|  |
| --- |
|  |
| Völundr Pipeline  Version 0.5.0 |
| |  |  |  | | --- | --- | --- | | Dennis Simpson | 7/3/17 |  | |

Table of Contents

[REQUIREMENTS 2](#_Toc506452128)

[Bugs, Quirks, and Improvements 3](#_Toc506452129)

[Völundr Quick Start Instructions 4](https://d.docs.live.net/3ad448385884809f/Bioinformatics/Dev_Scripts/Volundr/Völundr%20Beta.docx#_Toc506452130)

[Quick Start Introduction 5](#_Toc506452131)

[Synthetic\_Lethal 5](#_Toc506452132)

[Single\_Cell 6](#_Toc506452133)

[Permutation\_Analysis 8](#_Toc506452134)

[Installation 10](#_Toc506452135)

[run\_Volundr.txt 10](#_Toc506452136)

[INDEX File 10](#_Toc506452137)

[Synthetic\_Lethal Analysis 11](#_Toc506452138)

[**Figure 1.** 11](https://d.docs.live.net/3ad448385884809f/Bioinformatics/Dev_Scripts/Volundr/Völundr%20Beta.docx#_Toc506452139)

[Options File 12](#_Toc506452140)

[Figure 2. 12](https://d.docs.live.net/3ad448385884809f/Bioinformatics/Dev_Scripts/Volundr/Völundr%20Beta.docx#_Toc506452141)

[library\_param\_date code.txt File 13](#_Toc506452142)

[library\_index sequence\_data.txt 13](#_Toc506452143)

[Index File 14](#_Toc506452144)

[Target File 14](#_Toc506452145)

[Single Cell Analysis 14](#_Toc506452146)

[Segment Analyzer 14](#_Toc506452147)

[shRNA Target Test 14](#_Toc506452148)

[Credits 14](#_Toc506452149)

[library\_param\_date code 14](#_Toc506452150)

[library\_index sequence\_data 16](#_Toc506452151)

Introduction

Völundr was the blacksmith to the Norse gods and had the reputation of being able to transform raw materials into anything. The Völundr bioinformatics pipeline is intended to analyze single cell sequencing data for copy number alterations and map shRNA or sgRNA distributions in cell populations for a modified synthetic lethal type assay. It is also for analyzing whole genome single cell sequencing data for copy number calls. With minor tweaking other functions could be added. This is a fully functional beta version. It runs without crashing and as intended. Certain features surrounding how it runs and the data it gives back could be better.

# Installation

Read through the requirements listed below and get a working copy of Python and the other programs installed first. Make sure you are doing all this on a box that is 64-bit architecture. As of February 15, 2018, the automated setup.py does not work. Currently this package resides in a private GitHub repository. You will need to contact Dennis Simpson ([dennis@email.unc.edu](mailto:dennis@email.unc.edu)) to obtain access. Once published the Völundr files will be moved to a public repository. From the private repository, download everything except Mímir.py and the mimir folder. Place these in a directory that you have read/write access to. I generally recommend in your home directory under a “scripts” folder or somewhere in the cloud. I currently run this from OneDrive using my local machine and from ~/scripts from a computing cluster.

## Requirements

* Designed to run on 64-bit Linux. Not tested on Mac. Don’t even bother trying on Windows.
  + Scientific Linux v7.4
  + RHEL v7.4
* System Requirements
  + ≥16 Gb Ram
    - 8 Gb is the minimum
  + ≥4 Processors
    - Running on a single core can take many days.
* Python v3.4 or greater. Should run on v3.3 but will not run on earlier versions.
* Python modules needed to be installed. Unless noted use the current version.
  + Cython
  + Pysam v0.13.0
    - Not compatible with earlier versions. Pysam v0.13 must be downloaded from GitHub and built. As of February 15, 2018, the new v0.14 had not been tested.
  + Pathos. Provides better pickling and multiprocessor functions.
  + Pathlib
  + Pybedtools
  + Pyfaidx
  + Matplotlib
  + Python- Levenshtein. This is the Levenshtein distance module implemented in python.
  + Python-magic v0.4.15 or greater
    - Will crash when using earlier versions. Allows Python to identify a file type by using libmagic. Make sure python-libmagic is not installed before installing this one. The two are not compatible.
  + Pyensemble v1.1.0 or greater. This is the python API that allows access to Ensemble. There is a bug in this version. See note below for which file to edit to fix.
  + Natsort
  + Numpy
  + Scipy
  + Sortedcontainers
  + setuptools
* HTSlib v1.6. Make sure this is matched with the correct version of Pysam.
* Samtools v1.6. Make sure this is matched with the correct version of Pysam.
* BCFtools v1.6. Make sure this is matched with the correct version of Pysam.
* BWA
  + Appropriate indexed references as well.
* Bowtie2
  + Appropriate indexed references as well.
* BedTools
* Muscle Aligner
  + Install in your path, usually usr/bin.
  + Download from http://www.drive5.com/muscle/

**IMPORTANT:**

**There is a known bug in pyensembl v1.1.0 that does not allow use of database versions greater than 87. The discussion is found in issue #188. Edit ensembl\_release\_versions.py line 18 to read MAX\_RELEASE\_VERSION = 91 to correct this.**

**Pysam v0.13.0 must be downloaded from GitHub and built, DO NOT use pip install.**

## Bugs, Quirks, and Improvements

**Völundr**

* Entry to package. Works well but contains some dead code. Possibly make self-executable.

**Setup**

* Setup file does not work.
* Online documentation lacking



|  |  |
| --- | --- |
| The University of North Carolina at Chapel Hill | Völundr Quick Start Instructions |

## Quick Start Introduction

This section is intended as a reference for users that are familiar with this package and with running programs on Linux. I recommend using Seiðr to write the options file however the minimal options required for each module is listed here. When running on Killdevil at UNC the recommended start commands are

bsub -q week -M 5 -x -n 14 -R span[hosts=1] python3/nas02/home/<usr\_path\_to>/scripts/Volundr/Volundr.py

--options\_file /nas02/home/<usr\_path\_to>/scripts/Volundr/run\_Volundr\_Killdevil.txt

The 5 Gb of RAM defined by the -M 5 should be sufficient for most runs. The maximum number of cores currently allowed on Killdevil with the standard login is 14.

## Synthetic\_Lethal

This module takes a single, multiplexed FASTQ file as input and determines the number of targets in the FASTQ file grouped by index. The required options to run this module are listed below.

#Options File to run module: --Synthetic\_Lethal

#File generated on 2017-06-23T12:26:23.604

--Segment\_Analyzer False

--Bin\_Generator False

--Single\_Cell False

--Synthetic\_Lethal True

--process\_target False

--shRNA\_Target\_Test False

--Control\_Processing False

--Permutation\_Analysis False

--Heatmap False

--FASTQ\_QC False

--fastq1 /Sequencing/Wanjuan\_2017a\_Q15.fastq.gz

--index\_file /Google\_Drive/scripts/indices/2017a\_barcodes.bed

--target\_file /Google\_Drive/scripts/Targets/mDDR\_sgRNA.bed

--compress False

--rcomp False

--delete\_fastq False

--min\_length 100

--target\_length 20

--target\_start 21

--target\_padding 2

--library DeBug

--spawn 1

--prog\_check 1000

--index\_mismatch 2

--target\_mismatch 2

--expected\_position 112

--position True

--smith\_waterman False

--muscle False

--fastq1

Give the full path to the FASTQ file. Keep in mind that the temporary and final output files are written to this location.

--index\_file

Give the full path to the index file.

--target\_file

Give the full path to the target file.

--compress

If saving the demultiplexed FASTQ files set this to true to reduce final disk space. This takes several hours to complete.

--rcomp

Set True for reverse complimenting the target. Needed for CRISPR.

--delete\_fastq

If not saving the demultiplexed FASTQ files set this True in order to delete them.

--min\_length

Minimum length of a sequence read to be considered a good read. Only integers are allowed.

--target\_length

How long of a target to extract from the target file? Only integers are allowed. **Currently shRNA should be 21 and CRISPR should be 20.**

--target\_start

The sequence in the target file is from the oligonucleotide used to construct the clone. Where in this sequence does the actual target begin? Only integers are allowed. **Currently shRNA should be 60 and CRISPR should be 21.**

--target\_padding

How many nucleotides at the beginning and end of the target to include as part of the target? Only integers are allowed. **When using --position this should generally be 2. If using --smith\_waterman or**

**--muscle use 6 or 7.**

--library

Provide a name that is used for an analysis run. Used to make the file names unique.

--spawn

Number of parallel jobs to run. This value must be ≥1.

--prog\_check

How often to have the program output information to stdout. Used to confirm that program is still running. Does not behave as intended on killdevil. Recommend using a value of >300000. Setting value to 0 will turn this off.

--index\_mismatch

How many nucleotide mismatches to allow for index identity? Must be an integer. Recommended value is ≤2.

--target\_mismatch

How many nucleotide mismatches to allow for target identity? Must be an integer. Recommended value is 2 or 3.

--expected\_position

Where in the sequence read is the target expected to be? Must be an integer. **Currently for shRNA this should be 65. For CRISPR this should be 112.**

--position

Base target location on expected position.

--smith\_waterman

Find target using Smith-Waterman alignment.

--muscle

Find target using Muscle aligner.

## Single\_Cell

**This section is not complete!!**

This module analyzes the segment copy file generated from Ginko or other segmentation programs. Generates data files with information about copy data. The required options to run this file are listed below

--seg\_copy\_file

#Options File to run module: --Single\_Cell

#File generated on 2016-08-15T11:10:45.250

--Segment\_Analyzer False

--Single\_Cell Talse

--Synthetic\_Lethal Frue

--process\_target False

--shRNA\_Target\_Test False

--Control\_Processing False

--Permutation\_Analysis False

--seg\_copy\_file /path/to/segCopyFile.txt

--cell\_line WTNEU

--library some\_name

--spawn 14

--prog\_check 300000

Full path to segment copy file.

## Permutation\_Analysis

This module will do various permutation analysis runs on the segmentation data. The output is useful for determining statistical significance of the intersects of the segmentation data with chromosomal features. The required options file information to run this module is shown below

#Options File to run module: --Permutation\_Analysis

#File generated on 2016-08-15T13:45:12.432

--Segment\_Analyzer False

--Single\_Cell False

--Synthetic\_Lethal False

--process\_target False

--shRNA\_Target\_Test False

--Control\_Processing False

--Permutation\_Analysis True

--seg\_copy\_file /path/to/segCopyFile.txt

--target\_file /path/to/targetFile.bed

--cell\_name WTNEU

--library Some\_Library

--spawn 1

--freq\_iterations 20000

--repeat\_count 1

--bed\_group 2

--prog\_check 100000

--exclude\_chrY True

--write\_map\_file True

--seg\_perm\_file True

--breakpoint\_dist\_file True

--breakpoint\_chrom\_dist\_file True

--target\_perm\_file False

--seg\_copy\_file

Full path to the segment copy file.

--target\_file

Full path to the target file.

--cell\_name

Cell group name.

--library

Provide a name that is used for an analysis run. Used to make the file names unique.

--spawn

Number of parallel jobs to run. This value must be >0 and ≤ CPU’s-1.

--freq\_iterations

How many iterations to test for each permutation repeat?

--bed\_group

How many segments to combine when doing permutations. Integer >0.

--prog\_check

How often to have the program output information to stdout. Used to confirm that program is still running. Does not behave as intended on killdevil. Recommend using a value of >300000. Setting value to 0 will turn this off.

--exclude\_chrY

True or False. Exclude ChrY from analysis.

--write\_map\_file

Write a file mapping the targets to the given segment file.

--seg\_perm\_file

Write a file with the permutation data.

--breakpoint\_dist\_file

Write a file with the break point distribution data.

--breakpoint\_chrom\_dist\_file

Really cannot remember.

--target\_perm\_file

Permute the target data. This does not work as intended and will likely be removed in the future.

# Installation

To install the program extract Völundr folder somewhere that you have full access. The program is run from the command line using the following;

python3.x /path/to/Volundr.py --options\_file path/to/run\_Volundr.txt

While it is not strictly necessary, I recommend keeping the run\_Volundr.txt options\_file in the same directory as the program. On the UNC research computing cluster, while a submitted job is pending the files for that job have not been read into memory. Any changes to those files will then give unexpected results. In this environment I recommend saving the options file with different name to avoid unexpected results. Because the program writes so many new files it is recommended that each FASTQ or BAM file to be analyzed reside in their own directory. The output file names written by the program should be unique thus preventing naming collisions but that has not been extensively tested.

## run\_Volundr.txt

This text file is used to pass all the option variables to the program. An all-inclusive options file is not recommended because it is too easy to make mistakes. This file is best written using the stand-alone Java program Seiðr or by using a text editor to modify a module specific options file. Seiðr will only write the options that are required for a given module. If you do not have access to Seiðr, a basic file can be regenerated by copying the block of text from the chapter describing the module you wish to run into a text editor and saving the file as “run\_Volundr.txt”. The options file can be saved anywhere however I recommend placing it in the same folder with the Python scripts. This is a tab delimited two column text file. Column one is the option and column two is the value. Blank lines are there to make it easier for humans to read and are ignored by Völundr. Any line or variable that begins with “#” is a comment. Comments can be at the beginning of a line, resulting in the line being ignored or at the end of a column. When editing manually it is generally simpler to include all lines that begin with “--“. Some of them can have no value in column two. The options can be listed in any order. All options and variables in this file are case sensitive.

## INDEX File

A file containing all the indices must always be provided as a tab delimited text file. Even if the FASTQ file is not multiplexed this file is still required with the index. The file must contain at least three columns. Additional columns are permitted but are ignored. Any line beginning with a “#” is treated as a comment and ignored. Blank lines are also ignored. The structure of this file must be column 1 containing an index name or definition. Column 2 is for the name of a group. Column 3 must contain the sequence of the index in all caps. Only G, A, T, or C are allowed as nucleotides. The first two lines of an index file that might be used for single cell sequencing are shown below;

|  |  |  |
| --- | --- | --- |
| #Oligo\_Name | Cell\_Pool | Oligo\_Sequence |
| pool4\_1 | GGCT4 | ATTACTCTATAGCC |

Ambiguous nucleotides are not allowed. The sequence of the indices must be unique. The program will check this and exit with an error if they are not. This file can reside anywhere.

# Synthetic\_Lethal Analysis

This module has an absolute requirement for the multiplexed FASTQ file or uBAM file. If the sequencing facility demultiplexed the FASTQ file and you have no access to the multiplexed version, then you will need to run the Add Index Module on each FASTQ file. This will write the index back at the beginning of the sequence and write a new FASTQ file. Once the index has been added back to all of the FASTQ files, run cat to combine them all into a single file and, to save space, compress everything again with gzip. Once the options file, index file, and target file are all set for the run, execute the program by typing

python3.x /path/to/Volundr.py --options\_file path/to/run\_Volundr.txt

at a command prompt. This module takes about 36 hours in real time to complete a run IF no files are being compressed. This time translates into approximately 150 cpu hours. The default que on the UNC Killdevil cluster will only run for 24 hours of real time. I recommend using the following settings when defining the que and processors;

bsub -q week -x -n 13 -R span[hosts=1]

### **Figure 1.**

#Options File to run module: --Synthetic\_Lethal

#File generated on 2017-06-23T12:26:23.604

--Segment\_Analyzer False

--Bin\_Generator False

--Single\_Cell False

--Synthetic\_Lethal True

--process\_target False

--shRNA\_Target\_Test False

--Control\_Processing False

--Permutation\_Analysis False

--Heatmap False

--FASTQ\_QC False

--fastq1 /Sequencing/Wanjuan\_2017a\_Q15.fastq.gz

--index\_file /Google\_Drive/scripts/indices/2017a\_barcodes.bed

--target\_file /Google\_Drive/scripts/Targets/mDDR\_sgRNA.bed

--compress False

--rcomp False

--delete\_fastq False

--min\_length 100

--target\_length 20

--target\_start 21

--target\_padding 2

--library DeBug

--spawn 1

--prog\_check 1000

--index\_mismatch 2

--target\_mismatch 2

--expected\_position 112

--position True

--smith\_waterman False

--muscle False

The program will output a file with the name “library\_param.txt” and one file for each index with the name “library\_index sequence\_data.txt”. You may also notice a file called “library\_multiprocessor\_tmp\_data.txt” being written. This a temporary file containing the number of reads that have no identifiable target for each index. The data in this file is added to the parameter file when all the parallel jobs have completed and then is removed.

## Options File

The options file parameters required for this module are shown in Figure 1. The options through FASTQ1 should be self-explanatory.

--fastq1

Give the full path to the FASTQ file. Keep in mind that the temporary and final output files are written to this location.

--index\_file

Give the full path to the index file. The format of the [index file](#_INDEX_File) has been explained elsewhere.

--target\_file

Give the full path to the target file. The format of the target file is a three column, tab delimited text file similar to the index file. The first column is the probe or oligo ID. This column should be unique but is not used by the program. The second column should be a unique name that identifies the target such as geneName\_1. The third column should be the oligonucleotide sequence used to clone the target. This sequence must be unique in the file. If it is not the program will exit with an error identifying the duplicated sequence (Figure 2).

### Figure 2.

# mouse\_DDR\_ID Unique\_ID Oligo\_Seq

mDDR\_1 Aicda\_1 GTGGAAAGGACGAAACACCGACCATTTCAAAAATGTCCGCGTTTTAGAGCTAGAAATAGCAAGTTAAAATAAGG

--compress

If set to “True” the program will run gzip -9 on the demultiplexed FASTQ files after the analysis is done. This adds about 24 hours of world time to the run however the data can be downloaded while this is happening. If this is “True” then --delete\_fastq must be “False”.

--rcomp

When “True” the program will reverse-compliment the sequence derived from the the FASTQ read. This is required to be “True” for the shRNA target files used Aug. 2016.

--delete\_fastq

When “True” the demultiplexed FASTQ files are removed from the system after analysis. If this is “True” then --compress must be “False.”

--min\_length

Minimum length of sequence read from FASTQ file to attempt analysis on. This is very important when using sequences from an Ion platform.

--target\_length

How long of a target to extract from the target file? Only integers are allowed. **Currently shRNA should be 21 and CRISPR should be 20.**

--target\_start

The sequence in the target file is from the oligonucleotide used to construct the clone. Where in this sequence does the actual target begin? Only integers are allowed. **Currently shRNA should be 60 and CRISPR should be 21.**

--target\_padding

How many nucleotides at the beginning and end of the target to include as part of the target? Only integers are allowed. **When using --position this should generally be 2. If using --smith\_waterman or**

**--muscle use 6 or 7.**

--library

Provide a name that is used for an analysis run. Used to make the file names unique.

--spawn

Number of parallel jobs to run. This value must be >0 and ≤ CPU’s-1.

--prog\_check

How often to have the program output information to stdout. Used to confirm that program is still running. Does not behave as intended on killdevil. Recommend using a value of >300000. Setting value to 0 will turn this off.

--index\_mismatch

How many nucleotide mismatches to allow for index identity? Must be an integer. Recommended value is ≤2.

--target\_mismatch

How many nucleotide mismatches to allow for target identity? Must be an integer. Recommended value is 2 or 3.

--expected\_position

Where in the sequence read is the target expected to be? Must be an integer. **Currently for shRNA this should be 65. For CRISPR this should be 112.**

--position

Base target location on expected position.

--smith\_waterman

Find target using Smith-Waterman alignment.

--muscle

Find target using Muscle aligner.

### library\_param\_date code.txt File

An example of this file is shown at the end of this document ([library\_param\_date code](#_Figure_1)). The file contains tab delimited information and is designed to look best when viewed with something like Excel. The first part of the file contains information about the program run that is primarily gleaned from the options file. The text in *italics* was added as part of the figure and thus is not found in the actual output. The second part of the file contains information about the reads in the file grouped by the index sequence. The index mismatch counts are only done on the full reads as is the shRNA search.

### library\_index sequence\_data.txt

An example of this file is shown in at the end of this document ([library\_index sequence\_data](#_Figure_2)). There will be one of these for each index found during the shRNA module run. The current version of the Synthetic Lethal module sorts the target list by the target name before printing. This should make all the output files be in the same order. The data structure of these files should be fairly self-explanatory. The first column is the target sequence used for the search. This is extracted from the target file based on the parameters given. The next column is the target name. In general, this should be a gene name with a number as shown. The next columns show the number of times a given target was seen and a breakdown of the counts based on the number of mismatches.

### Index File

Three column tab delimited text file. Additional columns are ignored by the program. Column 1 and column 2 are combined by the program to help uniquely identify the index. Column 3 is the sequence of the index. Column 3 must be unique. The file can contain blank lines and any line that begins with a “#” is a comment that is ignored when parsed. A comment may also be placed in the third column AFTER the sequence. The values in column

### Target File

This is a three column tab delimited text file. The file can contain blank lines and any line that begins with a “#” is a comment that is ignored when parsed. Column 1 can be anything; it is not used by the program. Column 2 should be a unique name for the shRNA or the sgRNA. Column 3 is the sequence. The sequence can be either the oligo itself or the entire fragment cloned in. The values in columns 2 and 3 must all be unique. The program will exit and report an error with any of the targets are duplicated. An example shRNA target file is below;

|  |  |  |
| --- | --- | --- |
| #EntrizID | Protein Target | Oligo\_Sequence |
| 11545 | Parp1\_2550 | TGCTGTTGACAGTGAGCGCGGAAGTGATCGATATCTTCAATAGTGAAGCCACAGATGTATTGAAGATATCGATCACTTCCATGCCTACTGCCTCGGA |

## Single Cell Analysis

This module takes as input a demultiplexed FASTQ file or a mapped BAM file. If a demultiplexed FASTQ file is given the index is moved from the sequence to the header, and the whole genome primer is trimmed from the end of the sequence. This is all written to a new FASTQ file. The new file is then aligned with Bowtie2. The resulting BAM file or if a BAM file is given as input is then demultiplexed and the demultiplexed files are converted to BED files. The BED files can be gzipped or left as they are. Although it takes longer, I recommend gzipping the BED files. Saves time when transferring them to Ginko.

## Segment Analyzer

The segment analyzer takes the segment copy file from the Ginko website and returns data about copy number variations. This module is fairly fast and does not require any special settings to run.

## shRNA Target Test

This module takes as input the shRNA target file and compares each potential shRNA to all the others in the list using the internal match-maker functions. The program will out put a file showing how many times each is found and the number of mismatches. It is hard coded to use up to three mismatches.

# Credits

Cover page image from http://www.cancer.gov/research/nci-role/bioinformatics.

# library\_param\_date code

This is a parameter output file from the shRNA module. This is one of the output files for the [Synthetic Lethal](#_library_param_date_code.txt_File) analysis.

Running: Volundr module shRNA.py v1.0

Start\_Time: Sun Jan 3 00:00:00 2016

Stop\_Time Sun Jan 3 00:00:00 2016

FASTQ\_File: indexed.fastq.gz

INDEX\_File: /nas02/home/d/e/dennis/scripts/Volundr/shRNA\_INDEX\_list.txt

Target\_File: /nas02/home/d/e/dennis/scripts/Volundr/shRNA\_list.txt

Index\_Mismatches 1

Target\_Mismatches 2

Target\_Padding 2

shRNA\_Target\_Start 50

Min\_Read\_length 60

shRNA\_Start 60

shRNA\_Length 21

Total\_Reads: 60848387 *“Read count in FASTQ, uBAM, or BAM file”*

Indexed\_Reads: 123456 *“Total number of reads that contained an identifiable index”*

No\_Index\_Count: 0 “Total number of reads that did not contain an identifiable index”

No\_Index\_Short: 0 *“Number of reads without an index that were shorter than Min\_Read\_length”*

No\_Index\_Full: 0 *“Number of reads without an index that were longer than Min\_Read\_length”*

Index Cell Total Short\_Reads Full\_Reads 0\_mismatches 1\_mismatches no\_shRNA\_target

GTGACCCAA shRNA\_pool\_control3 0 0 0 0 0

TCCTCGAAT 26\_2ATLD\_8 2284959 13057 2271902 2271902 0 789347

TACCAAGAT SV\_MEF\_DL\_8 1248086 9373 1238713 1238713 0 442247

TCTAGAGGT Ku70\_0 195171 1571 193600 193600 61557

TCTAACGGA PolQ\_4 3457407 32141 3425266 3425266 0 1281953

CAGAAGGAA SV\_MEF\_0 2421566 107974 2313592 2313592 0 1336175

TAGGTGGTT PolQ\_0 4451557 27144 4424413 4424413 0 1488188

GTGACTCAA Ku70\_PolQ\_4 0 0 0 0 0

TCTGGATGA Ku70\_4 0 0 0 0 0

GTGACGTAA Ku70\_8 0 0 0 0 0

TTGGAGTGT PolQ\_8 745467 7397 738070 745467 0 318407

GAGAACGAA shRNA\_pool\_control1 834686 4998 829688 834686 0 282339

GTGCAACTT shRNA\_pool\_control2 0 0 0 0 0

TTCCGATAA ATLD1\_4 2301657 13586 2288071 2301657 0 748444

AAGAGGATT SV\_MEF\_DL\_4 2805535 121215 2684320 2805535 0 1510355

GTAACGTAA Ku70\_PolQ\_8 0 0 0 0 0

TGAGCGGAA 26\_2ATLD\_0 2075136 12412 2062724 2075136 0 660249

CTGCAAGTT SV\_MEF\_4 17163799 672979 16490820 17163799 0 9332466

CTAAGGTAA SV\_MEF\_8 8070570 385059 7685511 8070570 0 4433268

TAAGGAGAA SV\_MEF\_DL\_0 9707521 845313 8862208 9707521 0 5289600

TTCGTGATT ATLD1\_0 2090839 13093 2077746 2090839 0 746333

CTGACCGAA 26\_2ATLD\_4 994431 6195 988236 994431 0 334121

# library\_index sequence\_data

This is a data output file from the shRNA module. Only part of the file is shown. The example this is taken from has over 3000 shRNA targets. This is one of the output files for the [Synthetic Lethal](#_library_param_date_code.txt_File) analysis.

Target\_Key Target Counts 0\_mismatches 1\_mismatches 2\_mismatches

TTACACTATGAATATCTCGCA Ube2a\_1204 175 166 9 0

CTAGAACCTGACTGACTACTG Fen1\_437 1 0 1 0

TCACATCTCTGTTTCTTCCTC Apex2\_103 4196 4010 186 0

TCTGTTTCACTTTGAAGGTGT Tdp1\_1699 36 35 1 0

CTGTTAACTGAATACCACTCT Hmgb1\_1076 6817 4307 2510 0

GAAATTCCAATATGAAGATGA Telo2\_1670 3774 3300 474 0

AGACTATGAAGATGGTGTGGG Ddb2\_354 3553 2 3551 0

CAGATTGTAATTCATGGTGCT Dclre1b\_1931 13 1 12 0

GTAAAGTGAAGAATCTCTGTC Prpf19\_1785 5598 8 5590 0

AAAGACTTGCATAGGTCTGTG Tinf2\_17 2492 17 2475 0

TAAGTAGAACATGAATGTGAA Carm1\_2844 1 1 0 0

TAAAACTGCACTTCCTGCTGA Atrip\_8 525 494 31 0

ACAATAAAGTTTAGCTTCTGG Med10\_162 11935 6532 5403 0

TCTTCATGATAATCTCCTGCG Mutyh\_30 603 564 39 0

TCAATCACATTGAGGAATCGT Kdm4b\_2648 0 0 0 0

AAGAGAATAGCTCAACAGTCT Neil1\_153 0 0 0 0

TTCATGAAGAAAATCTTGCAC Atr\_3860 7342 2126 5216 0

TATAGACTACAACAGCCGTAT Rbbp8\_3657 234149 216881 17268 0

AAATCATTACTGTCATCAGCA Top2b\_4588 1 1 0 0

ACTGTACCTGCCTACCTTGTA Ung\_1953 1 1 0 0

TCATTTGAGAGCTCTAATGGG Eme1\_2143 2153 2085 68 0

TCAATGTCAACATGGCTTTCA Hdac4\_1164 147 137 10 0

ACTTTACACAGAAGGAGACGA Fance\_1243 2007 1896 111 0

TGTTAACATGAATGTTTCTGC Pcna\_98 10973 7437 3536 0

TAAACAACTTATTTAGCTACG Slx4\_5549 6597 6354 243 0

TGTAGAGCTTGATGGTACCAT Top3b\_2243 55 5 50 0

TACGTGACTGTTGCATCTTGA Arid4a\_5302 16 15 1 0

ACATTAATAGTTTCAGACAAT Gtf2h5\_1114 7142 4097 3045 0