

SICILIAN

- The main issue I encountered was configuring SICILIAN to run with *10x single cell data, with paired-end* reads. It appears that for 10x data, SICILIAN is only expecting single-end reads. The main area in the sicilian.py script that deals with the configuration of which steps of SICILIAN to run is pictured below, in which you can see 5 boolean variables which act as toggles for determining which steps to run (i.e., if a variable is True, then that indicates you wish for that process to be run).

```
single = False
tenX = True # was originally False -- tied to UMI_bar
stranded_library = True # was originally False -- tied to tenX
bc_pattern = "C"*16 + "N"*12 # was originally commented out!
#####

#####

#####

## Toggles for deciding which steps in SICILIAN should be run ####
run_whitelist = True # was false
run_extract = True # was false
run_map = True
run_class = True
run_GLM = True
#####

out_path = out_dir + "{}".format(run_name)
⚡
✓ if not single:
    run_whitelist = False
    run_extract = False
```

-
- The original state of the sicilian.py script configured the pipeline to deal with non-10x data, so variables related to necessary steps to process 10x data were marked as false, and thus those processes were not run. Notably, the following variables were false in the original state of the script: tenX, stranded_library, run__whitelist, run_extract. Furthermore, the bc_pattern variable was commented out.
- The most interesting and puzzling thing to note is that the script disables the run_whitelist and run_extract processes from being run if the data you are working with is paired-end *even if you set them to be true in the couple of lines of code before*

nf-sicilian

Quick Start

1. Install [nextflow](#)
2. Install any of [Docker](#), [Singularity](#), [Podman](#), [Shifter](#) or [Charliecloud](#) for full pipeline reproducibility (please only use [Conda](#) as a last resort; see [docs](#))
3. Download the pipeline and test it on a minimal dataset with a single command:

```
nextflow run nf-core/sicilian -profile test,<docker/singularity/podman/shifter/charliecloud
```

Please check [nf-core/configs](#) to see if a custom config file to run nf-core pipelines already exists for your Institute. If so, you can simply use `-profile <institute>` in your command. This will enable either `docker` or `singularity` and set the appropriate execution settings for your local compute environment.

4. Start running your own analysis!

```
nextflow run nf-core/sicilian -profile <docker/singularity/podman/shifter/charliecloud/conda
```

See [usage docs](#) for all of the available options when running the pipeline.

- Note: The above is the extent of the documentation which is offered for getting started

Step 2

- Installing nextflow was ok, but step 2 is where issues started to arise. The UVA HPC does not grant sudo access to typical users (which is a good thing), but that means that installing certain things like Docker is not possible
 - Talked to HPC help, and they said to use Apptainer which can be loaded using **module load apptainer**

Step 3

- Step 3 was where the biggest issues occurred. The documentation makes it seem like the command is a one-and-done type of situation, but trying commands like `nextflow run nf-core -profile test,apptainer` or `nextflow run nf-core -profile test,conda` does not work
 - Also, I am still confused as to whether the comma that's included is a typo or not because I have not seen commas used in command line arguments before
 - First of all, I speculate that the argument in the place of `nf-core/sicilian` is used to create a string of a github page for nextflow to fetch configuration files from. In that case, the github repo associated with `nf-core/sicilian` that nextflow is trying to reach does not exist, leading to a 404. Based on this speculation, I tried different arguments of potential repository names/branches that might work. A logical thing to test would be `salzmanlab/nf-sicilian`.
 - **nextflow run salzmanlab/nf-sicilian -profile test** itself did not work, so I, alongside UVA HPC office hours staff, dug around and looked at the issues log of the GitHub repository. We arrived at this command: **nextflow run salzmanlab/nf-sicilian -profile test -r develop**, with the last

argument being the iteration of the tool. We previously tried different numbers, such as something along the lines of 1.3.3 which we found was an example in another page of the documentation, but whatever we tried did not work.