

Package: ComplexMap (via r-universe)

October 9, 2026

Title A Toolset for the Functional Analysis and Interactive Visualization of Protein Complex Data

Version 2.0.0

Date 2026-02-18

Description Provides a comprehensive workflow for the quality control, refinement, functional enrichment, and network-based analysis of protein complex datasets. ComplexMap allows for the quantitative evaluation of predicted complexes against a reference and provides an interactive Cytoscape.js-powered Shiny explorer to navigate the functional landscape of a protein complex map.

URL <https://zqzneptune.github.io/ComplexMap>,
<https://github.com/zqzneptune/ComplexMap>

BugReports <https://github.com/zqzneptune/ComplexMap/issues>

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (≥ 4.5)

Imports AnnotationDbi, clue, dplyr, future, future.apply, GO.db, igraph, jsonlite, magrittr, Matrix, msigdb, RColorBrewer, reactome.db, rlang, shiny, stringr, tibble, tidyr

Suggests colorspace, knitr, org.Hs.eg.db, philentropy, progressr, rmarkdown, testthat ($\geq 3.0.0$)

BiocViews Software, Proteomics, NetworkAnalysis, Visualization, Enrichment, GeneSetEnrichment

LazyData true

VignetteBuilder knitr

Config/testthat/edition 3

Config/pak/sysreqs

cmake libgplk-dev make libicu-dev libpng-dev libuv1-dev libxml2-dev libssl-dev zlib1g-dev

Repository <https://zqzneptune.r-universe.dev>
Date/Publication 2026-02-24 02:28:27 UTC
RemoteUrl <https://github.com/zqzneptune/ComplexMap>
RemoteRef HEAD
RemoteSha 83415bf695207de6f524601d6689030a23f7a9ee

Contents

benchmarkParameters	2
buildComplexNetwork	4
computeMapTopology	5
createComplexMap	6
demoComplexes	8
evaluateComplexes	9
explore	10
exportNetwork	11
generateNodeAttributes	12
getEdgeTable	12
getExampleGmt	13
getGmtFromFile	14
getGoGmt	14
getMsigdbGmt	15
getNodeTable	15
getReactomeGmt	16
print.ComplexMap	16
pruneNetwork	17
qcComplexList	18
queryMap	19
referenceComplexes	19
refineComplexList	20
runComplexEnrichment	21
runComplexMapApp	22
scaleLayoutForCytoscape	23
toCytoscapeJSON	24
Index	25

benchmarkParameters	<i>Benchmark Complex Refinement Parameters</i>
----------------------------	--

Description

Evaluates how different merging thresholds affect the quality AND diversity of the resulting complex map.

Usage

```
benchmarkParameters(
  complexList,
  referenceComplexes,
  threshold_range = seq(0.5, 1, by = 0.05),
  similarityMethod = "jaccard",
  ...
)
```

Arguments

complexList A list of predicted protein complexes to be refined.

referenceComplexes A list of reference (gold standard) complexes.

threshold_range Numeric vector of thresholds to test. Defaults to 'seq(0.5, 1.0, by = 0.05)'.
similarityMethod The metric used for merging. Defaults to `"jaccard"`.

... Additional arguments passed to 'refineComplexList' (e.g., minSize).

Details

****Systems Biology Rationale:**** When optimizing the 'mergeThreshold', there is a trade-off between removing redundancy (increasing PPV) and losing specific variants (decreasing Sensitivity and Diversity).

This function returns the standard metrics (PPV, Sn, Acc, MMR) but also adds ****NumComplexes****. - ****Collapse Warning:**** If 'NumComplexes' drops precipitously between two thresholds, you have likely crossed the point where distinct biological variants are being merged into generic blobs. - ****Recommendation:**** Choose the threshold that maximizes F1/MMR while maintaining a 'NumComplexes' count close to your expected biological complexity.

Value

A 'tibble' with columns: - 'mergeThreshold': The threshold tested. - 'NumComplexes': Number of complexes remaining (Diversity metric). - 'PPV': Positive Predictive Value (Precision). - 'Sn': Sensitivity (Recall). - 'F1': Harmonic mean of PPV and Sn. - 'Acc': Accuracy (Geometric mean of PPV and Sn). - 'MMR': Maximum Matching Ratio.

Author(s)

Qingzhou Zhang <zqzneptune@hotmail.com>

buildComplexNetwork *Build a Complex-Complex Interaction Network*

Description

Constructs a network of complexes where edges represent similarity.

Usage

```
buildComplexNetwork(
  complexes,
  enrichments,
  mode = "combined",
  similarityMethod = "jaccard",
  alpha = 0.75,
  nCores = NULL,
  chunkSize = 1000,
  verbose = TRUE
)
```

Arguments

complexes	A named list of protein complexes.
enrichments	A named list of enrichment results (from 'runComplexEnrichment').
mode	Edge weight mode: "compositional", "functional", or "combined". Defaults to "combined".
similarityMethod	The metric for similarity. Defaults to "jaccard" . **Warning:** Using "overlap" is discouraged as it tends to collapse diverse hierarchies into single blobs.
alpha	Numeric (0-1). Weight given to Compositional Similarity in "combined" mode. Defaults to 0.75 (favoring physical structure).
nCores	Number of cores for parallel processing.
chunkSize	Size of processing chunks.
verbose	Logical.

Details

****Systems Biology Rationale:**** To generate a diverse and physically meaningful landscape, this function defaults to prioritizing ****Compositional Similarity**** (shared proteins) over functional similarity.

- ****Composition (Protein Identity)**** is treated as the "Physical Truth". - ****Function (Enrichment)**** is treated as an attribute.

The 'alpha' parameter controls this balance. The new default (0.75) gives 75 (like "Cell Cycle") from artificially pulling distinct physical complexes into a single cluster, while still allowing functionally related complexes to drift closer than unrelated ones.

Value

A ‘tibble’ of network edges with ‘weight’ representing the calculated similarity.

Author(s)

Qingzhou Zhang <zqzneptune@hotmail.com>

`computeMapTopology` *Compute Network Topology and Layout Coordinates*

Description

Calculates the 2D layout coordinates and key centrality metrics for a complex network. This function serves as the final step in preparing network data for visualization.

Usage

```
computeMapTopology(nodeAttributes, network, seed = 123, verbose = TRUE)
```

Arguments

nodeAttributes	A ‘tibble’ or ‘data.frame’ containing attributes for each node (complex). Must contain a column with node identifiers that matches the source/target columns in the ‘network’ data.
network	A ‘tibble’ or ‘data.frame’ representing the network edges. Must contain columns for source, target, and edge ‘weight’.
seed	Integer seed for reproducible layout. Defaults to ‘123’.
verbose	A logical value indicating whether to print progress messages. Defaults to ‘TRUE’.

Details

This function takes a node attribute table and an edge list (network) and performs the following steps:

1. Constructs an ‘igraph’ graph object from the provided data.
2. Computes a force-directed layout using the Fruchterman-Reingold algorithm via ‘igraph::layout_with_fr’. Edge weights are used to influence the layout, pulling strongly connected nodes closer together.
3. Calculates node centrality metrics:
 - **Betweenness Centrality:** Measures how often a node lies on the shortest path between other nodes (normalized).
 - **Degree Centrality:** The number of edges connected to a node.
4. Merges the layout coordinates and centrality scores back into the original node attribute table.

Value

A ‘tibble’ containing all original columns from ‘nodeAttributes’ plus four new columns: ‘x’, ‘y’ (layout coordinates), ‘betweenness’, and ‘degree’. The table is arranged in descending order of betweenness and degree.

Author(s)

Qingzhou Zhang <zqzneptune@hotmail.com>

See Also

‘generateNodeAttributes()’, ‘buildComplexNetwork()’

Examples

```
# --- Sample Data ---
# 1. Node attributes
nodes <- tibble::tibble(
  complexId = c("Cpx1", "Cpx2", "Cpx3"),
  proteinCount = c(10, 8, 12)
)

# 2. Network edges
net <- tibble::tibble(
  source = c("Cpx1", "Cpx2"),
  target = c("Cpx2", "Cpx3"),
  weight = c(0.8, 0.6)
)

# --- Compute Topology ---
masterLayout <- computeMapTopology(nodes, net)
print(masterLayout)
```

createComplexMap

Create a Complete Complex Map Object

Description

A high-level wrapper function that executes the entire ‘ComplexMap’ workflow. It takes a list of protein complexes and a gene set matrix (GMT) and performs refinement, enrichment, network construction, and topology calculation.

Usage

```
createComplexMap(
  complexList,
  gmt,
  similarityMethod = "jaccard",
```

```

    alpha = 0.75,
    ifRefineCpx = FALSE,
    layout_seed = 123,
    verbose = TRUE,
    ...
)

```

Arguments

complexList	A named list where each element is a character vector of protein identifiers representing a complex.
gmt	A named list where each element is a character vector of genes.
similarityMethod	The metric used for comparing complexes and functional terms. Defaults to "jaccard" to penalize size differences and maintain diversity. Avoid "overlap" if you wish to see specific pathways distinct from generic ones.
alpha	Numeric (0-1). The weight given to physical protein composition versus functional similarity in the network layout. Defaults to 0.75 (75% physical, 25% functional).
ifRefineCpx	Logical. If TRUE , executes Step 1 to refine the input complex list (merge redundant complexes). If FALSE , Step 1 is skipped, and the raw 'complexList' is used for downstream analysis. Defaults to FALSE .
layout_seed	Integer seed for reproducible layout. Defaults to 123.
verbose	A logical value indicating whether to print progress messages.
...	Additional arguments passed to core functions: - 'minSize', 'maxSize', 'mergeThreshold' (for 'refineComplexList' if 'ifRefineCpx = TRUE') - 'pAdjustMethod', 'pValueCutoff' (for 'runComplexEnrichment') - 'geneSetDb' (for 'generateNodeAttributes' semantic clustering)

Details

This function is tuned to generate a **functionally diverse landscape**. It enforces the following logic: 1. **Refinement** (Optional) Uses Jaccard similarity to merge only highly redundant complexes, preserving biological variants (subsets/supersets). Controlled by 'ifRefineCpx'. 2. **Enrichment** Calculates 'Fold Enrichment' to prioritize specific biological functions over generic ones. 3. **Network** Builds the layout primarily based on **Physical Composition** ($\alpha = 0.75$), using functional annotations only to group related clusters, not to collapse them.

Value

A validated 'ComplexMap' S3 object.

Author(s)

Qingzhou Zhang <zqzneptune@hotmail.com>

demoComplexes*Example Protein Complex List*

Description

A dataset containing a list of 622 putative protein complexes identified in the study using an integrative global proteomic profiling approach. This dataset is intended for use in package examples and vignettes.

Usage

`demoComplexes`

Format

A named list with 622 elements. Each element is a character vector of protein identifiers representing a single complex.

Details

These complexes were derived from chromatographic separation of cultured human cell extracts into more than one thousand biochemical fractions, followed by quantitative tandem mass spectrometry analysis. Most of the 622 complexes are linked to core biological processes and include candidate disease genes and unannotated proteins.

Source

Supplementary Table 3 (first tab) from: Havugimana PC, Hart GT, Nepusz T, Yang H, Turinsky AL, Li Z, Wang PI, Boutz DR, Fong V, Phanse S, Babu M, Craig SA, Hu P, Wan C, Vlasblom J, Dar VU, Bezginov A, Clark GW, Wu GC, Wodak SJ, Tillier ER, Paccanaro A, Marcotte EM, Emili A. *A census of human soluble protein complexes.* Cell. 2012 Aug 31;150(5):1068-81. doi: 10.1016/j.cell.2012.08.011. PMID: 22939629; PMCID: PMC3477804.

See Also

[referenceComplexes](#)

<code>evaluateComplexes</code>	<i>Evaluate Predicted Protein Complexes Against a Reference Set</i>
--------------------------------	---

Description

Calculates four standard metrics for evaluating protein complex predictions: Positive Predictive Value (PPV), Sensitivity (Sn), Accuracy (Acc), and the Maximum Matching Ratio (MMR).

Usage

```
evaluateComplexes(
  predictedComplexes,
  referenceComplexes,
  nCores = NULL,
  verbose = TRUE
)
```

Arguments

<code>predictedComplexes</code>	A list of predicted protein complexes.
<code>referenceComplexes</code>	A list of reference (gold standard) complexes.
<code>nCores</code>	The number of CPU cores to use for parallel computation. Defaults to one less than the total number of detected cores.
<code>verbose</code>	A logical value indicating whether to print progress messages. Defaults to 'TRUE'.

Details

This function is optimized for speed by calculating a shared intersection matrix between predicted and reference complexes in parallel. This matrix is then used as the basis for all four metric calculations.

- **PPV, Sn, and Acc** are calculated based on the confusion matrix between predicted and reference complexes, as described in the literature (e.g., Zhang et al., 2012).

- **MMR** is calculated by first deriving an overlap score matrix, where the score for a predicted complex (P) and a reference complex (R) is $|P \cap R|^2 / (|P| * |R|)$. The [Hungarian algorithm](https://en.wikipedia.org/wiki/Hungarian_algorithm) is then used to solve the maximum weight bipartite matching problem. This approach is based on the method described by Nepusz et al. (2012).

The parallel computation uses 'future.apply::future_lapply' for cross-platform compatibility (unlike 'parallel::mclapply', which is not available on Windows).

Value

A named list containing four numeric values: ‘PPV’, ‘Sn’, ‘Acc’, and ‘MMR’. Returns ‘NA’ for all metrics if either input list is empty.

Author(s)

Qingzhou Zhang <zqzneptune@hotmail.com>

References

Nepusz, T., Yu, H. & Paccanaro, A. (2012). Detecting overlapping protein complexes in protein-protein interaction networks. *Nature Methods*, 9, 471–472. doi:10.1038/nmeth.1938

Zhang XF, Dai DQ, Ou-Yang L, Wu MY (2012). Exploring Overlapping Functional Units with Various Structure in Protein Interaction Networks. *PLOS ONE*, 7(8): e43092. doi:10.1371/journal.pone.0043092

Examples

```
# --- Sample Data ---
# Predicted complexes
pred1 <- c("A", "B", "C")
pred2 <- c("D", "E", "F")
pred3 <- c("A", "G", "H")
predicted <- list(P1 = pred1, P2 = pred2, P3 = pred3)

# Reference complexes (gold standard)
ref1 <- c("A", "B", "C", "X") # Good match for pred1
ref2 <- c("D", "E", "F")      # Perfect match for pred2
ref3 <- c("I", "J", "K")      # Unmatched complex
reference <- list(R1 = ref1, R2 = ref2, R3 = ref3)

# --- Evaluation ---
# Use 2 cores for the example
metrics <- evaluateComplexes(predicted, reference, nCores = 2)
print(metrics)
```

explore

Explore a ComplexMap Object Interactively

Description

A convenient alias for ‘runComplexMapApp()’. Launches the Cytoscape.js interactive explorer.

Usage

```
explore(cmap, ...)
```

Arguments

<code>cmap</code>	A ‘ComplexMap’ object.
<code>...</code>	Additional arguments passed to ‘runComplexMapApp’.

<code>exportNetwork</code>	<i>Export a ComplexMap Network for External Tools</i>
----------------------------	---

Description

Exports the ‘ComplexMap’ nodes and edges to compatible file formats, preserving systems biology attributes like Specificity Scores and Functional Domains.

Usage

```
exportNetwork(
  complexMapObject,
  filePrefix,
  format = "cytoscape",
  verbose = TRUE
)
```

Arguments

<code>complexMapObject</code>	A ‘ComplexMap’ object.
<code>filePrefix</code>	Character string for output filenames (e.g., "results/my_map").
<code>format</code>	Output format. One of: - "cytoscape": Writes separate node and edge TSV files for Cytoscape Desktop. - "graphml": Writes a GraphML file (via ‘igraph’). - "tsv": Writes a single node-attribute TSV file.
<code>verbose</code>	Logical.

Details

****Systems Biology Rationale:**** To verify the landscape in external tools (e.g., Cytoscape Desktop), this function ensures that the specific calculated attributes are correctly formatted:

- ****‘score’****: The Specificity Score. - ****‘primaryFunctionalDomain’****: The Specific Label.
- ****‘colorHex’****: The calculated blended color.

It performs a "Sanitization" step to convert any R-specific list columns into semi-colon separated strings and fills ‘NA’ values to ensure safe import.

Value

None (Writes files to disk).

Author(s)

Qingzhou Zhang <zqzneptune@hotmail.com>

```
generateNodeAttributes
```

Generate Node Attributes for a Complex Network

Description

Creates a detailed attribute table for each complex, emphasizing functional specificity for network visualization, and assigns professional HCL colors.

Usage

```
generateNodeAttributes(
  complexes,
  enrichments,
  geneSetDb = NULL,
  similarityMethod = "jaccard",
  verbose = TRUE
)
```

Arguments

<code>complexes</code>	A named list of protein complexes.
<code>enrichments</code>	A named list of enrichment results. Must contain 'Fold'.
<code>geneSetDb</code>	Optional named list of gene sets for semantic clustering.
<code>similarityMethod</code>	Distance method for clustering ("jaccard", etc.).
<code>verbose</code>	Logical.

Value

A 'tibble' with complex attributes.

Author(s)

Qingzhou Zhang <zqzneptune@hotmail.com>

```
getEdgeTable
```

Get the Edge Table from a ComplexMap Object

Description

Access the network edge list (weights, similarity types).

Usage

```
getEdgeTable(cm)
```

Arguments

cm A 'ComplexMap' object.

Value

A 'tibble'.

getExampleGmt	<i>Get the Path to an Example GMT File</i>
---------------	--

Description

Provides the full system path to an example gene set (GMT) file included with the ComplexMap package.

Usage

```
getExampleGmt()
```

Details

The included file is the BioCarta gene set collection from the Molecular Signatures Database (MSigDB v2025.1). This function makes it easy to access the file for use in examples and vignettes.

Value

A character string containing the full path to the example GMT file.

Examples

```
# Get the path
gmtPath <- getExampleGmt()

# You can then read the file using the path
if (file.exists(gmtPath)) {
  exampleGmt <- getGmtFromFile(gmtPath)
}
```

<code>getGmtFromFile</code>	<i>Read a GMT File from a Local Path</i>
-----------------------------	--

Description

Parses a local GMT (Gene Matrix Transposed) file into the standard named-list format.

Usage

```
getGmtFromFile(filepath, verbose = TRUE)
```

Arguments

<code>filepath</code>	The path to the local .gmt file.
<code>verbose</code>	A logical value indicating whether to print progress messages.

Value

A named list where names are gene set names and values are character vectors of genes.

<code>getGoGmt</code>	<i>Fetch Gene Ontology (GO) Gene Sets</i>
-----------------------	---

Description

Fetches GO terms and their associated genes from a Bioconductor AnnotationDb package.

Usage

```
getGoGmt(
  speciesDb,
  ontology = "BP",
  minGmtSize = 10,
  maxGmtSize = 500,
  verbose = TRUE
)
```

Arguments

<code>speciesDb</code>	An AnnotationDb object for the target species (e.g., 'org.Hs.eg.db').
<code>ontology</code>	The GO ontology to fetch. One of "BP", "MF", or "CC".
<code>minGmtSize</code>	The minimum number of genes for a GO term to be included.
<code>maxGmtSize</code>	The maximum number of genes for a GO term to be included.
<code>verbose</code>	A logical value indicating whether to print progress messages.

Value

A named list where names are GO terms and values are character vectors of Entrez IDs.

getMsigdbGmt	<i>Fetch Gene Sets from MSigDB</i>
--------------	------------------------------------

Description

A wrapper for ‘msigdb::msigdb’ to fetch gene sets and format them as a named list (GMT format).

Usage

```
getMsigdbGmt(species = "Homo sapiens", collection = "H", verbose = TRUE)
```

Arguments

species	The scientific name for the species (e.g., "Homo sapiens").
collection	The MSigDB collection code (e.g., "H" for hallmark).
verbose	A logical value indicating whether to print progress messages.

Value

A named list where names are gene set names and values are character vectors of gene symbols.

getNodeTable	<i>Get the Node Table from a ComplexMap Object</i>
--------------	--

Description

Access the node attributes (layout, function, specificity scores).

Usage

```
getNodeTable(cm)
```

Arguments

cm	A ‘ComplexMap’ object.
-----------	------------------------

Value

A ‘tibble’.

<code>getReactomeGmt</code>	<i>Fetch Reactome Pathway Gene Sets</i>
-----------------------------	---

Description

Fetches Reactome pathways and their associated genes from the ‘reactome.db’ Bioconductor annotation package.

Usage

```
getReactomeGmt(speciesDb, verbose = TRUE)
```

Arguments

<code>speciesDb</code>	An AnnotationDb object for the target species (e.g., ‘org.Hs.eg.db’). This is used to filter pathways for the correct species.
<code>verbose</code>	A logical value indicating whether to print progress messages.

Value

A named list where names are Reactome pathway names and values are character vectors of Entrez IDs.

<code>print.ComplexMap</code>	<i>Print Method for a ComplexMap Object</i>
-------------------------------	---

Description

Provides a “Systems Biology Dashboard” summary of the ComplexMap object.

Usage

```
## S3 method for class 'ComplexMap'
print(x, ...)
```

Arguments

<code>x</code>	A ‘ComplexMap’ object.
<code>...</code>	Additional arguments.

Details

This summary is designed to give immediate feedback on the **Functional Diversity** of the landscape:

- **Functional Diversity:** The count of distinct functional labels (colors). A high number (e.g., >15) indicates a rich, specific landscape. A low number indicates over-clustering or generic annotation.
- **Annotation Coverage:** The percentage of complexes that could be assigned a function. High coverage with high diversity is the ideal state.

Value

Invisibly returns the object.

pruneNetwork	<i>Prune Network Edges Before Cytoscape Export</i>
---------------------	--

Description

Reduces the number of edges in a ‘ComplexMap’ object to prevent browser overload during interactive exploration.

Usage

```
pruneNetwork(  
  cmap,  
  method = "top_k",  
  k = 5,  
  weight_quantile = 0.75,  
  verbose = TRUE  
)
```

Arguments

cmap	A ‘ComplexMap’ object.
method	Pruning method: “top_k” (keep top-k edges per node) or “quantile” (keep edges above a weight quantile).
k	Integer. Number of top edges to retain per node when ‘method = “top_k”’.
weight_quantile	Numeric (0-1). Quantile threshold when ‘method = “quantile”’. Defaults to ‘0.75’.
verbose	Logical.

Value

A modified ‘ComplexMap’ object with pruned edges.

qcComplexList
Perform Quality Control on a List of Protein Complexes

Description

A diagnostic tool to assess complex size distribution and distinguish between true redundancy (synonyms) and biological hierarchy (subsets).

Usage

```
qcComplexList(
  complexList,
  redundancyThreshold = 0.9,
  subsetThreshold = 0.9,
  verbose = TRUE
)
```

Arguments

complexList	A list of character vectors (protein complexes).
redundancyThreshold	Threshold for Jaccard similarity warning. Defaults to 0.9.
subsetThreshold	Threshold for Simpson coefficient (subset) reporting. Defaults to 0.9.
verbose	Logical.

Details

****Systems Biology Rationale:**** To maintain a diverse functional landscape, it is critical to distinguish between two types of similarity:

1. ****Jaccard Similarity (Redundancy):**** Complex A and B are nearly identical. ****Action:**** Merge these (using 'refineComplexList').
2. ****Overlap/Simpson Similarity (Hierarchy):**** Complex A is a subset of Complex B. ****Action:**** ****Keep these.**** Do not merge. These often represent distinct biological states (e.g., Core Complex vs. Holo-Complex).

This report calculates both metrics to guide your refinement strategy.

Value

Invisibly returns the input 'complexList'.

Author(s)

Qingzhou Zhang <zqzneptune@hotmail.com>

queryMap	<i>Query a ComplexMap Object</i>
----------	----------------------------------

Description

Targeted querying of complexes or proteins.

Usage

```
queryMap(complexMapObject, query, type)
```

Arguments

complexMapObject	A 'ComplexMap' object.
query	Search string.
type	"protein" or "complex".

Details

- 'type="protein"': Regex search for protein members. - 'type="complex"': Exact match for Complex ID.

Value

A tibble of matching nodes.

referenceComplexes	<i>Reference Protein Complex Set (CORUM)</i>
--------------------	--

Description

A reference dataset containing 324 merged CORUM protein complexes used in training protein-protein interaction scoring and clustering optimization procedures in the study.

Usage

```
referenceComplexes
```

Format

A named list with 324 elements. Each element is a character vector of protein identifiers.

Details

These complexes were curated from the CORUM database and merged to reduce redundancy for benchmarking purposes.

Source

Supplementary Table 3 (second tab) from: Havugimana PC, Hart GT, Nepusz T, Yang H, Turinsky AL, Li Z, Wang PI, Boutz DR, Fong V, Phanse S, Babu M, Craig SA, Hu P, Wan C, Vlasblom J, Dar VU, Bezginov A, Clark GW, Wu GC, Wodak SJ, Tillier ER, Paccanaro A, Marcotte EM, Emili A. *A census of human soluble protein complexes.* Cell. 2012 Aug 31;150(5):1068-81. doi: 10.1016/j.cell.2012.08.011. PMID: 22939629; PMCID: PMC3477804.

See Also

[demoComplexes](#)

refineComplexList	<i>Refine a List of Protein Complexes by Size and Redundancy</i>
-------------------	--

Description

This function filters complexes by size and merges highly redundant ones.

Usage

```
refineComplexList(
  complexList,
  minSize = 3,
  maxSize = 500,
  mergeThreshold = 0.9,
  similarityMethod = "jaccard",
  verbose = TRUE
)
```

Arguments

complexList	A named list where each element is a character vector of protein identifiers representing a complex.
minSize	Minimum complex size. Defaults to 3.
maxSize	Maximum complex size. Defaults to 500.
mergeThreshold	Numeric (0-1). Threshold for merging. Defaults to 0.90 (strict) to preserve distinct variants.
similarityMethod	Metric for merging. Defaults to "jaccard" . Options: * "jaccard" : (Default) 'Intersection / Union'. Penalizes size differences. Best for preserving specific variants. * "overlap" : 'Intersection / Min(A,B)'. Merges subsets into parents. Use only if you want to aggressively reduce data. * "matching_score" : 'Intersection ² / (SizeA * SizeB)'. Geometric approach. * "dice" : '2 * Intersection / (SizeA + SizeB)'.
verbose	Logical.

Details

****Systems Biology Rationale:**** To preserve the functional diversity of the input landscape, this function defaults to a conservative ****Jaccard**** similarity with a high threshold (0.90).

*** **Trust the Input:**** We assume the upstream clustering method identified biologically relevant variants (e.g., a complex with vs. without a regulatory subunit). *** **Minimal Merging:**** We only merge complexes if they are effectively synonyms (nearly identical composition).

We discourage using "overlap" or "matching_score" for this step, as they tend to merge sub-complexes into parents, destroying the specific functional signals you wish to visualize.

Value

A list containing: * 'refinedComplexes': The final list of merged complexes. * 'mergeMap': A tibble mapping original IDs to final IDs.

Author(s)

Qingzhou Zhang <qzqneptune@hotmail.com>

runComplexEnrichment *Run Gene Set Enrichment Analysis on a List of Complexes*

Description

Performs enrichment analysis for each complex against a GMT background.

Usage

```
runComplexEnrichment(
  complexList,
  gmt,
  pAdjustMethod = "Benjamini",
  pValueCutoff = 0.05,
  verbose = TRUE
)
```

Arguments

complexList	A named list of protein complexes (character vectors).
gmt	A named list of gene sets (character vectors).
pAdjustMethod	Method for multiple testing correction ("Benjamini", "Bonferroni", "FDR"). Defaults to "Benjamini".
pValueCutoff	Adjusted p-value cutoff. Defaults to 0.05.
verbose	Logical.

Details

****Systems Biology Rationale:**** To support the generation of a specific and diverse landscape, this function calculates and returns ****Fold Enrichment**** and ****Term Size**** in addition to standard p-values.

- ****Fold Enrichment**** is used by downstream functions to prioritize specific biological labels over generic ones. - ****Term Size**** (PopHits) allows you to filter out overly broad terms (e.g., those containing > 10

Value

A named list of tibbles. Each tibble contains: - 'ID': Term ID/Name - 'Description': Term Description - 'p.adjust': Adjusted p-value - 'Fold': Fold Enrichment (Observed/Expected) - 'TermSize': Number of genes in the term (Background) - 'Count': Number of genes in the term (Overlap)

Author(s)

Qingzhou Zhang <zqzneptune@hotmail.com>

runComplexMapApp

Run the ComplexMap Interactive Explorer

Description

Launches a Shiny application that renders a 'ComplexMap' object as an interactive Cytoscape.js network. Nodes are colored by functional domain, sized by protein count, and fully filterable via sidebar controls.

Usage

```
runComplexMapApp(
  cmap,
  prune = TRUE,
  prune_method = "top_k",
  k = 5,
  weight_quantile = 0.75,
  port = NULL,
  launch.browser = TRUE,
  verbose = TRUE
)
```

Arguments

cmap	A 'ComplexMap' object.
prune	Logical. Whether to prune edges before launching. Default 'TRUE'.
prune_method	Pruning method: "top_k" or "quantile". Default "top_k".

k	Integer. Top-k edges per node when ‘prune_method = "top_k"’. Default ‘5’.
weight_quantile	Numeric (0-1). Quantile threshold for “quantile” method. Default ‘0.75’.
port	Optional integer port for the Shiny server.
launch.browser	Logical. Open browser automatically. Default ‘TRUE’.
verbose	Logical.

Value

Launches the Shiny app (invisibly returns ‘NULL’).

scaleLayoutForCytoscape

Scale Layout Coordinates for Cytoscape.js

Description

Mean-centers and rescales raw layout coordinates to a fixed pixel range suitable for Cytoscape.js.

Usage

```
scaleLayoutForCytoscape(x, y, range = c(-500, 500))
```

Arguments

x	Numeric vector of x coordinates.
y	Numeric vector of y coordinates.
range	Numeric vector of length 2 giving target range. Default ‘c(-500, 500)’.

Value

A ‘list’ with elements ‘x’ and ‘y’, both rescaled.

toCytoscapeJSON

Export a ComplexMap Object to Cytoscape.js JSON

Description

Converts a ‘ComplexMap’ object into a Cytoscape.js-compatible JSON string, with scaled layout positions. This JSON is consumed directly by the Shiny/Cytoscape.js interactive explorer.

Usage

```
toCytoscapeJSON(
  cmap,
  prune = TRUE,
  prune_method = "top_k",
  k = 5,
  weight_quantile = 0.75,
  coord_range = c(-500, 500),
  verbose = TRUE
)
```

Arguments

cmap	A ‘ComplexMap’ object (with layout coordinates in the node table).
prune	Logical. Whether to prune the network before export. Default ‘TRUE’.
prune_method	Pruning method passed to ‘pruneNetwork()’. Default “top_k”.
k	Integer. Top-k edges per node if ‘prune_method = "top_k"’. Default ‘5’.
weight_quantile	Numeric. Quantile threshold if ‘prune_method = "quantile"’. Default ‘0.75’.
coord_range	Numeric length-2 vector for coordinate scaling. Default ‘c(-500, 500)’.
verbose	Logical.

Value

A JSON string suitable for ‘window.complexmapElements’ in JavaScript.

Index

- * **datasets**
 - demoComplexes, [8](#)
 - referenceComplexes, [19](#)
- benchmarkParameters, [2](#)
- buildComplexNetwork, [4](#)
- computeMapTopology, [5](#)
- createComplexMap, [6](#)
- demoComplexes, [8](#), [20](#)
- evaluateComplexes, [9](#)
- explore, [10](#)
- exportNetwork, [11](#)
- generateNodeAttributes, [12](#)
- getEdgeTable, [12](#)
- getExampleGmt, [13](#)
- getGmtFromFile, [14](#)
- getGoGmt, [14](#)
- getMsigdbGmt, [15](#)
- getNodeTable, [15](#)
- getReactomeGmt, [16](#)
- print.ComplexMap, [16](#)
- pruneNetwork, [17](#)
- qcComplexList, [18](#)
- queryMap, [19](#)
- referenceComplexes, [8](#), [19](#)
- refineComplexList, [20](#)
- runComplexEnrichment, [21](#)
- runComplexMapApp, [22](#)
- scaleLayoutForCytoscape, [23](#)
- toCytoscapeJSON, [24](#)