UNDERSTANDING BIOINFORMATICS PIPELINES

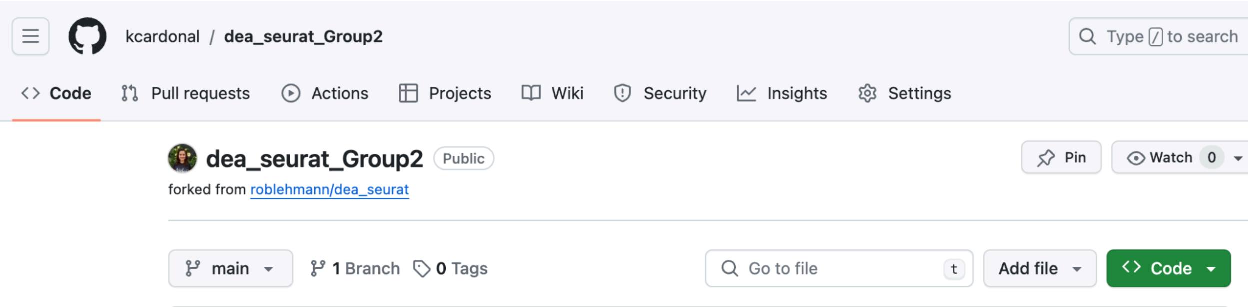
Week 4 assignment report: scRNA-Seq data analysis

Group 2: Kelly J. Cardona | Alejandra López V

I. SETTING UP THE SNAKEMAKE WORKFLOW FOR SINGLE-CELL ANALYSIS WITH SEURAT

1. Fork the repository from https://github.com/roblehmann/dea_seurat

The first thing we need to do is to fork the given GitHub repository for the single-cell analysis pipeline. Forking allows us to have our own version of the repository to make changes without affecting the original project.

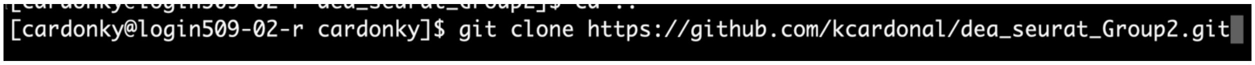


dea_seurat_Group2 Public

forked from roblehmann/dea_seurat

2. Clone the forked into IBEX

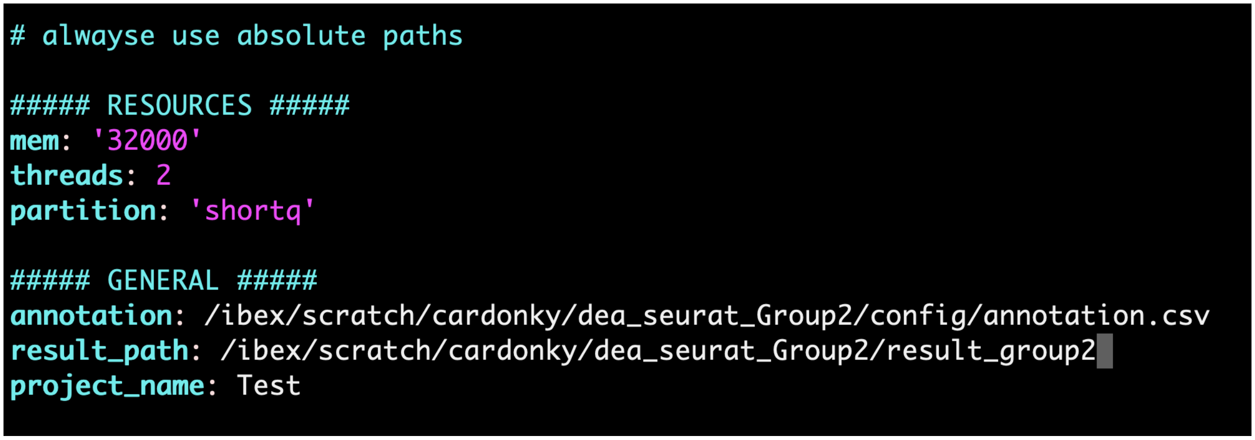
Then, we cloned the forked repository into IBEX and start exploring the workflow. The workflow implements a differential gene expression analysis using R starting from a Seurat object derived from the provided test datasets.



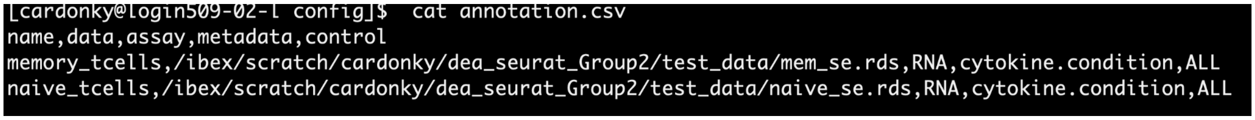
```
[cardonky@login509-02-r dea_seurat_Group2]$ cd ..
[cardonky@login509-02-r cardonky]$ git clone https://github.com/kcardonal/dea_seurat_Group2.git
```

3. Modify the configurations

In the config folder, we modified the config.yaml file to add the correct paths and the annotation.csv file where we specified the column name for grouping the data for the barplot.

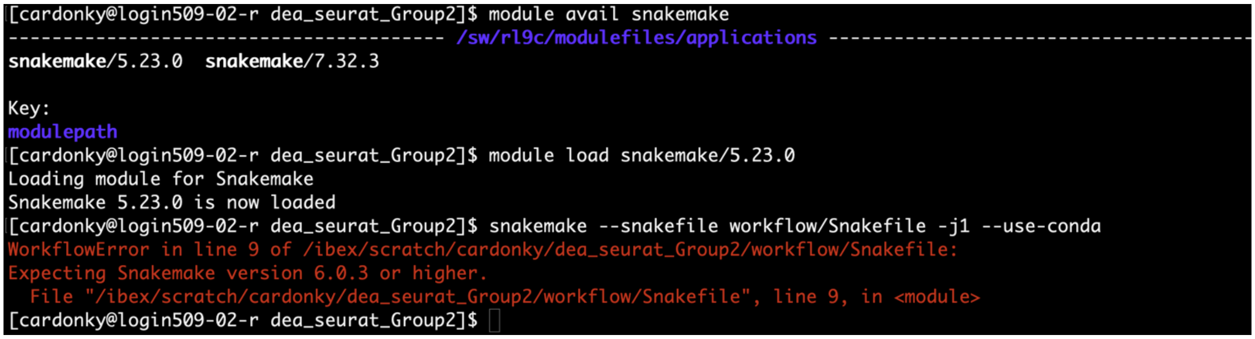


```
# alwayse use absolute paths

##### RESOURCES #####
mem: '32000'
threads: 2
partition: 'shortq'

##### GENERAL #####
annotation: /ibex/scratch/cardonky/dea_seurat_Group2/config/annotation.csv
result_path: /ibex/scratch/cardonky/dea_seurat_Group2/result_group2
project_name: Test
```



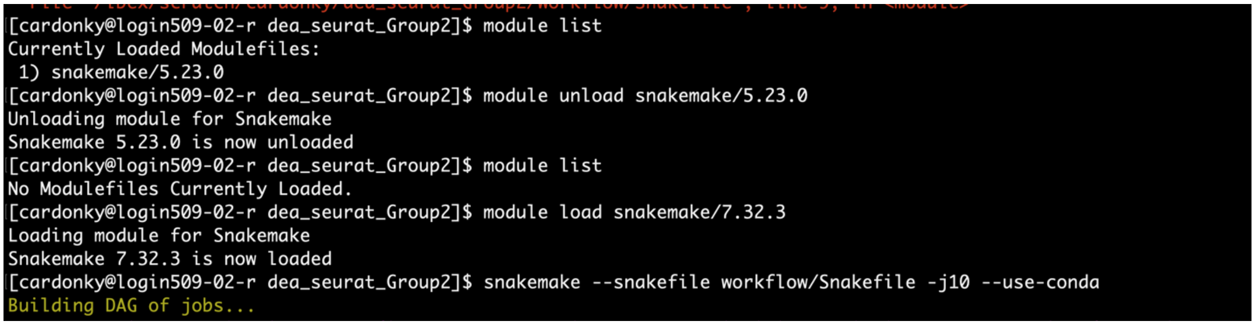
```
[cardonky@login509-02-l config]$  cat annotation.csv
name,data,assay,metadata,control
memory_tcells,/ibex/scratch/cardonky/dea_seurat_Group2/test_data/mem_se.rds,RNA,cytokine.condition,ALL
naive_tcells,/ibex/scratch/cardonky/dea_seurat_Group2/test_data/naive_se.rds,RNA,cytokine.condition,ALL
```

4. **Run Snakemake workflow on IBEX**
   We loaded the Ibex module for snakemake following the instructions of the snakemake tutorial repository provided in assignment (https://github.com/roblehmann/snakemake-tutorial), and then we ran the workflow



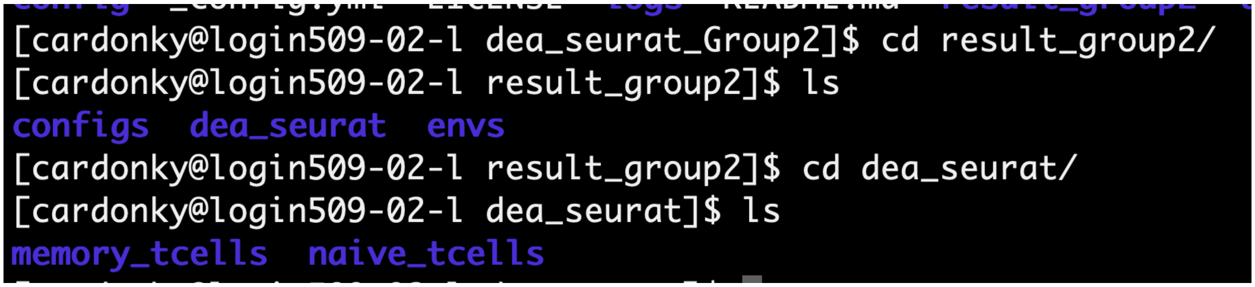
```
[cardonky@login509-02-r dea_seurat_Group2]$ module avail snakemake
---------------------------------------- /sw/rl9c/modulefiles/applications ----------------------------------------
snakemake/5.23.0  snakemake/7.32.3

Key:
modulepath
[cardonky@login509-02-r dea_seurat_Group2]$ module load snakemake/5.23.0
Loading module for Snakemake
Snakemake 5.23.0 is now loaded
[cardonky@login509-02-r dea_seurat_Group2]$ snakemake --snakefile workflow/Snakefile -j1 --use-conda
WorkflowError in line 9 of /ibex/scratch/cardonky/dea_seurat_Group2/workflow/Snakefile:
Expecting Snakemake version 6.0.3 or higher.
  File "/ibex/scratch/cardonky/dea_seurat_Group2/workflow/Snakefile", line 9, in <module>
[cardonky@login509-02-r dea_seurat_Group2]$ 
```

Since we got an error because the minimal expected snakemake version is 6.0.3 or higher, We retry using a more updated version of snakemake available at ibex. We also increased the number of threads from 1 to 10, modifying the parameter -j to run the workflow faster.



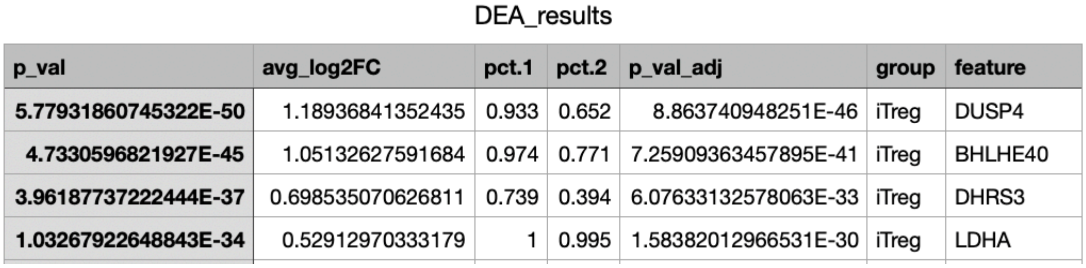
```
[cardonky@login509-02-r dea_seurat_Group2]$ module list
Currently Loaded Modulefiles:
 1) snakemake/5.23.0
[cardonky@login509-02-r dea_seurat_Group2]$ module unload snakemake/5.23.0
Unloading module for Snakemake
Snakemake 5.23.0 is now unloaded
[cardonky@login509-02-r dea_seurat_Group2]$ module list
No Modulefiles Currently Loaded.
[cardonky@login509-02-r dea_seurat_Group2]$ module load snakemake/7.32.3
Loading module for Snakemake
Snakemake 7.32.3 is now loaded
[cardonky@login509-02-r dea_seurat_Group2]$ snakemake --snakefile workflow/Snakefile -j10 --use-conda
Building DAG of jobs...
```

After a successful run, we got two folders with the results of running the workflow in the two datasets: Memory and Naive T cells.



```
[cardonky@login509-02-l dea_seurat_Group2]$ cd result_group2/
[cardonky@login509-02-l result_group2]$ ls
configs  dea_seurat  envs
[cardonky@login509-02-l result_group2]$ cd dea_seurat/
[cardonky@login509-02-l dea_seurat]$ ls
memory_tcells  naive_tcells
```

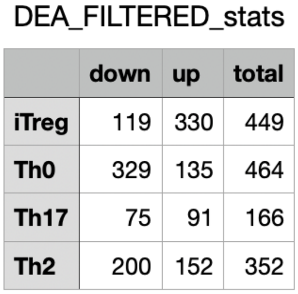
As a result we obtained a general table for the differential expression analysis **(1)**. For each of the datasets analyzed we got a summary of the up-regulated and down-regulated features per celltype (citokyne.condition). For the memory t-cells **(2)**, we get 119 down-regulated and 330 up-regulated features in the iTreg condition, 329 and 135 in Th0, 75 and 91 in Th17, and 200 and 152 in Th2. Whereas for the naive t-cells **(3),** we get 189 down-regulated and 245 up-regulated features in the iTreg condition, 54 and 191 in Th0, 230 and 165 in Th17, and 252 and 330 in Th2.



DEA_results

p_val	avg_log2FC	pct.1	pct.2	p_val_adj	group	feature
5.77931860745322E-50	1.18936841352435	0.933	0.652	8.863740948251E-46	iTreg	DUSP4
4.7330596821927E-45	1.05132627591684	0.974	0.771	7.25909363457895E-41	iTreg	BHLHE40
3.96187737222444E-37	0.698535070626811	0.739	0.394	6.07633132578063E-33	iTreg	DHRS3
1.03267922648843E-34	0.52912970333179	1	0.995	1.58382012966531E-30	iTreg	LDHA

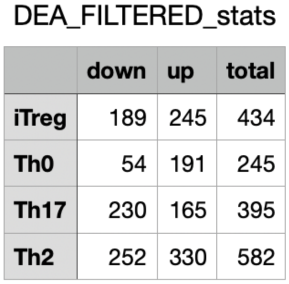
(1)



DEA_FILTERED_stats

	down	up	total
iTreg	119	330	449
Th0	329	135	464
Th17	75	91	166
Th2	200	152	352

(2)

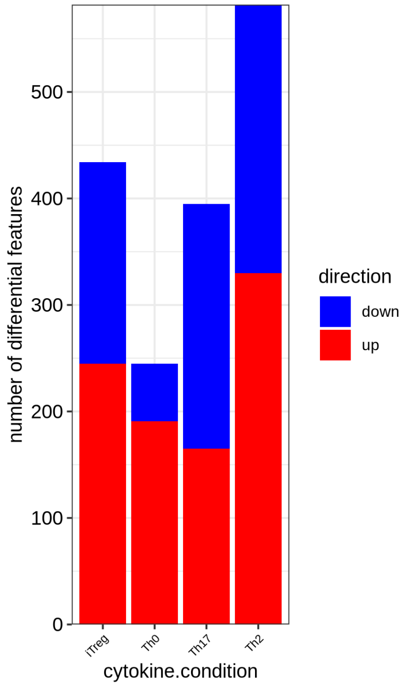


DEA_FILTERED_stats

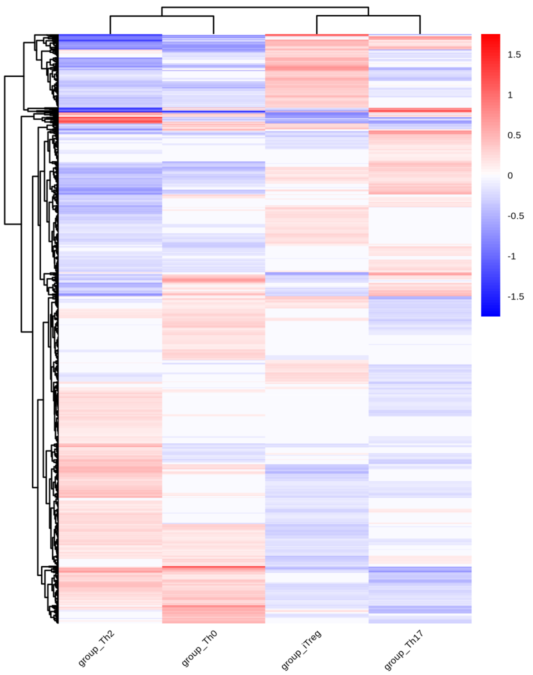
	down	up	total
iTreg	189	245	434
Th0	54	191	245
Th17	230	165	395
Th2	252	330	582

(3)

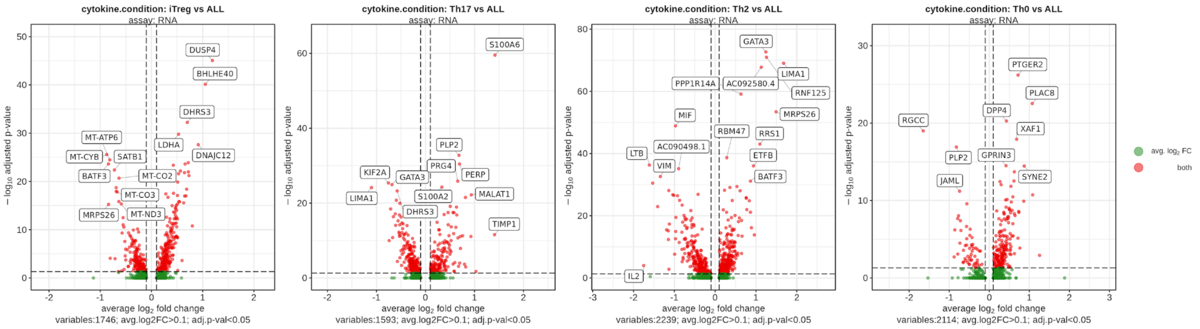
Moreover, we get a *feature lists* folder that represents a report of all the feature names (genes) that are up and down-regulated in the different conditions and a *plots* folder that contains some visualizations of the data through a barplot of the number of differential features (4), a heatmap of how gene expression is changing across the different conditions (5), and 4 volcano plots for each of the contrasts tested on the differential expression analysis (6).



(4)



(5)



(6)

II. MODIFYING THE WORKFLOW

As we are in Group 2, our task was to add a new rule to generate a bar plot showing the number of cells in groups as per the metadata table. To modify the provided workflow we performed three main steps:

1. **Write an R script to produce bar plots**
   We wrote an R script to generate the bar plot taking into account that our script should take inputs as defined in the Snakemake workflow and produce the desired bar plot as output. We added the barplot.R file in the *scripts* folder

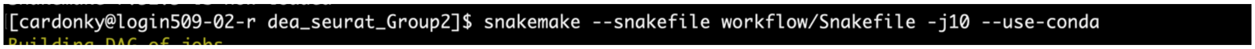
2. **Integrate the new R script into snakemake workflow, adding a new rule**
   Once our R script was ready, we added a new rule (barplot.smk) in the *rules* folder to execute our script as part of the workflow. This rule specifies the input, output, and the script to run.

3. **Add the new rule in the Snakefile**

We properly integrated the new rule into the workflow by adding it to the snakefile, in this way we guarantee it fits into the existing rule dependencies and execution order.

III. TESTING MODIFICATIONS AND ANALYZING THE RESULTS

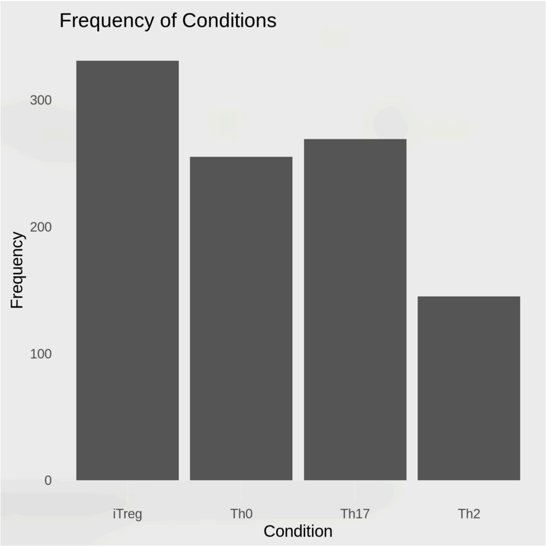
After integrating the new rule, we tested and debugged the workflow to ensure that it executed correctly and generated the expected bar plot. To execute the workflow we ran again the command:



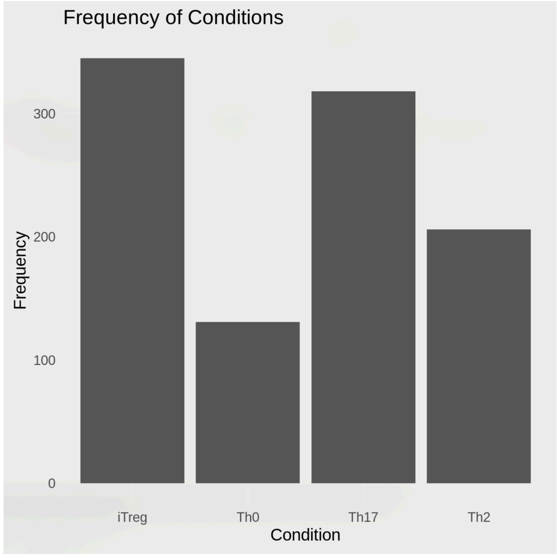
```
[cardonky@login509-02-r dea_seurat_Group2]$ snakemake --snakefile workflow/Snakefile -j10 --use-conda
```

As a result, we got two bar plots, one for each dataset analyzed (Memory and Naive T cells) the bar plots are saved in the corresponding *plots* folder of each experiment. Within the memory T-cell group **(7)**, the distribution of cell types was as follows: iTreg was the most abundant with 331 cells, followed by Th17 with 269 cells, Th0 with 255 cells, and Th2 with 145 cells. In contrast, within the naive T-cell group **(8)**, iTreg also represented the largest population with 345 cells, then Th17 with 318 cells, Th2 with 206 cells, and Th0 with 131 cells.

(7)



(8)



CONCLUSION

Throughout this activity we became familiarized with how to understand, create, and modify Snakemake workflows. Snakemake is a powerful tool for creating reproducible and scalable data analyses. We faced some challenges learning from scratch the basis of the workflows, and how to properly link the R scripts to the Snakemake rules, and then implementing new rules to the general Snakemake file. However, this exercise allowed us to learn how to manage complex data analysis workflow to make our projects more organized and reproducible, preparing us for future research projects. Moreover, we learned about version control, open-source projects, and the importance of contributions.

REFERENCES

Cano-Gamez, E., Soskic, B., Roumeliotis, T.I. et al. Single-cell transcriptomics identifies an effectorness gradient shaping the response of CD4+ T cells to cytokines. Nat Commun 11, 1801 (2020). https://doi.org/10.1038/s41467-020-15543-y

DEG analysis of Single-Cell Sequencing data using R and Seurat: https://github.com/roblehmann/dea_seurat

Snakemake tutorials
https://github.com/roblehmann/snakemake-tutorial
https://snakemake.readthedocs.io/en/stable/tutorial/tutorial.html