**Comprehensive Guide to Running squirrel for APOBEC3 Analysis**

squirrel (Some QUIck Rearranging to Resolve Evolutionary Links) is a powerful bioinformatics tool designed to streamline workflows for analyzing sequences, particularly in a phylogenetic context. This guide will walk you through its general usage, specific options, and how to generate interactive tree visualizations with APOBEC3 mutation annotations, incorporating a multi-step workflow.

For other details see [aineniamh/squirrel](https://github.com/aineniamh/squirrel)

**1. Understanding squirrel's Core Functionality**

At its heart, squirrel automates a series of common bioinformatics tasks, often used in viral or bacterial genomics:

* **Input Processing:** Handles input FASTA files.
* **Alignment:** Aligns your sequences against a reference or each other.
* **Quality Control & Masking:** Identifies and optionally masks problematic sites or regions (e.g., repetitive regions, ITRs, specific SNP calls).
* **Phylogenetics:** Runs phylogenetic inference (e.g., using IQ-TREE) to build a tree.
* **Mutation Reconstruction:** Reconstructs ancestral states and identifies mutations along tree branches.
* **APOBEC3 Analysis:** Specifically identifies and annotates APOBEC3-associated mutations (requires a dedicated flag).
* **Reporting & Visualization:** Generates various output files, including reports, static tree images, and interactive HTML trees.

**2. Installation (Brief)**

**(anything in blue should only need to be done once unless your environment changes or updates, you may need to redo)**

The recommended way to install squirrel is via conda or mamba, as it manages both Python and R dependencies.

Bash

# Example (adjust if squirrel is in a specific channel)

conda create -n squirrel\_env\_name -c bioconda -c conda-forge squirrel

conda activate squirrel\_env\_name

(Replace squirrel\_env\_name with your preferred environment name).

**3. Multi-Step squirrel Workflow Example**

This section outlines a common iterative workflow for squirrel analysis, incorporating your specific notes for detailed steps. We'll use generic names for files and directories.

First, **activate your squirrel environment** for all steps:

Bash

conda activate squirrel\_env\_name

* **Step 1: Initial Alignment, QC, and Masking**

This step performs initial alignment and quality control. You can specify a clade, but the alignment itself doesn't need to be pre-separated by clade.

Bash

squirrel your\_raw\_input\_sequences.fasta \

-o analysis\_results\_dir \

--clade cladei(i,ia,ib,ii,iia,iib) \

-qc

* your\_raw\_input\_sequences.fasta: Your starting FASTA file.
* -o analysis\_results\_dir: Outputs will go into this directory, this is optional and if not specified output will generate in current working directory.
* --clade cladeiib: Specifies the reference clade for alignment and subsequent analysis, other options: cladei,cladeia,cladeib,cladeii,cladeiia,cladeiib
* -qc: Runs sequence quality control, which will suggest problematic sequences and sites.

**Expected Output:** After this step, squirrel will generate several files within analysis\_results\_dir, including:

* your\_raw\_input\_sequences.report.html: An HTML report summarizing QC findings.
* suggested\_to\_exclude.csv: A CSV file listing sequences squirrel suggests you might want to remove.
* your\_raw\_input\_sequences.suggested\_mask.csv: A CSV file listing sites squirrel suggests you might want to mask.
* your\_raw\_input\_sequences.aln.fasta: The initial alignment FASTA formatted for squirrel.
* **Step 2: Refine Alignment with Masking and Exclusions (Iterative)**

Based on the output from Step 1, you can refine your alignment by applying masks and excluding sequences. This is often an iterative step where you "STOP and look at output" to decide on further reductions.

Bash

squirrel your\_raw\_input\_sequences.aln.fasta \

-o analysis\_results\_dir \

--clade cladeiib \

--additional-mask ./analysis\_results\_dir/your\_raw\_input\_sequences.suggested\_mask.csv \

--exclude ./analysis\_results\_dir/suggested\_to\_exclude.csv

* your\_raw\_input\_sequences.aln.fasta: **Note that we're now using the *aligned* FASTA from Step 1 as input.**
* --additional-mask ...suggested\_mask.csv: Applies site masks suggested by the QC step.
* --exclude ...suggested\_to\_exclude.csv: Excludes sequences suggested by the QC step.
* **Important:** After this step, you'll get new output files (e.g., your\_raw\_input\_sequences.aln.aln.fasta), which represent your more refined dataset. You might need to **repeat this step** if you perform manual curation of the suggested\_mask.csv or suggested\_to\_exclude.csv files, leading to new rounds of refinement.
* **Step 3: Run Phylogenetics and APOBEC3 Reconstruction**

This is the core analysis step where the phylogenetic tree is built and APOBEC3 mutations are identified.

Bash

squirrel your\_refined\_input\_sequences.aln.aln.fasta \

-o analysis\_results\_dir \

--clade cladeiib \

--run-apobec3-phylo \

# --outgroups your\_outgroup\_sequence\_id \ # run if you already included your outgroup of choice

# --include-background # Add this flag if your alignment doesn't have background sequences

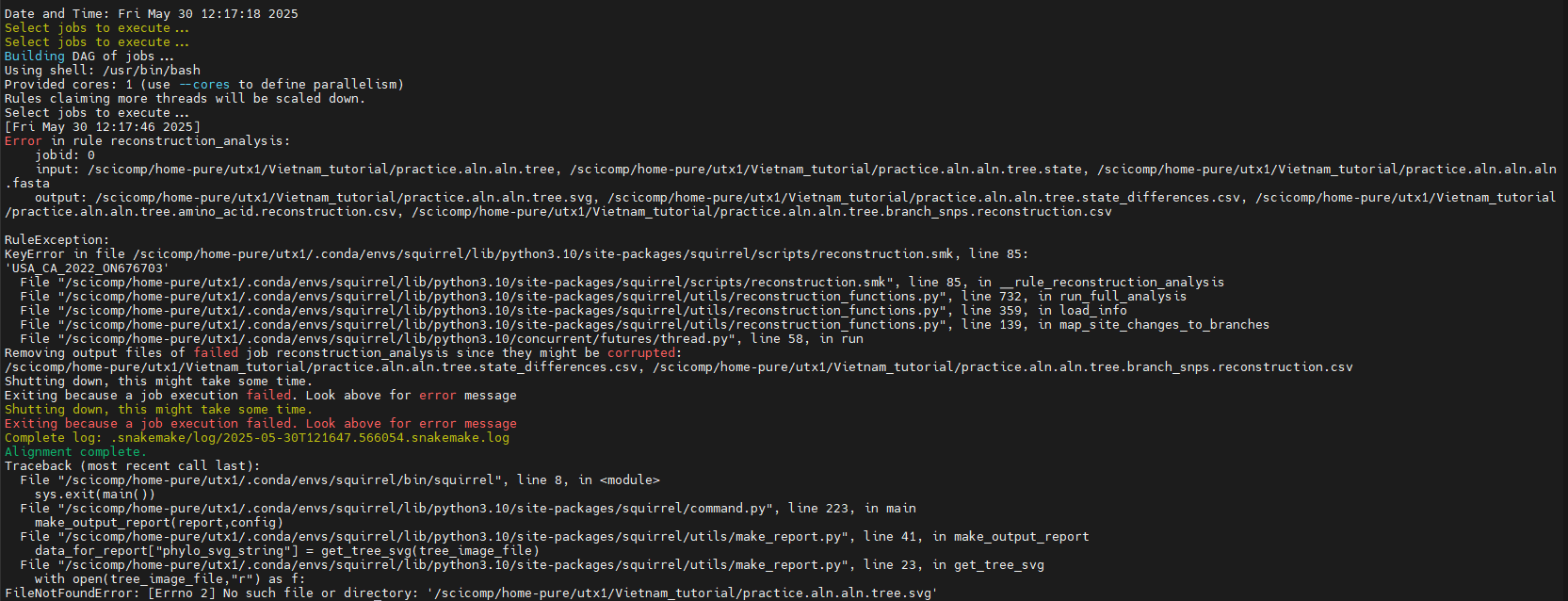
* your\_refined\_input\_sequences.aln.aln.fasta: **Use the most refined alignment FASTA from Step 2.** (The exact name will depend on how many times you re-ran Step 2; it might have multiple .aln suffixes).
* --run-apobec3-phylo (or its shorthand -a): **This is critical!** It triggers the full phylogenetics pipeline and, importantly, the APOBEC3 mutation reconstruction, which generates the amino\_acid.reconstruction.csv file needed for the interactive tree.
* --outgroups your\_outgroup\_sequence\_id: Specify the ID(s) of outgroup sequences in your alignment. These will be used for rooting the tree but pruned from the final displayed tree. Replace your\_outgroup\_sequence\_id with an actual sequence ID.
* --include-background: **Add this if your input FASTA does not contain sufficient background sequences for phylogenetic context.** squirrel will then automatically add a default background set relevant to your --clade choice.
  + - cladei: "KJ642615|human|Nigeria||1978"
    - cladeia: "KJ642615|human|Nigeria||1978"
    - cladeib: "KJ642613|human|DRC|Equateur|1970-09-01"
    - cladeii: "KJ642613|human|DRC|Equateur|1970-09-01"
    - cladeiia: "KJ642613|human|DRC|Equateur|1970-09-01"
    - cladeiib: "KJ642615|human|Nigeria||1978"

**Important Files generated in analysis\_results\_dir from this step:**

* your\_refined\_input\_sequences.aln.aln.aln.tree: The final phylogenetic tree file.
* your\_refined\_input\_sequences.aln.aln.aln.tree.amino\_acid.reconstruction.csv: **This is the crucial file for the interactive tree! It contains the apobec column.**
* your\_refined\_input\_sequences.aln.aln.aln.tree.branch\_snps.reconstruction.csv: Another mutation file, but typically **lacks the apobec column** and isn't used for the interactive tree in this context.

**Expected Errors:**

**You can by-pass if branch\_snps.reconstruction.csv file generated**

****

Bash

squirrel –tree-figure-only \

–fig-height ## \

--fig-width ## \

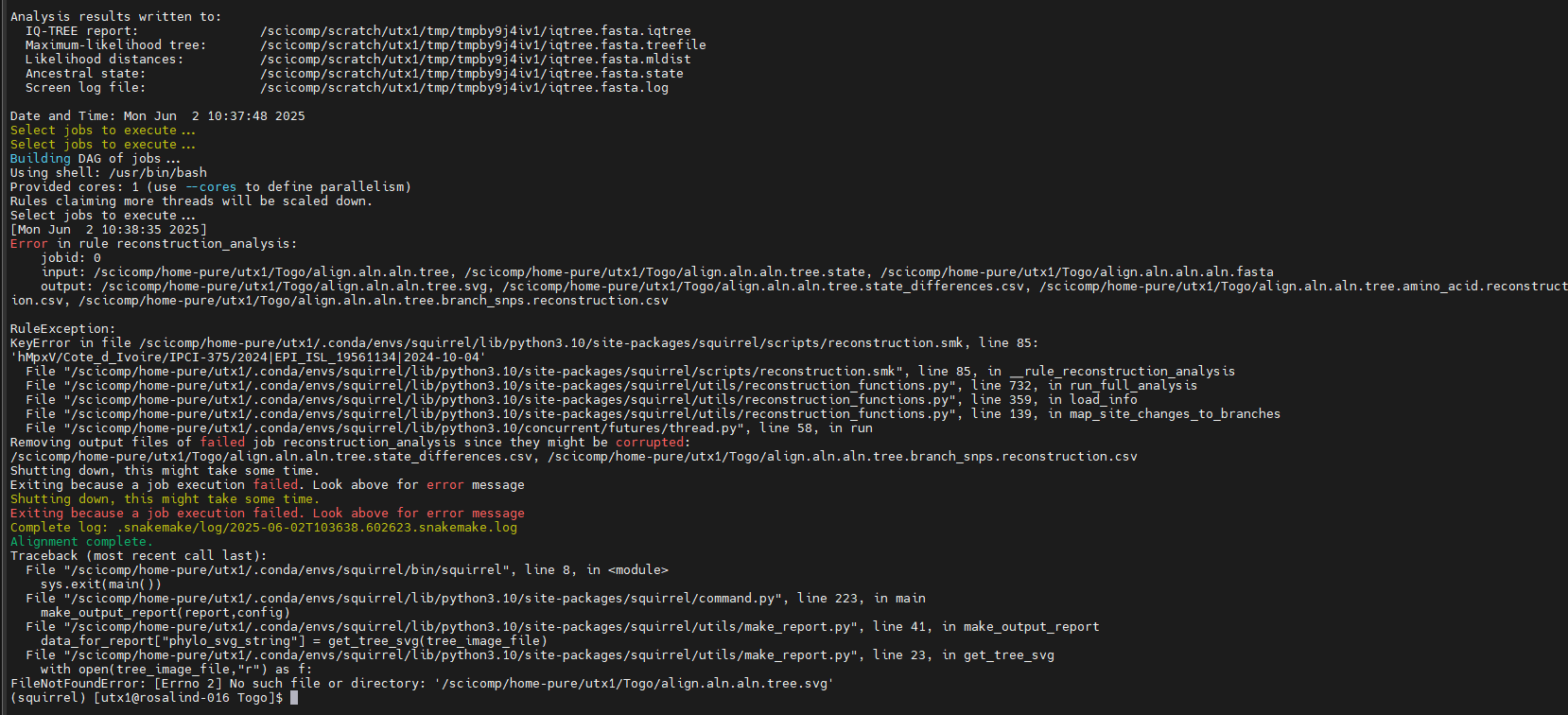
-brf ./ your\_refined\_input\_sequences.aln.aln.tree.branch\_snps.reconstruction.csv \

-tf ./ your\_refined\_input\_sequences.aln.aln.tree \

#--interactive-tree

Note if this does not work, there is likely a naming issue, probably due to an extra space in your alignment after one of the names

**If branch\_snps.reconstruction.csv file didn’t generate**

****

This typically occurs due to an ID mismatch across the input files for the APOBEC3 reconstruction analysis (e.g., .tree, .state, or .fasta files). Specifically, a sequence ID present in one file is unexpectedly absent from another required input file. May occur if you open a file and accidently edit.

Options:

* + Make IDs consistent: ensure exact ID consistency (including case, special characters, and absence of leading/trailing spaces) across all relevant input files, starting from the original FASTA headers
    - grep ‘missing\_id’ ./file\_path\_to/tree
    - grep ‘missing\_id’ ./file\_path\_to/tree.state
    - grep ‘missing\_id’ ./file\_path\_to/input\_align.aln.aln.fasta
  + Regenerate input files, and restart from the top, often this works best. Rerun without opening any intermediate files, this has often been worked for me.
    - Also consider renaming sequence that it can’t find, sometimes this works.
  + Regenerate input file with suggested outgroup and run with -- outgroup instead of --include-background, this also usually works for me.
  + Very rarely squirrel will drop a sequences in .aln or .aln.aln files, so you retry the --run-apobec step with the original alignment (no .aln)
* If you try all this, and it keeps failing, it’s likely that a sequence has a too high of N’s that squirrel can’t differentiate between the N’s generated when an INPUT.aln.fasta file is generated. If all else fails, remove sequences with high N values (>20,000 N’s usually is a good cutoff) prior to first step.
* **Step 4A: Generate the Interactive HTML Tree**

Finally, use the outputs from Step 3 to create your interactive visualization.

Bash

squirrel --tree-figure-only --interactive-tree \

-tf ./analysis\_results\_dir/your\_refined\_input\_sequences.aln.aln.tree \

-brf ./analysis\_results\_dir/your\_refined\_input\_sequences.aln.aln.tree.amino\_acid.reconstruction.csv \

--fig-width 23 \

--fig-height 33 \

--outfile interactive\_apobec\_tree.html

* --tree-figure-only: Instructs squirrel to only re-render a figure.
* --interactive-tree: **This enables the generation of the interactive HTML plot.**
* -tf ...your\_refined\_input\_sequences.aln.aln.aln.tree: Path to your final phylogenetic tree.
* -brf ...your\_refined\_input\_sequences.aln.aln.aln.tree.amino\_acid.reconstruction.csv: **Crucially, ensure this points to the amino\_acid.reconstruction.csv file** generated in Step 3, as this file contains the apobec column that the R script needs for coloring.
* --fig-width 23 --fig-height 33: Adjust these values to control the initial size of your tree figure.
* --outfile interactive\_apobec\_tree.html: This tells squirrel to save the interactive HTML output to this specific file name.

**Expected Output:**

* You should see Success! New tree figure written.
* An HTML file named interactive\_apobec\_tree.html (or whatever you named it) will be created in your current working directory (or the --outdir specified in *this* command, if you chose to redirect output for this command as well). You can open this file in any web browser.
* THIS HAS NEVER WORKED FOR ME

**6. Setting up the R Environment for Interactive Plots (One-Time Setup) for Manual Generation**

The interactive HTML tree feature relies on several R packages, some of which are from Bioconductor. Even if your squirrel conda environment is updated, these R packages might not be installed or configured correctly for the R interpreter that squirrel calls internally. This is a one-time setup per squirrel environment.

1. **Activate your squirrel conda environment:**

Bash

conda activate squirrel\_env\_name

1. **Launch R from within that activated environment:**

Bash

R

This ensures that any packages you install are associated with the R installation managed by your squirrel conda environment.

1. **Install Required R Packages:** Run the following commands in the R console.

# Install BiocManager (if not already installed) if (!requireNamespace("BiocManager", quietly = TRUE)) { install.packages("BiocManager", repos = "https://cran.r-project.org") }

# Install CRAN packages

install.packages(c("ape", "stringr", "scales", "htmlwidgets", "plotly", "dplyr"), repos = "https://cran.r-project.org")

# Install Bioconductor packages (ggtree and treeio are typically from Bioconductor)

BiocManager::install(c("treeio", "ggtree"))

```

\* When prompted to select a CRAN mirror, choose one **geographically close** to you.

\* Allow `BiocManager` to install any additional dependencies.

1. **Verify Package Installation (Optional but Recommended):** After installation, try loading each package to ensure they are available:

R

library(ape)

library(stringr)

library(scales)

library(htmlwidgets)

library(treeio)

library(plotly)

library(ggtree)

library(dplyr)

You should see version information or no output, but no errors.

1. **Exit the R console:**

R

q() # Type 'n' if it asks to save workspace image

**7. Interactive Plots Manual Generation**

Bash

Rscript ~/path/to/installed/squirrel/direcotory/squirrel/squirrel/scripts/interactive\_tree.R \

./your\_refined\_input\_sequences.aln.aln.aln.tree\

./your\_refined\_input\_sequences.aln.aln.tree.amino\_acid.reconstruction.csv manual\_interactive\_tree.html

* Rscript: Instructs cluster to only read and execute R script.
* ~/path/to/installed/squirrel/direcotory/squirrel/squirrel/scripts/interactive\_tree.R: **Points to script we want to run**
* your\_refined\_input\_sequences.aln.aln.aln.tree: Path to your final phylogenetic tree, DO NOT SPECIFY -TF HERE.
* your\_refined\_input\_sequences.aln.aln.aln.tree.amino\_acid.reconstruction.csv: **Crucially, ensure this points to the amino\_acid.reconstruction.csv file** generated in Step 3, as this file contains the apobec column that the R script needs for coloring, DO NOT SPECIFY -TF HERE.
* manual\_interactive\_tree.html: **Output file name for interactive .html file output.**