DComboNet-vignette

Fangyoumin Feng

Abstract

Description of your vignette

```r
knitr::opts_chunk$set(
  echo = TRUE,
  collapse = TRUE,
  comment = "#>",
  eval = FALSE
)
library(DComboNet)
```

#Introduction

`DComboNet` is an R package for personalized anti-cancer drug combination prediction based on multi-level data integration. There are two main prediction models contained in the package. The level one model is for generalized anti-cancer drug combination effectiveness prediction and level two model is for cancer sample specific drug combination prediction. The two-level model based on a network-based method which integrates five subnetwork including drug-drug, drug-gene, drug-pathway, gene-gene and pathway-pathway association networks. Random walk with restart(RWR) algorithm is used to capture global proximity between drugs and the result is based on the rank returned via RWR algorithm. `DComboNet` also provide clues for the potential mechanisms of drug combinations by extracting the top ranked genes/drugs between predicted drug combinations.

This tutorial provides the instruction of the main usage of this package fitting different scenario. This tutorial will lead to know the basic usage of the prediction, the description of prepared data how to extend the network construction to fit your own dataset. This tutorial will not present detail description of all functions contain in the packagem but you can easily learn those in help document with the R package.

#Package installation

DComboNet has been upload in Github and can be install as follow:

```r
  install.packages("devtools")
  devtools::install_github(".../DComboNet")
  library(DComboNet)
```

Due to the large size of drug-induced gene expression profiles for Level 2 model, we prepared an extra data folder. After install the package, you should download this compressed data set, unzip it and put in a reachable path. This path should be used as `load_dir` for functions in DComboNet.

#Overview of DComboNet()

The prediction of drug combinable tendency is packed in function `DComboNet`. It provides two level prediction models and includes network construction, transition matrix generating and global similarity calculation:

```r
DComboNet(load_dir,
          resultdir,
          model = c('L1','L2'),
          drugcandidate = NULL,
          manual_input = c('TRUE','T','FALSE','F'),
          CDK_FP = NULL,
          pubchemFP = NULL,
          MACCS_FP = NULL,
          drugnetWeight = c('TRUE','T','FALSE','F'),
          featuretype = c('ATCsim','STsim','SEsim','integrated_score'),
          drugtarget = NULL,
          druggene = NULL,
          dataset = NULL,
          cellline = NULL,
          treatment_time = NULL,
          foldchange_DEG = NULL,
          pvalue_DEG = NULL,
          foldchange_DEP = NULL,
          pvalue_DEP = NULL,
          drugDEG = NULL,
          drugDEP = NULL,
          cancergene = NULL,
          dtweight = 2,
          dgweight = 1,
          dDEGweight = 1,
          dgpweight = 1,
          dDEpweight = 1,
          gpweight=1,
          x = 0.5,
          y = 0.5,
          z = 0,
          A = 0.5,
          B = 0.5,
          eta = 1,
          r = 0.7)
```

• The function provide internal datasets for network construction. User should provide the path (`load_dir`) to access the data folder downloaded with the package.

• Path to save result files should be specified in `result_dir`.

• Two level models are provided for combination prediction, `L1` and `L2` denote **level one** and **level two** model, respectively. Different models require differnent following inputs. The details about differnet usages of the two-level models will describes in next session.

• Two solutions are provided here for users to input their interested drugs. If `manual_input = TRUE`, after seeing the tip **"Please type in the drug you are interested:"**, user can manually input drug name; if `manual_input = FALSE`, `drugcandidate` should be feed with `data.frame` contain drug name(s) and their drugbank ID. Note that if the drug you are interested in are not contained in the provided drug list, `drugcandidate` together with their chemical fingerprints generated via PADEL (CDK fingerprint `CDK_FP`, pubchem fingerprint `pubchemFP` and MACCS fingerprint `MACCS_FP`) should be provided for adding newly inputed drug(s) in the drug-drug association network.

• Next two parameters `drugnetWeight` and `featuretype` are necessary. User needs to inform whether it is necessary to assign values to the edge of drug-drug association network and which feature should be used to assign. Three drug-drug similarities(`ATCsim`, `STsim`, `SEsim`) together with integrated pharmacological score (`integrated_score`) are provided as options.

• To update drug-gene association network, gene-gene association network and drug-pathway assocition netwrok, if new drug(s) was taken as input in `drugcandidate`, User must provide drug(s) and their target genes table in `drugtarget` for both **level one** and **level two** models. For **level one** model, `druggene` can be provided to extend the connections between drug and gene or pathways. Here, `druggene` table can be extracted from multiple resources, such as IPA tools and publications. For **level two** model, cancer sample specific expressed gene list and drug induced transcriptome change should be provided. `DComboNet()` gives two options: if the cell line and drug(s) are in our provided datasets (which is generated from LINCS database and CCLE database), user can put in `cellline`, `dataset` and `treatment_time` to create network. Differentially expressed genes were selected by functions `lmFIt()` and `eBayes()` in the `Limma` package. Differential regulated pathways (DEpathway) were obtained by the `GSVA` algorithm. Note that gene expression filtering criteria (`foldchange` and `pvalue`) provides a detailed customizing solution. If cell line or drugs are not in provided dataset, you can simply provided your own cancer sample/cell line specific expressed gene list and drug-induced differentially expressed gene list in `cancergene` and `drugDEG`.

• Other then weight method for drug-drug assocaition network described above, for drug-gene association, drug-pathway association and gene-pathway association, different edge weight strategies are made as parameters. For **level one** model, the default weights are:

· `dtweight = 2`: the edge weight of drug-target gene associations sets as 2.

· `dgweight = 1`: the edge weight of drug-related gene associations sets as 1.

· `dgpweight = 1`: the edge weight of drug-pathway associations sets as 1.

For **level two** model, drug-targeted gene association will be weighted with the value set to `dtweight`, and drug-DEG association will be weighted according to the foldchange between drug-induced gene expression profile and control; the associations between drugs and their differentially regulated pathways will be weighted by parameter `dDEpWeight`.

• Parameters `x`, `y`, `z`, `A` and `B` denote the jumping probability in different subnetworks) and these jumping probabilities are not independent ($x = 1 - A$, $y = 1 - A$, $B = 1 - x$):

· `x` denotes the jumping probability in drug-gene association network (inter-network crossing event happens between drug and gene nodes).

· `y` denotes the jumping probability in drug-pathway association network (inter-network crossing event happens between gene and pathway nodes).

· `z` denotes the jumping probability in gene-pathway association network.

· `A` denotes the jumping probability in drug-drug association network (when the inter-network crossing events between drug and gene nodes and between drug and pathway nodes are neither existing).

· `B` denotes the jumping probability in gene-gene assocaition network (when the inter-network crossing events between drug and gene nodes and between gene and pathway nodes are neither existing).

• Numeric parameter `eta` is to controls the probability of restarting in the corresponding network. in `DComboNet()`, it always starts from drug seed in drug-drug association network, therefore `eta` sets to be 1.

• Numeric parameter `r` denotes global restart probability. It can be set as any value from 0 to 1, but according to previous researches and our tuning result, 0.7 shows the best prediction power.

#Quick start

To quickly get the usage of `DComboNet`, prediction assignment for one drug is taken as an example to go through the basic functions. Here, `Sorafenib` is taken as example, note that `Sorafenib` is within our pre-prepared data set, if you want to know how to do prediction for new drugs, please check session 5 for more details.

Before start any prediction assignment, please make sure `data` folder is fully downloaded and put in a fetchable path. `data` folder contains data files for construct networks as well as testing datasets used in our publication.

```r
# 1. Make sure the path of `data` folder assign to variable correctly
## Here is an example path, you should switch to your own
load_dir = "G:/lab/DCcomboNet/Rpackage/input_data/"
# 2. Specify the path to save result files
## Here is an example path, you should switch to your own
resultdir = "G:/lab/DCcomboNet/Rpackage/tryout_result/"
```

As we introduced in session 3, `DComboNet()` function incorporate two level models, you can choose level one model which is a generalized anti-cancer drug combination model and level two model which is a cancer-sample specific drug combination prediction model.

##Level one model

To start generalized prediction assignment, you should choose `L1` in parameter `model`. Then drug that you are interested in predict combinable drugs should be input. Here, we provide two possible solutions to input the drug you are interested in `DComboNet`:

1) Input one or more drug name(s) together with their drugbank ID in a `data.frame`. In this case, you should shield manual input by `manual_input = FALSE`.

```r
# runing level one model (L1)
drugcandidate = data.frame(drug = "Sorafenib", drugbankID = "DB00398")

DComboNet(load_dir = load_dir,
          resultdir = resultdir,
          model = "L1", # To choose level one model
          manual_input = FALSE, # To shield manually input drug name
          drugcandidate = drugcandidate,
          drugnetWeight = TRUE, # Confirm if drug network should be weighted
          featuretype = 'integrated_score') # Select which drug-drug similarity should be use
```

Note that if neither has `drugcandidate` been inputed nor has `manual_input` been allowed, all drug nodes in constructed network will be traversed for prediction.

2) Submit the name of drug in an interactive manner. Drug for prediction can be entered interactively after the question 'Please type in the drug you are interested:' pop out. Note that, if `manual_input = TRUE`, even with input `drugcandidate`, the final result will be generates for the manually input drug.

```r
DComboNet(load_dir = load_dir,
          resultdir = resultdir,
          model = "L1",
          manual_input = TRUE, # Select manually input function
          drugcandidate = NULL,
          drugnetWeight = TRUE,
          featuretype = 'integrated_score')

# Don't run here
# Please type in the drug you are interested:
# Sorafenib
```

Three results will be generated and saved under the folder `drugrank`, `generank` and `pathwayrank` in `resultdir` path. Prediction result for drug candidates is saved in `drugrank` folder. The name of result file indicate `drugseed` and inside the file, drug candidates within network is ranked based on global similarity.

```r
# Take Sorafenib as an example
model = "L1"
```

```r
drugrank = read.csv(paste0(resultdir,model,"_result/drugrank/Sorafenib_rank.csv"))
DT::datatable(drugrank, options = list(pageLength = 5))
```

Genes and pathways are also ranked according to global similarity. This two files will be used for subnetwork extraction and visualization.

```r
# Take Sorafenib as an example
generank = read.csv(paste0(resultdir,model,"_result/generank/Sorafenib_rank.csv"))
DT::datatable(generank, options = list(pageLength = 5))
pathwayrank = read.csv(paste0(resultdir,model,"_result/pathwayrank/Sorafenib_rank.csv"))
DT::datatable(pathwayrank, options = list(pageLength = 5))
```

##Level two model

To run cancer sample specific model requires more input parameters, especially to define cancer cell line and drug treatment related variables. Drug induced gene expression changes are from LINCS database where two datasets (`GSE70138` and `GSE92742`) are provided from GEO database. Multiple cancer cell lines together with different treatment time are also included in the two datasets. Depends on the specific purpose, you can choose cell line, treatment time and filtering criteria to generate your own gene set.

Still take `Sorafenib` as example, if aiming at predicting combinable drug for `Sorafenib` in Hepatocellular carcinoma cell line `HEPG2`, since corresponding dataset for `HEPG2` is provided in LINCS and CCLE database, you can choose `dataset`, `cellline` and `treatment_time` for calling the drug-induced differential expressed gene table and HEPG2 specific expressed gene table, then the model can be set as follow:

```r
# Prepare input drugseed
drugcandidate = data.frame(drug = "Sorafenib", drugbankID = "DB00398")

# The setting of other parameters can be the same as how we run level one model
DComboNet(load_dir = load_dir,
          resultdir = resultdir,
          model = "L2", # Choose level two model
          manual_input = FALSE, # You can use manual input function (see level one model example)
          drugcandidate = drugcandidate,
          drugnetWeight = TRUE,
          featuretype = 'integrated_score',
          dataset = "92742",
          cellline = "HEPG2",
          treatment_time = 6,
          # the absolute value of fold-change between drug-treated group and control group will be abov
          foldchange = 0.5,
          # the p-value from the significent test (t.test) between drug-treated group and control group
          pvalue = 0.05)
```

If you have other gene list related to the cancer cell line you are interested in and want to included in drug-gene assocaition network and/or gene-gene association network, you can input drugDEG and cancergene table (in data.frame format) manually.

Similar to level one model, three results will be generated and saved under the folder `drugrank`, `generank` and `pathwayrank` in `resultdir` path. Prediction result for drug candidates is saved in `drugrank` folder. The name of result file indicate `drugseed` and inside the file, drug candidates within network is ranked based on global similarity.

```r
# Take Sorafenib as an example
model = "L2"
drugrank = read.csv(paste0(resultdir,model,"_result/drugrank/Sorafenib_rank.csv"))
DT::datatable(drugrank, options = list(pageLength = 5))
```

Genes and pathways are also ranked according to global similarity. This two files will be used for subnetwork extraction and visualization.

```
# Take Sorafenib as an example
generank = read.csv(paste0(resultdir,model,"_result/generank/Sorafenib_rank.csv"))
DT::datatable(generank, options = list(pageLength = 5))
pathwayrank = read.csv(paste0(resultdir,model,"_result/pathwayrank/Sorafenib_rank.csv"))
DT::datatable(pathwayrank, options = list(pageLength = 5))
```

##Network visualization

Other than prediction function, `DComboNet` package provides functions to extract and visualize subnetwork between user-interested drug seed and its corresponding predicted combinable drug candidate, which may help infer the possible mechanism of drug combinations. This step only works when prediction assignment finished.

To run network visualization function, you need to check if the prediction result has been saved to given path `resultdir`. Note that the result files including drugrank table, generank table and pathwayrank table. Then capitalized drug names that you are interested can be inputed as `drugseed` and `drugcandidate`. `drugtarget` table should also be prepared as a dataframe and take as input. Genes and pathways for subnetwork construction are according to their rank corresponding to drugseed, the default value of gene rank (controlled via parametre `generank_filter`)is 0.01 meaning that only genes ranked on top 1% will be kept in subnetwork while default value of pathway rank(controlled via parametre `pathwayrank_filter`) is 0.1 meaning that only pathway ranked on top 10% will be kept in subnetwork.

After preparation of network visualization input, `network_visualization()` is used for visualizing network. This function was built based on `visNetwork` package. Drugseed, drugcandidate and their targeted genes are colorred coded. User can either click to select interesting nodes and drag around for better layout or select by id or group. An `.html` file will be generated and saved in under the result path. Furthermore, a `.graphml` file will also be saved. This file can be easily import as a network in Cytoscape and network style can be shown via choosing **column** type (e.g. **color**) and **Mapping type** set as "Passthrough Mapping".

```
library(visNetwork)

drugseed = "Sorafenib"
drugcandidate = "Vorinostat"
drugtarget = data.frame(drug = c(rep(drugseed,10),rep(drugcandidate,5)),
                        target = c("BRAF", "FGFR1", "FLT1", "FLT3", "FLT4", "KDR", "KIT", "PDGFRB", "RA

model = "L2"

network_extract(drugseed = drugseed,
                drugcandidate = drugcandidate,
                drugtarget = drugtarget,
                generank_filter = 0.01,
                pathwayrank_filter = 0.1,
                model = model,
                cellline = "HEPG2",
                load_dir = load_dir,
                resultdir = resultdir)

network_visualization(drugseed = drugseed,
                      drugcandidate =  drugcandidate,
                      drugtarget = drugtarget,
                      model = model,
                      cellline = "HEPG2",
                      load_dir = load_dir,
                              resultdir = resultdir)
```

Case study

Provided Data for network construction

Drug-drug associations

Drug-drug pharmacological score table was provided for drug-drug association network:

```
library(DT)

drugnet_feature = read.table(paste0(load_dir,"drug_net/features.csv"), sep=",",header=TRUE, stringsAsFa

# To fit the page, the values showing in the table below only kept two decimal places
# DT::datatable(drugnet_feature[1:20,], options = list(pageLength = 5))
# Only first 20 rows will be shown here as example.
```

The way to generate same feature table is packed in function `DComboNet`, user can also call function `DrugNetFeature`. To obtain features for newly inputed drug(s), extra informations need to provide, including drugs' name and their drugbank ID, as well as their chemical structure fingerprints:

```
# druglist including drug names and their drugbank ID, note that the drug name better use name provided
druglist = read.csv(paste0(load_dir, "/druglist_example.csv"), sep = ",", header = T, stringsAsFactors
# DT::datatable(druglist, options = list(pageLength = 5))
```

```
# For level one model
cellline = NULL
model = "L1"

# For level two model
# cellline = "HEPG2"
# model = "L2"
# feature will saved under the name of cell line

CDK_FP <- read.csv(paste0(load_dir,"data/fingerprints/fingerprints.csv"), sep = ",", header = T, string
pubchemFP <- read.csv(paste0(load_dir,"data/fingerprints/pubchem_fingerprints.csv"), sep = ",", header
MACCS_FP<- read.csv(paste0(load_dir,"data/fingerprints/MACCS_fingerprints.csv"), sep = ",", header = T,

# Otherwise, please input PaDEL generated three fingerprint files
drugnet_feature = DComboNet::DrugNetFeature(druglist = druglist,
                              cellline = NULL,
                              model = 'L1',
                              CDK_FP=CDK_FP,
                              pubchemFP=pubchemFP,
                              MACCS_FP=MACCS_FP,
                              load_dir = load_dir)
```

###Drug-gene associations

For level one model, drug-target gene associations and drug-related gene associations contribute to drug-gene association network:

```
drugtarget = read.table(paste0(load_dir,'data/drugtarget.csv'), sep =',', header = TRUE, stringsAsFacto
# DT::datatable(drugtarget[1:20,], options = list(pageLength = 5))
# Only first 20 rows will be shown here as example.
```

```
druggene = unique(read.csv(paste0(load_dir,'data/drug_gene/drug_ipa.inPPI.csv'), header = T, stringsAsFa
# DT::datatable(druggene[1:20, ], options = list(pageLength = 5))
# Only first 20 rows will be shown here as example.
```

For level two model, drug-related gene associations will be replaced by drug induced differentially expressed genes(DEG):

```
cellline = 'HEPG2'
dataset = '92742'
treatment_time = 6
drugDEG <- unique(read.csv(paste0(load_dir,'/LINCS_data/pathway_enrich_gsva_GSE',dataset,'/',cellline,'
# DT::datatable(drugDEG[1:20, ], options = list(pageLength = 5))
# Only first 20 rows will be shown here as example.
```

To generate drug-DEG association file, we provide function `DEG_DEP_preparation()` to obtain gene expression changes before and after by drug(s), then `drugDEG_preparation()` to filter with certain criteria.

```
druglist = read.csv(paste0(load_dir, "/druglist_example.csv"), sep = ",", header = T, stringsAsFactors =
cellline = 'HEPG2'
dataset = "92742" # which LINCS dataset to use
t = 6 # treatment time
# Before runing the function, please check how many available core can be use for calculation
# library(parallel)
# num_cores<-detectCores(logical=F)
num_cores = 2
# provide path to save results, please make sure it is saved in the same path with other modeling datas
load_dir = "G:/lab/DCcomboNet/Rpackage/input_data/"
DEG_DEP_preparation(druglist = druglist,
                    dataset = dataset,
                    cellline = cellline,
                    treatment_time = t,
                    core = num_cores,
                    load_dir = load_dir)
drugDEG_preparation(cellline = cellline,
                    dataset = dataset,
                    treatment_time = t,
                    foldchange = 0.5,
                    pvalue = 0.05,
                    load_dir = load_dir)
```

###Gene-gene associations

Gene-gene association network contains both cancer-related and drug-related genes. For level one model, cancers related genes were extracted from KEGG cancer related pathway including ‘pathway in cancer’:

```
cancer_gene = read.csv(paste0(load_dir,'data/network/protein-coding cancer genes name.csv'),header = F,
head(cancer_gene[,1],10)
```

For level two model, cancer sample/cell line specific expressed gene list extracted from CCLE database is provided:

```
cellline = "HEPG2"
load_dir = "G:/lab/DCcomboNet/Rpackage/input_data/"
cancer_gene = read.csv(paste0(load_dir,'data/cellline_genes/',cellline,'_genelist.csv', sep = ''),heade
head(cancer_gene[,1], 20)
```

If the gene list is not provided but the cancer cell line is included in CCLE database, function `cancerGene()`

can be used to generate cancer cell line specific expressed gene list:

```
# provide path to save results, please make sure it is saved in the same path with other modeling datas
cancer_gene = cancerGene(cellline = cellline,
                         load_dir = load_dir)
```

###Drug-pathway associations

For level one model, the associations between drugs and pathways are via a target-pathway mapping solution, that is, the drug-pathway association exists if the target gene(s) of a drug in drug-drug association network can be mapped to pathway(s). This process is packed in function `drugGeneNet.L1()` when construct drug-gene association network adjacency matrix. If new drugs are inputed, in this step, drug-target genes table should be provided.

For level two model, pathways(DEpathway) that differentially regulated by drugs was added. Drug-DEpathway associations are obtained by `gsva()` function. The process can be done via function `DEG_DEP_preparation()` and then use certain filter criteria to select pathways through `drugDEP_preparation()`:

```
druglist = read.csv(paste0(load_dir, "/druglist_example.csv"), sep = ",", header = T, stringsAsFactors =
cellline = 'HEPG2'
dataset = "92742" # which LINCS dataset to use
t = 6 # treatment time
# Before runing the function, please check how many available core can be use for calculation
# library(parallel)
# num_cores<-detectCores(logical=F)
num_cores = 2
# provide path to save results, please make sure it is saved in the same path with other modeling datas
load_dir = "G:/lab/DCcomboNet/Rpackage/input_data/"
DEG_DEP_preparation(druglist = druglist,
                    dataset = dataset,
                    cellline = cellline,
                    treatment_time = t,
                    core = num_cores,
                    load_dir = load_dir)
drugDEP_preparation(cellline = cellline,
                    dataset = dataset,
                    treatment_time = t,
                    foldchange = 0.5,
                    pvalue = 0.05,
                    load_dir = load_dir)
```

Some drug-DEpahtway tables are provided:

```
cellline = 'HEPG2'
dataset = '92742'
treatment_time = 6
drugDEP <- unique(read.csv(paste0(load_dir,'/LINCS_data/pathway_enrich_gsva_GSE',dataset,'/',cellline,'
# DT::datatable(drugDEP[1:20, ], options = list(pageLength = 5))
# Only first 20 rows will be shown here as example.
```

Predition with drug within provided dataset

After learning the pre-prepared network data files, let's try some more prediction task. In session 4, we tried some basic usage of prediction function `DComboNet` with drug that is within our provided dataset. We provided some more ways to personalized models for your dataset.

The first kind is a simple extension for drug within provided dataset. Let's continue taking "Sorafenib" as example. If there are new targeted genes or related gene got discovered (those genes that is not from Drugbank or IPA tools, or new interesting genes from publications), you can prepare your interested drug-targeted gene table (or drug-related gene table). For level one model, the prediction task can be as follow:

```
# runing level one model (L1)
drugcandidate = data.frame(drug = "Sorafenib", drugbankID = "DB00398")

# only as example
dt_table = data.frame(drug = 'Sorafenib', target  = 'RAF1')
# only as example
dg_table = data.frame(drug = 'Sorafenib', gene  = 'RAF1')
DComboNet(load_dir = load_dir,
          resultdir = resultdir,
          model = "L1", # To choose level one model
          manual_input = FALSE, # To shield manually input drug name
          drugcandidate = drugcandidate,
          drugnetWeight = TRUE, # Confirm if drug network should be weighted
          featuretype = 'integrated_score',
          drugtarget = dt_table,
          druggene = dg_table) # Select which drug-drug similarity should be use
```

Similarly, level two model can be run as:

```
# The setting of other parameters can be the same as how we run level one model
DComboNet(load_dir = load_dir,
          resultdir = resultdir,
          model = "L2", # Choose level two model
          manual_input = FALSE, # You can use manual input function (see level one model example)
          drugcandidate = drugcandidate,
          drugnetWeight = TRUE,
          featuretype = 'integrated_score',
          drugtarget = dt_table,
          druggene = dg_table,
          dataset = "92742",
          cellline = "HEPG2",
          treatment_time = 6,
          # the absolute value of fold-change between drug-treated group and control group will be abov
          foldchange = 0.5,
          # the p-value from the significent test (t.test) between drug-treated group and control group
          pvalue = 0.05)
```

For level two model, if you have other processed drug-DEG table and drug-DEP for cell line ‘HEPG2’ and/or HEPG2 specific expressed gene list, you can also provided them in the model:

```
# only as example

drugDEG = read.table(paste0(load_dir, '/test/Sorafenib_DEG_test.txt'), sep = '\t', header = T, stringsA
drugDEP = read.table(paste0(load_dir, '/test/Sorafenib_DEP_test.txt'), sep = '\t', header = T, stringsA
cancergene = read.table(paste0(load_dir, '/test/HEPG2_genelist.csv'), sep = ',', header = T, stringsAsF

# The setting of other parameters can be the same as how we run level one model
DComboNet(load_dir = load_dir,
          resultdir = resultdir,
          model = "L2", # Choose level two model
          manual_input = FALSE, # You can use manual input function (see level one model example)
          drugcandidate = drugcandidate,
          drugnetWeight = TRUE,
          featuretype = 'integrated_score',
          drugtarget = dt_table,
          druggene = dg_table,
          cellline = "HEPG2", # You must provide cell line name for saving results and generating netwo
          drugDEG = drugDEG,
          drugDEP = drugDEP,
          cancergene = cancergene)
```

Prediction with newly inputed drugs

`DComboNet` offers the possibility to extend network. You can provided your own drug list, drug-target data table, drug-related gene data table, drug-DEG/DEP tables etc to custmoized model to fit your purpose. In 5.2, you have learned how to provided extra drug-gene information for drug within provided dataset. Here, we are going to extend drug network. Take OCI-LY3 dataset as example.

Data preparation

In 5.1, you have seen how the provided data tables are like. For newly inputed drug(s) that you are interested, the basic rules is to prepare the same format of data tables to extend different subnetworks. The example tables for OCI-LY3 dataset is showing below to give a better hint of how to prepare your own data tables.

1) Drug table preparation

With newly added drug(s), `DComboNet` will automatically call function `DrugNetFeature` for generating integrated pharmacological score with all the provided drugs in network. For this, the drug table requires not only drug names (official names that matching ATC code system and drugbank database) but also their drugbank ID. For OCI-LY3 dataset, the drug table should prepare like this:

```
drugcandidate <-  read.csv(paste0(load_dir,"test/OCILY3_data/druglist_oci_ly3.csv"), sep = ',', header 
DT::datatable(drugcandidate, options = list(pageLength = 3))
```

2) Drug chemical structure fingerprints preparation

Three kinds of chemical structure fingerprints generated from software PaDEL-Descriptor. Before PaDEL-Descriptor, you should download 2D chemical structure in `.sdf` format (recommond from drugbank or Pubchem database), named after the name of drugs you provided before. Put the structure files together in one directory, PaDEL-Descriptor can generate fingerprints for all drugs at once. After loading the file, open `Advanced` and choose “Use filename as molecule name” so that rownames of fingerprint matrix in the output result are drug names. After setting these, you should choose “Fingerprints” and cancle “1D&2D” and “3D” in `Descriptors`, then open `Fingerprints` to select certain fingerprinter type. For CDK fingerprint, “Fingerprinters” should be choose, then go back to `General` page, make sure you edit the result name in a recognizable way (for example, as “fingerprint.csv”). Keep other as default, click `Start` button, results will be generated in few seconds or mins depends on the amount drugs you loaded. For pubchem fingerprint, similar setting as description before, but choose “PubchemFingerprinter” in `Fingerprints` page and name the result file as “PubChem.csv”; for MACCS fingerprints, you should choose “MACSSFingerprinter” in `Fingerprints` page and name as a corresponding name (for example “MACCF.csv”). To name the result files carefully, otherwise PaDEL will name it as “fingerprint.csv”, then it will be hard to provide the correct fingerprint table to `DComboNet`. After generating tables, you can provide different fingerprint table to the function. The example of how the fingerprint table lookes like is below:

```
CDK_FP  <- read.csv(paste0(load_dir,'test/OCILY3_data/structure_sim/fingerprint.csv'), sep = ',',header
DT::datatable(CDK_FP[,1:5], options = list(pageLength = 3))
```

```
pubchemFP <- read.csv(paste0(load_dir,'test/OCILY3_data/structure_sim/PubChem.csv'), sep = ',',header =
DT::datatable(pubchemFP[,1:5], options = list(pageLength = 3))
```

```r
MACCS_FP <- read.csv(paste0(load_dir,'test/OCILY3_data/structure_sim/MACCF.csv'), sep = ',',header = T,
DT::datatable(MACCS_FP[,1:5], options = list(pageLength = 3))
```

3) Drug-target gene table preparation

The next table you should provide is drug-targeted genes table. You can collect drug-targeted genes information from drugbank, TTD or other databases and publications. Afterwards, it is better to convert gene names to official gene names to be able to map in PPI network. You can use R package `bioMart` or other solutions to convert. The format of drug-target gene table should be as follow:

```r
drugtarget = read.csv(paste0(load_dir,'test/OCILY3_data/drug_target_oci_ly3.csv'), sep = ',',header = T
DT::datatable(drugtarget, options = list(pageLength = 3))
```

4) Extra information preparation for level two model

For level two model, you can first input `NULL` to parameters `drugDEG`, `drugDEP` and `cancergene`, if gene expression profiles before and after drug treatment of interested cancer cell line are not provided in LINCS and CCLE database, the function will reture an ERROR information. Then you should prepare your own tables for these three parameters. Here we provide examples for your preparation:

```r
drugDEG = read.table(paste0(load_dir,'test/OCILY3_data/drug_DEG_12hrs.txt'),sep='\t',header=T,stringsAs
DT::datatable(drugDEG, options = list(pageLength = 3))
```

```r
drugDEP = read.table(paste0(load_dir,'test/OCILY3_data/drug_DEP_12hrs.txt'),sep='\t',header=T,stringsAs
DT::datatable(drugDEP, options = list(pageLength = 3))
```

```r
cancergene = read.table(paste0(load_dir,'test/OCILY3_data/OCILY3_genelist.txt'),sep='\t',header=T,strin
DT::datatable(cancergene, options = list(pageLength = 3))
```

Prediction

After preparing data tables as above, you can run the prediction task simply as follow:

```r
DComboNet(load_dir = load_dir,
          resultdir = resultdir,
          model = 'L2',
          manual_input = FALSE,
          drugcandidate=drugcandidate,
          CDK_FP = CDK_FP,
          pubchemFP = pubchemFP,
          MACCS_FP = MACCS_FP,
          drugnetWeight = TRUE,
          featuretype = "integrated_score",
          drugtarget = drugtarget,
          cellline = 'OCILY3',
          drugDEG = drugDEG,
          drugDEG = drugDEP,
          cancergene = cancergene
         )
```

#sessionInfo

```r
sessionInfo()
```