Sep 01, 2024

NGAP Data Analysis Protocol

DOI
dx.doi.org/10.17504/protocols.io.14egn614pl5d/v1

Gordon Qian[1], Ryan Davis[1], Jennifer Johnston[2]

[1]University of Sydney; [2]NysnoBio

courtney.wright Wright
University of Sydney

OPEN ACCESS

DOI: dx.doi.org/10.17504/protocols.io.14egn614pl5d/v1

Protocol Citation: Gordon Qian, Ryan Davis, Jennifer Johnston 2024. NGAP Data Analysis Protocol . **protocols.io** https://dx.doi.org/10.17504/protocols.io.14egn614pl5d/v1

Protocol status: Working
We use this protocol and it's working

Created: September 01, 2024

Last Modified: September 01, 2024

Protocol Integer ID: 106799

Keywords: ASAPCRN, Neuronal Genome Atlas for Parkinsons (NGAP), Illumina Connected Analytics (ICA), WGS, Variants, iPSC

Funders Acknowledgement:
Michael J Fox Foundation
Grant ID: ASAP-000497

Abstract

Protocol for processing and analysis of data in the Neuronal Genome Atlas for Parkinsons (NGAP) on Illumina Connected Analytics (ICA). The protocol details the processing of data through the germline and somatic variant call pipelines on ICA and custom analysis of cell line metrics (Parkinsons Polygenic Risk Score, Tumour Normal comparison of parent and daughter lines for somatic variation, Mitochondrial local constraint, Mitochondrial DNA copy number and ploidy of autosomes) for comparison between cell lines.

Materials

Equipment

Computer with internet access

Consumables

Time and cloud storage

Before start

User account required to access ICA

Code and Data availability

Scripts and auxiliary data referenced in this guide can be found in the GitHub repository:
https://github.com/gqworks/ICA/

Running Dragen Germline/Somatic Whole Genome 4.2.4 v2 pipeline

1

- Running the Illumina Connected Analytics (ICA) DRAGEN pipeline can be done via a semi-automated python script that allows jobs to be submitted through the Command Line Interface (CLI).
- Make sure the following requirements are available:
  - Jupyter lab or notebook environment (see next section for setup).
  - Python libraries: subprocess, pandas, numpy, re.
  - icav2 (ICA's command-line interface)
- The script requires a metadata table (all_fastq_list.csv) containing upload information on all your samples. Instructions on creating this file are found below in the "Preparing an all_fastq_list file" section.

Installing the CLI tool

2

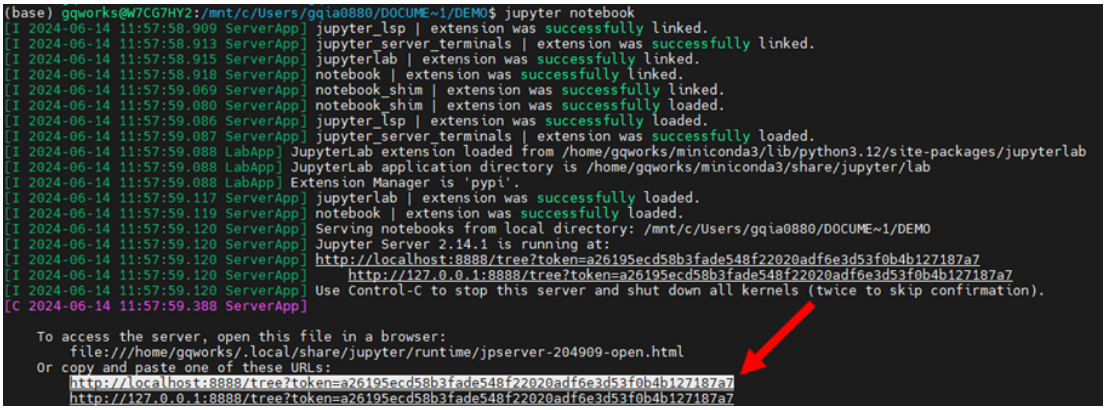
1. Download the version appropriate for your operating system here: https://help.ica.illumina.com/command-line-interface/cli-releasehistory
2. Follow the install instructions here: https://help.ica.illumina.com/command-line-interface/cli-installation
3. Note for Windows users: it is recommended to use the Windows Subsystem for Linux (WSL), this will minimize troubleshooting at later stages.

Install WSL: https://learn.microsoft.com/en-us/windows/wsl/install

Setting up a Jupyter environment

3

1. Run Jupyter lab/notebook from your working directory.

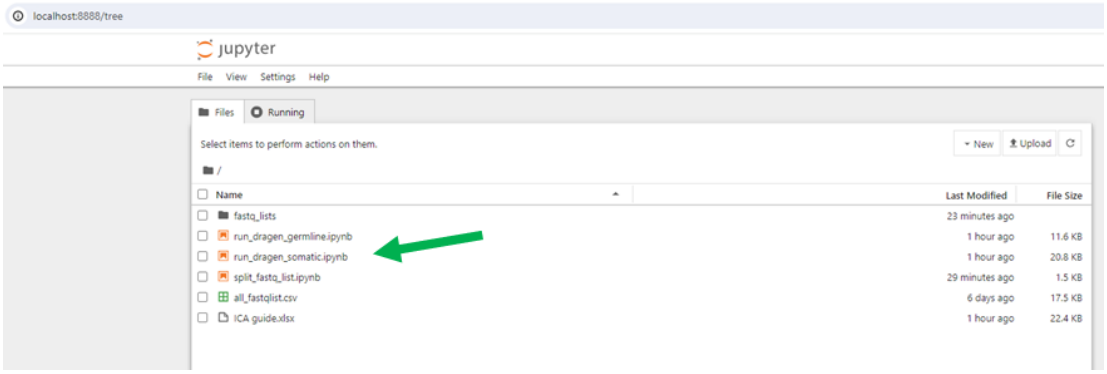


2. Copy and paste the unique link (indicated by the red arrow above) into your preferred web browser.

Note: If you are on Windows and not using WSL, you can use Anaconda to launch Jupyter.

3. Ensure all required files are in your working directory.

4. Double click a Jupyter notebook (.ipynb – green arrow below) to start it.



Preparing the “all_fastq_list” file

4 1. Follow the instructions on the “Info_DRAGEN” tab in the excel spreadsheet “ICA guide” to populate the table.

2. Save this file into your working directory and name it “all_fastq_list.csv”.

3. The file must then be split into sample-specific fastq lists.

4. Create a folder called “fastq_lists” in your working directory.

5. Run the “split_fastq_list.ipynb” Jupyter script to do this automatically. Make sure “all_fastq_list.csv” is in your working directory or edit the path in the script.

6. You can split the file manually. However, you must follow the naming convention of “{RGSM}_fastq_list.csv”, where {RGSM} is the name of the sample.

7. Upload all fastq_list files for individual samples into your ICA project. It is recommended to store these in the folder “fastq_lists” created in step 4. for better organization of files.

Launching the Dragen Whole Genome Germline/Somatic pipeline

5 1. Launch“run_dragen_germline.ipynb” or “run_dragen_somatic.ipynb” in Jupyter.

2. In the 2nd cell, edit your job parameters. These include:

a. ICA project name (target_project_NAME)

b. Output folder name on ICA (out_folder_NAME). Set this to the results folder you created in ICA (e.g. 01-DRAGEN.Output)

c. Location of all_fastq_list.csv (fastq_list)

d. Sample RGSM ID (RGSM, or normal_RGSM/ tumour_RGSM for T/N somatic pipeline)

e. Run name (run_name). Default is “RGSM_CLI”, but can be customized.

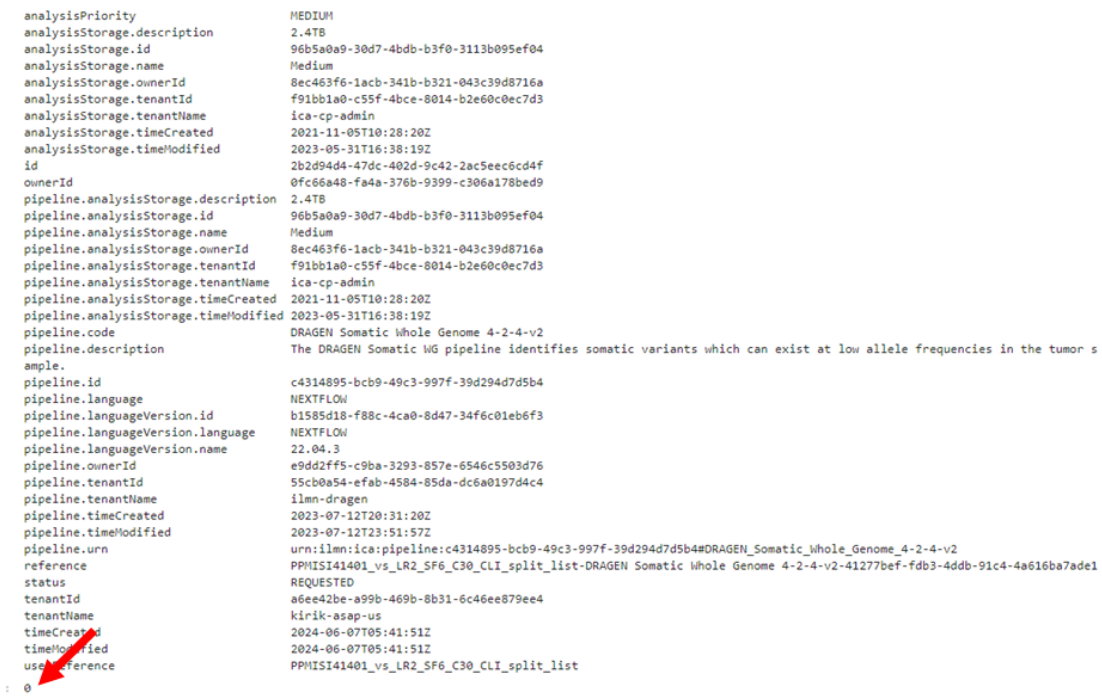
f. Run storage size (storage_size). Default = “Medium”

g. Runoutput prefix (output_prefix).

h. Sample sex (sample_sex). Default = “auto”

i. Enable germline on normal (enable_germline_on_normal). For T/N only.

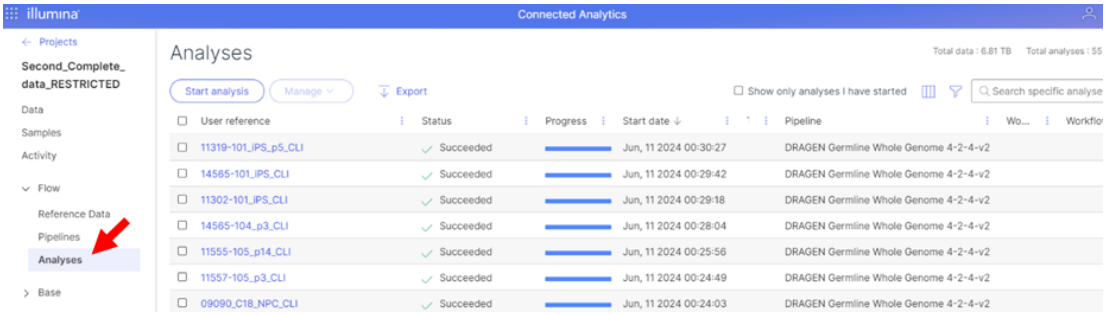
3. Once parameters are set, run all cells to submit the job (Shift+Tab on all cells, or select “Run all cells” from the Run panel at the top). Run is successfully submitted if you receive a 0 exit-status (red arrow below) and a similar output to below from the last cell



analysisPriority	MEDIUM
analysisStorage.description	2.4TB
analysisStorage.id	96b5a0a9-30d7-4bdb-b3f0-3113b095ef04
analysisStorage.name	Medium
analysisStorage.ownerId	8ec463f6-1acb-341b-b321-043c39d8716a
analysisStorage.tenantId	f91bb1a0-c55f-4bce-8014-b2e60c0ec7d3
analysisStorage.tenantName	ica-cp-admin
analysisStorage.timeCreated	2021-11-05T10:28:20Z
analysisStorage.timeModified	2023-05-31T16:38:19Z
id	2b2d94d4-47dc-402d-9c42-2ac5eec6cd4f
ownerId	0fc66a48-fa4a-376b-9399-c306a178bed9
pipeline.analysisStorage.description	2.4TB
pipeline.analysisStorage.id	96b5a0a9-30d7-4bdb-b3f0-3113b095ef04
pipeline.analysisStorage.name	Medium
pipeline.analysisStorage.ownerId	8ec463f6-1acb-341b-b321-043c39d8716a
pipeline.analysisStorage.tenantId	f91bb1a0-c55f-4bce-8014-b2e60c0ec7d3
pipeline.analysisStorage.tenantName	ica-cp-admin
pipeline.analysisStorage.timeCreated	2021-11-05T10:28:20Z
pipeline.analysisStorage.timeModified	2023-05-31T16:38:19Z
pipeline.code	DRAGEN Somatic Whole Genome 4-2-4-v2
pipeline.description	The DRAGEN Somatic WG pipeline identifies somatic variants which can exist at low allele frequencies in the tumor s ample.
pipeline.id	c4314895-bcb9-49c3-997f-39d294d7d5b4
pipeline.language	NEXTFLOW
pipeline.languageVersion.id	b1585d18-f88c-4ca0-8d47-34f6c01eb6f3
pipeline.languageVersion.language	NEXTFLOW
pipeline.languageVersion.name	22.04.3
pipeline.ownerId	e9dd2ff5-c9ba-3293-857e-6546c5503d76
pipeline.tenantId	55cb0a54-efab-4584-85da-dc6a0197d4c4
pipeline.tenantName	ilmn-dragen
pipeline.timeCreated	2023-07-12T20:31:20Z
pipeline.timeModified	2023-07-12T23:51:57Z
pipeline.urn	urn:ilmn:ica:pipeline:c4314895-bcb9-49c3-997f-39d294d7d5b4#DRAGEN_Somatic_Whole_Genome_4-2-4-v2
reference	PPMISI41401_vs_LR2_SF6_C30_CLI_split_list-DRAGEN Somatic Whole Genome 4-2-4-v2-41277bef-fdb3-4ddb-91c4-4a616ba7ade1
status	REQUESTED
tenantId	a6ee42be-a99b-469b-8b31-6c46ee879ee4
tenantName	kirik-asap-us
timeCreated	2024-06-07T05:41:51Z
timeModified	2024-06-07T05:41:51Z
userReference	PPMISI41401_vs_LR2_SF6_C30_CLI_split_list

: 0

5. Check the status of your job through the ‘Analyses’ tab under the ‘Flow’ menu in ICA (red arrow below).



Running auxiliary script for additional variant metrics

6

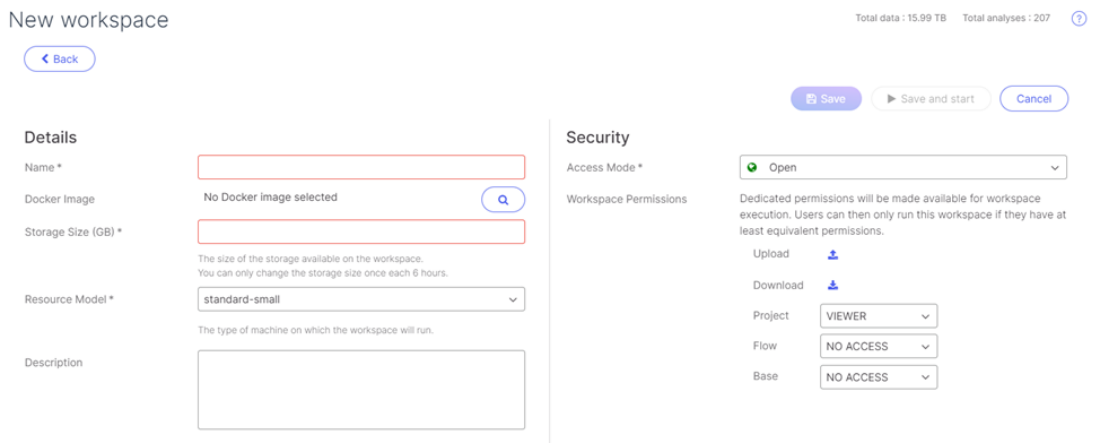
- The quality_check.py script provides the additional variant metrics:
  - Polygenic risk score (PRS) based on Nall's et.al 2019 Parkinsons disease risk variants.
  - Tumour Mutational Burden (TMB)
  - Mitochondrial Local Constraint (MLC)
  - Mitochondrial copy number (MCN)
  - Chromosome Ploidy

- Calculating/collecting these metrics require all sample Dragen Whole Genome results to be organised in a predefined folder hierarchical structure.
  - Place all **germline** results in the following ICA folder path: "/results/germline"
  - Place all **somatic** results in the following ICA folder path: "/results/somatic"

- Running the auxiliary script requires several file dependencies that should be located in the following ICA file path with the exact name.
  - /aux/hbnc.bb
  - /aux/MLC_supplementary_dataset_7.tsv
  - /aux/PGS000902_hg_38.txt

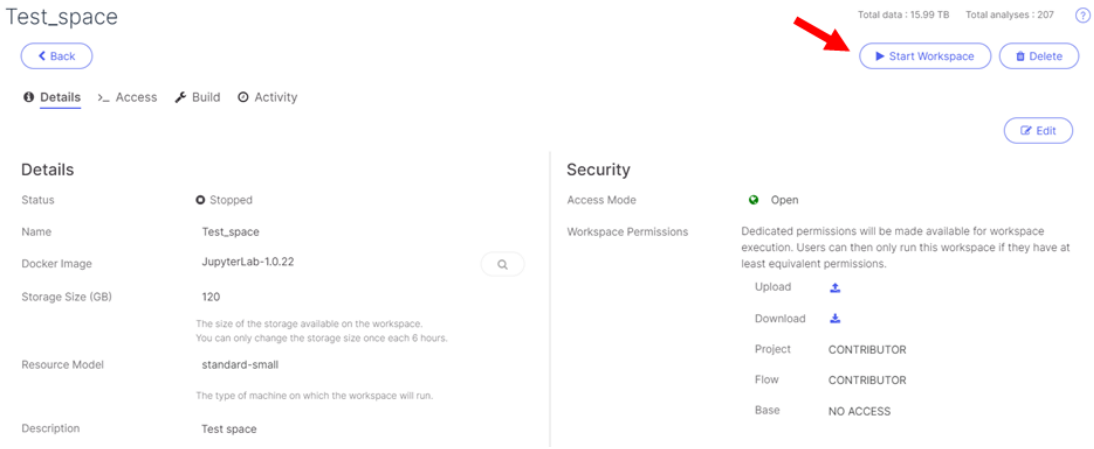
- The auxiliary script can be run in ICA through the workbench module by running a Jupyter lab Docker Image.

1. Create a new workspace in ICA



2. Provide a name for the workspace. Select the most recent JupyterLab docker image provided by ICA. Select an appropriate storage size for your analysis purposes. (Recommended: 64gb). Update Access mode and workspace permissions for Project (Recommended: Contributor role).

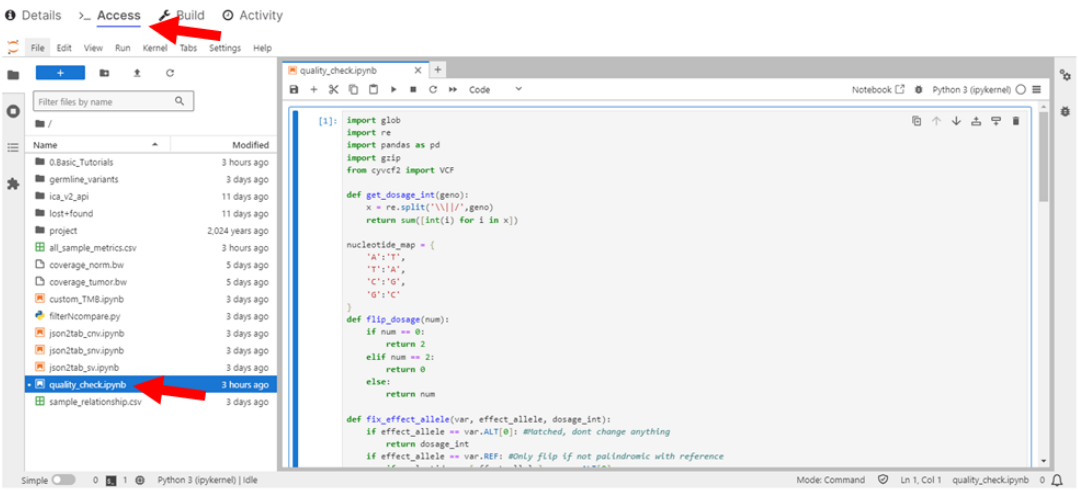
3. Start the workspace (red arrow below) once it is created. (This can take a few minutes)



4. Once the workspace has started, navigate to the ">_ Access" tab (red arrow below)

5. Upload the auxiliary script to the workspace root directory ~/data/

- **Note:** this is different to the ICA root directory which is mounted as ~/data/project/ in workspaces.



6. Open the Jupyter notebook “quality_check.ipynb” by double clicking it on the side bar (red arrow above)
and run the entire script.

7. The output file “all_sample_metrics.csv” will be found in the same directory after successfully running the script.

- **Troubleshooting:**
  o If there is a missing python library, install them by running pip install in a new cell. This can be removed after installation is successful.
  - E.g. !pip install cyvcf2

Running auxiliary script for comparing germline variants between samples

7

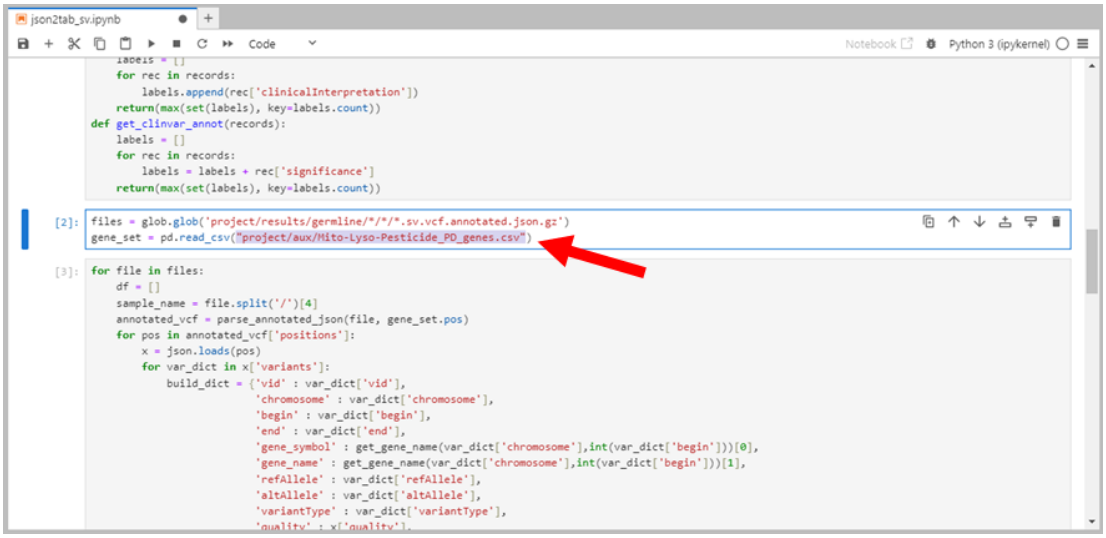
- This script converts DRAGEN’s annotated vcf files (.json) into a readable tabular format, whilst filtering variants by a given gene set. These variants can then be further filtered based on potential pathogenicity. For SNVs, additional ACMG classification is calculated. Once filtered, the script will automatically compare the difference in variants between parent and progenitor samples.

- This analysis is also performed in ICA workbench. Follow the instructions above to create a workbench session if you do not already have one created.

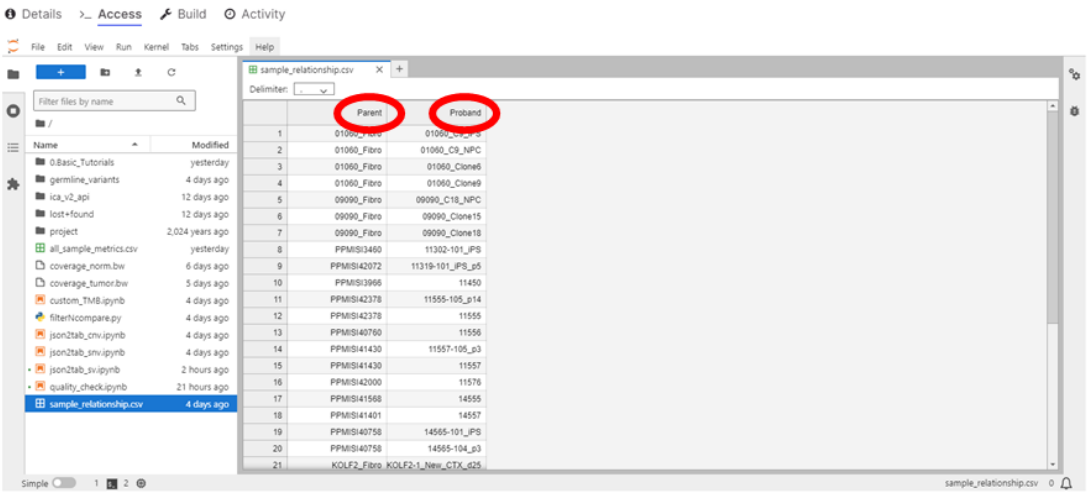
- The gene set of interest (for filtering variants) is required to be located in the ICA aux data folder.
  Currently the default gene set being used is (red arrow below):

o /aux/Mito-Lyso-Pesticide_PD_genes.csv

- If a different gene set is used, change the file name in the Jupyter notebook.



- In order to compare variants between parent and daughter samples, a samples relationship meta data table is required. This file should be a comma separated file with 2 columns, the parent sample name (Parent) and the daughter sample name (Proband). Save this file as “sample_relationship.csv” in the root directory (/data/).



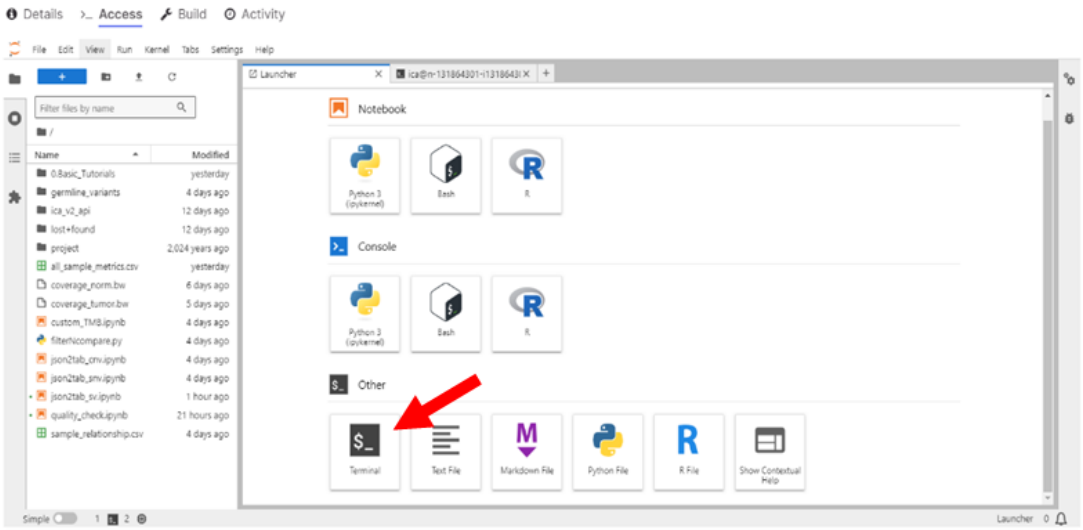
- Proceed with the following the steps:

1. Copy the json2tab_*.ipynb scripts into the root directory of workbench.

2. Create the following folders to contain intermediate files and results:
   - § /data/germline_variants/
   - § /data/germline_variants/snv
   - § /data/germline_variants/sv
   - § /data/germline_variants/cnv
   - § /data/germline_variants/snv_filtered
   - § /data/germline_variants/sv_filtered
   - § /data/germline_variants/cnv_filtered
   - § /data/germline_variants/snv_unique
   - § /data/germline_variants/sv_unique
   - § /data/germline_variants/cnv_unique

3. Run the json2tab_*.ipynb Jupyter notebook to convert annotated vcf json files into tabular format.

4. To filter and compare variants between samples run the filterNcompare.py python script. This can be done by opening a new terminal from the root (/data/) directory.



5. Run the script:

```
ica@n-131864301-i131864301-6c7k5:~$ python3 filterNcompare.py
```

6. Results can be found in the germline_variants folders.

Protocol references

Installations

- https://help.ica.illumina.com/command-line-interface/cli-releasehistory

- https://help.ica.illumina.com/command-line-interface/cli-installation

- https://learn.microsoft.com/en-us/windows/wsl/install