

# Package: hdWGCNA (via r-universe)

May 24, 2026

**Title** High-dimensional Weighted Gene Co-expression Network Analysis

**Version** 0.5.00

**Description** hdWGCNA is an R package for performing weighted gene co-expression network analysis in high dimensional -omics such as single-cell RNA-seq or spatial transcriptomics. It provides tools for constructing metacells, building co-expression networks, computing module eigengenes, and performing various downstream analyses including module trait correlation, module preservation, and transcription factor network analysis.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.3

**URL** <https://zaoqu-liu.github.io/hdWGCNA/>,  
<https://github.com/Zaoqu-Liu/hdWGCNA>

**BugReports** <https://github.com/Zaoqu-Liu/hdWGCNA/issues>

**Imports** WGCNA, Seurat (>= 4.0.0), SeuratObject (>= 4.0.0), Matrix, harmony (>= 1.0.1), igraph, ggraph, tidygraph, enrichR, ggplot2, ggrepel (>= 0.9.0), dplyr, tester, proxy, patchwork, magrittr, purrr, uwot, FNN, reshape2, methods, stats, utils, data.table, GeneOverlap, GenomicRanges

**Suggests** testthat (>= 3.0.0), BiocParallel, xgboost, NetRep, SingleCellExperiment, SummarizedExperiment, GenomeInfoDb, ensemblDb, motifmatchr, monocle3, ggforestplot, ggradar, ggtrastr, ggseqlogo, corrplot, qgraph, future.apply, Hmisc, scales, stringr, tidyr, gtools, devtools

**Config/testthat/edition** 3

**Depends** R (>= 4.0.0)

**Config/pak/sysreqs** cmake libfontconfig1-dev libfreetype6-dev libglpk-dev make libicu-dev libpng-dev libuv1-dev libxml2-dev libssl-dev perl python3 zlib1g-dev

**Repository** <https://zaoqu-liu.r-universe.dev>  
**Date/Publication** 2026-01-24 06:17:08 UTC  
**RemoteUrl** <https://github.com/Zaoqu-Liu/hdWGCNA>  
**RemoteRef** main  
**RemoteSha** 5c381c7246d672a0eebddf62619976ffb9f1399e

## Contents

AssignTFRegulons . . . . .	5
AvgModuleExpr . . . . .	6
BinPseudotime . . . . .	6
check_inputs . . . . .	7
CheckSeurat5 . . . . .	7
CheckWGCNAName . . . . .	8
ComputeModuleEigengene . . . . .	8
ConstructMetacells . . . . .	9
ConstructMetaspots . . . . .	10
ConstructNetwork . . . . .	11
ConstructPseudobulk . . . . .	12
ConstructTFNetwork . . . . .	13
corSparse . . . . .	14
DimPlotMetacells . . . . .	15
DoHubGeneHeatmap . . . . .	15
EnrichrBarPlot . . . . .	16
EnrichrDotPlot . . . . .	17
FindAllDMEs . . . . .	19
FindDifferentialRegulons . . . . .	20
FindDMEs . . . . .	21
FindMajorIsoforms . . . . .	22
GetActiveWGCNA . . . . .	23
GetActiveWGCNAName . . . . .	23
GetAvgModuleExpr . . . . .	24
GetDatExpr . . . . .	24
GetDegrees . . . . .	24
GetEnrichrRegulonTable . . . . .	25
GetEnrichrTable . . . . .	25
GetHubGenes . . . . .	26
GetMELoadings . . . . .	26
GetMEs . . . . .	27
GetMetacellObject . . . . .	27
GetMetacellParams . . . . .	28
GetModulePreservation . . . . .	28
GetModules . . . . .	29
GetModuleScores . . . . .	29
GetModuleTraitCorrelation . . . . .	30
GetModuleUMAP . . . . .	30

GetMotifMatrix . . . . .	31
GetMotifOverlap . . . . .	31
GetMotifs . . . . .	32
GetMotifScores . . . . .	32
GetMotifTargets . . . . .	33
GetMultiExpr . . . . .	33
GetNetworkData . . . . .	34
GetPFMList . . . . .	34
GetPowerTable . . . . .	35
GetRegulonScores . . . . .	35
GetTFEval . . . . .	36
GetTFNetwork . . . . .	36
GetTFRegulons . . . . .	36
GetTFTargetGenes . . . . .	37
GetTOM . . . . .	38
GetWGCNAGenes . . . . .	38
GetWGCNAParams . . . . .	39
HubGeneNetworkPlot . . . . .	39
MetacellsByGroups . . . . .	40
MetaspotsByGroups . . . . .	42
ModuleConnectivity . . . . .	42
ModuleCorrelogram . . . . .	44
ModuleCorrNetwork . . . . .	45
ModuleEigengenes . . . . .	45
ModuleExprScore . . . . .	47
ModuleFeaturePlot . . . . .	47
ModuleNetworkPlot . . . . .	48
ModulePreservation . . . . .	49
ModulePreservationNetRep . . . . .	50
ModuleRadarPlot . . . . .	52
ModuleRegulatoryHeatmap . . . . .	53
ModuleRegulatoryNetwork . . . . .	54
ModuleRegulatoryNetworkPlot . . . . .	55
ModuleTopologyBarplot . . . . .	57
ModuleTopologyHeatmap . . . . .	58
ModuleTraitCorrelation . . . . .	60
ModuleUMAPPlot . . . . .	61
MotifOverlapBarPlot . . . . .	62
MotifScan . . . . .	62
NormalizeMetacells . . . . .	64
OverlapBarPlot . . . . .	64
OverlapDotPlot . . . . .	65
OverlapModulesDEGs . . . . .	65
OverlapModulesMotifs . . . . .	66
PlotDendrogram . . . . .	67
PlotDifferentialRegulons . . . . .	67
PlotDMEsLollipop . . . . .	68
PlotDMEsVolcano . . . . .	69

PlotKMEs . . . . .	70
PlotLollipop . . . . .	71
PlotModulePreservation . . . . .	72
PlotModulePreservationLollipop . . . . .	73
PlotModuleTraitCorrelation . . . . .	74
PlotModuleTrajectory . . . . .	75
PlotSoftPowers . . . . .	76
ProjectModules . . . . .	76
ReassignModules . . . . .	78
RegulonBarPlot . . . . .	79
RegulonScores . . . . .	80
ResetModuleColors . . . . .	81
ResetModuleNames . . . . .	81
RunEnrichr . . . . .	82
RunEnrichrRegulons . . . . .	82
RunHarmonyMetacells . . . . .	83
RunModuleUMAP . . . . .	84
RunPCAMetacells . . . . .	84
RunUMAPMetacells . . . . .	85
scale01 . . . . .	85
ScaleMetacells . . . . .	85
SelectNetworkGenes . . . . .	86
SetActiveWGCNA . . . . .	87
SetAvgModuleExpr . . . . .	87
SetDatExpr . . . . .	88
SetDegrees . . . . .	89
SetEnrichrRegulonTable . . . . .	89
SetEnrichrTable . . . . .	90
SetMELoadings . . . . .	90
SetMEs . . . . .	91
SetMetacellObject . . . . .	91
SetMetacellParams . . . . .	92
SetModulePreservation . . . . .	92
SetModules . . . . .	93
SetModuleScores . . . . .	93
SetModuleTraitCorrelation . . . . .	94
SetModuleUMAP . . . . .	94
SetMotifMatrix . . . . .	95
SetMotifOverlap . . . . .	95
SetMotifs . . . . .	96
SetMotifScores . . . . .	96
SetMotifTargets . . . . .	97
SetMultiExpr . . . . .	97
SetNetworkData . . . . .	98
SetPFMList . . . . .	99
SetPowerTable . . . . .	99
SetRegulonScores . . . . .	100
SetTFEval . . . . .	100

SetTFNetwork . . . . .	101
SetTFRegulons . . . . .	101
SetupForWGCNA . . . . .	102
SetWGCNAGenes . . . . .	102
SetWGCNAParams . . . . .	103
shuffle_points . . . . .	103
TestSoftPowers . . . . .	104
TestSoftPowersConsensus . . . . .	105
TFNetworkPlot . . . . .	106
to_pseudobulk . . . . .	107
TransferModuleGenome . . . . .	108
umap_theme . . . . .	109

<b>Index</b>	<b>110</b>
--------------	------------

---

AssignTFRegulons	<i>AssignTFRegulons</i>
------------------	-------------------------

---

## Description

Define the set of likely target genes (Regulons) for each transcription factor

## Usage

```
AssignTFRegulons(
  seurat_obj,
  strategy = "A",
  reg_thresh = 0.01,
  n_tfs = 10,
  n_genes = 50,
  wgcna_name = NULL
)
```

## Arguments

seurat_obj	A Seurat object
strategy	method for defining regulons, "A", "B", or "C". See Details for more info.
reg_thresh	threshold for regulatory score in strategies A, B, and C
n_tfs	for strategy A, the number of top TFs to keep for each gene
n_genes	for strategy B, the number of top target genes to keep for each TF
wgcna_name	name of the WGCNA experiment

## Details

AssignTFRegulons uses the TF network information from ConstructTFNetwork to define sets of confident TF-gene pairs. A "regulon" is the set of target genes for a given TF, and this function provides three different strategies to define TF regulons. Strategy "A" selects the top TFs for each gene, strategy "B" selects the top genes for each TF, and strategy "C" retains all TF-gene pairs above a certain regulatory score (reg\_thresh).

**Value**

seurat\_obj with the TF Regulon information added

---

AvgModuleExpr	<i>AverageModuleExpr</i>
---------------	--------------------------

---

**Description**

Computes module eigengenes for all WGCNA co-expression modules

**Usage**

```
AvgModuleExpr(seurat_obj, n_genes = 25, wgcna_name = NULL, ...)
```

**Arguments**

seurat\_obj      A Seurat object

---

BinPseudotime	<i>BinPseudotime</i>
---------------	----------------------

---

**Description**

Makes evenly-spaced bins of cells along a pseudotime trajectory.

**Usage**

```
BinPseudotime(seurat_obj, n_bins = 50, pseudotime_col = "pseudotime")
```

**Arguments**

seurat\_obj      A Seurat object

n\_bins            number of hub genes to use in the UMAP computation

pseudotime\_col   name of the column in seurat\_obj@meta.data containing pseudotime information

---

check_inputs	<i>Check inputs</i>
--------------	---------------------

---

**Description**

Check inputs prior to running to\_pseudobulk

**Usage**

```
check_inputs(
  input,
  meta = meta,
  replicate_col = "replicate",
  cell_type_col = "cell_type",
  label_col = "label"
)
```

**Arguments**

input	a single-cell matrix to be converted, with features (genes) in rows and cells in columns. Alternatively, a Seurat, monocle3, or SingleCellExperiment object can be directly input.
meta	the accompanying meta data whereby the rownames match the column names of input.
replicate_col	the vector in meta containing the replicate information. Defaults to replicate.
cell_type_col	the vector in meta containing the cell type information. Defaults to cell_type.
label_col	the vector in meta containing the experimental label. Defaults to label.
min_cells	the minimum number of cells in a cell type to retain it. Defaults to 3.
min_reps	the minimum number of replicates in a cell type to retain it. Defaults to 2.
min_features	the minimum number of expressing cells (or replicates) for a gene to retain it. Defaults to 0.

**Value**

a cleaned up expression matrix and meta data object

---

CheckSeurat5	<i>CheckSeurat5</i>
--------------	---------------------

---

**Description**

Function to check if Seurat version is v5

**Usage**

```
CheckSeurat5()
```

---

CheckWGCNAName	<i>CheckWGCNAName</i>
----------------	-----------------------

---

**Description**

CheckWGCNAName

**Usage**

```
CheckWGCNAName(seurat_obj, wgcna_name)
```

**Arguments**

seurat_obj	A Seurat object
------------	-----------------

---

ComputeModuleEigengene	<i>ComputeModuleEigengene</i>
------------------------	-------------------------------

---

**Description**

Internal helper function that computes module eigengene for a single module.

**Usage**

```
ComputeModuleEigengene(
  seurat_obj,
  cur_mod,
  modules,
  group.by.vars = NULL,
  verbose = TRUE,
  vars.to.regress = NULL,
  scale.model.use = "linear",
  pc_dim = 1,
  assay = NULL,
  wgcna_name = NULL,
  ...
)
```

**Arguments**

seurat_obj	A Seurat object
cur_mod	name of a module found in <code>seurat_obj@misc[seurat_obj@misc\$active_wgcna]\$wgcna_net\$colors</code>
modules	table containing module / gene assignments, as in <code>GetModules(seurat_obj)</code> .
group.by.vars	groups to harmonize by

verbose	logical indicating whether to print messages
vars.to.regress	character vector of variables in <code>seurat_obj@meta.data</code> to regress when running <code>ScaleData</code>
scale.model.use	model to scale data when running <code>ScaleData</code> choices are "linear", "poisson", or "negbinom"
pc_dim	Which PC to use as the module eigengene? Default to 1.
assay	Assay in <code>seurat_obj</code> to compute module eigengenes. Default is <code>DefaultAssay(seurat_obj)</code>
wgcna_name	name of the WGCNA experiment

---

ConstructMetacells      *ConstructMetacells*

---

### Description

This function takes a Seurat object and constructs averaged 'metacells' based on neighboring cells.

### Usage

```
ConstructMetacells(
  seurat_obj,
  name = "agg",
  ident.group = "seurat_clusters",
  k = 25,
  reduction = "pca",
  dims = NULL,
  assay = "RNA",
  cells.use = NULL,
  slot = "counts",
  layer = "counts",
  meta = NULL,
  return_metacell = FALSE,
  mode = "average",
  max_shared = 15,
  target_metacells = 1000,
  max_iter = 5000,
  verbose = FALSE,
  wgcna_name = NULL
)
```

### Arguments

seurat_obj	A Seurat object
name	A string appended to resulting metalcells. Default = 'agg'
k	Number of nearest neighbors to aggregate. Default = 50

reduction	A dimensionality reduction stored in the Seurat object. Default = 'umap'
dims	A vector representing the dimensions of the reduction to use. Either specify the names of the dimensions or the indices. Default = NULL to include all dims.
assay	Assay to extract data for aggregation. Default = 'RNA'
slot	Slot to extract data for aggregation. Default = 'counts'. Slot is used with Seurat v4 instead of layer.
layer	Layer to extract data for aggregation. Default = 'counts'. Layer is used with Seurat v5 instead of slot.
return_metacell	Logical to determine if we return the metacell seurat object (TRUE), or add it to the misc in the original Seurat object (FALSE). Default to FALSE.
mode	determines how to make gene expression profiles for metacells from their constituent single cells. Options are "average" or "sum".
max_shared	the maximum number of cells to be shared across two metacells
target_metacells	the maximum target number of metacells to construct
max_iter	the maximum number of iterations in the metacells bootstrapping loop
verbose	logical indicating whether to print additional information
wgcna_name	name of the WGCNA experiment

---

ConstructMetaspots

*ConstructMetaspots*


---

## Description

Computes metaspots in a given Seurat object containing spatial transcriptomics data. This function is called by MetaspotsByGroups and should NOT be run directly!

## Usage

```
ConstructMetaspots(
  cur_seurat,
  mode = "sum",
  assay = "Spatial",
  slot = "counts",
  layer = "counts"
)
```

## Arguments

cur_seurat	A Seurat object
mode	"sum" or "average"
assay	Assay to extract data for aggregation. Default = 'Spatial'
slot	Slot to extract data for aggregation. Default = 'counts'
layer	Layer to extract data for aggregation. Default = 'counts'. Layer is used with Seurat v5 instead of slot.

---

ConstructNetwork	<i>ConstructNetwork</i>
------------------	-------------------------

---

### Description

Constructs a co-expression network and groups genes into modules given a Seurat object that has been prepared for network analysis.

### Usage

```
ConstructNetwork(
  seurat_obj,
  soft_power = NULL,
  min_power = 3,
  tom_outdir = "TOM",
  tom_name = NULL,
  consensus = FALSE,
  overwrite_tom = FALSE,
  wgcna_name = NULL,
  blocks = NULL,
  maxBlockSize = 30000,
  randomSeed = 12345,
  corType = "pearson",
  consensusQuantile = 0.3,
  networkType = "signed",
  TOMType = "signed",
  TOMDenom = "min",
  scaleTOMs = TRUE,
  calibrationQuantile = 0.95,
  sampleForCalibration = TRUE,
  sampleForCalibrationFactor = 1000,
  useDiskCache = TRUE,
  chunkSize = NULL,
  deepSplit = 4,
  pamStage = FALSE,
  detectCutHeight = 0.995,
  minModuleSize = 50,
  mergeCutHeight = 0.2,
  saveConsensusTOMs = TRUE,
  ...
)
```

### Arguments

seurat_obj	A Seurat object
soft_power	the soft power used for network construction. Automatically selected by default.
min_power	the smallest soft power to be selected if soft_power=NULL

tom_outdir	path to the directory where the TOM will be written
tom_name	prefix name given to the TOM output file
consensus	flag indicating whether or not to perform Consensus network analysis
wgcna_name	name of the WGCNA experiment
...	additional parameters passed to blockwiseConsensusModules

### Details

ConstructNetwork builds a co-expression network and identifies clusters of highly co-expressed genes (modules) from the metacell or metaspot expression matrix stored in the Seurat object. Before running this function, the following functions must be run on the input Seurat object:

1. SetupForWGCNA
2. MetacellsByGroups or MetaspotsByGroups, and NormalizeMetacells
3. SetDatExpr or SetMultiExpr
4. TestSoftPowers or TestSoftPowersConsensus

This function can also be used to perform consensus network analysis if consensus=TRUE is selected. ConstructNetwork calls the WGCNA function blockwiseConsensusModules to compute the adjacency matrix, topological overlap matrix, and to run the Dynamic Tree Cut algorithm to identify gene modules. blockwiseConsensusModules has numerous parameters but here we have selected default parameters that we have found to provide reasonable results on a variety of single-cell and spatial transcriptomic datasets.

### Value

seurat\_obj with the co-expression network and gene modules computed for the selected wgcna experiment

---

ConstructPseudobulk    *ConstructPseudobulk*

---

### Description

Constructs a "pseudobulk" gene expression matrix summarizing the expression levels of each gene across a grouping variable (cell types for example) in each biological replicate.

### Usage

```
ConstructPseudobulk(
  seurat_obj,
  group.by,
  replicate_col,
  label_col = NULL,
  assay = "RNA",
  slot = "counts",
```

```

    layer = "counts",
    min_reps = 20,
    wgcna_name = NULL
  )

```

### Arguments

seurat_obj	A Seurat object
group.by	column in seurat_obj@meta.data containing grouping info, ie clusters or cell-types
replicate_col	column in seurat_obj@meta.data denoting each replicate / sample
label_col	column in seurat_obj@meta.data denoting an additional label of interest, for example disease status or biological sex. This is not a required argument and is typically only used for consensus WGCNA
assay	Assay in seurat_obj containing isoform expression information.
slot	Slot to extract data for aggregation. Default = 'counts'
layer	Layer to extract data for aggregation. Default = 'counts'. Layer is used with Seurat v5 instead of slot.
min_reps	The minimum number of different biological replicates allowed. Error will be thrown if the number of reps is too low.
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

### Details

This function constructs pseudobulk gene expression profiles across the provided cell groups and the provided biological replicates. We note that low numbers of replicates are typical in single-cell and spatial transcriptomics due to the large monetary cost of running these experiments, and pseudobulk-ing your data for hdWGCNA is only recommended in the case where you have a sufficient number of replicates. Here we have set the minimum recommended number to 20. Using fewer than 20 replicates risks the results not being reproducible or robust, and therefore are not biologically meaningful due to spurious noisy correlations.

### Value

a matrix containing pseudobulk expression profiles

---

ConstructTFNetwork	<i>ConstructTFNetwork</i>
--------------------	---------------------------

---

### Description

Construct a network of transcription factors and target genes based on gene co-expression

**Usage**

```
ConstructTFNetwork(
  seurat_obj,
  model_params,
  nfold = 5,
  callbacks = NULL,
  wgcna_name = NULL
)
```

**Arguments**

seurat_obj	A Seurat object
model_params	a list of model parameters to pass to xgboost
nfold	number of folds for cross validation
callbacks	callback functions for the xgboost model
wgcna_name	name of the WGCNA experiment

**Details**

ConstructTFNetwork uses motif-gene information to build a directed network of transcription factors (TFs) and target genes. XGBoost regression is leveraged to model the expression of each gene based on its candidate TF regulators. This analysis gives us information about how important each TF is for predicting each gene, allowing us to prioritize the most likely regulators of each gene. This process is done on the gene expression matrix stored with SetDatExpr, which is typically the hdWGCNA metacell gene expression matrix.

**Value**

seurat\_obj with the TF network added

---

corSparse	<i>Sparse matrix correlation</i>
-----------	----------------------------------

---

**Description**

Compute the Pearson correlation matrix between columns of two sparse matrices.

**Usage**

```
corSparse(X, Y = NULL, cov = FALSE)
```

**Arguments**

X	A matrix
Y	A matrix
cov	return covariance matrix

**Details**

Originally from <http://stackoverflow.com/questions/5888287/running-cor-or-any-variant-over-a-sparse-matrix> and the qtcMatrix & Signac packages.

**Author(s)**

Michael Cysouw, Karsten Looschen

---

DimPlotMetacells      *DimPlotMetacells*

---

**Description**

Wrapper function to run Seurat's DimPlot function on the metacell object.

**Usage**

```
DimPlotMetacells(seurat_obj, ...)
```

**Arguments**

seurat\_obj      A Seurat object

---

DoHubGeneHeatmap      *Plots gene expression of hub genes as a heatmap*

---

**Description**

This function makes an expression heatmap of the top n hub genes per module using Seurat's DoHeatmap, and then assembles them all into one big heatmap.

**Usage**

```
DoHubGeneHeatmap(  
  seurat_obj,  
  n_hubs = 10,  
  n_cells = 200,  
  group.by = NULL,  
  module_names = NULL,  
  combine = TRUE,  
  draw.lines = TRUE,  
  disp.min = -2.5,  
  disp.max = 2.5,  
  wgcna_name = NULL  
)
```

**Arguments**

seurat_obj	A Seurat object
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

**Examples**

```
DoHubGeneHeatmap
```

---

EnrichrBarPlot

*EnrichrBarPlot*


---

**Description**

Generates bar plots from Enrichr data to visualize enriched terms for hdWGCNA modules in a Seurat object. The function outputs a PDF file for each module, with separate bar plots for each database.

**Usage**

```
EnrichrBarPlot(
  seurat_obj,
  outdir = "enrichr_plots",
  n_terms = 25,
  p_cutoff = 0.05,
  p_adj = TRUE,
  plot_size = c(6, 15),
  logscale = FALSE,
  plot_bar_color = NULL,
  plot_text_color = NULL,
  wgcna_name = NULL
)
```

**Arguments**

seurat_obj	A Seurat object
outdir	A string specifying the directory where the output PDF files will be saved. Default is "enrichr_plots".
n_terms	An integer indicating the number of top enriched terms to include in each bar plot. Default is 25.
p_cutoff	A numeric value specifying the significance threshold for including terms (p-value or adjusted p-value). Only terms with p-values below this threshold are plotted. Default is 0.05.
p_adj	Logical indicating whether to use the adjusted p-value (default: TRUE) or raw p-value for filtering terms.
plot_size	A numeric vector of length 2 specifying the width and height of the output PDF files in inches. Default is c(6, 15).

logscale	Logical specifying whether to log-transform the enrichment scores before plotting. Default is FALSE.
plot_bar_color	A string specifying the color of the bars in the bar plots. If NULL (default), bars are colored according to the module's assigned color.
plot_text_color	A string specifying the color of the text labels on the bar plots. If NULL (default), the color is automatically determined based on the bar color.
wgcna_name	The name of the hdWGCNA experiment in the <code>seurat_obj@misc</code> slot

### Details

This function processes the Enrichr output stored in a Seurat object, filters enriched terms by significance, and generates bar plots for each WGCNA module. Separate plots are created for each database included in the Enrichr results. The top enriched terms for each module and database are ordered by their combined enrichment score. If there are ties in the enrichment scores, the function uses all tied terms.

The bar plot text and bar colors can be customized, and plots can be saved with log-transformed enrichment scores if specified. Text wrapping is applied to long term names for better readability.

### Value

Generates PDF files in the specified output directory, with one file per WGCNA module. Each file contains bar plots for all databases associated with that module.

---

EnrichrDotPlot	<i>EnrichrDotPlot</i>
----------------	-----------------------

---

### Description

Generate a dot plot visualizing enrichment results from Enrichr for hdWGCNA modules.

### Usage

```
EnrichrDotPlot(  
  seurat_obj,  
  database,  
  mods = "all",  
  n_terms = 3,  
  p_cutoff = 0.05,  
  p_adj = TRUE,  
  break_ties = TRUE,  
  term_size = 10,  
  wgcna_name = NULL  
)
```

## Arguments

seurat_obj	A Seurat object
database	A character string specifying the name of the Enrichr database to use (e.g., "GO_Biological_Process_2021").
mods	A character vector specifying the names of modules to include in the plot. If mods = "all" (default), all modules except the "grey" module are included.
n_terms	An integer specifying the number of top enriched terms to plot for each module (default = 3).
p_cutoff	A numeric value specifying the p-value threshold for filtering enriched terms (default = 0.05).
p_adj	A logical value indicating whether to use adjusted p-values (TRUE, default) or raw p-values (FALSE) for filtering.
break_ties	A logical value indicating whether to randomly select among tied terms to enforce n_terms (default = TRUE).
term_size	A numeric value specifying the font size of the enriched terms displayed on the y-axis (default = 10).
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

## Details

This function creates dot plots from Enrichr results associated with hdWGCNA modules. Each module is represented by its most enriched terms from the specified Enrichr database. The size of the dots indicates the enrichment score, and the color indicates the statistical significance ( $-\log_{10}$  transformed p-value).

- The function first retrieves WGCNA module and Enrichr data from the specified Seurat object.
- Modules are filtered based on the mods parameter, and enriched terms are filtered by significance using p\_cutoff and p\_adj.
- The top n\_terms terms for each module are selected based on the Combined Score. If ties occur, the break\_ties parameter determines how they are resolved.
- A dot plot is generated where each dot represents an enriched term, its size corresponds to the Combined Score (log-transformed), and its color indicates the significance ( $-\log_{10}$  transformed p-value).

## Value

A ggplot2 object representing the dot plot of enriched terms for the specified modules and database.

---

FindAllDMEs

*FindAllDMEs*


---

## Description

Function to compare expression levels of co-expression modules between two sets of cell barcodes.

## Usage

```
FindAllDMEs(
  seurat_obj,
  group.by,
  features = "MEs",
  harmonized = TRUE,
  add_missing = FALSE,
  wgcna_name = NULL,
  test.use = "wilcox",
  only.pos = FALSE,
  logfc.threshold = 0,
  min.pct = 0,
  verbose = FALSE,
  pseudocount.use = 0,
  ...
)
```

## Arguments

seurat_obj	A Seurat object
group.by	column in seurat_obj@meta.data containing cell grouping information
features	indicate whether to use "MEs" or "ModuleScores" for the comparison
harmonized	logical determining whether or not to use harmonized MEs
add_missing	logical determining whether or not to add missing modules back into the resulting dataframe with NA values.
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot
...	Additional parameters for the FindMarkers function

## Value

A dataframe containing differential ME results

## Examples

```
FindAllDMEs
```

---

FindDifferentialRegulons

*FindDifferentialRegulons*


---

### Description

Function to compare TF regulon scores between two sets of cell barcodes.

### Usage

```
FindDifferentialRegulons(
  seurat_obj,
  barcodes1,
  barcodes2,
  assay = "RNA",
  slot = "data",
  layer = "data",
  wgcna_name = NULL,
  test.use = "wilcox",
  only.pos = FALSE,
  logfc.threshold = 0,
  min.pct = 0,
  verbose = FALSE,
  pseudocount.use = 0,
  ...
)
```

### Arguments

seurat_obj	A Seurat object
barcodes1	character vector containing cell barcodes for the first group to test. Positive fold-change means up-regulated in this group.
barcodes2	character vector containing cell barcodes for the second group to test. Negative fold-change means up-regulated in this group.
assay	Assay to extract data for aggregation. Default = 'RNA'
slot	Slot to extract data for aggregation. Default = 'counts'. Slot is used with Seurat v4 instead of layer.
layer	Layer to extract data for aggregation. Default = 'counts'. Layer is used with Seurat v5 instead of slot.
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot
...	Additional parameters for the Seurat FindMarkers function

### Details

FindDifferentialRegulons compares two groups based on their TF regulon scores. Three comparisons are made for different sets of features: positive regulon scores, negative regulon scores, and gene expression. The same settings for the test will be used for all three tests.

**Value**

A dataframe containing differential regulon results

---

FindDMEs

*FindDMEs*

---

**Description**

Function to compare expression levels of co-expression modules between two sets of cell barcodes.

**Usage**

```
FindDMEs(
  seurat_obj,
  barcodes1,
  barcodes2,
  features = "MEs",
  harmonized = TRUE,
  wgcna_name = NULL,
  add_missing = FALSE,
  test.use = "wilcox",
  only.pos = FALSE,
  logfc.threshold = 0,
  min.pct = 0,
  verbose = FALSE,
  pseudocount.use = 0,
  ...
)
```

**Arguments**

seurat_obj	A Seurat object
barcodes1	character vector containing cell barcodes for the first group to test. Positive fold-change means up-regulated in this group.
barcodes2	character vector containing cell barcodes for the second group to test. Negative fold-change means up-regulated in this group.
features	indicate whether to use "MEs" or "ModuleScores" for the comparison
harmonized	logical determining whether or not to use harmonized MEs
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot
add_missing	logical determining whether or not to add missing modules back into the resulting dataframe with NA values.
...	Additional parameters for the FindMarkers function

**Value**

A dataframe containing differential ME results

**Examples**

```
FindDMEs
```

---

```
FindMajorIsoforms      FindMajorIsoforms
```

---

**Description**

Finds the set of "major isoforms" for each gene in each cell population. The set of major isoforms consists of the isoforms accounting for a desired proportion of a gene's expression. Pseudobulk replicates in each sample and cell population are used for this calculation.

**Usage**

```
FindMajorIsoforms(
  seurat_obj,
  group.by,
  replicate_col,
  isoform_delim = "[.]",
  proportion_thresh = 0.8,
  low_thresh = 25,
  assay = "iso",
  slot = "counts",
  cluster_markers = NULL,
  wgcna_name = NULL
)
```

**Arguments**

seurat_obj	A Seurat object
group.by	column in seurat_obj@meta.data containing grouping info, ie clusters or cell-types
replicate_col	column in seurat_obj@meta.data denoting each replicate / sample
isoform_delim	
proportion_thresh	desired proportion of expression to define the set of major isoforms. Default = 0.8.
low_thresh	lower bound of expression level for considering an isoform as part of the major isoform set.
assay	Assay in seurat_obj containing isoform expression information.
slot	Slot in seurat_obj, default to counts slot.
cluster_markers	Cell population marker gene table from Seurat FindAllMarkers for the same cell populations specified in group.by. Optional parameter, will exclude isoforms that are not from marker genes.
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

**Details**

FindMajorIsoforms computes the set of major isoforms in a given Seurat object that contains isoform-level expression information. First, pseudobulk replicates are computed for the given cell populations and samples present in the Seurat object. For each gene in each cell population, we rank the gene's isoforms by expression level and take the top expressing isoforms that make up the desired proportion of the gene's total expression, making sure to exclude any very lowly expressed isoforms.

Optionally, the user may supply a marker gene table for each cell population (formatted like the output of Seurat FindAllMarkers), and then the algorithm will only return major isoforms of the given marker genes.

**Value**

a list of major isoforms for each cell population

---

GetActiveWGCNA	<i>GetActiveWGCNA</i>
----------------	-----------------------

---

**Description**

GetActiveWGCNA

**Usage**

```
GetActiveWGCNA(seurat_obj)
```

**Arguments**

seurat\_obj      A Seurat object

---

GetActiveWGCNAName	<i>GetActiveWGCNAName</i>
--------------------	---------------------------

---

**Description**

GetActiveWGCNAName

**Usage**

```
GetActiveWGCNAName(seurat_obj)
```

**Arguments**

seurat\_obj      A Seurat object

---

GetAvgModuleExpr	<i>GetAvgModuleExpr</i>
------------------	-------------------------

---

**Description**

GetAvgModuleExpr

**Usage**

```
GetAvgModuleExpr(seurat_obj, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

GetDatExpr	<i>GetDatExpr</i>
------------	-------------------

---

**Description**

This function gets the WGCNA expression matrix.

**Usage**

```
GetDatExpr(seurat_obj, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
------------	-----------------

---

GetDegrees	<i>GetDegrees</i>
------------	-------------------

---

**Description**

GetDegrees

**Usage**

```
GetDegrees(seurat_obj, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

GetEnrichRegulonTable  
*GetEnrichRegulonTable*

---

**Description**

GetEnrichRegulonTable

**Usage**

```
GetEnrichRegulonTable(seurat_obj, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

GetEnrichTable      *GetEnrichTable*

---

**Description**

GetEnrichTable

**Usage**

```
GetEnrichTable(seurat_obj, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

 GetHubGenes

*GetHubGenes*


---

### Description

Extract the top N hub genes for a given set of modules. This function outputs a table with the gene name, the module, and the kME for that module for the top N hub genes.

### Usage

```
GetHubGenes(seurat_obj, n_hubs = 10, mods = NULL, wgcna_name = NULL)
```

### Arguments

seurat_obj	A Seurat object
n_hubs	the number of hub genes to select for each module
mods	list of modules, selects all modules by default
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

 GetMELoadings

*GetMELoadings*


---

### Description

Function to retrieve module eigengens from Seurat object.

### Usage

```
GetMELoadings(seurat_obj, harmonized = TRUE, wgcna_name = NULL)
```

### Arguments

seurat_obj	A Seurat object
harmonized	logical indicating whether MEs have been harmonized
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

GetMEs	<i>GetMEs</i>
--------	---------------

---

**Description**

Function to retrieve module eigengens from Seurat object.

**Usage**

```
GetMEs(seurat_obj, harmonized = TRUE, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
harmonized	logical indicating whether MEs have been harmonized
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

GetMetacellObject	<i>GetMetacellObject</i>
-------------------	--------------------------

---

**Description**

GetMetacellObject

**Usage**

```
GetMetacellObject(seurat_obj, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

GetMetacellParams      *GetMetacellParams*

---

**Description**

GetMetacellParams

**Usage**

```
GetMetacellParams(seurat_obj, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

GetModulePreservation      *GetModulePreservation*

---

**Description**

GetModulePreservation

**Usage**

```
GetModulePreservation(seurat_obj, mod_name, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
mod_name	name of the module preservation test to store
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

GetModules	<i>GetModules</i>
------------	-------------------

---

**Description**

GetModules

**Usage**

```
GetModules(seurat_obj, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

GetModuleScores	<i>GetModuleScores</i>
-----------------	------------------------

---

**Description**

GetModuleScores

**Usage**

```
GetModuleScores(seurat_obj, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

GetModuleTraitCorrelation  
*GetModuleTraitCorrelation*

---

**Description**

GetModuleTraitCorrelation

**Usage**

```
GetModuleTraitCorrelation(seurat_obj, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

GetModuleUMAP            *GetModuleUMAP*

---

**Description**

GetModuleUMAP

**Usage**

```
GetModuleUMAP(seurat_obj, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

GetMotifMatrix	<i>GetMotifMatrix</i>
----------------	-----------------------

---

**Description**

GetMotifMatrix

**Usage**

```
GetMotifMatrix(seurat_obj)
```

**Arguments**

seurat\_obj      A Seurat object

**Examples**

```
GetMotifMatrix
```

---

GetMotifOverlap	<i>GetMotifOverlap</i>
-----------------	------------------------

---

**Description**

GetMotifOverlap

**Usage**

```
GetMotifOverlap(seurat_obj, wgcna_name = NULL)
```

**Arguments**

seurat\_obj      A Seurat object  
wgcna\_name      The name of the hdWGCNA experiment in the seurat\_obj@misc slot

---

`GetMotifs`*GetMotifs*

---

**Description**`GetMotifs`**Usage**`GetMotifs(seurat_obj)`**Arguments**`seurat_obj`     A Seurat object**Examples**`GetMotifs`

---

`GetMotifScores`*GetMotifScores*

---

**Description**`GetMotifScores`**Usage**`GetMotifScores(seurat_obj, wgcna_name = NULL)`**Arguments**`seurat_obj`     A Seurat object`wgcna_name`     The name of the hdWGCNA experiment in the `seurat_obj@misc` slot

---

GetMotifTargets	<i>GetMotifTargets</i>
-----------------	------------------------

---

**Description**

GetMotifTargets

**Usage**

```
GetMotifTargets(seurat_obj)
```

**Arguments**

seurat\_obj      A Seurat object

**Examples**

```
GetMotifTargets
```

---

GetMultiExpr	<i>GetMultiExpr</i>
--------------	---------------------

---

**Description**

This function gets the expression matrix from the metacell object.

**Usage**

```
GetMultiExpr(seurat_obj, wgcna_name = NULL)
```

**Arguments**

seurat\_obj      A Seurat object

---

GetNetworkData	<i>GetNetworkData</i>
----------------	-----------------------

---

**Description**

GetNetworkData

**Usage**

```
GetNetworkData(seurat_obj, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

GetPFMList	<i>GetPFMList</i>
------------	-------------------

---

**Description**

GetPFMList

**Usage**

```
GetPFMList(seurat_obj)
```

**Arguments**

seurat_obj	A Seurat object
------------	-----------------

**Examples**

```
GetPFMList
```

---

GetPowerTable	<i>GetPowerTable</i>
---------------	----------------------

---

**Description**

GetPowerTable

**Usage**

```
GetPowerTable(seurat_obj, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

GetRegulonScores	<i>GetRegulonScores</i>
------------------	-------------------------

---

**Description**

GetRegulonScores

**Usage**

```
GetRegulonScores(seurat_obj, target_type, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
target_type	dataframe storing the TF regulon scores
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

GetTFEval	<i>GetTFEval</i>
-----------	------------------

---

**Description**

GetTFEval

**Usage**

```
GetTFEval(seurat_obj, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

GetTFNetwork	<i>GetTFNetwork</i>
--------------	---------------------

---

**Description**

GetTFNetwork

**Usage**

```
GetTFNetwork(seurat_obj, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

GetTFRegulons	<i>GetTFRegulons</i>
---------------	----------------------

---

**Description**

GetTFRegulons

**Usage**

```
GetTFRegulons(seurat_obj, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

GetTFTargetGenes	<i>GetTFTargetGenes</i>
------------------	-------------------------

---

### Description

Retrieve target genes for a set of transcription factors (TFs) from a Seurat object, based on either regulons or the full TF network. The function allows exploration of direct targets or extending the network by adding target genes of TFs that are downstream in the network.

### Usage

```
GetTFTargetGenes(
  seurat_obj,
  selected_tfs,
  depth = 1,
  target_type = "both",
  use_regulons = TRUE,
  wgcna_name = NULL
)
```

### Arguments

seurat_obj	A Seurat object containing the TF network or regulon information. This must include WGCNA results in the @misc slot.
selected_tfs	A list of transcription factors (TFs) to use as the starting point for network exploration.
depth	Integer value specifying the number of layers to extend the TF network from the selected TFs. For example, if depth=2, the target genes of selected_tfs are shown, and additional target genes are shown for TFs that are targets of the original selected_tfs.
target_type	The type of target genes to include in the output. Options are "positive" (genes with expression positively correlated with TFs), "negative" (genes with expression negatively correlated with TFs), or "both" (default, includes all targets).
use_regulons	Logical flag indicating whether to use regulons (default = TRUE) or the entire TF network for target gene discovery.
wgcna_name	Character string specifying the name of the WGCNA experiment. This should be stored in the @misc slot of the Seurat object.

### Details

This function retrieves direct and extended targets of a set of TFs based on the regulatory network stored in the Seurat object. The depth parameter controls how many layers of TF-target interactions to explore, while the target\_type parameter allows filtering based on correlation direction. The use\_regulons flag specifies whether to use regulon data, which may provide a more confident set of interactions, or the full TF network.

**Value**

A data frame containing the TF-target interactions at each specified depth level, with additional information such as regulatory score, correlation, and depth.

---

GetTOM	<i>GetTOM</i>
--------	---------------

---

**Description**

GetTOM

**Usage**

```
GetTOM(seurat_obj, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

GetWGCNAGenes	<i>GetWGCNAGenes</i>
---------------	----------------------

---

**Description**

GetWGCNAGenes

**Usage**

```
GetWGCNAGenes(seurat_obj, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

GetWGCNAParams	<i>GetWGCNAParams</i>
----------------	-----------------------

---

**Description**

GetWGCNAParams

**Usage**

```
GetWGCNAParams(seurat_obj, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

HubGeneNetworkPlot	<i>HubGeneNetworkPlot</i>
--------------------	---------------------------

---

**Description**

Construct a unified network plot comprising hub genes for multiple modules.

**Usage**

```
HubGeneNetworkPlot(  
  seurat_obj,  
  mods = "all",  
  n_hubs = 6,  
  n_other = 3,  
  sample_edges = TRUE,  
  edge_prop = 0.5,  
  return_graph = FALSE,  
  edge.alpha = 0.25,  
  vertex.label.cex = 0.5,  
  hub.vertex.size = 4,  
  other.vertex.size = 1,  
  wgcna_name = NULL,  
  ...  
)
```

**Arguments**

seurat_obj	A Seurat object
mods	Names of the modules to plot. If mods = "all", all modules are plotted.
n_hubs	The number of hub genes to plot for each module.
n_other	The number of non-hub genes to sample from each module
edge_prop	The proportion of edges in the graph to sample.
return_graph	logical determining whether we return the graph (TRUE) or plot the graph (FALSE)
edge.alpha	Scaling factor for the edge opacity
vertex.label.cex	The font size of the gene labels
hub.vertex.size	The size of the hub gene nodes
other.vertex.size	The size of the other gene nodes
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

**Examples**

```
HubGeneNetworkPlot
```

---

MetacellsByGroups	<i>MetacellsByGroups</i>
-------------------	--------------------------

---

**Description**

This function takes a Seurat object and constructs averaged 'metacells' based on neighboring cells in provided groupings, such as cluster or cell type.

**Usage**

```
MetacellsByGroups(
  seurat_obj,
  group.by = c("seurat_clusters"),
  ident.group = "seurat_clusters",
  k = 25,
  reduction = "pca",
  dims = NULL,
  assay = NULL,
  slot = "counts",
  layer = "counts",
  mode = "average",
  cells.use = NULL,
  min_cells = 100,
  max_shared = 15,
```

```

    target_metacells = 1000,
    max_iter = 5000,
    verbose = FALSE,
    wgcna_name = NULL
  )

```

### Arguments

seurat_obj	A Seurat object
group.by	A character vector of Seurat metadata column names representing groups for which metacells will be computed.
k	Number of nearest neighbors to aggregate. Default = 50
reduction	A dimensionality reduction stored in the Seurat object. Default = 'pca'
dims	A vector representing the dimensions of the reduction to use. Either specify the names of the dimensions or the indices. Default = NULL to include all dims.
assay	Assay to extract data for aggregation. Default = 'RNA'
slot	Slot to extract data for aggregation. Default = 'counts'. Slot is used with Seurat v4 instead of layer.
layer	Layer to extract data for aggregation. Default = 'counts'. Layer is used with Seurat v5 instead of slot.
mode	determines how to make gene expression profiles for metacells from their constituent single cells. Options are "average" or "sum".
min_cells	the minimum number of cells in a particular grouping to construct metacells
max_shared	the maximum number of cells to be shared across two metacells
target_metacells	the maximum target number of metacells to construct
max_iter	the maximum number of iterations in the metacells bootstrapping loop
verbose	logical indicating whether to print additional information
wgcna_name	name of the WGCNA experiment
name	A string appended to resulting metacells. Default = 'agg'

### Details

MetacellsByGroups merges transcriptomically similar cells into "metacells". Given a dimensionally-reduced representation of the input dataset, this algorithm first uses KNN to identify similar cells. A bootstrapped sampling procedure is then used to group together similar cells until convergence is reached. Importantly, this procedure is done in a context-specific manner based on the provided group.by parameters. Typically this means that metacells will be constructed separately for each biological replicate, cell type or cell state, disease condition, etc. The metacell representation is considerably less sparse than the original single-cell dataset, which is preferable for co-expression network analysis or other analyses that rely on correlations.

### Value

seurat\_obj with a metacell seurat\_obj stored in the specified WGCNA experiment

---

MetaspotsByGroups      *MetaspotsByGroups*

---

### Description

Computes metaspots in a given Seurat object containing spatial transcriptomics data.

### Usage

```
MetaspotsByGroups(
  seurat_obj,
  group.by = c("seurat_clusters"),
  ident.group = "seurat_clusters",
  assay = "Spatial",
  slot = "counts",
  layer = "counts",
  mode = "sum",
  min_spots = 50,
  wgcna_name = NULL
)
```

### Arguments

group.by	A character vector of Seurat metadata column names representing groups for which metacells will be computed.
assay	Assay to extract data for aggregation. Default = 'Spatial'
slot	Slot to extract data for aggregation. Default = 'counts'
layer	Layer to extract data for aggregation. Default = 'counts'. Layer is used with Seurat v5 instead of slot.
mode	determines how to make gene expression profiles for metacells from their constituent single cells. Options are "average" or "sum".
min_spots	the minimum number of spots in a particular grouping to construct metaspots
wgcna_name	name of the WGCNA experiment
cur_seurat	A Seurat object

---

ModuleConnectivity      *ModuleConnectivity*

---

### Description

Computes eigengene-based connectivity (kME) based on module eigengenes and the feature expression in selected cell populations / spatial regions.

**Usage**

```
ModuleConnectivity(
  seurat_obj,
  group.by = NULL,
  group_name = NULL,
  corFnc = "bicor",
  corOptions = "use='p'",
  harmonized = TRUE,
  assay = NULL,
  slot = "data",
  layer = "data",
  sparse = TRUE,
  reassign_modules = TRUE,
  TOM_use = NULL,
  wgcna_name = NULL,
  ...
)
```

**Arguments**

<code>seurat_obj</code>	A Seurat object
<code>group.by</code>	column in <code>seurat_obj@meta.data</code> containing grouping info, ie clusters or cell-types
<code>group_name</code>	name of the group(s) in <code>group.by</code> to use for kME calculation
<code>corFnc</code>	character string specifying the function to be used to calculate co-expression similarity. Defaults to <code>bicor</code> . Any function returning values
<code>corOptions</code>	character string specifying additional arguments to be passed to the function given by <code>corFnc</code> . Use <code>"use = 'p', method = 'spearman'"</code> to obtain Spearman correlation. Use <code>"use = 'p'"</code> to obtain Pearson correlation.
<code>harmonized</code>	logical determining whether to use harmonized MEs for kME calculation
<code>assay</code>	Assay in <code>seurat_obj</code> containing expression information.
<code>slot</code>	Slot in <code>seurat_obj</code> , default to normalized 'data' slot.
<code>sparse</code>	logical indicating whether or not to run the correlation using a sparse matrix.
<code>reassign_modules</code>	logical indicating whether or not to reassign genes to different co-expression modules if their kME value in the assigned module is negative.
<code>TOM_use</code>	name of the hdWGCNA experiment that contains the TOM which will be used to compute the intramodular degree of each gene. If the TOM is not found, this calculation will be skipped.
<code>wgcna_name</code>	The name of the hdWGCNA experiment in the <code>seurat_obj@misc</code> slot

**Details**

`ModuleConnectivity` computes the eigengene-based connectivity (kME) of each feature and each module. This is done by correlating the gene expression signal with the module eigengenes for each

module. Features with high kME values have greater network connectivity within that module, and we can identify module hub genes by taking the top genes for each module by kME values.

We recommend using the `group.by` and `group_name` parameters that were previously used in the `SetDatExpr` function, so only the relevant cell populations / spatial regions are considered for computing kMEs.

Depending on the parameters for network analysis, sometimes a feature has a negative kME in the module that it was originally assigned. This discrepancy arises since the network construction is done on the metacell/metaspot matrix but the kME calculation is done on the cell/spot matrix. By default, we account for this by reassigning features with negative kMEs in their assigned module to the module that had the highest kME for that feature.

### Value

`seurat_obj` with the kMEs computed for the selected wgcna experiment

---

ModuleCorrelogram      *ModuleCorrelogram*

---

### Description

Plot Module Eigengene correlogram

### Usage

```
ModuleCorrelogram(
  seurat_obj,
  MEs2 = NULL,
  features = "hMEs",
  order = "original",
  method = "ellipse",
  exclude_grey = TRUE,
  type = "upper",
  tl.col = "black",
  tl.srt = 45,
  sig.level = c(1e-04, 0.001, 0.01, 0.05),
  pch.cex = 0.7,
  col = NULL,
  ncolors = 200,
  wgcna_name = NULL,
  wgcna_name2 = NULL,
  ...
)
```

### Arguments

<code>seurat_obj</code>	A Seurat object
<code>features</code>	What to plot? Can select hMEs, MEs, scores, or average

**Examples**

```
MECorrelogram
```

---

```
ModuleCorrNetwork      ModuleCorrNetworks
```

---

**Description**

Plot Module Eigengene correlogram

**Usage**

```
ModuleCorrNetwork(  
  seurat_obj,  
  cluster_col = NULL,  
  exclude_grey = TRUE,  
  features = "hMEs",  
  reduction = "umap",  
  cor_cutoff = 0.2,  
  label_vertices = FALSE,  
  edge_scale = 5,  
  vertex_size = 15,  
  niter = 100,  
  vertex_frame = FALSE,  
  wgcna_name = NULL  
)
```

**Arguments**

seurat\_obj A Seurat object

**Examples**

```
ModuleCorrNetwork
```

---

```
ModuleEigengenes      ModuleEigengenes
```

---

**Description**

Computes module eigengenes for co-expression modules

**Usage**

```
ModuleEigengenes(
  seurat_obj,
  group.by.vars = NULL,
  modules = NULL,
  vars.to.regress = NULL,
  scale.model.use = "linear",
  verbose = TRUE,
  assay = NULL,
  pc_dim = 1,
  wgcna_name = NULL,
  ...
)
```

**Arguments**

<code>seurat_obj</code>	A Seurat object
<code>group.by.vars</code>	groups to harmonize by
<code>modules</code>	table containing module / gene assignments, as in <code>GetModules(seurat_obj)</code> .
<code>vars.to.regress</code>	character vector of variables in <code>seurat_obj@meta.data</code> to regress when running <code>ScaleData</code>
<code>scale.model.use</code>	model to scale data when running <code>ScaleData</code> choices are "linear", "poisson", or "negbinom"
<code>verbose</code>	logical indicating whether to print messages
<code>assay</code>	Assay in <code>seurat_obj</code> to compute module eigengenes. Default is <code>DefaultAssay(seurat_obj)</code>
<code>pc_dim</code>	Which PC to use as the module eigengene? Default to 1.
<code>wgcna_name</code>	name of the WGCNA experiment

**Details**

`ModuleEigengenes` summarizes the gene expression signatures of entire co-expression modules. This is done by performing singular value decomposition (SVD) on a subset of the scaled expression matrix containing only features assigned to each module. The module eigengene (ME), defined as the first dimension of the SVD matrix, retains the most variation, and we use this vector as a summary of gene expression for the whole module.

The module gene expression matrix is first scaled using the `Seurat ScaleData` function. The user can optionally adjust for covariates of interest in this step using the `vars.to.regress` parameter. Additionally, the module eigengenes themselves can be adjusted for technical biases such as sequencing batch, dataset of origin, or other factors using the Harmony algorithm with the `group.by.vars` parameter.

**Value**

`seurat_obj` with the module eigengenes computed for the selected `wgcna` experiment

---

ModuleExprScore	<i>ModuleExprScore</i>
-----------------	------------------------

---

**Description**

Computes a module score for each co-expression module using Seurat AddModuleScore or UCell.

**Usage**

```
ModuleExprScore(
  seurat_obj,
  n_genes = 25,
  method = "Seurat",
  wgcna_name = NULL,
  ...
)
```

**Arguments**

seurat_obj	A Seurat object
n_genes	the number of genes to use for each module, ranked by kME. Setting n_genes = 'all' uses all of the genes in a module
method	selected method for module scoring, valid choices are "Seurat" or "UCell"
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

**Details**

ModuleExprScore provides an alternative function to ModuleEigengenes for summarizing the expression level of each module. The user can choose between Seurat AddModuleScore or UCell using the method parameter.

**Value**

seurat\_obj with module scores computed for the selected wgcna experiment

---

ModuleFeaturePlot	<i>ModuleFeaturePlot</i>
-------------------	--------------------------

---

**Description**

Plot module eigengenes as a FeaturePlot

**Usage**

```
ModuleFeaturePlot(
  seurat_obj,
  module_names = NULL,
  wgcna_name = NULL,
  reduction = "umap",
  features = "hMEs",
  order_points = TRUE,
  restrict_range = TRUE,
  point_size = 0.5,
  alpha = 1,
  label_legend = FALSE,
  ucell = FALSE,
  raster = FALSE,
  raster_dpi = 500,
  raster_scale = 1,
  plot_ratio = 1,
  title = TRUE
)
```

**Arguments**

seurat_obj	A Seurat object
features	What to plot? Can select hMEs, MEs, scores, or average
order	TRUE, FALSE, or "shuffle" are valid options

**Examples**

```
ModuleFeaturePlot
```

---

ModuleNetworkPlot	<i>ModuleNetworkPlot</i>
-------------------	--------------------------

---

**Description**

Visualizes the top hub genes for selected modules as a circular network plot with one inner circle and one outer circle.

**Usage**

```
ModuleNetworkPlot(
  seurat_obj,
  n_inner = 10,
  n_outer = 15,
  n_conns = 500,
  mods = "all",
```

```

    outdir = "ModuleNetworks",
    wgcna_name = NULL,
    plot_size = c(6, 6),
    edge.alpha = 0.25,
    edge.width = 1,
    vertex.label.cex = 1,
    vertex.size = 6,
    ...
)

```

### Arguments

seurat_obj	A Seurat object
n_inner	Number of genes to plot on the inner circle
n_outer	Number of genes to plot on the outer circle.
n_conns	Number of gene-gene co-expression connections to plot, sorted by the strongest connections.
mods	Names of the modules to plot. If mods = "all", all modules are plotted.
outdir	The directory where the plots will be stored.
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot
plot_size	A vector containing the width and height of the network plots. example: c(5,5)
edge.alpha	value between 0 and 1 determining the alpha (transparency) scaling factor for the network edges
edge.width	value determining the width of the network edges
vertex.label.cex	vertex label font size
vertex.size	vertex size

---

ModulePreservation      *ModulePreservation*

---

### Description

Computes module preservation statistics in a query dataset for a given reference dataset

### Usage

```

ModulePreservation(
  seurat_obj,
  seurat_ref,
  name,
  n_permutations = 500,
  parallel = FALSE,
  seed = 12345,
)

```

```

    gene_mapping = NULL,
    genome_ref_col = NULL,
    genome_query_col = NULL,
    return_raw = FALSE,
    wgcna_name = NULL,
    wgcna_name_ref = NULL,
    ...
)

```

### Arguments

<code>seurat_obj</code>	A Seurat object
<code>seurat_ref</code>	A Seurat object serving as the reference for the module preservation analysis
<code>name</code>	The name to give the module preservation analysis.
<code>n_permutations</code>	Number of permutations for the module preservation test.
<code>parallel</code>	logical determining whether to run preservation analysis in parallel
<code>seed</code>	random seed for the permutation analysis.
<code>gene_mapping</code>	a dataframe containing gene name mappings between the query and the reference dataset. One column should have the gene name in the query dataset, and another column should have the corresponding gene name in the reference dataset.
<code>genome_ref_col</code>	the column name containing the gene names for the reference dataset
<code>genome_query_col</code>	the column name containing the gene names for the query dataset
<code>return_raw</code>	if TRUE, returns the module preservation statistics, else returns <code>seurat_obj</code> with the stats added to the hdWGCNA experiment.
<code>wgcna_name</code>	The name of the hdWGCNA experiment in the <code>seurat_obj@misc</code> slot
<code>wgcna_name_ref</code>	The name of the hdWGCNA experiment in the <code>seurat_ref@misc</code> slot

### Details

ModulePreservation performs a statistical test to assess the preservation of co-expression modules identified in one dataset in an independent dataset. This method is originally described by Langfelder et al. in the 2011 paper "Is My Network Module Preserved and Reproducible?". This method can be used to assess biological differences in networks, as well as technical differences / reproducibility across different batches.

---

ModulePreservationNetRep

*ModulePreservationNetRep*

---

### Description

Computes module preservation statistics in a query dataset for a given reference dataset

**Usage**

```
ModulePreservationNetRep(
  seurat_query,
  seurat_ref,
  name,
  n_permutations = 10000,
  n_threads = 8,
  gene_mapping = NULL,
  genome_ref_col = NULL,
  genome_query_col = NULL,
  TOM_use = NULL,
  ref_power = 1,
  query_power = 1,
  wgcna_name = NULL,
  wgcna_name_ref = NULL,
  ...
)
```

**Arguments**

<code>seurat_query</code>	A Seurat object
<code>seurat_ref</code>	A Seurat object serving as the reference for the module preservation analysis
<code>name</code>	The name to give the module preservation analysis.
<code>n_permutations</code>	Number of permutations for the module preservation test.
<code>n_threads</code>	Number of parallel threads to use for the module preservation test
<code>gene_mapping</code>	a dataframe containing gene name mappings between the query and the reference dataset. One column should have the gene name in the query dataset, and another column should have the corresponding gene name in the reference dataset.
<code>genome_ref_col</code>	the column name containing the gene names for the reference dataset
<code>genome_query_col</code>	the column name containing the gene names for the query dataset
<code>TOM_use</code>	The name of the hdWGCNA experiment containing the TOM that will be used for plotting
<code>wgcna_name</code>	The name of the hdWGCNA experiment in the <code>seurat_obj@misc</code> slot
<code>wgcna_name_ref</code>	The name of the hdWGCNA experiment in the <code>seurat_ref@misc</code> slot

**Details**

ModulePreservationNetRep performs a statistical test to assess the preservation of co-expression modules identified in one dataset in an independent dataset using the NetRep implementation rather than the WGCNA implementation. This method is originally described by Langfelder et al. in the 2011 paper "Is My Network Module Preserved and Reproducible?". The NetRep implementation of the module preservation test is described by Ritchie et al. in the 2016 paper "A Scalable Permutation Approach Reveals Replication and Preservation Patterns of Networks Modules in Large

Datasets" This method can be used to assess biological differences in networks, as well as technical differences / reproducibility across different batches. Note that the outputs of ModulePreservation and ModulePreservationNetRep are not identical. This function requires separate installation of the NetRep package, which is not directly included as a dependency in hdWGCNA.

### Value

A Seurat Object containing the module preservation statistics from NetRep.

---

ModuleRadarPlot	<i>ModuleRadarPlot</i>
-----------------	------------------------

---

### Description

Plots the expression level (module eigengene) of each co-expression module for different groups as a radar plot.

### Usage

```
ModuleRadarPlot(
  seurat_obj,
  group.by = NULL,
  barcodes = NULL,
  combine = TRUE,
  ncol = 4,
  features = "hMEs",
  wgcna_name = NULL,
  fill = TRUE,
  draw.points = FALSE,
  ...
)
```

### Arguments

seurat_obj	A Seurat object
group.by	the column name of the selected comparison in the DMEs dataframe
barcodes	A list of barcodes from colnames(seurat_obj) which will be used to subset the data before plotting.
combine	logical indicating whether or not to combine plots using patchwork
ncol	The number of columns for the combined plot if patchwork is being used
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot
...	additional parameters for ggradar

**Details**

ModuleRadarPlot visualizes the expression level (module eigengene, ME) of each co-expression module for different groups in a radial coordinate system. The ME for each module is averaged for each group, scaled, and then is plotted radially. The resulting plots help us to interpret which cell groups (clusters, cell types, etc) are expressing each module.

**Value**

ggplot object containing the ModuleRadarPlot

---

ModuleRegulatoryHeatmap

*ModuleRegulatoryHeatmap*

---

**Description**

This function visualizes the regulatory network between gene modules as a heatmap, where each cell represents the regulatory score between a source and target module. The heatmap can display either positive, negative, or delta (positive minus negative) regulatory scores. Optionally, a dendrogram can be plotted to cluster modules based on their regulatory interactions.

**Usage**

```
ModuleRegulatoryHeatmap(
  seurat_obj,
  feature = "delta",
  TFs_only = TRUE,
  dendrogram = TRUE,
  coord_fixed = TRUE,
  max_val = 1,
  min_val_label = 3,
  high_color = "orange2",
  mid_color = "white",
  low_color = "dodgerblue",
  wgcna_name = NULL
)
```

**Arguments**

seurat_obj	A Seurat object containing single-cell data and WGCNA results.
feature	A character string specifying the type of regulatory score to plot. Options are 'positive' (positive regulatory score), 'negative' (negative regulatory score), or 'delta' (difference between positive and negative scores). Default is 'delta'.
TFs_only	Logical; if TRUE (default), only transcription factor (TF) genes are included in the heatmap. If FALSE, all genes are considered in the regulatory network.

dendrogram	Logical; if TRUE (default), a dendrogram is added to the heatmap to cluster modules based on their regulatory interactions. Default is TRUE.
coord_fixed	Logical; if TRUE (default), aspect ratio in x and y axes are equal. Default is TRUE.
max_val	Numeric; sets the maximum absolute value for the regulatory score. Any values exceeding this threshold are capped. Default is 1.
min_val_label	Numeric; the minimum number of interactions required for a label to be displayed on a heatmap cell. Default is 3.
high_color	Character string; the color representing high regulatory scores in the heatmap. Default is 'orange2'.
mid_color	Character string; the color representing intermediate regulatory scores in the heatmap. Default is 'white'.
low_color	Character string; the color representing low regulatory scores in the heatmap. Default is 'dodgerblue'.
wgcna_name	The name of the hdWGCNA experiment in the <code>seurat_obj@misc</code> slot

### Details

The function visualizes the regulatory network between gene modules by plotting regulatory scores as a heatmap. Each cell in the heatmap represents the regulatory interaction between a source module (columns) and a target module (rows). The color of the cell represents the magnitude and direction (positive or negative) of the regulatory score. Users can choose to plot positive, negative, or delta scores and can adjust the color gradient and score thresholds. A dendrogram can be added to cluster modules based on their regulatory patterns, and labels can be shown for cells with a sufficient number of interactions.

### Value

A `ggplot2` object visualizing the module regulatory network as a heatmap, which can be further customized or displayed using `plot()`.

---

ModuleRegulatoryNetwork

*ModuleRegulatoryNetwork*

---

### Description

Summarizes Transcription Factor regulatory networks across co-expression modules.

### Usage

```
ModuleRegulatoryNetwork(seurat_obj, TFs_only = TRUE, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object containing the single-cell data and WGCNA results.
Tfs_only	Logical; if TRUE (default), only transcription factor (TF) genes are included in the regulatory network. If FALSE, the network includes all genes.
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

**Details**

This function summarizes transcription factor regulatory networks across modules to infer module-module relationships. The number of directed TF links across modules are counted and normalized. The strength of these links from the XGBoost model are also tracked.

**Value**

A data frame with the following columns:

source	The source module in the regulatory interaction.
target	The target module in the regulatory interaction.
n_pos	The number of positive regulatory links from the source module to the target module.
sum_pos	The sum of the gain values for the positive links.
n_neg	The number of negative regulatory links from the source module to the target module.
sum_neg	The sum of the gain values for the negative links.
mean_pos	The average gain of the positive links.
mean_neg	The average gain of the negative links.
score_pos	The number of positive links normalized by the number of TFs in the source module.
score_neg	The number of negative links normalized by the number of TFs in the source module.

---

ModuleRegulatoryNetworkPlot

*ModuleRegulatoryNetworkPlot*

---

**Description**

This function visualizes the regulatory network between gene modules from a Seurat object. The plot displays transcription factor (TF) regulatory interactions between modules, with edges representing positive or negative regulatory links.

**Usage**

```
ModuleRegulatoryNetworkPlot(
  seurat_obj,
  feature = "delta",
  TFs_only = TRUE,
  layout = "umap",
  umap_background = FALSE,
  max_val = 1,
  cutoff = 0,
  focus_source = NULL,
  focus_target = NULL,
  loops = TRUE,
  label_modules = TRUE,
  high_color = "orange2",
  mid_color = "white",
  low_color = "dodgerblue",
  wgcna_name = NULL,
  ...
)
```

**Arguments**

seurat_obj	A Seurat object containing single-cell data and WGCNA results.
feature	A character string specifying the type of regulatory score to plot. Options are 'positive' (positive regulatory score), 'negative' (negative regulatory score), or 'delta' (difference between positive and negative scores). Default is 'delta'.
TFs_only	Logical; if TRUE (default), only transcription factor (TF) genes are included in the plot. If FALSE, all genes are considered in the regulatory network.
layout	A character string specifying the layout for the plot. Default is 'umap', which arranges the modules according to UMAP coordinates. Other layout options from ggraph can also be used.
umap_background	Logical; if TRUE, the UMAP of module eigengenes is plotted in the background (if layout = 'umap'). Default is FALSE.
max_val	Numeric; sets the maximum absolute value for the regulatory score. Any values exceeding this threshold are capped. Default is 1.
cutoff	Numeric; edges with absolute regulatory scores below this value are excluded from the plot. Default is 0.
focus_source	Character vector; optionally restricts the plot to only show regulatory links from specified source modules. Default is NULL (all modules included).
focus_target	Character vector; optionally restricts the plot to only show regulatory links targeting specified modules. Default is NULL (all modules included).
loops	Logical; if TRUE (default), loops (self-regulatory connections) are shown in the plot.
label_modules	Logical; if TRUE (default), module names are displayed as labels in the plot.

high_color	Character string; the color representing high regulatory scores in the color gradient. Default is 'orange2'.
mid_color	Character string; the color representing intermediate regulatory scores in the color gradient. Default is 'white'.
low_color	Character string; the color representing low regulatory scores in the color gradient. Default is 'dodgerblue'.
wgcna_name	The name of the hdWGCNA experiment in the <code>seurat_obj@misc</code> slot
...	Additional arguments passed to the <code>ggraph::create_layout</code> function.

### Details

The function visualizes the regulatory network between gene modules by plotting transcription factor regulatory interactions. Positive and negative regulatory scores are displayed as edges connecting nodes (modules), with node sizes proportional to the number of genes in each module. Users can customize the network layout (e.g., UMAP or other graph layouts), filter edges based on score thresholds, and highlight specific modules by using the focus parameters. Edge colors represent the strength and direction (positive or negative) of regulatory interactions, and loops can optionally be plotted to indicate self-regulation.

### Value

A `ggplot2` object visualizing the regulatory network, which can be further customized or displayed using `plot()`.

---

ModuleTopologyBarplot *ModuleTopologyBarplot*

---

### Description

Plots a ranked barplot of genes in a co-expression module by intramodular connectivity

### Usage

```
ModuleTopologyBarplot(
  seurat_obj,
  mod,
  features = "kME",
  plot_color = NULL,
  alpha = TRUE,
  genes_order = NULL,
  return_genes = FALSE,
  wgcna_name = NULL
)
```

**Arguments**

seurat_obj	A Seurat object
mod	the name of the co-expression module to plot
features	specify the features to use in the barplot, 'kME' or 'degree' or 'weighted_degree' (degree scaled to 0 or 1)
plot_color	color used for the bar plot, default is the module's unique color
alpha	logical indicating whether or not to add opacity to the barplot based on the strength (kME or degree)
genes_order	a character vector of genes to plot in this specific order, this option will override the order_by parameter
return_genes	logical indicating whether or not to return
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

**Details**

ModuleTopologyBarplot generates a barplot showing the intramodular connectivity of each gene in a specific co-expression module. Each bar in this plot represents a single gene, and they are ranked based on the strength of their connections within that particular module. A custom gene ordering can be supplied, which is helpful when comparing the module topologies side by side with more than one dataset.

**Value**

ggplot object containing the ModuleTopologyBarplot

---

ModuleTopologyHeatmap *ModuleTopologyHeatmap*

---

**Description**

Plots a heatmap of the co-expression network topology of a given module.

**Usage**

```
ModuleTopologyHeatmap(
  seurat_obj,
  mod,
  matrix = "TOM",
  matrix_name = NULL,
  order_by = "kME",
  high_color = NULL,
  low_color = "white",
  raster = TRUE,
  raster_dpi = 200,
  plot_max = "q99",
```

```

    plot_min = 0,
    return_genes = FALSE,
    genes_order = NULL,
    TOM_use = NULL,
    datExpr_use = NULL,
    wgcna_name = NULL,
    ...
)

```

### Arguments

seurat_obj	A Seurat object
mod	the name of the co-expression module to plot
matrix	specify which matrix to plot, use 'TOM' (topological overlap matrix) or 'Cor' (correlation matrix), or pass a custom square matrix where the rownames and colnames match the genes in this module
matrix_name	name of the matrix plotted that will be used as the label in the plot legend
order_by	order genes in this module by 'kME' (default) or by 'degree' (sum of all connections to this gene in the co-expression network)
high_color	color used for high values in the heatmap, default is the module's unique color
low_color	color used for low values in the heatmap, default is 'white'
raster	logical indicating whether or not to rasterise the plot
raster_dpi	dpi used for a rasterised plot
plot_max	maximum value to plot on the heatmap, can pass a numeric value or a string indicating the quantile ('q99' would be the 99th percentile)
plot_min	minimum value to plot on the heatmap, can pass a numeric value or a string indicating the quantile ('q1' would be the 1st percentile)
return_genes	logical indicating whether or not to return
genes_order	a character vector of genes to plot in this specific order, this option will override the order_by parameter
TOM_use	The name of the hdWGCNA experiment containing the TOM that will be used for plotting
datExpr_use	The name of the hdWGCNA experiment containing the datExpr that will be used for calculating the correlation matrix
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

### Details

ModuleTopologyHeatmap generates a triangular heatmap plot showing the network "topology" of a specific co-expression module. Each cell in the heatmap represents a gene-gene pair, and the heatmap is colored by the strength of the connection between these two genes. By default the genes in this heatmap are ordered in both the rows and the columns based on their importance in the module, ranked either by eigengene-based connectivity (kME) or by network degree.

**Value**

ggplot object containing the ModuleTopologyHeatmap

---

ModuleTraitCorrelation

*ModuleTraitCorrelation'*

---

**Description**

Correlates categorical and numeric variables with Module Eigengenes or hub-gene scores.

**Usage**

```
ModuleTraitCorrelation(
  seurat_obj,
  traits,
  group.by = NULL,
  features = "hMEs",
  cor_method = "pearson",
  subset_by = NULL,
  subset_groups = NULL,
  wgcna_name = NULL,
  ...
)
```

**Arguments**

seurat_obj	A list of column names in the Seurat object's metadata that you wish to correlate with each module. Traits must be a categorical variable (not a character vector), or a numeric variable.
features	Which features to use to summarize each modules? Valid choices are hMEs, MEs, or scores
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot
cor_meth	Which method to use for correlation? Valid choices are pearson, spearman, kendall.

**Examples**

```
ModuleTraitCorrelation
```

---

ModuleUMAPPlot	<i>ModuleUMAPPlot</i>
----------------	-----------------------

---

**Description**

Makes a igraph network plot using the module UMAP

**Usage**

```
ModuleUMAPPlot(
  seurat_obj,
  sample_edges = TRUE,
  edge_prop = 0.2,
  label_hubs = 5,
  edge.alpha = 0.25,
  vertex.label.cex = 0.5,
  label_genes = NULL,
  return_graph = FALSE,
  keep_grey_edges = TRUE,
  wgcna_name = NULL,
  ...
)
```

**Arguments**

seurat_obj	A Seurat object
sample_edges	logical determining whether we downsample edges for plotting (TRUE), or take the strongest edges.
edge_prop	proportion of edges to plot. If sample_edges=FALSE, the strongest edges are selected.
label_hubs	the number of hub genes to label in each module
edge.alpha	scaling factor for edge opacity
vertex.label.cex	font size for labeled genes
return_graph	logical determining whether to plot the graph (FALSE) or return the igraph object (TRUE)
keep_grey_edges	logical determining whether to show edges between genes in different modules (grey edges)
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

**Examples**

```
ModuleUMAPPlot
```

---

MotifOverlapBarPlot     *Displays the top n TFs in a set of modules as a bar plot*

---

### Description

Displays the top n TFs in a set of modules as a bar plot

### Usage

```
MotifOverlapBarPlot(
  seurat_obj,
  n_tfs = 10,
  plot_size = c(5, 6),
  outdir = "MotifOverlaps/",
  motif_font = "helvetica_regular",
  module_names = NULL,
  wgcna_name = NULL
)
```

### Arguments

seurat_obj	A Seurat object
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

### Examples

```
MotifOverlapBarPlot
```

---

MotifScan     *MotifScan*

---

### Description

This function scans the promoter regions of protein-coding genes for transcription factor (TF) motifs. It extracts promoter sequences using an Ensembl database (EnsDb) and then searches these sequences for TF binding motifs using position frequency matrices (PFMs). The Seurat object is updated with a matrix of motif-gene matches, a list of target genes for each TF, and additional motif information.

### Usage

```
MotifScan(seurat_obj, pfm, EnsDb, species_genome, wgcna_name = NULL)
```

### Arguments

<code>seurat_obj</code>	A Seurat object that will be updated with motif scan results.
<code>pfm</code>	A list of position frequency matrices (PFMs), such as those from the JASPAR2020 database.
<code>EnsDb</code>	An Ensembl database object (e.g., <code>EnsDb.Hsapiens.v86</code> or <code>EnsDb.Mmusculus.v79</code> ) containing gene and promoter annotations.
<code>species_genome</code>	A character string specifying the genome version (e.g., "hg38" for human or "mm10" for mouse).
<code>wgcna_name</code>	A character string specifying the name of the WGCNA experiment to associate with the motif data (optional). If NULL, the active WGCNA experiment in <code>seurat_obj@misc</code> will be used.

### Details

The MotifScan function performs the following steps:

- It extracts promoter regions (typically 2 kb upstream of the transcription start site) of protein-coding genes from the provided EnsDb.
- It uses the `motifmatchr` package to search these promoters for TF motifs, using the input PFMs.
- The function returns the Seurat object updated with several key pieces of information:
  - A motif-gene match matrix indicating the presence or absence of each motif in the promoter of each gene.
  - A list of target genes for each TF based on motif presence.
  - A summary of motifs, including the number of target genes for each TF.
- This data is stored in the `seurat_obj`'s metadata and can be used for downstream analysis, such as regulatory network inference.

### Value

A modified Seurat object containing the results of the motif scan, including:

- `seurat_obj@misc$motif_matrix`: A binary matrix indicating motif matches for each gene.
- `seurat_obj@misc$motif_info`: A data frame containing motif names, IDs, and the number of target genes.
- `seurat_obj@misc$motif_targets`: A list of genes targeted by each motif.
- `seurat_obj@misc$pfm`: The original PFMs used for the motif scan.

---

NormalizeMetacells	<i>NormalizeMetacells</i>
--------------------	---------------------------

---

**Description**

Wrapper function to run Seurat's NormalizeData function on the metacell object.

**Usage**

```
NormalizeMetacells(seurat_obj, wgcna_name = NULL, ...)
```

**Arguments**

seurat_obj	A Seurat object
------------	-----------------

---

OverlapBarPlot	<i>OverlapBarPlot</i>
----------------	-----------------------

---

**Description**

Plots the results from OverlapModulesDEGs as a bar plot

**Usage**

```
OverlapBarPlot(
  overlap_df,
  plot_var = "odds_ratio",
  logscale = FALSE,
  neglog = FALSE,
  label_size = 2,
  ...
)
```

**Arguments**

overlap_df	the Module/DEG overlap table from OverlapModulesDEGs
plot_var	the name of the overlap statistic to plot
logscale	logical controlling whether to plot the result on a log scale, useful for odds ratio
neglog	logical controlling whether to plot the result as a negative log, useful for p-value / FDR
label_size	the size of the module labels in the bar plot

**Examples**

```
OverlapBarPlot
```

---

OverlapDotPlot	<i>OverlapDotPlot</i>
----------------	-----------------------

---

**Description**

Makes barplots from Enrichr data

**Usage**

```
OverlapDotPlot(
  overlap_df,
  plot_var = "odds_ratio",
  logscale = TRUE,
  neglog = FALSE,
  plot_significance = TRUE,
  ...
)
```

**Arguments**

overlap_df	the Module/DEG overlap table from OverlapModulesDEGs
plot_var	the name of the overlap statistic to plot
logscale	logical controlling whether to plot the result on a log scale, useful for odds ratio
neglog	logical controlling whether to plot the result as a negative log, useful for p-value / FDR
plot_significance	logical controlling whether to plot the significance levels on top of the dots

**Examples**

```
OverlapDotPlot
```

---

OverlapModulesDEGs	<i>OverlapModulesDEGs</i>
--------------------	---------------------------

---

**Description**

Performs Fisher's Exact Test for overlap between DEGs and hdWGCNA modules.

**Usage**

```
OverlapModulesDEGs(
  seurat_obj,
  deg_df,
  fc_cutoff = 0.5,
  group_col = "cluster",
  wgcna_name = NULL,
  ...
)
```

**Arguments**

seurat_obj	A Seurat object
deg_df	DEG table formatted like the output from Seurat's FindMarkers
fc_cutoff	log fold change cutoff for DEGs to be included in the overlap test
group_col	the name of the column in deg_df containing the cell grouping information
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

OverlapModulesMotifs *Overlap modules with TF target genes*

---

**Description**

Overlap modules with TF target genes

**Usage**

```
OverlapModulesMotifs(seurat_obj, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

**Examples**

```
OverlapModulesDEGs
```

---

PlotDendrogram      *PlotDendrogram*

---

### Description

Plot WGCNA dendrogram

### Usage

```
PlotDendrogram(  
  seurat_obj,  
  groupLabels = "Module colors",  
  wgcna_name = NULL,  
  dendroLabels = FALSE,  
  hang = 0.03,  
  addGuide = TRUE,  
  guideHang = 0.05,  
  main = "",  
  ...  
)
```

### Arguments

seurat\_obj      A Seurat object

### Examples

```
PlotDendrogram
```

---

PlotDifferentialRegulons  
*PlotDifferentialRegulons*

---

### Description

Function to visualize differential TF regulon activity between two groups of cells based on regulon scores. The plot shows the average log fold-change (logFC) for positive and negative regulon scores in a scatter plot, with the points colored by their associated module and optionally labeled with the most differentially active regulons.

**Usage**

```
PlotDifferentialRegulons(
  seurat_obj,
  dregs,
  n_label = 10,
  logfc_thresh = 0.1,
  lm = TRUE,
  wgcna_name = NULL
)
```

**Arguments**

seurat_obj	A Seurat object that contains the WGCNA experiment and TF regulon scores.
dregs	Dataframe output from the FindDifferentialRegulons function containing differential regulon results.
n_label	Integer specifying the number of top up- and down-regulated regulons to label on the plot. If 'all', all significant regulons will be labeled. Default = 10.
logfc_thresh	Numeric threshold for labeling and annotating regulons based on logFC values. Default = 0.1.
lm	Logical indicating whether to plot a linear regression line for the logFC values of positive vs. negative regulons. Default = TRUE.
wgcna_name	The name of the WGCNA experiment in the seurat_obj@misc slot. Default is the active WGCNA experiment.

**Details**

This function generates a scatter plot where each point represents a TF regulon, with the x-axis showing the average log fold-change for positive regulon scores and the y-axis showing the average log fold-change for negative regulon scores. Points are colored by their module assignment, and the size of the points reflects the module eigengene correlation (kME) value of the TF.

Differentially expressed regulons can be highlighted, and the most significantly up- and down-regulated regulons can be labeled. Additional options allow adding a linear regression line and controlling label density based on logFC thresholds.

**Value**

A ggplot object representing the differential regulon scatter plot.

---

PlotDMEsLollipop

*PlotDMEsLollipop*


---

**Description**

Plotting function for the results of FindDMEs and FindAllDMEs

**Usage**

```
PlotDMEsLollipop(
  seurat_obj,
  DMEs,
  wgcna_name,
  group.by = NULL,
  comparison = NULL,
  pvalue,
  avg_log2FC = "avg_log2FC"
)
```

**Arguments**

seurat_obj	A Seurat object
DMEs	dataframe output from FindDMEs or FindAllDMEs
group.by	the column name of the selected comparison in the DMEs dataframe
comparison	character vector or a list of character vectors containing the comparison in the DMEs dataframe
pvalue	p_value or fdr used for the comparison in the DMEs dataframe

**Value**

A ggplot object PlotDMEsLollipop

---

PlotDMEsVolcano	<i>PlotDMEsVolcano</i>
-----------------	------------------------

---

**Description**

Plotting function for the results of FindDMEs and FindAllDMEs.

**Usage**

```
PlotDMEsVolcano(
  seurat_obj,
  DMEs,
  plot_labels = TRUE,
  mod_point_size = 4,
  label_size = 4,
  show_cutoff = TRUE,
  wgcna_name = NULL,
  xlim_range = NULL,
  ylim_range = NULL
)
```

**Arguments**

<code>seurat_obj</code>	A Seurat object containing the WGCNA analysis in the <code>@misc</code> slot.
<code>DMEs</code>	A dataframe output from <code>FindDMEs</code> or <code>FindAllDMEs</code> containing DME results.
<code>plot_labels</code>	Logical, determines whether to plot the module labels on the volcano plot. Default is <code>TRUE</code> .
<code>mod_point_size</code>	Numeric, the size of the points on the volcano plot. Default is 4.
<code>label_size</code>	Numeric, the size of the module labels on the plot. Default is 4.
<code>show_cutoff</code>	Logical, determines whether to plot the significance cutoff. Set this to <code>FALSE</code> if using <code>facet_wrap</code> . Default is <code>TRUE</code> .
<code>wgcna_name</code>	Character, the name of the hdWGCNA experiment in the <code>seurat_obj@misc</code> slot. Default is <code>NULL</code> , in which case it pulls the active WGCNA experiment from <code>seurat_obj@misc\$active_wgcna</code> .
<code>xlim_range</code>	A numeric vector of length 2 specifying the x-axis limits for the log2 fold change. Default is <code>NULL</code> , which automatically calculates limits based on the data.
<code>ylim_range</code>	A numeric vector of length 2 specifying the y-axis limits for the $-\log_{10}(\text{p-value})$ . Default is <code>NULL</code> , which automatically calculates limits based on the data.

**Details**

This function generates a volcano plot for differential module expression (DME) analysis results. It can handle both two-group comparisons (using the output of `FindDMEs`) and one-vs-all comparisons (using the output of `FindAllDMEs`).

**Value**

A ggplot object containing the volcano plot for the DME results.

**Examples**

```
# Example usage:
# Assuming `seurat_obj` is your Seurat object and `DMEs` is the output from FindDMEs
PlotDMEsVolcano(seurat_obj, DMEs, wgcna_name = "MG")
```

---

PlotKMEs

*PlotKMEs*


---

**Description**

Plotting function to show genes by kME value in each module

**Usage**

```
PlotKMEs(
  seurat_obj,
  n_hubs = 10,
  text_size = 2,
  ncol = 5,
  plot_widths = c(3, 2),
  wgcna_name = NULL
)
```

**Arguments**

seurat_obj	A Seurat object
n_hubs	number of hub genes to display
text_size	controls the size of the hub gene text
ncol	number of columns to display individual plots
plot_widths	the relative width between the kME rank plot and the hub gene text
wgcna_name	the name of the WGCNA experiment in the seurat object

---

 PlotLollipop

*PlotLollipop*


---

**Description**

An Internal function for PlotDMEsLollipop to Plotting function for the results of FindDMEs and FindAllDMEs

**Usage**

```
PlotLollipop(modules, cur_DMEs, pvalue, avg_log2FC = "avg_log2FC")
```

**Arguments**

pvalue	p_value or fdr used for the comparison in the DMEs dataframe
seurat_obj	A Seurat object
DMEs	dataframe output from FindDMEs or FindAllDMEs
group.by	the column name of the selected comparison in the DMEs dataframe
comparison	character vector or a list of character vectors containing the comparison in the DMEs dataframe

**Value**

A ggplot object

**Examples**

```
plot_list <- PlotLollipop(modules, DMEs, cur_title= c("Group1_vs_Group2"), pvalue, avg_log2FC = 'avg_log2FC')
```

---

PlotModulePreservation

*PlotModulePreservation*

---

## Description

Plotting function for Module Preservation statistics

## Usage

```
PlotModulePreservation(  
  seurat_obj,  
  name,  
  statistics = "summary",  
  plot_labels = TRUE,  
  label_size = 4,  
  mod_point_size = 4,  
  wgcna_name = NULL  
)
```

## Arguments

seurat_obj	A Seurat object
name	The name of the module preservation analysis to plot given in ModulePreservation
statistics	Which module preservation statistics to plot? Choices are summary, all, or a custom list
plot_labels	logical determining whether to plot the module labels
label_size	the size of the module labels
mod_point_size	the size of the points in each plot
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

## Details

This function creates a scatter plot showing the module preservation statistics for each module compared to the size of the module (number of genes).

---

PlotModulePreservationLollipop  
*PlotModulePreservationLollipop*

---

### Description

Plots a ranked lollipop plot of co-expression modules based on the results of module preservation analysis.

### Usage

```
PlotModulePreservationLollipop(  
  seurat_obj,  
  name,  
  features = NULL,  
  fdr = TRUE,  
  wgcna_name = NULL  
)
```

### Arguments

seurat_obj	A Seurat object
name	The name to give the module preservation analysis.
features	The name of the module preservation features to plot.
fdr	logical indicating whether or not to plot FDR-corrected p-values
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

### Details

PlotModulePreservationLollipop generates a lollipop plot showing module preservation results. If the module preservation test was performed using the WGCNA method, the statistic that will be shown is the Z-summary preservation statistic. If the analysis was performed using NetRep, then the statistic that will be shown is the FDR corrected averaged p-values from the module preservation permutation test.

### Value

ggplot object containing the PlotModulePreservationLollipop

---

PlotModuleTraitCorrelation

*PlotModuleTraitCorrelation*

---

## Description

Plotting function for Module Preservation statistics

## Usage

```
PlotModuleTraitCorrelation(  
  seurat_obj,  
  high_color = "red",  
  mid_color = "grey90",  
  low_color = "blue",  
  label = NULL,  
  label_symbol = "stars",  
  plot_max = NULL,  
  text_size = 2,  
  text_color = "black",  
  text_digits = 3,  
  combine = TRUE,  
  wgcna_name = NULL  
)
```

## Arguments

seurat_obj	A Seurat object
high_color	color for positive correlation
mid_color	color for zero correlation
low_color	color for negative correlation
label	logical determining whether to add p-val label in each cell of the heatmap
label_symbol	show the labels as 'stars' or as 'numeric'
plot_max	maximum value of correlation to show on the colorbar
text_size	size of the labels
text_color	color of the text labels
text_digits	how many digits to show in the text labels
combine	logical determining whether to plot as one combined plot (TRUE) or to return individual plots as a list (FALSE)

---

PlotModuleTrajectory *PlotModuleTrajectory*

---

### Description

Plots module eigengene dynamics over a pseudotime trajectory.

### Usage

```
PlotModuleTrajectory(  
  seurat_obj,  
  pseudotime_col = "pseudotime",  
  n_bins = 20,  
  harmonized = TRUE,  
  ncol = 4,  
  point_size = 1,  
  line_size = 1,  
  se = TRUE,  
  group_colors = NULL,  
  wgcna_name = NULL  
)
```

### Arguments

<code>seurat_obj</code>	A Seurat object
<code>pseudotime_col</code>	name of the column in <code>seurat_obj@meta.data</code> containing pseudotime information. Multiple names can be passed to plot multiple trajectories simultaneously.
<code>n_bins</code>	number of pseudotime bins/windows used to smooth the MEs
<code>harmonized</code>	logical determining whether or not to use the harmonized MEs
<code>point_size</code>	size of the points in the plot
<code>line_size</code>	width of the lines in the plot
<code>se</code>	logical determining whether or not to show the standard error on the plot
<code>group_colors</code>	optional list of colors to differentiate multiple pseudotime trajectories. Must be the same length as <code>pseudotime_col</code>
<code>wgcna_name</code>	The name of the hdWGCNA experiment in the <code>seurat_obj@misc</code> slot
<code>n_col</code>	number of columns for different modules in the plot

---

PlotSoftPowers	<i>PlotSoftPowers</i>
----------------	-----------------------

---

**Description**

Plot Soft Power Threshold results

**Usage**

```
PlotSoftPowers(
  seurat_obj,
  selected_power = NULL,
  point_size = 5,
  text_size = 3,
  plot_connectivity = TRUE,
  wgcna_name = NULL
)
```

**Arguments**

seurat_obj	A Seurat object
selected_power	power to highlight in the plots
point_size	the size of the points in the plot
text_size	the size of the text in the plot
plot_connectivity	logical indicating whether to plot the connectivity in addition to the scale free topology fit.
wgcna_name	The name of the WGCNA experiment in seurat_obj

**Examples**

```
PlotSoftPowers
```

---

ProjectModules	<i>ProjectModules</i>
----------------	-----------------------

---

**Description**

Project a set of co-expression modules from a reference to a query dataset

**Usage**

```
ProjectModules(
  seurat_obj,
  seurat_ref = NULL,
  modules = NULL,
  group.by.vars = NULL,
  gene_mapping = NULL,
  genome1_col = NULL,
  genome2_col = NULL,
  overlap_proportion = 0.5,
  vars.to.regress = NULL,
  scale.model.use = "linear",
  wgcna_name = NULL,
  wgcna_name_proj = NULL,
  ...
)
```

**Arguments**

<code>seurat_obj</code>	A Seurat object where the modules will be projected to
<code>seurat_ref</code>	A Seurat object containing the co-expression modules to be projected
<code>modules</code>	optionally provide a dataframe containing gene-module information to bypass <code>seurat_ref</code>
<code>group.by.vars</code>	groups to harmonize by
<code>gene_mapping</code>	dataframe to map gene names between genomes
<code>genome1_col</code>	column in <code>gene_mapping</code> containing the genes for <code>seurat_ref</code> (reference)
<code>genome2_col</code>	column in <code>gene_mapping</code> containing the genes for <code>seurat_obj</code> (query)
<code>overlap_proportion</code>	the proportion of genes that must be present in <code>seurat_obj</code> for a given module to be projected. Default = 0.5 (50% of genes)
<code>vars.to.regress</code>	character vector of variables in <code>seurat_obj@meta.data</code> to regress when running <code>ScaleData</code>
<code>scale.model.use</code>	model to scale data when running <code>ScaleData</code> choices are "linear", "poisson", or "negbinom"
<code>wgcna_name</code>	The name of the hdWGCNA experiment in the <code>seurat_ref@misc</code> slot
<code>wgcna_name_proj</code>	The name of the hdWGCNA experiment to be created for the projected modules in <code>seurat_obj</code>

**Examples**

```
ProjectModules
```

---

ReassignModules	<i>ReassignModules</i>
-----------------	------------------------

---

**Description**

Reassigns features to different modules

**Usage**

```
ReassignModules(
  seurat_obj,
  harmonized = TRUE,
  features = NULL,
  new_modules = NULL,
  ignore = FALSE,
  auto_reassign = FALSE,
  wgcna_name = NULL
)
```

**Arguments**

seurat_obj	A Seurat object
harmonized	logical indicating whether or not to use harmonized MEs
features	character vector containing features for manual module reassignment,
new_modules	character vector containing modules to reassign the genes
ignore	logical indicating whether or not to ignore error message about reassigning non-grey features
auto_reassign	logical, automatically re-assign selected features to the module with the highest kME?
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

**Details**

ReassignModules reassigns features with negative kMEs in their assigned module to the module that had the highest kME for that feature. Alternatively, this function can manually assign features to different modules, which can be helpful if certain genes of interest are assigned to the grey module. We generally do not advise reassigning features to new modules if they are in non-grey modules, but the user can do so at their own risk by setting ignore=TRUE.

**Value**

seurat\_obj with the an updated modules table for the selected wgcna experiment

---

RegulonBarPlot	<i>RegulonBarPlot</i>
----------------	-----------------------

---

### Description

Plots the top target genes within a specific TF regulon

### Usage

```
RegulonBarPlot(
  seurat_obj,
  selected_tf,
  cutoff = 0.2,
  top_n = Inf,
  TFs_only = FALSE,
  high_color = "orange2",
  mid_color = "white",
  low_color = "dodgerblue",
  wgcna_name = NULL
)
```

### Arguments

seurat_obj	A Seurat object
cutoff	Cutoff for the edge weights, links below this value will not be included.
top_n	Number of top and bottom target genes to include in the plot.
TFs_only	Logical indicating whether or not to use only TFs in the plot or to use TFs and other genes.
high_color	Color corresponding to positive edge weights
mid_color	Color corresponding to edge weights close to 0
low_color	Color corresponding to negative edge weights
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot
selected_tfs	A TF whose target genes will be shown

### Details

RegulonBarPlot creates a bar plot showing the top target genes for a particular TF based on the TF regulatory network analysis. Target genes are ranked by the predicted interaction strength from XGBoost (Gain) multiplied by the sign of the correlation between the TF and the target gene.

### Value

ggplot object containing the RegulonBarPlot

---

 RegulonScores

*RegulonScores*


---

### Description

Calculate expression scores for TF Regulons

### Usage

```
RegulonScores(
  seurat_obj,
  target_type = "positive",
  cor_thresh = 0.05,
  exclude_grey_genes = TRUE,
  wgcna_name = NULL,
  ...
)
```

### Arguments

<code>seurat_obj</code>	A Seurat object
<code>target_type</code>	Which types of TF target genes to compute scores for? "positive", "negative", or "both"
<code>cor_thresh</code>	threshold for TF-gene correlation for genes to be included in the regulon score
<code>exclude_grey_genes</code>	option to exclude genes that are in the grey module from the regulon scores
<code>wgcna_name</code>	name of the WGCNA experiment

### Details

RegulonScores calculates expression signatures for each TF regulon using the UCell algorithm. This function can calculate separate scores for TF regulons for target genes that are positively or negatively correlated with the TF, representing putative activated or repressed genes. These scores conveniently summarize the expression levels of the entire TF regulon, similar to module eigengenes for the co-expression network analysis.

### Value

`seurat_obj` with the TF Regulon Scores added

---

ResetModuleColors	<i>ResetModuleColors</i>
-------------------	--------------------------

---

**Description**

Reset the unique color for each hdWGCNA module

**Usage**

```
ResetModuleColors(seurat_obj, new_colors, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
new_colors	a character vector containing the new colors
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

ResetModuleNames	<i>ResetModuleNames</i>
------------------	-------------------------

---

**Description**

Reset the unname of each hdWGCNA module

**Usage**

```
ResetModuleNames(  
  seurat_obj,  
  new_name = "M",  
  reset_levels = FALSE,  
  wgcna_name = NULL  
)
```

**Arguments**

seurat_obj	A Seurat object
new_name	string containing the base name to re-name the modules
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

 RunEnrichr

*RunEnrichr*


---

### Description

Run Enrichr gene set enrichment tests on hdWGCNA modules

### Usage

```
RunEnrichr(
  seurat_obj,
  dbs = c("GO_Biological_Process_2021", "GO_Cellular_Component_2021",
    "GO_Molecular_Function_2021"),
  max_genes = 100,
  wait = TRUE,
  wait_time = 5,
  wgcna_name = NULL,
  ...
)
```

### Arguments

seurat_obj	A Seurat object
dbs	character vector of EnrichR databases
max_genes	Max number of genes to include per module, ranked by kME.
wait	logical indicating whether or not to wait some time between sending requests to the EnrichR server.
wait_time	the number of seconds to wait between sending requests to the EnrichR server. Value must be less than 60.
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

 RunEnrichrRegulons

*RunEnrichrRegulons*


---

### Description

Run Enrichr gene set enrichment tests on hdWGCNA modules

**Usage**

```
RunEnrichRegulons(
  seurat_obj,
  dbs = c("GO_Biological_Process_2021", "GO_Cellular_Component_2021",
    "GO_Molecular_Function_2021"),
  depth = 1,
  use_regulons = TRUE,
  min_genes = 5,
  wait = TRUE,
  wait_time = 5,
  wgcna_name = NULL,
  ...
)
```

**Arguments**

seurat_obj	A Seurat object
dbs	character vector of EnrichR databases
depth	Include primary TF target genes (depth=1) or primary + secondary (depth=2)
use_regulons	Use the regulons (default=TRUE) or the full set of TF target genes?
wait	logical indicating whether or not to wait some time between sending requests to the EnrichR server.
wait_time	the number of seconds to wait between sending requests to the EnrichR server. Value must be less than 60.
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot
max_genes	Max number of genes to include per module, ranked by kME.

---

RunHarmonyMetacells    *RunHarmonyMetacells*

---

**Description**

Wrapper function to run harmony's RunHarmony function on the metacell object.

**Usage**

```
RunHarmonyMetacells(seurat_obj, ...)
```

**Arguments**

seurat_obj	A Seurat object
------------	-----------------

---

RunModuleUMAP	<i>RunModuleUMAP</i>
---------------	----------------------

---

**Description**

Run UMAP on co-expression matrix using hub genes as features.

**Usage**

```
RunModuleUMAP(
  seurat_obj,
  n_hubs = 10,
  exclude_grey = TRUE,
  genes_use = NULL,
  wgcna_name = NULL,
  n_neighbors = 25,
  metric = "cosine",
  spread = 1,
  min_dist = 0.4,
  supervised = FALSE,
  ...
)
```

**Arguments**

seurat_obj	A Seurat object
n_hubs	number of hub genes to use in the UMAP computation
exclude_grey	logical indicating whether to include grey module
genes_use	character vector of genes to use for the UMAP, must already be present in GetModules(seurat_obj)
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot
...	Additional parameters supplied to uwot::umap

---

RunPCAMetacells	<i>RunPCAMetacells</i>
-----------------	------------------------

---

**Description**

Wrapper function to run Seurat's RunPCA function on the metacell object.

**Usage**

```
RunPCAMetacells(seurat_obj, wgcna_name = NULL, ...)
```

**Arguments**

seurat_obj	A Seurat object
------------	-----------------

---

RunUMAPMetacells	<i>RunUMAPMetacells</i>
------------------	-------------------------

---

**Description**

Wrapper function to run Seurat's RunUMAP function on the metacell object.

**Usage**

```
RunUMAPMetacells(seurat_obj, ...)
```

**Arguments**

seurat_obj	A Seurat object
------------	-----------------

---

scale01	<i>scale01</i>
---------	----------------

---

**Description**

Function to scale a numeric vector between 0 and 1.

**Usage**

```
scale01(x)
```

**Arguments**

x	numeric vector
---	----------------

---

ScaleMetacells	<i>ScaleMetacells</i>
----------------	-----------------------

---

**Description**

Wrapper function to run Seurat's ScaleData function on the metacell object.

**Usage**

```
ScaleMetacells(seurat_obj, wgcna_name = NULL, ...)
```

**Arguments**

seurat_obj	A Seurat object
------------	-----------------

---

 SelectNetworkGenes      *SelectNetworkGenes*


---

## Description

Select genes that will be used for co-expression network analysis

## Usage

```
SelectNetworkGenes(
  seurat_obj,
  gene_select = "variable",
  fraction = 0.05,
  group.by = NULL,
  gene_list = NULL,
  assay = NULL,
  wgcna_name = NULL
)
```

## Arguments

seurat_obj	A Seurat object
fraction	A numeric that determines the minimum cells that a gene must be expressed in order to be included. For example, fraction = 0.05 means that 5% of cells must express a gene (count > 0) for it to be included.
gene_list	A character string of gene names, only used if type = "custom"
assay	Assay in seurat_obj to compute module eigengenes. Default is DefaultAssay(seurat_obj)
wgcna_name	name of the WGCNA experiment
type	How to select genes? Select "variable", "fraction", "all", or "custom".

## Details

SelectNetworkGenes allows us to specify the genes that will be used for co-expression network analysis. This function is called by SetupForWGCNA. By default, the variable features in VariableFeatures(seurat\_obj) are used. A custom gene list can also be used with the gene\_list parameter and setting gene\_select="custom".

We can also identify genes that are expressed above 0 in a certain proportion of the dataset by setting gene\_select='fraction'. For example, by setting fraction=0.1 and group.by='seurat\_clusters', this function will identify the set of genes that are expressed in 10% of cells in at least one of the clusters.

---

SetActiveWGCNA	<i>SetActiveWGCNA</i>
----------------	-----------------------

---

**Description**

SetActiveWGCNA

**Usage**

```
SetActiveWGCNA(seurat_obj, wgcna_name)
```

**Arguments**

seurat_obj	A Seurat object
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

SetAvgModuleExpr	<i>SetAvgModuleExpr</i>
------------------	-------------------------

---

**Description**

SetAvgModuleExpr

**Usage**

```
SetAvgModuleExpr(seurat_obj, avg_mods, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
avg_mods	dataframe storing the average expression of all genes in the same module
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

SetDatExpr

*SetDatExpr***Description**

This function specifies the gene expression matrix for co-expression network analysis.

**Usage**

```
SetDatExpr(
  seurat_obj,
  group_name,
  use_metacells = TRUE,
  group.by = NULL,
  multi.group.by = NULL,
  multi_group_name = NULL,
  return_seurat = TRUE,
  assay = NULL,
  slot = "data",
  layer = "data",
  mat = NULL,
  features = NULL,
  wgcna_name = NULL,
  ...
)
```

**Arguments**

<code>seurat_obj</code>	A Seurat object
<code>group_name</code>	A string containing a group present in the provided <code>group.by</code> column or in the Seurat Idents. A character vector can be provided to select multiple groups at a time.
<code>use_metacells</code>	A logical determining if we use the metacells (TRUE) or the full expression matrix (FALSE)
<code>group.by</code>	A string containing the name of a column in the Seurat object with cell groups (clusters, cell types, etc). If NULL (default), <code>hdWGCNA</code> uses the Seurat Idents as the group.
<code>multi.group.by</code>	A string containing the name of a column in the Seurat object with groups for consensus WGCNA (dataset, sample, condition, etc)
<code>multi_group_name</code>	A string containing the name of a group present in the <code>multi.group.by</code> column.
<code>assay</code>	The name of the assay in the Seurat object
<code>slot</code>	Slot to extract data for aggregation. Default = 'counts'. Slot is used with Seurat v4 instead of layer.

layer	Layer to extract data for aggregation. Default = 'counts'. Layer is used with Seurat v5 instead of slot.
mat	A Matrix containing gene expression data. Supplying a matrix using this parameter ignores other options. This is almost exclusively used for pseudobulk analysis.
features	A list of features to use to override the features that have been previously set.
wgcna_name	A string containing the name of the WGCNA slot in <code>seurat_obj@misc</code> . Default = NULL which retrieves the currently active WGCNA data

### Details

SetDatExpr is a critical function of the hdWGCNA pipeline that determines the gene expression matrix that will be used for network analysis. We typically use this function to select a cell type or group of cell types for network analysis using the `group.by` parameter, but we provide additional parameters for further customization.

---

SetDegrees	<i>SetDegrees</i>
------------	-------------------

---

### Description

SetDegrees

### Usage

```
SetDegrees(seurat_obj, degree_df, wgcna_name = NULL)
```

### Arguments

seurat_obj	A Seurat object
degree_df	dataframe containing gene module assignments
wgcna_name	The name of the hdWGCNA experiment in the <code>seurat_obj@misc</code> slot

---

SetEnrichRegulonTable	<i>SetEnrichRegulonTable</i>
-----------------------	------------------------------

---

### Description

SetEnrichRegulonTable

### Usage

```
SetEnrichRegulonTable(seurat_obj, enrich_table, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
enrich_table	dataframe storing the results of running enrichr
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

SetEnrichrTable	<i>SetEnrichrTable</i>
-----------------	------------------------

---

**Description**

SetEnrichrTable

**Usage**

```
SetEnrichrTable(seurat_obj, enrich_table, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
enrich_table	dataframe storing the results of running enrichr
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

SetMELoadings	<i>SetMELoadings</i>
---------------	----------------------

---

**Description**

SetMELoadings

**Usage**

```
SetMELoadings(seurat_obj, loadings, harmonized = TRUE, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
loadings	named numeric vector with eigengene loadings
harmonized	logical indicating whether MEs have been harmonized
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

SetMEs	<i>SetMEs</i>
--------	---------------

---

**Description**

SetMEs

**Usage**

```
SetMEs(seurat_obj, MEs, harmonized = TRUE, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
MEs	dataframe or matrix containing module eigengenes
harmonized	logical indicating whether MEs have been harmonized
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

SetMetacellObject	<i>SetMetacellObject</i>
-------------------	--------------------------

---

**Description**

SetMetacellObject

**Usage**

```
SetMetacellObject(seurat_obj, metacell_obj, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
metacell_obj	metacell Seurat object
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

SetMetacellParams      *SetMetacellParams*

---

### Description

SetMetacellParams

### Usage

```
SetMetacellParams(seurat_obj, params, wgcna_name = NULL)
```

### Arguments

seurat_obj	A Seurat object
params	list of WGCNA parameters
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

SetModulePreservation      *SetModulePreservation*

---

### Description

SetModulePreservation

### Usage

```
SetModulePreservation(seurat_obj, mod_pres, mod_name, wgcna_name = NULL)
```

### Arguments

seurat_obj	A Seurat object
mod_name	name of the module preservation test to store
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot
mt_cor	matrix of module-trait correlation results

---

SetModules	<i>SetModules</i>
------------	-------------------

---

**Description**

SetModules

**Usage**

```
SetModules(seurat_obj, modules, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
modules	dataframe containing gene module assignments
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

SetModuleScores	<i>SetModuleScores</i>
-----------------	------------------------

---

**Description**

SetModuleScores

**Usage**

```
SetModuleScores(seurat_obj, mod_scores, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
mod_scores	dataframe storing the module expression scores
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

SetModuleTraitCorrelation  
*SetModuleTraitCorrelation*

---

**Description**

SetModuleTraitCorrelation

**Usage**

```
SetModuleTraitCorrelation(seurat_obj, mt_cor, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
mt_cor	matrix of module-trait correlation results
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

SetModuleUMAP                    *SetModuleUMAP*

---

**Description**

SetModuleUMAP

**Usage**

```
SetModuleUMAP(seurat_obj, umap_df, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
umap_df	dataframe of UMAP coordinates
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

SetMotifMatrix	<i>SetMotifMatrix</i>
----------------	-----------------------

---

**Description**

SetMotifMatrix

**Usage**

```
SetMotifMatrix(seurat_obj, tf_match)
```

**Arguments**

seurat_obj	A Seurat object
tf_match	matrix containing tf-promoter matches

**Examples**

```
SetMotifMatrix
```

---

SetMotifOverlap	<i>SetMotifOverlap</i>
-----------------	------------------------

---

**Description**

SetMotifOverlap

**Usage**

```
SetMotifOverlap(seurat_obj, overlap_df, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
overlap_df	dataframe containing motif-module overlap info
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

SetMotifs

*SetMotifs*

---

### Description

SetMotifs

### Usage

```
SetMotifs(seurat_obj, motif_df)
```

### Arguments

seurat_obj	A Seurat object
motif_df	dataframe containing info about the motifs being analyzed

### Examples

```
SetMotifs
```

---

SetMotifScores

*SetMotifScores*

---

### Description

SetMotifScores

### Usage

```
SetMotifScores(seurat_obj, tf_scores, wgcna_name = NULL)
```

### Arguments

seurat_obj	A Seurat object
tf_scores	dataframe of tf motif target scores
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

SetMotifTargets	<i>SetMotifTargets</i>
-----------------	------------------------

---

**Description**

SetMotifTargets

**Usage**

```
SetMotifTargets(seurat_obj, motif_targets)
```

**Arguments**

seurat_obj	A Seurat object
motif_targets	list of motifs and their target genes

**Examples**

```
SetMotifTargets
```

---

SetMultiExpr	<i>SetMultiExpr</i>
--------------	---------------------

---

**Description**

This function sets up the expression matrix input for consensus WGCNA based on the metacell expression matrix, the full expression matrix, or a provided pseudobulk expression matrix.

**Usage**

```
SetMultiExpr(  
  seurat_obj,  
  group_name,  
  use_metacells = TRUE,  
  group.by = NULL,  
  multi.group.by = NULL,  
  multi_groups = NULL,  
  assay = NULL,  
  slot = "data",  
  layer = "data",  
  mat = NULL,  
  mat_group_delim = 3,  
  wgcna_name = NULL,  
  ...  
)
```

**Arguments**

seurat_obj	A Seurat object
group_name	A string containing a group present in the provided group.by column or in the Seurat Idents.
use_metacells	A logical determining if we use the metacells (TRUE) or the full expression matrix (FALSE)
group.by	A string containing the name of a column in the Seurat object with cell groups (clusters, cell types, etc). If NULL (default), hdWGCNA uses the Seurat Idents as the group.
multi.group.by	A string containing the name of a column in the Seurat object with groups for consensus WGCNA (dataset, sample, condition, etc)
multi_groups	A character vector containing the names of groups to select
assay	The name of the assay in the Seurat object
slot	The name of the slot in the Seurat object (counts, data)
layer	Layer to extract data for aggregation. Default = 'counts'. Layer is used with Seurat v5 instead of slot.
mat	A Matrix containing gene expression data. Supplying a matrix using this parameter ignores other options. This is almost exclusively used for pseudobulk analysis.
wgcna_name	A string containing the name of the WGCNA slot in seurat_obj@misc. Default = NULL which retrieves the currently active WGCNA data

---

SetNetworkData

*SetNetworkData*


---

**Description**

SetNetworkData

**Usage**

SetNetworkData(seurat\_obj, net, wgcna\_name = NULL)

**Arguments**

seurat_obj	A Seurat object
net	list of network data from WGCNA
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

`SetPFMList`*SetPFMList*

---

**Description**`SetPFMList`**Usage**`SetPFMList(seurat_obj, pfm_list)`**Arguments**

<code>seurat_obj</code>	A Seurat object
<code>pfm_list</code>	list of pfm objects

**Examples**`SetPFMList`

---

`SetPowerTable`*SetPowerTable*

---

**Description**`SetPowerTable`**Usage**`SetPowerTable(seurat_obj, power_table, wgcna_name = NULL)`**Arguments**

<code>seurat_obj</code>	A Seurat object
<code>power_table</code>	a dataframe containing the results of the soft power test
<code>wgcna_name</code>	The name of the hdWGCNA experiment in the <code>seurat_obj@misc</code> slot

---

SetRegulonScores	<i>SetRegulonScores</i>
------------------	-------------------------

---

### Description

SetRegulonScores

### Usage

```
SetRegulonScores(seurat_obj, regulon_scores, target_type, wgcna_name = NULL)
```

### Arguments

seurat_obj	A Seurat object
regulon_scores	dataframe storing the TF regulon scores
target_type	dataframe storing the TF regulon scores
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

SetTFEval	<i>SetTFEval</i>
-----------	------------------

---

### Description

SetTFEval

### Usage

```
SetTFEval(seurat_obj, tf_eval, wgcna_name = NULL)
```

### Arguments

seurat_obj	A Seurat object
tf_eval	dataframe storing the TF network evaluation info from ConstructTFNetwork
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

SetTFNetwork	<i>SetTFNetwork</i>
--------------	---------------------

---

**Description**

SetTFNetwork

**Usage**

```
SetTFNetwork(seurat_obj, tf_net, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
tf_net	dataframe storing the TF network info in ConstructTFNetwork
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

SetTFRegulons	<i>SetTFRegulons</i>
---------------	----------------------

---

**Description**

SetTFRegulons

**Usage**

```
SetTFRegulons(seurat_obj, tf_regulons, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
tf_regulons	dataframe storing the TF regulon info from AssignTFRegulons
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

 SetupForWGCNA

*SetupForWGCNA*


---

### Description

Create a slot in a Seurat object to store hdWGCNA data

### Usage

```
SetupForWGCNA(
  seurat_obj,
  wgcna_name,
  features = NULL,
  metacell_location = NULL,
  ...
)
```

### Arguments

<code>seurat_obj</code>	A Seurat object
<code>wgcna_name</code>	name of the WGCNA experiment
<code>features</code>	list of features to use for WGCNA
<code>metacell_location</code>	name of the WGCNA experiment to copy the metacell object from
<code>...</code>	additional parameters to pass to <code>SelectNetworkGenes</code>

### Details

`SetupForWGCNA` creates a new slot in the Seurat object (`seurat_obj`) to store an hdWGCNA experiment with a given name (`wgcna_name`). This function calls on `SelectNetworkGenes` to specify which features will be used for network analysis. If there is another hdWGCNA experiment already in the `seurat_obj`, the same `metacell/metaspot` object can be used by specifying the name of the hdWGCNA experiment to the `metacell_location` parameter.

---

 SetWGCNAGenes

*SetWGCNAGenes*


---

### Description

SetWGCNAGenes

### Usage

```
SetWGCNAGenes(seurat_obj, gene_list, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
gene_list	vector of genes to be used for WGCNA
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

SetWGCNAParams	<i>SetWGCNAParams</i>
----------------	-----------------------

---

**Description**

SetWGCNAParams

**Usage**

```
SetWGCNAParams(seurat_obj, params, wgcna_name = NULL)
```

**Arguments**

seurat_obj	A Seurat object
params	list of WGCNA parameters
wgcna_name	The name of the hdWGCNA experiment in the seurat_obj@misc slot

---

shuffle_points	<i>shuffle_points</i>
----------------	-----------------------

---

**Description**

Function to shuffle the rows of a dataframe.

**Usage**

```
shuffle_points(df)
```

**Arguments**

df	dataframe
----	-----------

---

TestSoftPowers	<i>TestSoftPowers</i>
----------------	-----------------------

---

### Description

Compute the scale-free topology model fit for different soft power thresholds

### Usage

```
TestSoftPowers(
  seurat_obj,
  powers = c(seq(1, 10, by = 1), seq(12, 30, by = 2)),
  networkType = "signed",
  corFnc = "bicor",
  wgcna_name = NULL,
  ...
)
```

### Arguments

seurat_obj	A Seurat object
powers	numeric vector specifying soft powers to test
networkType	The type of network to use for network analysis. Options are "signed" (default), "unsigned", or "signed hybrid". This should be consistent with the network chosen for ConstructNetwork
corFnc	Correlation function for the gene-gene correlation adjacency matrix.
wgcna_name	name of the WGCNA experiment
...	additional parameters passed to WGCNA::pickSoftThreshold

### Details

TestSoftPowers aims to find the optimal value of the "soft-power threshold" for co-expression network analysis. For varying values of this `soft_power` parameter, this function assesses the scale-free topology model fit of the resulting network and the function also calculates other network statistics such as the connectivity. The user may also vary the type of network used and the correlation function. For downstream functions, the lowest `soft_power` that reaches a fit of 0.8 is selected for network construction, but the user may select their own values.

---

 TestSoftPowersConsensus

*TestSoftPowersConsensus*


---

### Description

Compute the scale-free topology model fit for different soft power thresholds separately for each input dataset

### Usage

```
TestSoftPowersConsensus(
  seurat_obj,
  powers = c(seq(1, 10, by = 1), seq(12, 30, by = 2)),
  use_metacells = TRUE,
  networkType = "signed",
  corFnc = "bicolor",
  setDatExpr = FALSE,
  group.by = NULL,
  group_name = NULL,
  multi.group.by = NULL,
  multi_groups = NULL,
  wgcna_name = NULL,
  ...
)
```

### Arguments

<code>seurat_obj</code>	A Seurat object
<code>powers</code>	numeric vector specifying soft powers to test
<code>use_metacells</code>	logical flag for whether to use the metacell expression matrix
<code>networkType</code>	The type of network to use for network analysis. Options are "signed" (default), "unsigned", or "signed hybrid". This should be consistent with the network chosen for ConstructNetwork
<code>corFnc</code>	Correlation function for the gene-gene correlation adjacency matrix.
<code>setDatExpr</code>	logical flag indicating whether to run setDatExpr.
<code>group.by</code>	A string containing the name of a column in the Seurat object with cell groups (clusters, cell types, etc). If NULL (default), hdWGCNA uses the Seurat Idents as the group.
<code>group_name</code>	A string containing a group present in the provided group.by column or in the Seurat Idents. A character vector can be provided to select multiple groups at a time.
<code>multi.group.by</code>	A string containing the name of a column in the Seurat object with groups for consensus WGCNA (dataset, sample, condition, etc)
<code>multi_groups</code>	A character vector containing the names of groups to select
<code>...</code>	additional parameters passed to SetDatExpr

**Examples**

```
# TestSoftPowers(pbmc)
```

---

TFNetworkPlot

*TFNetworkPlot*


---

**Description**

Plots a network of TFs and predicted target genes.

**Usage**

```
TFNetworkPlot(
  seurat_obj,
  selected_tfs,
  depth = 2,
  edge_weight = "Cor",
  cutoff = 0.01,
  color_cutoff = 0.75,
  target_type = "both",
  use_regulons = TRUE,
  label_genes = NULL,
  label_TFs = 1,
  no_labels = FALSE,
  TFs_only = FALSE,
  high_color = "orange2",
  mid_color = "white",
  low_color = "dodgerblue",
  node_colors = c("black", "darkorchid4", "mediumpurple2"),
  wgcna_name = NULL
)
```

**Arguments**

seurat_obj	A Seurat object
selected_tfs	A list of TFs
depth	Number of layers to extend the TF network from the selected_tfs. For example, if depth=2 (default), the target genes of selected_tfs are shown, and additional target genes are shown for other TFs that are target genes of the original selected_tfs.
edge_weight	Attribute to use to color the network edges. "Cor" to show the pearson correlation, or "Gain" to show the importance score from the XGBoost model.
cutoff	Cutoff for the edge weights, links below this value will not be included.
color_cutoff	Maximum value for the colorscale for the edge weights.

target_type	Type of target genes to show in the network. "Positive" shows genes with expression positively correlated with selected_tfs, "Negative" shows genes that are negatively correlated, and "Both" shows both (default).
use_regulons	Logical indicating whether to use the TF-gene links defined in the TF Regulons (TRUE, default), or all putative TF-gene links (FALSE).
label_genes	List of target genes to label in the network plot.
label_TFs	The depth level in the network to plot the names of TFs.
no_labels	Logical indicating whether or not to remove all labels.
TFs_only	Logical indicating whether or not to use only TFs in the plot or to use TFs and other genes.
high_color	Color corresponding to positive edge weights
mid_color	Color corresponding to edge weights close to 0
low_color	Color corresponding to negative edge weights
node_colors	List of color names for the TFs and genes at each depth. For example, if depth=2, you should supply a list of 3 colors corresponding to the selected_tfs (depth 0), the primary target genes (depth 1), and the secondary target genes (depth 2).
wgcna_name	The name of the hdWGCNA experiment in the <code>seurat_obj@misc</code> slot

### Details

TFNetworkPlot plots a network of TFs and their predicted target genes, based on the results from ConstructTFNetwork.

### Value

ggplot object containing the TFNetworkPlot

---

to_pseudobulk	<i>Create a pseudobulk matrix</i>
---------------	-----------------------------------

---

### Description

Convert a single-cell expression matrix (i.e., genes by cells) to a pseudobulk matrix by summarizing counts within biological replicates. This function is

### Usage

```
to_pseudobulk(
  input,
  meta = NULL,
  replicate_col = "replicate",
  cell_type_col = "cell_type",
  label_col = "label",
```

```

    min_cells = 3,
    min_reps = 2,
    min_features = 0,
    external = T
  )

```

### Arguments

<code>input</code>	a single-cell matrix to be converted, with features (genes) in rows and cells in columns. Alternatively, a Seurat, monocle3, or SingleCellExperiment object can be directly input.
<code>meta</code>	the accompanying meta data whereby the rownames match the column names of input.
<code>replicate_col</code>	the vector in meta containing the replicate information. Defaults to replicate.
<code>cell_type_col</code>	the vector in meta containing the cell type information. Defaults to cell_type.
<code>label_col</code>	the vector in meta containing the experimental label. Defaults to label.
<code>min_cells</code>	the minimum number of cells in a cell type to retain it. Defaults to 3.
<code>min_reps</code>	the minimum number of replicates in a cell type to retain it. Defaults to 2.
<code>min_features</code>	the minimum number of expressing cells (or replicates) for a gene to retain it. Defaults to 0.

### Value

a list of pseudobulk matrices, for each cell type.

---

TransferModuleGenome *TransferModuleGenome*

---

### Description

Takes a module table and a gene mapping table (like from biomaRt) with gene names from two genomes in order to switch

### Usage

```

TransferModuleGenome(
  modules,
  gene_mapping,
  genome1_col = NULL,
  genome2_col = NULL
)

```

**Arguments**

- modules            module table from GetModules function
- gene\_mapping     table that has the gene names for both genomes
- genome1\_col      the column in gene\_mapping that has gene names for the genome currently present in modules
- genome2\_col      the column in gene\_mapping that has gene names for the genome to transfer to

**Examples**

TransferModuleGenome

---

umap\_theme                    *umap\_theme*

---

**Description**

ggplot theme to remove axes etc.

**Usage**

umap\_theme()

# Index

- \* **WGCNA**
  - PlotDMesVolcano, 69
- \* **differential**
  - PlotDMesVolcano, 69
- \* **expression,**
  - PlotDMesVolcano, 69
- \* **helper**
  - scale01, 85
  - shuffle\_points, 103
  - umap\_theme, 109
- \* **plot,**
  - PlotDMesVolcano, 69
- \* **scRNA-seq,**
  - PlotDMesVolcano, 69
- \* **scRNA-seq**
  - CheckWGCNAName, 8
  - DimPlotMetacells, 15
  - DoHubGeneHeatmap, 15
  - FindAllDMes, 19
  - FindDMes, 21
  - GetActiveWGCNA, 23
  - GetActiveWGCNAName, 23
  - GetAvgModuleExpr, 24
  - GetDatExpr, 24
  - GetEnrichrRegulonTable, 25
  - GetEnrichrTable, 25
  - GetHubGenes, 26
  - GetMELoadings, 26
  - GetMEs, 27
  - GetMetacellObject, 27
  - GetMetacellParams, 28
  - GetModulePreservation, 28
  - GetModuleTraitCorrelation, 30
  - GetModuleUMAP, 30
  - GetMotifMatrix, 31
  - GetMotifOverlap, 31
  - GetMotifs, 32
  - GetMotifScores, 32
  - GetMotifTargets, 33
  - GetMultiExpr, 33
  - GetNetworkData, 34
  - GetPFMList, 34
  - GetPowerTable, 35
  - GetRegulonScores, 35
  - GetTFEval, 36
  - GetTFNetwork, 36
  - GetTFRegulons, 36
  - GetTOM, 38
  - GetWGCNAGenes, 38
  - GetWGCNAParams, 39
  - HubGeneNetworkPlot, 39
  - ModuleCorrelogram, 44
  - ModuleCorrNetwork, 45
  - ModuleFeaturePlot, 47
  - ModuleTraitCorrelation, 60
  - ModuleUMAPPlot, 61
  - MotifOverlapBarPlot, 62
  - OverlapBarPlot, 64
  - OverlapDotPlot, 65
  - OverlapModulesDEGs, 65
  - OverlapModulesMotifs, 66
  - PlotDendrogram, 67
  - PlotDMesLollipop, 68
  - PlotKMEs, 70
  - PlotLollipop, 71
  - PlotModuleTraitCorrelation, 74
  - PlotSoftPowers, 76
  - ProjectModules, 76
  - ResetModuleColors, 81
  - ResetModuleNames, 81
  - RunEnrichr, 82
  - RunUMAPMetacells, 85
  - SetActiveWGCNA, 87
  - SetDatExpr, 88
  - SetEnrichrRegulonTable, 89
  - SetEnrichrTable, 90
  - SetMELoadings, 90
  - SetMEs, 91

- SetMetacellObject, 91
- SetMetacellParams, 92
- SetModulePreservation, 92
- SetModules, 93
- SetModuleTraitCorrelation, 94
- SetModuleUMAP, 94
- SetMotifMatrix, 95
- SetMotifOverlap, 95
- SetMotifs, 96
- SetMotifScores, 96
- SetMotifTargets, 97
- SetMultiExpr, 97
- SetNetworkData, 98
- SetPFMList, 99
- SetPowerTable, 99
- SetRegulonScores, 100
- SetTFEval, 100
- SetTFNetwork, 101
- SetTFRegulons, 101
- SetWGCNAGenes, 102
- SetWGCNAParams, 103
- TestSoftPowersConsensus, 105
- TransferModuleGenome, 108
- \* **volcano**
  - PlotDMEsVolcano, 69
- AssignTFRegulons, 5
- AvgModuleExpr, 6
- BinPseudotime, 6
- check\_inputs, 7
- CheckSeurat5, 7
- CheckWGCNAName, 8
- ComputeModuleEigengene, 8
- ConstructMetacells, 9
- ConstructMetaspots, 10
- ConstructNetwork, 11
- ConstructPseudobulk, 12
- ConstructTFNetwork, 13
- corSparse, 14
- DimPlotMetacells, 15
- DoHubGeneHeatmap, 15
- EnrichrBarPlot, 16
- EnrichrDotPlot, 17
- FindAllDMEs, 19
- FindDifferentialRegulons, 20
- FindDMEs, 21
- FindMajorIsoforms, 22
- GetActiveWGCNA, 23
- GetActiveWGCNAName, 23
- GetAvgModuleExpr, 24
- GetDatExpr, 24
- GetDegrees, 24
- GetEnrichrRegulonTable, 25
- GetEnrichrTable, 25
- GetHubGenes, 26
- GetMELoadings, 26
- GetMEs, 27
- GetMetacellObject, 27
- GetMetacellParams, 28
- GetModulePreservation, 28
- GetModules, 29
- GetModuleScores, 29
- GetModuleTraitCorrelation, 30
- GetModuleUMAP, 30
- GetMotifMatrix, 31
- GetMotifOverlap, 31
- GetMotifs, 32
- GetMotifScores, 32
- GetMotifTargets, 33
- GetMultiExpr, 33
- GetNetworkData, 34
- GetPFMList, 34
- GetPowerTable, 35
- GetRegulonScores, 35
- GetTFEval, 36
- GetTFNetwork, 36
- GetTFRegulons, 36
- GetTFTargetGenes, 37
- GetTOM, 38
- GetWGCNAGenes, 38
- GetWGCNAParams, 39
- HubGeneNetworkPlot, 39
- MetacellsByGroups, 40
- MetaspotsByGroups, 42
- ModuleConnectivity, 42
- ModuleCorrelogram, 44
- ModuleCorrNetwork, 45
- ModuleEigengenes, 45
- ModuleExprScore, 47
- ModuleFeaturePlot, 47
- ModuleNetworkPlot, 48

ModulePreservation, [49](#)  
 ModulePreservationNetRep, [50](#)  
 ModuleRadarPlot, [52](#)  
 ModuleRegulatoryHeatmap, [53](#)  
 ModuleRegulatoryNetwork, [54](#)  
 ModuleRegulatoryNetworkPlot, [55](#)  
 ModuleTopologyBarplot, [57](#)  
 ModuleTopologyHeatmap, [58](#)  
 ModuleTraitCorrelation, [60](#)  
 ModuleUMAPPlot, [61](#)  
 MotifOverlapBarPlot, [62](#)  
 MotifScan, [62](#)  
  
 NormalizeMetacells, [64](#)  
  
 OverlapBarPlot, [64](#)  
 OverlapDotPlot, [65](#)  
 OverlapModulesDEGs, [65](#)  
 OverlapModulesMotifs, [66](#)  
  
 PlotDendrogram, [67](#)  
 PlotDifferentialRegulons, [67](#)  
 PlotDMEsLollipop, [68](#)  
 PlotDMEsVolcano, [69](#)  
 PlotKMEs, [70](#)  
 PlotLollipop, [71](#)  
 PlotModulePreservation, [72](#)  
 PlotModulePreservationLollipop, [73](#)  
 PlotModuleTraitCorrelation, [74](#)  
 PlotModuleTrajectory, [75](#)  
 PlotSoftPowers, [76](#)  
 ProjectModules, [76](#)  
  
 ReassignModules, [78](#)  
 RegulonBarPlot, [79](#)  
 RegulonScores, [80](#)  
 ResetModuleColors, [81](#)  
 ResetModuleNames, [81](#)  
 RunEnrichr, [82](#)  
 RunEnrichrRegulons, [82](#)  
 RunHarmonyMetacells, [83](#)  
 RunModuleUMAP, [84](#)  
 RunPCAMetacells, [84](#)  
 RunUMAPMetacells, [85](#)  
  
 scale01, [85](#)  
 ScaleMetacells, [85](#)  
 SelectNetworkGenes, [86](#)  
 SetActiveWGCNA, [87](#)  
 SetAvgModuleExpr, [87](#)  
 SetDatExpr, [88](#)  
 SetDegrees, [89](#)  
 SetEnrichrRegulonTable, [89](#)  
 SetEnrichrTable, [90](#)  
 SetMELoadings, [90](#)  
 SetMEs, [91](#)  
 SetMetacellObject, [91](#)  
 SetMetacellParams, [92](#)  
 SetModulePreservation, [92](#)  
 SetModules, [93](#)  
 SetModuleScores, [93](#)  
 SetModuleTraitCorrelation, [94](#)  
 SetModuleUMAP, [94](#)  
 SetMotifMatrix, [95](#)  
 SetMotifOverlap, [95](#)  
 SetMotifs, [96](#)  
 SetMotifScores, [96](#)  
 SetMotifTargets, [97](#)  
 SetMultiExpr, [97](#)  
 SetNetworkData, [98](#)  
 SetPFMList, [99](#)  
 SetPowerTable, [99](#)  
 SetRegulonScores, [100](#)  
 SetTFEval, [100](#)  
 SetTFNetwork, [101](#)  
 SetTFRegulons, [101](#)  
 SetupForWGCNA, [102](#)  
 SetWGCNAGenes, [102](#)  
 SetWGCNAParams, [103](#)  
 shuffle\_points, [103](#)  
  
 TestSoftPowers, [104](#)  
 TestSoftPowersConsensus, [105](#)  
 TFNetworkPlot, [106](#)  
 to\_pseudobulk, [107](#)  
 TransferModuleGenome, [108](#)  
  
 umap\_theme, [109](#)