Package ‘Pi’

December 22, 2016

**Type** Package

**Title** Leveraging Genetic Evidence to Prioritise Drug Targets at the Gene, Pathway and Network Level

**Version** 1.0.0

**Date** 2016-10-12

**Author** Hai Fang, the ULTRA-DD Consortium, Julian C Knight

**Maintainer** Hai Fang <hfang@well.ox.ac.uk>

**Depends** XGR, igraph, dnet, ggplot2

**Imports** Matrix, MASS, ggbio, GenomicRanges, GenomeInfoDb, supraHex, scales, grDevices, graphics, stats, ggrepel, glmnet

**Suggests** foreach, doMC, BiocStyle, knitr, rmarkdown, png, GGally

**Description** Priority index or Pi is developed as a genomic-led target prioritisation system, with the focus on leveraging human genetic data to prioritise potential drug targets at the gene, pathway and network level. The long term goal is to use such information to enhance early-stage target validation. Based on evidence of disease association from genome-wide association studies (GWAS), this prioritisation system is able to generate evidence to support identification of the specific modulated genes (seed genes) that are responsible for the genetic association signal by utilising knowledge of linkage disequilibrium (co-inherited genetic variants), distance of associated variants from the gene, and evidence of independent genetic association with gene expression in disease-relevant tissues, cell types and states. Seed genes are scored in an integrative way, quantifying the genetic influence. Scored seed genes are subsequently used as baits to rank seed genes plus additional (non-seed) genes; this is achieved by iteratively exploring the global connectivity of a gene interaction network. Genes with the highest priority are further used to identify/prioritise pathways that are significantly enriched with highly prioritised genes. Prioritised genes are also used to identify a gene network interconnecting highly prioritised genes and a minimal number of less prioritised genes (which act as linkers bringing together highly prioritised genes).

**URL** [http://pi314.r-forge.r-project.org](http://pi314.r-forge.r-project.org)

**Collate** 'xRWR.r' 'xPier.r' 'xPierGenes.r' 'xPierSNPs.r'

'xPierPathways.r' 'xPierManhattan.r' 'xPierSubnet.r'

'xSNPeqtl.r' 'xSNP2eGenes.r' 'xPierSNPsConsensus.r'

'xPredictPR.r'

**License** GPL-3

**VignetteBuilder** knitr
biocViews  Software, Genetics, GraphAndNetwork, Pathways,
GeneExpression, GeneTarget, GenomeWideAssociation,
LinkageDisequilibrium, Network

NeedsCompilation no

R topics documented:

- xPier ........................................... 2
- xPierGenes ................................... 4
- xPierManhattan ............................... 7
- xPierPathways ................................. 8
- xPierSNPs .................................... 12
- xPierSNPsConsensus .......................... 17
- xPierSubnet .................................. 22
- xPredictPR .................................... 24
- xRWR ........................................... 26
- xSNP2eGenes .................................. 28
- xSNPeqtl ...................................... 30

Index 32

---

xPier  Function to do prioritisation through random walk techniques

Description

xPier is supposed to prioritise nodes given an input graph and a list of seed nodes. It implements Random Walk with Restart (RWR) and calculates the affinity score of all nodes in the graph to the seeds. The priority score is the affinity score. Parallel computing is also supported for Linux or Mac operating systems. It returns an object of class "pNode".

Usage

xPier(seeds, g, normalise = c("laplacian", "row", "column", "none"),
      restart = 0.75, normalise.affinity.matrix = c("none", "quantile"),
      parallel = TRUE, multicores = NULL, verbose = TRUE)

Arguments

- seeds  a named input vector containing a list of seed nodes. For this named vector, the element names are seed/node names (e.g. gene symbols), the element (non-zero) values used to weight the relative importance of seeds. Alternatively, it can be a matrix or data frame with two columns: 1st column for seed/node names, 2nd column for the weight values
- g  an object of class "igraph" to represent network. It can be a weighted graph with the node attribute 'weight'
- normalise  the way to normalise the adjacency matrix of the input graph. It can be 'laplacian' for laplacian normalisation, 'row' for row-wise normalisation, 'column' for column-wise normalisation, or 'none'
restart

The restart probability used for Random Walk with Restart (RWR). The restart probability takes the value from 0 to 1, controlling the range from the starting nodes/seeds that the walker will explore. The higher the value, the more likely the walker is to visit the nodes centered on the starting nodes. At the extreme when the restart probability is zero, the walker moves freely to the neighbors at each step without restarting from seeds, i.e., following a random walk (RW).

normalise.affinity.matrix

The way to normalise the output affinity matrix. It can be 'none' for no normalisation, 'quantile' for quantile normalisation to ensure that columns (if multiple) of the output affinity matrix have the same quantiles.

parallel

Logical to indicate whether parallel computation with multicores is used. By default, it sets to true, but not necessarily does so. Partly because parallel backends available will be system-specific (now only Linux or Mac OS). Also, it will depend on whether these two packages "foreach" and "doMC" have been installed. It can be installed via: source("http://bioconductor.org/biocLite.R"); biocLite(c("foreach","doMC")). If not yet installed, this option will be disabled.

multicores

An integer to specify how many cores will be registered as the multicore parallel backend to the 'foreach' package. If NULL, it will use a half of cores available in a user's computer. This option only works when parallel computation is enabled.

verbose

Logical to indicate whether the messages will be displayed in the screen. By default, it sets to true for display.

Value

An object of class "pNode", a list with following components:

- priority: a matrix of nNode X 4 containing node priority information, where nNode is the number of nodes in the input graph, and the 4 columns are "name" (node names), "seed" (1 for seeds, 0 for non-seeds), "weight" (weight values), "priority" (the priority scores that are rescaled to the range [0,1]), "rank" (ranks of the priority scores)
- g: an input "igraph" object
- call: the call that produced this result

Note

The input graph will treat as an unweighted graph if there is no 'weight' edge attribute associated with

See Also

xRDataLoader, xRWR, xPierSNPs, xPierGenes, xPierPathways

Examples

```r
## Not run:
# Load the library
library(Pi)
## End(Not run)

# a) provide the input nodes/genes with the significance info
sig <- rbeta(500, shape1=0.5, shape2=1)
```
## Not run:
## load human genes
org.Hs.eg <- xRDataLoader(RData='/'
org.Hs.eg
]

data <- data.frame(symbols=org.Hs.eg$gene_info$Symbol[1:500], sig)

# b) provide the network
g <- xRDataLoader(RData.customised='org.Hs.string')

# c) perform priority analysis
pNode <- xPier(seeds=data, g=g, restart=0.75)

## End(Not run)

---

xPierGenes

Function to priorise genes from an input network and the weight info imposed on its nodes

### Description

xPierGenes is supposed to prioritise genes given an input graph and a list of seed nodes. It implements Random Walk with Restart (RWR) and calculates the affinity score of all nodes in the graph to the seeds. The priority score is the affinity score. Parallel computing is also supported for Linux or Mac operating systems. It returns an object of class "pNode".

### Usage

```r
xPierGenes(data, network = c("STRING_highest", "STRING_high",
"STRING_medium", "STRING_low", "PCommonsUN_high", "PCommonsUN_medium",
"PCommonsDN_high", "PCommonsDN_medium", "PCommonsDN_Reactome", "PCommonsDN_KEGG",
"PCommonsDN_HumanCyc", "PCommonsDN_PID", "PCommonsDN_PANTHER",
"PCommonsDN_ReconX", "PCommonsDN_TRANSFAC", "PCommonsDN_PhosphoSite",
"PCommonsDN_CTD"), weighted = FALSE, network.customised = NULL,
normalise = c("laplacian", "row", "column", "none"), restart = 0.75,
normalise.affinity.matrix = c("none", "quantile"), parallel = TRUE,
multicores = NULL, verbose = TRUE,
RData.location = "https://github.com/hfang-bristol/RDataCentre/blob/master/Portal")
```

### Arguments

- **data**
  - A named input vector containing a list of seed nodes (ie gene symbols). For this named vector, the element names are seed/node names (e.g. gene symbols), the element (non-zero) values used to weight the relative importance of seeds. Alternatively, it can be a matrix or data frame with two columns: 1st column for seed/node names, 2nd column for the weight values.

- **network**
  - The built-in network. Currently two sources of network information are supported: the STRING database (version 10) and the Pathways Commons database (version 7). STRING is a meta-integration of undirect interactions from the functional aspect, while Pathways Commons mainly contains both undirect and direct interactions from the physical/pathway aspect. Both have scores to control
the confidence of interactions. Therefore, the user can choose the different quality of the interactions. In STRING, "STRING_highest" indicates interactions with highest confidence (confidence scores >= 900), "STRING_high" for interactions with high confidence (confidence scores >= 700), "STRING_medium" for interactions with medium confidence (confidence scores >= 400), and "STRING_low" for interactions with low confidence (confidence scores >= 150). For undirect/physical interactions from Pathways Commons, "PCommonsUN_high" indicates undirect interactions with high confidence (supported with the PubMed references plus at least 2 different sources), "PCommonsUN_medium" for undirect interactions with medium confidence (supported with the PubMed references). For direct (pathway-merged) interactions from Pathways Commons, "PCommonsDN_high" indicates direct interactions with high confidence (supported with the PubMed references plus at least 2 different sources), and "PCommonsDN_medium" for direct interactions with medium confidence (supported with the PubMed references). In addition to pooled version of pathways from all data sources, the user can also choose the pathway-merged network from individual sources, that is, "PCommonsDN_Reactome" for those from Reactome, "PCommonsDN_KEGG" for those from KEGG, "PCommonsDN_HumanCyc" for those from HumanCyc, "PCommonsDN_PID" for those from PID, "PCommonsDN_PANTHER" for those from PANTHER, "PCommonsDN_ReconX" for those from ReconX, "PCommonsDN_TRANSFAC" for those from TRANSFAC, "PCommonsDN_PhosphoSite" for those from PhosphoSite, and "PCommonsDN_CTD" for those from CTD.

weighted logical to indicate whether edge weights should be considered. By default, it sets to false. If true, it only works for the network from the STRING database.

network.customised an object of class "igraph". By default, it is NULL. It is designed to allow the user analysing their customised network data that are not listed in the above argument 'network'. This customisation (if provided) has the high priority over built-in network. If the user provides the "igraph" object with the "weight" edge attribute, RWR will assume to walk on the weighted network.

normalise the way to normalise the adjacency matrix of the input graph. It can be 'laplacian' for laplacian normalisation, 'row' for row-wise normalisation, 'column' for column-wise normalisation, or 'none'.

restart the restart probability used for Random Walk with Restart (RWR). The restart probability takes the value from 0 to 1, controlling the range from the starting nodes/seeds that the walker will explore. The higher the value, the more likely the walker is to visit the nodes centered on the starting nodes. At the extreme when the restart probability is zero, the walker moves freely to the neighbors at each step without restarting from seeds, i.e., following a random walk (RW).

normalise.affinity.matrix the way to normalise the output affinity matrix. It can be 'none' for no normalisation, 'quantile' for quantile normalisation to ensure that columns (if multiple) of the output affinity matrix have the same quantiles.

parallel logical to indicate whether parallel computation with multicore is used. By default, it sets to true, but not necessarily does so. Partly because parallel backends available will be system-specific (now only Linux or Mac OS). Also, it will depend on whether these two packages "foreach" and "doMC" have been installed. It can be installed via: source("http://bioconductor.org/biocLite.R"); biocLite(c("foreach","doMC")). If not yet installed, this option will be disabled.
multicores   an integer to specify how many cores will be registered as the multicore parallel backend to the 'foreach' package. If NULL, it will use a half of cores available in a user's computer. This option only works when parallel computation is enabled

verbose     logical to indicate whether the messages will be displayed in the screen. By default, it sets to true for display

RData.location the characters to tell the location of built-in RData files. See xRDataLoader for details

Value

an object of class "pNode", a list with following components:

• priority: a matrix of nNode X 4 containing node priority information, where nNode is the number of nodes in the input graph, and the 4 columns are "name" (node names), "seed" (1 for seeds, 0 for non-seeds), "weight" (weight values), "priority" (the priority scores that are rescaled to the range [0,1]), "rank" (ranks of the priority scores)

• g: an input "igraph" object

• call: the call that produced this result

Note

The input graph will treat as an unweighted graph if there is no 'weight' edge attribute associated with

See Also

xRDataLoader, xPierSNPs, xPier, xPierPathways

Examples

```r
## Not run:
# Load the library
library(Pi)

## End(Not run)

# a) provide the seed nodes/genes with the weight info
## load ImmunoBase
ImmunoBase <- xRDataLoader(RData.customised=’/quotesingle.Var
ImmunoBase/quotesingle.Var

## get genes within 500kb away from AS GWAS lead SNPs
seeds.genes <- ImmunoBase$AS$genes_variants

## seeds weighted according to distance away from lead SNPs
data <- 1-seeds.genes/500000

## Not run:
# b) perform priority analysis
pNode <- xPierGenes(data=data, network="PCommonsDN_medium", restart=0.7)

## End(Not run)

# c) save to the file called 'Genes_priority.txt'
write.table(pNode$priority, file="Genes_priority.txt", sep="\t", row.names=FALSE)

## End(Not run)
```
**xPierManhattan**  

Function to visualise prioritised genes using manhattan plot

---

**Description**

xPierManhattan is supposed to visualise prioritised genes using manhattan plot. Genes with the top priority are highlighted. It returns an object of class "ggplot".

**Usage**

```r
xPierManhattan(pNode, color = c("darkred", "darkgreen"), cex = 0.5,
highlight.top = 20, highlight.col = "deepskyblue",
highlight.label.size = 2, highlight.label.offset = 0.02,
highlight.label.col = "darkblue", y.scale = c("normal", "sqrt",
"log10"),
GR.Gene = c("UCSC_knownGene", "UCSC_knownCanonical"), verbose = TRUE,
RData.location = "https://github.com/hfang-bristol/RDataCentre/blob/master/Portal")
```

**Arguments**

- `pNode`: an object of class "pNode"
- `color`: a character vector for point colors to alternate
- `cex`: a numeric value for point size
- `highlight.top`: the number of the top targets to be highlighted
- `highlight.col`: the highlight colors
- `highlight.label.size`: the highlight label size
- `highlight.label.offset`: the highlight label offset
- `highlight.label.col`: the highlight label color
- `y.scale`: how to transform the y scale. It can be "normal" for no transformation, "sqrt" for square transformation, and "log10" for 10-based log-transformation
- `GR.Gene`: the genomic regions of genes. By default, it is 'UCSC_knownGene', that is, UCSC known genes (together with genomic locations) based on human genome assembly hg19. It can be 'UCSC_knownCanonical', that is, UCSC known canonical genes (together with genomic locations) based on human genome assembly hg19. Alternatively, the user can specify the customised input. To do so, first save your RData file (containing a GR object) into your local computer, and make sure the GR object content names refer to Gene Symbols. Then, tell "GR.Gene" with your RData file name (with or without extension), plus specify your file RData path in "RData.location"
- `verbose`: logical to indicate whether the messages will be displayed in the screen. By default, it sets to false for no display
- `RData.location`: the characters to tell the location of built-in RData files. See `xRDataLoader` for details
xPierPathways

Function to prioritise pathways based on enrichment analysis of top prioritised genes

Description

xPierPathways is supposed to prioritise pathways given prioritised genes and the ontology in query. It returns an object of class "eTerm". It is done via enrichment analysis.
Usage

```r
xPierPathways(pNode, priority.top = 100, background = NULL,
ontology = c("GOBP", "GOMF", "GOCC", "PS", "PS2", "SF", "DO", "HPPA",
"HPMI", "HPMA", "MP", "MsigdbBH", "MsigdbC1", "MsigdbC2CGP",
"MsigdbC2BIOCARTA", "MsigdbC3TFT", "MsigdbC3MIR", "MsigdbC4CGN",
"MsigdbC4CM",
"MsigdbC5BP", "MsigdbC5MF", "MsigdbC5CC", "MsigdbC6", "MsigdbC7",
"DGIdb"),
size.range = c(10, 2000), min.overlap = 3, which.distance = NULL,
test = c("hypergeo", "fisher", "binomial"), p.adjust.method = c("BH",
"BY", "bonferroni", "holm", "hochberg", "hommel"),
onontology.algorithm = c("none", "pc", "elim", "lea"), elim.pvalue = 0.01,
lea.depth = 2, path.mode = c("all_paths", "shortest_paths",
"all_shortest_paths"), true.path.rule = FALSE, verbose = TRUE,
RData.location = "https://github.com/hfang-bristol/RDataCentre/blob/master/Portal")
```

Arguments

- **pNode**: an object of class "pNode"
- **priority.top**: the number of the top targets to be analysed for pathway enrichment
- **background**: a background vector. It contains a list of Gene Symbols as the test background. If NULL, by default all annotatable are used as background
- **ontology**: the ontology supported currently. It can be "GOBP" for GeneOntology Biological Process, "GOMF" for Gene Ontology Molecular Function, "GOCC" for Gene Ontology Cellular Component, "PS" for phylostratific age information, "PS2" for the collapsed PS version (inferred ancestors being collapsed into one with the known taxonomy information), "SF" for domain superfamilly assignments, "DO" for Disease Ontology, "HPPA" for Human Phenotype Phenotypic Abnormality, "HPMI" for Human Phenotype Mode of Inheritance, "HPCM" for Human Phenotype Clinical Modifier, "HPMA" for Human Phenotype Mortality Aging, "MP" for Mammalian Phenotype, and Drug-Gene Interaction database (DGIdb) for drugable categories, and the molecular signatures database (Msigdb, including "MsigdbH", "MsigdbC1", "MsigdbC2CGP", "MsigdbC2CPall", "MsigdbC2CP", "MsigdbC2KEGG", "MsigdbC2REACTOME",
"MsigdbC2BIOCARTA", "MsigdbC3TFT", "MsigdbC3MIR", "MsigdbC4CGN",
"MsigdbC4CM", "MsigdbC5BP", "MsigdbC5MF", "MsigdbC5CC", "MsigdbC6",
"MsigdbC7")
- **size.range**: the minimum and maximum size of members of each term in consideration. By default, it sets to a minimum of 10 but no more than 2000
- **min.overlap**: the minimum number of overlaps. Only those terms with members that overlap with input data at least min.overlap (by default) will be processed
- **which.distance**: which terms with the distance away from the ontology root (if any) is used to restrict terms in consideration. By default, it sets to 'NULL' to consider all distances
- **test**: the statistic test used. It can be "fisher" for using fisher's exact test, "hypergeo" for using hypergeometric test, or "binomial" for using binomial test. Fisher's exact test is to test the independence between gene group (genes belonging to
a group or not) and gene annotation (genes annotated by a term or not), and thus compare sampling to the left part of background (after sampling without replacement). Hypergeometric test is to sample at random (without replacement) from the background containing annotated and non-annotated genes, and thus compare sampling to background. Unlike hypergeometric test, binomial test is to sample at random (with replacement) from the background with the constant probability. In terms of the ease of finding the significance, they are in order: hypergeometric test > binomial test > fisher's exact test. In other words, in terms of the calculated p-value, hypergeometric test < binomial test < fisher's exact test

**p.adjust.method**

the method used to adjust p-values. It can be one of "BH", "BY", "bonferroni", "holm", "hochberg" and "hommel". The first two methods "BH" (widely used) and "BY" control the false discovery rate (FDR: the expected proportion of false discoveries amongst the rejected hypotheses); the last four methods "bonferroni", "holm", "hochberg" and "hommel" are designed to give strong control of the family-wise error rate (FWER). Notes: FDR is a less stringent condition than FWER

**ontology.algorithm**

the algorithm used to account for the hierarchy of the ontology. It can be one of "none", "pc", "elim" and "lea". For details, please see 'Note' below

**elim.pvalue**

the parameter only used when "ontology.algorithm" is "elim". It is used to control how to declare a significantly enriched term (and subsequently all genes in this term are eliminated from all its ancestors)

**lea.depth**

the parameter only used when "ontology.algorithm" is "lea". It is used to control how many maximum depth is used to consider the children of a term (and subsequently all genes in these children term are eliminated from the use for the recalculation of the significance at this term)

**path.mode**

the mode of paths induced by vertices/nodes with input annotation data. It can be "all_paths" for all possible paths to the root, "shortest_paths" for only one path to the root (for each node in query), "all_shortest_paths" for all shortest paths to the root (i.e. for each node, find all shortest paths with the equal lengths)

**true.path.rule**

logical to indicate whether the true-path rule should be applied to propagate annotations. By default, it sets to false

**verbose**

logical to indicate whether the messages will be displayed in the screen. By default, it sets to false for no display

**RData.location**

the characters to tell the location of built-in RData files. See [xRDataLoader](#) for details

### Value

an object of class "eTerm", a list with following components:

- **term_info**: a matrix of nTerm X 4 containing snp/gene set information, where nTerm is the number of terms, and the 4 columns are "id" (i.e. "Term ID"), "name" (i.e. "Term Name"), "namespace" and "distance"

- **annotation**: a list of terms containing annotations, each term storing its annotations. Always, terms are identified by "id"

- **data**: a vector containing input data in consideration. It is not always the same as the input data as only those mappable are retained
• background: a vector containing the background data. It is not always the same as the input data as only those mappable are retained
• overlap: a list of overlapped snp/gene sets, each storing snps overlapped between a snp/gene set and the given input data (i.e. the snps of interest). Always, gene sets are identified by "id"
• zscore: a vector containing z-scores
• pvalue: a vector containing p-values
• adjp: a vector containing adjusted p-values. It is the p value but after being adjusted for multiple comparisons
• call: the call that produced this result

Note

The interpretation of the algorithms used to account for the hierarchy of the ontology is:

• "none": does not consider the ontology hierarchy at all.
• "lea": computers the significance of a term in terms of the significance of its children at the maximum depth (e.g. 2). Precisely, once snps are already annotated to any children terms with a more significance than itself, then all these snps are eliminated from the use for the recalculation of the significance at that term. The final p-values takes the maximum of the original p-value and the recalculated p-value.
• "elim": computes the significance of a term in terms of the significance of its all children. Precisely, once snps are already annotated to a significantly enriched term under the cutoff of e.g. pvalue<1e-2, all these snps are eliminated from the ancestors of that term.
• "pc": requires the significance of a term not only using the whole snps as background but also using snps annotated to all its direct parents/ancestors as background. The final p-value takes the maximum of both p-values in these two calculations.
• "Notes": the order of the number of significant terms is: "none" > "lea" > "elim" > "pc".

See Also

xRDataLoader, xEnricher

Examples

```r
## Not run:
# Load the library
library(Pi)

## End(Not run)

# a) provide the seed nodes/genes with the weight info
## load ImmunoBase
ImmunoBase <- xRDataLoader(RData.customised='ImmunoBase')
## get genes within 500kb away from AS GWAS lead SNPs
seeds.genes <- ImmunoBase$AS$genes_variants
## seeds weighted according to distance away from lead SNPs
data <- 1- seeds.genes/500000

## Not run:
# b) perform priority analysis
pNode <- xPierGenes(data=data, network="PCommonsDN_medium", restart=0.7)
```
# c) derive pathway-level priority
eTerm <- xPierPathways(pNode=pNode, priority.top=100,
ontology="MsigdbC2CP")

# d) view enrichment results for the top significant terms
xEnrichViewer(eTerm)

# e) save enrichment results to the file called 'Pathways_priority.txt'
res <- xEnrichViewer(eTerm, top_num=length(eTerm$adjp), sortBy="adjp",
details=TRUE)
output <- data.frame(term=rownames(res), res)
utils::write.table(output, file="Pathways_priority.txt", sep="\t",
row.names=FALSE)

## End(Not run)

xPierSNPs

Function to priorise genes given a list of seed SNPs together with the
significance level (e.g. GWAS reported p-values)

Description

xPierSNPs is supposed to priorise genes given a list of seed SNPs together with the significance
level. To priorise genes, it first defines and scores seed genes: nearby genes and eQTL genes. With
seed genes and their scores, it then uses Random Walk with Restart (RWR) to calculate the affinity
score of all nodes in the input graph to the seed genes. The priority score is the affinity score.
Parallel computing is also supported for Linux or Mac operating systems. It returns an object of
class "pNode".

Usage

xPierSNPs(data, include.LD = NA, LD.customised = NULL, LD.r2 = 0.8,
significance.threshold = 5e-05, distance.max = 2e+05,
decay.kernel = c("rapid", "slow", "linear"), decay.exponent = 2,
GR.SNP = c("dbSNP_GWAS", "dbSNP_Common"), GR.Gene = c("UCSC_knownGene",
"UCSC_knownCanonical"), include.eQTL = c(NA, "JKscience_TS2A",
"JKscience_TS2B", "JKscience_TS3A", "JKng_bcell", "JKng_mono",
"JKnc_neutro",
"JK_nk", "GTex_V4_Artery_Aorta", "GTex_V4_Artery_Tibial", "GTex_V4_Eosophagus_Mucosa",
"GTex_V4_Eosophagus_Muscularis", "GTex_V4_Heart_Left_Ventricle",
"GTex_V4_Lung", "GTex_V4_Muscle_Skeletal", "GTex_V4_Nerve_Tibial",
"GTex_V4_Skin_Sun_Exposed_Lower_leg", "GTex_V4_Stomach",
"GTex_V4_Thyroid",
"GTex_V4_Whole_Blood", "eQTLdb_NK", "eQTLdb_CD14", "eQTLdb_LPS2",
"eQTLdb_LPS24", "eQTLdb_IFN"), eQTL.customised = NULL,
cdf.function = c("empirical", "exponential"), relative.importance = c(0.5,
0.5), scoring.scheme = c("max", "sum", "sequential"),
network = c("STRING_highest", "STRING_high", "STRING_medium",
"STRING_low",
"PCommonsUN_high", "PCommonsUN_medium", "PCommonsDN_high",
..."...
"PCommonsDN_medium", "PCommonsDN_Reactome", "PCommonsDN_KEGG",
"PCommonsDN_HumanCyc", "PCommonsDN_PID", "PCommonsDN_PANTHER",
"PCommonsDN_ReconX", "PCommonsDN_TRANSFAC", "PCommonsDN_PhosphoSite",
"PCommonsDN_CTD"), weighted = FALSE, network.customised = NULL,
normalise = c("laplacian", "row", "column", "none"), restart = 0.75,
normalise.affinity.matrix = c("none", "quantile"), parallel = TRUE,
multicores = NULL, verbose = TRUE,
RData.location = "https://github.com/hfang-bristol/RDataCentre/blob/master/Portal")

Arguments

data

A named input vector containing the significance level for nodes (dbSNP). For
this named vector, the element names are dbSNP ID (or in the format such as
'chr16:28525386'), the element values for the significance level (measured as p-
value or fdr). Alternatively, it can be a matrix or data frame with two columns:
1st column for dbSNP, 2nd column for the significance level

include.LD

Additional SNPs in LD with Lead SNPs are also included. By default, it is 'NA'
to disable this option. Otherwise, LD SNPs will be included based on one or
more of 26 populations and 5 super populations from 1000 Genomics Project
data (phase 3). The population can be one of 5 super populations ('AFR',
"AMR", "EUR", "SAS"), or one of 26 populations ('ACB', "ASW", "BEB", "CDX", "CEU", "CHB", "CHS", "CLM", "ESN", "FIN", "GBR", "GIH",
"PUR", "STU", "TSI", "YRI"). Explanations for population code can be found
at http://www.1000genomes.org/faq/which-populations-are-part-your-study

LD.customised

A user-input matrix or data frame with 3 columns: 1st column for Lead SNPs,
2nd column for LD SNPs, and 3rd for LD r2 value. It is designed to allow the
user analysing their pre-calculated LD info. This customisation (if provided)
has the high priority over built-in LD SNPs

LD.r2

The LD r2 value. By default, it is 0.8, meaning that SNPs in LD (r2>=0.8) with
input SNPs will be considered as LD SNPs. It can be any value from 0.8 to 1

significance.threshold

The given significance threshold. By default, it is set to NULL, meaning there is
no constraint on the significance level when transforming the significance level
of SNPs into scores. If given, those SNPs below this are considered significant
and thus scored positively. Instead, those above this are considered insignificant
and thus receive no score

distance.max

The maximum distance between genes and SNPs. Only those genes no far way
from this distance will be considered as seed genes. This parameter will influ-
ence the distance-component weights calculated for nearby SNPs per gene

decay.kernel

A character specifying a decay kernel function. It can be one of 'slow' for slow
decay, 'linear' for linear decay, and 'rapid' for rapid decay

decay.exponent

An integer specifying a decay exponent. By default, it sets to 2

GR.SNP

The genomic regions of SNPs. By default, it is 'dbSNP_GWAS', that is, SNPs
from dbSNP (version 146) restricted to GWAS SNPs and their LD SNPs (hg19).
It can be 'dbSNP_Common', that is, Common SNPs from dbSNP (version 146)
plus GWAS SNPs and their LD SNPs (hg19). Alternatively, the user can specify
the customised input. To do so, first save your RData file (containing an GR
object) into your local computer, and make sure the GR object content names
PierSNPs refer to dbSNP IDs. Then, tell "GR.SNP" with your RData file name (with or without extension), plus specify your file RData path in "RData.location". Note: you can also load your customised GR object directly

GR.Gene
the genomic regions of genes. By default, it is 'UCSC_knownGene', that is, UCSC known genes (together with genomic locations) based on human genome assembly hg19. It can be 'UCSC_knownCanonical', that is, UCSC known canonical genes (together with genomic locations) based on human genome assembly hg19. Alternatively, the user can specify the customised input. To do so, first save your RData file (containing an GR object) into your local computer, and make sure the GR object content names refer to Gene Symbols. Then, tell "GR.Gene" with your RData file name (with or without extension), plus specify your file RData path in "RData.location". Note: you can also load your customised GR object directly

include.eQTL
genes modulated by eQTL (also Lead SNPs or in LD with Lead SNPs) are also included. By default, it is 'NA' to disable this option. Otherwise, those genes modulated by eQTL will be included: immune stimulation in monocytes ("JKscience_TS1A" and "JKscience_TS2B" for cis-eQTLs or "JKscience_TS3A" for trans-eQTLs) from Science 2014, 343(6175):1246949; cis- and trans-eQTLs in B cells ("JKng_bcell") and in monocytes ("JKng_mono") from Nature Genetics 2012, 44(3):502-510; cis- and trans-eQTLs in neutrophils ("JKne_neutro") from Nature Communications 2015, 7(6):7545; cis-eQTLs in NK cells ("JK_nk") which is unpublished. Also supported are GTEx cis-eQTLs from Science 2015, 348(6235):648-60, including 13 tissues: "GTEx_V4_Adipose_Subcutaneous", "GTEx_V4_Artery_Aorta", 'exponential' based on exponential cdf, 'empirical' for empirical cdf

eQTL.customised
a user-input matrix or data frame with 3 columns: 1st column for SNPs/eQTLs, 2nd column for Genes, and 3rd for eQTL mapping significance level (p-values or FDR). It is designed to allow the user analysing their eQTL data. This customisation (if provided) has the high priority over built-in eQTL data.

cdf.function
a character specifying a Cumulative Distribution Function (cdf). It can be one of 'exponential' based on exponential cdf, 'empirical' for empirical cdf

relative.importance
a vector specifying the relative importance of the distance weight and eQTL weight

scoring.scheme
the method used to calculate seed gene scores under a set of SNPs. It can be one of "sum" for adding up, "max" for the maximum, and "sequential" for the sequential weighting. The sequential weighting is done via: $\sum_{i=1}^{R_i} \frac{R_i}{i}$, where $R_i$ is the $i^{th}$ rank (in a decreasing order)

network
the built-in network. Currently two sources of network information are supported: the STRING database (version 10) and the Pathways Commons database (version 7). STRING is a meta-integration of undirect interactions from the functional aspect, while Pathways Commons mainly contains both undirect and direct interactions from the physical/pathway aspect. Both have scores to control the confidence of interactions. Therefore, the user can choose the different quality of the interactions. In STRING, "STRING_highest" indicates interactions with highest confidence (confidence scores>=900), "STRING_high" for interactions with high confidence (confidence scores>=700), "STRING_medium" for interactions with medium confidence (confidence scores>=400), and "STRING_low" for interactions with low confidence (confidence scores=150). For undirect/physical interactions from Pathways Commons, "PCommonsUN_high" indicates undirect interactions with high confidence (supported with the PubMed references
For direct (pathway-merged) interactions from Pathways Commons, "PCommonsDN_high" indicates direct interactions with high confidence (supported with the PubMed references plus at least 2 different sources), and "PCommonsUN_medium" for direct interactions with medium confidence (supported with the PubMed references). In addition to pooled version of pathways from all data sources, the user can also choose the pathway-merged network from individual sources, that is, "PCommonsDN_Reactome" for those from Reactome, "PCommonsDN_KEGG" for those from KEGG, "PCommonsDN_HumanCyc" for those from HumanCyc, "PCommonsDN_PID" for those from PID, "PCommonsDN_PANTHER" for those from PANTHER, "PCommonsDN_ReconX" for those from ReconX, "PCommonsDN_TRANSFAC" for those from TRANSFAC, "PCommonsDN_PhosphoSite" for those from PhosphoSite, and "PCommonsDN_CTD" for those from CTD

weighted logical to indicate whether edge weights should be considered. By default, it sets to false. If true, it only works for the network from the STRING database

data

network.customised

an object of class "igraph". By default, it is NULL. It is designed to allow the user analysing their customised network data that are not listed in the above argument 'network'. This customisation (if provided) has the high priority over built-in network. If the user provides the "igraph" object with the "weight" edge attribute, RWR will assume to walk on the weighted network

normalise

the way to normalise the adjacency matrix of the input graph. It can be 'laplacian' for laplacian normalisation, 'row' for row-wise normalisation, 'column' for column-wise normalisation, or 'none'

restart

the restart probability used for Random Walk with Restart (RWR). The restart probability takes the value from 0 to 1, controlling the range from the starting nodes/seeds that the walker will explore. The higher the value, the more likely the walker is to visit the nodes centered on the starting nodes. At the extreme when the restart probability is zero, the walker moves freely to the neighbors at each step without restarting from seeds, i.e., following a random walk (RW)

normalise.affinity.matrix

the way to normalise the output affinity matrix. It can be 'none' for no normalisation, 'quantile' for quantile normalisation to ensure that columns (if multiple) of the output affinity matrix have the same quantiles

parallel

logical to indicate whether parallel computation with multicores is used. By default, it sets to true, but not necessarily does so. Partly because parallel backends available will be system-specific (now only Linux or Mac OS). Also, it will depend on whether these two packages "foreach" and "doMC" have been installed. It can be installed via: source("http://bioconductor.org/biocLite.R"); biocLite(c("foreach","doMC")). If not yet installed, this option will be disabled

multicores

an integer to specify how many cores will be registered as the multicore parallel backend to the 'foreach' package. If NULL, it will use a half of cores available in a user's computer. This option only works when parallel computation is enabled

verbose

logical to indicate whether the messages will be displayed in the screen. By default, it sets to true for display

RData.location

the characters to tell the location of built-in RData files. See xRDataLoader for details
Value

an object of class "pNode", a list with following components:

- priority: a matrix of nNode X 4 containing node priority information, where nNode is the number of nodes in the input graph, and the 4 columns are "name" (node names), "seed" (1 for seeds, 0 for non-seeds), "weight" (weight/score values for seed genes), "priority" (the priority scores that are rescaled to the range [0,1]), "rank" (ranks of the priority scores)

- g: an input "igraph" object

- SNP: a data frame of nSNP X 3 containing input SNPs and/or LD SNPs info, where nSNP is the number of input SNPs and/or LD SNPs, and the 4 columns are "SNP" (dbSNP), "Score" (the SNP score), "Pval" (the SNP p-value), "Flag" (indicative of Lead SNPs or LD SNPs)

- Gene2SNP: a data frame of nPair X 3 containing Gene-SNP pair info, where nPair is the number of Gene-SNP pairs, and the 3 columns are "Gene" (seed genes), "SNP" (dbSNP), "Score" (an SNP's genetic influential score on a seed gene), "Pval" (the SNP p-value)

- call: the call that produced this result

Note

The search procedure is heuristic to find the subgraph with the maximum score:

- i) xSNPscores used to calculate the SNP score.
- ii) xSNP2nGenes used to define and score the nearby genes.
- iii) xSNP2eGenes used to define and score the eQTL genes.
- iv) define seed genes as the nearby genes in ii) and the eQTL genes in iii), which are then scored in an integrative manner.
- v) xPierGenes used to prioritise genes using an input graph and a list of seed genes and their scores from iv). The priority score is the affinity score estimated by Random Walk with Restart (RWR), measured as the affinity of all nodes in the graph to the seeds.

See Also

xSNPscores, xSNP2nGenes, xSNP2eGenes, xSparseMatrix, xSM2DF, xPier, xPierGenes, xPierPathways

Examples

```r
## Not run:
# Load the library
library(Pi)
## End(Not run)
# a) provide the SNPs with the significance info
## get lead SNPs reported in AS GWAS and their significance info (p-values)
#data.file <- "http://galahad.well.ox.ac.uk/bigdata/AS.txt"
#AS <- read.delim(data.file, header=TRUE, stringsAsFactors=FALSE)
ImmuNoBase <- xRDataLoader(RData.customised="ImmunoBase")
gr <- ImmunoBase$AS$variants
AS <- as.data.frame(GenomicRanges::mcols(gr)[, c('Variant','Pvalue')])
## Not run:
# b) perform priority analysis
pNode <- xPierSNPs(data=AS, include.LD=NA,
```

**Description**

`xPierSNPsConsensus` is supposed to prioritise genes given a list of seed SNPs together with the significance level. It is a parameter-free version of `xPierSNPs` identifying the consensus rank (less sensitive to the relative importance of the distance weight and eQTL weight). It returns an object of class "pNode" but appended with components on optimal distance weight and consensus info.

**Usage**

```r
xPierSNPsConsensus(data, include.LD = NA, LD.customised = NULL, 
  LD.r2 = 0.8, significance.threshold = 5e-05, distance.max = 2e+05, 
  decay.kernel = c("rapid", "slow", "linear"), decay.exponent = 2, 
  GR.SNP = c("dbSNP_GWAS", "dbSNP_Common"), GR.Gene = c("UCSC_knownGene", 
    "UCSC_knownCanonical"), include.eQTL = c(NA, "JKscience_TS2A", 
    "JKscience_TS2B", "JKscience_TS3A", "JKng_bcell", "JKng_mono", 
    "JKnc_neutro", 
    "JK_nk", "GTEX_V4_Artery_Aorta", "GTEX_V4_Artery_Tibial", "GTEX_V4_Esophagus_Mucosa", 
    "GTEX_V4_Esophagus_Muscularis", "GTEX_V4_Heart_Left_Ventricle", 
    "GTEX_V4_Lung", "GTEX_V4_Muscle_Skeletal", "GTEX_V4_Nerve_Tibial", 
    "GTEX_V4_Skin_Sun_Exposed_Lower_leg", "GTEX_V4_Stomach", 
    "GTEX_V4_Thyroid", "GTEX_V4_Whole_Blood", "eQTLdb_NK", "eQTLdb_CD14", "eQTLdb_LPS2", 
    "eQTLdb_LPS24", "eQTLdb_IFN"), eQTL.customised = NULL, 
  cdf.function = c("empirical", "exponential"), scoring.scheme = c("max", 
    "sum", "sequential"), network = c("STRING_highest", "STRING_high", 
    "STRING_medium", "STRING_low", "PCommonsUN_high", "PCommonsUN_medium", 
    "PCommonsDN_high", "PCommonsDN_medium", "PCommonsDN_Reactome", 
    "PCommonsDN_KEGG", "PCommonsDN_HumanCyc", "PCommonsDN_PID", 
    "PCommonsDN_PANTHER", "PCommonsDN_ReconX", "PCommonsDN_TRANSFAC", 
    "PCommonsDN_PhosphoSite", "PCommonsDN_CTD"), weighted = FALSE, 
  network.customised = NULL, normalise = c("laplacian", "row", "column", 
    "column"), 
  c) save to the file called 'SNPs_priority.txt'
  write.table(pNode$priority, file="SNPs_priority.txt", sep="\t", 
    row.names=FALSE)

  # d) manhattan plot
  mp <- xPierManhattan(pNode, highlight.top=10)
  pdf(file="Gene_manhattan.pdf", height=6, width=12, compress=TRUE)
  print(mp)
  #dev.off()

  ## End(Not run)
```

**Function to resolve relative importance of distance weight and eQTL weight priorising consensus gene ranks given a list of seed SNPs together with the significance level (e.g. GWAS reported p-values)**

```r
xPierSNPsConsensus(data, include.LD = NA, LD.customised = NULL, 
  LD.r2 = 0.8, significance.threshold = 5e-05, distance.max = 2e+05, 
  decay.kernel = c("rapid", "slow", "linear"), decay.exponent = 2, 
  GR.SNP = c("dbSNP_GWAS", "dbSNP_Common"), GR.Gene = c("UCSC_knownGene", 
    "UCSC_knownCanonical"), include.eQTL = c(NA, "JKscience_TS2A", 
    "JKscience_TS2B", "JKscience_TS3A", "JKng_bcell", "JKng_mono", 
    "JKnc_neutro", 
    "JK_nk", "GTEX_V4_Adipose_Subcutaneous", "GTEX_V4_Artery_Aorta", 
    "GTEX_V4_Artery_Tibial", "GTEX_V4_Esophagus_Mucosa", 
    "GTEX_V4_Esophagus_Muscularis", "GTEX_V4_Heart_Left_Ventricle", 
    "GTEX_V4_Lung", "GTEX_V4_Muscle_Skeletal", "GTEX_V4_Nerve_Tibial", 
    "GTEX_V4_Skin_Sun_Exposed_Lower_leg", "GTEX_V4_Stomach", 
    "GTEX_V4_Thyroid", "GTEX_V4_Whole_Blood", "eQTLdb_NK", "eQTLdb_CD14", "eQTLdb_LPS2", 
    "eQTLdb_LPS24", "eQTLdb_IFN"), eQTL.customised = NULL, 
  cdf.function = c("empirical", "exponential"), scoring.scheme = c("max", 
    "sum", "sequential"), network = c("STRING_highest", "STRING_high", 
    "STRING_medium", "STRING_low", "PCommonsUN_high", "PCommonsUN_medium", 
    "PCommonsDN_high", "PCommonsDN_medium", "PCommonsDN_Reactome", 
    "PCommonsDN_KEGG", "PCommonsDN_HumanCyc", "PCommonsDN_PID", 
    "PCommonsDN_PANTHER", "PCommonsDN_ReconX", "PCommonsDN_TRANSFAC", 
    "PCommonsDN_PhosphoSite", "PCommonsDN_CTD"), weighted = FALSE, 
  network.customised = NULL, normalise = c("laplacian", "row", "column", 
    "column")
```

"none"), restart = 0.75, normalise.affinity.matrix = c("none", "quantile"), parallel = TRUE, multicores = NULL, verbose = TRUE, RData.location = "https://github.com/hfang-bristol/RDataCentre/blob/master/Portal")

Arguments

data  a named input vector containing the significance level for nodes (dbSNP). For this named vector, the element names are dbSNP ID (or in the format such as 'chr16:28525386'), the element values for the significance level (measured as p-value or fdr). Alternatively, it can be a matrix or data frame with two columns: 1st column for dbSNP, 2nd column for the significance level

include.LD  additional SNPs in LD with Lead SNPs are also included. By default, it is 'NA' to disable this option. Otherwise, LD SNPs will be included based on one or more of 26 populations and 5 super populations from 1000 Genomics Project data (phase 3). The population can be one of 5 super populations ('AFR', 'AMR', 'EAS', 'EUR', 'SAS'), or one of 26 populations ('ACB', 'ASW', 'BEB', 'CDX', 'CEU', 'CHB', 'CHS', 'CLM', 'ESN', 'FIN', 'GBR', 'GIH', 'GWD', 'IBS', 'ITU', 'JPT', 'KHV', 'LWK', 'MSL', 'MXL', 'PEL', 'PIL', 'PUR', 'STU', 'TSI', 'YRI'). Explanations for population code can be found at http://www.100genomes.org/faq/which-populations-are-part-your-study

LD.customised  a user-input matrix or data frame with 3 columns: 1st column for Lead SNPs, 2nd column for LD SNPs, and 3rd for LD r2 value. It is designed to allow the user analysing their pre-calculated LD info. This customisation (if provided) has the high priority over built-in LD SNPs

LD.r2  the LD r2 value. By default, it is 0.8, meaning that SNPs in LD (r2>=0.8) with input SNPs will be considered as LD SNPs. It can be any value from 0.8 to 1

significance.threshold  the given significance threshold. By default, it is set to NULL, meaning there is no constraint on the significance level when transforming the significance level of SNPs into scores. If given, those SNPs below this are considered significant and thus scored positively. Instead, those above this are considered insignificant and thus receive no score

distance.max  the maximum distance between genes and SNPs. Only those genes no far way from this distance will be considered as seed genes. This parameter will influence the distance-component weights calculated for nearby SNPs per gene

decay.kernel  a character specifying a decay kernel function. It can be one of 'slow' for slow decay, 'linear' for linear decay, and 'rapid' for rapid decay

decay.exponent  an integer specifying a decay exponent. By default, it sets to 2

GR.SNP  the genomic regions of SNPs. By default, it is 'dbSNP_GWAS', that is, SNPs from dbSNP (version 146) restricted to GWAS SNPs and their LD SNPs (hg19). It can be 'dbSNP_Common', that is, Common SNPs from dbSNP (version 146) plus GWAS SNPs and their LD SNPs (hg19). Alternatively, the user can specify the customised input. To do so, first save your RData file (containing an GR object) into your local computer, and make sure the GR object content names refer to dbSNP IDs. Then, tell "GR.SNP" with your RData file name (with or without extension), plus specify your file RData path in "RData.location"

GR.Gene  the genomic regions of genes. By default, it is 'UCSC_knownGene', that is, UCSC known genes (together with genomic locations) based on human genome assembly hg19. It can be 'UCSC_knownCanonical', that is, UCSC known
canonical genes (together with genomic locations) based on human genome assembly hg19. Alternatively, the user can specify the customised input. To do so, first save your RData file (containing an GR object) into your local computer, and make sure the GR object content names refer to Gene Symbols. Then, tell "GR.Gene" with your RData file name (with or without extension), plus specify your file RData path in "RData.location"

include.eQTL

genes modulated by eQTL (also Lead SNPs or in LD with Lead SNPs) are also included. By default, it is 'NA' to disable this option. Otherwise, those genes modulated by eQTL will be included: immune stimulation in monocytes (‘JKscience_TS1A’ and ‘JKscience_TS2B’ for cis-eQTLs or ‘JKscience_TS3A’ for trans-eQTLs) from Science 2014, 343(6175):1246949; cis- and trans-eQTLs in B cells (‘JKng_bcell’) and in monocytes (‘JKng_mono’) from Nature Genetics 2012, 44(5):502-510; cis- and trans-eQTLs in neutrophils (‘JKnc_neutro’) from Nature Communications 2015, 7(6):7545; cis-eQTLs in NK cells (‘JK_nk’) which is unpublished. Also supported are GTEx cis-eQTLs from Science 2015, 348(6235):648-60, including 13 tissues: ‘GTEx_Adipose_Subcutaneous’,’GTEx_Artery_Aorta’,’GTEx_Brain’,’GTEx_Brain_Amygdala’,’GTEx_Brain_Caudate’,’GTEx_Brain_Cerebellum’,’GTEx_Brain_Hippocampus’,’GTEx_Brain_Hypothalamus’,’GTEx_Brain_Non_Cerebellum’,’GTEx_Brain_Thalamus’,’GTEx_Lung’,’GTEx_Skull’,’GTEx_Skin_Sun_Exposed_Lower_leg’.

eQTL.customised

a user-input matrix or data frame with 3 columns: 1st column for SNPs/eQTLs, 2nd column for Genes, and 3rd for eQTL mapping significance level (p-values or FDR). It is designed to allow the user analysing their eQTL data. This customisation (if provided) has the high priority over built-in eQTL data.

cdf.function

a character specifying a Cumulative Distribution Function (cdf). It can be one of ‘exponential’ based on exponential cdf, ‘empirical’ for empirical cdf

scoring.scheme

the method used to calculate seed gene scores under a set of SNPs. It can be one of "sum" for adding up, "max" for the maximum, and "sequential" for the sequential weighting. The sequential weighting is done via: \[ \sum_{i=1}^{R_i} \frac{R_i}{t} \], where \( R_i \) is the \( i^{th} \) rank (in a descreasing order)

network

the built-in network. Currently two sources of network information are supported: the STRING database (version 10) and the Pathways Commons database (version 7). STRING is a meta-integration of undirect interactions from the functional aspect, while Pathways Commons mainly contains both undirect and direct interactions from the physical/pathway aspect. Both have scores to control the confidence of interactions. Therefore, the user can choose the different quality of the interactions. In STRING, "STRING_highest" indicates interactions with highest confidence (confidence scores>=900), "STRING_high" for interactions with high confidence (confidence scores>=700), "STRING_medium" for interactions with medium confidence (confidence scores>=400), and "STRING_low" for interactions with low confidence (confidence scores>=150). For undirect/physical interactions from Pathways Commons, "PCommonsUN_high" indicates undirect interactions with high confidence (supported with the PubMed references plus at least 2 different sources), "PCommonsUN_medium" for undirect interactions with medium confidence (supported with the PubMed references). For direct (pathway-merged) interactions from Pathways Commons, "PCommonsDN_high" indicates direct interactions with high confidence (supported with the PubMed references plus at least 2 different sources), and "PCommonsDN_medium" for direct interactions with medium confidence (supported with the PubMed references). In addition to pooled version of pathways from all data sources, the user can also choose the pathway-merged network from individual sources, that is, "PCommonsDN_Reactome" for those from Reactome, "PCommonsDN_KEGG" for those from KEGG, "PCommonsDN_HumanCyc" for those from HumanCyc, "PCommonsDN_PID" for those from PID, "PCommonsDN_PANTHER" for those from PANTHER, "PCommonsDN_ReconX"
for those from ReconX, "PCommonsDN_TRANSFAC" for those from TRANSFAC, "PCommonsDN_PhosphoSite" for those from PhosphoSite, and "PCommonsDN_CTD" for those from CTD

weighted logical to indicate whether edge weights should be considered. By default, it sets to false. If true, it only works for the network from the STRING database

network.customised
an object of class "igraph". By default, it is NULL. It is designed to allow the user analysing their customised network data that are not listed in the above argument 'network'. This customisation (if provided) has the high priority over built-in network. If the user provides the "igraph" object with the "weight" edge attribute, RWR will assume to walk on the weighted network

normalise
the way to normalise the adjacency matrix of the input graph. It can be 'laplacian' for laplacian normalisation, 'row' for row-wise normalisation, 'column' for column-wise normalisation, or 'none'

restart
the restart probability used for Random Walk with Restart (RWR). The restart probability takes the value from 0 to 1, controlling the range from the starting nodes/seeds that the walker will explore. The higher the value, the more likely the walker is to visit the nodes centered on the starting nodes. At the extreme when the restart probability is zero, the walker moves freely to the neighbors at each step without restarting from seeds, i.e., following a random walk (RW)

normalise.affinity.matrix
the way to normalise the output affinity matrix. It can be 'none' for no normalisation, 'quantile' for quantile normalisation to ensure that columns (if multiple) of the output affinity matrix have the same quantiles

parallel logical to indicate whether parallel computation with multicores is used. By default, it sets to true, but not necessarily does so. Partly because parallel backends available will be system-specific (now only Linux or Mac OS). Also, it will depend on whether these two packages "foreach" and "doMC" have been installed. It can be installed via: source("http://bioconductor.org/biocLite.R"); biocLite(c("foreach","doMC")). If not yet installed, this option will be disabled

multicores
an integer to specify how many cores will be registered as the multicore parallel backend to the 'foreach' package. If NULL, it will use a half of cores available in a user’s computer. This option only works when parallel computation is enabled

verbose
logical to indicate whether the messages will be displayed in the screen. By default, it sets to true for display

RData.location
the characters to tell the location of built-in RData files. See xRDataLoader for details

Value
an object of class "pNode", a list with following components:

• priority: a matrix of nNode X 4 containing node priority information, where nNode is the number of nodes in the input graph, and the 4 columns are "name" (node names), "seed" (1 for seeds, 0 for non-seeds), "weight" (weight/score values for seed genes), 'priority' (the priority scores that are rescaled to the range [0,1]), "rank" (ranks of the priority scores), and two additional columns: 'driver' telling who drives the prioritisation ('nGenes','eGenes' or 'both'), and 'consensus_rank'

• g: an input "igraph" object
• SNP: a data frame of nSNP X 3 containing input SNPs and/or LD SNPs info, where nSNP is the number of input SNPs and/or LD SNPs, and the 3 columns are “SNP” (dbSNP), “Score” (the SNP score), “Pval” (the SNP p-value)
• Gene2SNP: a matrix of Genes X SNPs, each non-zero cell telling an SNP’s genetic influential score on a seed gene
• nGenes: the relative weight for nearby genes
• consensus: a matrix containing details on rank results by decreasing the relative importance of nGenes. In addition to rank matrix, it has columns ‘rank_median’ for median rank excluding two extremes ‘n_1’ (nGenes only) and ‘n_0’ (eGenes only), ‘rank_MAD’ for median absolute deviation, ‘driver’ telling who drives the prioritisation (‘nGenes’, ‘eGenes’ or ‘both’), ‘consensus_rank’ for the rank of the median rank list
• call: the call that produced this result

Note
none

See Also
xPierSNPs

Examples

```r
## Not run:
# Load the library
library(Pi)

## End(Not run)

# a) provide the SNPs with the significance info
# get lead SNPs reported in AS GWAS and their significance info (p-values)
data.file <- "http://galahad.well.ox.ac.uk/bigdata/AS.txt"
AS <- read.delim(data.file, header=TRUE, stringsAsFactors=FALSE)
ImmunoBase <- xRDataLoader(RData.customised="\'Var ImmunoBase\'"
) gr <- ImmunoBase$AS$variants
AS <- as.data.frame(GenomicRanges::mcols(gr[, c('Variant', 'Pvalue')])

## Not run:
# b) perform priority analysis
pNode <- xPierSNPsConsensus(data=AS, include.LD="EUR",
 include.eQTL=c("JKscience_TS2A","JKscience_TS3A"),
network="PCommonsUN_medium", restart=0.7)

# c) save to the file called 'SNPs_priority.consensus.txt'
write.table(pNode$priority, file="SNPs_priority.consensus.txt",
sep="\t", row.names=FALSE)

# d) manhattan plot
mp <- xPierManhattan(pNode, highlight.top=10)
#pdf(file="Gene_manhattan.pdf", height=6, width=12, compress=TRUE)
print(mp)
#dev.off()

## End(Not run)
```
xPierSubnet Function to identify a gene network from top prioritised genes

Description

xPierSubnet is supposed to identify maximum-scoring gene subnetwork from a graph with the
node information on priority scores, both are part of an object of class "pNode". It returns an object
of class "igraph".

Usage

xPierSubnet(pNode, priority.quantite = 0.1, network = c(NULL,
"STRING_highest", "STRING_high", "STRING_medium", "STRING_low",
"PCommonsUN_high", "PCommonsUN_medium", "PCommonsDN_high",
"PCommonsDN_medium", "PCommonsDN_Reactome", "PCommonsDN_KEGG",
"PCommonsDN_HumanCyc", "PCommonsDN_PID", "PCommonsDN_PANThER",
"PCommonsDN_ReconX", "PCommonsDN_TRANSFAC", "PCommonsDN_PhosphoSite",
"PCommonsDN_CTD"), network.customised = NULL, subnet.significance = 0.01,
subnet.size = NULL, verbose = TRUE,
RData.location = "https://github.com/hfang-bristol/RDataCentre/blob/master/Portal")

Arguments

pNode an object of class "pNode"

priority.quantite the quantite of the top priority genes. By default, 10 analysis. If NULL or NA,
all prioritised genes will be used

network the built-in network. If NULL, the network used for prioritisation will be used,
which is part of the object of class "pNode". Otherwise, choose the other net-
work of interest. Currently two sources of network information are supported:
the STRING database (version 10) and the Pathways Commons database (ver-
sion 7). STRING is a meta-integration of undirect interactions from the func-
tional aspect, while Pathways Commons mainly contains both undirect and di-
rect interactions from the physical/pathway aspect. Both have scores to control
the confidence of interactions. Therefore, the user can choose the different qual-
ity of the interactions. In STRING, "STRING_highest" indicates interactions
with highest confidence (confidence scores>=900), "STRING_high" for interac-
tions with high confidence (confidence scores>=700), "STRING_medium" for
interactions with medium confidence (confidence scores>=400), and "STRING_low"
for interactions with low confidence (confidence scores>=150). For undirect/physical
interactions from Pathways Commons, "PCommonsUN_high" indicates undi-
rect interactions with high confidence (supported with the PubMed references
plus at least 2 different sources), "PCommonsUN_medium" for undirect inter-
actions with medium confidence (supported with the PubMed references).
For direct (pathway-merged) interactions from Pathways Commons, "PCom-
monsDN_high" indicates direct interactions with high confidence (supported with
the PubMed references plus at least 2 different sources), and "PCommons-
unsUN_medium" for direct interactions with medium confidence (supported with
the PubMed references). In addition to pooled version of pathways from all data sources, the user can also choose the pathway-merged network from individual sources, that is, "PCommonsDN_Reactome" for those from Reactome, "PCommonsDN_KEGG" for those from KEGG, "PCommonsDN_HumanCyc" for those from HumanCyc, "PCommonsDN_PID" for those from PID, "PCommonsDN_PANTHER" for those from PANTHER, "PCommonsDN_ReconX" for those from ReconX, "PCommonsDN_TRANSFAC" for those from TRANSFAC, "PCommonsDN_PhosphoSite" for those from PhosphoSite, and "PCommonsDN_CTD" for those from CTD.

network.customised

an object of class "igraph". By default, it is NULL. It is designed to allow the user analysing their customised network data that are not listed in the above argument `network`. This customisation (if provided) has the high priority over built-in network.

subnet.significance

the given significance threshold. By default, it is set to NULL, meaning there is no constraint on nodes/genes. If given, those nodes/genes with p-values below this are considered significant and thus scored positively. Instead, those p-values above this given significance threshold are considered insignificant and thus scored negatively.

subnet.size

the desired number of nodes constrained to the resulting subnet. It is not null, a wide range of significance thresholds will be scanned to find the optimal significance threshold leading to the desired number of nodes in the resulting subnet. Notably, the given significance threshold will be overwritten by this option.

verbose

logical to indicate whether the messages will be displayed in the screen. By default, it sets to true for display.

RData.location

the characters to tell the location of built-in RData files. See `xRDataLoader` for details.

Value

a subgraph with a maximum score, an object of class "igraph". It has ndoe attributes: significance, score, priority (part of the "pNode" object).

Note

The priority score will be first scaled to the range x=[0 100] and then is converted to pvalue-like significant level: 10^{(-x)}. Next, `xSubneterGenes` is used to identify a maximum-scoring gene subnetwork that contains as many highly prioritised genes as possible but a few lowly prioritised genes as linkers. An iterative procedure of scanning different priority thresholds is also used to identify the network with a desired number of nodes/genes. Notably, the preferential use of the same network as used in gene-level prioritisation is due to the fact that gene-level affinity/priority scores are smoothly distributed over the network after being walked. In other words, the chance of identifying such a gene network enriched with top prioritised genes is much higher.

See Also

`xSubneterGenes`

Examples

```r
## Not run:
```
# Load the library
library(Pi)

## End(Not run)

# a) provide the SNPs with the significance info
## get lead SNPs reported in AS GWAS and their significance info (p-values)
data.file <- "http://galahad.well.ox.ac.uk/bigdata/AS.txt"
AS <- read.delim(data.file, header=TRUE, stringsAsFactors=FALSE)
ImmunoBase <- xRDataLoader(RData.customised='ImmunoBase')
gr <- ImmunoBase$AS$variants
AS <- as.data.frame(GenomicRanges::mcols(gr)[, c('Variant','Pvalue')])

## Not run:
# b) perform priority analysis
pNode <- xPierSNPs(data=AS, network="PCommonsUN_medium", restart=0.7)

# c) perform network analysis
# find maximum-scoring subnet with the desired node number=50
subnet <- xPierSubnet(pNode, priority.quantite=0.1, subnet.size=50)

# d) save subnet results to the files called 'subnet_edges.txt' and 'subnet_nodes.txt'
output <- igraph::get.data.frame(subnet, what="edges")
utils::write.table(output, file="subnet_edges.txt", sep="\t", row.names=FALSE)
output <- igraph::get.data.frame(subnet, what="vertices")
utils::write.table(output, file="subnet_nodes.txt", sep="\t", row.names=FALSE)

# e) visualise the identified subnet
## do visualisation with nodes colored according to the priority
xVisNet(g=subnet, pattern=V(subnet)$priority, vertex.shape="sphere")
## do visualisation with nodes colored according to pvalue-like significance
xVisNet(g=subnet, pattern=-log10(as.numeric(V(subnet)$significance)),
vertex.shape="sphere", colormap="wyr")

# f) visualise the identified subnet as a circos plot
library(RCircos)
xCircos(g=subnet, entity="Gene")

## End(Not run)

---

**xPredictPR**

*Function to assess the prediction performance via Precision-Recall (PR) analysis*

**Description**

xPredictPR is supposed to assess the prediction performance via Precision-Recall (PR) analysis. It requires two inputs: 1) Gold Standard Positive (GSP) containing targets; 2) prediction containing predicted targets and predictive scores.
Usage

xPredictPR(GSP, prediction, num.threshold = 20, bin = c("quantile", "uniform"), recall.prediction = FALSE, GSN = NULL, plot = TRUE, smooth = FALSE, verbose = TRUE)

Arguments

GSP a vector containing Gold Standard Positive (GSP)
prediction a data frame containing predictions along with predictive scores. It has two columns: 1st column for target, 2nd column for predictive scores (the higher the better)
num.threshold an integer to specify how many PR points (as a function of the score threshold) will be calculated
bin how to bin the scores. It can be "uniform" for binning scores with equal interval (ie with uniform distribution), and "quantile" for binning scores with equal frequency (ie with equal number)
recall.prediction logical to indicate whether the calculation of recall is based on predictable GSP. By default, it sets to FALSE
GSN a vector containing Gold Standard Negative (GSN). It is optional. By default (NULL), GSN is not provided
plot logical to indicate whether to return an object of class "ggplot" for plotting PR curve. By default, it sets to FALSE. If TRUE, it will return a ggplot object after being appended with 'PR' and 'Fmax'
smooth logical to indicate whether to smooth the curve by making sure a non-increasing order for precision. By default, it sets to FALSE
verbose logical to indicate whether the messages will be displayed in the screen. By default, it sets to TRUE for display

Value

If plot is FALSE (by default), a data frame containing three columns: 1st column ‘Threshold’ for the score threshold, 2nd column ‘Precision’ for precision, 3rd ‘Recall’ for recall. If plot is TRUE, it will return a ggplot object after being appended with ‘PR’ (a data frame containing three columns: 1st column ‘Threshold’ for the score threshold, 2nd column ‘Precision’ for precision, 3rd ‘Recall’ for recall), and ‘Fmax’ for maximum F-measure.

Note

F-measure: the maximum of a harmonic mean between precision and recall along PR curve

Examples

## Not run:
PR <- xPredictPR(GSP, prediction)
## End(Not run)
Function to implement Random Walk with Restart (RWR) on the input graph

Description

XRWR is supposed to implement Random Walk with Restart (RWR) on the input graph. If the seeds (i.e. a set of starting nodes) are given, it intends to calculate the affinity score of all nodes in the graph to the seeds. If the seeds are not given, it will pre-compute affinity matrix for nodes in the input graph with respect to each starting node (as a seed) by looping over every node in the graph. Parallel computing is also supported for Linux or Mac operating systems.

Usage

XRWR(g, normalise = c("laplacian", "row", "column", "none"),
        setSeeds = NULL, restart = 0.75, normalise.affinity.matrix = c("none",
        "quantile"), parallel = TRUE, multicores = NULL, verbose = TRUE)

Arguments

g

an object of class "igraph" or "graphNEL"

normalise

the way to normalise the adjacency matrix of the input graph. It can be 'laplacian' for laplacian normalisation, 'row' for row-wise normalisation, 'column' for column-wise normalisation, or 'none'

setSeeds

an input matrix used to define sets of starting seeds. One column corresponds to one set of seeds that a walker starts with. The input matrix must have row names, coming from node names of input graph, i.e. V(g)$name, since there is a mapping operation. The non-zero entries mean that the corresponding rows (i.e. the gene/row names) are used as the seeds, and non-zero values can be viewed as how to weight the relative importance of seeds. By default, this option sets to "NULL", suggesting each node in the graph will be used as a set of the seed to pre-compute affinity matrix for the input graph. This default does not scale for large input graphs since it will loop over every node in the graph; however, the pre-computed affinity matrix can be extensively reused for obtaining affinity scores between any combinations of nodes/seeds, allows for some flexibility in the downstream use, in particular when sampling a large number of random node combinations for statistical testing

restart

the restart probability used for RWR. The restart probability takes the value from 0 to 1, controlling the range from the starting nodes/seeds that the walker will explore. The higher the value, the more likely the walker is to visit the nodes centered on the starting nodes. At the extreme when the restart probability is zero, the walker moves freely to the neighbors at each step without restarting from seeds, i.e., following a random walk (RW)

normalise.affinity.matrix

the way to normalise the output affinity matrix. It can be 'none' for no normalisation, 'quantile' for quantile normalisation to ensure that columns (if multiple) of the output affinity matrix have the same quantiles

parallel

logical to indicate whether parallel computation with multicores is used. By default, it sets to true, but not necessarily does so. Partly because parallel backends
available will be system-specific (now only Linux or Mac OS). Also, it will de-
pend on whether these two packages "foreach" and "doMC" have been installed.
It can be installed via: source("http://bioconductor.org/biocLite.R");
biocLite(c("foreach","doMC")). If not yet installed, this option will be dis-
abled

**multicores**

an integer to specify how many cores will be registered as the multicore parallel
backend to the 'foreach' package. If NULL, it will use a half of cores available in
a user's computer. This option only works when parallel computation is enabled

**verbose**

logical to indicate whether the messages will be displayed in the screen. By
default, it sets to true for display

**Value**

It returns a sparse matrix, called 'PTmatrix':

- When the seeds are NOT given: a pre-computed affinity matrix with the dimension of n
  X n, where n is the number of nodes in the input graph. Columns stand for starting nodes
  walking from, and rows for ending nodes walking to. Therefore, a column for a starting node
  represents a steady-state affinity vector that the starting node will visit all the ending nodes in
  the graph

- When the seeds are given: an affinity matrix with the dimension of n X nset, where n is
  the number of nodes in the input graph, and nset for the number of the sets of seeds (i.e.
  the number of columns in setSeeds). Each column stands for the steady probability vector,
  storing the affinity score of all nodes in the graph to the starting nodes/seeds. This steady
  probability vector can be viewed as the "influential impact" over the graph imposed by the
  starting nodes/seeds.

**Note**

The input graph will treat as an unweighted graph if there is no 'weight' edge attribute associated
with

**See Also**

- xPier

**Examples**

```r
# 1) generate a random graph according to the ER model
g <- erdos.renyi.game(100, 1/100)

## Not run:
# 2) produce the induced subgraph only based on the nodes in query
subg <- dNetInduce(g, V(g), knn=0)
V(subg)$name <- 1:vcount(subg)

# 3) obtain the pre-computed affinity matrix
PTmatrix <- xRWR(g=subg, normalise="laplacian", restart=0.75,
parallel=FALSE)
# visualise affinity matrix
visHeatmapAdv(PTmatrix, Rowv=FALSE, Colv=FALSE, colormap="wyr",
KeyValueName="Affinity")

# 4) obtain affinity matrix given sets of seeds
```
# define sets of seeds
# each seed with equal weight (i.e. all non-zero entries are '1')
aSeeds <- c(1,0,1,0,1)
bSeeds <- c(0,0,1,0,1)
setSeeds <- data.frame(aSeeds,bSeeds)ownames(setSeeds) <- 1:5
# calculate affinity matrix
PTmatrix <- xRWR(g=subg, normalise="laplacian", setSeeds=setSeeds,
restart=0.75, parallel=FALSE)

## End(Not run)

xSNP2eGenes  

Function to define eQTL genes given a list of SNPs or a customised eQTL mapping data

Description

xSNP2eGenes is supposed to define eQTL genes given a list of SNPs or a customised eQTL mapping data. The eQTL weight is calculated as Cumulative Distribution Function of negative log-transformed eQTL-reported significance level.

Usage


Arguments

data  
a input vector containing SNPs. SNPs should be provided as dbSNP ID (ie starting with rs). Alternatively, they can be in the format of 'chrN:xxx', where N is either 1-22 or X, xxx is number; for example, 'chr16:28525386'

include.eQTL  
genes modulated by eQTL (also Lead SNPs or in LD with Lead SNPs) are also included. By default, it is 'NA' to disable this option. Otherwise, those genes modulated by eQTL will be included: immune stimulation in monocytes ('JKScience_TS1A' and 'JKScience_TS2B' for cis-eQTLs or 'JKScience_TS3A' for trans-eQTLs) from Science 2014, 343(6175):1246949; cis- and trans-eQTLs
In B cells ('JKng_bcell') and in monocytes ('JKng_mono') from Nature Genetics 2012, 44(5):502-510; cis- and trans-eQTLs in neutrophils ('JKnc_neutro') from Nature Communications 2015, 7(6):7545; cis-eQTLs in NK cells ('JK_nk') which is unpublished. Also supported are GTEx cis-eQTLs from Science 2015, 348(6235):648-60, including 13 tissues: "GTEx_V4_Adpose_Subcutaneous","GTEx_V4_Artery_Aorta","GTEx_V4_Brain_Base_of_Brain","GTEx_V4_Brain_Cerebellum","GTEx_V4_Brain_Frontal","GTEx_V4_Brain_Hippocampus","GTEx_V4_Brain_Medulla_Ornalis","GTEx_V4_Brain_Pineal","GTEx_V4_Brain_Cerebellar_Plate","GTEx_V4_Brain_Cerebellum","GTEx_V4_Brain_Cerebral_Cortex","GTEx_V4_Brain_Hypothalamus","GTEx_V4_Brain_Thyroid","GTEx_V4_Fetal_Brain","GTEx_V4_Fetal_Cerebellar_Plate","GTEx_V4_Fetal_Cerebellum","GTEx_V4_Fetal_Cerebral_Cortex","GTEx_V4_Mouse_Brain","GTEx_V4_Skin_Sun_Exposed_Lower_leg","GTEx_V4_Stomach","GTEx_V4_Thyroid","GTEx_V4_Human_Brain","GTEx_V4_Whole_Blood".

eQTL.customised

A user-input matrix or data frame with 3 columns: 1st column for SNPs/eQTLs, 2nd column for Genes, and 3rd for eQTL mapping significance level (p-values or FDR). It is designed to allow the user analysing their eQTL data. This customisation (if provided) has the high priority over built-in eQTL data.

cdf.function

A character specifying a Cumulative Distribution Function (cdf). It can be one of 'exponential' based on exponential cdf, 'empirical' for empirical cdf.

plot

Logical to indicate whether the histogram plot (plus density or CDF plot) should be drawn. By default, it sets to false for no plotting.

verbose

Logical to indicate whether the messages will be displayed in the screen. By default, it sets to true for display.

RData.location

The characters to tell the location of built-in RData files. See xRDataLoader for details.

Value

A data frame with following columns:

- Gene: eQTL-containing genes
- SNP: eQTLs
- Sig: the eQTL mapping significant level (the best/minimum)
- Weight: the eQTL weight

Note

None

See Also

xRDataLoader

Examples

```R
## Not run:
# Load the library
library(Pi)

## End(Not run)

# a) provide the SNPs with the significance info
# get lead SNPs reported in AS GWAS and their significance info (p-values)
data.file <- "http://galahad.well.ox.ac.uk/bigdata/AS.txt"
AS <- read.delim(data.file, header=TRUE, stringsAsFactors=FALSE)
ImmunoBase <- xRDataLoader(RData.customised="/quotesingle.Var
ImmunoBase/quotesingle.Var
ImmunoBase")
gr <- ImmunoBase$AS$variants
AS <- as.data.frame(GenomicRanges::mcols(gr)[, c('Variant', 'Pvalue')])

df_eGenes <- xSNP2eGenes(data=AS[,1], include.eQTL="JKscience_TS2A")
```

# b) define eQTL genes

df_eGenes <- xSNP2eGenes(data=AS[,1], include.eQTL="JKscience_TS2A")
xSNPeqtl  

Function to extract eQTL-gene pairs given a list of SNPs or a customised eQTL mapping data

Description

xSNPeqtl is supposed to extract eQTL-gene pairs given a list of SNPs or a customised eQTL mapping data.

Usage

xSNPeqtl(data = NULL, include.eQTL = c(NA, "JKscience_TS2A", "JKscience_TS2B", "JKscience_TS3A", "JKng_bcell", "JKng_mono", "JKnk_neutro", "JK_neutro", "GTEx_V4_Adipose_Subcutaneous", "GTEx_V4_Artery_Aorta", "GTEx_V4_Artery_Tibial", "GTEx_V4_Artery_Muscularis", "GTEx_V4_Heart_Left_Ventricle", "GTEx_V4_Lung", "GTEx_V4_Muscle_Skeletals", "GTEx_V4_Nerve_Tibial", "GTEx_V4_Skin_Sun_Exposed_Lower_leg", "GTEx_V4_Stomach", "GTEx_V4_Thyroid", "GTEx_V4_Whole_Blood", "eQTLdb_NK", "eQTLdb_CD14", "eQTLdb_LPS2", "eQTLdb_LPS24", "eQTLdb_IFN"), eQTL.customised = NULL, verbose = TRUE, RData.location = "https://github.com/hfang-bristol/RDataCentre/blob/master/Portal")

Arguments

data
Null or a input vector containing SNPs. If NULL, all SNPs will be considered. If a input vector containing SNPs, SNPs should be provided as dbSNP ID (ie starting with rs). Alternatively, they can be in the format of 'chrN:xxx', where N is either 1-22 or X, xxx is number; for example, 'chr16:28525386'

include.eQTL
genes modulated by eQTL (also Lead SNPs or in LD with Lead SNPs) are also included. By default, it is 'NA' to disable this option. Otherwise, those genes modulated by eQTL will be included: immune stimulation in monocytes ('JKscience_TS1A' and 'JKscience_TS2B' for cis-eQTLs or 'JKscience_TS3A' for trans-eQTLs) from Science 2014, 343(6175):1246949; cis- and trans-eQTLs in neutrophils ('JKnk_neutro') from Nature Genetics 2012, 44(5):502-510; cis- and trans-eQTLs in neutrophils ('JKnk_neutro') from Nature Communications 2015, 7(6):7545; cis-eQTLs in NK cells ('JKnk_kn') which is unpublished. Also supported are GTEx cis-eQTLs from Science 2015, 348(6235):648-60, including 13 tissues: "GTEx_V4_Adipose_Subcutaneous","GTEx_V4_Artery_Aorta",...

eQTL.customised
a user-input matrix or data frame with 3 columns: 1st column for SNPs/eQTLs, 2nd column for Genes, and 3rd for eQTL mapping significance level (p-values or FDR). It is designed to allow the user analysing their eQTL data. This customisation (if provided) has the high priority over built-in eQTL data.

verbose
logical to indicate whether the messages will be displayed in the screen. By default, it sets to true for display

RData.location
the characters to tell the location of built-in RData files. See xRDataLoader for details
Value

a data frame with following columns:

- SNP: eQTLs
- Gene: eQTL-containing genes
- Sig: the eQTL mapping significant level
- Context: the context in which eQTL data was generated

Note

None

See Also

xRDataLoader

Examples

```R
## Not run:
# Load the library
library(Pi)

## End(Not run)

# a) provide the SNPs with the significance info
# get lead SNPs reported in AS GWAS and their significance info (p-values)
data.file <- "http://galahad.well.ox.ac.uk/bigdata/AS.txt"
AS <- read.delim(data.file, header=TRUE, stringsAsFactors=FALSE)
ImmunoBase <- xRDataLoader(RData.customised="ImmunoBase")
gr <- ImmunoBase$AS$variants
AS <- as.data.frame(GenomicRanges::mcols(gr)[, c('Variant', 'Pvalue')])

# b) define eQTL genes
df_SGS <- xSNPeqtl(data=AS[,1], include.eQTL="JKscience_TS2A")
```
Index

xEEnricher, 11
xPier, 2, 6, 8, 16, 27
xPierGenes, 3, 4, 8, 16
xPierManhattan, 7
xPierPathways, 3, 6, 8, 8, 16
xPierSNPs, 3, 6, 8, 12, 21
xPierSNPsConsensus, 17
xPierSubnet, 22
xPredictPR, 24
xRDataLoader, 3, 6–8, 10, 11, 15, 20, 23, 29–31
xRWR, 3, 26
xSM2DF, 16
xSNP2eGenes, 16, 28
xSNP2nGenes, 16
xSNPeqt1, 30
xSNPscores, 16
xSparseMatrix, 16
xSubneterGenes, 23