

Developing a Shiny Interface for "Maplet" Analysis Pipelines

Jinfeng Lu, Yifan Wu, Xiang Zhu

July 9th, 2021

Abstract

This paper presents the user guide and developer guide for a R Shiny-based, interactive interface prototype for maplet. Maplet is an extensive R toolbox for modular and reproducible statistical pipelines for omics data analysis. Shiny is a powerful R package that enables the developers to build elegant interactive web applications. Our maplet Shiny interface is based on the SummarizedExperiment data structure created by the maplet pipelines. The application is constituted by six modules with a variety of functionalities, including data uploading, exploration and visualization. The concise and user-friendly page design facilitates exploratory analysis, statistical analysis, result reporting, and visualization for researchers. In addition, the interface allows the user to download all interactive plots and to customize the plot annotations, which efficiently helps users to better inspect the data and gain insights into the statistical results. The developer guide describes in detail the structure of the interface for future revision and development.

Availability and implementation

The Shiny interface for maplet is freely available at:

<https://github.com/krumsieklab/maplet-shiny>.

Introduction

Maplet is an R toolbox for the statistical analysis of metabolomics and other omics data. It was developed by the Krumsiek Lab at Weill Cornell Medicine (Chetnik, Kelsey, et al). The processed data from maplet is a SummarizedExperiment (SE) object, which is a matrix-like container with features as rows and samples as columns. The routine analysis work generally formulated into pipelines covers the exploratory analysis, statistical analysis, result reporting, and visualization. The pipe operator (`%>%`) from the magrittr R package allows for intuitive sequential function calls.

To streamline the routine analysis and assist researchers in better grasping the information in the exploratory data analysis, we developed an interactive interface based on R Shiny that supports interactive exploration and visualization of the results produced by maplet. Shiny is a convenient way to build interactive web apps from R, and it is an open-source R package that turns graphical objects into interactive web applications without requiring web development knowledge ("Help Users Upload Files"). The interaction with the app is primarily "point and click". To run the app,

users need to install the maplet package first and run “code_for_SE.R” to generate an example SE object (or upload one of their choice). The interactive interface offers all functionalities in 6 modules, including the data uploading, exploration and visualization. All plots and tables with the hover text can be downloaded for the inspection and understanding of plots. Despite being a prototype, the developed interface gives the user several tools for the exploration of complex data and results in an interactive and user-friendly setting.

1. Vignettes

Six modules are developed in the interface. The work flow and feature begin from the data uploading in the first module, Real-Time Pipeline Module. The Annotations Explorer Module supports the data exploration. The data visualization covers the 2D Projection Module, All Results Explorer Module, Feature Results Explorer Module, and Pathway Results Explorer Module. Additionally, the All Results Explorer Module accesses plots and tables stored in maplet pipelines, while the Pathway Results Explorer Module, 2D Projection Module, Feature Results Explorer Module generate the new interactive plots and tables.

1.1 - Real-Time Pipeline Module

The Real-Time Pipeline Module enables the functionality of internally creating and running a pipeline. Users can first upload the data and then preprocess it by setting parameters. The parameter setting includes 1) file name; 2) file sheets (data and annotations separately) and corresponding identification columns; 3) outcome column; 4) parameters and attributes (statistical method, significance level, coloring column, etc). After the parameter setting, users can perform a basic differential analysis for the preprocessed data. In addition, if users want to navigate to other modules for results exploration of this uploaded data, they need to download the new SE object in the local directory and use this new SE object for visualization in other modules.

The sidebar options of the Real-time Pipeline Module is displayed in the figure below:

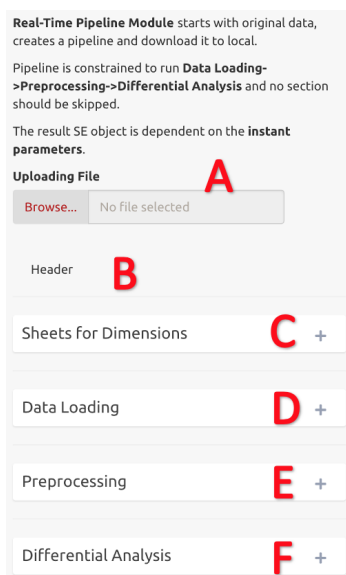


Figure 1.1. The Sidebar Options of the Real-Time Pipeline Module

- A. Select the file to upload;
- B. Indicate whether to specify the first row as a header for the uploaded file;
- C. Select the corresponding sheets for assay, rowData, and colData when generating the SE object;
- D. Select the ID columns for assay, rowData, and colData for the SE object;
- E. Set the values for preprocessing to allow the user to perform typical preprocessing operations on the data, for example, set the color column for box plot;
- F. Set the values in differential analysis.

1.2 - Annotations Explorer Module

The Annotations Explorer Module guides users to explore the SE object by creating tables, distribution plots, or other graphics after determining a dimension (assay, rowData, colData) and variables of interest (all or only a subset). For the feature annotations (rowData), a series of interactive pie charts are generated; for the patients' annotations (colData), an interactive plot (scatter/box/bar plots) with multiple choices of hover text displays.

The sidebar options of the Annotations Explorer Module is displayed in the figure below:

Figure 1.2. The Sidebar Options of the Annotations Explorer Module

(the left picture for displaying the feature annotations (rowData); the right picture for displaying the patients' annotations (colData))
A. Select which data dimension the user wants to explore in the SE object;

B. Select the x column for plotting;

C. Select if the x column should be treated as continuous;

D. Select the y column for plotting;

E. Select if the y column should be treated as continuous;

F. Add hover text information to the plot;

G. Select the grouping column if inspecting the row data.

1.3 - 2D Projection Module

The 2D Projection Module supports the visualization of the SE object in an interactive 2D PCA/UMAP projection with the hover text. A drop-down menu of all colData columns allows the plot coloring. It gives users the option to regenerate a new plot for the 2D projection by resetting the parameters. The interface allows users to choose the data type for the 2D PCA project and the number of neighbors for the 2D UMAP project.

The sidebar options of the 2D Projection Module are shown below:

2D Projection Module generates an interactive 2D projection of PCA/UMAP.

It displays a drop-down menu of all colData columns for coloring.

Select one plot type:

☒ PCA

☐ UMAP

Select data type for PCA:

scores

Scaled data

Select one coloring variable:

BOX.NUMBER

☐ Categorical Coloring

Select hovering text:

sample

Hint: Outputs are delayed until you click 'UPDATE' button after selection.

Update

2D Projection Module generates an interactive 2D projection of PCA/UMAP.

It displays a drop-down menu of all colData columns for coloring.

Select one plot type:

☐ PCA

☒ UMAP

Number of neighbors for UMAP:

15

Scaled data

Select one coloring variable:

BOX.NUMBER

☐ Categorical Coloring

Select hovering text:

sample

Hint: Outputs are delayed until you click 'UPDATE' button after selection.

Update

Figure 1.3. The Sidebar Options of 2D Projection Module

(the left picture for the 2D PCA project;
the right picture for the 2D UMAP project)

A. Select the method used for dimensionality reduction;

B. Select the matrix type if using the PCA method;

(PCA is based on a decomposition of the data matrix X into two matrices V and U . The matrix V is called the loadings matrix, and the matrix U is called the scores matrix);

C. Option to scale the data;

D. Select the color column;

(For PCA_scores and UMAP, it is the column names of the colData(D), and for PCA_loadings it is the column names of the rowData(D));

E. Whether to treat the color legend variable as categorical;

F. Add hover text information to the plot;

G. Set the number of neighbors when using the UMAP method.

(a parameter to balance local versus global structure in the data)

1.4 - All Results Explorer Module

The All Results Explorer Module offers the functionality of extracting all the result objects from the SE object in a plot or a table. Users are able to visualize the desired statistical results by choosing a statistic name (statname) in a drop-down menu. For example, if the user wants to output a specific plot, they need to select a when given a SE and a statname and a plot type (“Diagnosis Analysis”, “volcano”).

The sidebar options of the All Results Explorer Module are shown below:

All Results Explorer Module extracts all the result objects one at a time.

Users can assess results in a drop-down menu that offers a list of a stat_name and a plot type (e.g. "missingness", "pval").

Select output type:

☒ Plot

☐ Table

Select one stat name:

(no stat_name)

Select one object:

missingness

Hint: Outputs are delayed until you click 'UPDATE' button after selection. Some plots such as box plot or multiple plots may take dozens of seconds to show up.

Update

All Results Explorer Module extracts all the result objects one at a time.

Users can assess results in a drop-down menu that offers a list of a stat_name and a plot type (e.g. "missingness", "pval").

Select output type:

☐ Plot

☒ Table

Select one stat name:

Fisher's test - Missingness

Hint: Outputs are delayed until you click 'UPDATE' button after selection. Some plots such as box plot or multiple plots may take dozens of seconds to show up.

Update

Figure 1.4. The Sidebar Options of the All Results Explorer Module

(the left picture for the output of a plot;

the right picture for the output of a table)

A. Select the output type;

B. Select the name under which a statistical result is stored;

C. Select the plot type (only available for the output of a plot).

1.5 - Feature Results Explorer Module

The Feature Results Explorer Module visualizes the collection of all statistical results for a given variable in a table. Once the table appears, users can click on any row in the table to trigger to display an interactive volcano plot and then a box/scatter plot.

The sidebar options of the Feature Results Explorer Module are shown below:

Feature Results Explorer requires collection on all statistical results in a table given one metabolite name.

When clicking on one row, it should display interactive plots following the same orders in Module 2.

Select one metabolite:

acisoga

☐ Treat as categorical

Hint:

Outputs are delayed until you click 'UPDATE' button after selection.

Update

Figure 1.5. The Sidebar Options of the Feature Results Explorer Module

A. Select one metabolite name;

B. Option to treat the variable on x-axis in scatter/box plot as categorical.

1.6 - Pathway Results Explorer Module

The Pathway Results Explorer Module offers a series of interactive plots that can be triggered in fixed order when given a SE and a statname. The interface establishes the connection between one plot and the other plots, which follows any of two orders below. For the box/scatter plot, users can choose to visualize the variables in the categorical form or the numeric form.

1) Equalizer/Volcano plot → Box/Scatter plot

2) Pathway Bar plot → Equalizer/Volcano plot → Box/Scatter plot

The sidebar options of the Pathway Results Explorer Module are shown below:

Pathway Results Explorer: Displays a series of interactive plots at different granularities given a SE and a statname.

StatsBar plot -> Equalizer/Volcano plot -> Box/Scatter plot.

Select one stat name: **A**

Age Regression

Select plot1: **B**

Barplot

☐ No Barplot

Select plot2 type: **C**

☐ Equalizer

☐ Volcano

Select plot3 type: **D**

☐ Box

☐ Scatter

☐ Treat as categorical **E**

Hint:
Outputs are delayed untill you click 'UPDATE' button after selection.

Update

Figure 1.6. The Sidebar Options of the Pathway Results Explorer Module

- A. Select the name under which a statistical result was stored;
- B. Choose whether to start with the bar plot when generating the series of interactive plots;
- C. Select the second plot type;
- D. Select the third plot type;
- E. Option to treat the x-axis variable in box/scatter plot as categorical

2. Developer Guide

For future revision and development, the developer guide is essential to describe in detail an overview of data flow, the Shiny App structure, and internal functions we used in the Shiny interface (see Appendix). At this stage, our Shiny interface supports the functionalities of data uploading, exploration and visualization that are most necessary for the interface. Limitations of the current version of the interfaces will benefit the future improvement and extension.

2.1 - Data Flow

Maplet-shiny App starts by loading the “SE.Rdata” file of interest generated by the maplet package. The data flow of maplet-shiny is data uploading → data exploration → data visualization. The first module does not refer to the SE object of interest. The user usually begins with uploading an excel file and then sets parameters for the three sections of data loading, preprocessing, and differential analysis to customize an analysis pipeline. The generated resulting SE object from the Real-Time Pipeline module will not be updated for other modules automatically. If users want to explore and visualize the new pipeline, they need to download the generated SE object to the same local directory with this Shiny App and rename it to “SE.Rdata” before running the software again. This enables the other modules to access the data stored in the

SE object of interest. The second module allows the user to explore the annotations of features and samples. The last four modules provide interactive visualization of the plots and statistics tables stored in the metadata of the SE object either by displaying them exactly as they are or by generating new interactive plots.

2.2 - Shiny App Structure

- App.R
 - ui: controls the appearance and layout for the app
 - server: contains the functions and instructions to build the app
- help_functions.R: stores some functions used in the main script App.R
- code_for_SE.R: allows users to generate an example SummarizedExperiment (SE) object that can be loaded into the Shiny app. The example SE object is not uploaded to github due to the large size of data.
- www: a folder containing logos and CSS file

Limitations

There are some limitations to this prototype maplet interface. The first is the waiting time problem. There is a delay between the moment when one clicks the update/run button and the moment when the intended objects show up in the main panel. The time lag ranges from milliseconds to dozens of seconds. The uncertain duration of the waiting time depends on the size of the object to be displayed and the complexity of the plots. Conventional loading messages indicating the data processing cannot work well in this prototype interface. The reason is that the waiting time involves not only the data processing but also the process of printing objects. Another open issue for future extension is the robustness of these features against any combination of user inputs. The software is designed to accommodate any generated SE object, but the functionality is constrained especially in the Real-time Pipeline Module. This module only supports the basic parameter setting and attributes for customization. There is always more imagination space for aesthetical customization such as more informative labels and more consistent attributes, including fonts, margins, the layout and instant mark of user's footprint, etc. Moreover, this interface does not allow the automatic update of the new generated SE object for the other five modules. The user has to manually make sure the resulting SE object of interest is in the same local repository with the software to ensure the operation of the interface.

Acknowledgements

We appreciate the assistance and significant contributions from Dr. Elizabeth Sweeney, Dr. Jan Krumsiek, Elisa Benedetti and Kelsey Chetnik during the development of the first prototype of the Shiny interface for maplet analysis pipelines.

Reference

Chetnik, Kelsey, et al. "Maplet: An Extensible R Toolbox for Modular and Reproducible Omics Pipelines." *ArXiv.org*, 6 May 2021, arxiv.org/abs/2105.04305.

"Help Users Upload Files to Your App." *Shiny*, shiny.rstudio.com/articles/upload.html.

Appendix

- `get_obj_name`: extract the object names from the results stored in `metadata(D)$results`
 - `D`: SummarizedExperiment (SE) object
- `get_pathway_annotations` : get pathway annotations
 - `D`: SummarizedExperiment (SE) object
 - `pwvar`: pathway annotation column (extracted from corresponding `stat_bar`)
- `get_data_by_name`: get pathway annotations
 - `D`: SummarizedExperiment (SE) object
 - `args.name`: args name
 - `plot.name`: plot name
 - `stat.name`: name under which a statistical result will be stored
- `reverselog_trans` : create `reverselog_trans` for log10-SCALE in volcano plot
- `mod2_plot_vol`: get the volcano plot in Pathway Results Explorer Module
 - `D`: SummarizedExperiment (SE) object
 - `inputs`: selected values for the sidebar options for Pathway Results Explorer Module
 - `legend_name`: legend name for volcano plot
 - `d`: click information for bar plot
 - `pwvar`: pathway annotation column name
 - `alpha`: threshold for significance (extracted from corresponding `stat_bar` plot)
- `mod2_plot_eq`: get the equalizer plot in Pathway Results Explorer Module
 - `D`: SummarizedExperiment (SE) object
 - `inputs`: selected values for the sidebar options for Pathway Results Explorer Module
 - `rd`: pathway annotations
 - `alpha`: threshold for significance (extracted from corresponding `stat_bar` plot)
 - `pwvar`: pathway annotation column name
 - `path_name`: pathway/subpathy name if plot1 is null
 - `d`: click information for bar plot
- `mod2_plot_box_scatter`: get the box/scatter plot in Pathway Results Explorer Module
 - `D`: SummarizedExperiment (SE) object
 - `inputs`: selected values for the sidebar options for Pathway Results Explorer Module
 - `d.bar`: click information for bar plot
 - `d.eq`: click information for equalizer plot
 - `d.vol`: click information for volcano plot
 - `rd`: pathway annotations
 - `pwvar`: pathway annotation column (extracted from corresponding `stat_bar`)
 - `path_name`: pathway/subpathy name if plot1 is null
 - `alpha`: threshold for significance (extracted from corresponding `stat_bar` plot)
 - `is_categorical`: whether to treat the x-axis variable as categorical
 - `data`: data set used to draw scatter/box plots
- `mod3_plots_pca`: define PCA output function for 2D Projection Module referring the built-in meta-function '`mt_plots_pca`' of the maplet package
 - `D`: SummarizedExperiment (SE) object
 - `scale_data`: whether to scale the data
 - `color`: color column name for plot

- categorizing: whether to categorize color legend
 - data_type: scores or loadings
 - hover: hover text
- mod3_plots_umap: define UMAP output function for 2D Projection Module referring the built-in meta-function 'mt_plots_umap' of the maplet package
 - D: SummarizedExperiment (SE) object
 - scale_data: whether to scale the data
 - color: color column name for plot
 - categorizing: whether to categorize color legend
 - n_neighbors: set the parameter of n_neighbors when using UMAP function
 - hover: hover text
- mod5_boxplot: define boxplot function in Annotations Explorer Module
 - D: SummarizedExperiment (SE) object
 - x: name of x-axis variable (one of names(colData(D)))
 - x_cate: whether to categorize x-axis variable
 - y: name of y-axis variable (one of names(colData(D)))
 - y_cate: whether to categorize y-axis variable
 - fill: legend column name
 - hover: hover text
- mod5_scatter: define scatterplot function in Annotations Explorer Module
 - D: SummarizedExperiment (SE) object
 - x: name of x-axis variable
 - y: name of y-axis variable
 - hover: hover text
- mod5_barplot: define barplot function in Annotations Explorer Module
 - D: SummarizedExperiment (SE) object
 - x: name of x-axis variable
 - fill: legend column name
- get_log_text: get log text from a SE object
 - D: SummarizedExperiment (SE) object