**Code and Software Submission Checklist Information**

1. System requirements:

-Script compiled and run in PyCharm Community Edition 2021.2.2

-Python Community version 3.8

-Pandas version 1.2.4, installed as part of package: Anaconda Individual Edition 2021.05

1. Installation guide:

-Typical install time for PyCharm/Python – 5 minutes

-Typical install time for Anaconda package – 5 minutes

-Once both are installed, establish Conda interpreter and install pandas within that environment

1. Demo library and output:

-Use demo.csv and WT-C-3A.csv; you will need both of these files in the same folder as the script to run the demo

-File demo.csv will direct the code to WT-C-3A sequence reads file, which is a library generated from wild type mouse embryonic fibroblasts that were a “no guide RNA” negative control. The vast majority of reads should be identified as germline sequence (no deletions or insertions). In my analysis of this library, 99.29% were classified as germline sequences (for confirmation, see “demo\_full\_compiled\_freq\_upload.csv” file).

-Script will output 3 files (I have uploaded a copy of each of these files from my own analysis of WT-C-3A for direct comparison. These are labeled as below with the addition of “\_upload”):

WT-C-3A\_jxn\_analysis (characterization of ever junction passing filters in the library)

WT-C-3A\_miscall\_results (characterization of all reads filtered out for ambiguitites, failed matches, or substitutions)

demo\_full\_compiled\_freq (master file containing all unique junction characterizations that passed the above filters, including junction counts, raw calculated frequency of that junction, and repair adjusted frequency of that junction (adjusted for mutation frequency, ignoring germline reads) for each individual library

-Printed script outputs should be as follows:

reads: 165130 (denotes number of raw reads provided, unfiltered)

jxns with ambiguity: 33 (reads containing ambiguous base calls)

failed matches: 1 (reads with failed matches upstream or downstream)

jxns with substitution: 774 (reads with substituted base)

final junctions: 164321 (reads passing all filters, used for frequency analysis)

fin jxn analysis - WT-C-3A

compiled 164321 (compiled jxns from all libraries)

unique 118 (unique jxns from the compiled list)

fin frequency calc - WT-C-3A (frequency calculation completed)

fin fin (output files successfully generated)

Process finished with exit code 0 (code complete with no errors)

-Running this single small demo library (165130 reads) should take less than one minute. Larger libraries or more libraries take exponentially more time as this drastically increases the number of unique junctions being counted (see “Instructions for use” document for more info). Estimated time for 1-10 libraries at approx. 200,000 reads per library is under 1 minute per library. estimated time for >10 libraries at approx. 200,000 reads per library is 1-2 minutes per library. Analysis for this paper included 15 libraries and approx. 2,800,000 reads. This analysis took approx. 25 minutes.

1. Instructions for use (brief). For more detailed instructions, reference “Instructions for use” document:

-To reproduce analysis on all 16 libraries used for this paper, do the following using “MEL\_PARP\_triplicate.csv” as the master input file. **Demultiplexed library sequence files can**  **be found using BioProject ID: PRJNA806204.**

-Define locus and cut site, one of three options (if using locus other than R26-MHD or R26-TINS, refer to options B or C):

A. Use R26-MHD (remove # signs to use lines 38-42) or R26-TINS (remove # signs on lines 45-49) accordingly, for demo library or complete re-analysis, use R26-MHD

B. Manually enter locus and guide sequence in same orientation as sequence reads (use lines 52-56 for NGG PAM orientation, lines 59-63 for CCN PAM orientation)

C. Manually enter left and right flank sequences of cut site, ensure orientation matches that of input reads (use lines 66-68)

-Generate .csv file (“master” .csv) listing all library files wanting to be analyzed, use “demo.csv” for the demo library (should take less than 1 minute), use “MEL\_PARP\_triplicate.csv” for full re- analysis (took approx. 25 minutes on my laptop)

-Place this .csv file as well as all library .csvs in same folder as script

-When prompted, input name of master .csv to identify list of library files for analysis

-Script will print benchmarks as it progresses through analysis and will print the following if analysis completed without error:

fin fin

Process finished with exit code 0

-If it prints Process finished with exit code 1, an error occurred along with a brief description of the error