This file contains step by step instructions about how to carry out the sequencing data analysis for the Cysteine mutant library. The parameters that need to be modified by the user are indicated by comments preceded by 'INPUT' in all the Perl scripts. So, users can ctrl+F for 'INPUT' and change values of those fields as per the requirement. It is recommended that the user does not change the folder structure, file names and extensions mentioned in the workflow. Users can dowload 'seq\_analysis' folder containing all the scripts from GitHub (https://github.com/skshrutikhare/cys\_library\_analysis), copy to their location of choice and run all the scripts. All the commands in this document are preceded by a '$' sign. Sample ‘.fastq’ files are also available on GitHub repository.

This workflow requires that the Water program is installed on your system for alignments of reads with the WT sequence (http://manpages.ubuntu.com/manpages/trusty/man1/water.1e.html). All scripts generate a LOG file indicating the time required to run the script. Some scripts generate summary\_counts files which provide some additional information about the data.

**Dividing sequencing output files into smaller parts**

Sequencing was performed using Illumina MiSeq platform. Sequencing output is in the form of two fastq files called R1 and R2.

On a Linux/Unix operating system let 'seq\_analysis' be the main directory where all the analysis scripts are executed. All the commands mentioned below can be executed within this directory.

$ mkdir seq1\_split

$ mkdir seq2\_split

Copy R1.fastq file to seq\_analysis/seq1\_split

Copy R2.fastq file to seq\_analysis/seq2\_split

Split the files into smaller chunks in order to make the processing faster. We have calculated the no. of lines such that we get 4 parts containing approximately equal number of lines. We suggest users to do the same so that they don't have to alter the filenames in the scripts.

For example,

$ cd seq\_analysis /seq1\_split

$ split -d -a 3 -l X R1.fastq initial1a

$ cd seq\_analysis /seq2\_split

$ split -d -a 3 -l X R2.fastq initial2b

X: number of lines in the split parts of the file

Use command 'man split' to get information about all the options.

**Initial filtering**

Before sending the data for sequencing, 3 random nucleotides were added at the start of the reads to ensure proper sequencing of reads with identical MID tags. These 3 N's are removed from the reads at this step.

$ cd seq\_analysis/seq1\_split/

$ perl initial\_filter\_fwd.pl

$ cd seq\_analysis/seq2\_split/

$ perl initial\_filter\_rev.pl

Ignore the file handler related warnings generated by this script.

**Separation into bins**

Data is separated into different bins depending on the MID tag and forward or reverse primer. After this step all reads in a given bin will have the same MID.

$ mkdir seq\_analysis/cys\_library/seq1\_split

$ mkdir seq\_analysis/cys\_library/seq2\_split

$ cd seq\_analysis/cys\_library/seq1\_split

$ perl bin\_all\_fwd.pl

$ cd seq\_analysis/cys\_library/seq2\_split

$ perl bin\_all\_rev.pl

**Combining the bins**

R1 and R2 files have both fwd and rev reads. This step combines all the forward reads from seq1\_split and seq2\_split folders having the same MID tag into one file and this is done for all reverse reads as well.

$ mkdir seq\_analysis/cys\_library/BINS\_ALL

$ cd seq\_analysis/cys\_library/BINS\_ALL

$ perl import\_bins.pl

$ sh import\_bins.sh

**Quality filtering**

Reads are filtered according to their quality scores. Minimum read length cutoff was taken as 75 and quality score cutoff was taken as Q20. If a low quality base was observed at a position, the read was trimmed at that position.

$ cd seq\_analysis/cys\_library/BINS\_ALL

$ perl filter\_scores\_Q20\_0-75.pl

A file named 'next\_badbase\_freq.txt' get created which contains frequencies of interval between first and next low quality base. If this interval is very large then user can modify the script not to trim the read at occurrence of first low quality base. This will result in longer reads. However, the authors do not recommend ignoring the occurrence of a low quality base.

A LOG file is created which lists some details about accepted and rejected reads.

$ mkdir seq\_analysis/cys\_library/BINS\_ALL/Q20\_0-75

$ mv \*.Q20\_0-75 Q20\_0-75/

**Conversion to FASTA format**

Reads are converted to FASTA format as they need to be aligned with the WT sequence.

$ cd seq\_analysis/cys\_library/BINS\_ALL/Q20\_0-75

$ perl Convert2Fasta.pl

$ mkdir fasta\_files\_fwd

$ mv \*.fasta fasta\_files\_fwd #This is a necessary step as revComp\_Convert2Fasta.pl will otherwise act on these .fasta files

$ perl revComp\_Convert2Fasta.pl

$ mkdir fasta\_files\_rev

$ mv \*.fasta fasta\_files\_rev

**Alignment with WT sequence**

$ mkdir seq\_analysis/cys\_library/BINS\_ALL/Q20\_0-75/Align\_input

$ cd seq\_analysis/cys\_library/BINS\_ALL/Q20\_0-75/Align\_input

$ mkdir fwd\_reads

$ mkdir rev\_reads

$ mv seq\_analysis/cys\_library/BINS\_ALL/Q20\_0-75/fasta\_files\_fwd/\* Align\_input/fwd\_reads/

$ mv seq\_analysis/cys\_library/BINS\_ALL/Q20\_0-75/fasta\_files\_rev/\* Align\_input/rev\_reads/

Also copy the WT sequence in fasta format to the 'fwd\_reads' and 'rev\_reads' folders

$ cd seq\_analysis/cys\_library/BINS\_ALL/Q20\_0-75/Align\_input/fwd\_reads/

$ perl /run\_water\_fwd.pl

$ cd seq\_analysis/cys\_library/BINS\_ALL/Q20\_0-75/Align\_input/rev\_reads/

$ perl run\_water\_rev.pl

All parameters were kept at default values except the Gap Opening Penalty, which was

increased to 20 while running the WATER program. Users can modify the command in the run\_water\_\*.pl script to make changes to the parameters.

The script creates wt\_seq.out file. As the script reads all .fasta files, the wt seq file is also read

and is aligned with itself. This file can be deleted as it is not required for analysis.

**Reformatting the alignment**

$ mkdir /seq\_analysis/cys\_library/BINS\_ALL/Q20\_0-75/Align\_input/fwd\_reads/fwd\_watout\_Q20

$ mkdir /seq\_analysis/cys\_library/BINS\_ALL/Q20\_0-75/Align\_input/rev\_reads/rev\_watout\_Q20

$ mv /Align\_input/fwd\_reads/\*.out fwd\_watout\_Q20/

$ mv /Align\_input/rev\_reads/\*.out rev\_watout\_Q20/

$ cd /seq\_analysis/cys\_library/BINS\_ALL/Q20\_0-75/Align\_input/fwd\_reads/fwd\_watout\_Q20

$ perl reformat\_watout\_fwd.pl

$ cd /seq\_analysis/cys\_library/BINS\_ALL/Q20\_0-75/Align\_input/rev\_reads/rev\_watout\_Q20

$ perl reformat\_watout\_rev.pl

**Combining the aligments**

In this step, alignmnets for overlapping reads will be combined. Reads which are not identical in the overlapping region are omitted. For non-overlapping reads, the gap between fwd and rev reads is filled by the WT sequence. Reads without a corresponding fwd or rev partner are used without any modifications, assuming that the missing sequence is WT.

$ mkdir /seq\_analysis/cys\_library/BINS\_ALL/Q20\_0-75/Align\_input/combine\_all

$ cd /seq\_analysis/cys\_library/BINS\_ALL/Q20\_0-75/Align\_input/combine\_all

$ perl combine\_all.pl

**Obtaining substitution information**

This step generates a table listing substitutions observed in each read. Single mutants were identified from this table using 'classify\_substitutions.pl' script.

$ cd /seq\_analysis/cys\_library/BINS\_ALL/Q20\_0-75/Align\_input/combine\_all

$ perl get\_substitution\_table.pl

$ perl classify\_substitutions.pl

**Calculation of CYS mutant frequencies**

$ cd /seq\_analysis/cys\_library/BINS\_ALL/Q20\_0-75/Align\_input/combine\_all

$ perl get\_CYS\_freq.pl

This script generates 'Freq\_CYS\_mutants.txt' file. This file lists the frequency of a CYS mutation at each position in the protein of interest at each experimental condition (corresponding to each MID tag). The counts from this file can be normalized with respect to total no. of reads at each MID and then used for comparison across MIDs. For our CcdB work, we also normalized the counts with respect to the expression MID. Details are explained in 'Deep Sequencing of Sorted Population' section of Najar et al., 2017 (doi:10.1016/j.str.2016.12.016).