# ScarMapper User Guide

ScarMapper is a Python program designed to analyze the DNA repair products at targeted loci that result from double strand breaks. In this case the double strand breaks are induced using CRISPR. This method has advantages over other methods such as CRISPResso and HiFiBR because it does not depend on aligning the sequence reads to each other when creating a consensus or to a reference to identify INDELS.

## Prerequisites for running ScarMapper

1. Linux OS, tested on RHEL 7.x, Scientific Linux 7.x, and CentOS 7.x. Will possibly run on a Mac OS, although it has not been tested. Will not run on Windows because ScarMapper uses Pysam to parse the reference FASTA file.
2. Ideally ≥45 Gb RAM. The amount of RAM depends on the number of reads being processed. This amount will allow processing of a FASTQ file containing ~8.0 million read pairs. The minimum recommended amount of RAM is 24 Gb. This will allow processing of ~4 million read pairs. The RAM does not scale equally with the number of reads.
3. >2 CPUs or threads. Processes one library per CPU/thread in parallel. The number of parallel jobs possible is CPUs or threads minus 1.
4. The minimum available drive space required is approximately 500 Mb. This does not include the space required for the FASTQ files.
5. Python ≥3.6
6. Python packages in no particular order, latest version of each
   1. scipy
   2. natsort
   3. pysam
   4. python-magic
   5. pathos
   6. numpy
   7. python-Levenshtein
   8. cython
   9. setuptools

## Installation

1. Download or clone ScarMapper from GitHub (https://github.com/pkMyt1/ScarMapper.git) to a location you have read/write access. Do not attempt to install using the setup.py file.
2. Follow these instructions to cythonize the sliding window library.
   1. From the command line, move into the scarmapper directory under the ScarMapper directory.
   2. Execute the command below. This command is also found in the setup.py file found in scarmapper.
      1. python3 setup.py build\_ext –inplace
   3. This will write a "build" and "scarmapper" directory in this location. In the "scarmapper" directory there will be a file similar to "SlidingWindow.cpython-37m-x86\_64-linux-gnu.so". Move this up one level.
3. Test installation by moving up to the ScarMapper directory and executing <python3 scarmapper.py> (without the <>). You should get the error message below.
   1. usage: scarmapper.py [-h] --options\_file OPTIONS\_FILE  
       scarmapper.py: error: the following arguments are required: --options\_file

## Required Files

ScarMapper requires the files listed below. Example the Master Index file and the Targets file can be found in the docs folder of ScarMapper. The formats and contents of each of these is discussed later.

1. Genome Reference Sequence in FASTA format.
   1. Obtain a reference genome of your choice such as GRCh38 or GRCm38. Make sure you also have the index file. If one was not available to download you will need to create it with Samtools. Place these files in a directory you have access to.
2. Index File
   1. Tab delimited text file. The excel template (ScarMapper Index Template.xlsm) is found in the docs folder. As a minimum, columns A through E are required for ScarMapper. Either copy and paste those columns into a text editor or export the sheet from the excel file as a tab delimited file. The rows that begin with a # are treated as a comment line.
3. Paired end FASTQ files.
   1. Place these somewhere you have read access to. ScarMapper is designed to use compressed, paired end, multiplexed FASTQ files as input. Uncompressed files or a demultiplexed FASTQ file will also work.
4. Master Index File.
   1. This file is derived by copying the first three columns from the sheet in the **ScarMapper Index Template.xlsm** file that corresponds to the to your sequencing platform. The file should be saved as a tab delimited text file. The table below shows the layout for Illumina dual indexing.

|  |  |  |
| --- | --- | --- |
| **# Index\_ID** | **Forward (D7s)** | **Reverse (D5s)** |
| D501+D701 | ATTACTCG | TATAGCCT |

1. Targets File
   1. This is a tab delimited text file. File ScarMapper\_Targets.txt, found in the docs folder, shows how the file is to be laid out. The first row must start with a #. Place this file in a location you have, at a minimum, read access. The same targeted region can contain multiple sgRNA sequences. These are treated as unique targets and go on individual lines. Each column is explained below.
   2. Target Name
      1. Unique name for the target. Avoid using special characters (`~!@#$%^&(\*}{).
   3. Chr
      1. Chromosome target is on.
   4. Start
      1. 5’ genomic position of PCR amplicon containing cut site.
   5. Stop
      1. 3’ genomic position of PCR amplicon containing cut site.
   6. sgRNA Sequence
      1. The 5’ → 3’ sequence of the sgRNA. This is not relative to the genomic sequence. The cleavage site will always be at the -3 position from the 3’ end.
   7. ReverseComp
      1. If the sgRNA sequence is not on the forward strand of the genomic DNA then answer “Yes”, otherwise answer “No”.
   8. Comments
      1. Any notes or other words of wisdom about this target.

## Options File

This must be a formatted as a tab delimited text file. In the docs folder you will find template options file run\_ScarMapper\_IndelProcessing.sh. The options file is a shell script file that starts the program and passes parameters to the program. It can also be used to pass commands to computing cluster load balancers such as SLURM or LSF. Each parameter is described below.

--IndelProcessing

Options are True or False. This exists to allow the indel search to be turned off in future versions.

--FASTQ1

Full path to FASTQ1

--FASTQ2

Full path to FASTQ2

--Ref\_Seq

Full path to the FASTA genomic reference. The reference index file must be in the same location as the genomic FASTA file

--Master\_Index\_File

Full path to the master index file described above.

--Index\_File

Full path to the pooled library index file.

--Target\_File

Full path to the target file.

--Working\_Folder

Full path to a working folder. This is were ScarMapper will write the output files.

--Verbose

Set verbosity level. Options are INFO, DEBUG, ERROR. For general runs leave this at INFO.

--Job\_Name

Provide a name for a run. This name will be incorporated into the output files simplifying identification. For this reason avoid special characters such as (`~!@#$%^&(\*}{).

--Spawn

Defines how many parallel jobs to create. Each library defined in the index file will be processes in its own job. Max setting should be number of CPUs – 1. Minimum value is 1.

--Demultiplex

Allowed values are True or False. Defines if demultiplexed FASTQ files are written. If the files are written, they are automatically compressed with gzip with a compression setting of 9.

--Species

Allowed values are Mouse or Human.

--Platform

Currently Illumina, Ion, Ramsden. This tells ScarMapper how to identify the indices.

--Atropos\_Trim

Allowed values are True or False. Use Atropos trimmer.

--Anchored\_Adapters\_5p

Full path to Atropos 5’ adapter file. Not required if --Atropos\_Trim is False.

--Anchored\_Adapters\_3p

Full path to Atropos 3’ adapter file. Not required if --Atropos\_Trim is False.

--Atropos\_Aligner

Atropos setting. Generally, use “adapter” option as seen in template file. Read Atropos documentation for additional options.

--NextSeq\_Trim

Atropos setting. Options are 0 or 1. Generally, use 1 as seen in template file.

--Adapter\_Mismatch\_Fraction

Atropos setting. For ScarMapper use 0.15 as seen in template file.

--Read\_Queue\_Size

Atropos setting. These are somewhat platform specific. Read Atropos documentation for additional information.

--Result\_Queue\_Size

Atropos setting. These are somewhat platform specific. Read Atropos documentation for additional information.

--N\_Limit

Atropos setting. These are somewhat platform specific. Read Atropos documentation for additional information.

--Minimum\_Length

Minimum read length. Default is 100 nucleotides.

--OutputRawData

Allowed values are True or False. Determines if additional files are written. These may or may not be useful.

## Output Files

The main output file will be labeled Job\_Name\_Index\_ScarMapper\_Frequency.txt. The Job\_Name component is from the --Job\_Name parameter described above. The “Index” component is from the first column in the --Index\_File. These files will be written in the location defined in the --Working\_Folder parameter. In all cases left and right is relative to the Cas9 cut site based on the sgRNA. If the sgRNA is reverse compliment as defined in the target file, then the sequence listed is the reverse compliment. The column contents are below.

1. Total
   1. Total number of times the pattern was observed in the sample.
2. Frequency
   1. Total number of times the pattern was observed normalized to the total number of sequence reads in the library that passed filters.
3. Left Deletions
   1. Nucleotides deleted to the left of the cut site.
4. Right Deletions
   1. Nucleotides deleted to the right of the cut site.
5. Deletion Size
   1. The size of the deletion is left + right + size of microhomology.
6. Microhomology
   1. Microhomology sequence.
7. Microhomology Size
   1. Size of microhomology.
8. Insertion
   1. Nucleotides inserted.
9. Insertion Size
   1. Size of insertion.
10. Left Template
    1. Nucleotides, based on insertions, that have homology within 50 nucleotides of the left junction.
11. Right Template
    1. Nucleotides, based on insertions, that have homology within 50 nucleotides of the right junction.
12. Consensus Left Junction
    1. Position of the left junction on the consensus sequence relative to the 5’ end.
13. Consensus Right Junction
    1. Position of the right junction on the consensus sequence relative to the 5’ end.
14. Target Left Junction
    1. Position of the left junction on the genomic target sequence relative to the 5’ end.
15. Target Right Junction
    1. Position of the right junction on the genomic target sequence relative to the 5’ end.
16. Consensus
    1. Consensus sequence.
17. Target
    1. Target regions sequence.