MInfer: MInfer: Bridging MetaboAnalyst and Jacobian analysis for metabolomic networks

Overview

MInfer is an R package designed to streamline the analysis of metabolomics data. It offers tools for:

- Data preparation
- Covariance matrix generation
- Jacobian matrix computation
- Visualization of metabolite interaction networks

This package enables a seamless integration of MetaboAnalyst's capabilities with advanced Jacobian analysis, allowing researchers to explore dynamic interactions between metabolites comprehensively.

Installation

To begin using MInfer, follow these steps:

Prerequisites

Ensure that the `devtools` package is installed on your system:

```
install.packages("devtools")
```

Install MInfer from GitHub

Install the package directly from its repository:

```
devtools::install_github("cellbiomaths/MInfer")
```

Load the Package

Once installed, load the package into your R session:

```
library(MInfer)
```

Key Features

Targeted and Non-Targeted Data Support

- MInfer is designed for identified metabolites (targeted analysis).
- It also supports non-targeted metabolomics data, but only annotated compounds are mapped to pathways to ensure unbiased analysis.

Dynamic Interaction Analysis

- Generate Jacobian matrices to model interactions between metabolites under varying conditions.
- Visualize networks using heatmaps and 3D plots.

Flexible Data Preparation

- Select experimental conditions to focus your analysis.
- Generate covariance matrices for dynamic and network modeling.

Usage Examples

Description of Example Data / Input data

Example data (represented input data structures) can be loaded using `data(example_data)`, and it includes three essential input datasets: interactions_fin, met_input, metabolites_fin

```
interactions_fin
```

A metabolomic interaction network where:

- `0` indicates no connection between metabolites,
- `1` indicates a connection between metabolites.

The matrix is not symmetrical. It can be created using the `minfer()` function, manually, or based on experiments.

```
met_input
```

Contains concentrations of various metabolites for different samples. Each row corresponds to a sample, and individual columns represent metabolites.

accession	**Glycine**	**Serine**	**Sucrose**	**Fructose**	**Glucose**	**...**	**condition**
410	8099.47	9065.15	38493.92	54829.86	33519.31	...	16
410	5203.81	15432.67	32679.50	32949.91	26036.59	...	16
410	10557.05	54895.31	430.38	46382.25	28286.06	...	16
428	11572.03	27038.25	47901.46	53650.70	45046.97	...	16
428	1683.61	5322.68	51136.01	36213.23	25736.04	...	6
428	2889.17	6284.49	88456.79	58978.30	56105.71	...	6

- `accession` – sample identifier,
- individual columns correspond to metabolites,
- `condition` – experimental condition.

```
metabolites_fin
```

List of metabolites included in the analysis.

V1
Glycine
Serine
Sucrose
Fructose
Glucose
Galactose
…

Each row corresponds to a single metabolite.

Workflow 1: Analyze a Metabolite Interaction Network (MIN)

Step 1: Define Metabolite IDs

Define a list of KEGG IDs to analyze. These IDs must be present in the dataset:

```
# Define metabolite IDs
input_ids <- c('C00042', 'C00149', 'C00036')

# Run the minfer function
results <- minfer(input_ids)

# Display the results
print(results)
```

Workflow 2: Compute and Visualize Jacobian Matrices

Step 1: Load Example Data

MInfer includes example datasets for quick testing:

```
# Load example data
data(example_data)
```

Step 2: Prepare Data

Select specific experimental conditions (e.g., 6°C and 16°C):

```
# Prepare data for two conditions
data_6C <- prepare_data(met_input, 6)
data_16C <- prepare_data(met_input, 16)
```

Step 3: Generate Covariance Matrices

Specify the number of time points using the `num_tp` parameter:

```
# Generate covariance matrices
cov_6C <- generate_covariance(data_6C, num_tp = 1)
cov_16C <- generate_covariance(data_16C, num_tp = 1)
```

Step 4: Calculate Jacobian Matrices

Use the covariance matrices to compute Jacobian matrices:

```
# Calculate Jacobian matrices
jacobian_6C <- calculate_jacobian(cov_6C[[1]], interactions_fin, icount=15)
jacobian_16C <- calculate_jacobian(cov_16C[[1]], interactions_fin, icount=15)
```

Step 5: Visualize Results

MInfer includes visualization tools for heatmaps and 3D plots:

```
# Heatmap visualization
visualize_heatmap(jacobian_6C$J,colnames(met_input)[2:37], title = "Jacobian
Matrix - 6C")
```

`jacobian_6C$J`: This is the Jacobian matrix for the 6°C condition, which was computed in the previous step using the `calculate_jacobian` function. `jacobian_6C$J` contains the resulting Jacobian matrix, which describes the dynamics of interactions between metabolites under the given experimental condition (6°C).

`colnames(met_input)[2:37]`: This is a vector of metabolite names used for the axis labels (both x and y) on the heatmap. `colnames(met_input)` returns the names of all columns in the `met_input` file, where the first column likely contains sample identifiers (accession), and columns from index 2 to 37 correspond to the metabolites whose concentrations are measured. This part of the command is used to assign metabolite names to the axes of the plot.

`title = "Jacobian Matrix - 6C"`: This is an optional argument that sets the title of the heatmap. In this case, the title is set to `"Jacobian Matrix - 6C"`, indicating that the heatmap shows results for the Jacobian matrix under the 6°C experimental condition.

```
# 3D visualization
visualize_3d(jacobian_16C$J)
```

Advanced Features

Targeted vs. Non-Targeted Data

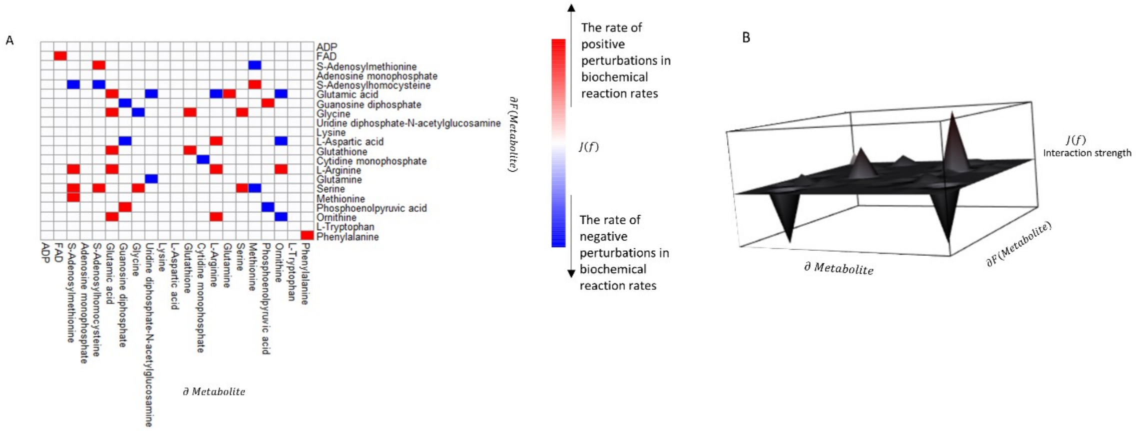
- **Targeted Analysis:** Works directly with identified metabolites for precise pathway mapping.
- **Non-Targeted Analysis:** Accepts datasets with unidentified metabolites but maps only annotated compounds to pathways, ensuring comprehensive yet unbiased results.

Time Point Flexibility

- The `num_tp` parameter allows modeling dynamic interactions across various time points.

Visualization

- Built-in tools offer both 2D (heatmap) (A) and 3D visualizations (B), facilitating clear interpretation of metabolite interactions.



Practical Tips

- **Ensure Data Compatibility:** Use datasets with KEGG IDs for accurate mapping.
- **Preprocess Data:** Focus on high-quality, annotated metabolites to enhance analysis reliability.
- **Combine Approaches:** Integrate MInfer results with external tools like MetaboAnalyst for broader insights.

Getting Help / Contact

For questions or support, contact the developers via the GitHub repository / https://github.com/cellbiomaths or https://github.com/JanaSchwarzerova / or email Jana.Schwarzerova@vut.cz / jana.schwarzerova2@fno.cz