FASTAptameR 2.0 - User Interface Tutorial

Skyler T. Kramer Paige R. Gruenke Khalid K. Alam
Rebecca N. Burke-Agüero Dong Xu Donald H. Burke

3/22/2021

Contents

1 Introduction

FASTAptameR 2.0 is an R-based update of FASTAptamer (Alam KK 2015). Like its predecessor, FASTAptameR 2.0 is an open-source toolkit designed to analyze populations of sequences resulting from combinatorial selections. This updated version features a user interface (UI), interactive graphics, more modules, and a faster implementation of the original clustering algorithm.

This user guide walks you through installation and each of the analytical modules, and it highlights what options you have when analyzing your data through the UI.

1.1 Overview

- **FASTAptameR-Count**
  - *This module is the entry point into FASTAptameR 2.0*
  - Input: preprocessed FASTQ/A
  - Workflow:
    1. count all unique sequences (`Reads`)
    2. sort by counts (`Rank`)
    3. normalize counts as reads per million (`RPM`)
  - Plotting:
    1. line plot of reads-per-rank
    2. histograms of sequence lengths - one for the unique sequences and one for all reads
    3. sequence abundance bar plot
  - Output: FASTA or CSV
- **FASTAptameR-Translate**
  - Input: counted FASTA
  - Workflow: translate D/RNA sequences to amino acid sequences
  - Plotting:
    1. line plot of reads-per-rank
    2. histograms of sequence lengths - one for the unique sequences and one for all reads
  - Output: FASTA or CSV
- **FASTAptameR-Motif_Search**
  - Input: counted FASTA and comma-separated patterns
  - Workflow: search for user-defined patterns in sequences
  - Output: FASTA or CSV
- **FASTAptameR-Motif_Tracker**
  - Input: at least two counted FASTAs and query list
  - Workflow: track how user-defined motifs or sequences change across populations
  - Plotting: line plot of each query's RPM across the populations
  - Output: CSV
- **FASTAptameR-Distance**
  - Input: counted FASTA and query sequence
  - Workflow: compute the Levenshtein edit distance (LED) between a single query sequence and all other provided sequences
  - Plotting: histograms of edit distances - one for the unique sequences and one for all reads
  - Output: FASTA or CSV
- **FASTAptameR-Enrich**

  - Input: at least two counted or clustered FASTAs
  - Workflow: calculate how each sequence enriches across populations
  - Plotting:
    1. bar plot of sequence persistence
    2. histogram(s) of `log2(Enrichment)`
    3. scatter plot(s) of RPM
    4. volcano plot of `log2(Enrichment)` and a term related to frequency
    5. box plot(s) of sequence enrichment per cluster; only available if clustered FASTAs are provided
  - Output: CSV

- **FASTAptameR-Positional_Enrichment**
  - Input: enrichment CSV and reference sequence
  - Workflow: for each position of the reference sequence, compute the average enrichment of non-reference residues in the data
  - Plotting:
    1. bar plot of average enrichment per position of reference sequence
    2. heat map of average enrichment per position of reference sequence, further grouped by residues
  - Output: None

- **FASTAptameR-Cluster**
  - Input: counted FASTA
  - Workflow:
    1. filter out low-read sequences based on user-defined input
    2. treat most abundant, non-clustered sequence as cluster seed
    3. add all sequences within a user-defined LED of the seed to the cluster
    4. Repeat until all sequences are clustered or a maximum number of clusters are created
  - Output: FASTA or CSV

- **FASTAptameR-Cluster_Diversity**
  - Input: clustered FASTA
  - Workflow: provide metadata for each cluster
  - Plotting:
    1. metaplots for count of unique sequences, count of total reads, and average LED per cluster
    2. k-mer PCA plot, colored by cluster identity
  - Output: CSV

- **FASTAptameR-Cluster_Enrich**
  - Input: at least two cluster-analysis CSVs
  - Workflow: calculate how each cluster enriches across populations
  - Plotting: line plot of each the total RPM per cluster for each seed per population
  - Output: CSV

A summary of input and output file types is given by **Table 1**. Please note that each module requires the user to upload a file or, in the case of **FASTAptameR-Count**, optionally provide a GitHub link to the data. At present, none of this data will "live" on the server to be passed between modules.

Table 1: Module Inputs and Outputs

Module	Input Files	Output Files
FASTAptameR-Count	Preprocessed FASTQ/A	FASTA or CSV
FASTAptameR-Translate	Counted FASTA	FASTA or CSV

Module	Input Files	Output Files
FASTAptameR-Motif_Search	Counted FASTA	FASTA or CSV
FASTAptameR-Motif_Tracker	2 or 3 counted FASTAs	CSV
FASTAptameR-Distance	Counted FASTA	CSV
FASTAptameR-Enrich	2 or 3 counted FASTAs	CSV
FASTAptameR-Cluster	Counted FASTA	FASTA or CSV
FASTAptameR-Cluster_Diversity	Clustered FASTA	CSV
FASTAptameR-Cluster_Enrich	2 or 3 cluster-analysis CSVs	CSV

Many function inputs / outputs are simply FASTA files, so FASTAptameR 2.0 can be easily integrated into most analytical pipelines. Note that *counted* FASTA files are the minimum input for most modules (*e.g.*, FASTAptameR-Translate needs *at least* a counted FASTA but could also accept a searched or clustered FASTA file).

Importantly, FASTAptameR 2.0 does not provide any functions that are easily addressed by other software (*e.g.*, merging paired-end reads, trimming constant regions, predicting structures, *etc.*). Rather, the focus of this application is to provide flexible downstream analyses for the selections field.

2 How to get started

Exactly like its predecessor, FASTAptameR 2.0 is designed to be **easy** to use. There are three main ways for users to interact with FASTAptameR 2.0. The *web server* is the easiest way to interact with this application because it only requires an internet connection and browser. However, you can run it locally as a *Docker container*.

2.1 User interface

2.1.1 Web Server

This is the easiest way to use the FASTAptameR 2.0 UI. The web server can be accessed from https://fastaptamer2.missouri.edu/, which is hosted by the Digital Biology Laboratory under the direction of Dr. Dong Xu. However, this option only works if the files you need to upload are less than 2 GB.

2.1.2 Docker

The web server can also be run locally on your machine(s) via Docker. In brief, Docker is a convenient tool that may be used to construct *images* of software. The *image* essentially functions as the blueprint for an application. The *image* of FASTAptameR 2.0, for example, contains all relevant software (*e.g.*, `R`), files (*e.g.*, this PDF), and packages (*e.g.*, `Shiny`). For details on Docker or its installation, please see https://www.docker.com/ and https://docs.docker.com/get-docker/, respectively.

The FASTAptameR 2.0 *image* must be pulled from a repository (*i.e.*, Docker Hub) by running

```
docker pull skylerkramer/fastaptamer2:publicupload03
```

in a Docker-active terminal. Windows and Mac users will have a Docker-active terminal after proper installation of Docker Desktop. Linux users will have a Docker-active terminal after proper installation of Docker. Windows, Mac, or Linux users can check if their terminal is Docker-active by running

```
docker version
```

which should indicate which version of Docker has been installed.

Importantly, the FASTAptameR 2.0 *image* is built on Linux. Thus, it is necessary to run it from a Linux environment or virtual machine. For Mac or Windows users, the installation instructions for Docker Desktop (https://docs.docker.com/get-docker/) will show you how to do this.

Once you have this application's *image*, running

```
docker run -d --rm -p 3838:3838 skylerkramer/fastaptamer2:publicupload03
```

from a Docker-active terminal will launch a local instance - a *container* - of FASTAptameR 2.0. You will then interact with this *container* in the same fashion as the web server by navigating to `localhost:3838` in your web browser.

Explanation of flags from the previous command:

- `-d`: enable detached mode, which allows you to use your command line / terminal even with the active *container* (*i.e.*, *container* is detached from your terminal and runs in the background)
- `--rm`: automatically remove the container upon exit
- `-p 3838:3838`: publish `3838` host port (1st number) to the `3838` container port (2nd number)
- `skylerkramer/fastaptamer2:publicupload03`: the local path to the **FASTAptameR 2.0** Docker *image*
- `localhost:3838`: navigate here from your web browser to start interacting with **FASTAptameR 2.0**

To recap:

1. Install Docker on a Linux machine or install Docker Desktop on a Windows or Mac machine to establish a 'Docker-active terminal' on your local system.
2. Pull the FASTAptameR 2.0 image from the Docker Hub repository.
3. Execute the `docker run ...` command and navigate to `localhost:3838` in your web browser.

2.2 Software usage

If you use, adapt, or modify FASTAptameR 2.0, please cite: **CITATION**.

For any questions or concerns, please email burkelab@missouri.edu or stk7c9@umsystem.edu.

3 Tutorial

3.1 Data requirements

FASTAptameR 2.0 utilizes many string-based functions in its backend. Thus, this program can be used to analyze many types of biological populations. However, all libraries must be initially saved in a FASTA or FASTQ format and passed through **FASTAptameR-Count** prior to any subsequent analyses. Further, any data preprocessing steps must be made outside of this application.

3.2 Sample Data and Uploading User Data

All data shown in this tutorial come from the 14th and 15th rounds of an aptamer selection against HIV-1 reverse transcriptase (Burke DH 1996; Ditzler MA 2013; Whatley AS 2013). These data are preprocessed (trimmed and filtered) and available from http://burkelab.missouri.edu/fastapamer.html.

To start analyzing the sample data or your own data, please do one of two things. Either **1)** upload a local copy of the file via the file browser in **FASTAptameR-Count** or **2)** supply a link to the data via the text box labeled as `Online source` in **FASTAptameR-Count**. This module is the entry point to **FASTAptameR 2.0**, so each analysis should start here.

3.3 FASTAptameR-Count

3.3.1 Description

FASTAptameR-Count serves as the entry point into this suite of modules, and, thus, it should be run prior to any of the following modules. This function accepts either a FASTQ/A file chosen with the file browser or a link to such a file (*e.g.*, the default GitHub link in the text box labeled as `Online source`) and returns a *counted* data table as output that can be downloaded as a FASTA or CSV file.

Input FASTQ files should be properly formatted (4 lines per entry with the 2nd line of each entry being the sequence). Input FASTA files are not required to have sequence identifiers. No pre-existing sequence identifiers will be conserved by this module. Instead, output sequence identifiers are defined by the statistical representation of each sequence. Sample input files are shown in **Fig. 1**. A screenshot of the module interface is shown in **Fig. 2**.

```
AGCGCGGCACCCAAAATCGAAATCCGAAGGCGAACGGGAGAATGCGACCAAAGATACCCTGTGAATGGC
AGCGCGGCACCCAAAATCGAAATCCGAAGGCGAACGGGAGAATGCGACCAAAGATACCCTGTGAATGGC
AGCGCGGCACCCAAAATCGAAATCCGAAGGCGAACGGGAGAATGCGACCAAAGATACCCTGTGAATGGC
ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCACGCTTGGTGT
ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCACGCTTGGTGT
ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCACGCTTGGTGT
ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCACGCTTGGTGT
ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCACGCTTGGTGT
```

(a) Sample FASTA input without sequence identifiers

```
>Description Line 1
AGCGCGGCACCCAAAATCGAAATCCGAAGGCGAACGGGAGAATGCGACCAAAGATACCCTGTGAATGGC
>Description Line 2
AGCGCGGCACCCAAAATCGAAATCCGAAGGCGAACGGGAGAATGCGACCAAAGATACCCTGTGAATGGC
>Description Line 3
AGCGCGGCACCCAAAATCGAAATCCGAAGGCGAACGGGAGAATGCGACCAAAGATACCCTGTGAATGGC
>Description Line 4
ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCACGCTTGGTGT
```

(b) Sample FASTA input with sequence identifiers

```
@HWI-700167R:511:C3N2DACXX:5:1101:4016:2214 1:N:0:GACGGCT
AGCGCGGCACCCAAAATCGAAATCCGAAGGCGAACGGGAGAATGCGACCAAAGATACCCTGTGAATGGC
+
IIJJJIJJJJJJJJJJIIIJIJJJHHHFF<ADBDDDDDDDDDDCCDDDDDDDDDDCACCDCDDDDECDD
@HWI-700167R:511:C3N2DACXX:5:1101:4997:2191 1:N:0:GACGGCT
ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCACGCTTGGTGT
+
IIJIJJGIJJJJIIJJJJIJIJJJJJJGHHIIJJJIJJJHHFFFEEEEDDDDDDDDDDDDDDDBB@DCDD
```

(c) Sample FASTQ input

Figure 1: Valid input types

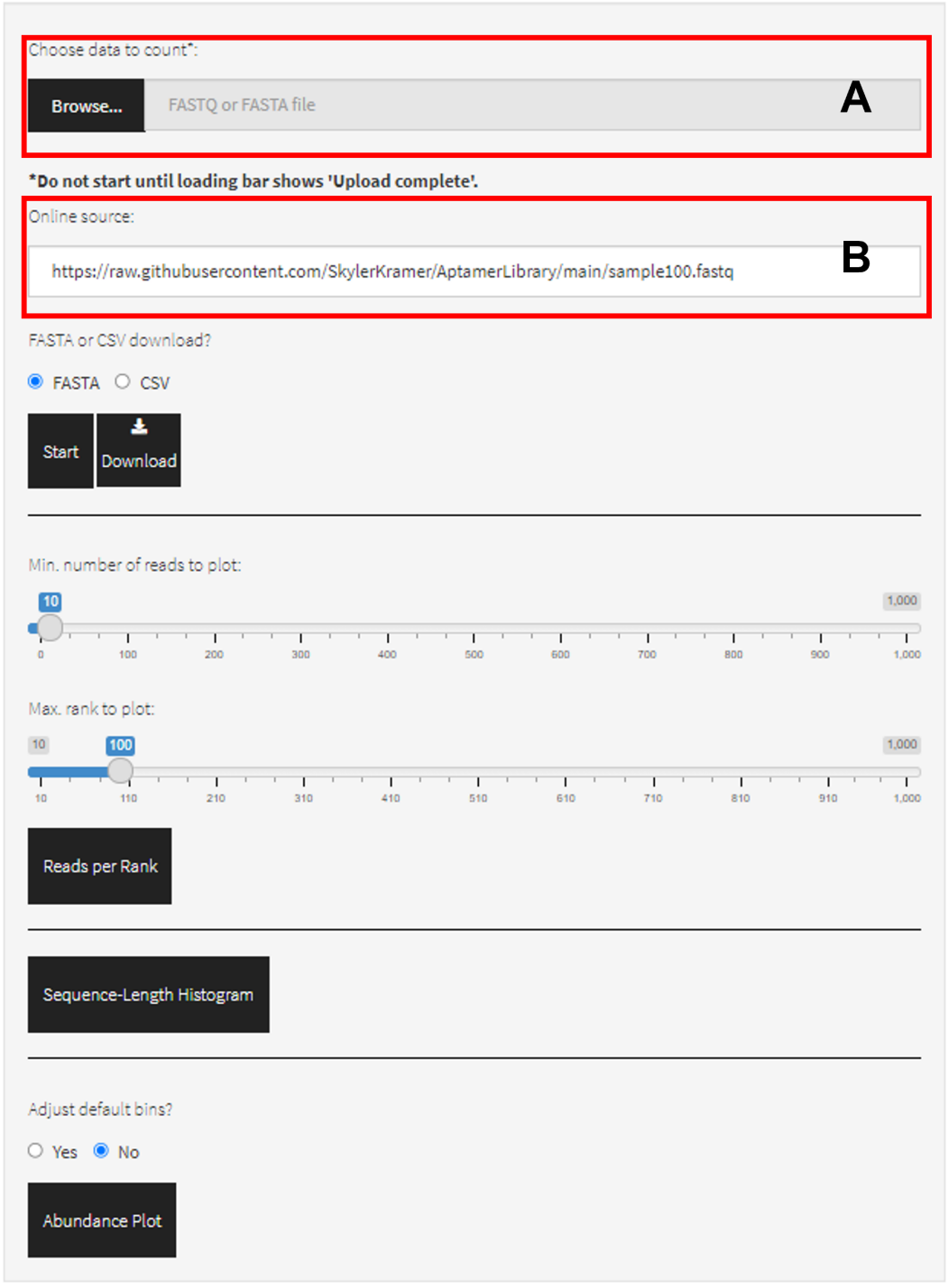


Figure 2: Screenshot of FASTAptameR-Count.

3.3.2 Usage

The input FASTQ/A file must be chosen with the file browser (**Fig. 2a**) or linked in the text box (**Fig. 2b**). A sample link is already provided in the text box. Note, if a file is uploaded via the file browser **AND** a link is provided, the uploaded file (**NOT** the linked one) will be analyzed.

The `Start` button will begin the counting process. The results will be displayed as a data table on the right side of the screen. For file uploads, please wait for the loading bar to show *Upload complete* before using the `Start` button.

If you start the module before the upload is finished **AND** a file link is provided, then the file link will be analyzed. If you start the module before the upload is finished **AND** no link is provided, then you will get an error that says `No file or link provided!`. If you start the module before the upload is finished **AND** no file link is provided **AND** you have previously uploaded a file to this module, then the previously uploaded file will be reanalyzed. In any of these cases, reuploading the file, *waiting for it to finish uploading*, and then starting the module should correctly analyze your data. If an error persists (*e.g.*, one that says `Error in nchar()`), please refresh the page.

A sample output data table is shown in **Fig. 3**.

Show 10 entries Search:

id	Rank	Reads	RPM	Length	seqs
All	All	All	All	All	All
>1-417696-193358.44	1	417696	193358.44	70	ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCACGCTTGGTGT
>2-313312-145037.35	2	313312	145037.35	70	CATAGCGACTGTCCACGAATCCGAAGCCTAACGGGACAAAAGGCAAGAGCGCGATACCAATGCTGGACTG
>3-174096-80591.94	3	174096	80591.94	70	AACCGCAAGCAACACCCAGCAAGAAACATCCGACGCACGACGGGAGAAAGTGCATTACCACGATGTCGAT
>4-94978-43966.9	4	94978	43966.9	70	CATAGCGACTGCCCACGAATCCGAAGCCTAACGGGACAAAAGGCAAGAGCGCGATACCAATGCTGGACTG
>5-74389-34435.91	5	74389	34435.91	70	ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCGCGCTTGGTGT
>6-57625-26675.57	6	57625	26675.57	69	CCCTCCTTGTATGACGCTAACTGAGAATCCGAAGTCCAACGGGAGAAAGGACACTTATGACGTGGCGCG
>7-53608-24816.04	7	53608	24816.04	70	ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCATGCTTGGTGT
>8-39793-18420.84	8	39793	18420.84	69	AGCGCGGCACCCAAAATCGAAATCCGAAGGCGAACGGGAGAATGCGACCAAAGATACCCTGTGAATGGC
>9-33800-15646.58	9	33800	15646.58	70	TTGACAATAACTCGAGAAGAACCGAGGTGCAAACGGGAGAACACAATGGATTACACCGAGCTCGGCTGAC
>10-29794-13792.14	10	29794	13792.14	70	GCGAACCAAACCCAGATTACTAACCGTGGGCCTGAAACACGGGACAAAACAGGCATCAATGGAGTGGTAC

Showing 1 to 10 of 72,921 entries Previous 1 2 3 4 5 … 7293 Next

Figure 3: FASTAptameR-Count Output.

Note that the *id* column has the following format: `>Rank-Reads-RPM`, where `Rank` is the order of sequences after sorting by `Reads`, which is the raw abundance of each sequence. `RPM` - Reads per Million - is the value of `Reads`, normalized by the total population size: `RPM = Reads / (populationSize / 1e6)`.

The total number of sequences, number of unique sequences, and module runtime will be displayed below the `Start` and `Download` buttons after running is finished. The `Download` button opens a file browser prior to downloading the output as a FASTA or CSV file (`DEFAULT = FASTA`, which is required for subsequent modules). Keep the *count* file in an easily accessible folder, as this file will serve as input for many other FASTAptameR 2.0 modules.

3.3.3 Plotting

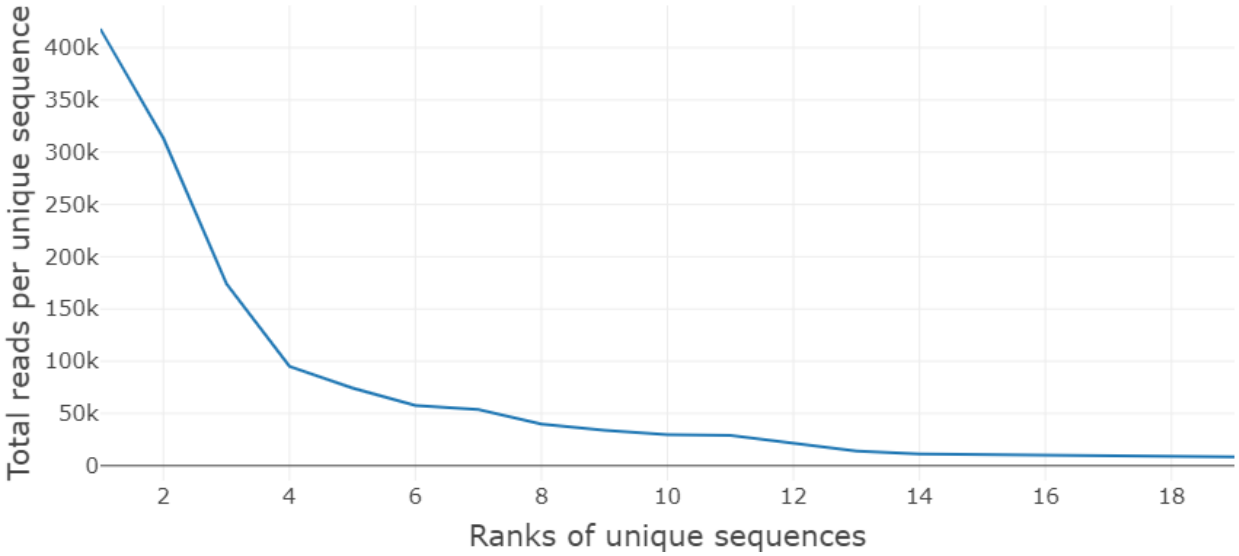
This module can also generate three types of interactive plots based on the counted data: a line plot of reads-per-rank (**Fig. 4**), two histograms of sequence lengths (**Fig. 5**), and a sequence abundance bar plot (**Fig. 6**). The line plot is filterable by 1) minimum number of reads to plot and 2) maximum rank to plot. Both values are chosen with a slider bar.

The 1st histogram corresponds to the sequence-length distribution of unique sequences, whereas the 2nd histogram corresponds to the sequence-length distribution of all reads. The histograms are not filterable.

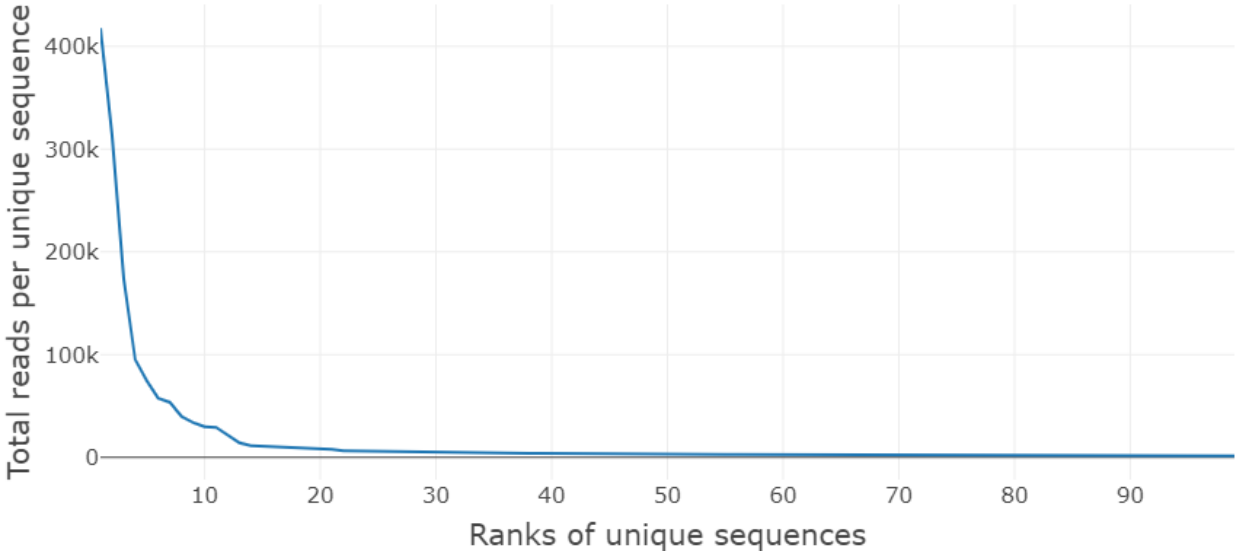
The sequence abundance bar plot first bins sequences based on their read counts and then plots these bins against their relative abundance (as fractions of the total population). Finally, the bars are colored according to the number of unique sequences in each bin. These breakpoints forming these bins are, by default, set to the following: `Reads = 1`, `1 < Reads < 10`, `10 <= Reads < 100`, `100 <= Reads < 1000`, `1000 <= max(Reads)`. However, the user can toggle singletons on/off or alter these breakpoints by selecting `Yes` to the respective prompt. New breakpoints should be entered as a comma-separated list.

3.3.4 A note on plotting

Though many plots are initially created with `ggplot2`, they are all shown as interactive `plotly` plots. As such, you will see a number of options appear along the top of the image in response to your mouse hover. These options will allow you to zoom in and out, select regions of interest, and, most importantly, download the plots. This last functionality is provided by the camera icon (1st icon on the left as you hover over the image). Finally, double-clicking the image should reset it (*e.g.* remove zoom or crop effects).



A) Top 20−ranked sequences (min. reads = 10)



B) Top 100−ranked sequences (min. reads = 10)

Figure 4: Reads-per-Rank line plots. Two views of the 70HRT14 data, where panels (A) and (B) include the top 20 and top 100 ranked sequences, respectively. These values are chosen with the corresponding slider bars (shown in Fig. 3).

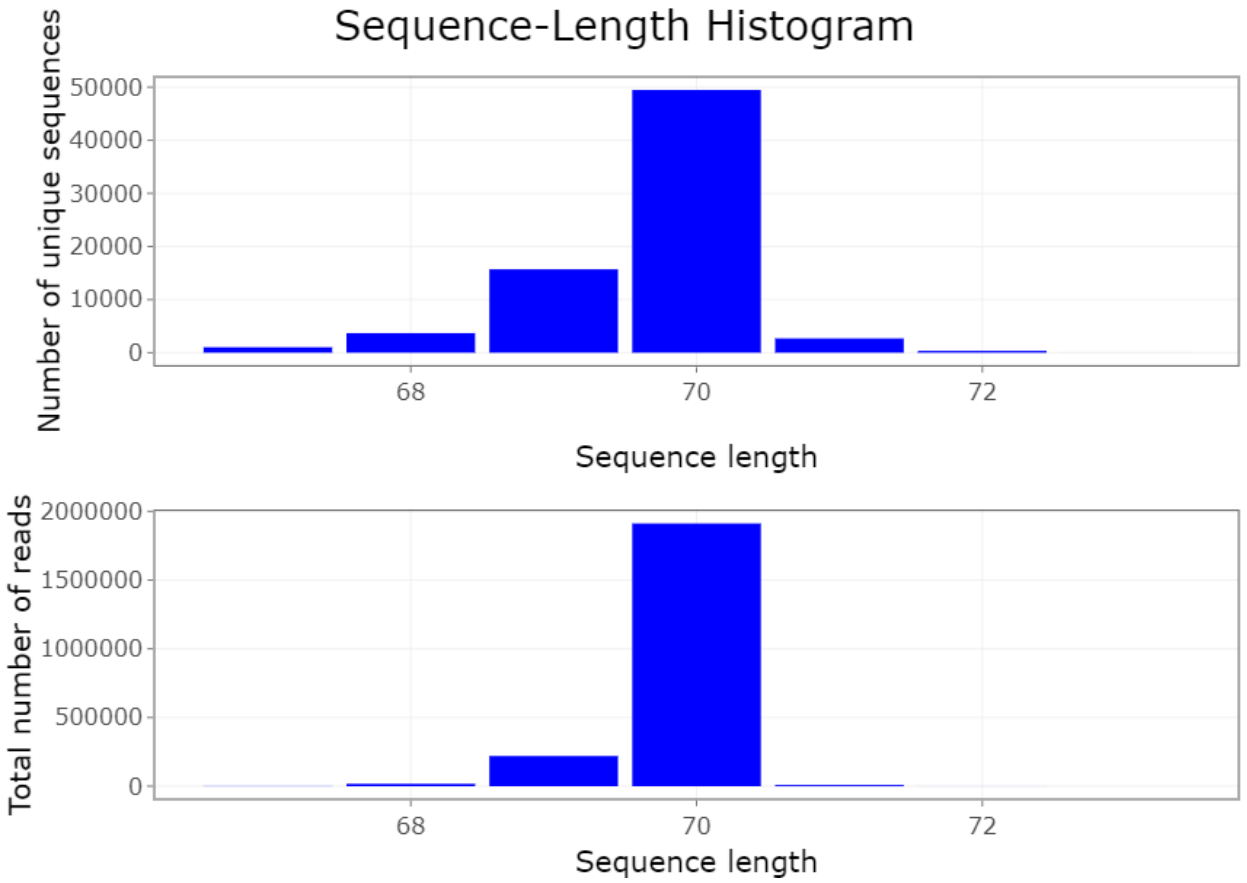


Figure 5: Sequence-Length Histogram. Two views of the 70HRT14 data, where the top and bottom panels correspond to unique sequences and total reads, respectively. Both plots are generated together in the UI.

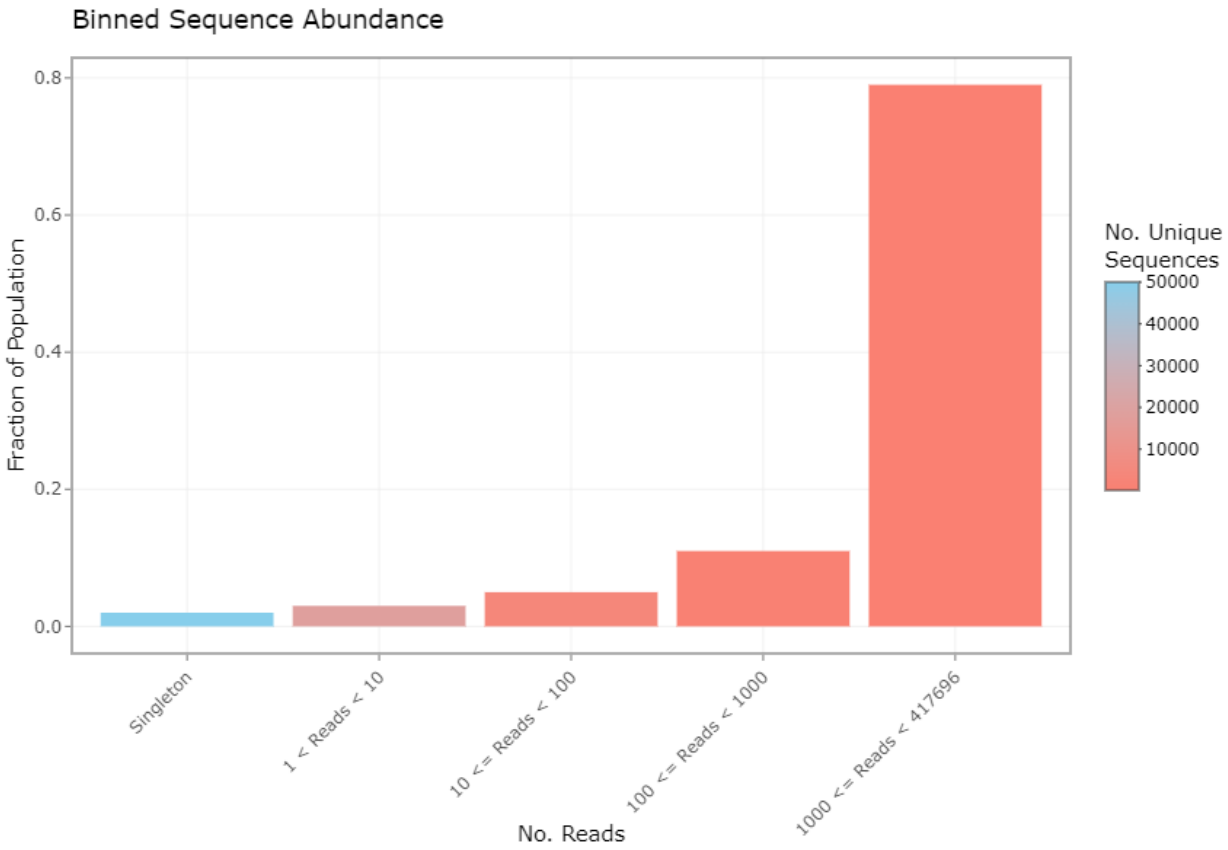


Figure 6: Sequence Abundance Bar Plot. Shows binned sequences as fractions of the total population, colored by the number of unique sequences per bin.

3.4 FASTAptameR-Translate

3.4.1 Description

FASTAptameR-Translate translates input nucleotide sequences into amino acid sequences following the standard genetic code, treating the input nucleotide sequences as positive-sense mRNA. This module accepts a *counted* FASTA file and returns a *translated* data table that can be downloaded as a FASTA or CSV file. A screenshot of the module interface is shown in **Fig. 7**.

3.4.2 Usage

The input FASTA file must be chosen with the file browser (**Fig. 7a**). The open reading frame may be selected by the 1st set of radio buttons (`DEFAULT = 1`) (**Fig. 7b**). The 2nd set of radio buttons indicates whether nucleotide sequences that encode the same amino acid sequence should be merged (`DEFAULT = Yes`) (**Fig. 7c**). If `Yes`, then redundantly encoded amino acid sequences are converged, and a new column (`Unique.Nt.Count`) will specify how many non-unique nucleotide sequences from the *counted* input were merged into each amino acid sequence. If `No`, then each unique nucleotide sequence is treated separately, even if multiple sequences encode the same amino acid sequence.

The user may also alter the standard genetic code by selecting `Yes` in the 3rd set of radio buttons (**Fig. 7d**) prior to translating. If `Yes`, then comma-separated codon / translation pairs may be entered in the resulting text box (*e.g.*, `GAT,Z`). If the codon already exists in the standard genetic code, then the user-supplied mapping will take precedence. If the codon does not exist in the standard genetic code, then it will be added to it. Please note that only 3-letter codons and 1-letter translations are currently accepted.

The `Start` button begins the translation process. The *translated* data table will be shown on the right side of the screen. The `Download` button opens a file browser prior to downloading the output as a FASTA or CSV file (`DEFAULT = FASTA`, which is required for subsequent modules).

3.4.3 Plotting

This module generates the same two types of plots as FASTAptameR-Count: a line plot of reads-per-rank and two histograms of sequence lengths. See that section for more details.

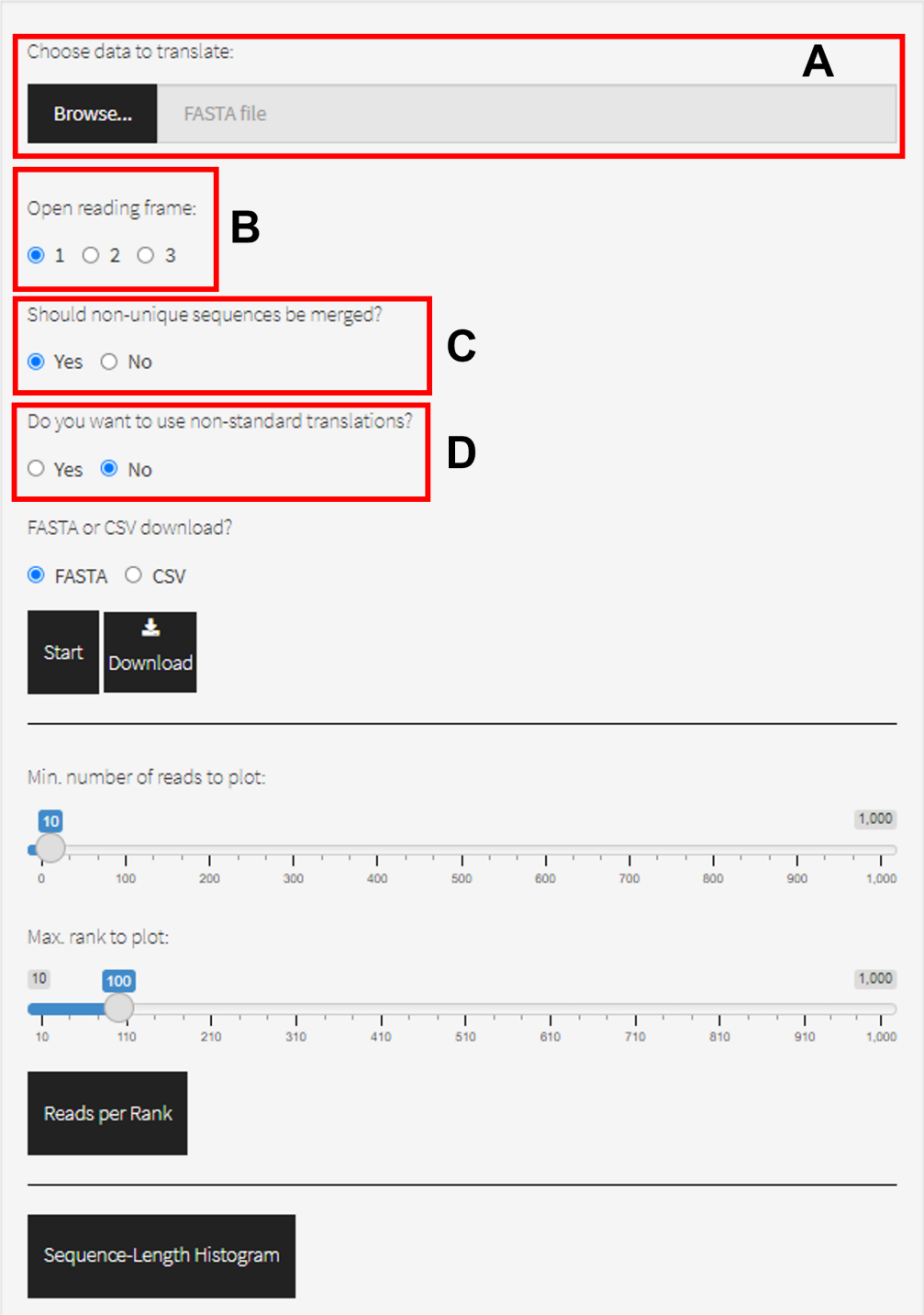


Figure 7: Screenshot of FASTAptameR-Translate.

3.5 FASTAptameR-Motif_Search

3.5.1 Description

FASTAptameR-Motif_Search identifies sequences that contain one or more user-specified sequence motifs, or 'patterns.' The module accepts a *counted* FASTA file and returns a *searched* data table that can be downloaded as a FASTA or CSV file. Sequences in the output must have at least one occurrence of each pattern or at least one occurrence of at least one pattern (see details below for the `partial match` radio button). A screenshot of the module interface is shown in **Fig. 8**.

3.5.2 Usage

The input FASTA file must be chosen with the file browser. The following text box (**Fig. 8a**) must contain at least one pattern (*e.g.*, `AAA`). If the user wishes to search for multiple patterns, the patterns must be separated by commas (*e.g.*, `AAA,GTG`).

The 1st set of radio buttons (**Fig. 8b**) determines whether the output has parentheses set around identified patterns (`DEFAULT = No`). For example, when `pattern = GGC` and `sequence = AAAGGCT`, the output is `AAA(GGC)T`. Note, when two or more patterns overlap, output only displays parentheses around the first search term that is matched. For example, when `pattern = AGGC,GGCT` and `sequence = AAAGGCT`, the output is `AA(AGGC)T`. Note that parentheses will be treated as individual characters by subsequent modules and may alter downstream analyses.

The 2nd set of radio buttons (**Fig. 8c**) governs how the software deals with multiple search terms. When the query contains multiple patterns, the search can be carried out either as a Boolean `AND` function by requiring all parts of the query to be present within a given sequence (this is the `default`, with button set to `No`), or as a Boolean `OR` function to identify sequences that contain any part of the query (set button to `Yes`). If `Yes`, filtered sequences must have at least one occurrence of **at least one** of the listed patterns. If `No` (`DEFAULT`), filtered sequences must have at least one occurrence of **each** of the listed patterns.

The 3rd set of radio buttons (**Fig. 8d**) determines the type of pattern (`DEFAULT = Nucleotide`). If `Nucleotide`, then degenerate nucleotide codes are allowed, and T/U are interchangeable. Degenerate search patterns are **not** allowed for other sequence types. Importantly, all patterns are converted to uppercase and have white spaces removed regardless of the pattern type.

1. **A**/**T**/**G**/**C**/**U** - single bases
2. **R** - puRine (A/G)
3. **Y** - pYrimidine (C/T)
4. **W** - Weak (A/T)
5. **S** - Strong (G/C)
6. **M** - aMino (A/C)
7. **K** - Keto (G/T)
8. **B** - not A
9. **D** - not C
10. **H** - not G
11. **V** - not T/U
12. **N** - aNy base (not *gap*)

The `Start` button begins the search process. The *searched* data table will be shown on the right side of the screen. The `Download` button opens a file browser prior to downloading the output as a FASTA or CSV file (`DEFAULT = FASTA`, which is required for subsequent modules).

A sample output data table is shown in **Fig. 9** with the following parameters: `comma-separated patterns = UCCG,CGGGAnAA`, `parentheses = No`, `partial filtering = No`, and `pattern type = Nucleotide`.

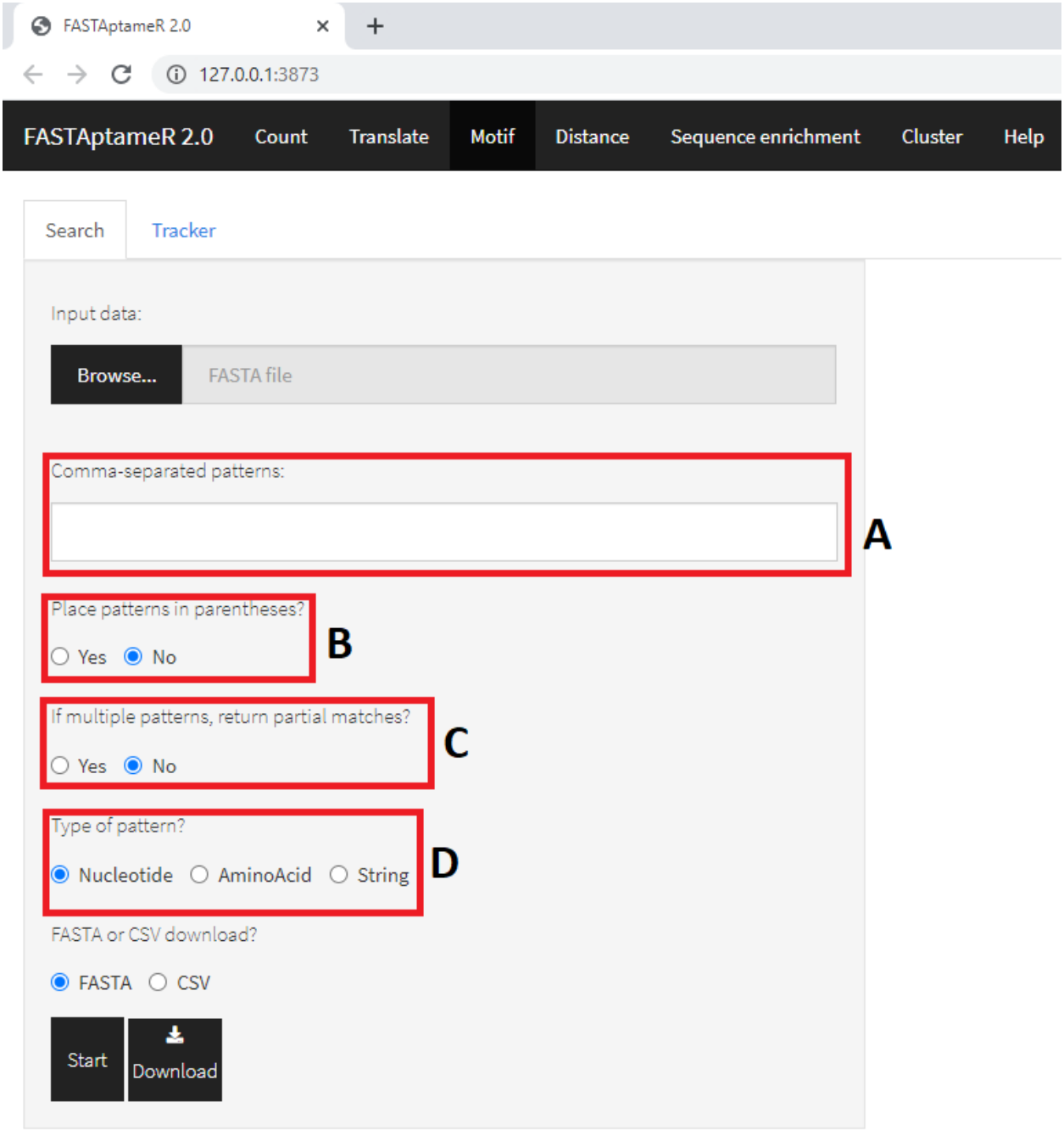
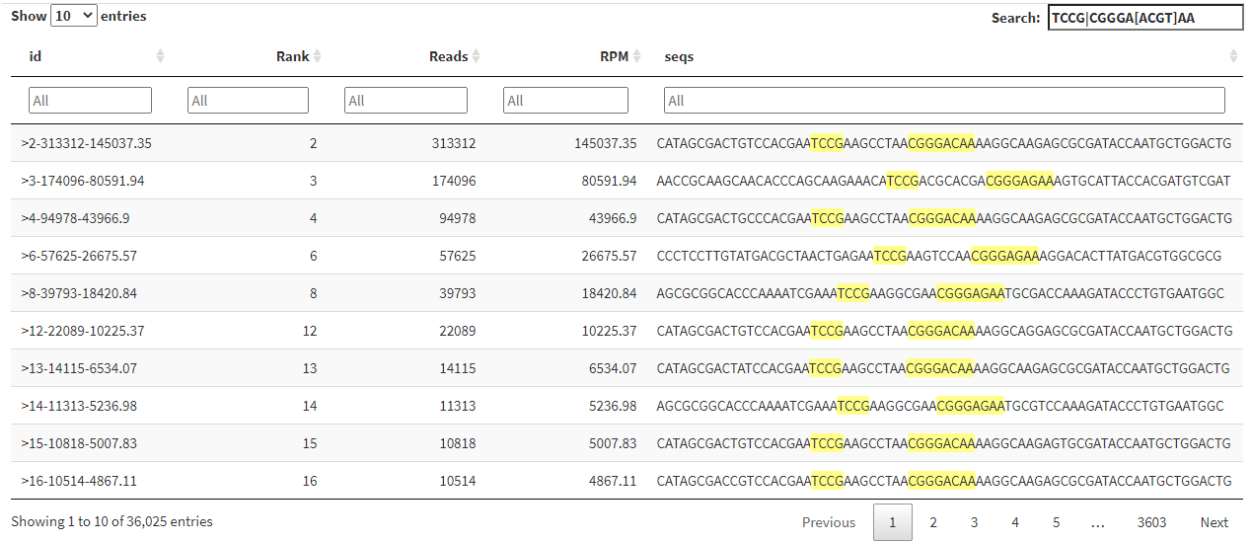


Figure 8: Screenshot of FASTAptameR-Motif_Search.


Show 10 entries Search: TCCG|CGGGA[ACGT]AA

id	Rank	Reads	RPM	seqs
>2-313312-145037.35	2	313312	145037.35	CATAGCGACTGTCCACGAATCCGAAGCCTAACGGGACAAAAGGCAAGAGCGCGATACCAATGCTGGACTG
>3-174096-80591.94	3	174096	80591.94	AACCGCAAGCAACACCCAGCAAGAAACATCCGACGCACGACGGGAGAAAGTGCATTACCACGATGTCGAT
>4-94978-43966.9	4	94978	43966.9	CATAGCGACTGCCCACGAATCCGAAGCCTAACGGGACAAAAGGCAAGAGCGCGATACCAATGCTGGACTG
>6-57625-26675.57	6	57625	26675.57	CCCTCCTTGTATGACGCTAACTGAGAATCCGAAGTCCAACGGGAGAAAGGACACTTATGACGTGGCGCG
>8-39793-18420.84	8	39793	18420.84	AGCGCGGCACCCAAAATCGAAATCCGAAGGCGAACGGGAGAATGCGACCAAAGATACCCTGTGAATGGC
>12-22089-10225.37	12	22089	10225.37	CATAGCGACTGTCCACGAATCCGAAGCCTAACGGGACAAAAGGCAGGAGCGCGATACCAATGCTGGACTG
>13-14115-6534.07	13	14115	6534.07	CATAGCGACTATCCACGAATCCGAAGCCTAACGGGACAAAAGGCAAGAGCGCGATACCAATGCTGGACTG
>14-11313-5236.98	14	11313	5236.98	AGCGCGGCACCCAAAATCGAAATCCGAAGGCGAACGGGAGAATGCGTCCAAAGATACCCTGTGAATGGC
>15-10818-5007.83	15	10818	5007.83	CATAGCGACTGTCCACGAATCCGAAGCCTAACGGGACAAAAGGCAAGAGTGCGATACCAATGCTGGACTG
>16-10514-4867.11	16	10514	4867.11	CATAGCGACCGTCCACGAATCCGAAGCCTAACGGGACAAAAGGCAAGAGCGCGATACCAATGCTGGACTG

Showing 1 to 10 of 36,025 entries Previous 1 2 3 4 5 … 3603 Next

Figure 9: FASTAptameR-Motif_Search Output.

3.6 FASTAptameR-Motif_Tracker

3.6.1 Description

FASTAptameR-Motif_Tracker reports on the occurrence of one or more query patterns / sequences across multiple populations. The module accepts at least two *counted* FASTA files as input and returns a data table of metadata related to the enrichment of the query pattern(s) across multiple populations. Multiple FASTA files should be selected from the file browser at the same time. Columns of the output data table include the following:

1. Population
2. File name
3. Query
4. Rank
5. Reads
6. RPM

Optionally, an alias list can be provided and will be included as a separate column. These aliases will be used in the legend of the line plot. If provided, there must be one alias per query per line.

This output can be downloaded as a CSV file and will include appended enrichment scores. A screenshot of the module interface is shown in **Fig. 10**.

3.6.2 Usage

The input FASTA files must be chosen with the file browser. The following text box must contain at least one pattern. If the user wishes to search for multiple patterns, the patterns must be separated by commas. The set of radio buttons determines the type of pattern (`DEFAULT = Nucleotide`). If `Nucleotide`, then degenerate nucleotide codes are allowed. Note, the pattern is converted to uppercase and white spaces are removed regardless of the pattern type.

The `Start` button begins the motif enrichment process. The resulting data table will be shown on the right side of the screen. The `Download` button opens a file browser prior to downloading the output as a CSV file.

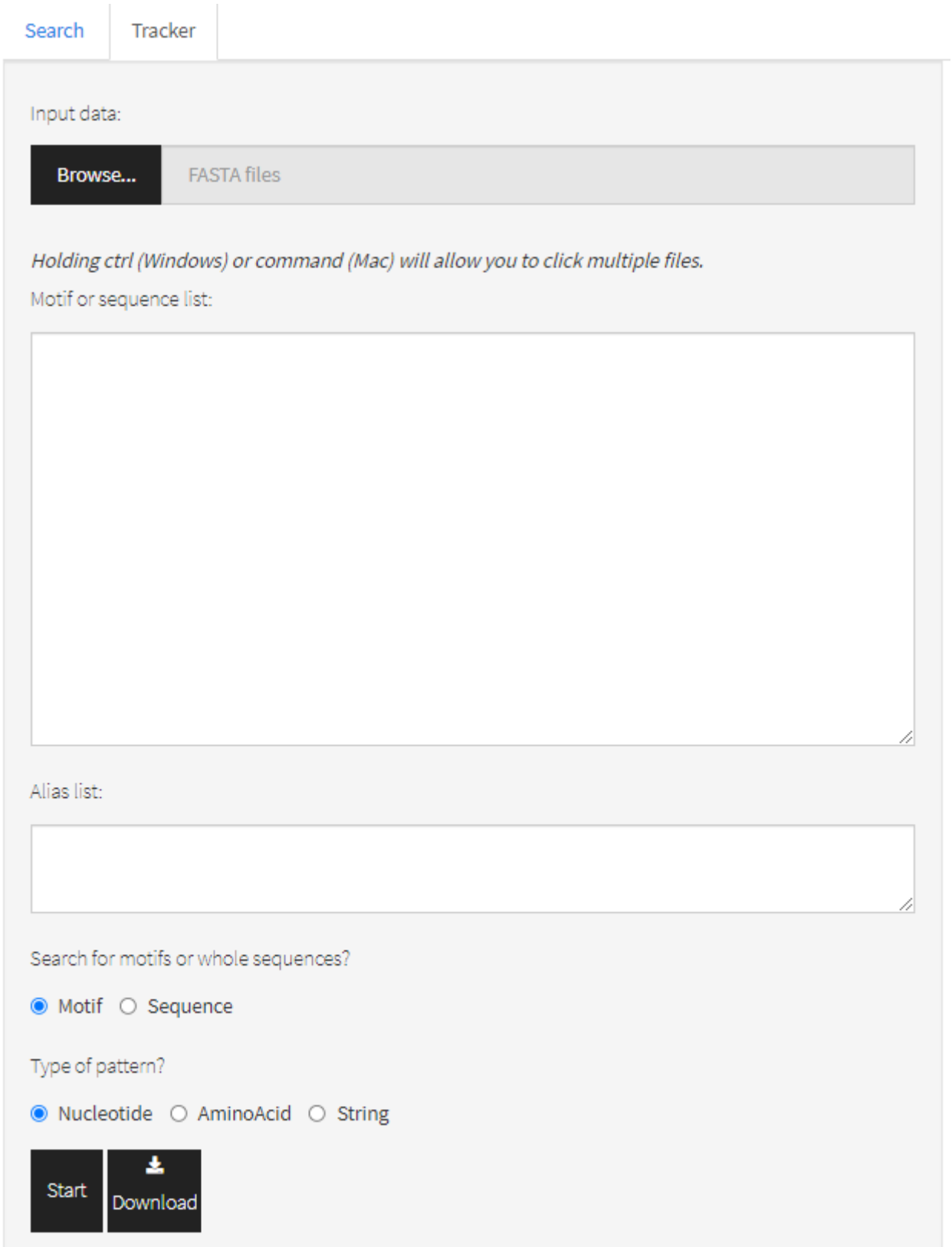


Figure 10: Screenshot of FASTAptameR-Motif_Tracker.

A screenshot of a sample output data table is shown in **Fig. 11** with the following pattern: `UCCGnnnnnnnnCGGGAnAA` (a Family 1 Pseudoknot).

Show 10 entries Search:

Population	FileName	seqs	Alias	Rank	Reads	RPM
1	70HRT14-count.fasta	AACCGCAAGCAACACCCAGCAAGAAACATCCGACGCACGACGGGAGAAAGTGCATTACCACGATGTCGAT	3rd	3	174096	80591.94
2	70HRT15-count.fasta	AACCGCAAGCAACACCCAGCAAGAAACATCCGACGCACGACGGGAGAAAGTGCATTACCACGATGTCGAT	3rd	5	104932	52786.23
1	70HRT14-count.fasta	ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCACGCTTGGTGT	1st	1	417696	193358.44
2	70HRT15-count.fasta	ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCACGCTTGGTGT	1st	3	161830	81408.87
1	70HRT14-count.fasta	CATAGCGACTGTCCACGAATCCGAAGCCTAACGGGACAAAAGGCAAGAGCGCGATACCAATGCTGGACTG	2nd	2	313312	145037.35
2	70HRT15-count.fasta	CATAGCGACTGTCCACGAATCCGAAGCCTAACGGGACAAAAGGCAAGAGCGCGATACCAATGCTGGACTG	2nd	1	382391	192362.47

Showing 1 to 6 of 6 entries Previous 1 Next

Figure 11: FASTAptameR-Motif_Tracker Output.

3.6.3 Plotting

This module can generate an interactive line plot showing the query's RPM across each population (**Fig. 12**).

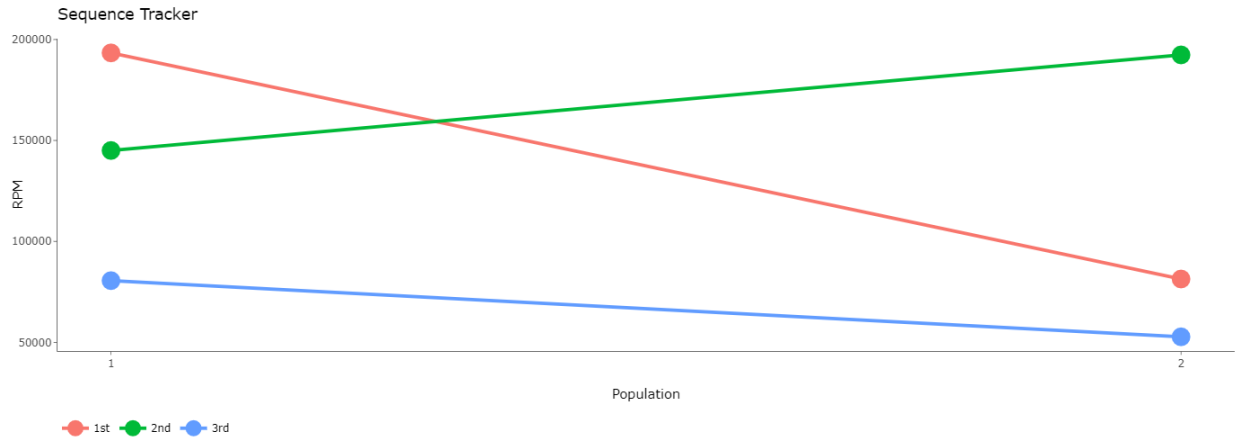


Figure 12: Sequence Tracking Line Plot. Shows the RPM of three sequences across the 70HRT14 and 70HRT15 populations. The aliases - 1st, 2nd, 3rd - refer to the 1st, 2nd, and 3rd most abundant sequences from the 70HRT14 population.

3.7 FASTAptameR-Distance

3.7.1 Description

FASTAptameR-Distance tabulates the distribution of distances from a user-defined reference sequence for all sequences in a population. The module accepts a *counted* FASTA file as input and returns a data table that contains a column for the Levenshtein edit distance (LED) between each input sequence and a query sequence. The output can be downloaded as a FASTA or CSV. A screenshot of the module interface is shown in **Fig. 13**.

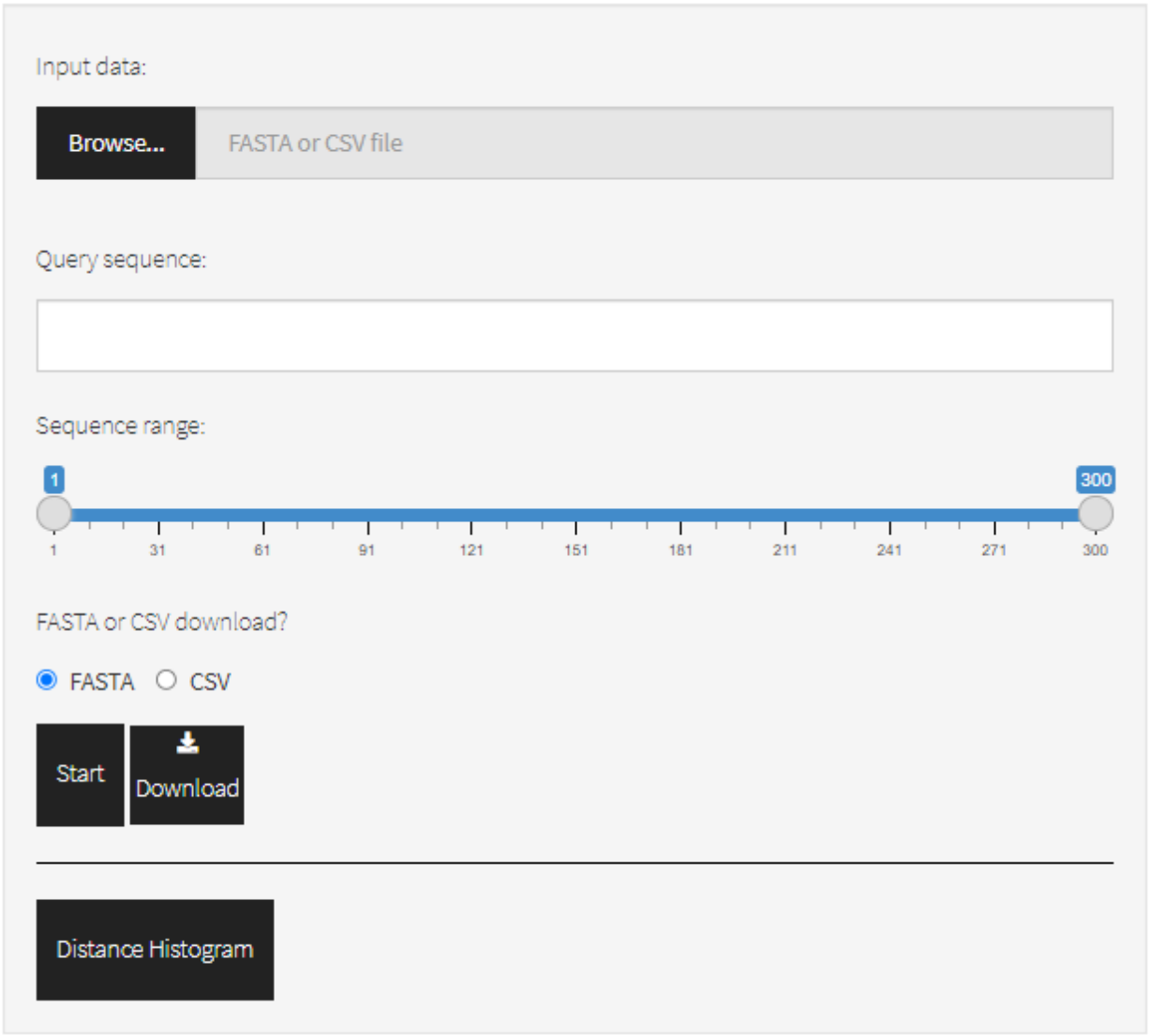


Figure 13: Screenshot of FASTAptameR-Distance.

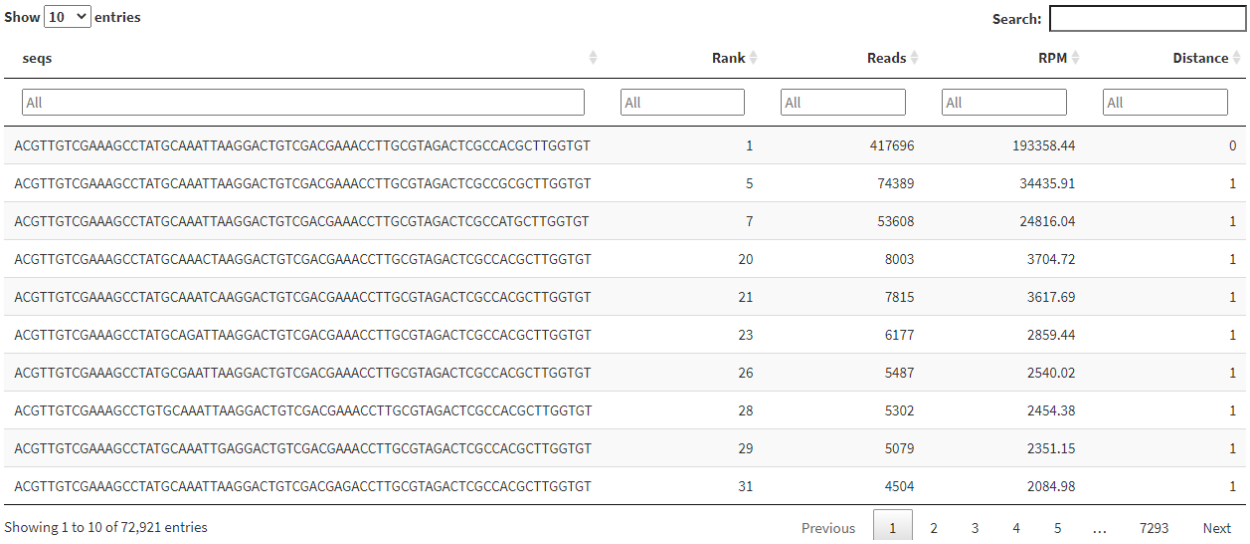
3.7.2 Usage

The input FASTA file must be chosen with the file browser, and the following text box must contain a single query sequence. Note, this query sequence may not have any degenerate nucleotide codes. The `Start` button

begins the distance calculations. The resulting data table will be shown on the right side of the screen. The `Download` button opens a file browser prior to downloading the output as a FASTA or CSV file.

A sample output data table is shown in **Fig. 14** with the following query sequence (the most abundant sequence from the 70HRT14 dataset):

`ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCACGCTTGGTGT`.



Show 10 entries — Search:

seqs	Rank	Reads	RPM	Distance
ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCACGCTTGGTGT	1	417696	193358.44	0
ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCGCGCTTGGTGT	5	74389	34435.91	1
ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCATGCTTGGTGT	7	53608	24816.04	1
ACGTTGTCGAAAGCCTATGCAAACTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCACGCTTGGTGT	20	8003	3704.72	1
ACGTTGTCGAAAGCCTATGCAAATCAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCACGCTTGGTGT	21	7815	3617.69	1
ACGTTGTCGAAAGCCTATGCAGATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCACGCTTGGTGT	23	6177	2859.44	1
ACGTTGTCGAAAGCCTATGCGAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCACGCTTGGTGT	26	5487	2540.02	1
ACGTTGTCGAAAGCCTGTGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCACGCTTGGTGT	28	5302	2454.38	1
ACGTTGTCGAAAGCCTATGCAAATTGAGGACTGTCGACGAAACCTTGCGTAGACTCGCCACGCTTGGTGT	29	5079	2351.15	1
ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAGACCTTGCGTAGACTCGCCACGCTTGGTGT	31	4504	2084.98	1

Showing 1 to 10 of 72,921 entries — Previous 1 2 3 4 5 … 7293 Next

Figure 14: FASTAptameR-Distance Output.

The slider bar allows the user to select a range of positions to query. For example, setting the two ends of this slider bar to 10 and 60 will truncate all of the sequences (**including the query sequence**) to be in that specific range. Thus, the resulting distance value will be the LED between positions 10-60 of the query sequence and positions 10-60 of every other sequence in the data. Note, this option is best used if the sequences have already been length-filtered in FASTAptameR-Count since this module will not align any sequences.

3.7.3 Plotting

This module can also generate interactive histograms of distances (**Fig. 15**). The top plot corresponds to the distances between the query and unique sequences, whereas the bottom plot corresponds to the distances between the query and all sequences. In both cases, the query sequence is displayed at the top of the plot.

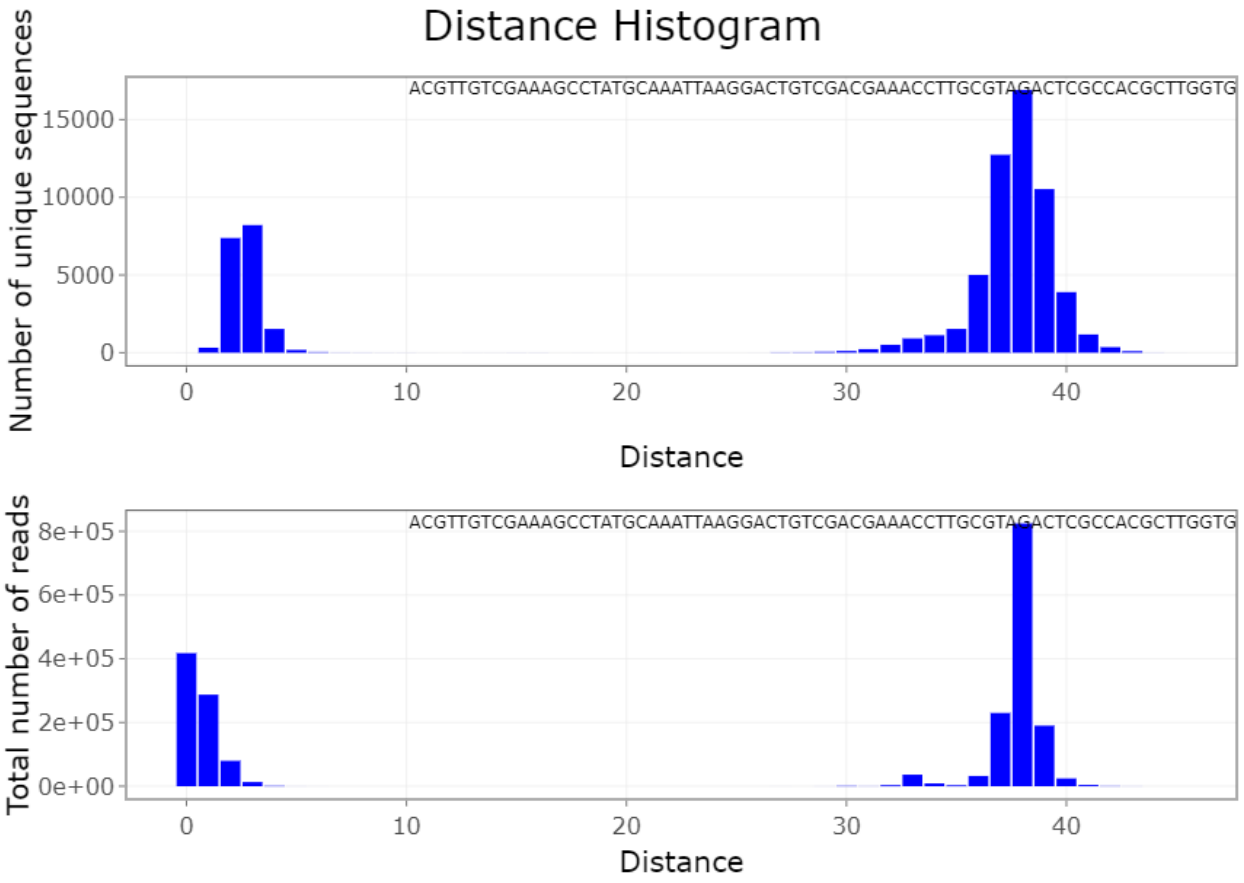


Figure 15: Distance Histogram. Two views of the 70HRT14 data, where the top and bottom panels correspond to unique sequences and total reads, respectively. Both plots are generated together in the UI.

3.8 FASTAptameR-Enrich

3.8.1 Description

FASTAptameR-Enrich calculates the enrichment (or depletion) of each sequence in one population relative to other populations. The module at least two *counted* FASTA files as input and returns a single data table after merging by sequences. Column headers for output data are appended with *.a*, *.b*, *.c*, *etc.*, depending on the order in which they are uploaded. Additional columns include enrichment scores ("Enrichment" = `RPM2 / RPM1`) and the base-two logarithm of the Enrichment ("log2(E)" = `log2(Enrichment)`). For simplicity, comparisons are only made between consecutive populations (*e.g.*, 2:1, 3:2, *etc.*). This output can be downloaded as a CSV.

A screenshot of the module interface is shown in **Fig. 16**.

3.8.2 Usage

The input FASTA files must be chosen with the file browser. The following set of radio buttons determine whether missing values are allowed in the output. Missing values result from sequences that are only present in a subset of the input files.

The `Start` button begins the enrichment calculations, and the resulting data table will be shown on the right side of the screen. All numeric columns in this data table are filterable by typing into the corresponding text box (*e.g.*, `1 ... 10` to keep values in the range `[1:10]`) or by using the slider bar that is displayed after clicking in the corresponding text box. Note, these filters apply the mask only to the displayed data, so calculations will **not** be repeated when the filters are altered. To display all data again, delete the filters from the text boxes. Note, many other outputs are similarly filterable.

The `Download` button opens a file browser prior to downloading the output as a CSV file. A sample filtered output data table is shown in **Fig. 17**.

3.8.3 Plotting

This module can generate five types of interactive plots: sequence persistence bar plots (**Fig. 18**), $\log_2(Enrichment)$ histograms (one per comparison - **Fig. 19**), RPM scatter plots (one per comparison - **Fig. 20**), volcano plots (one per comparison - **Fig. 21**), and a cluster box plot in the case where clustered (not counted) FASTAs are provided (**Fig. 22**).

The sequence persistence bar plot labels sequences with how many rounds they are found in all uploaded FASTA files. The slider bar just above this plot button filters these sequences by their respective read count.

The spread of the $\log_2(Enrichment)$ histogram relative to a vertical line at `x = 0` can indicate the magnitudes of enrichment (or depletion), while displacement of the centroid of the distribution from this line can indicate possible directionality of the population's evolution.

Similarly, the spread of the RPM scatter plot relative to the diagonal line at `y = x` can also indicate the magnitudes of enrichment and possible directionality.

The volcano plot is used to show the relationship between the respective $\log_2(Enrichment)$ and number of reads for each sequence, which is an indication of the statistical strength of the observed ratios. Specifically, for the purposes of this module, the y-axis is given by $y(seq) = \sqrt{\log_2(seq.Reads)}/\log_2(total.Reads)$, based on the fact that the variance of sample count for a species from a resampled population varies with the square root of the number of times a given species is resampled.

The cluster box plot shows the distribution of enrichment values for sequences after grouping by cluster. The red marker indicates where the seed sequence of the cluster falls.

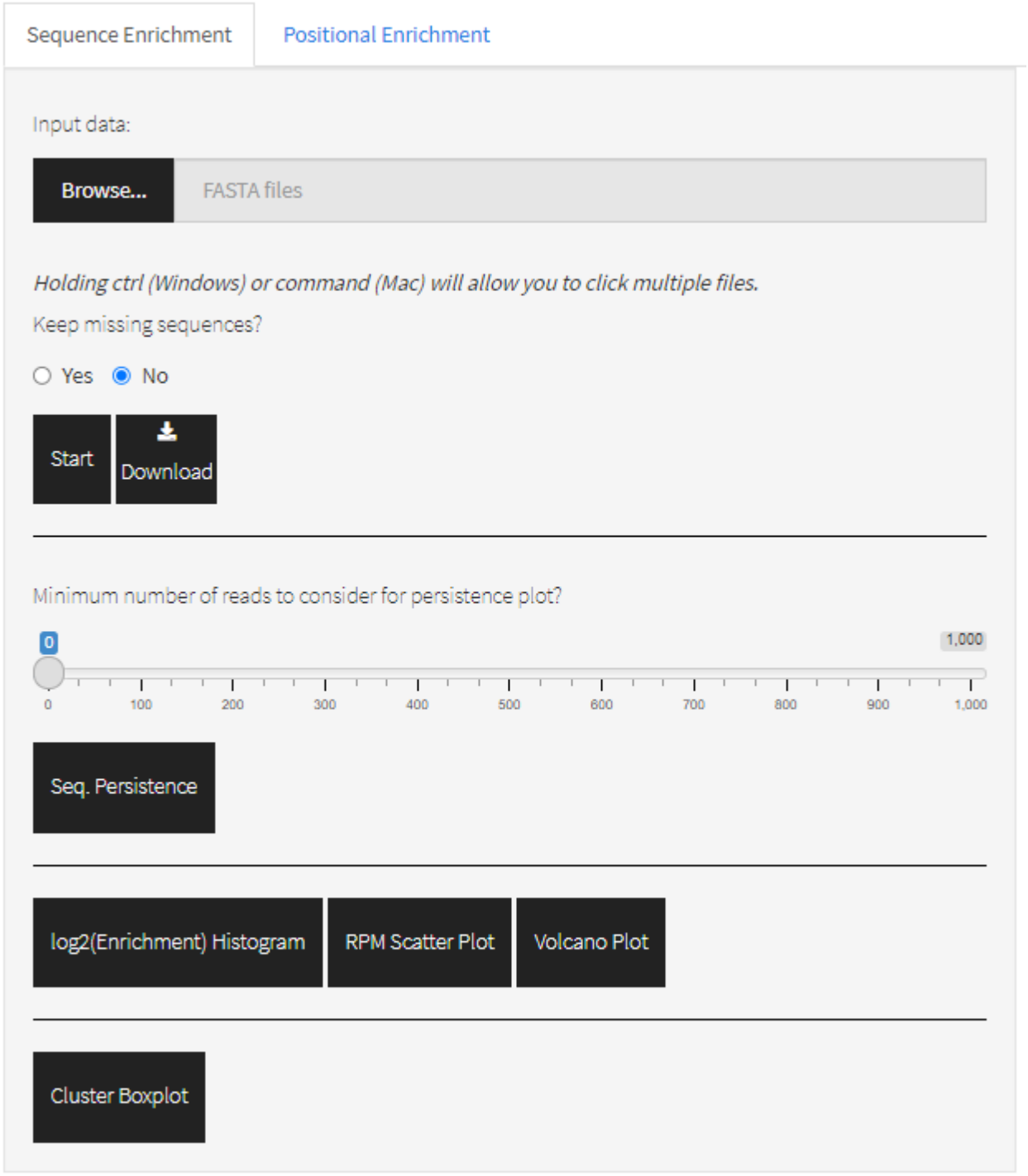


Figure 16: Screenshot of FASTAptameR-Enrich.

Show 10 entries Search:

seqs	Rank.a	Reads.a	RPM.a	Rank.b	Reads.b	RPM.b	enrichment_ba	log2E_ba
All	All	All	All	All	All	All	All	All
ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCACGCTTGGTGT	1	417696	193358.44	3	161830	81408.87	0.421	-1.248
CATAGCGACTGTCCACGAATCCGAAGCCTAACGGGACAAAAGGCAAGAGCGCGATACCAATGCTGGACTG	2	313312	145037.35	1	382391	192362.47	1.326	0.407
AACCGCAAGCAACACCCAGCAAGAAACATCCGACGCACGACGGGAGAAAGTGCATTACCACGATGTCGAT	3	174096	80591.94	5	104932	52786.23	0.655	-0.61
CATAGCGACTGCCCACGAATCCGAAGCCTAACGGGACAAAAGGCAAGAGCGCGATACCAATGCTGGACTG	4	94978	43966.9	6	42954	21608.09	0.491	-1.025
ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCGCGCTTGGTGT	5	74389	34435.91	9	32821	16510.66	0.479	-1.061
CCCTCCTTGTATGACGCTAACTGAGAATCCGAAGTCCAACGGGAGAAAGGACACTTATGACGTGGCGCG	6	57625	26675.57	7	37701	18965.55	0.711	-0.492
ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCATGCTTGGTGT	7	53608	24816.04	10	30749	15468.34	0.623	-0.682
AGCGCGGCACCCAAAATCGAAATCCGAAGGCGAACGGGAGAATGCGACCAAAGATACCCTGTGAATGGC	8	39793	18420.84	15	15414	7754.04	0.421	-1.248
TTGACAATAACTCGAGAAGAACCGAGGTGCAAACGGGAGAACACAATGGATTACACCGAGCTCGGCTGAC	9	33800	15646.58	4	136505	68669.08	4.389	2.134
GCGAACCAAACCCAGATTACTAACCGTGGGCCTGAAACACGGGACAAAACAGGCATCAATGGAGTGGTAC	10	29794	13792.14	176	732	368.23	0.027	-5.227

Showing 1 to 10 of 21,924 entries Previous 1 2 3 4 5 … 2193 Next

Figure 17: Filtered FASTAptameR-Enrich Output.

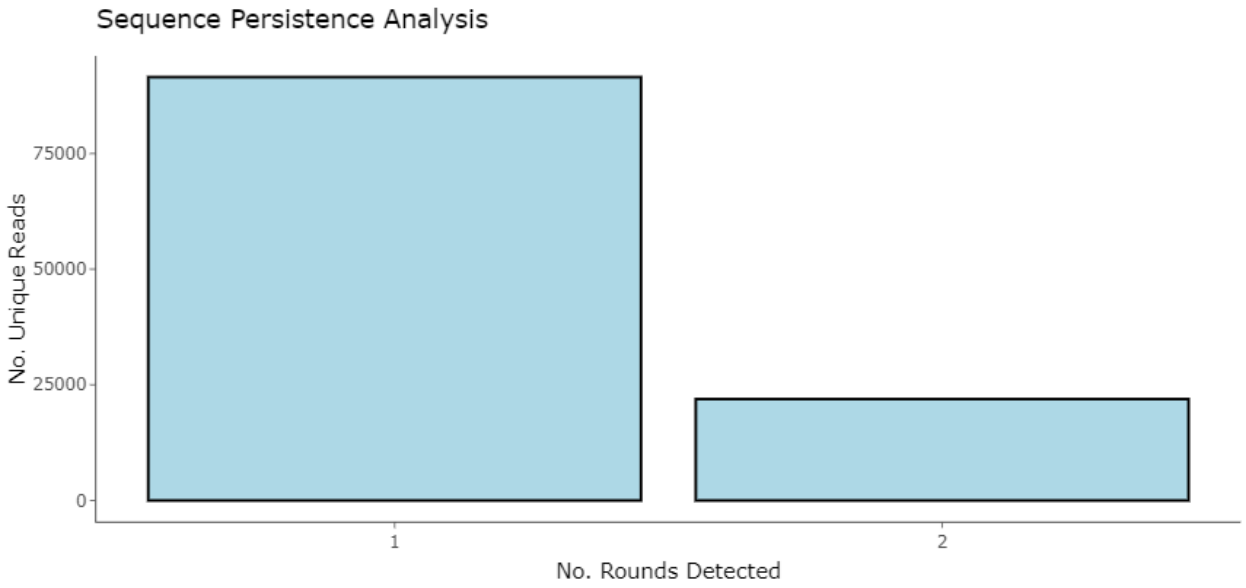


Figure 18: Sequence Persistence Bar Plot. Approximately 25000 sequences were found in both populations, whereas approximately 80000 sequences were found in only one population.

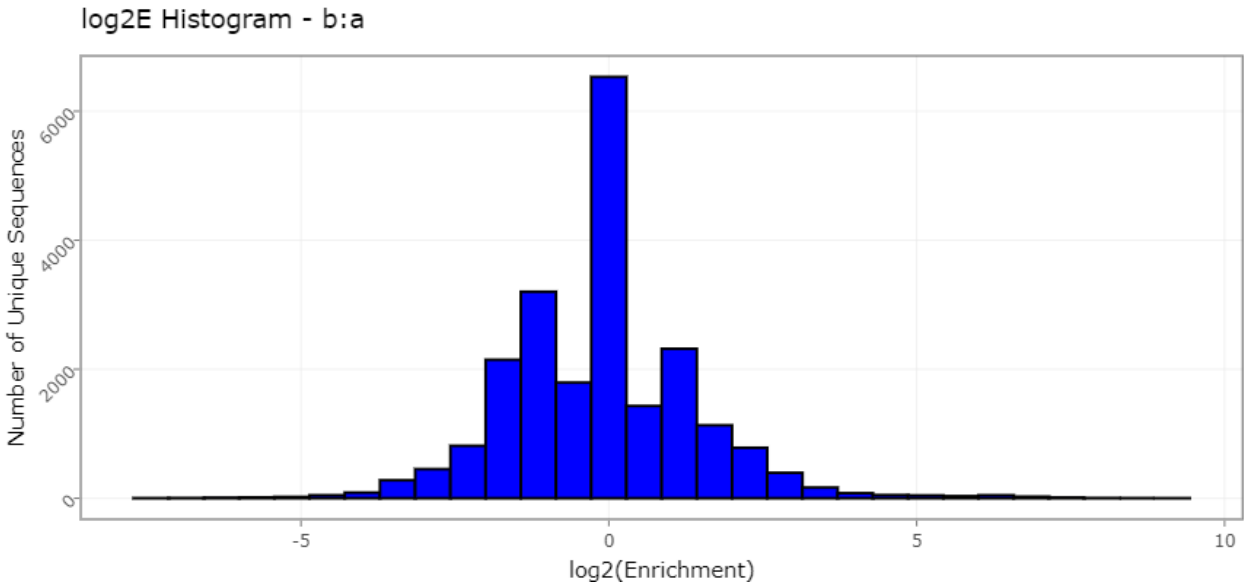


Figure 19: $\log_2(Enrichment)$ Histogram.

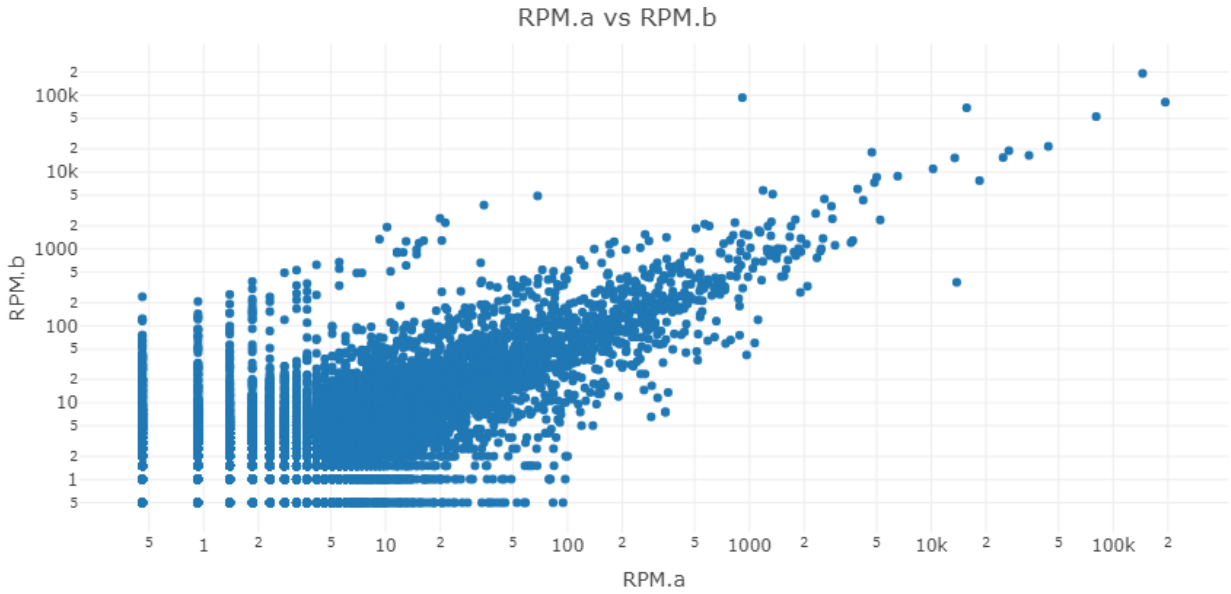


Figure 20: RPM Scatter Plot.

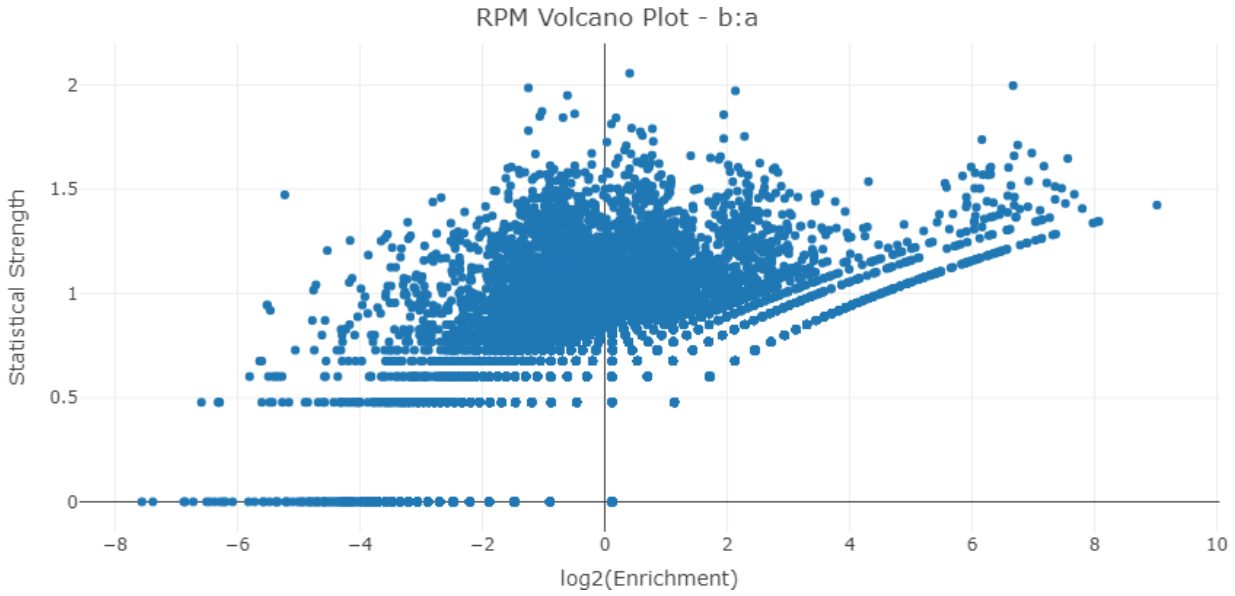


Figure 21: Volcano Plot.

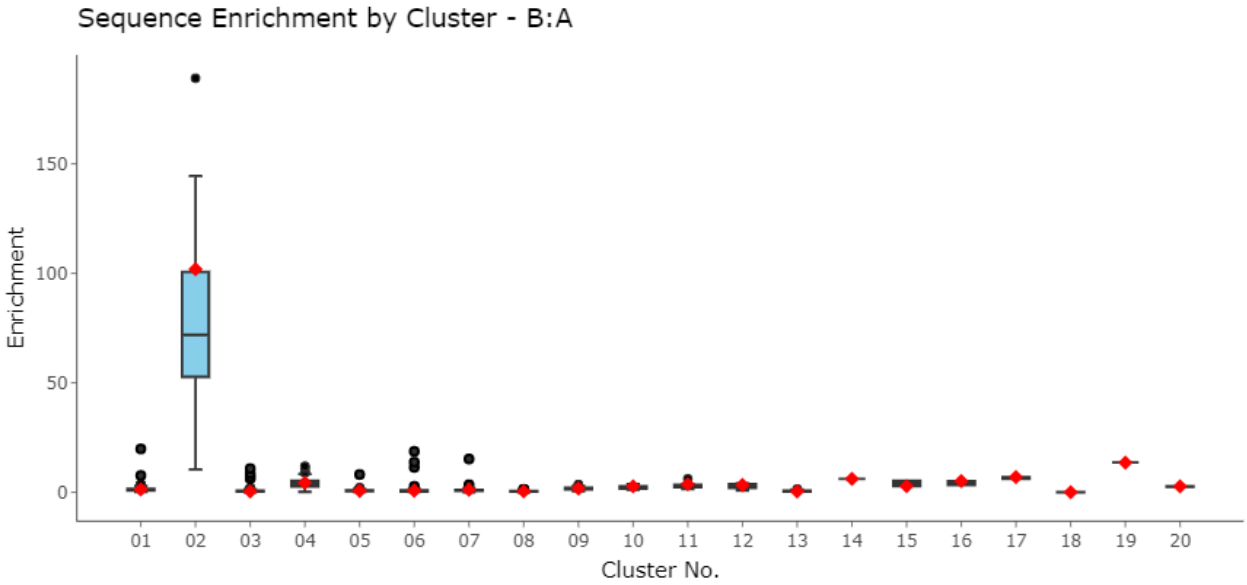


Figure 22: Cluster Box Plot of Sequence Enrichment.

3.9 FASTAptameR-Positional_Enrichment

3.9.1 Description

FASTAptameR-Positional_Enrichment calculates the average enrichment (or depletion) for each position that does not match the corresponding residue in the user-defined reference sequence. For example, if the first residue of the reference sequence is `E`, then this module will calculate the average enrichment of all sequences that do not have an `E` in the first position. Thus, it is recommended that all sequences are of the same length (can be done by applying a filter to the output table from FASTAptameR-Count).

This module accepts a CSV from the previous module, though it exclusively operates on the `enrichment_ba` column. Thus, an enrichment CSV with >2 populations can be uploaded here, but only the first enrichment column will be used.

The outputs of this module are two plots. The first is a bar plot showing the average enrichment value for each position. The second is a heat map that shows the average enrichment per position per residue.

A screenshot of the module interface is shown in **Fig. 23**.

3.9.2 Usage

The input CSV file must be chosen with the file browser, and the reference sequence must be added in the subsequent text box. The first set of radio buttons allows the standard genetic code to be altered. Each change should occupy a single line. To add a residue, enter its single-letter code. To replace a residue, enter a comma-separated pair (*e.g.*, `A,B` will replace `A` with `B` in the algorithm and resulting plots).

The slider bar allows the user to set the minimum and maximum enrichment values (*e.g.*, `0-10` means that any value greater than 10 is made equal to 10). The final set of radio buttons determines the set of residues for which the algorithm searches. The next three text boxes allow the user to set the "low", "middle", and "high" colors for the plots.

Finally, the `Start` button generates the two plots.

3.9.3 Plotting

FASTAptameR-Positional_Enrichment generates two types of plots: 1) average enrichment bar plot and 2) average enrichment heat map.

The bar plot (**Fig. 24**) shows the reference sequence on the x-axis and the average enrichment on the y-axis. The heat map (**Fig. 25**) shows the reference sequence on the x-axis, possible residues on the y-axis, and average enrichment in the color axis.

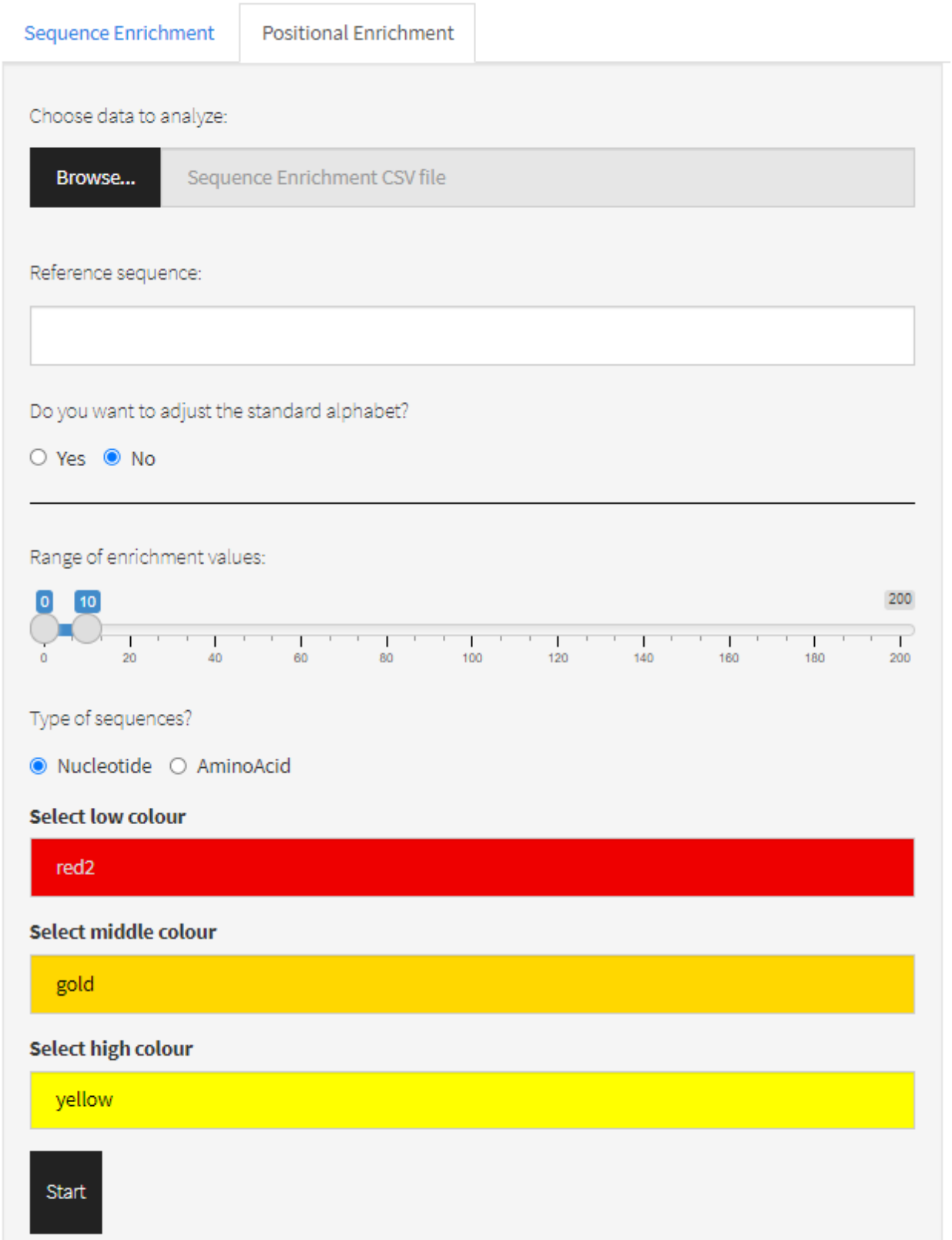


Figure 23: Screenshot of FASTAptameR-Positional_Enrichment.

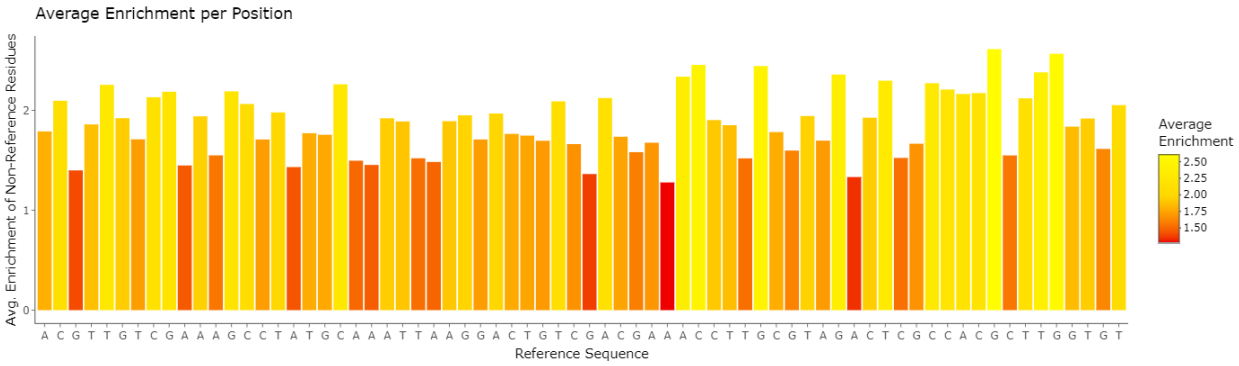


Figure 24: Average Enrichment Bar Plot. Shows the user-defined reference sequence on the x-axis and average enrichment of non-reference residues on the y-axis.

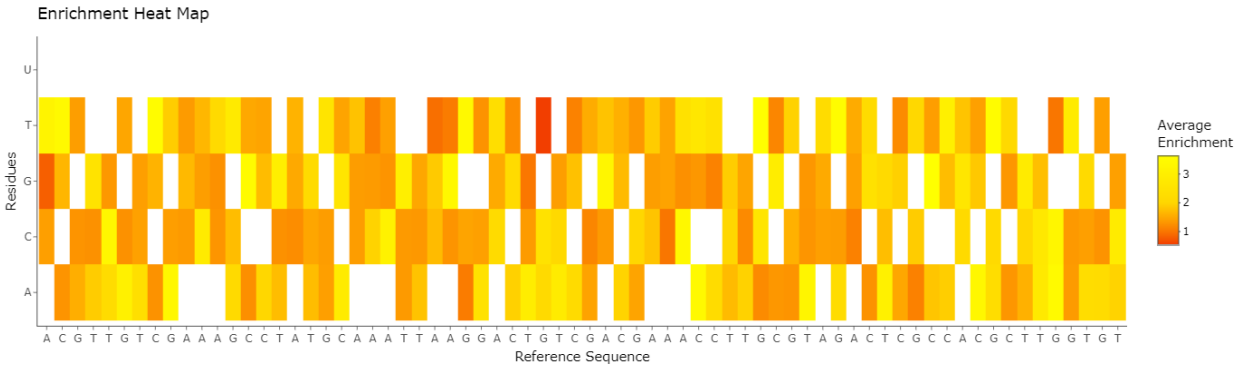


Figure 25: Average Enrichment Heat Map. Shows the user-defined reference sequence on the x-axis and all possible residues on the y-axis (nucleotides for this use case). Colors depict average enrichment.

3.10 FASTAptameR-Cluster

3.10.1 Description

FASTAptameR-Cluster groups sequences according to sequence relatedness for all sequences in the population within a user-defined threshold of similarity. The module accepts a *counted* FASTA file as input. If no output directory is specified (the default setting), the module returns a *clustered* data table to the screen. This data table contains all sequences and clusters and can be downloaded as a single FASTA or CSV file. If an output directory is specified, then no data table will be created, and one FASTA file per cluster will be written to the output directory.

Briefly, the module identifies clusters in an iterative manner. During each iteration, the most abundant sequence that has not yet been clustered becomes a cluster "seed" for that iteration. Any other sequences that have not yet been clustered and that are within a user-defined edit distance of this seed sequence are added to this cluster. This process repeats until all sequences are clustered or a predefined number of clusters is created.

A screenshot of the module interface is shown in **Fig. 26**.

3.10.2 Usage

The input FASTA file must be chosen with the file browser. The 1st slider bar (**Fig. 26a**) sets the minimum number of reads a sequence must have to be clustered (`DEFAULT = 10`). Sequences with fewer reads are removed prior to clustering. The 2nd slider bar (**Fig. 26b**) sets the maximum Levenshtein edit distance to consider between a seed sequence and all other sequences (`DEFAULT = 7`). The 3rd slider bar (**Fig. 26c**) sets the total number of desired clusters (`DEFAULT = 20`). Note, any remaining sequences will be grouped as `NC` ("not clustered").

The 1st set of radio buttons (**Fig. 26d**) indicates whether non-clustered sequences should be kept (`DEFAULT = No`). If `Yes` then the sequence IDs of non-clustered sequences will be appended with `NC`.

The 2nd set of radio buttons (**Fig. 26e**) indicate whether each cluster should be written to a different FASTA file (`DEFAULT = No`). If `No`, then all clusters are grouped together and downloaded in a single file. If `Yes`, then each cluster will be written to its own FASTA file, and no data table will be displayed. Note, a directory path **must be copied or typed** into the corresponding text box (**Fig. 26f**) if this option is `Yes`.

The `Start` button will begin the clustering process. The results will be displayed as a data table on the right side of the screen. The `Download` button opens a file browser prior to downloading the output as a FASTA or CSV file (`DEFAULT = FASTA`, which is required for subsequent modules).

Algorithm progress will be shown below these buttons and will update after each cluster finishes. These notifications occur regardless of whether the module is writing to one or many files.

A sample output data table is shown in **Fig. 27**.

Note that the new *id* column is the old *id* with three new values appended to the end: `Cluster Number`, `Rank in Cluster`, and `Distance to Cluster Seed`.

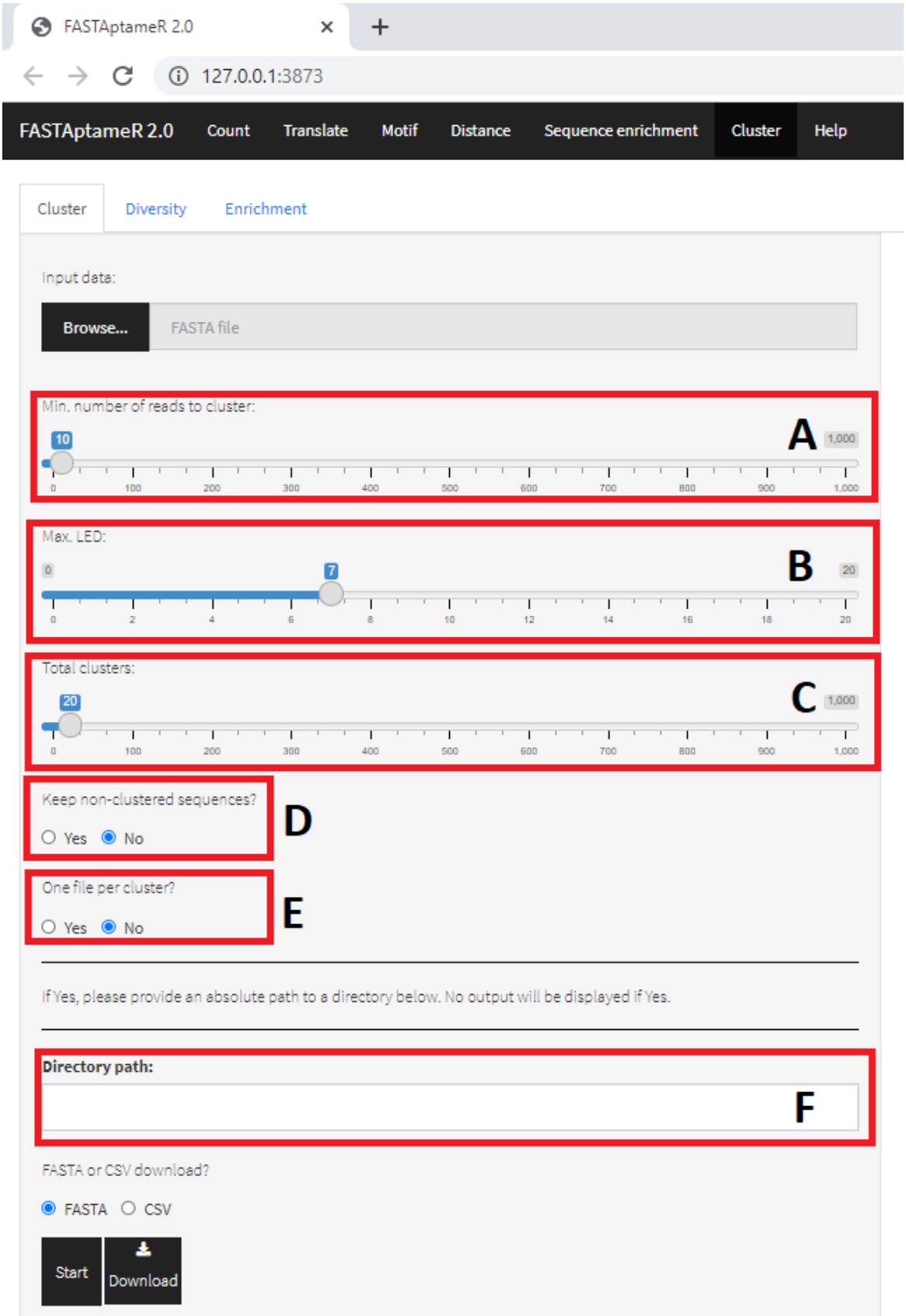


Figure 26: Screenshot of FASTAptameR-Cluster.

Show 10 entries Search:

id	Rank	Reads	RPM	cluster	rankInCluster	LED	seqs
All	All	All	All	All	All	All	All
>1-417696-193358.44-1-1-0	1	417696	193358.44	1	1	0	ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCACGCTTGGTGT
>2-313312-145037.35-2-1-0	2	313312	145037.35	2	1	0	CATAGCGACTGTCCACGAATCCGAAGCCTAACGGGACAAAAGGCAAGAGCGCGATACCAATGCTGGACTG
>3-174096-80591.94-3-1-0	3	174096	80591.94	3	1	0	AACCGCAAGCAACACCCAGCAAGAAACATCCGACGCACGACGGGAGAAAGTGCATTACCACGATGTCGAT
>4-94978-43966.9-2-2-1	4	94978	43966.9	2	2	1	CATAGCGACTGCCCACGAATCCGAAGCCTAACGGGACAAAAGGCAAGAGCGCGATACCAATGCTGGACTG
>5-74389-34435.91-1-2-1	5	74389	34435.91	1	2	1	ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCGCGCTTGGTGT

Figure 27: FASTAptameR-Cluster Output.

3.11 FASTAptameR-Cluster_Diversity

3.11.1 Description

FASTAptameR-Cluster_Diversity evaluates diversity across the *clustered* population and sequence relationships within and between clusters. The module accepts a *clustered* FASTA file as input and returns a data table with metadata for each cluster. This data table can be downloaded as a CSV file.

A screenshot of the module interface is shown in **Fig. 28**.

3.11.2 Usage

The input FASTA file must be chosen with the file browser. The `Start` button begins the analysis. The results will be displayed as a data table on the right side of the screen and include the following columns: `Cluster Number`, `Seed Sequence`, `Total Sequences`, `Total Reads`, and `Total RPM`. The `Download` button opens a file browser prior to downloading the output as a CSV file, which can be used by FASTAptameR-Cluster_Enrich. A sample output data table is shown in **Fig. 29**.

3.11.3 Plotting

This module can generate metaplots of the analyzed data (**Fig. 28a**). These line plots correspond to the number of unique sequences per cluster, total reads per cluster, and average LED to seed sequence per cluster (**Fig. 30**).

This module is also able to analyze clusters by converting all sequences into k-mer vectors and rendering an interactive 2D PCA plot (**Fig. 28b**), colored by cluster (**Fig. 31**). The value of `k` can be chosen with the 1st set of radio buttons (**Fig. 28c**) (`DEFAULT = 3`). The slider bar (**Fig. 28d**) indicates how many of the top clusters should be plotted (max = 21 clusters due to graphics limitations). The 2nd set of radio buttons (**Fig. 28e**) indicates whether non-clustered (`NC`) sequences should be plotted (`DEFAULT = Yes`). Note that non-clustered (`NC`) sequences in the output are marked as `NA` in this plot.

Importantly, only nucleotide sequences without ambiguities should be plotted in this module. The large k-mer matrix needed for peptide sequences may return errors related to memory usage. Further, this module will reject any set of sequences with characters outside of `[A, C, G, T/U]`. The resulting k-mer PCA plot will not display anything.

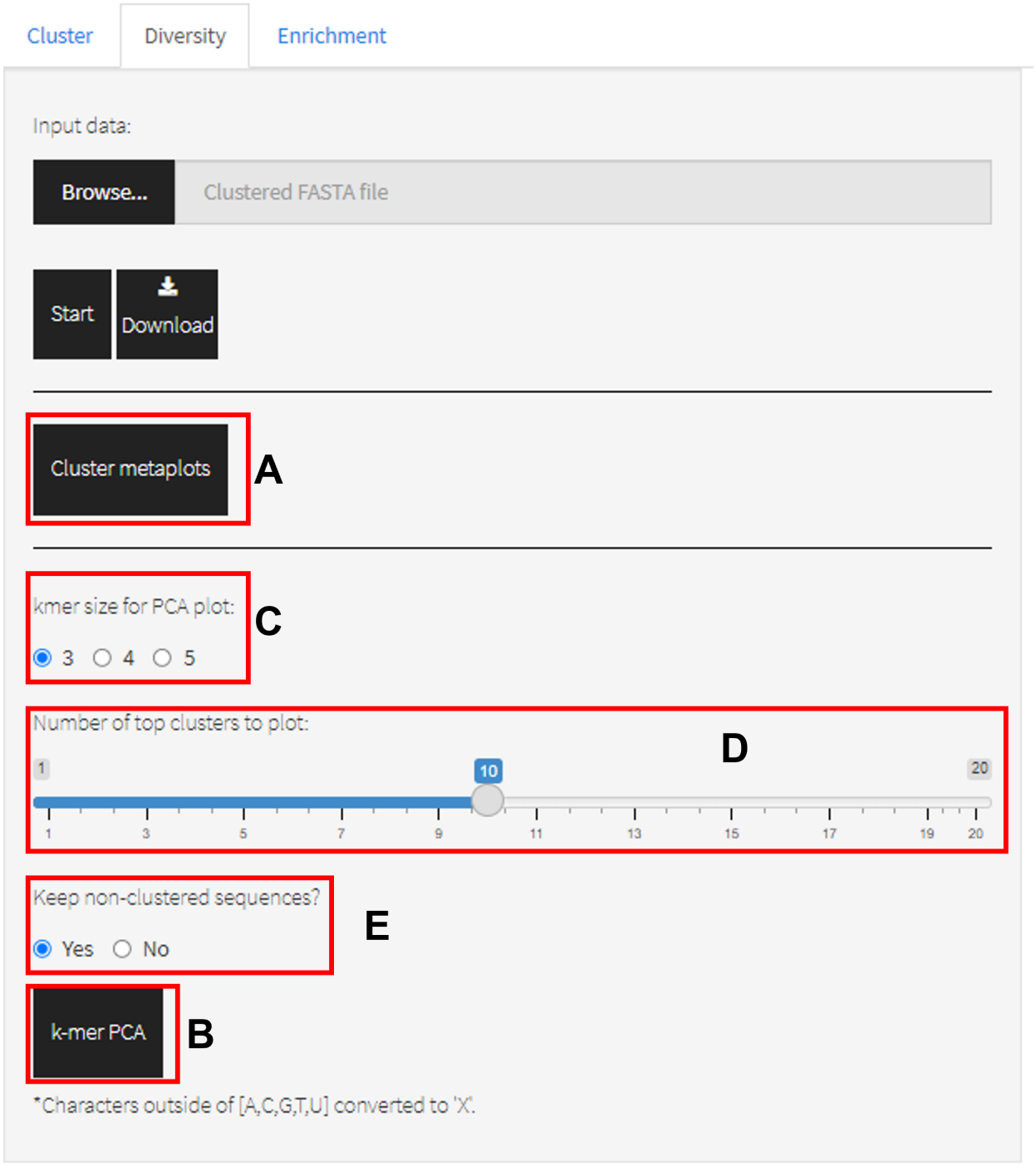


Figure 28: Screenshot of FASTAptameR-Cluster_Diversity.

Show 10 entries Search:

Cluster	Seeds	TotalSequences	TotalReads	TotalRPM	AverageLED
All	All	All	All	All	All
1	ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCACGCTTGGTGT	1259	770383	356623.54	1.86
2	CATAGCGACTGTCCACGAATCCGAAGCCTAACGGGACAAAAGGCAAGAGCGCGATACCAATGCTGGACTG	1295	652468	302038.75	1.97
3	AACCGCAAGCAACACCCAGCAAGAAACATCCGACGCACGACGGGAGAAAGTGCATTACCACGATGTCGAT	528	257675	119282.23	1.63
4	CCCTCCTTGTATGACGCTAACTGAGAATCCGAAGTCCAACGGGAGAAAGGACACTTATGACGTGGCGCG	324	101448	46962.14	1.49
5	AGCGCGGCACCCAAAATCGAAATCCGAAGGCGAACGGGAGAATGCGACCAAAGATACCCTGTGAATGGC	262	73809	34167.47	1.47
6	TTGACAATAACTCGAGAAGAACCGAGGTGCAAACGGGAGAACACAATGGATTACACCGAGCTCGGCTGAC	275	67908	31435.87	1.51
7	GCGAACCAAACCCAGATTACTAACCGTGGGCCTGAAACACGGGACAAAACAGGCATCAATGGAGTGGTAC	148	41566	19241.67	1.16
8	ACGTTGTGCACGGATGCCCACGGTCGCACGAAACCTTGTGTGGGATAGCGCGAATACTGACGAGTGTGCC	163	44693	20689.16	1.26
9	ACCAAATCCCGAACTACAAATCCGAACGCTAACGGGACAATTGCGAAATGGAACATACGGGCCTGGTTGA	67	9114	4219.07	1.1
10	GTGCGCTACCACATGATCCGAGGCAAAACGGGAAAAGATAGCATCGATTACGGAACCGGCCACGCACA	54	7292	3375.58	0.98

Showing 1 to 10 of 21 entries Previous 1 2 3 Next

Figure 29: FASTAptameR-Cluster_Diversity Output.

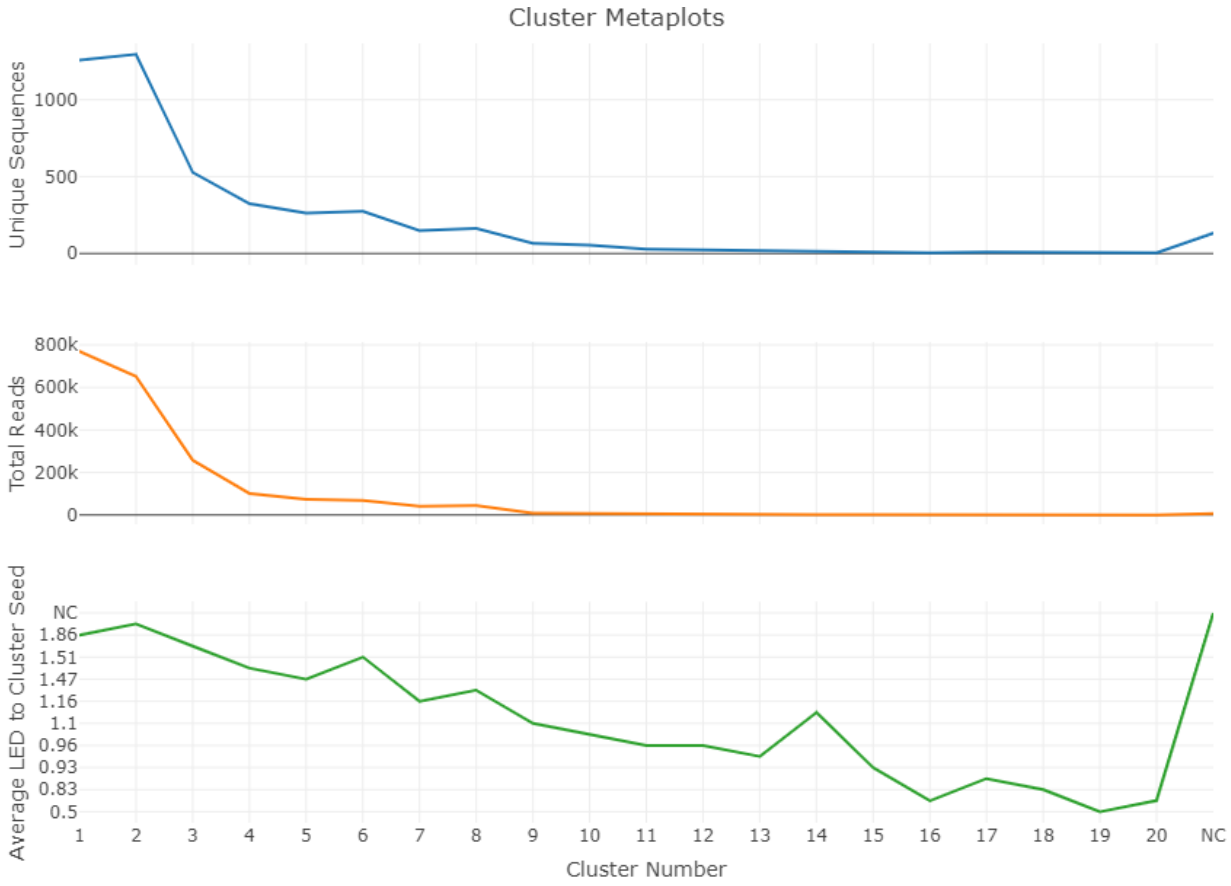


Figure 30: Cluster metaplots

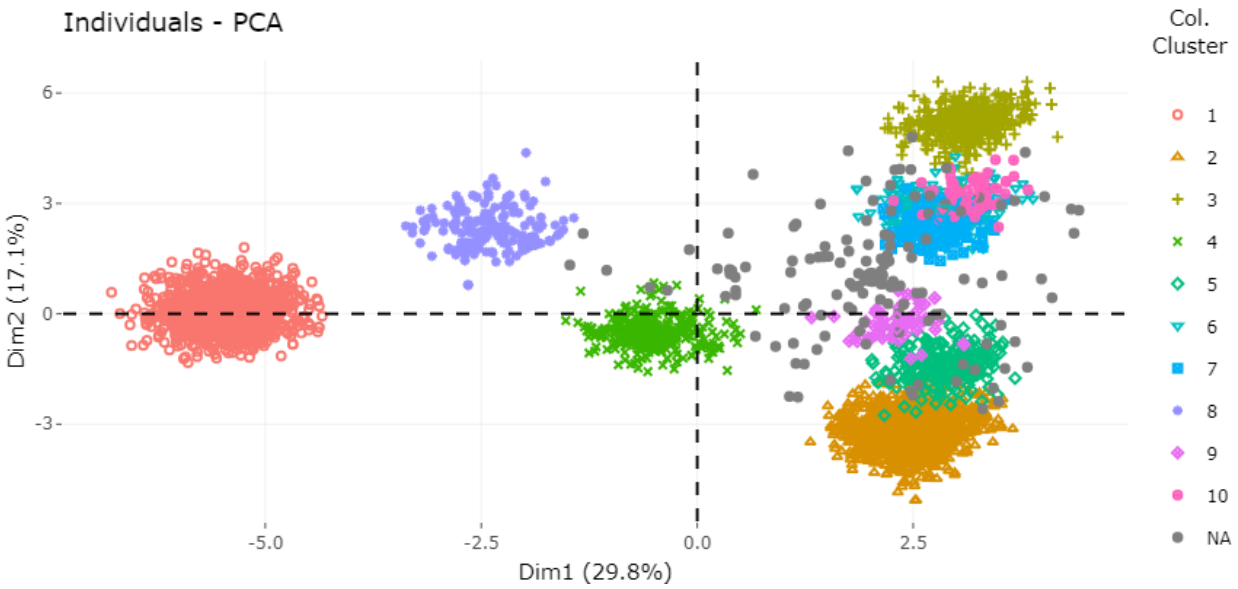


Figure 31: 4-mer PCA of top 20 clusters.

3.12 FASTAptameR-Cluster_Enrich

3.12.1 Description

FASTAptameR-Cluster_Enrich calculates the enrichment (or depletion) of each cluster in one population relative to other populations. The module accepts two or three *cluster-analysis* CSV files as input and returns a data table after merging by `Seed Sequence`. Thus, this module assumes that cluster seeds are consistent across populations, though this may not always be a valid assumption.

The output data table includes a column for `Enrichment` that can be downloaded as a CSV file. A screenshot of the module interface is shown in **Fig. 32**.

3.12.2 Usage

The input FASTA files must be chosen with the file browser. The `Start` button begins the enrichment calculation. The results will be displayed as a data table on the right side of the screen and include a column for `Enrichment`. The `Download` button opens a file browser prior to downloading the output as a CSV file.

A sample output data table is shown in **Fig. 33**.

Note that columns 3-5 and 7-9 refer to *total* values in the given cluster.

3.13 Plotting

After merging by seed sequence, this module will generate a line plot in which the x-axis corresponds to population, and the y-axis corresponds to the total RPM of the seed's cluster (**Fig. 34**).

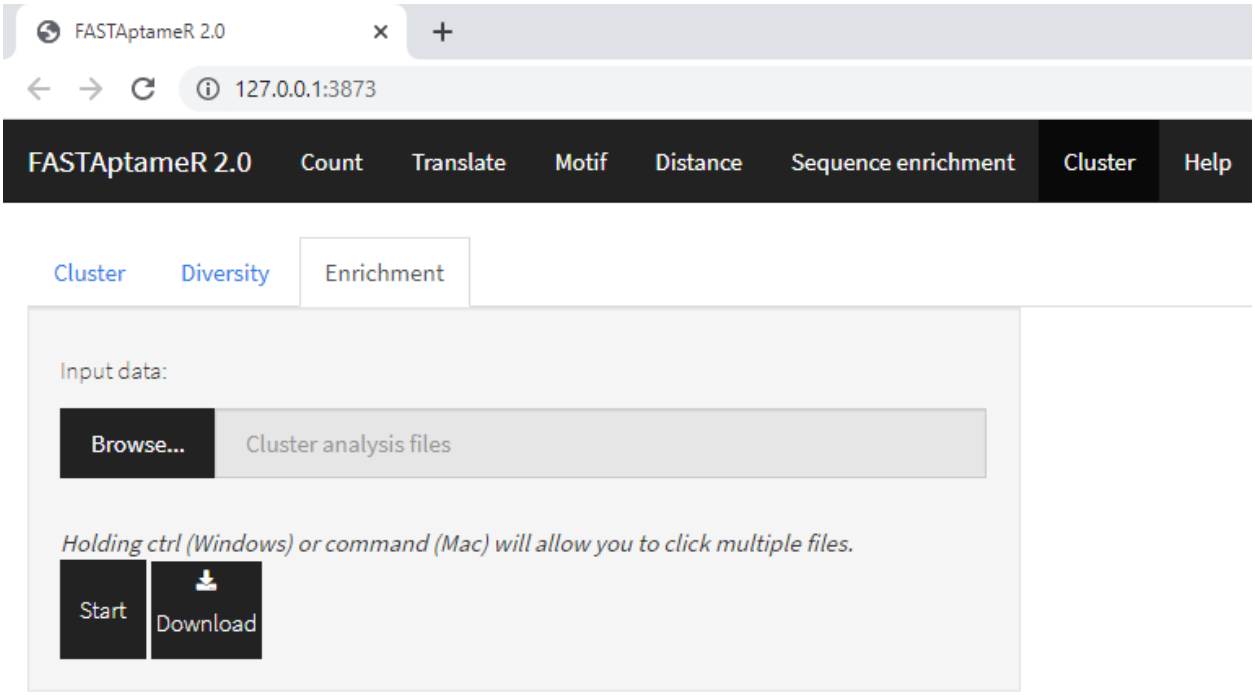


Figure 32: Screenshot of FASTAptameR-Cluster_Enrich.

Show 10 entries Search:

Cluster	Seeds	TotalSequences	TotalReads	TotalRPM	AverageLED	Population	FileName
All	All	All	All	All	All	All	All
1	ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCACGCTTGGTGT	1259	770383	356623.54	1.86	1	70HRT14-count-cluster-clusterDiversity.csv
2	CATAGCGACTGTCCACGAATCCGAAGCCTAACGGGACAAAAGGCAAGAGCGCGATACCAATGCTGGACTG	1295	652468	302038.75	1.97	1	70HRT14-count-cluster-clusterDiversity.csv
3	AACCGCAAGCAACACCCAGCAAGAAACATCCGACGCACGACGGGAGAAAGTGCATTACCACGATGTCGAT	528	257675	119282.23	1.63	1	70HRT14-count-cluster-clusterDiversity.csv
4	CCCTCCTTGTATGACGCTAACTGAGAATCCGAAGTCCAACGGGAGAAAGGACACTTATGACGTGGCGCG	324	101448	46962.14	1.49	1	70HRT14-count-cluster-clusterDiversity.csv
5	AGCGCGGCACCCAAAATCGAAATCCGAAGGCGAACGGGAGAATGCGACCAAAGATACCCTGTGAATGGC	262	73809	34167.47	1.47	1	70HRT14-count-cluster-clusterDiversity.csv

Figure 33: FASTAptameR-Cluster_Enrich Output.

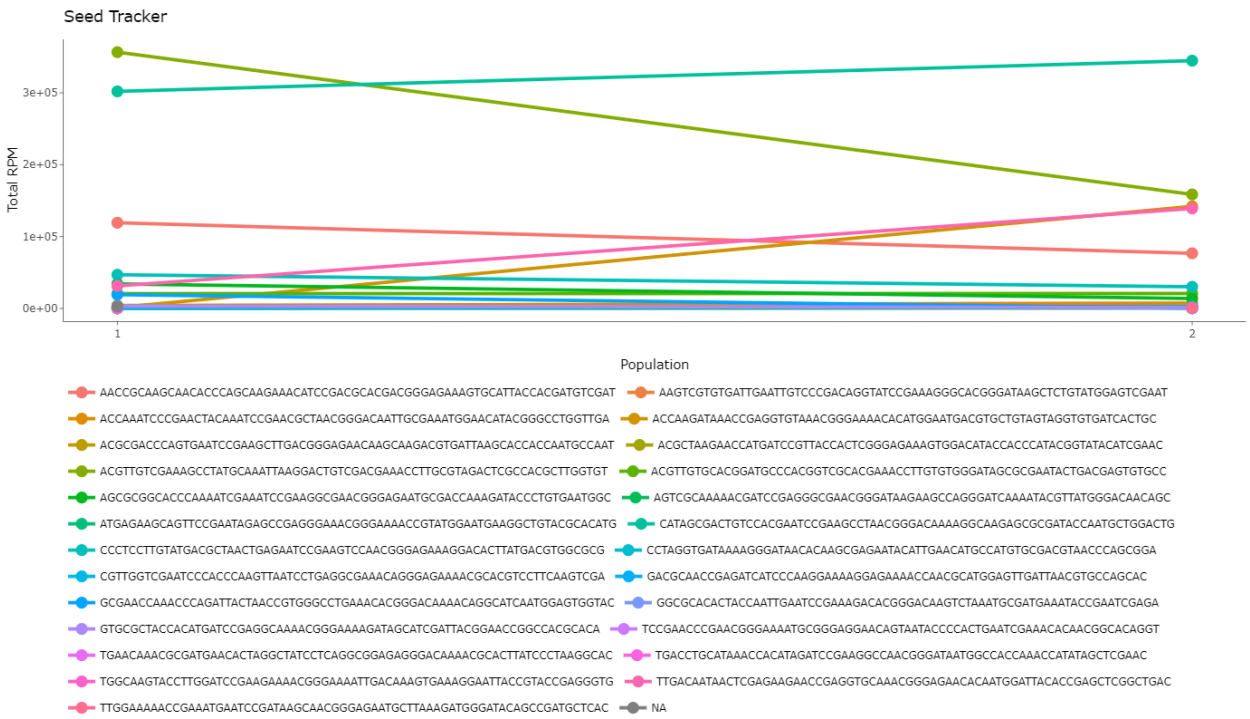


Figure 34: Cluster Tracker Line Plot.

4 Version history

References

Alam KK, Burke DH, Chang JL. 2015. "FASTAptamer: A Bioinformatic Toolkit for High-throughput Sequence Analysis of Combinatorial Selections." *Mol Ther Nucleic Acids* 4. https://doi.org/10.1038/mtna.2015.4.

Burke DH, Andrews K, Scates L. 1996. "Bent pseudoknots and novel RNA inhibitors of type 1 human immunodeficiency virus (HIV-1) reverse transcriptase." *J Mol Biol* 264. https://doi.org/10.1006/jmbi.1996.0667.

Ditzler MA, Bose D, Lange MJ. 2013. "High-throughput sequence analysis reveals structural diversity and improved potency among RNA inhibitors of HIV reverse transcriptase." *Nucleic Acids Res* 41. https://doi.org/10.1093/nar/gks1190.

Whatley AS, Lange MJ, Ditzler MA. 2013. "Potent Inhibition of HIV-1 Reverse Transcriptase and Replication by Nonpseudoknot, 'UCAA-motif' RNA Aptamers." *Mol Ther Nucleic Acids* 2. https://doi.org/10.1038/mtna.2012.62.