Master Regulators Gene Scoring Overview

The script performs the following key operations:

1. Input:
   * A gene regulatory network (GRN) file in a tab-delimited format (gene1, gene2, MI) where MI is the Mutual Information between genes.
   * A gene expression statistics file, typically containing differential expression data (gene, logFC, pvalue, fdr).
2. Network Processing:
   * Reads the GRN and constructs an igraph network object.
   * Optionally filters the network to a subnetwork containing only the top N genes based on a user-specified statistic (e.g., top 20% of genes with the highest logFC).
   * Calculates network centrality measures (degree, betweenness, coreness, PageRank, eigenvector centrality) for each gene if specified in the user options.
3. Master Regulator Score Calculation:
   * Implements a neighborhood-based scoring algorithm.
   * For each gene, it examines its neighborhood up to a certain number of steps (hops) away in the network.
   * It aggregates the gene expression statistics (e.g., logFC, p-value) of the neighbors, weighted by the strength of the connections (e.g., MI or correlation derived from MI) and the distance from the source gene.
   * The aggregation method can be sum, average, or median.
   * Optionally, it can incorporate the source gene's statistics into the score (e.g., by summing or multiplying it with the neighborhood score).
   * It can perform iterative calculations, where the scores from one round are used as input for the next round, potentially refining the results.
4. Output:
   * A table of genes with their calculated master regulator scores.
   * A ranking of genes based on these scores.
   * Optional: validation against a user-provided list of known master regulator genes.

User Guide

1. Prerequisites

* R: You need to have R installed on your system.
* Packages: The script requires the following R packages:
  + igraph
  + optparse
  + data.table
  + Rmpi (if you want to use parallel processing)

You can install these packages using the following commands in R:

install.packages(c("igraph", "optparse", "data.table"))

# For parallel computing (optional)

install.packages("Rmpi")

2. Input Files

* Network File ( -n or --network option):
  + This file defines the gene regulatory network.
  + Format: Tab-delimited text file.
  + Columns:
    1. gene1 (name of the first gene)
    2. gene2 (name of the second gene)
    3. MI (Mutual Information between gene1 and gene2)
  + Example:

geneA geneB 0.5

geneB geneC 0.8

geneA geneC 0.2

content\_copy Use code [with caution](https://support.google.com/legal/answer/13505487).

* Gene Expression Statistics File ( -x or --gene\_ex\_stat option):
  + This file provides statistics for each gene, typically from a differential expression analysis.
  + Format: Tab-delimited text file or an RDS file (R data file). If it's an RDS file, it should contain a data frame with the required columns or a list with an element named lr\_table that contains the data frame.
  + Columns (order matters):
    1. gene (gene name)
    2. logFC (log2 fold change)
    3. pvalue (p-value)
    4. fdr (false discovery rate)
  + Example:

gene logFC pvalue fdr

geneA 2.5 0.01 0.05

geneB -1.2 0.001 0.02

geneC 0.8 0.05 0.1

3. Running the Script

You can run the script from the command line using Rscript. Here's the basic syntax:

Rscript Master\_Regulator\_Scoring\_v12.R --cmd <command> [options]

Commands (--cmd option):

* run: This is the main command to calculate master regulator scores.
* btw: Calculates node-weighted betweenness centrality. This is often a separate analysis and might not be directly used in the master regulator score calculation.
* btw\_jobs: Generates job scripts for calculating betweenness on a cluster, useful for large networks.
* parameter\_optimization: (Not fully described in the script, but likely intended for optimizing the script's parameters.)
* help: Displays the help message, showing all available options.

Example: Running the run command:

Rscript Master\_Regulator\_Scoring\_v12.R --cmd run --network my\_network.txt --gene\_ex\_stat my\_gene\_stats.txt --step 2 --gene\_statistics\_list "logFC,pvalue" --edge\_weight rho --nround 3 --outdir my\_results

4. Key Options (for the run command)

* -n or --network: Path to the network file.
* -x or --gene\_ex\_stat: Path to the gene expression statistics file.
* -u or --consider\_positive\_values\_only: If TRUE, only positive values in the input statistics (e.g., logFC) are considered.
* -f or --fdr\_to\_confidence: If TRUE, converts FDR values to confidence scores (1 - FDR).
* -s or --step: The number of steps (hops) to consider in the network neighborhood (default: 2).
* -g or --top\_gene\_statistics: The statistic used to select the top genes for subnetwork extraction (e.g., "logFC", "pvalue", "fdr") (default: "pvalue").
* -l or --gene\_statistics\_list: A comma-separated list of statistics to use in the score calculation (e.g., "logFC,pvalue,degree", "fdr,betweenness") (default: "logFC,pvalue,pagerank").
* -r or --nround: The number of iterative rounds (default: 1).
* -e or --edge\_weight: The type of edge weight to use: "rho" (correlation-like, derived from MI), "MI" (Mutual Information), or "unw" (unweighted) (default: "unw").
* -p or --weight\_power: The power to which edge weights are raised (default: 1).
* -o or --step\_power: The power used in combining scores from different steps (default: 0).
* -z or --steps\_combined: If TRUE, scores from different steps are combined (default: FALSE).
* -y or --step\_normalization: If TRUE, scores at each step are normalized before combining (default: FALSE).
* -w or --input\_normalization: If TRUE, input statistics are normalized before calculations (default: FALSE).
* -t or --top\_genes: The proportion of top genes to include in the subnetwork (e.g., 0.2 for 20%) (default: 0.2).
* -k or --use\_rank: If TRUE, uses the rank of the statistic instead of its value (default: FALSE).
* -m or --master\_genes: A comma-separated list of known master regulator genes for validation (e.g., "geneA,geneB,geneC") (default: "POU5F1,SOX2,MYCN,NANOG,LIN28A").
* -d or --source\_node\_inclusion: How to include the source gene's statistics: "n" (no), "s" (sum with neighborhood score), "p" (multiply with neighborhood score), or "m" (use only source node statistics) (default: "p").
* -a or --neighbor\_aggregation\_method: How to aggregate neighbor scores: "s" (sum), "a" (average), or "m" (median) (default: "a").
* -v or --verbose: If TRUE, prints verbose messages during execution (default: TRUE).
* -i or --index: An index for the experiment (used for output file naming) (default: 1).
* --outdir: The output directory (default: "nSCORE").

5. Output Files (for the run command)

The script will create an output directory (specified by --outdir) and generate the following files:

* scores.csv: A table of genes with their calculated master regulator scores.
* ranks.csv: A table of genes ranked by their scores.
* master\_genes\_ranks.csv: The ranks of the user-provided master genes (if provided).
* experiment\_condition.csv: A record of the options used in the experiment.
* experiment<index>.log: A log file containing messages and results.
* <gep\_filename>\_master\_score\_result.RDS: An RDS file containing the results (scores, ranks, master gene ranks, and the input differential expression data).

6. Betweenness Centrality Calculation (btw and btw\_jobs commands)

* btw command:

Rscript Master\_Regulator\_Scoring\_v12.R --cmd btw --network my\_network.txt --gene\_ex\_stat my\_gene\_stats.txt --gene\_index <start\_index> --batch\_size <size> --outdir DFs

* + --gene\_index: The starting index of the genes to process.
  + --batch\_size: The number of genes to process in one batch.
  + --outdir: The output directory for intermediate files (default: "DFs").

This command calculates the node-weighted betweenness for a subset of genes and saves the results in intermediate files within the "DFs" directory.

* btw\_jobs command:

Rscript Master\_Regulator\_Scoring\_v12.R --cmd btw\_jobs --network my\_network.txt --gene\_ex\_stat my\_gene\_stats.txt --batch\_size <size> --outdir DFs

This command generates job scripts for running the betweenness calculation on a cluster. It divides the genes into batches and creates a job script for each batch. You can then submit these job scripts to your cluster's job scheduler. The final betweenness calculation is done by the nw\_betweenness\_nompi\_finalize function (which is called within the nw\_betweenness\_nompi\_manager function, which is itself called by the btw\_jobs command) after all the cluster jobs have completed.

7. Important Notes:

* Computational Cost: The script can be computationally intensive, especially for large networks and multiple steps. Using a subnetwork of top genes can significantly reduce computation time.
* Parameter Tuning: The choice of parameters (e.g., step, edge\_weight, weight\_power, source\_node\_inclusion, neighbor\_aggregation\_method) can affect the results. It's often necessary to experiment with different parameter settings to find the best configuration for your specific data.
* Parallel Processing: The script supports parallel processing using Rmpi. You'll need to have Rmpi installed and configured to use it. The script automatically detects if you're running on a cluster and adjusts the number of slaves accordingly.
* Memory Management: The script tries to manage memory by removing large objects when they're no longer needed. However, for very large networks, you might still encounter memory issues.
* Convergence: When using iterative calculations (nround > 1), the script checks for convergence by comparing the ranks of genes between consecutive rounds. The convergence criterion is based on the average change in ranks.

8. Example Workflow

1. Prepare Input Files: Create your network file and gene expression statistics file according to the specified formats.
2. Run the run command: Execute the script with appropriate options.
3. Examine Output: Analyze the scores.csv, ranks.csv, and master\_genes\_ranks.csv files to identify potential master regulators.
4. Parameter Optimization (Optional): Experiment with different parameter settings to see how they affect the results. You might consider using a smaller subset of your data for faster parameter exploration.
5. Betweenness Calculation (Optional): If you want to calculate node-weighted betweenness, use the btw or btw\_jobs commands.

Troubleshooting

* Errors: If you encounter errors, carefully check the log file (experiment<index>.log) for error messages. Make sure your input files are in the correct format and that all required packages are installed.
* Memory Issues: If you run out of memory, try using a smaller subnetwork (top\_genes option) or reducing the number of steps (step option). You might also need to run the script on a machine with more RAM.
* Slow Execution: If the script is running very slowly, consider using parallel processing (Rmpi) if available. You can also try reducing the number of steps or using a subnetwork.

This comprehensive guide should help you understand and use the Master\_Regulator\_Scoring\_v12.R script effectively. Remember to adapt the parameters and workflow to your specific research question and data.