting to no-operation (NOP) logger implementation

SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.

2024-04-19 16:54:23,522 [main] INFO org.reactome.fi.RFPredictionResultAnalyzer - Threshold 0.0...

2024-04-19 16:54:27,928 [main] INFO org.reactome.fi.RFPredictionResultAnalyzer - Threshold 0.01...

2024-04-19 16:54:31,217 [main] INFO org.reactome.fi.RFPredictionResultAnalyzer - Threshold 0.02...

2024-04-19 16:54:34,445 [main] INFO org.reactome.fi.RFPredictionResultAnalyzer - Threshold 0.03...

2024-04-19 16:54:37,618 [main] INFO org.reactome.fi.RFPredictionResultAnalyzer - Threshold 0.04...

2024-04-19 16:54:40,916 [main] INFO org.reactome.fi.RFPredictionResultAnalyzer - Threshold 0.05...

Two files are generated: NumberOfPredictedFIsVsThreshold\_02042022.csv, and NumberOfPredictedFIsVsThreshold\_02042022.html (The html file may have to saved manually).

Note: Try to pick up a threshold value giving us enough predicted FIs, usually less than 50% of pathway FIs and having good enough precision ~0.50. However, these numbers are determined heuristics. A better way is needed in the future. Also we may need to use 10% or more of Reactome FIs for testing purpose after removing the KEGG FIs.

* 1. Before running the method, set the cutoff value in the configuration file (See the notes in each version to see the reason. This may change every year) and assign at least 12G for the method, checkFIFiles() in org.reactome.fi.generateFIs():

# FI\_2024

CUT\_OFF\_VALUE=0.89d

# Related to prediction results from the trained RF

RF\_PREDICTION\_FILE = ${RESULT\_DIR}/prediction\_results\_04202024.csv

# The original feature file for prediction that should be used to create the FI network database

RF\_FEATURE\_FILE=${RESULT\_DIR}/prediction\_feature\_file\_04202024.csv

PREDICTED\_FI\_FILE=${RESULT\_DIR}/PredictedFIs\_${DATE}.txt

Note: Somehow there is an old gene symbol, ANP32C, is used in some FIs. Have to manually update ReferenceGeneProduct in the database (DB\_ID 49306) by adding ANP32C as a gene name so that synonym mapping can work without throwing null.

Output may be like the following:

2024-04-19 20:49:25,310 [main] INFO org.reactome.fi.RFPredictionResultAnalyzer - Total predicted pairs: 22972504

2024-04-19 20:49:28,235 [main] INFO org.reactome.fi.RFPredictionResultAnalyzer - Size of uni2gene: 20490

2024-04-19 20:49:28,408 [main] INFO org.reactome.fi.RFPredictionResultAnalyzer - Total predicted pairs after removing annotated FIs: 22838640

2024-04-19 20:49:29,782 [main] INFO org.reactome.fi.RFPredictionResultAnalyzer - Predicted FIs in Genes for 0.885:

FI: 74397

Ids: 10808

2024-04-19 20:49:30,806 [main] INFO org.reactome.fi.RFPredictionResultAnalyzer - Size of synonym2gene: 44215

2024-04-19 20:49:30,862 [main] INFO org.reactome.fi.RFPredictionResultAnalyzer - After name normalization:

FI: 74374

Ids: 10788

2024-04-19 20:49:31,121 [main] INFO org.reactome.fi.RFPredictionResultAnalyzer - Total SwissProt ids: 20434

2024-04-19 20:49:31,122 [main] INFO org.reactome.fi.RFPredictionResultAnalyzer - Annotated FIs in UniProt:

FI: 195006

Ids: 9616

2024-04-19 20:49:32,793 [main] INFO org.reactome.fi.RFPredictionResultAnalyzer - Size of uni2gene: 20490

2024-04-19 20:49:32,939 [main] INFO org.reactome.fi.RFPredictionResultAnalyzer - Annotated FIs in Genes:

FI: 188065

Ids: 9613

2024-04-19 20:49:33,930 [main] INFO org.reactome.fi.RFPredictionResultAnalyzer - Size of synonym2gene: 44215

2024-04-19 20:49:34,006 [main] INFO org.reactome.fi.RFPredictionResultAnalyzer - After name normalization:

FI: 188065

Ids: 9613

2024-04-19 20:49:34,015 [main] INFO org.reactome.fi.RFPredictionResultAnalyzer - Remove annotateFIs in normalized predicted FIs:

FI: 73962

Ids: 10722

2024-04-19 20:49:34,059 [main] INFO org.reactome.fi.RFPredictionResultAnalyzer - Total FIs in Genes:

FI: 262027

Ids: 13418

2024-04-19 20:49:34,059 [main] INFO org.reactome.fi.RFPredictionResultAnalyzer - Done.

1. **buildFIDb()**
   1. Before running the method, create an empty mysql database named as “FI\_yyyy” (yyyy should the year the FI network is building).
   2. Before running the method, make sure the following values in the hibernate configuration file, resources/funcIntHibernate.cfg.xml , are correct:

<property name=*"connection.url"*>jdbc:mysql://localhost:3306/FI\_2024</property>

<property name=*"connection.username"*>root</property>

<property name=*"connection.password"*>macmysql01</property>

* 1. Make sure –Xmx12G is set before running the method.

Output from the method running (logging information from hibernate has been removed):

2012-04-17 16:47:20,257 [main] INFO org.reactome.fi.FINetworkBuilder - Running FIDBBuilder.generateSchema()...

……

2012-04-17 16:47:22,447 [main] INFO org.hibernate.tool.hbm2ddl.SchemaExport - schema export complete

Interaction should be empty: 0

2012-04-17 16:47:22,548 [main] INFO org.reactome.fi.FINetworkBuilder - Running FIDBBuilder.dump()...

Total empty interactions: 0

Total empty interactions: 0

……

Total empty interactions: 505

……

Total empty interactions: 199

total interactions: 193901

Total time to extract: 130507

……

Total time for saving: 144991

2012-04-17 16:51:58,046 [main] INFO org.reactome.fi.FINetworkBuilder - Running FIDBBuilder.dumpPredicted()...

Total predicted FIs: 67892

Total protein from db: 9609

Total FIs will be added to the FI database: 67892

Save proteins: 5275

Save evidences: 67892

1. **generateCytoscapePlugInFiles()**
   1. Before running the method, the following properties should be set up in the configuration file:

# The minimum size used for a pathway

MINIMUM\_PAHTWAY\_SIZE = 3; // Change to 1 now for Pathway Enrichment Analysis to take care of the filtering.

GENE\_FI\_BIG\_COMP\_FILE\_NAME=${RESULT\_DIR}/FIsInGene\_121013\_BigComp.txt

GENE\_FI\_FILE\_NAME=${RESULT\_DIR}/FIsInGene\_121013.txt

GENE\_FI\_PATHWAY\_FILE\_NAME=${RESULT\_DIR}/FIsInGene\_Pathway\_121013.txt

GENE\_FI\_PREDICTED\_FILE\_NAME=${RESULT\_DIR}/FIsInGene\_Predicted\_121013.txt

# File for mapping accession to names dumped from the FI network

PROTEIN\_ACCESSION\_TO\_NAME\_FILE = ${RESULT\_DIR}/ProteinAccessionToName\_121013.txt

# We need a flattened list of pathways from Reactome for enrichment analysis

REACTOME\_PATHWAYS = ${RESULT\_DIR}/ReactomePathways121013.txt

# As of Decemember, 2013, we also want to dump all Reactome pathways in order

# to do a hierarhy based pathway enrichment analysis

PROTEIN\_ID\_TO\_REACTOME\_PATHWAYS = ${RESULT\_DIR}/ProteinIdToReactomePathways121013.txt

GENE\_TO\_REACTOME\_PATHWAYS = ${RESULT\_DIR}/ProteinNameToReactomePathways121013.txt

# Gene sets based on pathways

PROTEIN\_ID\_TO\_TOPIC = ${RESULT\_DIR}/ProteinIdToTopic121013.txt

GENE\_TO\_TOPIC = ${RESULT\_DIR}/ProteinNameToTopics121013.txt

REACTOME\_GMT\_FILE\_NAME = ${RESULT\_DIR}/ReactomePathways\_020720.gmt

Output from the method running should be similar to the following:

2012-04-17 17:33:41,044 [main] INFO org.reactome.fi.FINetworkBuilder - Running HibernateFIReader.generateFIFileInGeneInHiberante()...

……

Total interactions from prediction: 67892

Total interactions from pathways: 193901

Time for getting interactions: 5318

Total predicted FIs: 38450

Total pathway FIs:135432

Total FIs: 172235

Total predicted proteins: 8102

Total pathway proteins: 7402

Total proteins: 10696

2012-04-17 17:33:47,972 [main] INFO org.reactome.fi.FINetworkBuilder - Running FIGraphAnalyzer.analyzeComponents()...

Total interactions: 172235

Total components: 81

0: 10501

……

2012-04-17 17:33:49,860 [main] INFO org.reactome.fi.FINetworkBuilder - Running HiberanteFIReader.generateAccessionToProteinNameMap()...

2012-04-17 17:33:50,353 [main] INFO org.reactome.fi.FINetworkBuilder - Running ReactomeAnalyzer.generateListOfPathways()...

Total Pathways: 141

2012-04-17 17:34:12,681 [main] INFO org.reactome.fi.FINetworkBuilder - Running PathwayGeneSetGenerator.generateProteinNameToPathwayMap()...

……

* 1. This step has been disabled in 2020 since HotNet is not supported any more: The following two method calls at this step are related to create a matrix and heat kernel using R at the OICR cluster. The second method should not be run at the first time:

HotNetMatrixCalculator hotnetMatrixCalculator = **new** HotNetMatrixCalculator();

hotnetMatrixCalculator.testCalculateHeatKernel();

// The following method should be called after the kernel file was generated from using R

// hotnetMatrixCalculator.generateSerializedMatrixFile();

The generated matrix file, HotNet\_L\_matrix\_{year}.txt, should be zipped and moved to the cluster and processed by the script, runHeatKernel\_R.sh, in directory, ~/home/caBigR3. The script should be modified based on the actual file names, and submit using this: qsub –l h\_vmem=16G runHeatKernel\_R.sh. The results should be copied back and then ran the above method (second method).

* 1. The web application for ReactomeFIViz needs the annotated FI file, FIsInGene\_xxxx\_with\_annotations.txt too. This file is generated by using the method org.reactome.r3.fi.InteractionAnnotator.annotateAllFIs() in the FIVizWS\_corews project. There are some interactions may not be annotated, which may be mixed with known types and should be manually fixed.
     1. Make sure the InteractionTypeMapper.xml file is updated to the file having the same name in the FINetworkBuild project. This file should be copied into the webapp/WEB-INF folder.
     2. About 10 extracted FIs have no types because of gene name changes during sequence-based normalization (e.g. CALM1 nd CALM3 have the same sequence, causing ENCODE TargetedInteractions for CALM1 swapped). You may copy annotations from the previous year’s version or manually annotate them based on the original types in the database.
     3. This is a long process, taking about 2.5 hours.

1. **CompareFilesToPreviousVersion**:
   1. This method was added later on to check how similar the new version of the FINetwork, along with files generated for the web application, to previous one
   2. Output from this method run is something like this (partial results):

2018 2017

BP\_Domain\_Shared\_Pairs.txt 5600374 4252882

CGISurvivalAnalysis.R 5731 5731

Combined\_Logging.txt 2463140 39094755

FI\_2017.sql NA 37300517

FI\_2017\_Numbers.pptx NA 54591

FI\_2018.sql 54204498 NA

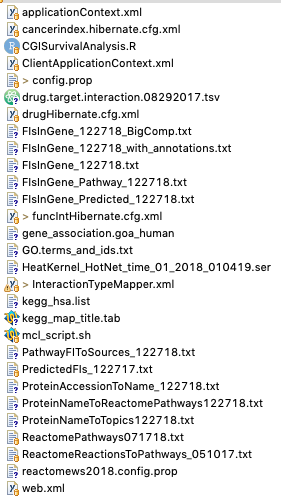
FI\_2018\_Numbers.pptx 54200 NA

FIsInGene\_071718.txt NA 2972316

……

1. **Set up Cytoscape Web Application and Plug-in**

Note: The ant script may need to be tested more and updated. The current way to generate war file is to use Eclipse: All needed files should be manually copied from the result folder to FIVizWS\_corews/WebContent/WEB-INF. All configurations should be edited manually. Here is a list of files needed by this web application:



The web application for the Reactome FI plug-in app is managed by another independent Java Web project, caBigR3WebApp. In this project, there is an ant build file, buil.xml, run this build.xml file to create a new war file and deploy this war file to the tomcat webapps folder in the deployment machine.

Notes: 1). These two values in the build.xml should be modified before running ant, version and buildDate.

2). There is a bug in the current version of the build code. The generated mysqldump file for the \_*plus*\_i database has corrupted the Ontology table. You may have to fix this table by re-install a correct version of Ontology table from gk\_central.

3). In order to view instances for newly added classes (Interaction and TargettedInteraction), the Ontology table needs to be modified by using a Perl script called updateDatabase.pl in the script folder. However, in order to use this script, a new Protégé project should be created by downloading a project file from gk\_central in the web site, adding a new attribute dataSource, and two new classes (Interaction and TargettedInteraction). Please follow the database model in the curator tool to make changes. (This is not needed any more since a simple view is used for instance).

4). For this web application and providing the download file, we need to add annotations to the FIsInGene\_XXX.txt file. To generate this file, call this method, annotateAllFIs() in class org.reactome.r3.fi.InteractionAnnotator, which is in the project caBigR3WebApp. Because these two genes, TRAPPC2 and TRAPPC2P1, have the same sequence, the following FI cannot be annotated, which is imported from ENCODE:

TRAPPC2P1 ZBTB33 unknown null 1.00

Based on the original data source, the above should be changed into the following manually in the file:

TRAPPC2P1 ZBTB33 expression regulated by <- 1.00

5). GSEA-based pathway enrichment analysis is provided by another WS service. Don’t forgot to copy the newly generated Reactome human gmt file (e.g. ReactomePathways\_122718.gmt) to that WS servlet and then restart it.

6). PathwyDiagramsFactorGraphs.xml.zip should be copied into the Cytoscape folder. All other files should be in the WEB-INF folder except two GMT files.

7). Don’t forget to enable file-based logging for all servlets when deploying the applications.

*======The END======*