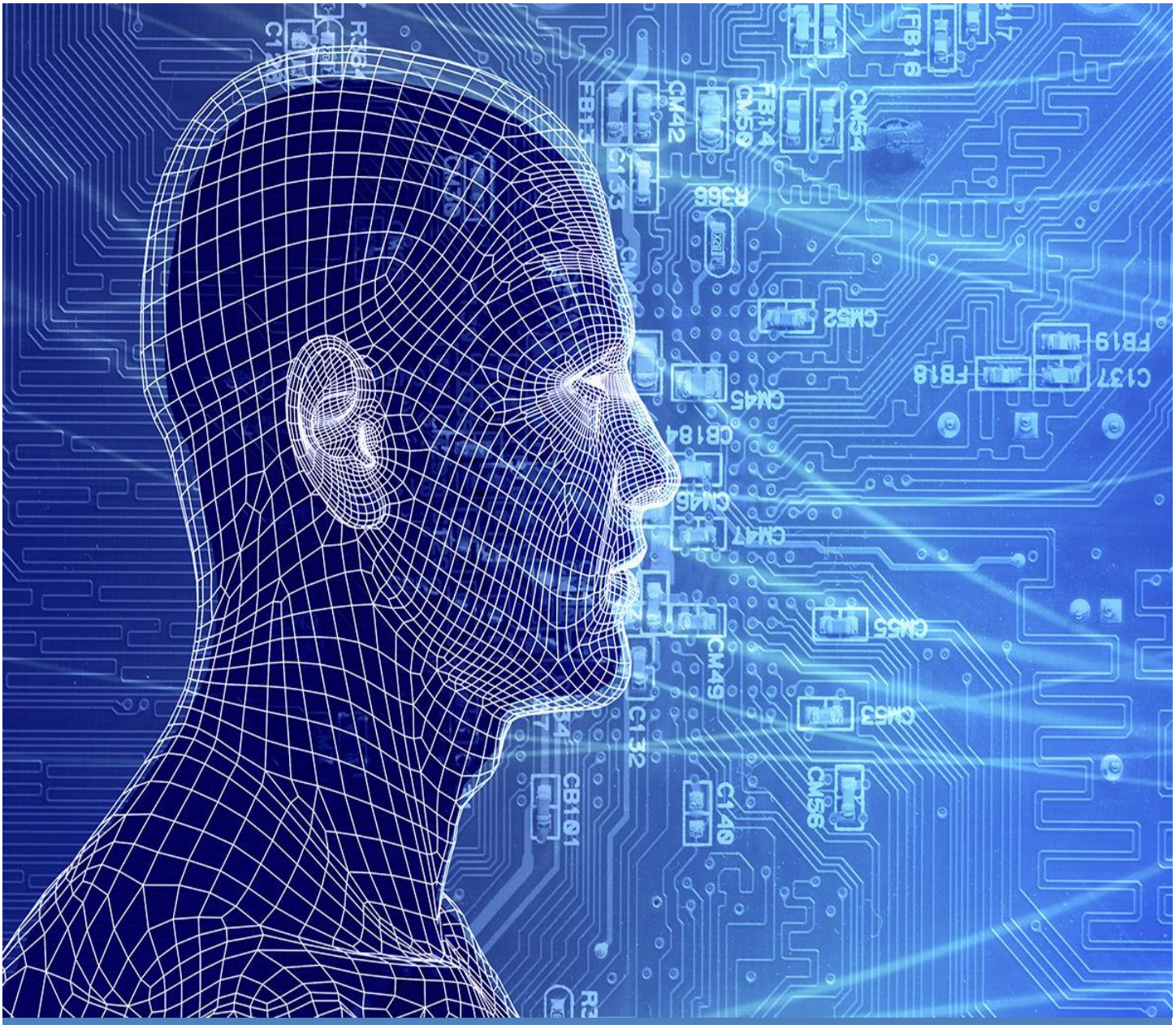

Völundr Pipeline

Version 1.0.0

Dennis Simpson

5/3/19

Table of Contents

Introduction

The Völundr bioinformatics pipeline is intended to analyze sgRNA distributions in cell populations for a modified synthetic lethal type assay. This pipeline has been used on libraries containing ~3900 sgRNAs up to libraries containing ~11,500 sgRNAs. To use this pipeline the sgRNA library must contain at least 500 control sgRNAs. These can either be non-targeting or designed to target intragenic regions. To install and use this read through the requirements listed below and get a working copy of Python installed first. Make sure you are doing all this on a Linux box that is 64-bit architecture. There are currently two methods to install this. The first is to clone or download the package to a location that you have access to. The second is to clone the package and then install using Python setuptools and the setup.py script. Unless otherwise noted ALL sequence position numbering uses the 0 based Python convention.

Requirements

- Python v3.4 or greater.
  - Python v3.5 is recommended.
- Designed to run on 64-bit Linux. Not tested on Mac. Don't even bother trying on Windows. Tested on these flavors:
  - Scientific Linux v7.4
  - RHEL v7.4
  - CentOS
- System Requirements
  - ≥20 Gb Ram
    - 16 Gb is the minimum
  - ≥4 CPUs or threads
    - The target search function runs in parallel, the more cores or threads available the faster it will run.
- Python modules needed to be installed. Unless noted use the current version.
  - Pathos. Provides better pickling and multiprocessor functions.
  - Python- Levenshtein. This is a 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.
  - wxPython (Needed for GUI API)
  - natsort
  - numpy
  - scipy
  - setuptools

Bugs, Quirks, and Improvements

General

- Would like a pip package
- Make Volundr_App.py GUI a standalone executable.
- Völundr_API option in the GUI does not allow for Völundr to have been installed with setuptools or pip.

Setup

- Online documentation lacking

Installation

If you have not already done so, first install the prerequisites. To install the program clone or download Völundr somewhere that you have full access. At this point the program is ready to use. You can also install as a Python module using setuptools and setup.py (instructions needed). The recommended method to run Völundr is to use a bash shell described below. Alternatively, if setuptools was not used to install it the program can be run from the command line with:

```
python3 /path/to/Volundr.py --options_file /path/to/run_Volundr.txt
```

If Völundr was installed as a Python module with setuptools then it can be run from the command line with:

```
python3 Volundr --options_file /path/to/run_Volundr.txt
```

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 5 columns. Additional columns are permitted but are ignored. Any blank line or line beginning with a "#" is treated as a comment and ignored. An example index file and an Excel template index file can be found in the docs folder of the Völundr package. The structure of this file must be column 1 containing an index name as defined in the Master Index File. Column 2 is for the sample or group. Column 3 defines the replicates. This is for the user and can be letters or integers. Column 4 is the species. Column 5 contains any user comments or notes.

```
# Index     Sample    Replicate  Species        User Comments
BC1         WT        A          Mouse          Gel Extracted
BC2         WT        B          Mouse          Gel Extracted
BC3         Plasmid   A          Mouse
```

This file can be edited during the analysis steps if any of the samples are to be excluded. The program will throw an error if any target count file defined by this index cannot be located. Version 1 of Völundr requires at least one sample named "Plasmid". This defines the maximum diversity of the targets.

Master Index File

A tab delimited text file containing at least 2 columns. Additional columns are permitted but are ignored. Any blank line or line beginning with a "#" is treated as a comment and ignored. Column 1 contains the index name and column 2 contains the index sequence. The sequence must be all caps.

```
# Index_ID   Sequence
BC1          CTAAGGTAAC
```

Target File

A tab delimited text file containing at least 3 columns. Additional columns are permitted but are ignored. Any blank line or line beginning with a "#" is treated as a comment and ignored. Column 1 is a unique identifier for each sgRNA. The order id for the sgRNA is a good value to place here. Column 2 is a unique name for each sgRNA. This must be of the format Gene_? As seen in the example. It is recommended that the first letter of the gene name be capitalized. Column 3 contains sequence information. This can be the actual

sequence of the cloning oligonucleotides ordered or just the sgRNA sequence. The sgRNA sequences can be of different lengths. This defines the two different types of target files. In example 1 the sgRNA begins at nucleotide 20 and is a fixed 20 nucleotides long for each sgRNA.

Example 1.

```
# mouse_DDR_ID    Gene       Sequence
mmDDR_1           Aicda_1    GTGGAAAGGACGAAACACCGACCATTTCAAAAATGTCCGCGTTTTAGAGCTAG
                             AAATAGCAAGTTAAAATAAGG
mmDDR_1           Aicda_2    GTGGAAAGGACGAAACACCGGTAGGTCTCATGCCGTCCCTGTTTTAGAGCTAG
                             AAATAGCAAGTTAAAATAAGG
```

In example 2 the sgRNA begins at position 0 and the sequences are of variable length. This example also shows a fourth column that the program ignores.

Example 2.

```
# ID                      Gene           sgRNA_Seq              code
ENSMUSG00000028232_0      Tmem68_221.1   GCTTGAGGAGTGGTTGGGTG   MPA
ENSMUSG00000044534_10     Ackr2_261.1    GGAACCGACGGTGGTGAG
```

run_Volundr.txt

This text file is used to pass all the option variables to the program. The recommended method to run Völundr is to use a bash shell file. This shell file doubles as the required options file. The simplest method to generate this bash file is to use the Volundr_App.py GUI located in the bin directory of the Völundr package. To execute this type at the command prompt:

```
python3 /path/to/Volundr_App.py
```

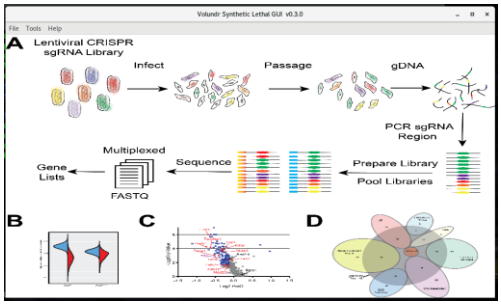

Figure 1. Splash Screen for Völundr GUI

This should bring up a window similar to that shown in Figure 1. This GUI will output either a Target Search script file or a Analyze Counts file. When filling in the forms found on the Target Search or Analyze Counts screens, everything shown on the screen must have a value assigned or the GUI will not allow you to save or run the script. Also, if you attempt to enter a value that is not allowed the data entry area will turn red. Some of the options have default values shown. These can be over-written on the screen. The default values can be edited. They are found in the *def default_dict_build()* near line 568 of Volundr_App.py. All options are case sensitive.

Target Search. Click on the Tools dropdown and then select TargetSearch. This will open the Target Search Parameters screen. Hovering over an option area will bring a balloon containing a tip for what belongs there. Several of the options have navigation buttons on the left that will bring up either a file or directory explorer window. Most options also have a dropdown button the right. This will show previously saved values for that option box.

Völundr_API

This is the full path to the Volundr.py API. This has a known issue in that it does not allow for Völundr to have been installed with setuptools. If setuptools was used, then the shell file needs to be edited after being saved.

--Options_File

This is the full path to where you want the shell file to live. I recommend placing and executing the script from within your working directory. Do not attempt to name the script file here and do not include the backslash if manually typing the path.

--Working_Folder

Full path to the working folder. Do not include the backslash if entering this manually.

--FASTQ1

Full path to FASTQ file including the name. The file can and is recommended to be gzipped. Uncompressed text files are also allowed. Other file formats will result in an error when you attempt to run the program.

--Target_File

Full path to this file including its name. The is a tab delimited text file containing the sgRNA sequences to be searched for.

--Master_Index_File

Full path to this file including its name. This is a tab delimited text file containing the name and sequence of the indices used for sequencing.

--Index_File

Full path to this file including its name. This is a tab delimited text file describing each library in the sequencing run.

Verbose

Sets the reporting level for how the program is running. Restricted options: INFO, DEBUG, ERROR, WARN

Job_Name

A short name for the job. This name is prepended to all the output files to make it easer to group them. Cannot contain spaces or special characters.

Spawn

Integer. How many parallel jobs to run during the search. Keep the maximum n-1 the number of CPUs or threads. The maximum number of parallel jobs will never exceed the number of libraries.

Species

Currently only Mouse or Human allowed.

Analyze_Unknowns

Restricted to True or False. There are always some reads in a sequencing run that do not have an identifiable index. If set to True these reads will be collected and searched like the ones with an identifiable index.

_Demultiplexed_FASTQ

Restricted to True or False. What this label really says is “Delete_Demultiplexed_FASTQ”. During the target search, temporary FASTQ files will be written that contain all the reads for a single index. If this is set to True these temporary files will be deleted when the program is finished. Default setting is True.

Compress

Restricted to True or False. If _Demultiplexed_FASTQ is False, then setting this to True will compress each temporary FASTQ file using gzip with a compression level of 9. If _Demultiplexed_FASTQ is True, then this option is ignored.

RevComp

Restricted to True or False. Set to True if the sequence in the target file is the reverse compliment of the sequence in the FASTQ file.

Target_Mismatch

Integer. How many mismatches to allow during target search? Generally recommend 1.

Min_Length

Integer. Minimum length of the sequence read

Target_Length

Either an integer value or Variable. This depends on the format used in the target file.

Target_Start

Integer value. Only used when target length is also an integer value. Defines the position of the sgRNA sequence in the target file string. See the target file section for more information.

Index_Mismatch

Integer value. How many mismatches are allowed when searching for indices. Recommended value is ≤1.

Target Padding

Integer value. If the program fails to find the start position of the sgRNA in the FASTQ read it will default to extracting a sequence. This allows for additional nucleotides on each end. Recommended value is 2.

Expected_Position

Integer value. Expected location of the first nucleotide of the sgRNA in the FASTQ read. Used only when the sgRNA anchor sequence is not found. If using Ion sequencing don't forget to account for the length of the Ion index which is removed prior to the search. For example, if the position is 121 in the raw FASTQ read it will be position 112 in the processed read that is being searched.

AnchorSeq

This is dependent on the viral backbone used to make the library. It is the 6 – 10 nucleotides immediately 5' of the first position of the sgRNA sequence.

AnchorMismatch

Integer value. How many mismatches to allow in the anchor. Generally, should be 0 or 1. Default is 1.

AnchorStart

Integer value. Position in the read to begin searching for the anchor sequence. This is based on the 5' end of the anchor sequence. Use same consideration as for the expected position. Don't start search at the exact expected position. You will recover more sgRNA sequences if you start 1 or 2 nucleotides back.

AnchorStop

Integer value. Define the position at which to stop searching for the anchor. This is based on the 5' end of the anchor sequence. Generally, stop searching 1 – 2 nucleotides past the expected start site of the sgRNA.

Once the form is filled in, click on "File" and then select "Run" if the computer you are working at has Völundr installed and all the paths are relative to that computer. Otherwise select "Save" to open a Save As dialog. You can then select the location to save the file. If you change the name of the file at this point you will need to edit the script file to reflect the name change before attempting to run it.

Analyze Counts. Click on the Tools dropdown and then select AnalyzeCounts. This will open the Statistics Parameters screen. Hovering over an option area will bring a balloon containing a tip for what belongs there. Several of the options have navigation buttons on the left that will bring up either a file or directory explorer window. Most options also have a dropdown button the right. This will show previously saved values for that option box.

Völundr_API

This is the full path to the Volundr.py API. This has a known issue in that it does not allow for Völundr to have been installed with setuptools. If setuptools was used, then the shell file needs to be edited after being saved.

--Options_File

This is the full path to where you want the shell file to live. I recommend placing and executing the script from within your working directory. Do not attempt to name the script file here and do not include the backslash if manually typing the path.

--Working_Folder

Full path to the working folder. Do not include the backslash if entering this manually.

--Target_File

Full path to this file including its name. The is a tab delimited text file containing the sgRNA sequences to be searched for.

--Master_Index_File

Full path to this file including its name. This is a tab delimited text file containing the name and sequence of the indices used for sequencing.

--Index_File

Full path to this file including its name. This is a tab delimited text file describing each library in the sequencing run.

Verbose

Sets the reporting level for how the program is running. Restricted options: INFO, DEBUG, ERROR, WARN

Job_Name

This must be the name used during the target search for this group. From the job name the program will be able to find the files containing the count data.

Spawn

Integer. How many parallel jobs to run during the analysis. Keep the maximum n-1 the number of CPUs or threads. The maximum number of parallel jobs will never exceed the number of libraries. The analysis runs so fast that it takes longer to initialize the parallel jobs than run them. Therefore the recommended value here is 2.

Species

Currently only Mouse or Human allowed.

Control_Sample

Sample name from the index file to use as a control. See index file section for more information.

Target_Mismatch

Integer. The target count files contain columns of counts for each mismatch. This can be set to any value ≤ the value used during the target search.

Target_Length

This should be the same value used during the target search.

Target_Start

This should be the same value used during the target search.

Target Search

Once all the files are in place it is time to do a target search run. Execute the shell script. If you don't want to run it from the bash shell, the options file needs to be edited to remove the python3 line. The program will write information to stdout and to a log file. The amount of information is dependent on the Verbose setting. All files will begin with the job name. Besides a log file the program will write temporary FASTQ files

that will be deleted unless the user has set Delete_Demultiplexed_FASTQ to True; a target count file that will have the name format of <Job Name>_<Index Name>_target_counts.txt; a position frequency file that will have the name format of <Job Name>_<Index Name>_Target_Position_Frequencies.txt; and finally a summary file that will have the name format <Job Name>_summary.txt. All these files are tab delimited text files.

Summary File

The summary file contains information about the number and length of reads in the FASTQ file; number and length of indexed reads found; number and length of indexed reads with targets found. The contents of a summary file are shown in Figure 2.

Figure 2. Summary file.

```
Running:        Volundr Synthetic_Lethal v0.10.0
Start_Time:     Sun Apr 28 20:44:00 2019
Stop_Time       Mon Apr 29 13:10:21 2019
FASTQ_File:     /pine/scr/d/e/dennis/25April2019b/25April2019b.fastq.gz
INDEX_File:     /pine/scr/d/e/dennis/25April2019b/25April2019b_Indices.bed
Target_File:    /nas/longleaf/home/dennis/Reference_Files/Targets/CRISPR/Mouse_Membrane_sgRNA.bed
Index_Mismatches        1
Target_Mismatches       1
Target_Padding 2
Expected_Position       153
Min_Read_length         185
Target_Start    20
Target_Length   Variable
Total_Reads:    24808149
Indexed_Reads: 20636975
Unknown_Count: 4171174
Unknown_Short_Count:    3273205
Unknown_Full_Length_Count:      897969

Index Name      Sample Name     Sample Replica Total    Short Reads     Full Reads      0_mismatches
        1_mismatches    Targeted        Not Targeted    Fraction Targeted
BC1     WT_Shield       A       1066505 978648  87857   976623  2025    754323  224325
        0.7707807097138093
```

Target Positions Frequency File

This tab delimited text file contains three tables listing the number and fraction of total reads containing anchor sequences, sgRNA sequences with anchor sequences, and sgRNA sequences without anchor sequences by position in the read. This information provides quality assurance that the anchor sequence and sgRNA sequences are location at or near the expected position.

Target Counts File

This tab delimited text file contains the raw counts for each sgRNA target. There are a minimum of three columns plus an additional column for each mismatch allowed in the target search. The example shown below is from a target search done allowing 1 mismatch

```
# Target        Target_Key              0_mismatches    1_mismatches
Tmem68_221.1    GCTTGAGGAGTGGTTGGGTG    90              0
Tmem68_222.2    GGGTGAAAACCCACAAAAGA    0               0
```

Analyze Counts

Once all the files are in place it is time to do a target search run. Execute the shell script. If you don't want to run it from the bash shell, the options file needs to be edited to remove the python3 line. The program will write information to stdout and to a log file. The amount of information is dependent on the Verbose setting. All files will begin with the job name. Any libraries that you don't wish to analyze must be commented out in the index file or you will get an error. Besides a log file this will write several different files.

<Job Name>_<Sample Name>_TD_norm.txt
<Job Name>_Log2_Control_Targets.txt
<Job Name>_Log2_Delta_<Control Sample Name>_Genes.txt
<Job Name>_Log2_Genes.txt
<Job Name>_<Sample Name>_KS_Log2_Delta_Genes.txt
<Job Name>_Permuted_Log2_GMeans.txt

TD Norm File

This file is a tab delimited text file containing 5 columns. Column 1 is the individual sgRNAs. Column 2 is the target sequence associated with the sgRNA. Column 3 is the TC_Norm value. This is the counts for a given sgRNA normalized to the total sgRNA counts for the library. Column 4 is the TD_Norm value for each sgRNA. This is the TC_Norm value for the individual sgRNA from the samples divided by the TC_Norm value for the same sgRNA from the Plasmid sample. Column 5 is a log2 transformation of the TD_Norm value.

Log2 Control Targets File

This file contains one column for each non-Plasmid sample plus a column of the individual control sgRNAs. The data in this file is derived from the TD Norm file by simply extracting the sgRNAs labeled Mouse_xx.

Log2_Delta_<Control Sample Name>_Genes File

This file contains one column for each non-Plasmid sample plus a column of the individual sgRNA names. The file always includes the control sample as well. This is derived from the TD Norm file by subtracting the log2 value of the control sample from the log2 value of the unknown sample.

Log2 Genes File

This file contains a column of gene names plus one column for each non-plasmid sample analyzed. These values are derived by taking the geometric means of the TD Norm values for the guides for each gene and log2 transforming it. The result is a difference with respect to the Plasmid sample.

KS Log2 Delta Genes File

This is probably the key file for the analysis. There is one file per sample group written containing 3 columns. Column 1 is the gene name. Column 2 is the log2 delta value also found in the Log2_Delta_<Control Sample Name>_Genes file. Column 3 is the p value from a Kolmogorov-Smirnoff test comparing the value of the individual sgRNAs for each gene group to the values of the group of control sgRNAs.

Permuted Log2 GMeans File

This file is the result of a permutation analysis where the value of 10 control sgRNAs are randomly selected and analyzed as if they were a gene 10,000 times.