Running the JGI Integration Workflow

Overview

The general stages of the workflow are:

- Loading and normalizing multi-omics datasets (e.g., transcriptomics and metabolomics)
- Integrating data (quantitative values) and metadata (sample info) between datasets
- Selecting features for analysis via statistical tests
- Building and visualizing a correlation network of integrated features
- Running multi-omics factor analysis

Note: This tutorial assumes you've already completed all stages of the workflow and environment setup detailed in *setup.pdf.*

Running the Notebook

1. You should see the JupyterLab interface rendered in your browser tab.
2. Double-click the workflow notebook `/integration_workflow.ipynb` in the left menu navigator to bring it into the workspace.
3. Double-click the configuration file in `/input_data/config/project_config.yml` to bring it into the workspace. The JupyterLab interface will open the file in a text editor.
4. Run a workflow with all default parameters:

   A. Start with the `integration_workflow.ipynb` in the workspace window.
   B. Ensure that the kernel is "JGI Integration" by checking the top right corner of the workspace. If not, click the kernel name and select "JGI Integration" from the dropdown menu.
   C. Run all notebook cells in order, either with the "play" button in the top menu bar or with the keyboard shortcut in each cell (ctrl/cmd/shift-click).
   D. Each cell performs a workflow step (data loading, normalization, integration, correlation analysis, etc.) and prints some information to the standard output for review.
   E. Review the cell print statements to see where plots and tables are saved to the `/output_data` directory; some key results or previews are also displayed in the notebook.

5. Run a workflow with customized parameters (optional):

   A. Edit the `/input_data/config/project_config.yml` file to update workflow parameters. Use the guides provided in the `/jgi_integration/docs/*_parameters_explained.md` files to understand how parameters work.

Note: Make sure to change the `data_processing_tag` and/or `data_analysis_tag` configuration parameters – these create a new output directory to store the results and keep them distinct from previous runs. Alternatively, you can set “overwrite=True” in the cells that create the dataset and/or analysis object to overwrite a previous run. If you do not change the tag or overwrite, the notebook console will print an error informing you of your options.

B. Rerun the notebook as described above from the beginning to re-load the updated configuration settings and run a full workflow with new parameters.

6. Examine the workflow results throughout the run with (A) the JupyterLab interface, (B) the notebook itself, or (C) on your local system file browser:

   A. To view the outputs within the JupyterLab web interface, navigate to the `/output_data` directory in the left side panel and click through the files and folders, or open a new workspace tab and use the built-in terminal to view files.

   B. To view the outputs within the notebook itself, you can access and view attributes of a dataset or analysis by creating a new cell and executing the command `<object>.<attribute>`. For example, running a cell with `tx_dataset.normalized_data` will show the transcriptomics count table (a dataframe) after all dataset normalization steps, or `mx_dataset.linked_metadata` will show the metabolomics metadata table after it has been linked to the other datasets.

   Note: to see the names of all possible attributes that you can view for each object (dataset or analysis), run a cell with the command `vars(<object>).keys()`. For example, `vars(analysis).keys()` or `vars(mx_dataset).keys()`.

   C. To view the outputs on your system (which is mounted to the docker container), navigate to `$INTEGRATION_DIR/project_data/output_data` directory (see *setup.pdf*) and view output files as you would normally on your local filesystem.

7. Close down the docker container and JupyterLab interface with (A) the Docker Desktop app, (B) the terminal/console running the container, or (C) a new terminal/console window.

   A. Open the Docker Desktop app and click on the Containers tab in the top left menu bar. Find the container named “jgi-integration” in the main panel and click the Stop button (black square) in the Actions column.

   B. Open the terminal/console window that is running the container (where you ran the … `docker compose` … command during instructions in *setup.pdf*) and press ctrl/cmd-C on your keyboard.

   C. Open a new terminal/console window and stop the container with a single command:

```
> docker stop jgi-integration
```

Notes on JupyterLab

- You will not be able to delete directories on the JupyterLab file browser (the left navigation panel), so if you need to remove a directory for any reason you can do so from your local file browser (for the directories mounted to your system) or via the built-in JupyterLab terminal.
- The default working directory for JupyterLab is `/home/joyvan/work` – this is set by the docker image and should not be changed.
- Details about how to navigate the JupyterLab interface can be found on the Jupyter documentation page.