Package 'MetPriCNet'

March 16, 2015

Version 1.0

Title Prioritizing the Candidate Metabolites Based on Composite Multi-omics Network

Author Qianlan Yao

Maintainer Qianlan Yao <yaoqianlan@sjtu.edu.cn>

Description MetPriCNet can implement prioritize the disease metabolites based on composite multi-omics network.

Depends R (>= 3.1.2),Matrix

Collate CNPM.R

LazyData Yes

License GPL (>= 2)

biocViews disease metabolites,multi-omics,composite network;

R topics documented:

`MetPriCNet-package` *Description of MetPriCNet Package*

Description

A R platform to prioritize candidate disease metabolites by MetPriCNet method which extend random walk analysis on multi-omics network.

Introduction

The main goal of MetPriCNet is to prioritize candidate disease metabolites by MetPriCNet method.

Author(s)

Qianlan Yao <yaoqianlan@sjtu.edu.cn>

envi *The Variables in The Environment Variable*

Description

The variables in the environment variable envi of the system.

Format

An environment variable

Details

The environment variable includes the variable `DiseaseInfList`, `genSeedExample`, `GMNetExample`, `GNetExample`, `MetaboliteInf`, `metSeedExample` etc. We can use the function `get` to obtain one of them.

Author(s)

Qianlan Yao <yaoqianlan@sjtu.edu.cn>

See Also

`initializeData`

getDiseaseInf *Get All Disease Information Provided by MetPriCNet Package*

Description

Get all disease information provided by MetPriCNet package by OMIMID. This function will provide you known disease genes and metabolites corresponding phenotype.

Usage

```
getDiseaseInf(OMIMID)
```

Arguments

`OMIMID`	A character. OMIM ID of one disease or phenotype.

Value

A list.

Author(s)

Qianlan Yao <yaoqianlan@sjtu.edu.cn>

See Also

getTopDiseaseMetabolites

Examples

```
## Not run:
#############    Get disease information   ################
##prostate cancer
diseaseInf<-getDiseaseInf("176807")
diseaseInf



## End(Not run)
```

---

getTopDiseaseMetabolites

Get the Disease Risk Metabolites

Description

prioritize the disease candidate metabolites by integrated multi-omics information.

Usage

```
getTopDiseaseMetabolites(pheSeed=NULL,genSeed=NULL,metSeed=NULL,
    candidates=NULL,showTop=30,gamma=0.7,x=1/3,y=1/3,a=1/3,b=1/3,
GNet,PNet,MNet,GMNet,PGNet,PMNet)
```

Arguments

pheSeed	A character vector, represents the disease users want to study. Users should input diseaseName.
genSeed	A character vector, The gene seeds are the known disease genes of corresponding phenotype. Users should input a gene seed vector.
metSeed	A character vector, The metabolite seeds are the known disease metabolites of corresponding phenotype. Users should input a metabolite seed vector.
candidates	A character vector. Users should input metabolite candidates, a metabolite seed vector.
showTop	An integer. The number of top ranked candidate metabolites users want to show.
gamma	Restart probability in RWR method. A value ranges from 0 to 1. The default value is 0.7.
x	Jumping probability between gene network and phenotype network. A value ranges from 0 to 1. The default value is 1/3.
y	Jumping probability between gene network and metabolite network. A value ranges from 0 to 1. The default value is 1/3.
a	a denote the importance of the gene network. A value ranges from 0 to 1. The default value is 1/3.

b	b denote the importance of the phenotype network. A value ranges from 0 to 1. The default value is 1/3.
GNet	An adjacency matrix of the gene network with colnames and rownames representing genes. The value in it ranges from 0 to 1.
PNet	An adjacency matrix of the phenotype network with colnames and rownames representing phenotypes. The value in it ranges from 0 to 1.
MNet	An adjacency matrix of the metabolite network with colnames and rownames representing metabolites. The value in it ranges from 0 to 1.
GMNet	An adjacency matrix of the gene-metabolite network with colnames representing metabolites and rownames representing genes. The value in it ranges from 0 to 1.
PGNet	An adjacency matrix of the phenotype-gene network with colnames representing genes and rownames representing phenotypes. The value in it ranges from 0 to 1.
PMNet	An adjacency matrix of the phenotype-metabolite network with colnames representing metabolites and rownames representing phenotypes. The value in it ranges from 0 to 1.

Details

The function `getTopDiseaseMetabolites` could get the top ranked candidate metabolites by prioritization of the disease candidate metabolites using known disease seed nodes to perform random walk on multi-omics network. Note: If users want to use seeds we provided, pheSeed, genSeed and metSeed can be obtained by `getDiseaseInf` and `envi`.

Value

A dataframe.

Author(s)

Qianlan Yao <yaoqianlan@sjtu.edu.cn>

See Also

`getDiseaseInf`

Examples

```
## Not run:
#############    Prioritize candidate metabolites   ################
#ls(envi)
## Here we use six fake matrix(networks).
GNetExample<-get("GNetExample",envir=envi);
MNetExample<-get("MNetExample",envir=envi);
PNetExample<-get("PNetExample",envir=envi);
GMNetExample<-get("GMNetExample",envir=envi);
PGNetExample<-get("PGNetExample",envir=envi);
PMNetExample<-get("PMNetExample",envir=envi);
## Get three types seeds from DiseaseInfList we provided.
diseaseName<-"Prostate cancer" ;
DiseaseInfList<-get("DiseaseInfList",envir=envi);
loci<-match(diseaseName,DiseaseInfList[["OMIMName"]]);
```

```
pheSeedExample<-DiseaseInfList[loci,"PheOMIM"];
genSeedExample<-unlist(strsplit(as.character(DiseaseInfList[loci,"DGenes"]),";"));
metSeedExample<-unlist(strsplit(as.character(DiseaseInfList[loci,"DMetabolites"]),";"));
PNodes<-colnames(PNetExample) ;
GNodes<-colnames(GNetExample);
MNodes<-colnames(MNetExample);
pheSeedResult<-pheSeedExample;
genSeedResult<-genSeedExample;
metSeedResult<-metSeedExample;

## Get candidates.
candidateResult<-setdiff(MNodes,metSeedResult);
##Run getTopDiseaseMetabolites
example<-getTopDiseaseMetabolites(pheSeed=pheSeedResult,genSeed=genSeedResult,metSeed=metSeedResult,
           candidates=candidateResult,showTop=200,GNet=GNetExample,PNet=PNetExample,MNet=MNetExample,
  GMNet=GMNetExample,PGNet=PGNetExample,PMNet=PMNetExample);
head(example)



## End(Not run)
```

`initializeData` *Initialize Environment*

Description

Initialize environment envi.

Usage

```
initializeData()
```

Details

The environment variable envi save many information. We can use the function `ls` to see the variable and use `ls(envi)` to see information in it, which includes `DiseaseInfList`, `genSeedExample`, `GMNetExample`, `GNetExample`, `MetaboliteInf`, `metSeedExample` etc. We can use the function `get` to obtain one of them.

Author(s)

Qianlan Yao <yaoqianlan@sjtu.edu.cn>

Examples

```
initializeData()

##see information in environment variable envi
ls(envi)

##obtain one variable in environment variable envi
DiseaseInfList<-get("DiseaseInfList",envir=envi)
```

Index