Molecular Therapy
Nucleic Acids
Original Article

FASTAptameR 2.0: A web tool for combinatorial sequence selections

Skyler T. Kramer,[1,2] Paige R. Gruenke,[2,3] Khalid K. Alam,[4] Dong Xu,[1,2,5] and Donald H. Burke[2,3,6]

[1]MU Institute for Data Science and Informatics, University of Missouri, Columbia, MO, USA; [2]Bond Life Sciences Center, University of Missouri, Columbia, MO, USA; [3]Department of Biochemistry, University of Missouri School of Medicine, Columbia, MO, USA; [4]Stemloop, Inc., Evanston, IL 60201, USA; [5]Department of Electrical Engineering and Computer Science, University of Missouri, Columbia, MO, USA; [6]Department of Molecular Microbiology and Immunology, University of Missouri School of Medicine, Columbia, MO, USA

Combinatorial selections are powerful strategies for identifying biopolymers with specific biological, biomedical, or chemical characteristics. Unfortunately, most available software tools for high-throughput sequencing analysis have high entrance barriers for many users because they require extensive programming expertise. FASTAptameR 2.0 is an R-based reimplementation of FASTAptamer designed to minimize this barrier while maintaining the ability to answer complex sequence-level and population-level questions. This open-source toolkit features a user-friendly web tool, interactive graphics, up to 100 times faster clustering, an expanded module set, and an extensive user guide. FASTAptameR 2.0 accepts diverse input polymer types and can be applied to any sequence-encoded selection.

INTRODUCTION

Combinatorial selections are powerful strategies for identifying biopolymers with specific characteristics such as target specificity or affinity, catalytic properties, or biological function. The strength and adaptability of this approach were recognized with the 2018 Nobel Prize in Chemistry for Francis Arnold, George Smith, and Gregory Winter.[1] While these biopolymers are generally composed of nucleotides or amino acids, the molecular alphabets can be extended or modified to include non-canonical amino acids[2] and chemically modified nucleotides such as AEGIS,[3] Hachimoji,[4] and others.[5] Selection strategies for nucleic acids have been applied to aptamers,[6,7] (deoxy)ribozymes,[8–10] synthetic genetic polymers (XNAs),[11,12] and other combinatorial chemistries. Selection strategies for peptides and proteins can be accomplished by selecting for bioactivity in cells or whole organisms[13] or by displaying on phage particles,[14] ribosomes,[15] mRNA,[16] whole bacteria,[17] and other platforms. The genes that encode the evolving proteins can be translated from nucleic acid libraries according to the standard genetic code or to natural or artificial genetic codes.[18] DNA sequence libraries have even been used as barcodes to track lipid nanoparticle formulations[19–21] and combinatorial chemical synthesis.[22,23] In short, any platform that links polymer sequence (genotype) with a selectable or screenable property (phenotype) can be adapted to combinatorial selections.

Under optimal circumstances, the evolutionary dynamics of populations undergoing selection reflect the relative fitness of each species, with high-fitness sequences typically enriching during selection and low-fitness sequences depleting. Thus, common analytic tasks of any combinatorial selection include counting the number of occurrences for each sequence,[24,25] calculating enrichment of sequences between two or more rounds,[25–27] filtering sequences based on the number of reads present in one or multiple rounds,[28] clustering related sequences,[25,29–31] and in some cases analyzing predicted structure motifs.[29,32–36] High-throughput sequencing (HTS) provides large volumes of data for these analyses and can yield high-resolution insights. Many specialized bioinformatics toolkits have been developed to enable this analytical workflow,[37] and several of these tools include graphical user interfaces to visualize HTS data during the analysis.[36,38,39] However, some of these toolkits require significant computational resources or coding expertise that together constitute barriers to entry for the average molecular biologist. Our lab previously developed and released the FASTAptamer toolkit[25] to address the primary, sequence-level needs in the field, such as those outlined above. FASTAptamer is an open-source toolkit consisting of five Perl scripts that can be used to count, normalize, and rank reads in a FASTQ or FASTA file, compare populations for sequence distribution, cluster related sequences, calculate fold-enrichment, and search for sequence motifs.[25] Since its publication, the FASTAptamer toolkit has been used and cited extensively for diverse types of molecular and biological selections on populations of functional nucleic acids and protein/peptides (see Supplementary Information), thereby demonstrating its ability to address many of the first-level bioinformatics needs of the field.

Although FASTAptamer can analyze sequences from many types of biomolecules, its original application was targeted at aptamers, which are structured nucleic acids capable of binding to a molecular target, usually with high specificity and affinity. Aptamers are generated through an iterative, *in vitro* selection process termed SELEX (Systematic Evolution of Ligands by EXponential enrichment).[6,7] After a determined number of selection rounds, the sequences of the enriched

Received 4 May 2022; accepted 18 August 2022;
https://doi.org/10.1016/j.omtn.2022.08.030.

Correspondence: Donald H. Burke, PhD, Bond Life Sciences Center, University of Missouri, Columbia, MO, USA.
E-mail: burkedh@missouri.edu

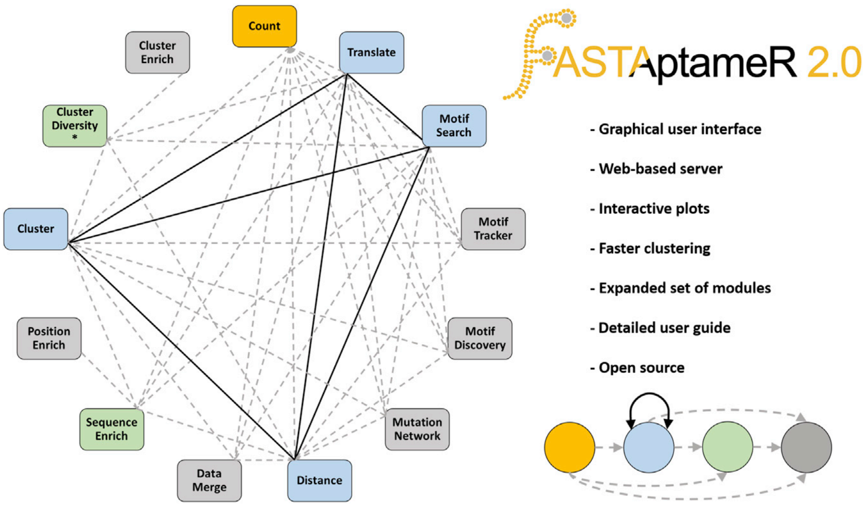


Figure 1. General overview, module connectivity, and major new features of FASTAptameR 2.0
The gold node (the *Count* module) is the required first step of every user-customized workflow. Blue nodes (such as the *Distance* module) are either intermediate or final steps of workflows. Gray nodes (such as the *Motif Tracker* module) are final steps of workflows, and green nodes (such as the *Sequence Enrich* module) exclusively feed into gray nodes. Solid black edges are bidirectional, whereas dashed gray edges are unidirectional. The asterisk with the *Cluster Diversity* module is to indicate that the population must be clustered at some step before its use. Most module outputs are downloadable as FASTA, CSV, or both.

aptamers have traditionally been obtained by cloning the aptamer libraries into a plasmid and sequencing each clone one at a time. With HTS, millions of sequence reads from multiple rounds of selection can be determined, and this information can be used to identify aptamer candidates for further characterization.[40–47] HTS investigation of *in vitro* selection pools has revealed the distribution and relative frequencies of individual sequences and groups of related sequences as the populations evolve through the course of the experiment.[48,49] Such data can inform on the success of the selection,[45,50] aptamer-target interactions,[51–54] the mutation and fitness landscape,[29,44,55] structure-function relations, biological constraints, and more.

While the initial release of FASTAptamer is generally user-friendly, it also has some limitations. First, as the FASTAptamer modules are Perl scripts, they must be run using a command line, which creates a modest barrier for practitioners of combinatorial selections who are unfamiliar or uncomfortable with a command line interface. Second, depending on the parameters used, the clustering module is time-consuming and computationally intensive.[31] Third, while the output data from FASTAptamer can be downloaded for offline visualization, it does not allow for visualization of results within the platform, which can constrain data exploration.

To address these limitations, we describe here the development of FASTAptameR 2.0, an R-based reimplementation of FASTAptamer. This program improves upon the original version while keeping the features that made FASTAptamer an accessible, easy-to-use toolkit for the analysis of HTS datasets. Like FASTAptamer, FASTAptameR 2.0 does not need external dependencies (especially when used through the web tool) and is easy to install and launch. FASTAptameR 2.0 is portable across multiple platforms, open source, and comes with a detailed user guide that includes screenshots of the user interface and sample output tables and graphs for each module (see Supplementary Information). Further, the generalizable outputs can be used as downstream inputs to this program or any other bioinformatics program that supports FASTA files. It has a user-friendly interface that can be accessed online at https://fastaptamer2.missouri.edu/ or in a downloadable form as a Docker image from Docker Hub (https://hub.docker.com/repository/docker/skylerkramer/fastaptamer2), and the code can be accessed from GitHub at https://github.com/SkylerKramer/FASTAptameR-2.0.[56] Additional improvements in FASTAptameR 2.0 include a faster clustering algorithm with speeds nearly 100X faster than FASTAptamer in some cases (e.g., for larger, more complex libraries) and an expanded set of interconnected modules (shown in Figure 1) that can be used to interactively analyze and visualize HTS data from new perspectives with custom, user-defined pipelines. Collectively, these improvements make exploration of HTS data from combinatorial selections significantly more accessible.

RESULTS

Count module

The first step in analyzing sequence data from combinatorial selections is nearly always to determine the read count (abundance) of each unique sequence. This information can indicate whether the population is relatively diverse with little convergence or has converged on one or a few dominant sequences. Either of these scenarios is immediately evident when analyzing the population with the *Count* module, which, as in FASTAptamer, is the entry point into FASTAptameR 2.0.

This module serves two main purposes. First, it condenses the original file size by returning a FASTA file with a single entry for each unique sequence. Second, it provides summary statistics for each unique sequence in the input population as three key metrics: abundance, rank by abundance, and reads per million (RPM), which is the read count divided by the population size in millions. It then incorporates these statistics into the sequence identifier for each entry. For example, a sequence with an identifier of ">4-94978-43966.9" is the fourth most abundant sequence in its population, has 94,978 identical reads, and is present at a frequency of 43,966.9 RPM. The distribution of those statistical values across a given population provides the first

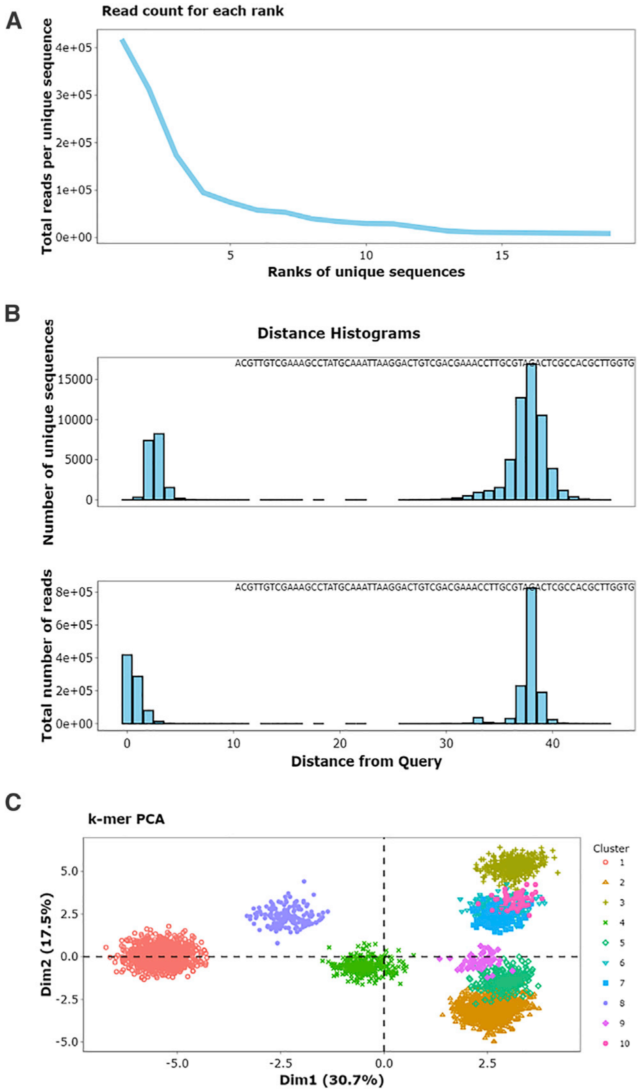


Figure 2. Example plots in a *Cluster* module workflow, using the 70HRT14 population as an example

(A) This line plot of the relationship between sequence rank and abundance (from the *Count* module) suggests that the population is dominated by a few sequences (convergence) due to its relatively steep slope and the magnitude of the y axis. (B) These histograms of LEDs suggest that many sequences in this population are similar to its most abundant sequence and that the region of sequence space surrounding the most abundant sequence is well-sampled, which can indicate biochemical significance. For the top plot, each unique sequence is equally weighted, whereas each unique sequence in the bottom plot is weighted by abundance. (C) A 3-mer matrix was generated from clustered sequences and visualized as a PCA plot where colors correspond to clusters.

insights into the degree to which that population has converged, which can be seen by visualizing the relationship between rank and abundance (Figure 2A, generated with the 70HRT14 population from the original FASTAptamer publication[51,57–59]; see Materials and Methods). A slowly decreasing function suggests that the population has not converged onto a small set of sequences, whereas a steeply decreasing line suggests convergence.

A new feature of FASTAptameR 2.0 is the ability to identify overlaps among two or more populations. To this end, multiple populations from the *Count* module or from several other, downstream modules can be merged and visualized together in the *Data Merge* module. Supported merge types return the set of all sequences from every population (union), the set of all sequences that are shared between every population (intersection), or the set of all sequences from the first population with information from the other population(s) appended to it (left join).

Distance module

The *Distance* module is a new feature of FASTAptameR 2.0 that computes the Levenshtein edit distance (LED) between a query sequence and every other sequence in the population. An in-house precursor of this module was previously used in an *in vivo* selection.[60] Distance analysis can be especially useful when monitoring the accumulation of point mutants, evaluating the effectiveness of a mutagenesis protocol, or monitoring diversity near the beginning of a selection from sequences that densely sample local sequence space (e.g., via mutagenic PCR or doped resynthesis). The distribution of these LEDs can then be visualized as a histogram of distances (Figure 2B) to provide additional perspectives on overall sequence relatedness within the population. For output libraries, an isolated cluster will be seen close to zero distance when the population consists predominantly of sequences that are closely related to the query (such as when the query is part of a cluster of sequences that have come to dominate the population) or after the accumulation of new mutations during the course of the selection (drift or divergent evolution from the founder sequences). A second peak will appear at large distances from the query when the remaining sequences are evolutionarily unrelated to the query, such as when many different founding members of a random sequence population are independently selected (Figure 2B). To illustrate, this analysis was applied to the 70HRT14 population, using the most abundant sequence as the query. Plotting this distribution *by equally weighting each unique sequence* (top plot of Figure 2B) reveals that many of these sequences are similar to the query, that the region of sequence space immediately surrounding this query is well-sampled (which can indicate its biochemical significance), that most species are within three mutations relative to the query, and that nearly all sequences related to the query are within an LED of 7. This visualization provides guidance in setting the maximum LED value to use in the *Cluster* module (see below). In contrast, plotting the distribution after *weighting the data by sequence abundance* (bottom plot of Figure 2B) shows that variants within one or two mutations of the query are far more abundant than those with higher-order mutations.

Mutation network module

Fitness landscapes and evolutionary histories can sometimes be revealed by looking at mutational intermediates and how they rise and fall during selection. The *Mutation Network* module is a new feature of FASTAptameR 2.0 that uses Dijkstra's shortest path algorithm to discover the shortest evolutionary path between two query sequences in a population. The maximum number of mutations per evolutionary step can be defined by the user, thereby allowing for highly constrained, incremental steps (e.g., no more than one mutation per step) or for larger, more saltatory steps. If all intermediates for a given path are present, the module then returns a data table for the intermediates along that path. This functionality allows researchers to better understand evolutionary trajectories in the fitness landscape created in the experiment.

Cluster algorithm and validation

The *Cluster* module groups closely related sequences into "clusters," thereby setting the stage for computing local fitness landscapes and further simplifying downstream analysis. The clustering algorithms for FASTAptamer and FASTAptameR 2.0 are both iterative, greedy processes that start by considering the most abundant, unclustered sequence as a cluster seed during each iteration. Given that the *Count* module sorts the population by abundance, the first sequence in that output becomes the cluster seed for the first iteration. All unique, unclustered sequences within a predetermined LED are added to this cluster. After considering all unique sequences in the population, the most abundant sequence that remains unclustered becomes the seed of the second cluster, and all unclustered sequences within the LED are added to that cluster. These steps are iterated until a predetermined stop condition is met (see below).

Clustering can be computationally intense and slow, a problem that has been observed for FASTAptamer and other platforms.[31] FASTAptameR 2.0 significantly reduces the clustering runtime by changing the underlying data structure for the computations. The original implementation stores clustered sequences in arrays (a static data structure), whereas the FASTAptameR 2.0 implementation uses linked lists (a dynamic data structure). The list structure more efficiently handles memory requirements, which grow with the size and complexity of the population. FASTAptameR 2.0 offers additional means of reducing clustering time, such as by allowing the user to filter out sequences with abundance less than a user-defined threshold or to set a maximum number of clusters for the module to generate.

The FASTAptameR 2.0 clustering algorithm was benchmarked against FASTAptamer by comparing runtimes on an Ubuntu subsystem (v18.04) on a desktop computer with 16 GB RAM and an Intel I7 processor. All 72,921 unique sequences from the 70HRT14 population were used to generate the top 30 clusters with a maximum LED of seven. While the original implementation finished in 35 min 27 s, the FASTAptameR 2.0 implementation finished in 24.6 s, roughly 86 times faster. When clustering times were compared for a number of other scenarios, FASTAptameR 2.0 was always significantly faster than FASTAptamer, and the magnitude of this difference grew with the size and complexity of the population being analyzed. Therefore, the FASTAptameR 2.0 clustering algorithm is strictly better than the algorithm used in the original FASTAptamer.

Outputs from the *Cluster* module can be visualized with the *Cluster Diversity* module, which is another new addition to FASTAptameR 2.0. This module uses all unique sequences within each of the user-defined clusters to create a *k*-mer matrix. This matrix is subsequently visualized as a two-dimensional PCA plot (Figure 2C). The *k*-mer plot for the top 10 clusters of the 70HRT14 population shows most clusters in well-defined regions with separation from most or all of the other clusters. The separations among the clusters reflect their origins from independent, unrelated founder sequences present in the initial population, while the spread of each cluster reflects the sampling of local sequence space around each founder sequence, resulting from the accumulation of functional point mutations through neutral drift and purifying selection.

Sample plots from a cluster-specific workflow are shown in Figure 2. The panels show evidence of a convergent population (Figure 2A), quantify the distance from a query sequence to the rest of the population and suggest LED values required for clustering (Figure 2B), and provide a visualization of the separation between and diversity within clusters (Figure 2C).

Motif discovery

Combinatorial selections often converge upon one or more sequence or structural motifs that are present in many otherwise unrelated members of the population. The new *Motif Discovery* module is included in FASTAptameR 2.0 to provide a preliminary assessment of shared sequence motifs. This module uses an implementation of the Fast String-Based Clustering (FSBC) algorithm[31] for *de novo* discovery for contiguous, over-enriched motifs in D/RNA sequences. There are many other excellent tools dedicated to *de novo* motif discovery, such as the MEME suite[61] for sequence-based approaches and Infernal[62] for RNA using both sequence-based and predicted structural similarities. FASTA-formatted output from any of the FASTAptameR 2.0 modules can be exported and analyzed by those other dedicated platforms.

Individual- and population-level tracking

Another new feature of FASTAptameR 2.0 is the ability to track individual motifs, sequences, or clusters across multiple rounds of a selection experiment. The *Motif Tracker* module tracks query motifs or sequences, and the *Enrich* module tracks how every sequence changes between populations, while the *Cluster Enrich* module allow the user to monitor the collective behavior of the cluster as a whole, analogous to the collective evolutionary dynamics of a viral quasispecies. These three modules additionally calculate how families or species enrich, which can indicate how they performed in the selection experiment. An in-house precursor of *Motif Tracker* was previously used in a selection for 2′-modified RNA aptamers with affinity for HIV-1 RT.[63] In the case when two clustered files are supplied to the *Enrich* module, the enrichment values of individual sequences can be grouped by clustering and visualized as a boxplot (Figure 3A). In the example shown in Figure 3A, Cluster 2 is enriched relative to the other clusters, suggesting that this set of sequences has a motif important for target binding (specifically

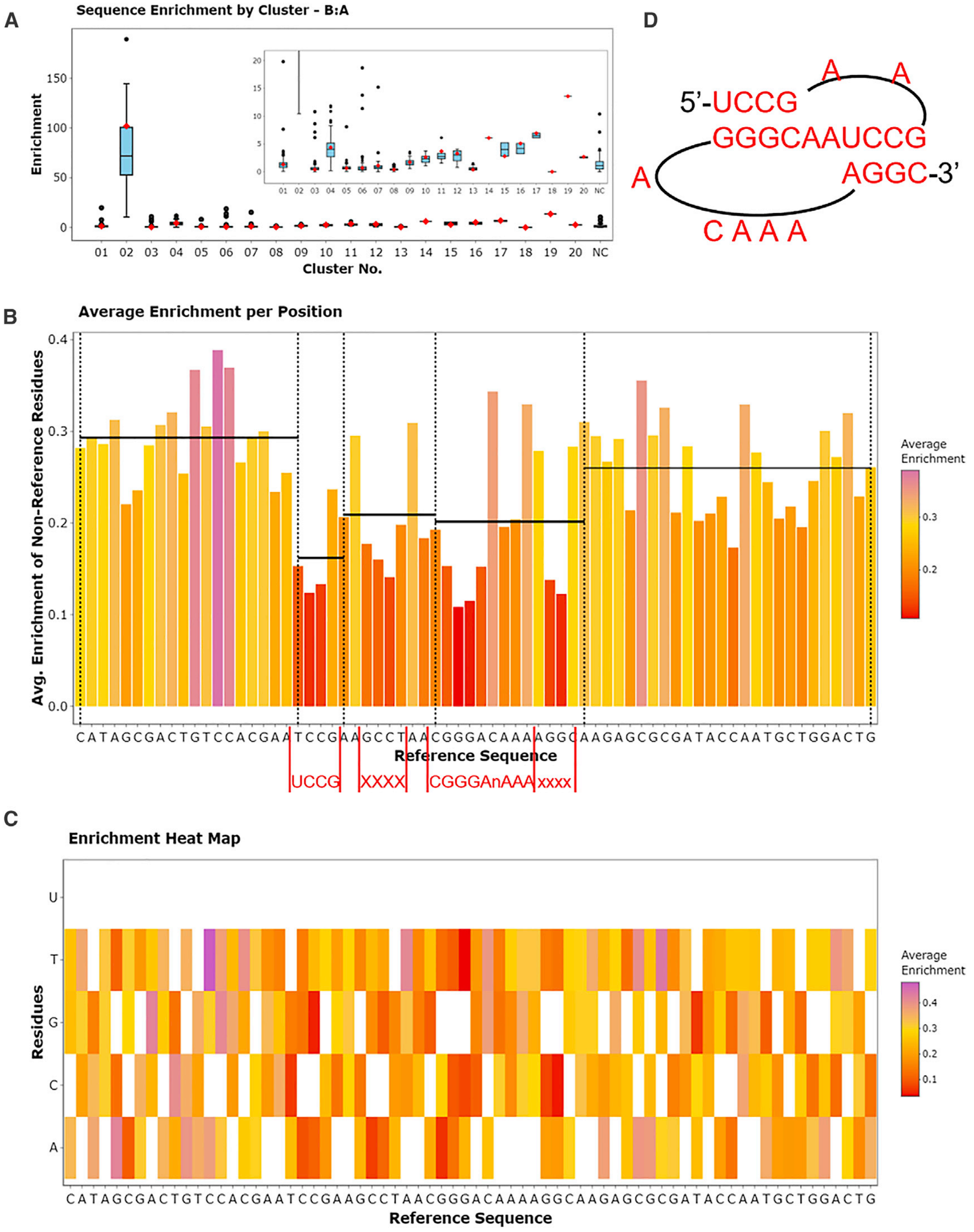


Figure 3. Cluster and position enrichment plots

(A) The cluster boxplot showing how clustered sequences in 70HRT14 enrich in 70HRT15. Cluster 2 of 70HRT14, for example, is highly enriched in 70HRT15 due to the presence of the F1Pk, which is implicated in target binding to HIV-1 RT. The 25th and 75th quartiles are respectively represented by the bottom and top of each box. The line in the middle of the box represents the median. Whiskers are at most 1.5 * IQR (interquartile range), and any points beyond that are shown as outliers. The red marker indicates where the seed sequence of the cluster falls. (B) The x axis of the bar plot also shows the user-defined reference sequence, and the y axis shows the average enrichment of each non-reference residue at each position. The red text below this panel shows the portion of the query sequence that matches the linear F1Pk motif. Black

(legend continued on next page)

the family 1 pseudoknot, or F1Pk, in this case). For each cluster in the boxplot, individual points that are well above or well below the median value represent species that are enriching or depleting relative to the cluster as a whole. Both the enriched and depleted species can be highly informative, as species that carry strongly advantageous variations may be emerging as future dominant species for that cluster, while species with strongly disadvantageous variations can illuminate critical portions of the biomolecule, as illustrated by the *Position Enrich* module.

Position enrichment

For a set of closely related sequences, mutations in some positions contribute directly to enrichment or depletion, while mutations at some other positions have little consequence. Uncovering these relationships can be enormously valuable for delineating the contributions of those positions to macromolecular functions. The *Position Enrich* module is a new feature of FASTAptameR 2.0 that calculates the average enrichment or depletion at each position for all sequences that do not match the corresponding user-defined reference residue at that position. This calculation is visualized as a bar plot (Figure 3B). Relatively short bars indicate functional conservation at that position, such that deviations from the reference residue identity contribute to depletion. Exceptionally tall bars may indicate positive selection for improved function relative to the reference sequence. As a result, highly conserved sections are immediately visible as regions with low bars because mutations in these positions contribute to depletion. The module calculates local averages across user-defined intervals and displays them as horizontal lines across those intervals, making the conserved and non-conserved regions especially evident from visual inspection (Figure 3B). *Position Enrich* further resolves enrichment and depletion patterns for each of the available substitute residues (e.g., three alternative nucleotides or 19 alternative amino acids when using standard alphabets) and displays the resulting patterns as a heatmap (Figure 3C). As in the *Translate* module, nonstandard nucleotides and amino acids can be analyzed with the *Position Enrichment* module.

To generate the plots in Figure 3, the 70HRT14 and 70HRT15 populations were counted and clustered following the workflow of Figure 2, although in this case the *Count* module was also used to omit any sequences that were not exactly 70 nucleotides long. The *Enrich* module created the boxplot from the full set of clustered sequences in both populations. The *Enrich* module then calculated enrichment scores for the first cluster from 70HRT15, which carries the F1Pk motif, and for the corresponding cluster from the preceding round of selection (second cluster from 70HRT14). The *Position Enrich* module used the output from the *Sequence Enrich* module as input, and the most abundant sequence in 70HRT15 was used as the reference sequence. The segments within the 70-nucleotide random region that contain the pseudoknot[64] at the functional core of the aptamer are shown in Figure 3D.

Expanded sequence support

While populations of any sequence type (e.g., nucleotides or amino acids) could be fed into the original release of FASTAptamer, it did not allow for sequence translations. The *Translate* module of FASTAptameR 2.0 translates nucleotide sequences to amino acids according to either the standard genetic code or any of 15 alternative genetic codes such as those used by vertebrate mitochondria, mycoplasma, and other organisms. This module also supports complete customization of the genetic code used for translation to support nonstandard nucleotide input and/or nonstandard amino acid output, both of which are useful for applications in synthetic biology. Thus, FASTAptameR 2.0 explicitly supports all linear biopolymers of diverse biological origins.

DISCUSSION

The integration of HTS with combinatorial selection experiments has created many opportunities and challenges for bioinformatics analyses. Though many tools exist to aid these analyses,[24,25,28,30,31,34–36,38,39] usually they are not designed for users without a relatively strong computational background. As such, a typical practitioner of combinatorial selections may need to devote significant time and effort to tasks such as software installation and dependency handling before they can even learn to properly use the tool, constituting a serious barrier to data exploration. A notable exception is the REVERSE platform,[65] which offers a user-friendly web service to analyze populations of RNA sequences from selection and evolution experiments. This tool is easy to use, supports preprocessing functionality, and offers helpful documentation, although it lacks the abilities to handle expanded/customized alphabets or to fully customize user workflows.

FASTAptameR 2.0 was designed with non-computational users in mind and according to best practices in the field of bioinformatics.[66–69] Like its predecessor FASTAptamer, FASTAptameR 2.0 is a powerful open-source toolkit to analyze combinatorial selection populations and is accompanied by an extensive user guide. The program simplifies data analysis by minimally requiring a web browser and internet access. For the web-based version, the UI can be accessed from any browser operating with any operating system. Alternatively, the user may choose to download the software and run it locally, which is compatible on any system with a functional Docker installation and does not require internet access. Further, the outputs are designed to be modular so that this platform can be easily integrated into existing workflows or used to develop custom ones. Module inputs and outputs are standard file types (e.g., FASTQ/A and CSV).

horizontal lines show the left-inclusive average enrichment score of each user-defined region. The regions corresponding to the F1Pk motif have the lowest regional average of enrichment scores, indicating the importance of this motif for this selection experiment. (C) The x axis of the heatmap shows the user-defined reference sequence, and the y axis shows all possible residues at each position. Colors depict the average enrichment of each possible non-reference residue. (D) The experimentally determined secondary structure of the F1Pk motif.

Modules in this platform can be used for a wide range of functions on subsets of individual populations or across many populations. *FASTAptameR-Count*, the starting point of the platform, counts and ranks unique sequences. *FASTAptameR-Translate* translates nucleotide sequences according to standard, nonstandard, or user-defined genetic codes. The trio of modules *FASTAptameR-Motif_Search*, *FASTAptameR-Motif_Tracker*, and *FASTAptameR-Motif_Discovery* serve the three functions of identifying occurrences of motifs, tracking motifs or sequences across multiple populations, and identifying over-enriched motifs, respectively. *FASTAptameR-Distance* computes the LED between all sequences and a query sequence. *FASTAptameR-Mutation_Network* identifies the shortest mutational path between two sequences in a population. *FASTAptameR-Data_Merge* merges sequences from multiple populations. *FASTAptameR-Sequence_Enrich* and *FASTAptameR-Position_Enrich* assess sequence trajectories across populations and provide insights into which residues contribute to the enrichment scores. The three linked modules of *FASTAptameR-Cluster*, *FASTAptameR-Cluster_Diversity*, and *FASTAptameR-Cluster_Enrich* cluster sequences, provide cluster-level metadata, and assess how they change across populations.

FASTAptameR 2.0 features substantial improvements relative to its predecessor. By increasing user accessibility, improving the original modules, and providing additional tools for data analysis, FASTAptameR 2.0 further lowers the technical barrier for analysis and exploration of HTS datasets and allows the user to gain more insights from their combinatorial selection experiments.

MATERIALS AND METHODS

Data description

Data from the original FASTAptamer publication[57] were used to build and test FASTAptameR 2.0. In brief, these data are two populations of RNA aptamers selected to target HIV-1 reverse transcriptase after 14 and 15 rounds of a SELEX experiment (designated 70HRT14 and 70HRT15, respectively).[51,58] These populations were trimmed via cutadapt[70] and filtered for high-quality reads via the FASTX-Toolkit (http://hannonlab.cshl.edu/fastx_toolkit/). These FASTQ files are available at SkylerKramer/AptamerLibrary: Data for FASTAptameR 2.0 (Zenodo).[59]

Implementation

FASTAptameR 2.0 is written in the R programming language[71] and made interactive with the Shiny package.[72] The platform uses ggplot2[73] to build plots and plotly[74] to make them interactive. The entire program (i.e., code, dependencies, and supporting files) is wrapped into a Docker image and deployed on a web server at the University of Missouri - Columbia. The web server has been tested in Google Chrome, FireFox, and Safari. Beta testers at five institutions confirmed platform independence and the absence of external dependencies.

DATA AVAILABILITY STATEMENT

All code and supporting files are available at https://github.com/SkylerKramer/FASTAptameR-2.0.[56] The Docker image is available at https://hub.docker.com/repository/docker/skylerkramer/fastaptamer2. Finally, the web-accessible version of FASTAptameR 2.0 is available at https://fastaptamer2.missouri.edu/. All data analyzed in this manuscript are available at https://github.com/SkylerKramer/AptamerLibrary.[59] FASTAptameR 2.0 is distributed under a GNU General Public License version 3.0.

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.omtn.2022.08.030.

ACKNOWLEDGMENTS

This research was supported by NASA Exobiology grant NNX17AE88G to D.H.B., US National Institutes of Health BD2K Training Grant T32HG009060 grant to S.T.K., US National Institute of Health R35-GM126985 grant to D.X., and University of Missouri Life Sciences Fellowship and Wayne L. Ryan Graduate Fellowship from the Ryan Foundation to P.R.G.

Feedback from beta testing was provided by Austin Prater, Marc Johnson, Carolyn Robinson, and Laurie Agosto from the University of Missouri - Columbia; Uli Müller from the University of California - San Diego; Andrej Lupták from the University of California - Irvine; and Elisa Biondi from the Foundation for Applied Molecular Evolution. The authors thank Rebecca Burke-Agüero for critical discussions and technical assistance early in the development of a faster clustering algorithm and Matt Stanley for network support during launch. The authors also thank members of the Burke and Xu research teams and the NASA Interdisciplinary Consortium for Astrobiology Research “Bringing RNA to Life” (NASA grant 80NSSC21K0596) for feedback, discussion, and advice throughout the project.

AUTHOR CONTRIBUTIONS

S.T.K. developed the front end and back end, prepared the tool for deployment, interacted with beta testers, and wrote the manuscript and user guide with input from P.R.G., K.K.A., D.X., and D.H.B. D.H.B. supervised the project, recruited and interacted with beta testers, conceived the project with P.R.G., and edited the manuscript. The authors read and approved the final manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

1. Gibney, E., Van Noorden, R., Ledford, H., Castelvecchi, D., and Warren, M. (2018). ’Test-tube’ evolution wins chemistry nobel prize. Nature *562*, 176.
2. Strack, R. (2020). Noncanonical amino acids on display. Nat. Methods *17*, 461.
3. Yang, Z., Chen, F., Chamberlin, S.G., and Benner, S.A. (2010). Expanded genetic alphabets in the polymerase chain reaction. Angew. Chem. Int. Ed. Engl. *49*, 177–180.
4. Hoshika, S., Leal, N.A., Kim, M.-J., Kim, M.-S., Karalkar, N.B., Kim, H.-J., Bates, A.M., Watkins, N.E., SantaLucia, H.A., Meyer, A.J., et al. (2019). Hachimoji DNA and RNA: a genetic system with eight building blocks. Science *363*, 884–887.
5. Hwang, G.T., and Romesberg, F.E. (2008). Unnatural substrate repertoire of a, b, and x family DNA polymerases. J. Am. Chem. Soc. *130*, 14872–14882.

6. Tuerk, C., and Gold, L. (1990). Systematic evolution of ligands by exponential enrichment: RNA ligands to bacteriophage T4 DNA polymerase. Science *249*, 505–510.

7. Ellington, A.D., and Szostak, J.W. (1990). In vitro selection of RNA molecules that bind specific ligands. Nature *346*, 818–822.

8. Pitt, J.N., and Ferré-D'Amaré, A.R. (2010). Rapid construction of empirical RNA fitness landscapes. Science *330*, 376–379.

9. Pressman, A.D., Liu, Z., Janzen, E., Blanco, C., Müller, U.F., Joyce, G.F., Pascal, R., and Chen, I.A. (2019). Mapping a systematic ribozyme fitness landscape reveals a frustrated evolutionary network for self-aminoacylating RNA. J. Am. Chem. Soc. *141*, 6213–6223.

10. Yokobayashi, Y. (2019). Applications of high-throughput sequencing to analyze and engineer ribozymes. Methods *161*, 41–45.

11. Burmeister, P.E., Lewis, S.D., Silva, R.F., Preiss, J.R., Horwitz, L.R., Pendergrast, P.S., et al. (2005). Direct in vitro selection of a 2'-O-methyl aptamer to VEGF. Chem. Biol. *12*, 25–33.

12. Taylor, A.I., and Holliger, P. (2015). Directed evolution of artificial enzymes (XNAzymes) from diverse repertoires of synthetic genetic polymers. Nat. Protoc. *10*, 1625–1642.

13. Szardenings, M., Törnroth, S., Mutulis, F., Muceniece, R., Keinänen, K., Kuusinen, A., and Wikberg, J.E. (1997). Phage display selection on whole cells yields a peptide specific for melanocortin receptor 1. J. Biol. Chem. *272*, 27943–27948.

14. Dias-Neto, E., Nunes, D.N., Giordano, R.J., Sun, J., Botz, G.H., Yang, K., Setubal, J.C., Pasqualini, R., and Arap, W. (2009). Next-generation phage display: integrating and comparing available molecular tools to enable cost-effective high-throughput analysis. PLoS One *4*, e8338.

15. Villemagne, D., Jackson, R., and Douthwaite, J.A. (2006). Highly efficient ribosome display selection by use of purified components for in vitro translation. J. Immunol. Methods *313*, 140–148.

16. Cotten, S.W., Zou, J., Wang, R., Huang, B., and Liu, R. (2011). mRNA display-based selections using synthetic peptide and natural protein libraries. In Ribosome Display and Related Technologies (New York: Springer), pp. 287–297.

17. Granhøj, J., Dimke, H., and Svenningsen, P. (2019). A bacterial display system for effective selection of protein-biotin ligase BirA variants with novel peptide specificity. Sci. Rep. *9*, 4118.

18. Xie, J., and Schultz, P.G. (2005). Adding amino acids to the genetic repertoire. Curr. Opin. Chem. Biol. *9*, 548–554.

19. Dahlman, J.E., Kauffman, K.J., Xing, Y., Shaw, T.E., Mir, F.F., Dlott, C.C., Langer, R., Anderson, D.G., and Wang, E.T. (2017). Barcoded nanoparticles for high throughput in vivo discovery of targeted therapeutics. Proc. Natl. Acad. Sci. USA *114*, 2060–2065.

20. Sago, C.D., Lokugamage, M.P., Paunovska, K., Vanover, D.A., Monaco, C.M., Shah, N.N., Gamboa Castro, M., Anderson, S.E., Rudoltz, T.G., Lando, G.N., et al. (2018). High-throughput in vivo screen of functional mRNA delivery identifies nanoparticles for endothelial cell gene editing. Proc. Natl. Acad. Sci. USA *115*, E9944–E9952.

21. Paunovska, K., Sago, C.D., Monaco, C.M., Hudson, W.H., Castro, M.G., Rudoltz, T.G., Kalathoor, S., Vanover, D.A., Santangelo, P.J., Ahmed, R., et al. (2018). A direct comparison of in vitro and in vivo nucleic acid delivery mediated by hundreds of nanoparticles reveals a weak correlation. Nano Lett. *18*, 2148–2157.

22. Brenner, S., and Lerner, R.A. (1992). Encoded combinatorial chemistry. Proc. Natl. Acad. Sci. USA *89*, 5381–5383.

23. Favalli, N., Bassi, G., Scheuermann, J., and Neri, D. (2018). DNA-encoded chemical libraries - achievements and remaining challenges. FEBS Lett. *592*, 2168–2180.

24. Thiel, W.H., and Giangrande, P.H. (2016). Analyzing HT-SELEX data with the galaxy project tools a web based bioinformatics platform for biomedical research. Methods *97*, 3–10.

25. Alam, K.K., Chang, J.L., and Burke, D.H. (2015). FASTAptamer: a bioinformatic toolkit for high-throughput sequence analysis of combinatorial selections. Mol. Ther. Nucleic Acids *4*, e230.

26. Cho, M., Xiao, Y., Nie, J., Stewart, R., Csordas, A.T., Oh, S.S., Thomson, J.A., and Soh, H.T. (2010). Quantitative selection of DNA aptamers through microfluidic selection and high-throughput sequencing. Proc. Natl. Acad. Sci. USA *107*, 15373–15378.

27. Schütze, T., Wilhelm, B., Greiner, N., Braun, H., Peter, F., Mörl, M., Erdmann, V.A., Lehrach, H., Konthur, Z., Menger, M., et al. (2011). Probing the SELEX process with next-generation sequencing. PLoS One *6*, e29604.

28. Thiel, W.H. (2016). Galaxy workflows for web-based bioinformatics analysis of aptamer high-throughput sequencing data. Mol. Ther. Nucleic Acids *5*, e345.

29. Nguyen Quang, N., Bouvier, C., Henriques, A., Lelandais, B., and Ducongé, F. (2018). Time-lapse imaging of molecular evolution by high-throughput sequencing. Nucleic Acids Res. *46*, 7480–7494.

30. Hoinka, J., Berezhnoy, A., Sauna, Z.E., Gilboa, E., and Przytycka, T.M. (2014). AptaCluster a Method to Cluster HT-SELEX Aptamer Pools and Lessons from its Application (Springer International Publishing), pp. 115–128.

31. Kato, S., Ono, T., Minagawa, H., Horii, K., Shiratori, I., Waga, I., Ito, K., and Aoki, T. (2020). FSBC: Fast string-based clustering for HT-SELEX data. BMC Bioinf. *21*, 263.

32. Hoinka, J., Zotenko, E., Friedman, A., Sauna, Z.E., and Przytycka, T.M. (2012). Identification of sequence-structure RNA binding motifs for SELEX-derived aptamers. Bioinformatics *28*, i215–i223.

33. Hoinka, J., Berezhnoy, A., Dao, P., Sauna, Z.E., Gilboa, E., and Przytycka, T.M. (2015). Large scale analysis of the mutational landscape in HT-SELEX improves aptamer discovery. Nucleic Acids Res. *43*, 5699–5707.

34. Dao, P., Hoinka, J., Takahashi, M., Zhou, J., Ho, M., Wang, Y., Costa, F., Rossi, J.J., Backofen, R., Burnett, J., and Przytycka, T.M. (2016). AptaTRACE elucidates RNA sequence-structure motifs from selection trends in HT-SELEX experiments. Cell Syst. *3*, 62–70.

35. Caroli, J., Taccioli, C., De La Fuente, A., Serafini, P., and Bicciato, S. (2016). APTANI: a computational tool to select aptamers through sequence-structure motif analysis of HT-SELEX data. Bioinformatics *32*, 161–164. btv545.

36. Shieh, K.R., Kratschmer, C., Maier, K.E., Greally, J.M., Levy, M., and Golden, A. (2020). AptCompare: optimized de novo motif discovery of RNA aptamers via HTS-SELEX. Bioinformatics *36*, 2905–2906.

37. Nguyen Quang, N., Perret, G., and Ducongé, F. (2016). Applications of high-throughput sequencing for in vitro selection and characterization of aptamers. Pharmaceuticals *9*, 76.

38. Hoinka, J., Dao, P., and Przytycka, T.M. (2015). AptaGUI - a graphical user interface for the efficient analysis of HT-SELEX data. Mol. Ther. Nucleic Acids *4*, e257.

39. Hoinka, J., Backofen, R., and Przytycka, T.M. (2018). AptaSUITE: a full-featured bioinformatics framework for the comprehensive analysis of aptamers from HT-SELEX experiments. Mol. Ther. Nucleic Acids *11*, 515–517.

40. Gotrik, M.R., Feagin, T.A., Csordas, A.T., Nakamoto, M.A., and Soh, H.T. (2016). Advancements in aptamer discovery technologies. Acc. Chem. Res. *49*, 1903–1910.

41. Berezhnoy, A., Stewart, C.A., Mcnamara, J.O., 2nd, Thiel, W., Giangrande, P., Trinchieri, G., Gilboa, E., and Gilboa, E. (2012). Isolation and optimization of murine IL-10 receptor blocking oligonucleotide aptamers using high-throughput sequencing. Mol. Ther. *20*, 1242–1250.

42. Thiel, W.H., Bair, T., Peek, A.S., Liu, X., Dassie, J., Stockdale, K.R., Behlke, M.A., Miller, F.J., and Giangrande, P.H. (2012). Rapid identification of cell-specific, internalizing RNA aptamers with bioinformatics analyses of a cell-based aptamer selection. PLoS One *7*, e43836.

43. Valenzano, S., De Girolamo, A., DeRosa, M.C., McKeague, M., Schena, R., Catucci, L., and Pascale, M. (2016). Screening and identification of DNA aptamers to tyramine using in vitro selection and high-throughput sequencing. ACS Comb. Sci. *18*, 302–313.

44. Hamada, M. (2018). In silico approaches to RNA aptamer design. Biochimie *145*, 8–14.

45. Takahashi, M., Wu, X., Ho, M., Chomchan, P., Rossi, J.J., Burnett, J.C., and Zhou, J. (2016). High throughput sequencing analysis of RNA libraries reveals the influences of initial library and PCR methods on SELEX efficiency. Sci. Rep. *6*, 33697.

46. Blind, M., and Blank, M. (2015). Aptamer selection technology and recent advances. Mol. Ther. Nucleic Acids *4*, e223.

47. Komarova, N., Barkova, D., and Kuznetsov, A. (2020). Implementation of high-throughput sequencing (HTS) in aptamer selection technology. Int. J. Mol. Sci. *21*, 8774.

48. Jijakli, K., Khraiwesh, B., Fu, W., Luo, L., Alzahmi, A., Koussa, J., Chaiboonchoe, A., Kirmizialtin, S., Yen, L., and Salehi-Ashtiani, K. (2016). The in vitro selection world. Methods *106*, 3–13.

49. Kinghorn, A., Fraser, L., Liang, S., Shiu, S., and Tanner, J. (2017). Aptamer bioinformatics. Int. J. Mol. Sci. *18*, 2516.

50. Zimmermann, B., Gesell, T., Chen, D., Lorenz, C., and Schroeder, R. (2010). Monitoring genomic sequences during SELEX using high-throughput sequencing: neutral SELEX. PLoS One *5*, e9169.

51. Ditzler, M.A., Lange, M.J., Bose, D., Bottoms, C.A., Virkler, K.F., Sawyer, A.W., Whatley, A.S., Spollen, W., Givan, S.A., and Burke, D.H. (2013). High-throughput sequence analysis reveals structural diversity and improved potency among RNA inhibitors of HIV reverse transcriptase. Nucleic Acids Res. *41*, 1873–1884.

52. Alam, K.K., Chang, J.L., Lange, M.J., Nguyen, P.D.M., Sawyer, A.W., and Burke, D.H. (2018). Poly-target selection identifies broad-spectrum RNA aptamers. Mol. Ther. Nucleic Acids *13*, 605–619.

53. Dupont, D.M., Larsen, N., Jensen, J.K., Andreasen, P.A., and Kjems, J. (2015). Characterisation of aptamer-target interactions by branched selection and high-throughput sequencing of SELEX pools. Nucleic Acids Res. *43*, e139.

54. Spiga, F.M., Maietta, P., and Guiducci, C. (2015). More DNA-aptamers for small drugs: a capture-SELEX coupled with surface plasmon resonance and high-throughput sequencing. ACS Comb. Sci. *17*, 326–333.

55. Levay, A., Brenneman, R., Hoinka, J., Sant, D., Cardone, M., Trinchieri, G., Przytycka, T.M., and Berezhnoy, A. (2015). Identifying high-affinity aptamer ligands with defined cross-reactivity using high-throughput guided systematic evolution of ligands by exponential enrichment. Nucleic Acids Res. *43*, e82.

56. Kramer, S. (2022). SkylerKramer/FASTAptameR-2.0: FASTAptameR-2.0 (Zenodo).

57. Burke, D.H., Scates, L., Andrews, K., and Gold, L. (1996). Bent pseudoknots and novel RNA inhibitors of type 1 human immunodeficiency virus (HIV-1) reverse transcriptase. J. Mol. Biol. *264*, 650–666.

58. Whatley, A.S., Ditzler, M.A., Lange, M.J., Biondi, E., Sawyer, A.W., Chang, J.L., Franken, J.D., and Burke, D.H. (2013). Potent inhibition of HIV-1 reverse transcriptase and replication by nonpseudoknot, "UCAA-motif" RNA aptamers. Mol. Ther. Nucleic Acids *2*, e71.

59. Kramer, S. (2022). SkylerKramer/AptamerLibrary: Data for FASTAptameR 2.0 (Zenodo).

60. Salamango, D.J., Alam, K.K., Burke, D.H., and Johnson, M.C. (2016). In vivo analysis of infectivity, fusogenicity, and incorporation of a mutagenic viral glycoprotein library reveals determinants for virus incorporation. J. Virol. *90*, 6502–6514.

61. Bailey, T.L., Johnson, J., Grant, C.E., and Noble, W.S. (2015). The MEME suite. Nucleic Acids Res. *43*, W39–W49.

62. Nawrocki, E.P., and Eddy, S.R. (2013). Infernal 1.1: 100-fold faster RNA homology searches. Bioinformatics *29*, 2933–2935.

63. Gruenke, P.R., Alam, K.K., Singh, K., and Burke, D.H. (2020). 2'-fluoro-modified pyrimidines enhance affinity of RNA oligonucleotides to HIV-1 reverse transcriptase. RNA *26*, 1667–1679.

64. Tuerk, C., Macdougal, S., and Gold, L. (1992). RNA pseudoknots that inhibit human immunodeficiency virus type 1 reverse transcriptase. Proc. Natl. Acad. Sci. USA *89*, 6988–6992.

65. Weiss, Z., and DasGupta, S. (2022). REVERSE: a user-friendly web server for analyzing next-generation sequencing data from in vitro selection/evolution experiments. Preprint at bioRxiv.

66. Prlić, A., and Procter, J.B. (2012). Ten simple rules for the open development of scientific software. PLoS Comput. Biol. *8*, e1002802.

67. List, M., Ebert, P., and Albrecht, F. (2017). Ten simple rules for developing usable software in computational biology. PLoS Comput. Biol. *13*, e1005265.

68. Sandve, G.K., Nekrutenko, A., Taylor, J., and Hovig, E. (2013). Ten simple rules for reproducible computational research. PLoS Comput. Biol. *9*, e1003285.

69. Leprevost, F.d.V., Barbosa, V.C., Francisco, E.L., Perez-Riverol, Y., and Carvalho, P.C. (2014). On best practices in the development of bioinformatics software. Front. Genet. *5*, 199.

70. Martin, M. (2011). Cutadapt removes adapter sequences from high-throughput sequencing reads. EMBnet. j. *17*, 10.

71. R Core Team (2019). R: A Language and Environment for Statistical Computing (R Foundation for Statistical Computing).

72. Chang, W., Cheng, J., Allaire, J., Sievert, C., Schloerke, B., Xie, Y., Allen, J., McPherson, J., Dipert, A., and Borges, B. (2021). Shiny: Web Application Framework for R.

73. Wickham, H. (2016). ggplot2: Elegant Graphics for Data Analysis (Springer-Verlag).

74. Sievert, C. (2020). Interactive Web-Based Data Visualization with R, Plotly, and Shiny (Chapman; Hall/CRC).

OMTN, Volume *29*

Supplemental information

FASTAptameR 2.0: A web tool
for combinatorial sequence selections

Skyler T. Kramer, Paige R. Gruenke, Khalid K. Alam, Dong Xu, and Donald H. Burke

Table S1: FASTAptamer Use Cases

	Type of Selection	Description of Study	FASTAptamer Modules Used	Reference
Nucleic Acids	DNA Aptamers	Cell-SELEX to CD44E/s	Count	[1]
		Selection to A549 cells for aptamers that internalize	Count, Cluster	[2]
		Selection to multidrug-resistant carcinoma	Did not specify	[3]
		Selection to *A. muciniphila* gut bacteria	Count, Compare, Enrich	[4]
		Selection to carbapenem resistant *P. aeruginosa*	Count, Cluster, Enrich	[5]
		Identification of aptamer to human monocytes and macrophages	Count, Enrich	[6]
		Ligand-guided SELEX to TCR-CD3ε	Count, Enrich	[7]
		Ligand-guided SELEX with AEGIS components to TCR-CDε	Count, Enrich	[8]
		Selection to clear cell renal cell carcinoma RCC-MF cell line	Count, Enrich	[9]
		Cell-SELEX to T-cell marker CD8	Count, Enrich	[10]
		Selection to CTLA-4	Did not specify	[11]
		Selection to bovine spermatozoa	Count, Cluster	[12]
		Selection to human TNFα	Count, Cluster	[13]
		Selection to LAG3	Count, Cluster	[14]
		Selection to human PD-L1	Did not specify	[15]
		Selection to the murine extracellular domain of PD-1	Did not specify	[16]
		Selection to human adenovirus	Count, Enrich	[17]
		Selection to N-terminal domain of SARS-CoV-2 spike protein	Did not specify	[18]
		Selection to Newcastle avian virus for detection applications	Count, Cluster	[19]
		Selection to HMG1 of *Plasmodium falciparum*	Count, Enrich	[20]
		Selection to Galectin-1	Did not specify	[21]
		Selection to kanamycin for development of structure-switching aptamer-based biosensor	Count, Cluster, Compare, Enrich	[22]
		Selection to diverse synthetic cathinones	Did not specify	[23]
		Selection to serotonin as part of single-walled carbon nanotube (SWCNT) sensor	Did not specify	[24]
	RNA Aptamers	Selection (2′FY, 2′dA RNA) to G-protein coupled receptors (GPCRs)	Count, Enrich	[25]
		Re-selection to diverse strains of HIV-1 RT	Count, Cluster, Compare, Enrich, Search	[26]
		2′-modified RNA re-selection to HIV-1 RT	Count, Cluster, Enrich, Search	[27]
		Mannose-modified RNA selection to lectin concanavalin A	Count, Enrich	[28]
		Selections to thrombin and TGFβ1	Count, Cluster	[29]
		2′FY selection to MRP1 used to create MRP1-CD28 bispecific aptamer	Did not specify	[30]
		2′FY selection to TIM3	Count, Cluster	[31]
	(Deoxy)Ribozymes	Selection for RNA ligase activity using libraries with different-sized random region	Count, Compare, Cluster	[32]

		Tenofovir-transferase ribozyme selection	Count, Cluster, Enrich	[33]
		Endonuclease deoxyribozyme selection	Count, Cluster, Enrich	[34]
		Self-cleaving ribozyme selection in presence of a mineral surface	Count, Cluster, Compare	[35]
		Self-cleaving ribozyme selection in diverse chemical environments	Count, Cluster, Compare	[36]
	Nucleobase editing	Development of base-editing assay coupled with HTS	Count, Enrich	[37]
		Used assay above to compare system using tethered APOBEC3B (A3B) to the Cas9n/gDNA complex and MagnEdit system when an A3B-interacting protein is tethered to the Cas9n/gDNA complex and recruits APOBEC3B to targeted edit sites	Count, Enrich	[38]
	Miscellaneous	Determine small RNA loading specificity onto *Drosophila* Argonaute proteins	Count, Enrich	[39]
Peptides and Proteins	Phage Display	Selection to traumatic brain injury biomarkers	Count, Enrich	[40]
		Selection of phagebodies that was started by mutagenizing the host-range-determining-region (HRDR) of phage tail fibers	Count	[41]
	Biological activity *in vivo*	Screening of HIV-1 Vif mutant library	Count, Enrich	[42]
		Screening of MLV Env mutant library	Count, Enrich	[43]
	mRNA/cDNA Display	Proof-of-principle selection to anti-FLAG M2 antibody identified consensus FLAG epitope motif	Count	[44]

References

1. Lo CW-S, Chan CKW, Yu J, He M, Choi CHJ, Lau JYW, Wong N. Development of CD44E/s dual-targeting DNA aptamer as nanoprobe to deliver treatment in hepatocellular carcinoma. Nanotheranostics. 2022, 6:161-174.
2. Tanaka K, Okuda T, Kasahara Y, Obika S. Base-modified aptamers obtained by cell-internalization SELEX facilitate cellular uptake of an antisense oligonucleotide. Mol Ther Nucleic Acids. 2021, 23:440-449.
3. Zhang L, Zhou L, Zhang H, Zhang Y, Li L, Xie T, Chen Y, Li X, Ling N, Dai J, et al. Development of a DNA aptamer against multidrug-resistant hepatocellular carcinoma for in vivo imaging. ACS Appl Mater Interfaces. 2021, 13:54656-54664.
4. Raber HF, Kubiczek DH, Bodenberger N, Kissmann A-K, D'souza D, Xing H, Mayer D, Xu P, Knippschild U, Spellerberg B, et al. FluCell-SELEX aptamers as specific binding molecules for diagnostics of the health relevant gut bacterium Akkermansia muciniphila. Int J Mo. Sci. 2021, 22:10425.
5. Kubiczek D, Raber H, Bodenberger N, Oswald T, Sahan M, Mayer D, Wiese S, Stenger S, Weil T, Rosenau F. The diversity of a polyclonal FluCell-SELEX library outperforms individual aptamers as emerging diagnostic tools for the identification of carbapenem resistant Pseudomonas aeruginosa. Chem Eur J. 2020, 26:14536-14545.
6. Sylvestre M, Saxby CP, Kacherovsky N, Gustafson H, Salipante SJ, Pun SH. Identification of a DNA aptamer that binds to human monocytes and macrophages. Bioconjug Chem. 2020, 31:1899-1907.

7. Zumrut HE, Batool S, Argyropoulos KV, Williams N, Azad R, Mallikaratchy PR. Integrating ligand-receptor interactions and in vitro evolution for streamlined discovery of artificial nucleic acid ligands. Mol Ther Nucleic Acids. 2019, 17:150-163.

8. Zumrut H, Yang Z, Williams N, Arizala J, Batool S, Benner SA, Mallikaratchy P. Ligand-guided selection with artificially expanded genetic information systems against TCR-CD3ε. Biochemistry. 2020, 59:552-562.

9. Pleiko K, Saulite L, Parfejevs V, Miculis K, Vjaters E, Riekstina U. Differential binding cell-SELEX method to identify cell-specific aptamers using high-throughput sequencing. Sci Rep. 2019, 9:8142.

10. Kacherovsky N, Cardle II, Cheng EL, Yu JL, Baldwin ML, Salipante SJ, Jensen MC, Pun SH. Traceless aptamer-mediated isolation of CD8(+) T cells for chimeric antigen receptor T-cell therapy. Nat Biomed Eng. 2019, 3:783-795.

11. Huang B-T, Lai W-Y, Chang Y-C, Wang J-W, Yeh S-D, Lin EP-Y, Yang P-C: A CTLA-4 antagonizing DNA aptamer with antitumor effect. Mol Ther Nucleic Acids. 2017, 8:520-528.

12. Vinod SP, Vignesh R, Priyanka M, Tirumurugaan KG, Sivaselvam SN, Dhinakar Raj G. Generation of single stranded DNA with selective affinity to bovine spermatozoa. Anim Biosci. 2021, 34:1579-1589.

13. Lai W-Y, Wang J-W, Huang B-T, Lin EP-Y, Yang P-C. A novel TNF-α-targeting aptamer for TNF-α-mediated acute lung injury and acute liver failure. Theranostics. 2019, 9:1741-1751.

14. Soldevilla MM, Hervas S, Villanueva H, Lozano T, Rabal O, Oyarzabal J, Lasarte JJ, Bendandi M, Inoges S, López-Díaz de Cerio A, Pastor F. Identification of LAG3 high affinity aptamers by HT-SELEX and Conserved Motif Accumulation (CMA). PLoS One. 2017, 12:e0185169.

15. Lai W-Y, Huang B-T, Wang J-W, Lin P-Y, Yang P-C. A novel PD-L1-targeting antagonistic DNA aptamer with antitumor effects. Mol Ther Nucleic Acids. 2016, 5:e397.

16. Prodeus A, Abdul-Wahid A, Fischer NW, Huang EHB, Cydzik M, Gariépy J. Targeting the PD-1/PD-L1 immune evasion axis with DNA aptamers as a novel therapeutic strategy for the treatment of disseminated cancers. Mol Ther Nucleic Acids. 2015, 4:e237.

17. Peinetti AS, Lake RJ, Cong W, Cooper L, Wu Y, Ma Y, Pawel GT, Toimil-Molares ME, Trautmann C, Rong L, et al. Direct detection of human adenovirus or SARS-CoV-2 with ability to inform infectivity using DNA aptamer-nanopore sensors. Sci Adv. 2021, 7:eabh2848.

18. Kacherovsky N, Yang LF, Dang HV, Cheng EL, Cardle II, Walls AC, McCallum M, Sellers DL, DiMaio F, Salipante SJ, et al. Discovery and characterization of spike N-terminal domain-binding aptamers for rapid SARS-CoV-2 detection. Angew Chem Int Ed. 2021, 60:21211-21215.

19. Marnissi B, Kamali-Moghaddam M, Ghram A, Hmila I. Generation of ssDNA aptamers as diagnostic tool for Newcastle avian virus. PLoS One. 2020, 15:e0237253.

20. Joseph DF, Nakamoto JA, Garcia Ruiz OA, Peñaranda K, Sanchez-Castro AE, Castillo PS, Milón P. DNA aptamers for the recognition of HMGB1 from Plasmodium falciparum. PLoS One. 2019, 14:e0211756.

21. Tsai Y-T, Liang C-H, Yu J-H, Huang K-C, Tung C-H, Wu J-E, Wu Y-Y, Chang C-H, Hong T-M, Chen Y-L. A DNA aptamer targeting Galectin-1 as a novel immunotherapeutic strategy for lung cancer. Mol Ther Nucleic Acids. 2019, 18:991-998.

22. Sanford AA, Rangel AE, Feagin TA, Lowery RG, Argueta-Gonzalez HS, Heemstra JM. RE-SELEX: restriction enzyme-based evolution of structure-switching aptamer biosensors. Chem Sci. 2021, 12:11692-11702.

23. Yang W, Yu H, Alkhamis O, Liu Y, Canoura J, Fu F, Xiao Y. In vitro isolation of class-specific oligonucleotide-based small-molecule receptors. Nucleic Acids Res. 2019, 47:e71.

24. Jeong S, Yang D, Beyene AG, Bonis-O'Donnell JTD, Gest AMM, Navarro N, Sun X, Landry MP. High-throughput evolution of near-infrared serotonin nanosensors. Sci Adv. 2019, 5:eaay3771.

25. Takahashi M, Amano R, Ozawa M, Martinez A, Akita K, Nakamura Y. Nucleic acid ligands act as a PAM and agonist depending on the intrinsic ligand binding state of P2RY2. Proc Natl Acad Sci USA. 2021, 118:e2019497118.
26. Alam KK, Chang JL, Lange MJ, Nguyen PDM, Sawyer AW, Burke DH. Poly-target selection identifies broad-spectrum RNA aptamers. Mol Ther Nucleic Acids. 2018, 13:605-619.
27. Gruenke PR, Alam KK, Singh K, Burke DH. 2′-fluoro-modified pyrimidines enhance affinity of RNA oligonucleotides to HIV-1 reverse transcriptase. RNA. 2020, 26:1667-1679.
28. Gordon CKL, Wu D, Pusuluri A, Feagin TA, Csordas AT, Eisenstein MS, Hawker CJ, Niu J, Soh HT. Click-particle display for base-modified aptamer discovery. ACS Chem Biol. 2019, 14:2652-2662.
29. Imashimizu M, Takahashi M, Amano R, Nakamura Y. Single-round isolation of diverse RNA aptamers from a random sequence pool. Biol Methods Protoc. 2018, 3: bpy004.
30. Soldevilla MM, Villanueva H, Casares N, Lasarte JJ, Bendandi M, Inoges S, López-Díaz de Cerio A, Pastor F. MRP1-CD28 bi-specific oligonucleotide aptamers: target costimulation to drug-resistant melanoma cancer stem cells. Oncotarget. 2016, 7:23182-23196.
31. Hervas-Stubbs S, Soldevilla MM, Villanueva H, Mancheño U, Bendandi M, Pastor F. Identification of TIM3 2'-fluoro oligonucleotide aptamer by HT-SELEX for cancer immunotherapy. Oncotarget. 2016, 7:4522-4530.
32. Popović M, Ellingson AQ, Chu TP, Wei C, Pohorille A, Ditzler MA. In vitro selections with RNAs of variable length converge on a robust catalytic core. Nucleic Acids Res. 2020, 49:674-683.
33. Ghaem Maghami M, Dey S, Lenz A-K, Höbartner C. Repurposing antiviral drugs for orthogonal RNA-catalyzed labeling of RNA. Angew Chem Int Ed. 2020, 59:9335-9339.
34. Liaqat A, Stiller C, Michel M, Sednev MV, Höbartner C. N6-isopentenyladenosine in RNA determines the cleavage site of endonuclease deoxyribozymes. Angew Chem Int Ed. 2020, 59:18627-18631.
35. Stephenson JD, Popović M, Bristow TF, Ditzler MA. Evolution of ribozymes in the presence of a mineral surface. RNA. 2016, 22:1893-1901.
36. Popović M, Fliss PS, Ditzler MA. In vitro evolution of distinct self-cleaving ribozymes in diverse environments. Nucleic Acids Res. 2015, 43:7070-7082.
37. Martin AS, Salamango DJ, Serebrenik AA, Shaban NM, Brown WL, Harris RS. A panel of eGFP reporters for single base editing by APOBEC-Cas9 editosome complexes. Sci Rep. 2019, 9:497.
38. McCann JL, Salamango DJ, Law EK, Brown WL, Harris RS. MagnEdit—interacting factors that recruit DNA-editing enzymes to single base targets. Life Sci Alliance. 2020, 3:e201900606.
39. Goh E, Okamura K. Hidden sequence specificity in loading of single-stranded RNAs onto Drosophila Argonautes. Nucleic Acids Res. 2018, 47:3101-3116.
40. Martinez BI, Stabenfeldt SE. In vivo phage display as a biomarker discovery tool for the complex neural injury microenvironment. Curr Protoc. 2021, 1:e67.
41. Yehl K, Lemire S, Yang AC, Ando H, Mimee M, Torres MDT, de la Fuente-Nunez C, Lu TK. Engineering phage host-range and suppressing bacterial resistance through phage tail fiber mutagenesis. Cell. 2019, 179:459-469.e9.
42. Salamango DJ, Ikeda T, Moghadasi SA, Wang J, McCann JL, Serebrenik AA, Ebrahimi D, Jarvis MC, Brown WL, Harris RS. HIV-1 Vif triggers cell cycle arrest by degrading cellular PPP2R5 phospho-regulators. Cell Rep. 2019, 29:1057-1065.e4.
43. Salamango DJ, Alam KK, Burke DH, Johnson MC. In vivo analysis of infectivity, fusogenicity, and incorporation of a mutagenic viral glycoprotein library reveals determinants for virus incorporation. J Virol. 2016, 90:6502-6514.
44. Reyes SG, Kuruma Y, Fujimi M, Yamazaki M, Eto S, Nishikawa S, Tamaki S, Kobayashi A, Mizuuchi R, Rothschild L, et al. PURE mRNA display and cDNA display provide rapid detection of core epitope motif via high-throughput sequencing. Biotechnol Bioeng. 2021, 118:1702-1715.

FASTAptameR 2.0 - User Interface Tutorial

Skyler T. Kramer Paige R. Gruenke Khalid K. Alam Dong Xu
Donald H. Burke

2022/July/28

Contents

1 Introduction

FASTAptameR 2.0 expands the bioinformatics pipeline that was first released as FASTAptamer (Alam KK 2015). Like its predecessor, FASTAptameR 2.0 is an open-source toolkit designed to quickly and easily analyze populations of sequences resulting from combinatorial selections. This updated version is an R-based platform that features a graphical user interface (UI), interactive graphics, more modules, and a faster implementation of the original clustering algorithm.

This user guide walks through installation of the software and operations for each of the modules, and it highlights what options are available to analyze data through the UI.

The key features of the ten modules of FASTAptameR 2.0 are outlined below in **Section 1.1**. **Figure 1** gives an overview of how these modules connect to one another. Additionally, a summary of input and output file types is given in **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 these data will be saved on the server to be passed between modules.

A feature of FASTAptameR 2.0 is that 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 from FASTAptameR-Count 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 that are especially applicable to, and useful for, the combinatorial selections field.

1.1 Overview

- **FASTAptameR-Count**
  - This module is the entry point into FASTAptameR 2.0
  - Input: one preprocessed FASTQ/A file
  - Workflow:
    1. count the occurrence of each unique sequence (`Reads`)
    2. sort by counts in descending order (`Rank`)
    3. normalize counts to reads per million (`RPM`)
  - Interactive plotting:
    1. line plot of reads for each unique sequence sorted by 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 file (Note: FASTA output from this module is referred to as "counted FASTA" files throughout this tutorial)
- **FASTAptameR-Translate**
  - Input: one counted FASTA file
  - Workflow: translate D/RNA sequences to amino acid sequences
  - Interactive plotting:
    1. line plot of reads for each unique sequence sorted by rank
    2. histograms of sequence lengths - one for the unique sequences and one for all reads
  - Output: FASTA or CSV file
- **FASTAptameR-Motif_Search**
  - Input: one counted FASTA file and user-defined, comma-separated query patterns

  - Workflow: search for user-defined query patterns in sequences
  - Output: FASTA or CSV file

- **FASTAptameR-Motif_Tracker**
  - Input: at least two counted FASTA files and query list (either motifs or full sequences)
  - Workflow: track how user-defined motifs or sequences from the query list change across populations
  - Interactive plotting: line plot of each query's RPM across the populations
  - Output: CSV file to summarize each query + CSV file of enrichment values

- **FASTAptameR-Motif_Discovery**
  - Input: one counted FASTA file
  - Workflow: apply Fast String-Based Clustering (FSBC) (Kato et al. 2020) to identify over-enriched, ungapped motifs with user-defined lengths
  - Interactive plotting: scatter plot of rank by normalized Z-score vs `-log10(p-value)`, with size and color representing the motif length
  - Output: CSV file

- **FASTAptameR-Mutation_Network**
  - Input: one counted FASTA file and two query sequences
  - Workflow:
    1. compute the Levenshtein edit distance (LED) for every pairwise combination of sequences in the file
    2. omit any combination with an LED greater than a user-specified value
    3. use Dijkstra's Shortest Path Algorithm to find the shortest evolutionary path between the two user-defined sequences
  - Output: CSV file

- **FASTAptameR-Distance**
  - Input: one counted FASTA file and query sequence
  - Workflow: compute the LED between the query sequence and all other provided sequences
  - Interactive plotting: histograms of edit distances - one for the unique sequences and one for all reads
  - Output: FASTA or CSV file

- **FASTAptameR-Data_Merge**
  - Input: at least two counted FASTA files
  - Workflow: merge each FASTA file with union, intersection, or left join
  - Interactive plotting:
    1. bar plot of sequence persistence
    2. UpSet plot of population intersections
  - Output: CSV file

- **FASTAptameR-Sequence_Enrich**
  - Input: two counted FASTA files
  - Workflow: calculate how each sequence enriches across populations
  - Interactive plotting:
    1. histogram of `log2(Enrichment)`
    2. scatter plot of RPM
    3. Ratio average (RA) plot, displaying average log-RPM (`A`) and log ratio (`R`)
    4. box plot of sequence enrichment per cluster; only available if clustered FASTA files
  - Output: CSV file

- **FASTAptameR-Position_Enrich**

  - Input: one enrichment CSV from FASTAptameR-Sequence_Enrich and reference sequence
  - Workflow: for each position of the reference sequence, compute the average enrichment of non-reference residues in the data
  - Interactive plotting:
    1. bar plot of average enrichment per position of reference sequence
    2. heat map of average enrichment per position of reference sequence grouped by residues
  - Output: No direct file output (only interactive plots)

- **FASTAptameR-Cluster**
  - Input: one counted FASTA file
  - Workflow:
    1. filter out low-read sequences based on user-defined input
    2. treat the 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 Steps 2 and 3 until all sequences are clustered or a maximum number of clusters are created
  - Output: FASTA or CSV file (Note: FASTA output from this module is referred to as "clustered FASTA" files throughout this tutorial)

- **FASTAptameR-Cluster_Diversity**
  - Input: one clustered FASTA file
  - Workflow: provide metadata for each cluster
  - Interactive 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 file

- **FASTAptameR-Cluster_Enrich**
  - Input: at least two CSVs from FASTAptameR-Cluster_Diversity
  - Workflow: calculate how each cluster enriches across populations
  - Interactive plotting: line plot of each the total RPM per cluster for each seed per population
  - Output: CSV file to summarize each cluster + CSV file of enrichment values

Please note that many module inputs require counted or clustered FASTA files, but this is only the minimum required file type. For example, **FASTAptameR-Cluster** requires at least a counted FASTA file, but the file may also be translated, filtered by distance to a query sequence, and filtered for a set of user-defined motifs before it is clustered.

1.2 A note on plotting in FASTAptameR 2.0

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 download the plots. This last functionality is provided by the camera icon (first icon on the left as you hover over the image of the plots). Finally, double-clicking the image should reset it (*e.g.*, remove zoom or crop effects).

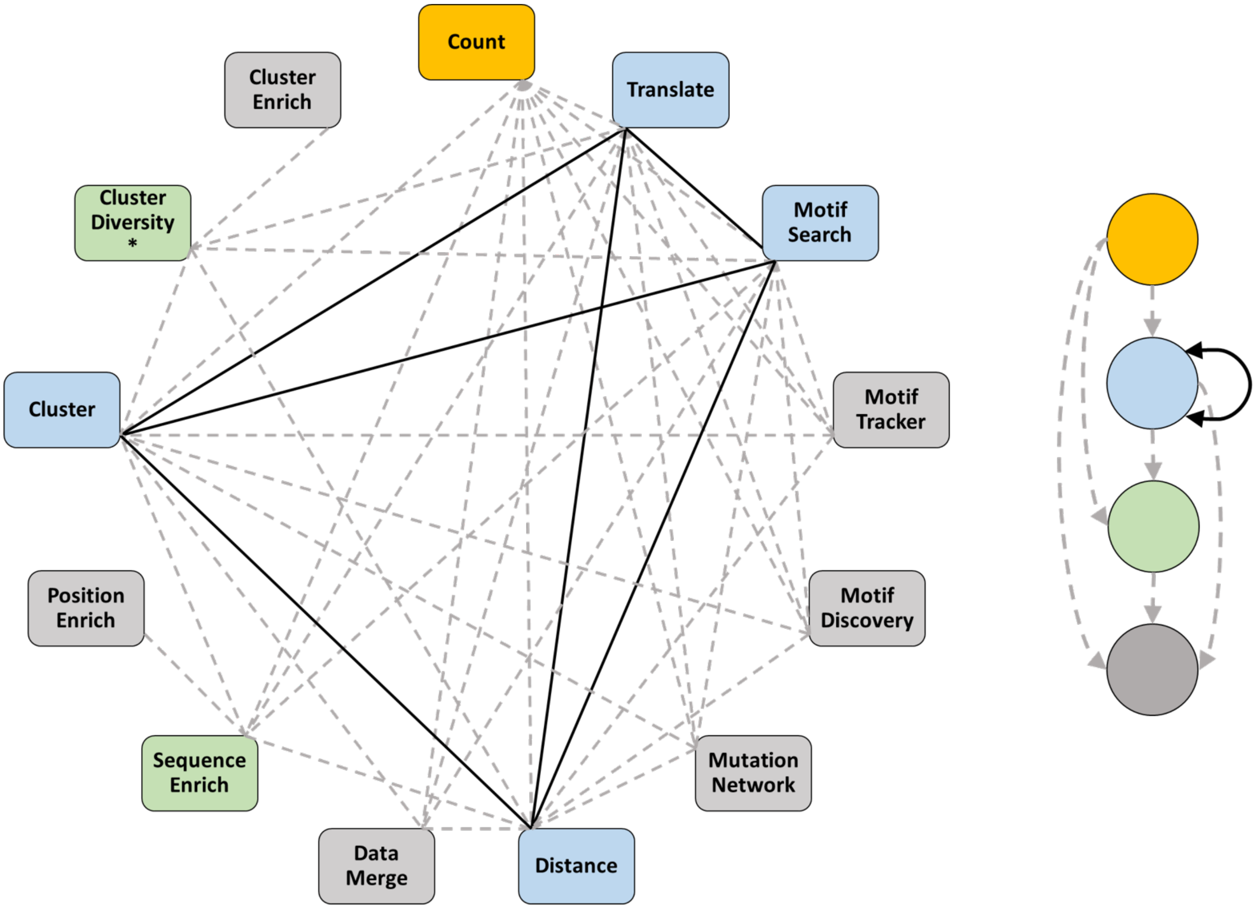


Figure 1: Graph Diagram of Module Connections. All workflows must start with the Count module (gold). Blue modules can be either intermediate or terminal steps of pipelines, and they can feed into other blue modules, green modules, or gray modules. Green modules can be intermediate or terminal steps of pipelines, but they can only feed into gray modules. Gray modules are always terminal steps of pipelines. Solid black lines are bidirectional, and dashed gray lines are unidirectional. The simplified graphical legend on the right indicates the directionality of the connections.

Table 1: Module Input and Output File Types

Module	Min. Input Files	Output Files
FASTAptameR-Count	Preprocessed FASTQ/A	FASTA or CSV
FASTAptameR-Translate	Counted FASTA	FASTA or CSV
FASTAptameR-Motif_Search	Counted FASTA	FASTA or CSV
FASTAptameR-Motif_Tracker	2+ counted FASTAs	CSV
FASTAptameR-Motif_Discovery	Counted FASTA	CSV
FASTAptameR-Mutation_Network	Counted FASTA	CSV
FASTAptameR-Distance	Counted FASTA	FASTA or CSV
FASTAptameR-Data_Merge	2+ counted FASTAs	CSV
FASTAptameR-Sequence_Enrich	2 counted FASTAs	CSV
FASTAptameR-Position_Enrichment	Enrich CSV	Plots
FASTAptameR-Cluster	Counted FASTA	FASTA or CSV
FASTAptameR-Cluster_Diversity	Clustered FASTA	CSV
FASTAptameR-Cluster_Enrich	2+ CSVs from Cluster_Diversity	CSV

2 How to get started

Like its predecessor FASTAptamer, FASTAptameR 2.0 is designed to be easy to use and accessible for practitioners of combinatorial selection, and to be open-source so that the community can interact with the source code. There are three ways for users to interact with FASTAptameR 2.0. The *web server* (https://fastaptamer2.missouri.edu/) is the easiest way to interact with this application because it only requires an internet connection and browser. The user interface can also run on your local system as a *Docker container*. Finally, all code can be pulled from GitHub at https://github.com/SkylerKramer/FASTAptameR-2.0.

2.1 Web server interface

The web server interface is the fastest and easiest way to use the FASTAptameR 2.0 User Interface, especially for relatively small data sets. 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 at the University of Missouri - Columbia. However, this option only works if the files uploaded to any given module are less than 2 GB.

2.2 Docker interface

The web server can also be run locally on your machine(s) via Docker. This interface functions identically to the web server interface and allows the user full access to all FASTAptameR 2.0 functions without having to rely on continuous internet connection and without the limitations of files sizes or file transfer speeds. In many cases, using the Docker interface is the most reliable and convenient way to explore your data set.

Docker is a convenient tool that may be used to construct *images* of software. The *image* 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`).

The **three steps** required for initial installation of FASTAptameR 2.0 via Docker. **Step 1** is required for the initial installation of Docker. **Step 2** is required to get access to FASTAptameR 2.0 and all subsequent versions. **Step 3** is required to run the application. After initial installation of a given version of FASTAptameR 2.0, only Step 3 needs to be repeated in subsequent sessions. Instructions for accessing newer versions as they become available are in the instructions below for Step 2.

Step 1. Install Docker to establish a "Docker-active terminal" on your local system.

The first step is either to install Docker (on a Linux machine) or to install Docker Desktop (on a Windows or Mac machine) to establish a "Docker-active terminal" on your local system. For reference, Windows users can access their terminal (the Command Prompt) by searching the machine for `cmd`. Mac users can access their terminal by searching for `Terminal` from the Launchpad, Finder, Spotlight, *etc.*

If you would like to see extensive details on Docker or its installation, please visit https://www.docker.com/ and https://docs.docker.com/get-docker/, respectively. Importantly, the FASTAptameR 2.0 *image* is built on Linux. Thus, it is necessary to run it from a Linux environment or virtual machine. This should automatically happen after installing Docker for Linux or Docker Desktop for Mac. Windows users must also install Docker Desktop and will typically need to enable virtualization through their Bios (see https://www.tutorialspoint.com/windows10/windows10_virtualization.htm for more details).

Successful completion of this step will yield a "Docker-active terminal". Linux, Windows, or Mac users can check if their terminal is Docker-active by running the following command in the terminal, which should return which version of Docker has been installed if it was installed successfully:

```
docker version
```

Step 2. Pull the FASTAptameR 2.0 image from the Docker Hub repository.

The FASTAptameR 2.0 *image* must be pulled from a repository (*i.e.*, Docker Hub) by running the following command in a Docker-active terminal:

```
docker pull skylerkramer/fastaptamer2:publicupload08
```

Please note that `publicupload08` is the most recent version of FASTAptameR 2.0 as of the time of writing this tutorial. To access future updates as they become available, please refer to this section of the dynamic User Guide at https://github.com/SkylerKramer/FASTAptameR-2.0 or at the main Docker Hub page at https://hub.docker.com/repository/docker/skylerkramer/fastaptamer2.

Step 3. Launch FASTAptameR 2.0.

Once you have this application's image, run the following from a Docker-active terminal:

```
docker run -d --rm -p 3838:3838 skylerkramer/fastaptamer2:publicupload08
```

This 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 entering `localhost:3838` into your web browser address bar.

Explanation of flags from Step 3:

- `-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 stop the container upon exit
- `-p 3838:3838`: publish `3838` host port (first number) to the `3838` container port (second number)
- `skylerkramer/fastaptamer2:publicupload08`: the local path to the **FASTAptameR 2.0** Docker *image*; please see **Step 2** for the most recent version
- `localhost:3838`: navigate here from your web browser to start interacting with **FASTAptameR 2.0**

2.3 R interface

The third way to interact with FASTAptameR 2.0 is through the source code. All code is publicly available at https://github.com/SkylerKramer/FASTAptameR-2.0. This will allow R developers to adjust the code to their specific needs. Please note that all dependencies in this app (*e.g.*, Shiny, ggplot2, *etc.*) must be installed to use this app. A full list of dependencies and their installation instructions are available on the GitHub page above.

2.4 Software usage

If you use, adapt, or modify FASTAptameR 2.0, please cite: (Alam KK 2015) and (Kramer et al. 2022).

For any questions or concerns, please email burkelab@missouri.edu or make a GitHub issue.

FASTAptameR 2.0 is distributed under a GNU General Public License version 3.0.

3 Tutorial

3.1 Data requirements

FASTAptameR 2.0 utilizes many string-based functions. 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, such as trimming flanking sequences or filtering for read quality, must be made outside of this application. Please note that there are many preprocessing tools. A few of them are shown here:

- cutadapt (Martin 2011)
- Trimmomatic (Bolger, Lohse, and Usadel 2014)
- fastp (Chen et al. 2018)
- FASTX-Toolkit

3.2 Sample Data and Uploading User Data

3.2.1 Sample data

All data shown in this tutorial come from published selections for RNA aptamers with affinity for HIV-1 reverse transcriptase (RT).

Rnd14: Briefly, 14 rounds of selection gave a population that was dominated by pseudoknots, especially those with a well-defined motif known as the Family 1 Pseudoknot (F1Pk) (Burke DH 1996).

Rnd15: Later analysis monitored population dynamics in response to increasing selection pressure and revealed two other RNA motifs - the '(6/5) asymmetric loop' and the 'UCAA' motif - that are more potent inhibitors of RT than the previously identified pseudoknots (Ditzler MA 2013; Whatley AS 2013). Notably, RT inhibition of F1Pk aptamers was abolished by the R277L mutation while '(6/5) asymmetric loop' and the 'UCAA' motif aptamers were insensitive to this mutation.

Rnd17: We then subjected the Rnd14 population above to three additional cycles of selection for affinity to RTs from phylogenetically diverse lentiviruses, including several that carried the R227K mutation. This "Poly-Target" selection approach identified aptamer subsets with broad target recognition and RT inhibition (Alam et al. 2018).

All of these data are available as preprocessed (trimmed and filtered) FASTA files from http://burkelab.missouri.edu/fastaptamer.html.

3.2.2 Moving on to your data

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

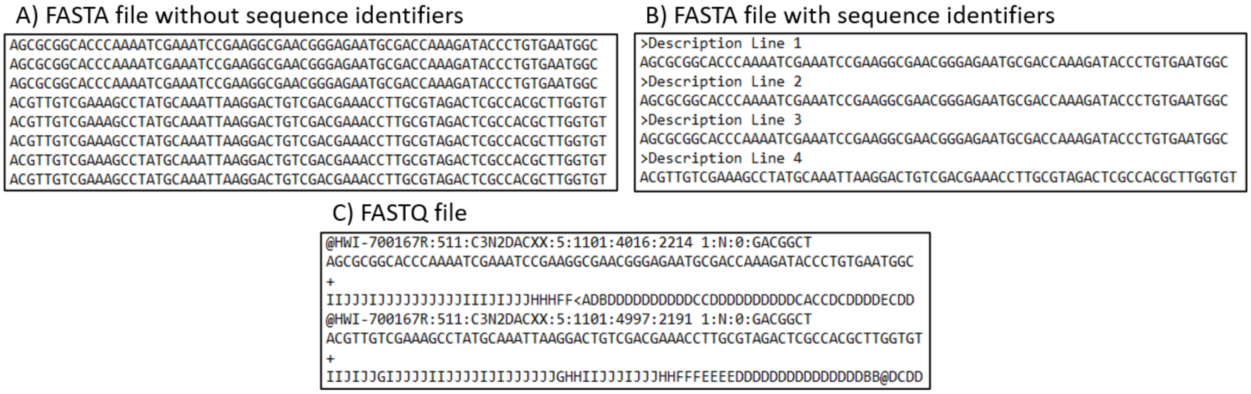
Analyzing sequences from combinatorial selections typically begins with counting how many of each species are present. This baseline information immediately establishes whether the population has converged on a small handful of dominant sequences or on a large number of enriched sequences, of if instead there is still an enormous diversity with little convergence. Each of these scenarios is immediately evident in a plot of the total number of reads ordered by rank. Similarly, a simple count of how many of a given species is present is the input data required for many other layers of analysis.

Thus, the pipeline begins with FASTAptameR-Count, which is the gateway to the rest of FASTAptameR 2.0. Sequence files must first be processed through the FASTAptameR-Count module to generate a new file that removes redundant copies and tabulates the abundance of each sequence, its rank within the population, and a normalized value of the abundance (reads per million, or 'RPM') that normalizes the raw number of reads to the population size. This normalization enables comparisons of relative abundances across populations of different sizes. These statistical values are written into the sequence identifiers of the downloadable FASTA file.

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. Keep the *count* file in an easily accessible folder, as this file will serve as input for many other FASTAptameR 2.0 modules (see **Figure 1**).

FASTAptameR-Count accepts a FASTQ/A file chosen with the file browser and returns a *counted* data table as output that can be downloaded as a FASTA or CSV file. The module includes a link to sample data.

Input FASTQ files should be properly formatted (4 lines per entry with the second 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 (specifically, the rank order of that sequence in the population, its read count, and its normalized read count - see below). **Importantly, all sequence data (including quality scores) should be on the same line, never spread across multiple lines**. Sample input files are shown in **Figure 2**.



A) FASTA file without sequence identifiers

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

B) FASTA file with sequence identifiers

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

C) FASTQ file

```
@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
```

Figure 2: Examples of valid inputs to FASTAptameR-Count. These are all valid inputs to FASTAptameR-Count, which is the first step in using this workflow. A) FASTA file without sequence identifier lines. B) FASTA file with sequence identifier lines. C) FASTQ file.

All modules directly connected to FASTAptameR-Count are shown in **Figure 3**, and a screenshot of the module interface is shown in **Figure 4**.

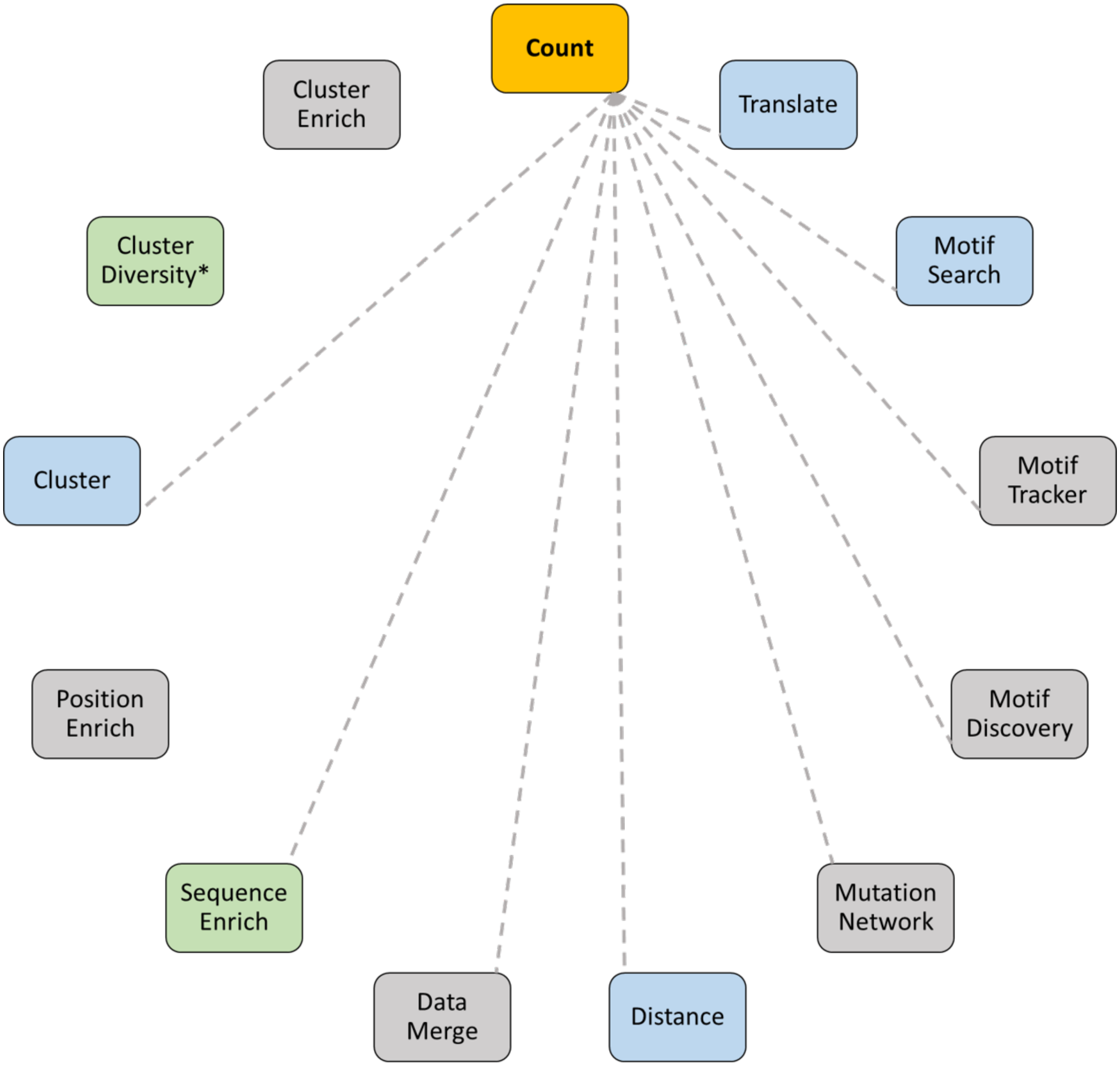
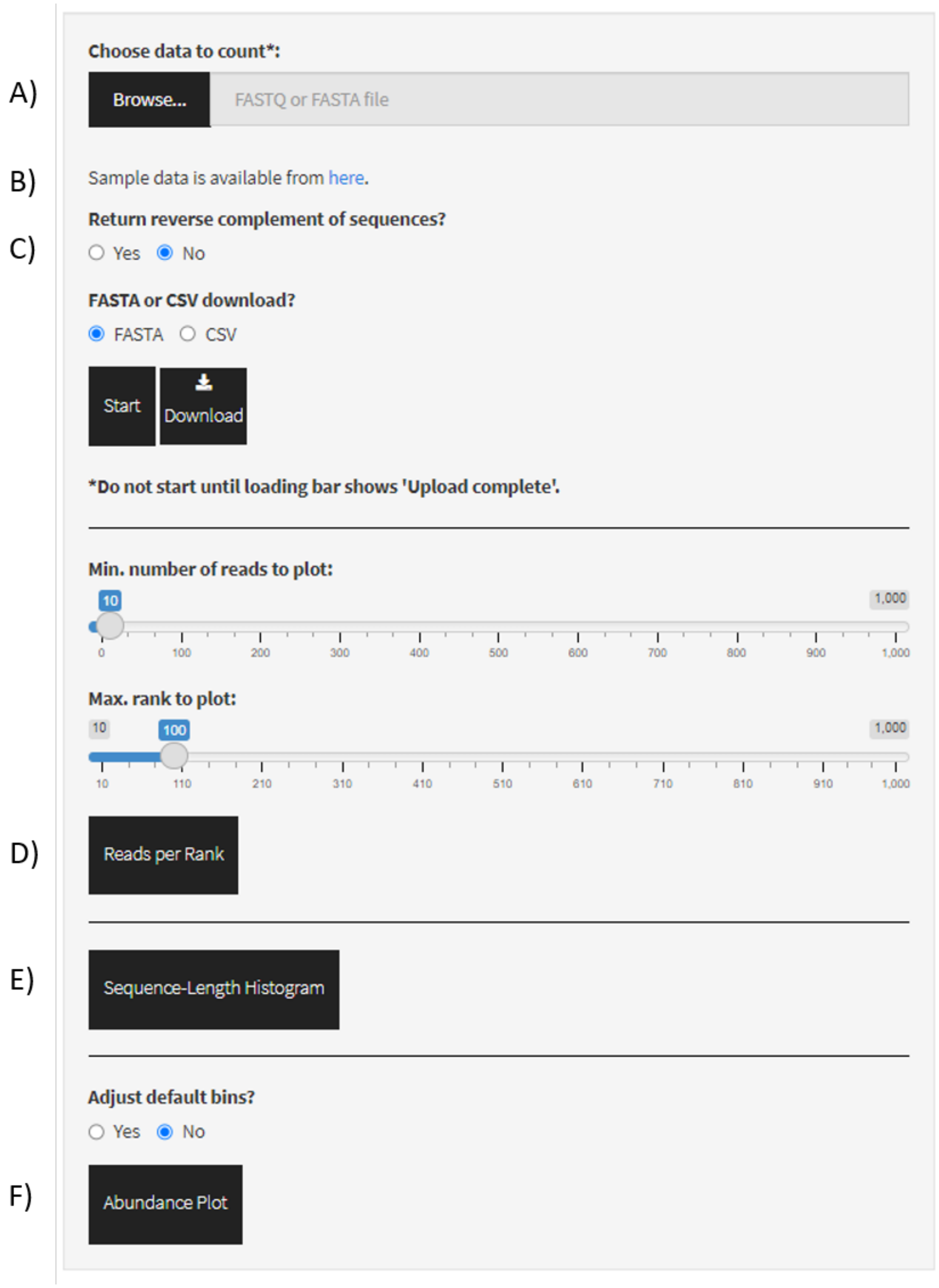


Figure 3: All modules connected to FASTAptameR-Count.



Choose data to count*:

FASTQ or FASTA file

Sample data is available from here.

Return reverse complement of sequences?

○ Yes ● No

FASTA or CSV download?

● FASTA ○ CSV

*Do not start until loading bar shows 'Upload complete'.

Min. number of reads to plot:

10 — 1,000 (0, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1,000)

Max. rank to plot:

10 — 100 — 1,000 (10, 110, 210, 310, 410, 510, 610, 710, 810, 910, 1,000)

Adjust default bins?

○ Yes ● No

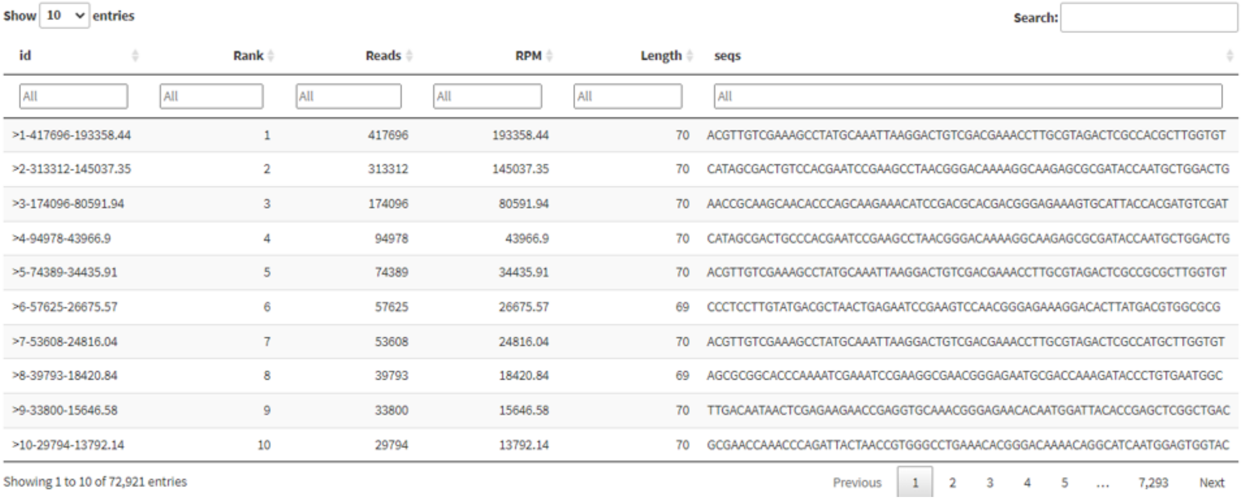
Figure 4: Screenshot of FASTAptameR-Count.

3.3.2 Usage

The input FASTQ/A file must be chosen with the file browser (**Figure 4A**). Sample data can be accessed from the link in **Figure 4B**.

For file uploads, please wait for the loading bar to show *Upload complete* before using the Start button. Once the files are fully uploaded, the `Start` button will begin the counting process. The results will be displayed as a data table on the right side of the screen. You can optionally count only the reverse complements of sequences by using the radio button shown in **Figure 4C**.

A sample output data table is shown in **Figure 5**. Importantly, numeric columns in this output table and all other output tables are filterable by range (*e.g.*, to select the top 100 ranked sequences, type `1 ... 100` into the `Rank` column). To use some later features of FASTAptameR 2.0 (*i.e.*, the Distance and Position Enrichment modules), you must apply such a filter to the `Length` column such that all sequences are of the same length (*e.g.*, `70 ... 70` only retains sequences of length 70).



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

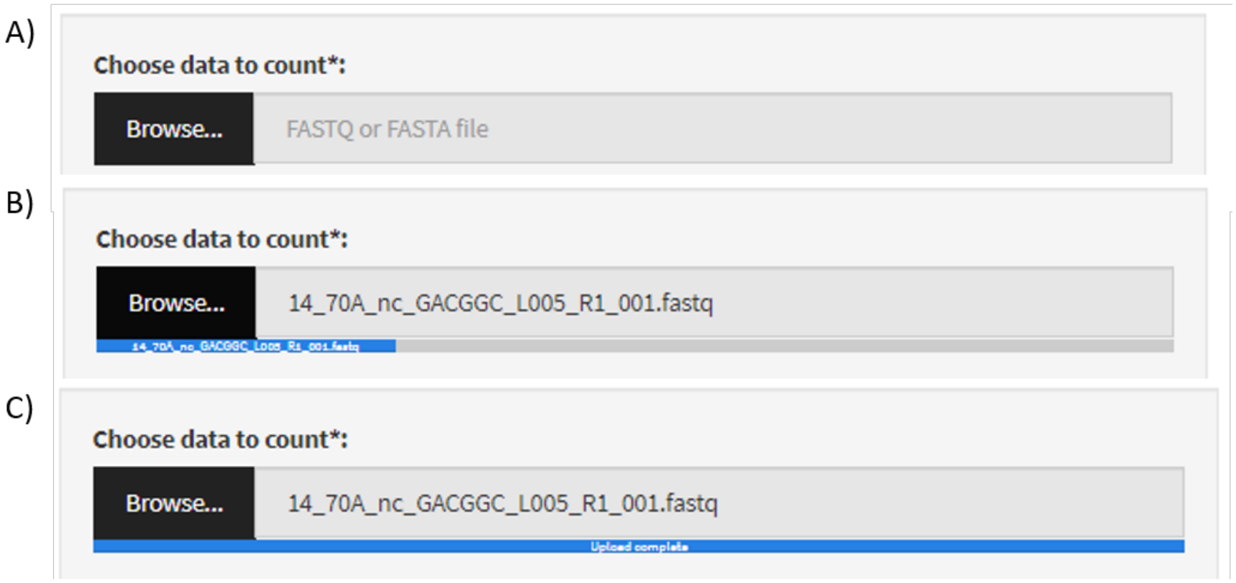
Figure 5: FASTAptameR-Count Output. Note the search box in the top right corner. This appears in most module outputs and allows the user to search the table for specific strings.

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` (*i.e.*, Reads per Million) is the value of `Reads`, normalized by the total population size: `RPM = Reads / (Population Size / 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). Again, 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 Troubleshooting

Do not start the module before the upload is finished. If you start the module before the upload is finished, then you will *get an error message* that says `No file or link provided!`. To fix this error, reupload the file, *wait for it to finish uploading*, and then start the module. If any errors persist, please refresh the page. Examples of incomplete and complete progress bar displays are given in **Figure 6**.



Choose data to count*: FASTQ or FASTA file

Choose data to count*: 14_70A_nc_GACGGC_L005_R1_001.fastq

14_70A_nc_GACGGC_L005_R1_001.fastq

Choose data to count*: 14_70A_nc_GACGGC_L005_R1_001.fastq

Upload complete

Figure 6: Example loading bars. A) Widget for file selection before upload. B) Loading bar (blue) during upload. C) Loading bar after successful upload. In any module requiring a file upload (*i.e.*, every module except for FASTAptameR-Count when a link is provided), this loading bar **must** show `Upload complete` before pressing the start button.

3.3.4 Plotting

This module can generate three types of interactive plots based on the counted data to illustrate the overall distribution of sequence abundances in the population: a line plot of reads for each unique sequence sorted by rank (button shown in **Figure 4D**, output shown in **Figure 7A,B**), two histograms of sequence lengths (button shown in **Figure 4E**, output shown in **Figure 7C**), and a sequence abundance bar plot (button shown in **Figure 4F**, output shown in **Figure 7D**). Line plots are filterable by 1) minimum number of reads to plot and 2) maximum rank to plot, and both values are chosen with a slider bar. The histograms, however, 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. The breakpoints forming these bins are, by default, set to the following: `Reads = 1`, `1 < Reads < 10`, `10 <= Reads < 100`, `100 <= Reads < 1000`, `1000 <= Reads <= max(Reads)`. However, the user can toggle singletons on/off or customize these breakpoints by selecting `Yes` in the respective prompt. New breakpoints should be entered as a comma-separated list. An example is given in **Figure 8**.

The first bin contains all sequences with `Reads < min(break point)` unless singletons are desired, in which case the set of sequences with `Reads = 1` becomes its own respective bin. The final bin contains all sequences with `max(break point) <= Reads`. Intermediate bins contain sequences between consecutive breakpoints, where the minimum and maximum values are inclusive and exclusive, respectively. For example, if singletons are not desired and the breakpoints are `20,2000`, then the bins are as follows: (`Reads < 20`), (`20 <= Reads < 2000`), and (`2000 <= Reads <= max(Reads)`).

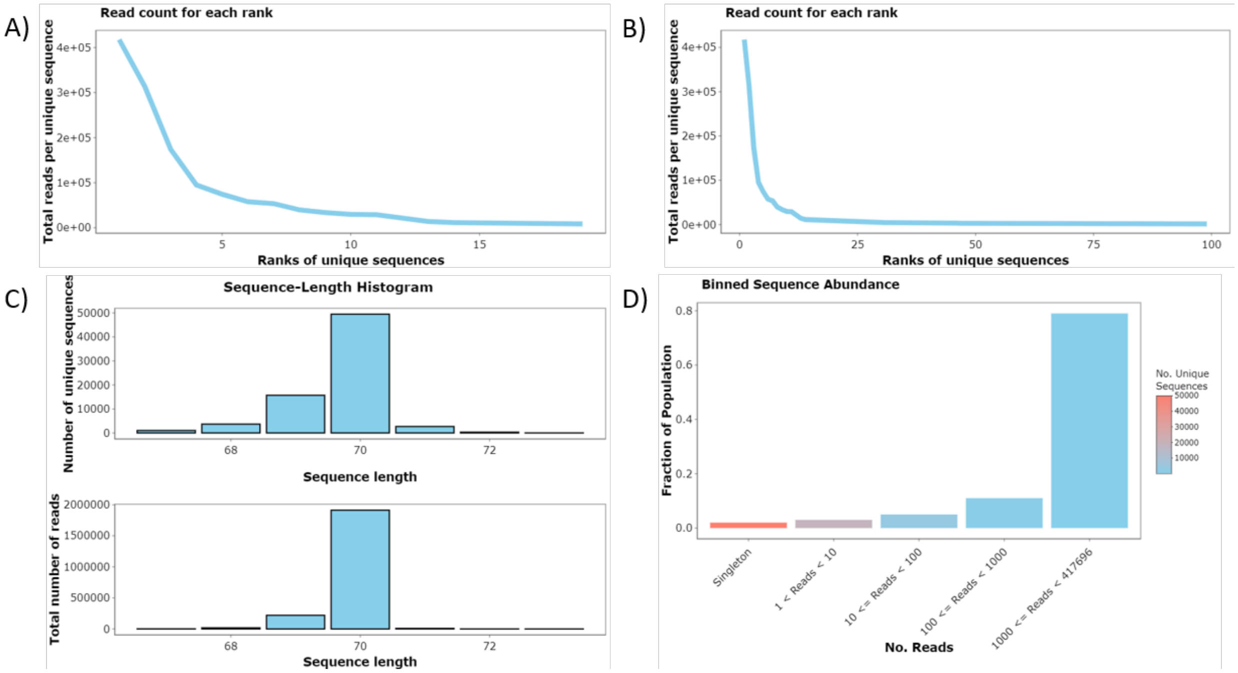
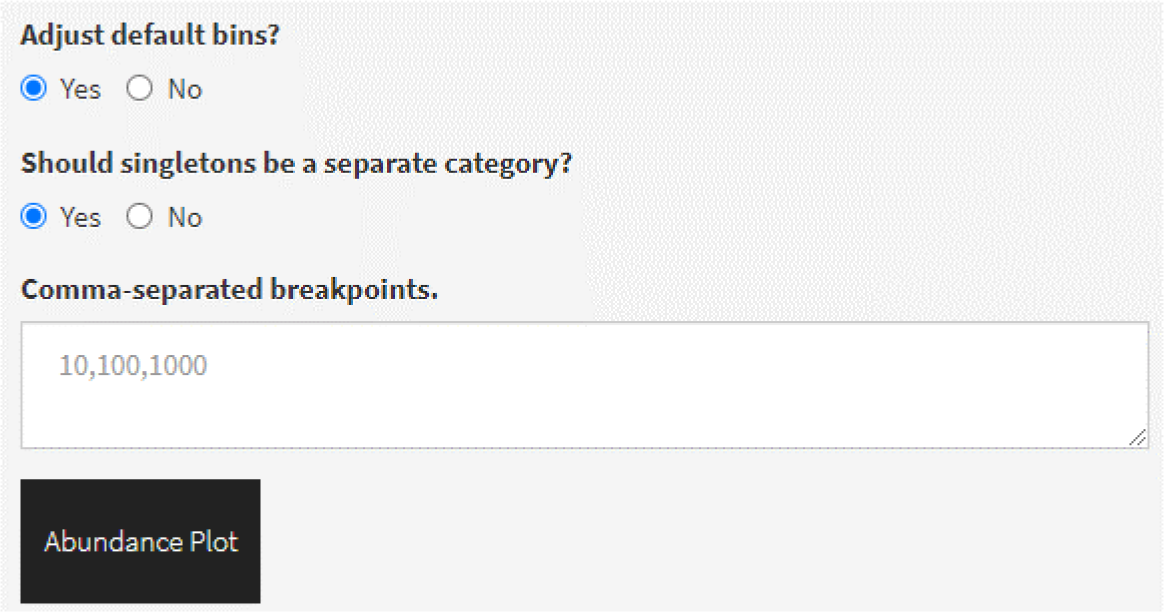


Figure 7: FASTAptameR-Count Plots. A) A line plot showing the total number of reads for the 20 most abundant sequences. B) The same plot with the 100 most abundant sequences. C) Histograms of sequence lengths for unique sequences (top) and all reads (bottom). D) Binned sequence abundance bar plot where `x` corresponds to discrete bins of read counts, `y` corresponds to the fraction of the total population, and color corresponds to the number of unique sequences per bin.



Adjust default bins?

● Yes ○ No

Should singletons be a separate category?

● Yes ○ No

Comma-separated breakpoints.

10,100,1000

Figure 8: Example of how to customize the bins of the sequence abundance plot. Top radio button must be set to `Yes`. The second radio button determines whether singletons are a separate bin. The text input takes a comma-separated list of breakpoints. This image reflects the default breakpoints (listed in the text).

3.4 FASTAptameR-Translate

3.4.1 Description

In addition to analyzing nucleic acid populations, FASTAptameR 2.0 can also analyze combinatorial libraries based on translated peptide or protein products such as phage display, partially randomized cellular proteins selected for bioactivity, and evolving viral populations from patients. In addition, protein engineering and synthetic biology research makes increasing use of non-standard amino acids and expanded genetic alphabets; both of these innovations can be analyzed with FASTAptameR 2.0.

FASTAptameR-Translate translates input nucleotide sequences into amino acid sequences following either the standard genetic code, a biologically derived, nonstandard code, or a user-defined code. The input nucleotide sequences are treated 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.

All modules directly connected to FASTAptameR-Translate are shown in **Figure 9**, and a screenshot of the module interface is shown in **Figure 10**.

3.4.2 Usage

The input FASTA file must be chosen with the file browser. The open reading frame may be selected by the first set of radio buttons (`DEFAULT = 1`) (**Figure 10A**). The second set of radio buttons indicates whether data for nucleotide sequences that encode the same amino acid sequence should be merged (`DEFAULT = Yes`) (**Figure 10B**). 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.

Merging the data for redundant sequences (button set to `No`) focuses attention on the encoded peptide sequences, irrespective of which nucleotide sequences gave rise to them, which can be especially informative for protein structure/function studies. In contrast, treating each evolving species separately (button set to `Yes`) allows users to discern contributions of nucleotide sequence to overall fitness by comparing relative enrichments of different sequences that that encode the same translated product.

The dropdown menu in this UI (**Figure 10C**) allows the user to select one of sixteen genetic codes for translation (`DEFAULT = Standard`) (Gasteiger 2003):

1. Standard
2. Vertebrate mitochondrial
3. Yeast mitochondrial
4. Mold, protozoan, and coelenterate mitochondrial + Mycoplasma / Spiroplasma
5. Invertebrate mitochondrial
6. Ciliate, dasycladacean and Hexamita nuclear
7. Echinoderm and flatworm mitochondrial
8. Euplotid nuclear
9. Alternative yeast nuclear
10. Ascidian mitochondrial
11. Alternative flatworm mitochondrial
12. Blepharisma nuclear
13. Chlorophycean mitochondrial
14. Trematode mitochondrial
15. Scenedesmus obliquus mitochondrial
16. Pterobranchia mitochondrial

The user may also customize the translation code by selecting `Yes` in the third set of radio buttons (**Figure 10D**) prior to translating. If `Yes`, then comma-separated codon / translation pairs may be entered in the

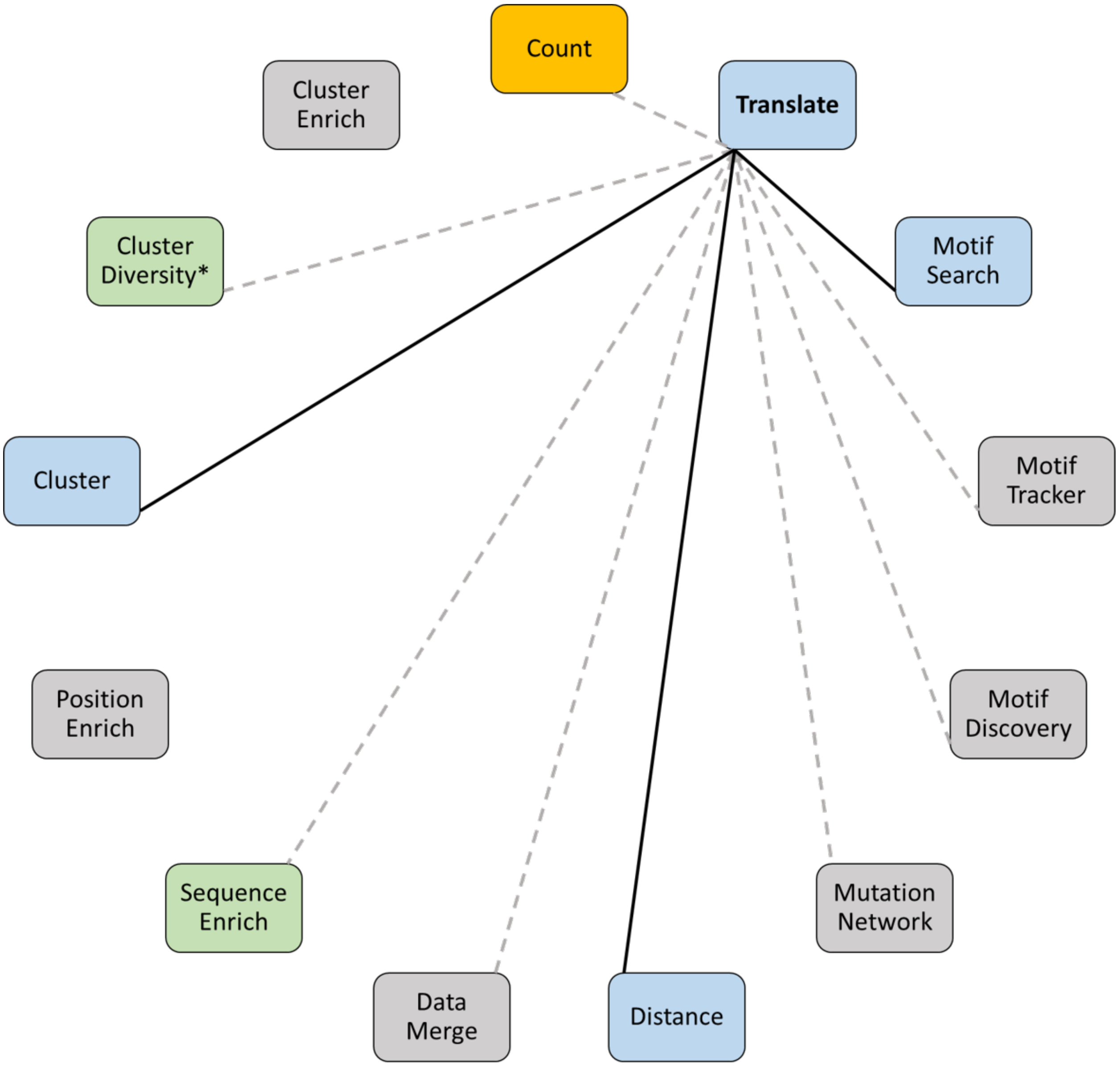
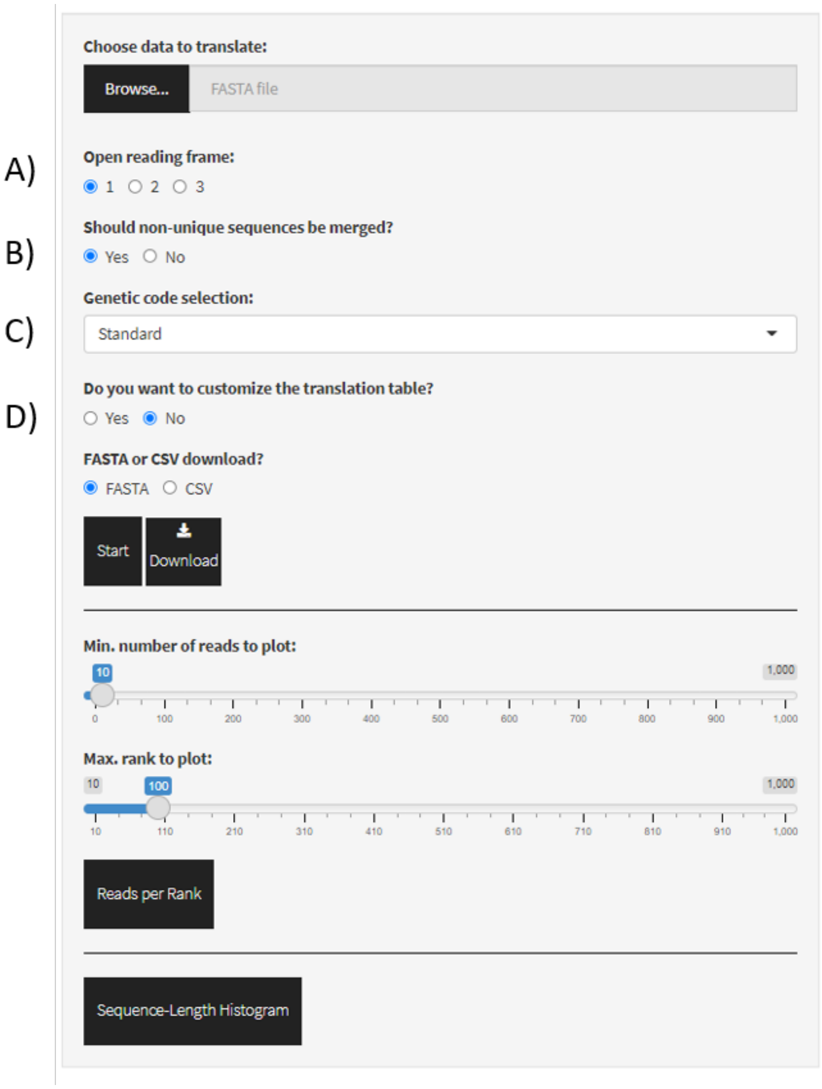


Figure 9: All modules connected to FASTAptameR-Translate.



Choose data to translate:

Browse... FASTA file

Open reading frame:

● 1 ○ 2 ○ 3

Should non-unique sequences be merged?

● Yes ○ No

Genetic code selection:

Standard

Do you want to customize the translation table?

○ Yes ● No

FASTA or CSV download?

● FASTA ○ CSV

Min. number of reads to plot:

10

Max. rank to plot:

100

Figure 10: Screenshot of FASTAptameR-Translate.

resulting text box (*e.g.*, `GAT,Q` to recode `GAT` from its standard `Glu (E)` to `Gln (Q)`. Additional pairs can be included but must be on separate lines. 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.

Only 3-letter codons and 1-letter translations are currently accepted.

Non-standard alphabets are allowed for both input and output. For example, to translate an input amber STOP codon (`UGA`) as a nonstandard amino acid designated as `2`, the resulting text box would be entered as `TGA,2`. Similarly, to translate a triplet that contains the AEGIS nucleotide nitropyridine (`Z`) in the first position of a glycine codon, with `Trp` as the resulting output, the resulting text box would be as `ZGG,W`.

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 a line plot of reads for each unique sequence sorted by rank and two histograms of sequence lengths. These plots illustrate overall population structure for the translated products and are analogous to the corresponding plots from FASTAptameR-Count. See that section for more details.

3.5 FASTAptameR-Motif_Search

3.5.1 Description

It is often useful to find all occurrences of a given sequence motif within a population. Even when parts of the functional structure are not defined simply by sequence (such as generic base-paired helices), many functional biomolecules contain 'signature sequences' that make useful first-pass filters. The motif of interest may be fully contiguous, such as the 8 nucleotide apical loop of the bacteriophage T4 gene 43 translational operator (`AAUACUC`) (Craig Tuerk and Gold 1990), or it may be discontinuous, such as the 11 nucleotides of the (6/5) asymmetric loop aptamer (abbreviated as "(6/5)AL"), in which a stem is interrupted by an asymmetric internal loop with the sequence `ARCGUY` on one strand and `RARAC` on the other. For the (6/5)AL motif, both elements must be present within the same sequence for that sequence to form a functional (6/5)AL aptamer. (Note that the `A` in `ARCGUY` is typically the last base of the 5' constant region and is removed when trimming sequences).

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).

All modules directly connected to FASTAptameR-Motif_Search are shown in **Figure 11**, and a screenshot of the module interface is shown in **Figure 12**.

3.5.2 Usage

The input FASTA file must be chosen with the file browser. The following text box (**Figure 12A**) 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 first set of radio buttons (**Figure 12B**) determines whether the output has parentheses set around identified patterns. The default is to set to `No`, and patterns are highlighted in yellow in the output data without parentheses. For example, when `pattern = GGC` and `sequence = AAAGGCT`, the default output is `AAAGGCT` with `GGC` highlighted. Setting this button to `Yes` returns the output as `AAA(GGC)T`, and `GGC` is still highlighted. When two or more patterns overlap, output highlights the complete match when the button is set to `No` but only displays parentheses around the first queried search term that is matched when the button is set to `Yes`. For example, when `pattern = AGGC,GGCT` and `sequence = AAAGGCT`, the default output is `AAAGGCT` (with `AGGCT` highlighted), while it is `AA(AGGC)T` (with `AGGC` highlighted) when the parentheses option is turned on. Note that parentheses will be treated as individual characters by subsequent modules and may alter downstream analyses.

The second set of radio buttons (**Figure 12C**) 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. For example, since both elements of the (6/5)AL aptamer must be present within the same sequence, the search for candidate (6/5)AL aptamers would be entered as `RCGUY,RARAC`, and this button would be set to `No`.

The third set of radio buttons (**Figure 12D**) 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 protein sequences or other sequence types. All patterns are converted to uppercase and have white spaces removed regardless of the pattern type.

1. **A/T/G/C/U** - single bases

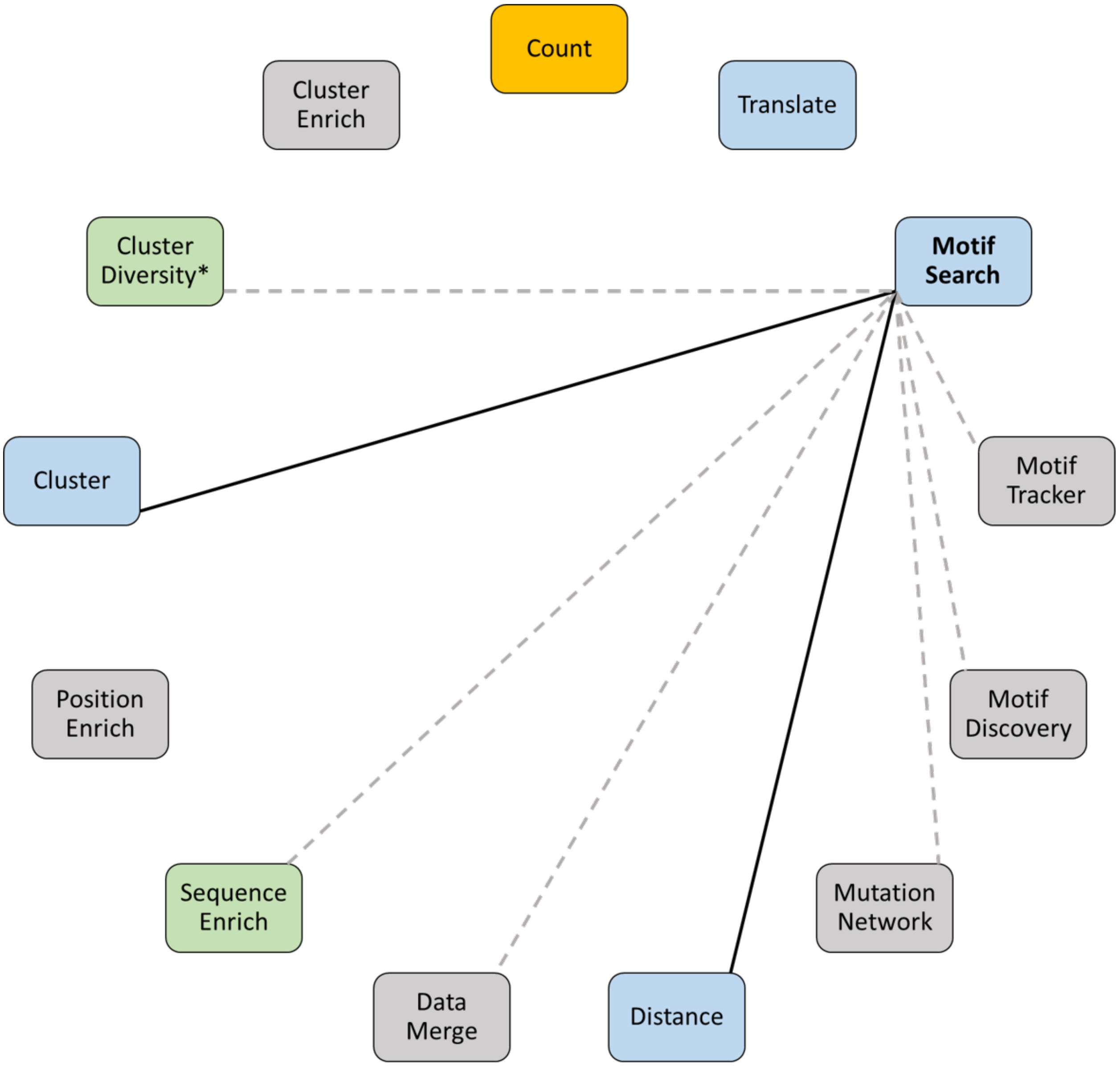
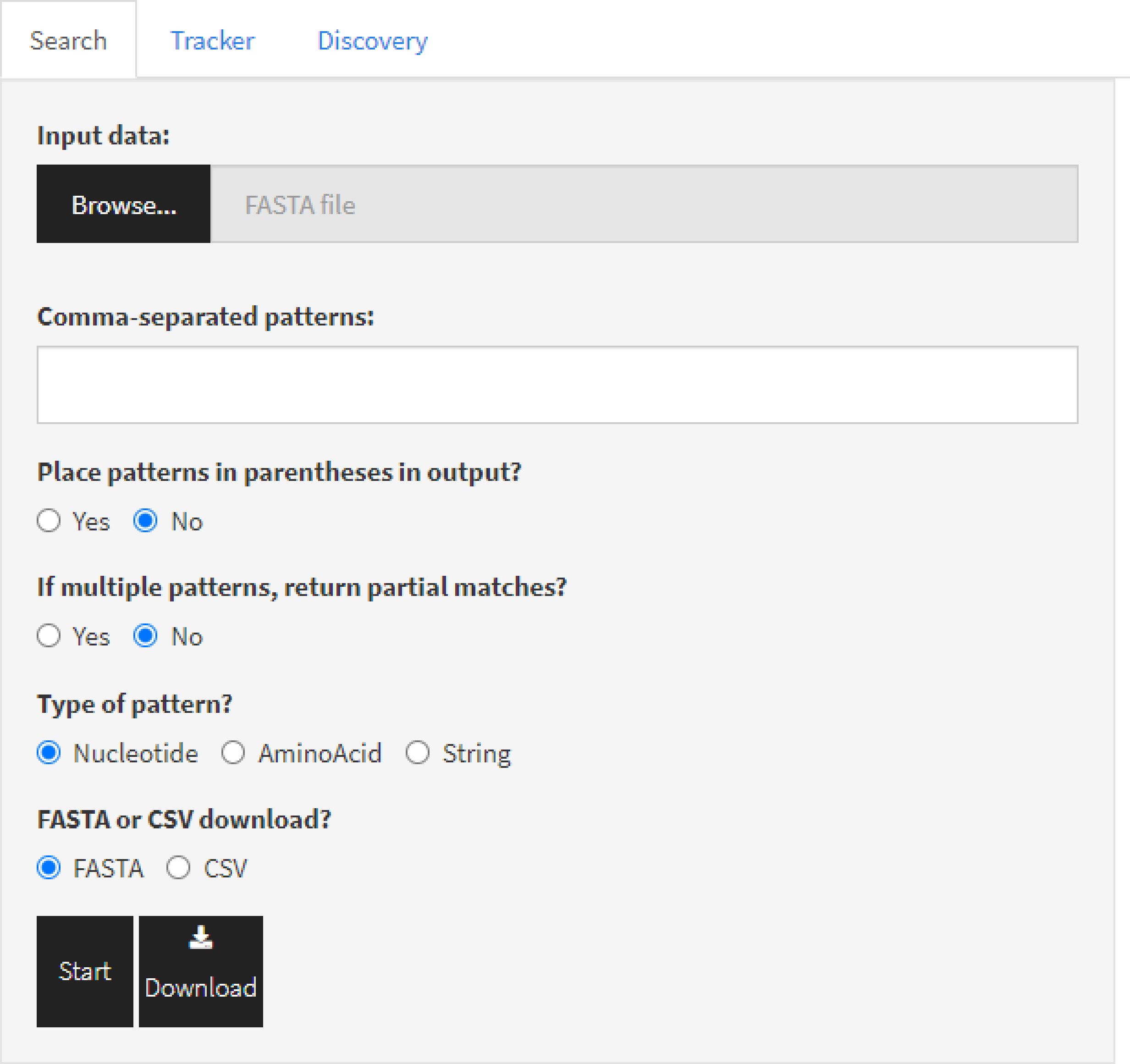


Figure 11: All modules connected to FASTAptameR-Motif_Search.



Input data:

Comma-separated patterns:

Place patterns in parentheses in output?

○ Yes ● No

If multiple patterns, return partial matches?

○ Yes ● No

Type of pattern?

● Nucleotide ○ AminoAcid ○ String

FASTA or CSV download?

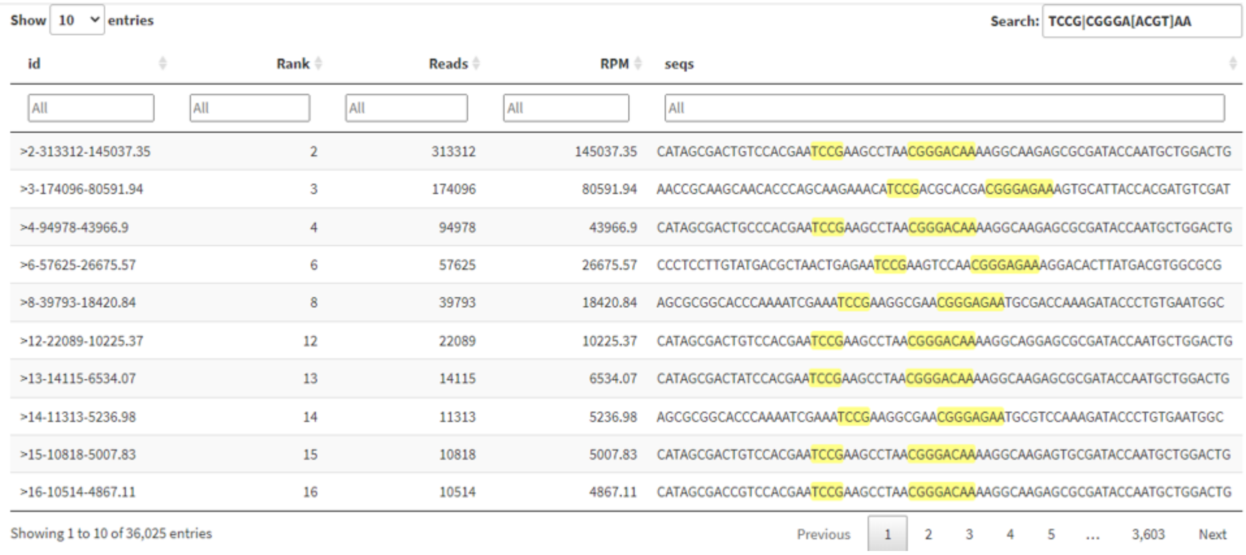
● FASTA ○ CSV

Figure 12: Screenshot of FASTAptameR-Motif_Search.

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 (no *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 **Figure 13** with the following parameters: `comma-separated patterns = UCCG,CGGGAnAA`; `parentheses = No`; `partial filtering = No`; and `pattern type = Nucleotide`.



Show 10 entries

Search: TCCG|CGGGA[ACGT]AA

id	Rank	Reads	RPM	seqs
All	All	All	All	All
>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

Figure 13: FASTAptameR-Motif_Search Output. Motifs are highlighted by default, and the specific patterns that get highlighted are shown in the search box. Omiting the search text from the search bar will remove the highlighting.

3.6 FASTAptameR-Motif_Tracker

3.6.1 Description

Individual species often rise and fall during combinatorial selections as a function of their overall fitness or other evolutionary forces, and they can enrich to different degrees in independently evolving populations. The same is true for sets of species that carry a given sequence motif. Tracking these changes across multiple populations can identify when certain species or sequence motifs first emerge (or disappear) and provides significant insights into their evolutionary dynamics.

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. Sequences that contain the query
4. Rank
5. Reads
6. RPM

Optionally, an alias list of alternative motif/sequences names 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 (*e.g.*, `Seq1`, `Seq2`, `Seq3`; with each alias on its own line).

This module generates two data tables that can be downloaded as CSV files. The first table provides summary statistics for each query, and the second table provides enrichment scores for each query.

All modules directly connected to FASTAptameR-Motif_Tracker are shown in **Figure 14**, and a screenshot of the module interface is shown in **Figure 15**.

3.6.2 Usage

The input FASTA files must be chosen with the file browser. The next line (**Figure 15A**) determines the order of the input files via a dynamically generated dropdown menu. the order in which the user selects the files in this list determines which populations are considered to be population 1, which population 2, *etc.* The following text box (**Figure 15B**) must contain at least one pattern or sequence. If the user wishes to search for multiple patterns or sequences, each pattern or sequence must be entered on separate lines. The text box for aliases (**Figure 15C**) allows the user to rename the motifs or sequences to a more convenient name (such as ‘Motif 1’, ‘G-rich’, or ‘F1Pk’), which will be used in the figure legend for ease of identification. Only one alias must be provided per line.

The first set of radio buttons (**Figure 15D**) determines whether the queries are motifs or sequences, which determines how matches are identified. If `Motif`, then commas in the line are interpreted as separating submotifs, and regex pattern matching is used. For now, commas are interpreted as Boolean `AND` functions. If `Sequence`, then exact matches for each query are returned. The second set of radio buttons (**Figure 15E**) determines the type of query (`DEFAULT = Nucleotide`). If `Nucleotide`, then degenerate nucleotide codes are allowed. Note, the query is converted to uppercase and white spaces are removed regardless of the 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.

A screenshot of a sample output data table from tracked sequences is shown in **Figure 16** with the three most abundant sequences from the 70HRT14 population as input queries. For the example search below, 70HRT14 is population 1 and 70HRT15 is population 2.

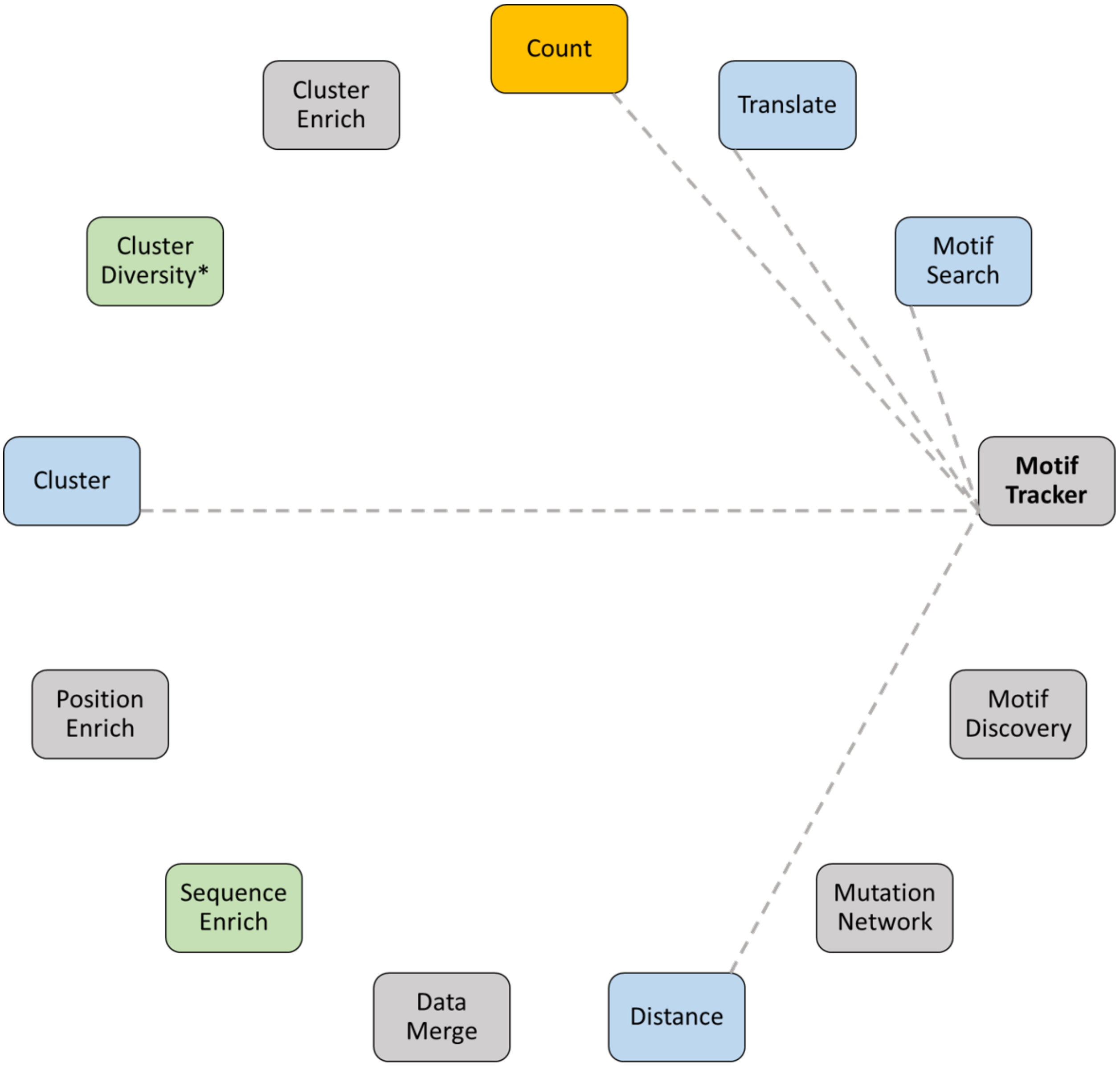
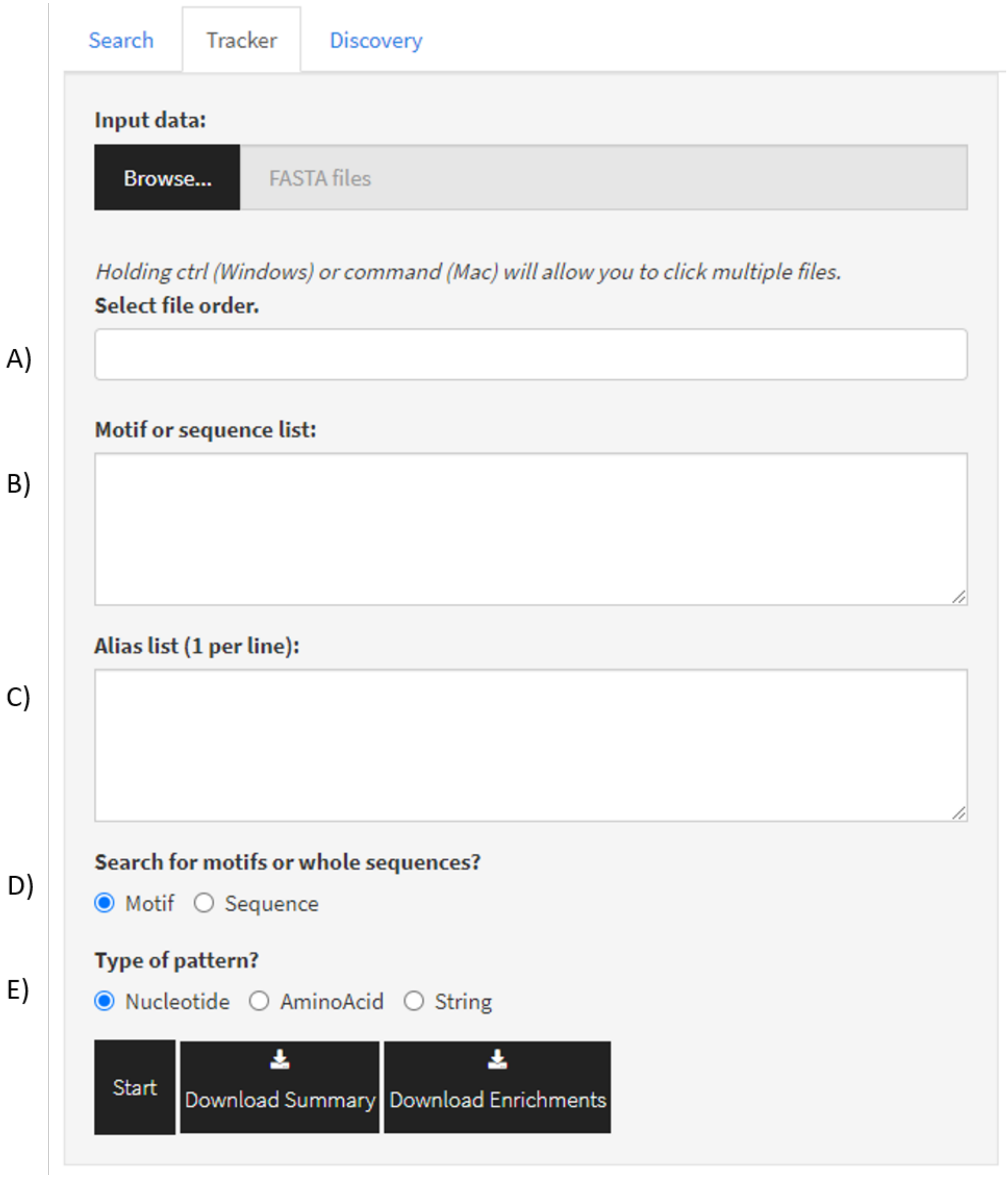


Figure 14: All modules connected to FASTAptameR-Motif_Tracker.



Input data:

FASTA files

Holding ctrl (Windows) or command (Mac) will allow you to click multiple files.

Select file order.

Motif or sequence list:

Alias list (1 per line):

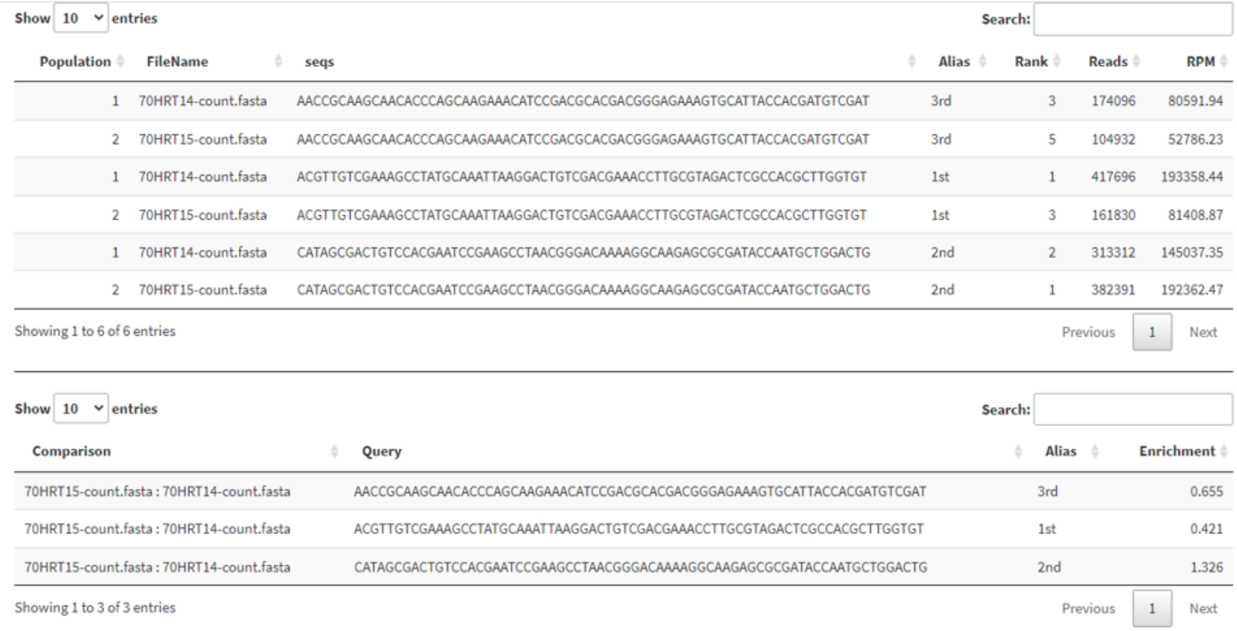
Search for motifs or whole sequences?

● Motif ○ Sequence

Type of pattern?

● Nucleotide ○ AminoAcid ○ String

Figure 15: Screenshot of FASTAptameR-Motif_Tracker.



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

Comparison	Query	Alias	Enrichment
70HRT15-count.fasta : 70HRT14-count.fasta	AACCGCAAGCAACACCCAGCAAGAAACATCCGACGCACGACGGGAGAAAGTGCATTACCACGATGTCGAT	3rd	0.655
70HRT15-count.fasta : 70HRT14-count.fasta	ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCACGCTTGGTGT	1st	0.421
70HRT15-count.fasta : 70HRT14-count.fasta	CATAGCGACTGTCCACGAATCCGAAGCCTAACGGGACAAAAGGCAAGAGCGCGATACCAATGCTGGACTG	2nd	1.326

Showing 1 to 3 of 3 entries

Figure 16: FASTAptameR-Motif_Tracker Output.

3.6.3 Plotting

This module can be used to visualize multiple user-defined motifs or sequences across multiple rounds of selection by generating an interactive line plot showing the RPMs of each query per round (**Figure 17**). Although only two populations and three query sequences were used here for illustration, this tool can be especially useful for observing the rise and fall of multiple query sequences over the course of multiple rounds of selection.

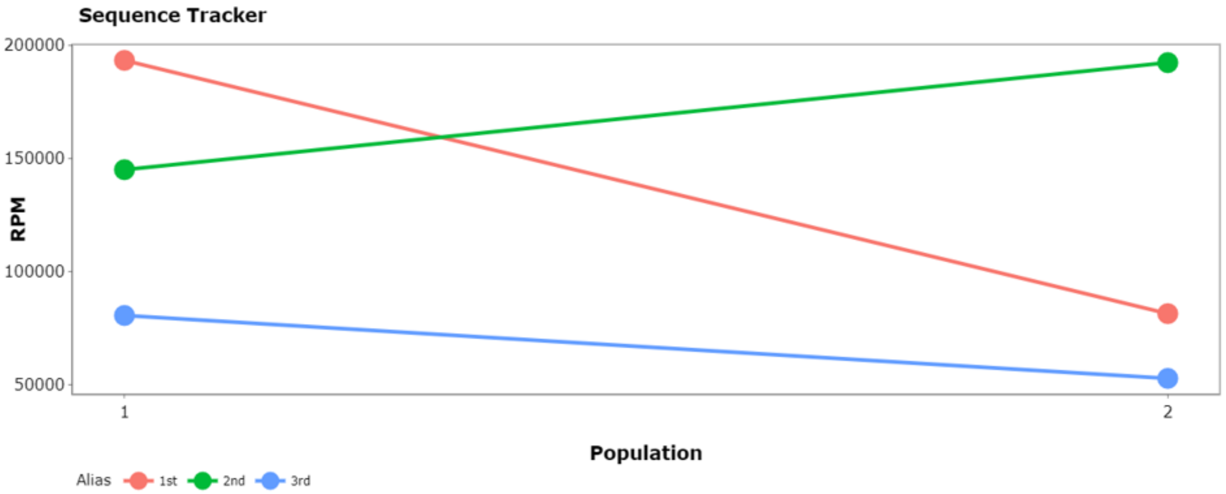


Figure 17: Sequence Tracking Line Plot. Shows the RPM of three sequences across the 70HRT14 and 70HRT15 populations. The aliases - ‘1st’, ‘2nd’, ‘3rd’ - refer to the first, second, and third most abundant sequences from the 70HRT14 population.

3.7 FASTAptameR-Motif_Discovery

3.7.1 Description

Combinatorial selections often converge upon one or more sequence or structural motifs that are present in many otherwise unrelated members of the population. FASTAptameR-Motif_Discovery is intended to provide a preliminary assessment of shared sequence motifs via its implementation of the Fast String-Based Clustering (FSBC) algorithm (Kato et al. 2020) for *de novo* discovery of **contiguous**, over-enriched motifs in D/RNA sequences. This module accepts a *counted* FASTA file with no degenerate codes as input and returns a data table that contains columns for over-enriched motifs and their associated p-value (P), normalized Z-score (ZZ), motif length (Motif_Length), and rank by normalized Z-score (Rank).

Please note that no motif discovery tool perfectly handles every situation. For example, FSBC is a **sequence-based** motif discovery tool that does not consider discontinuous motifs, degenerate codes, or structural motifs. Further, this method currently only supports D/RNA sequences, but we are working to generalize it for alternative alphabets, such as protein or XNA sequences, and to allow motif discovery strategies that are informed by predicted 2D/3D structures and/or by physicochemical properties. We are also interested in exploring the idea of cluster-specific alignments to identify sequence motifs in a future version of FASTAptameR.

Additionally, we note that there are many other excellent tools dedicated to *de novo* motif discovery, such as the MEME suite (Bailey et al. 2015) for sequence-based approaches and Infernal (Nawrocki and Eddy 2013) for RNA using both sequence-based and predicted structural similarities. FASTA-formatted output from any of the FASTAptameR 2.0 modules can be exported and analyzed by those other dedicated platforms.

All modules directly connected to FASTAptameR-Motif_Discovery are shown in **Figure 18**, and a screenshot of the module interface is shown in **Figure 19**.

3.7.2 Usage

The input FASTA file must be chosen with the file input browser. While a *counted* FASTA file is the minimum requirement for this module, we recommend supplying a single cluster (generated with FASTAptameR-Cluster) to reduce the module run-time. The following set of radio buttons determines whether over-enriched motifs should be found from the set of unique sequences (button set to `No`) or from the set of all unique

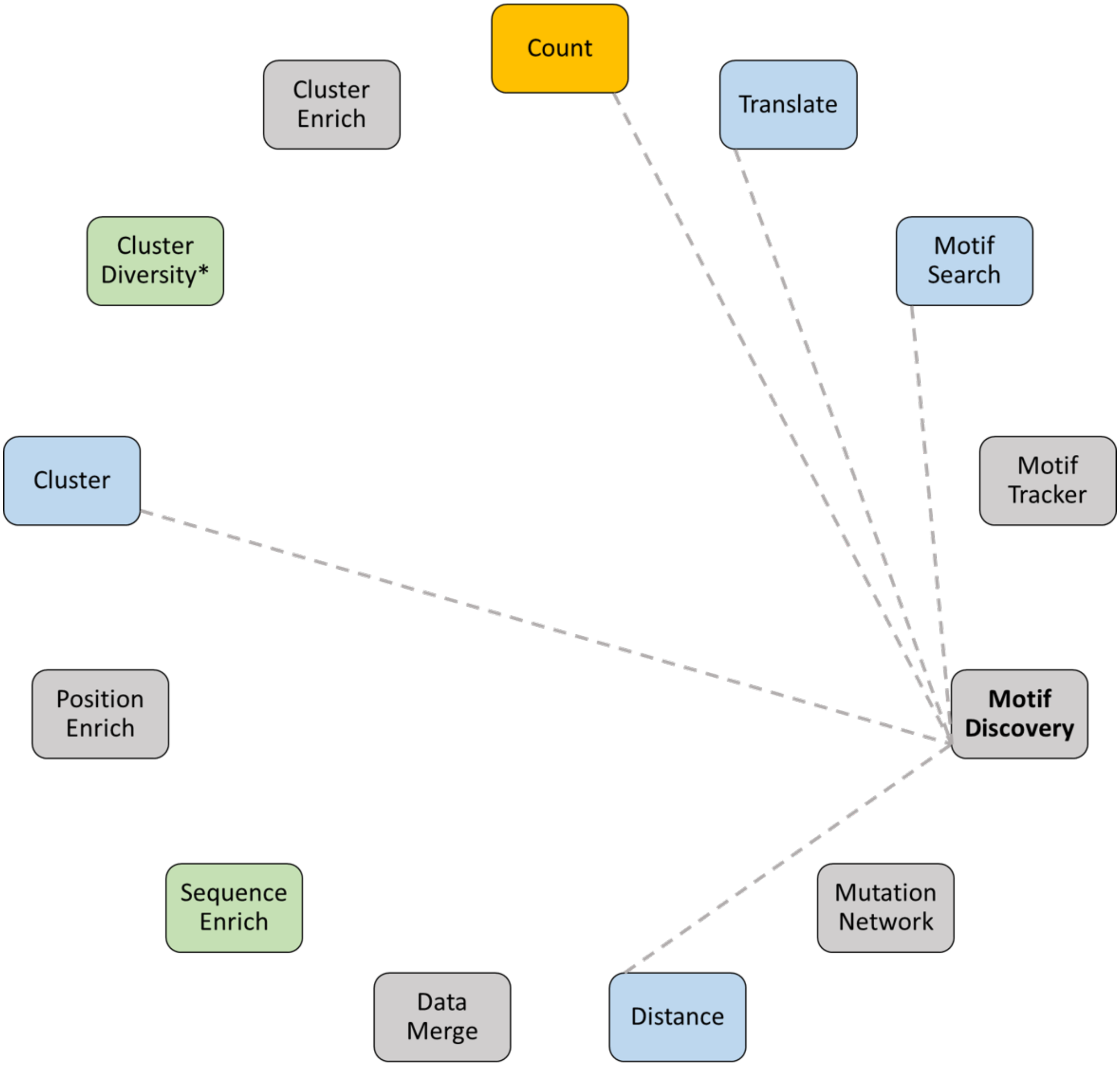
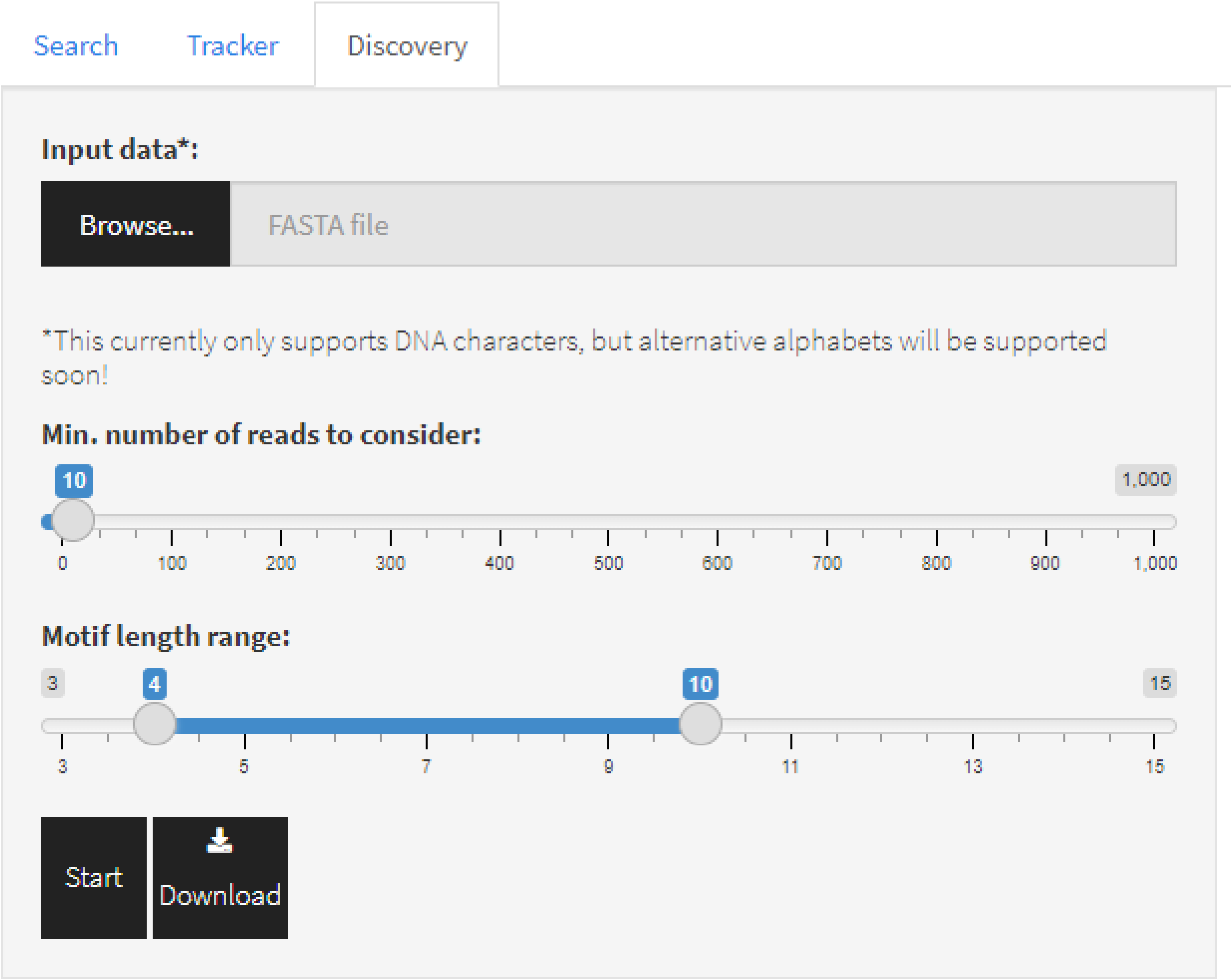


Figure 18: All modules connected to FASTAptameR-Motif_Discovery.



Input data*:

*This currently only supports DNA characters, but alternative alphabets will be supported soon!

Min. number of reads to consider: 10 (range 0 – 1,000)

Motif length range: 4 – 10 (range 3 – 15)

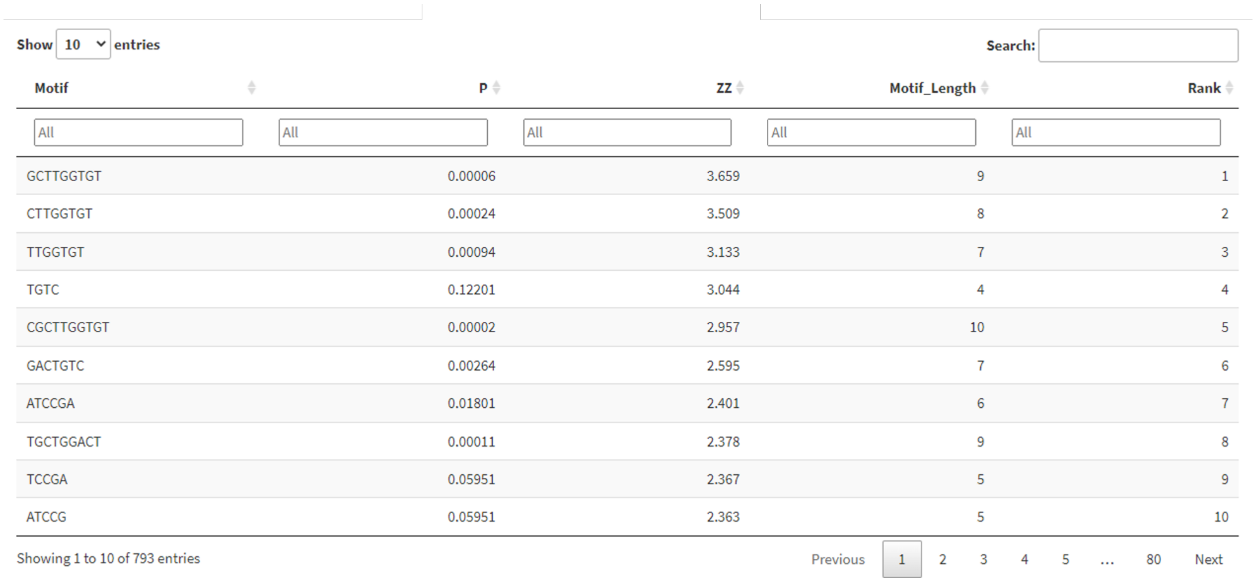
Figure 19: Screenshot of FASTAptameR-Motif_Discovery.

sequences weighted by their respective read counts (button set to `Yes`). The slider bar sets the range of motif lengths for which the algorithm will search. As shown in **Figure 19**, this run would search for motifs with a length between 4-10 (inclusive).

To improve the runtime of this module, we recommend that the user apply the read filter to their data prior to running the algorithm.

The `Start` button begins the *de novo* motif discovery. 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 CSV file.

A sample output data table is shown in **Figure 20**, which was run with the first cluster of the 70HRT14 population.



Motif	P	ZZ	Motif_Length	Rank
All	All	All	All	All
GCTTGGTGT	0.00006	3.659	9	1
CTTGGTGT	0.00024	3.509	8	2
TTGGTGT	0.00094	3.133	7	3
TGTC	0.12201	3.044	4	4
CGCTTGGTGT	0.00002	2.957	10	5
GACTGTC	0.00264	2.595	7	6
ATCCGA	0.01801	2.401	6	7
TGCTGGACT	0.00011	2.378	9	8
TCCGA	0.05951	2.367	5	9
ATCCG	0.05951	2.363	5	10

Showing 1 to 10 of 793 entries

Figure 20: FASTAptameR-Motif_Discovery Output.

3.7.3 Plotting

This module also generates a bubble plot where the x-axis corresponds to the rank by normalized Z-score (ZZ), the y-axis corresponds to `-log10(p-value)`, and both color and size are used to indicate the length of the associated motif. To interpret this plot, we recommend looking at the motifs according to the x-axis (rank by normalized z-score). An example plot is shown in **Figure 21**.

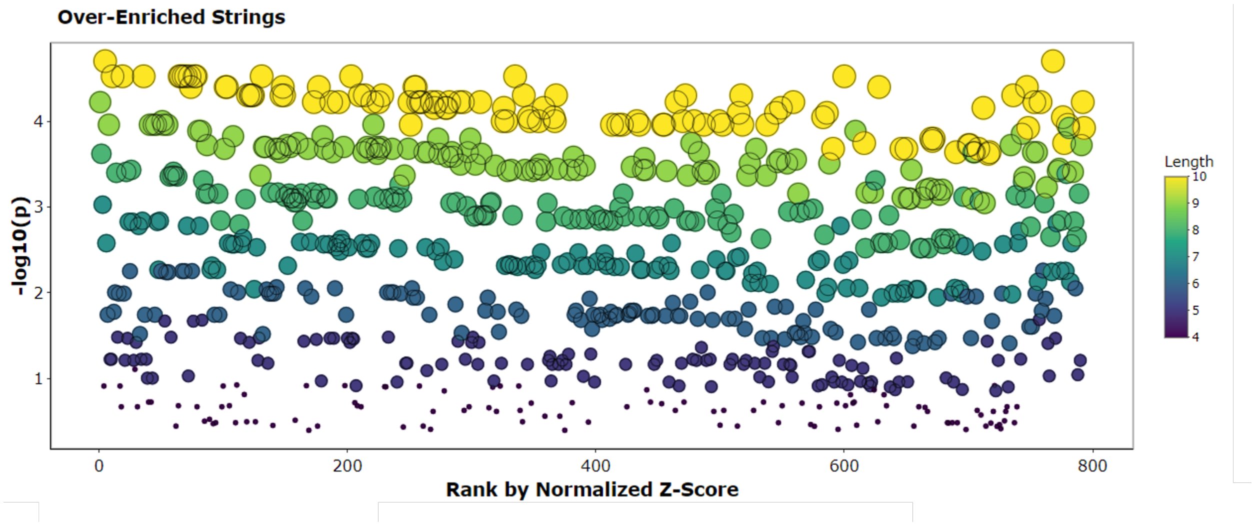


Figure 21: Bubble plot from FASTAptameR-Motif_Discovery. This serves as a visual interpretation of this module's results. Discovered motifs are ranked according to their normalized Z-score.

3.8 FASTAptameR-Mutation_Network

3.8.1 Description

Fitness landscapes and evolutionary histories can sometimes be evaluated by looking at the rise and fall of mutational intermediates during selection. FASTAptameR-Mutation_Network uses an implementation of Dijkstra's Shortest Path Algorithm to find the shortest mutational path between two query sequences in a population. The maximum number of mutations per evolutionary step can be defined by the user, allowing for incremental steps (*e.g.*, only one mutation per step) or larger steps. This module accepts a *counted* FASTA file as input and returns a data table summarizing each step of the path. Specifically, there is one row per step in the pathway, and the columns correspond to the current sequence, the nearest sequence, and the LED to get there.

All modules directly connected to FASTAptameR-Mutation_Network are shown in **Figure 22**, and a screenshot of the module interface is shown in **Figure 23**.

3.8.2 Usage

The input FASTA file must be chosen with the file input browser. Like in FASTAptameR-Motif_Discovery, a *counted* FASTA file is the minimum requirement for this module, but we recommend supplying a single cluster (generated with FASTAptameR-Cluster) to improve the module run-time. The following two text boxes accept the "start" and "end" sequences for the search; these sequences will be added to the uploaded sequence set if they are not already found in it. Finally, the slider determines the maximum allowable cost for a single mutational step in the pathway. For example, when the slider is set to 1, each step must be within 1 edit of the previous sequence. When the slider is set to 3, each step must be within 1-3 edits of the previous sequence.

It is possible that the algorithm fails to find a path given the sequence set and the maximum number of edits allowed between consecutive steps. This possibility returns an error message: "The minimal path for the given constraints is not represented in the dataset!". It is possible that increasing the maximum number of edits will lead to a mutational path, or it is possible that fundamental intermediates are just not present in the dataset. In this latter case, no change to the maximum number of edits will lead to a viable path being returned.

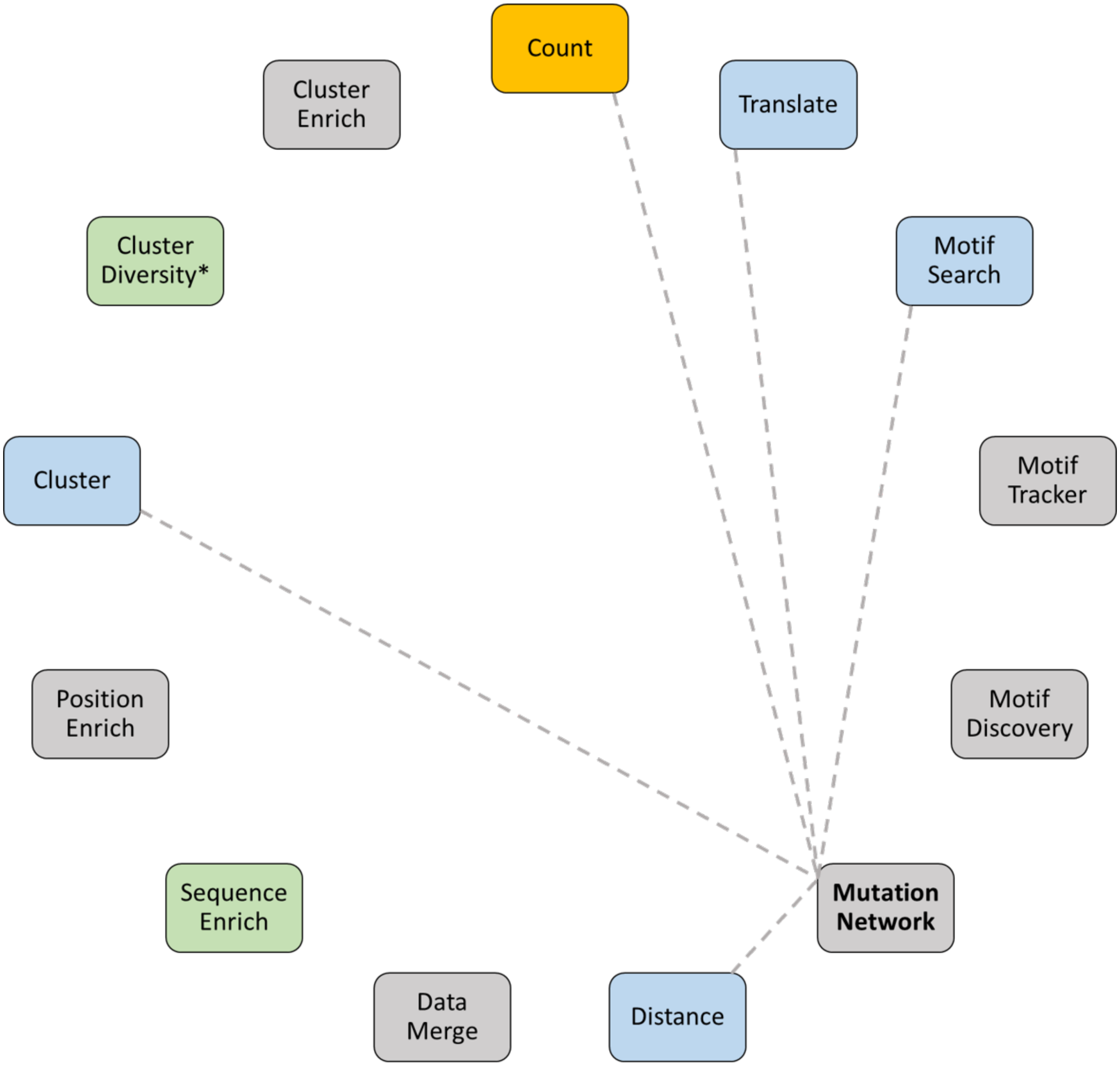


Figure 22: All modules connected to FASTAptameR-Mutation_Network

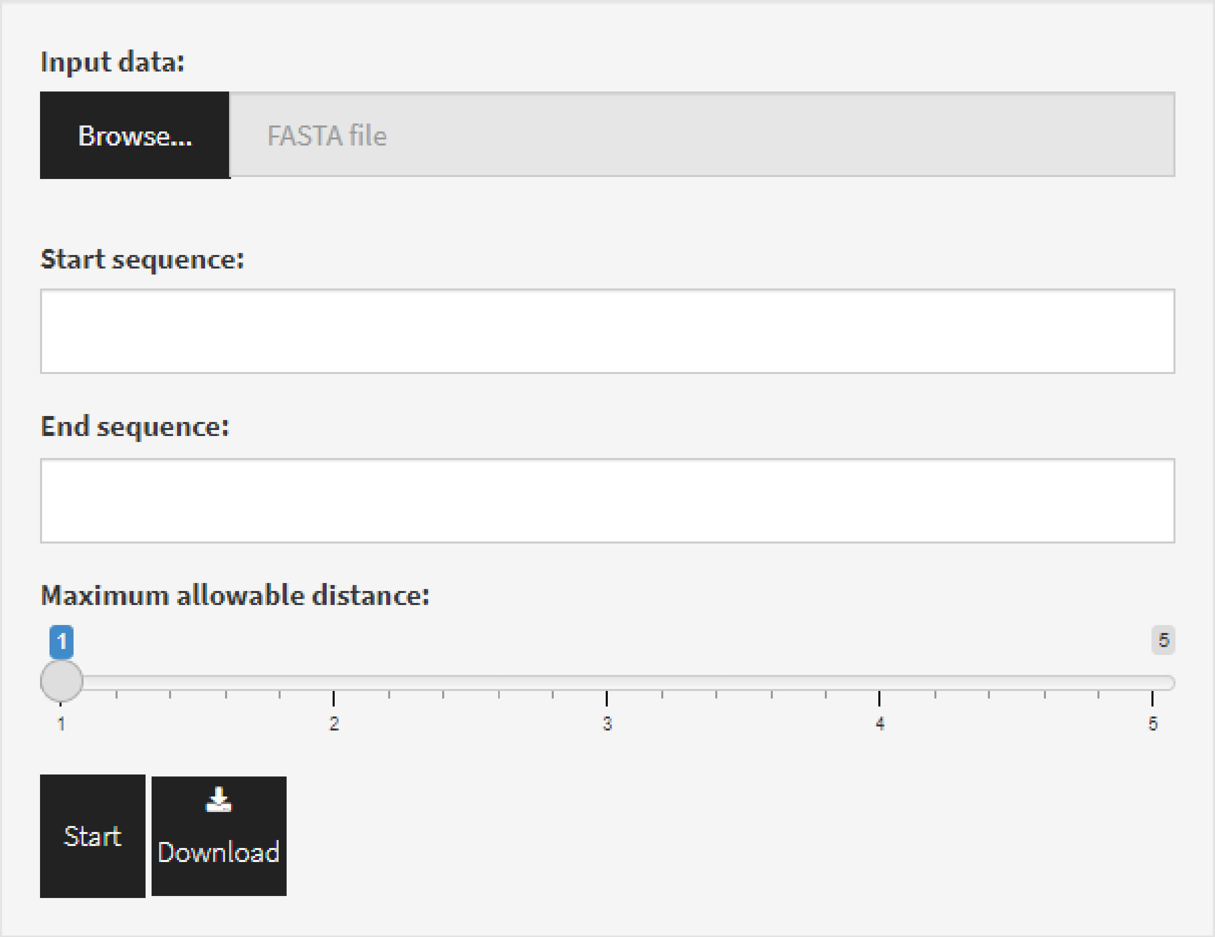
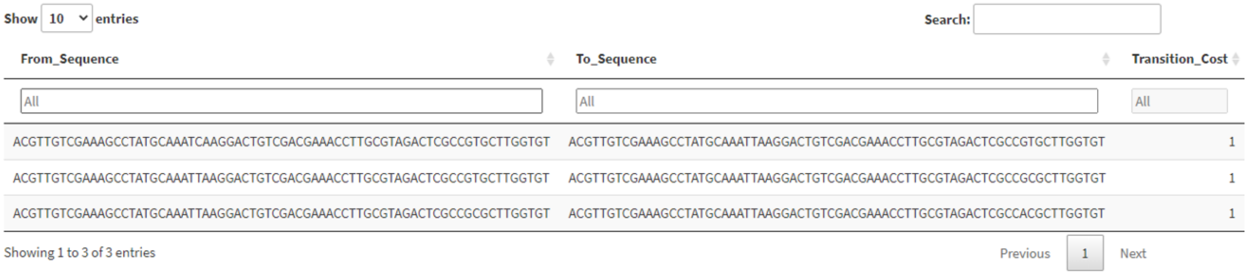


Figure 23: Screenshot of FASTAptameR-Mutation_Network

To help the runtime of this algorithm, we recommend that the user supply relatively small sequence populations.

The `Start` button begins the search for the shortest path between the two user-defined sequences. 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 CSV file.

A sample output data table is shown in **Figure 24**, which was run with the first cluster of the 70HRT14 population.



From_Sequence	To_Sequence	Transition_Cost
ACGTTGTCGAAAGCCTATGCAAATCAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCGTGCTTGGTGT	ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCGTGCTTGGTGT	1
ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCGTGCTTGGTGT	ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCGCGCTTGGTGT	1
ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCGCGCTTGGTGT	ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCACGCTTGGTGT	1

Showing 1 to 3 of 3 entries

Figure 24: FASTAptameR-Mutation_Network Output.

3.9 FASTAptameR-Distance

3.9.1 Description

The distribution of edit distances from a given “seed” sequence to the rest of the population informs population structure, evolutionary dynamics, and fitness landscapes. For example, a tight grouping of closely related sequences may reflect low mutation rates, rugged fitness landscapes, or strong purifying selective pressures. Alternatively, high diversity within a cluster may be the result of high mutation rates, smooth fitness landscapes, or weak selective pressures. A wide separation between that cluster and the rest of the population may indicate wide-spaced fitness peaks or sparsely sampled sequence space at the outset of the selection. In contrast, intermediate or closely spaced peaks may indicate that the starting population sampled sequence space more densely and that species selected for one fitness peak may be able to acquire the necessary mutations to sample other fitness peaks. Finally similar analyses can help to establish diversity or mutational density for a starting library prior to a selection.

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 LED is the minimum number of substitutions, insertions, or deletions required to transform one sequence into another. In that sense, it is more general than the Hamming distance, which only considers the minimum number of substitutions required to transform one sequence to another sequence of equal length. The output can be downloaded as a FASTA or CSV file.

All modules directly connected to FASTAptameR-Distance are shown in **Figure 25**, and a screenshot of the module interface is shown in **Figure 26**.

3.9.2 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.

The slider bar (**Figure 26A**) 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. Thus, it is recommended that the starts of the sequences (5’ ends, N-termini, *etc.*) are aligned.

A sample output data table is shown in **Figure 27** with the following query sequence (the most abundant sequence from the 70HRT14 data set):

`ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCACGCTTGGTGT`.

3.9.3 Plotting

This module can also generate interactive histograms of distances (button shown in **Figure 26B**, output shown in **Figure 28**). The top plot corresponds to the distances between the query and *all unique sequences*, which allows greater visibility for low-abundance sequences. In contrast, the bottom plot corresponds to the distances between the query and all sequences, factoring in their read counts. In both cases, the query sequence is displayed at the top of the plot.

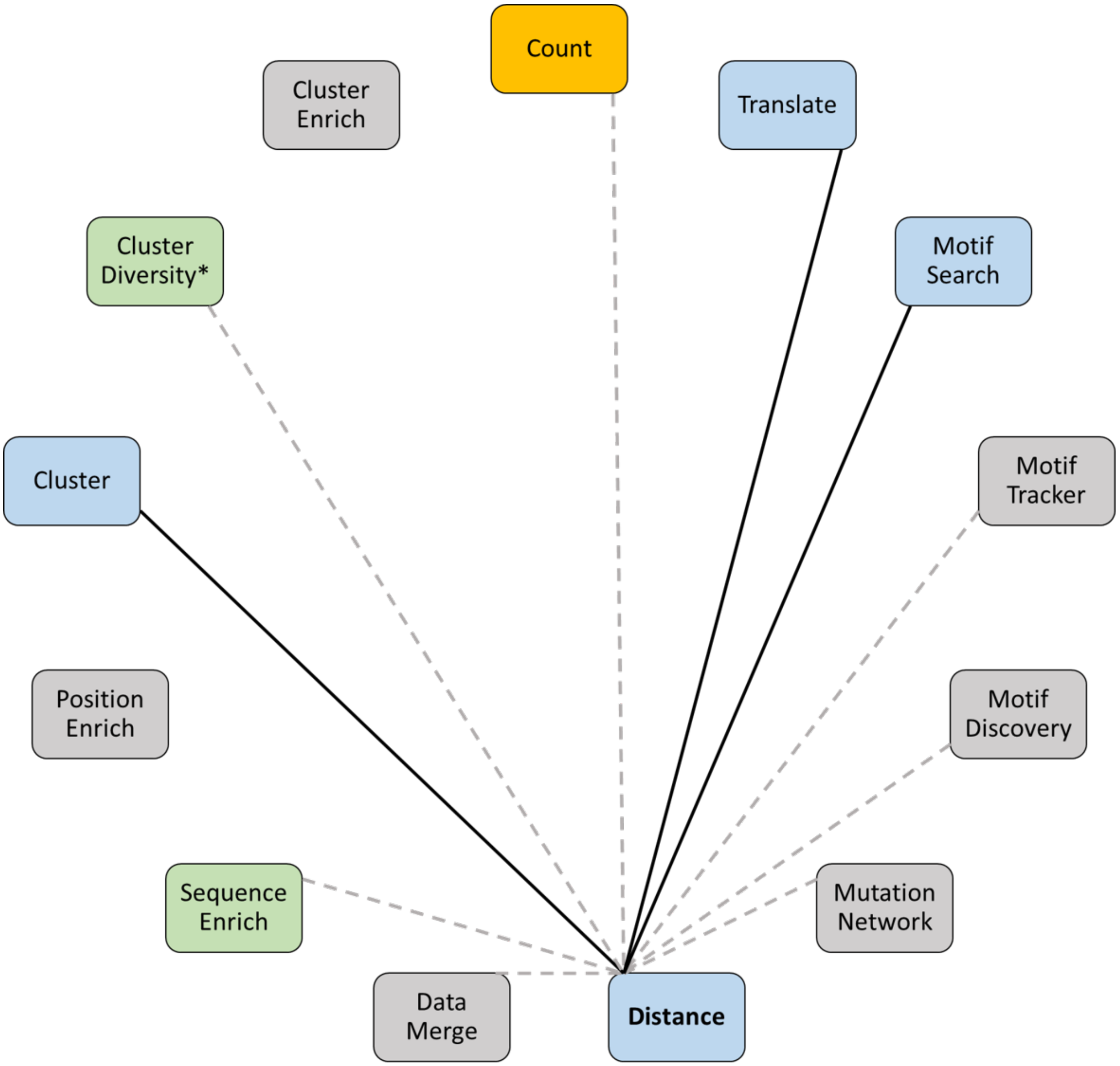
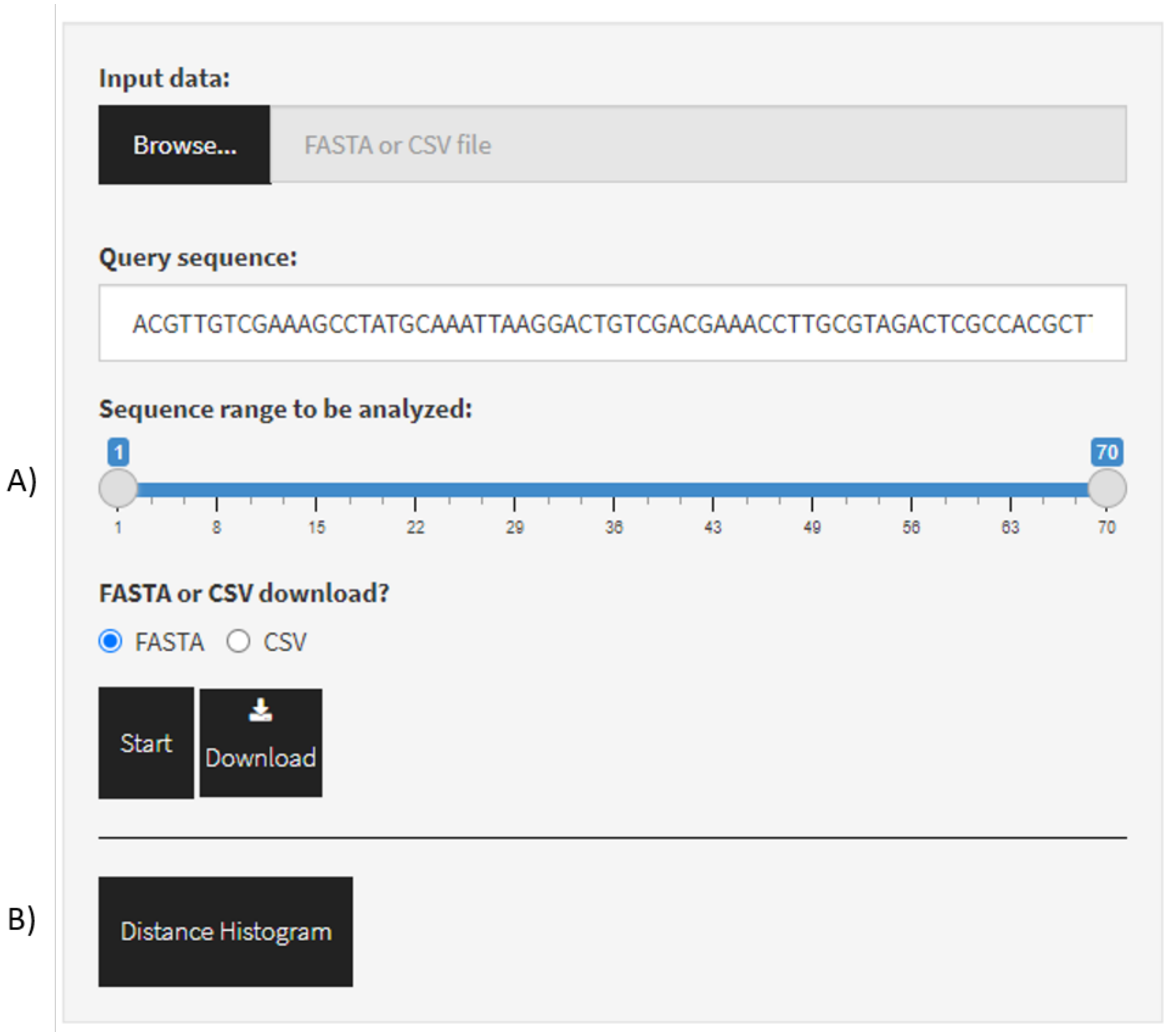


Figure 25: All modules connected to FASTAptameR-Distance.



Input data:

FASTA or CSV file

Query sequence:

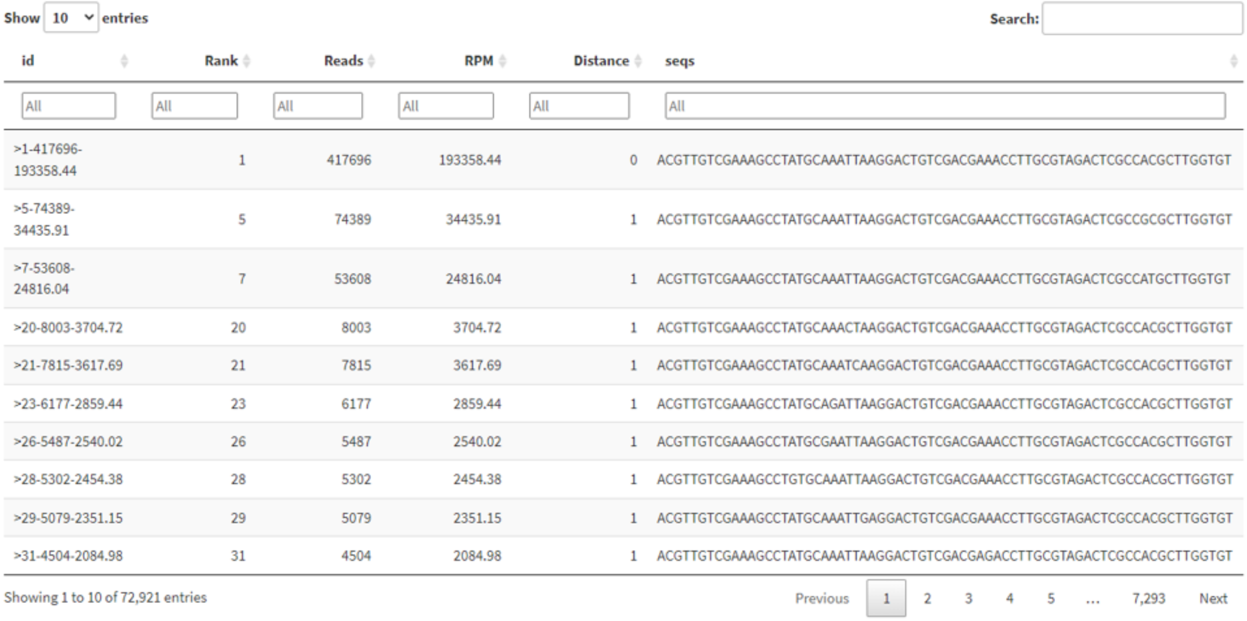
ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCACGCT

Sequence range to be analyzed: 1 – 70

FASTA or CSV download?

● FASTA ○ CSV

Figure 26: Screenshot of FASTAptameR-Distance.



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

Showing 1 to 10 of 72,921 entries

Figure 27: FASTAptameR-Distance Output.

3.10 FASTAptameR-Data_Merge

3.10.1 Description

To identify overlaps between two or more populations, FASTAptameR-Data_Merge accepts two or more *counted* FASTA files as input and returns a merged data table (downloadable as a CSV) that contains sequence metadata from each uploaded population. Currently supported merges include union, intersection, and left join. The **union** returns all sequences from every uploaded population. The **intersection** returns all sequences shared between every uploaded population. The **left join** returns one row for each sequence in the first population with columns from other populations appended to it. A left join is analogous to bringing a grocery list to several stores and recording the price of *only* the objects of the list, disregarding every other item in the store. Similarly, a left join between two populations (`A` and `B`) returns a table with one row for every sequence in `A` and appends 1) metadata for `B` if the sequence is shared or 2) nothing if the sequence is not found in `B`.

All modules directly connected to FASTAptameR-Data_Merge are shown in **Figure 29**, and a screenshot of the module interface is shown in **Figure 30**.

3.10.2 Usage

The input FASTA files must be chosen with the file browser and ordered with the dropdown menu. The following radio buttons determine if the data will be merged with a union, intersection, or left join. The `Start` button begins the merge, and 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.

A sample output data table is shown in **Figure 31** after merging 70HRT14 and 70HRT15 with set intersection.

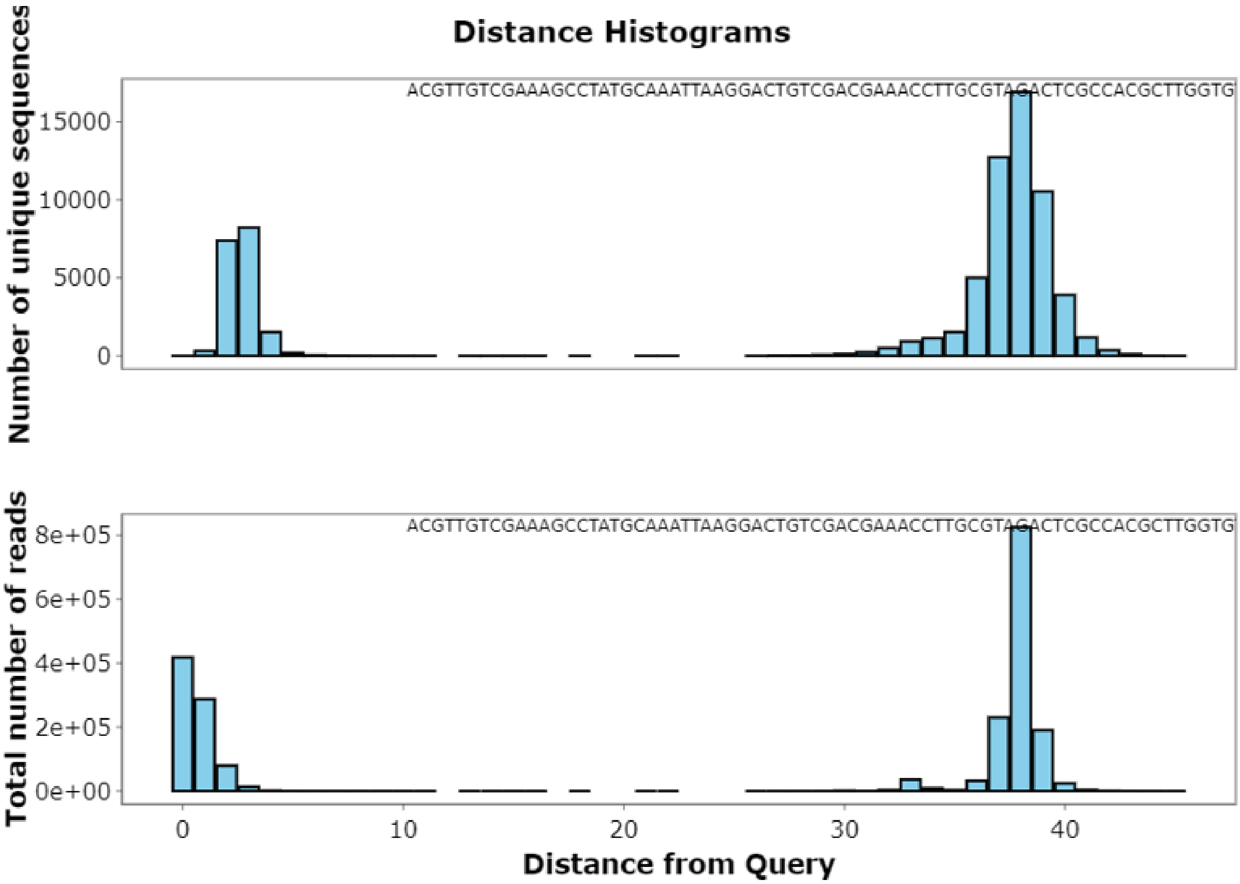


Figure 28: Distance Histograms with the 70HRT14 population as the target and the most abundant sequence as the query. A) Distance histogram with only unique sequences. B) Distance histogram with all reads. In both plots, the set of peaks on the left indicate the abundances of all sequences within a short edit distance of the seed sequence (that set constitutes a “cluster”, see **section 3.10**), while the large gap between group and the rest of the population reflects the dissimilarities between any two molecules with 70 random positions.

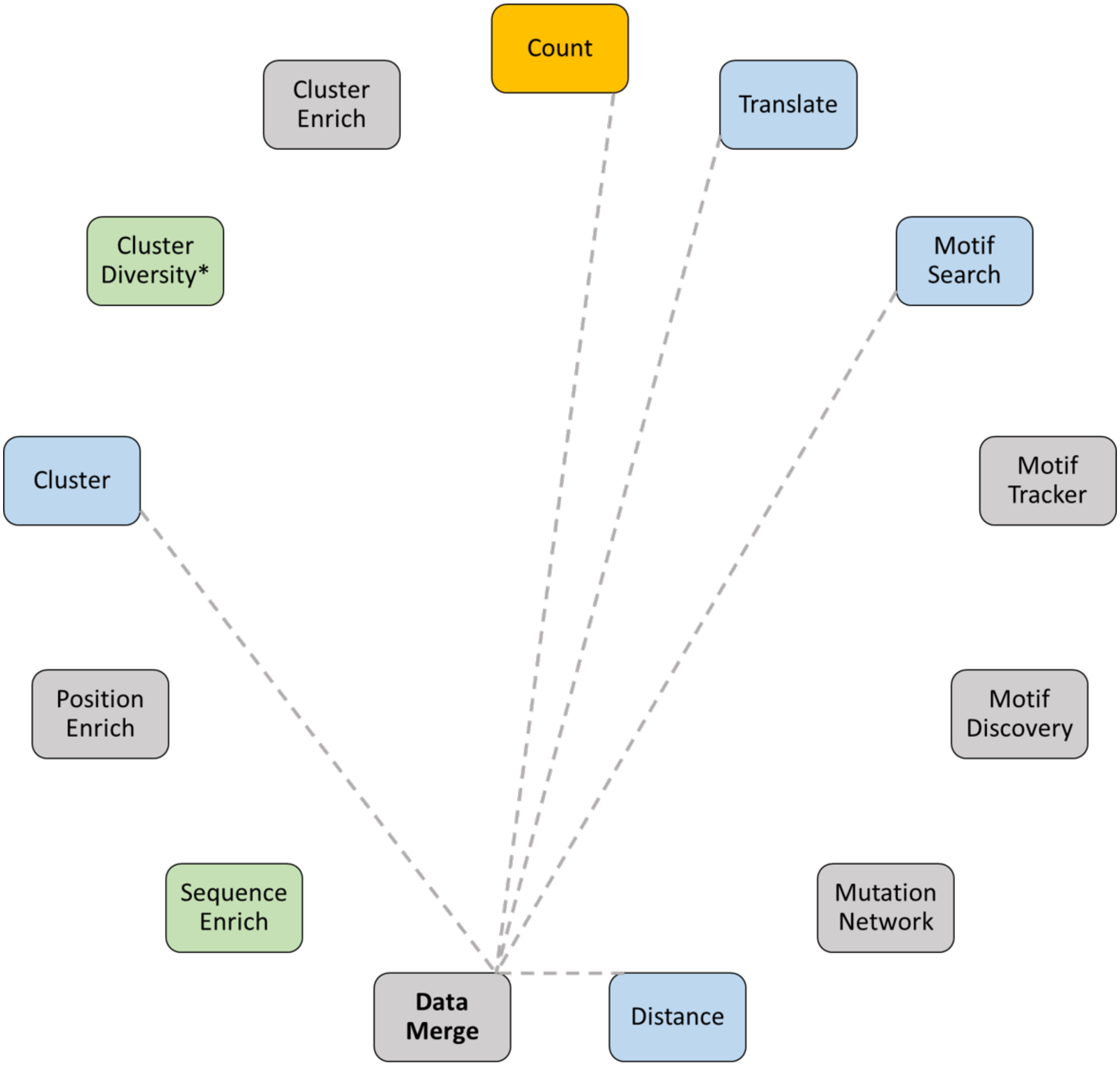
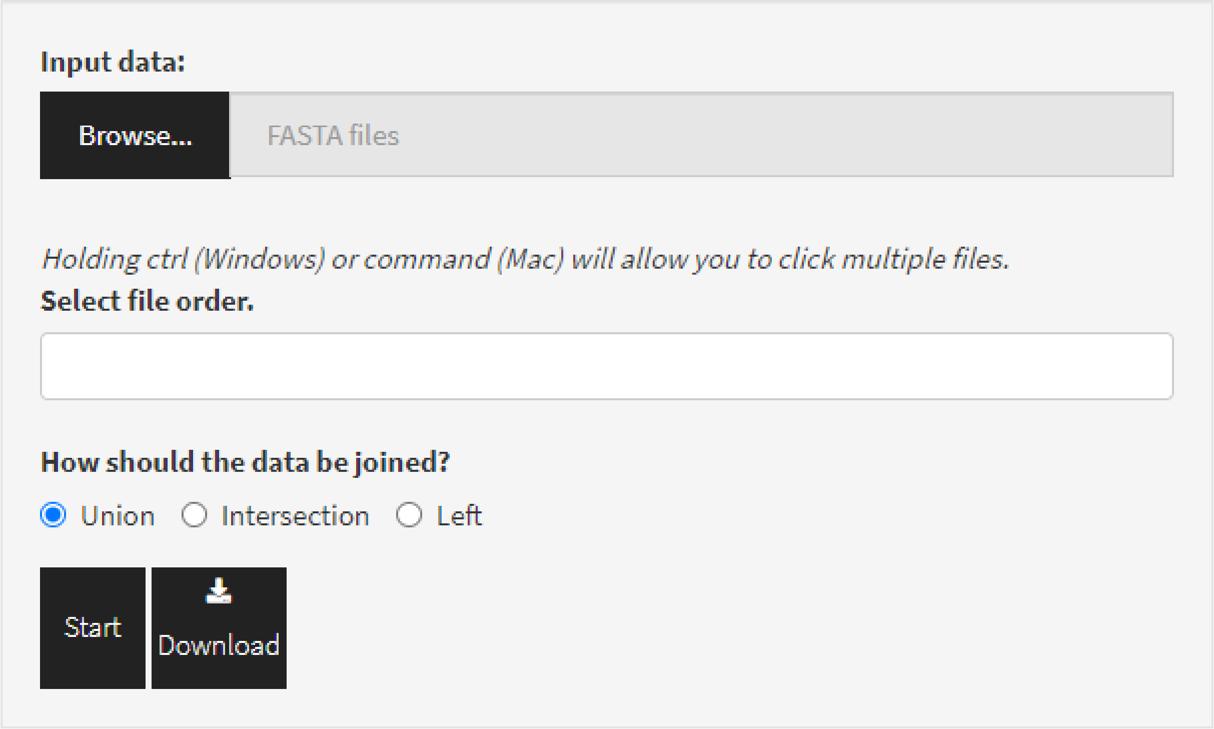


Figure 29: All modules connected to FASTAptameR-Data_Merge



Input data:

Browse... FASTA files

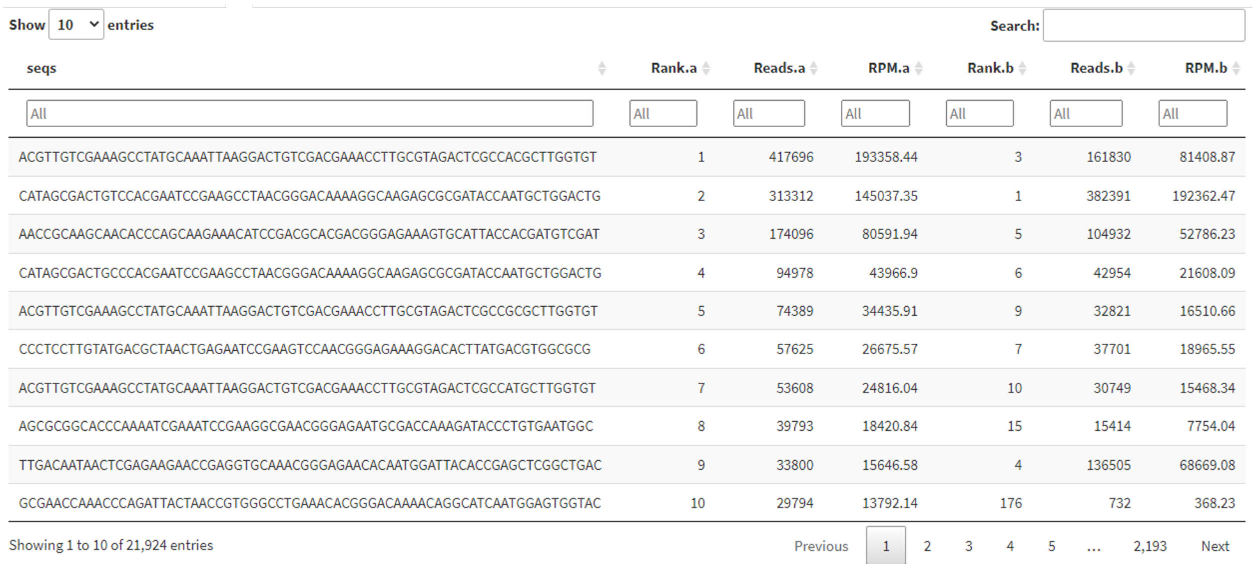
Holding ctrl (Windows) or command (Mac) will allow you to click multiple files.

Select file order.

How should the data be joined?

● Union ○ Intersection ○ Left

Figure 30: Screenshot of FASTAptameR-Data_Merge



Show 10 entries Search:

seqs	Rank.a	Reads.a	RPM.a	Rank.b	Reads.b	RPM.b
All	All	All	All	All	All	All
ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCACGCTTGGTGT	1	417696	193358.44	3	161830	81408.87
CATAGCGACTGTCCACGAATCCGAAGCCTAACGGGACAAAAGGCAAGAGCGCGATACCAATGCTGGACTG	2	313312	145037.35	1	382391	192362.47
AACCGCAAGCAACACCCAGCAAGAAACATCCGACGCACGACGGGAGAAAGTGCATTACCACGATGTCGAT	3	174096	80591.94	5	104932	52786.23
CATAGCGACTGCCCACGAATCCGAAGCCTAACGGGACAAAAGGCAAGAGCGCGATACCAATGCTGGACTG	4	94978	43966.9	6	42954	21608.09
ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCGCGCTTGGTGT	5	74389	34435.91	9	32821	16510.66
CCCTCCTTGTATGACGCTAACTGAGAATCCGAAGTCCAACGGGAGAAAGGACACTTATGACGTGGCGCG	6	57625	26675.57	7	37701	18965.55
ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCATGCTTGGTGT	7	53608	24816.04	10	30749	15468.34
AGCGCGGCACCCAAAATCGAAATCCGAAGGCGAACGGGAGAATGCGACCAAAGATACCCTGTGAATGGC	8	39793	18420.84	15	15414	7754.04
TTGACAATAACTCGAGAAGAACCGAGGTGCAAACGGGAGAACACAATGGATTACACCGAGCTCGGCTGAC	9	33800	15646.58	4	136505	68669.08
GCGAACCAAACCCAGATTACTAACCGTGGGCCTGAAACACGGGACAAAACAGGCATCAATGGAGTGGTAC	10	29794	13792.14	176	732	368.23

Showing 1 to 10 of 21,924 entries

Figure 31: FASTAptameR-Data_Merge Output.

3.10.3 Plotting

This module can also generate an interactive sequence persistence bar plot and a static UpSet plot (**Figure 32**). The sequence persistence plot (**Figure 32, top**) shows the number of unique sequences present in either 70HRT14 or 70HRT15 (~90,000 sequences) and the number of unique sequences shared between 70HRT14 and 70HRT15 (~20,000 sequences).

The horizontal bars of the UpSet plot (**Figure 32, bottom**) show the total number of sequences in each corresponding population. The vertical bars show the results of two types of set operations. Vertical bars associated with single "dots" in the matrix (1st and 2nd vertical bars) show the results of a **set difference** operation: the number of sequences in one population that are not found in the other population (~50,000 sequences for 70HRT14 and ~40,000 sequences for 70HRT15). Vertical bars associated with multiple "dots" in the matrix (3rd bar) show the results of a **set intersection** operation: the number of sequences shared between all implicated populations (~20,000 sequences are shared between 70HRT14 and 70HRT15). Thus, the UpSet plot can be thought of as a higher-resolution version of the sequence persistence plot.

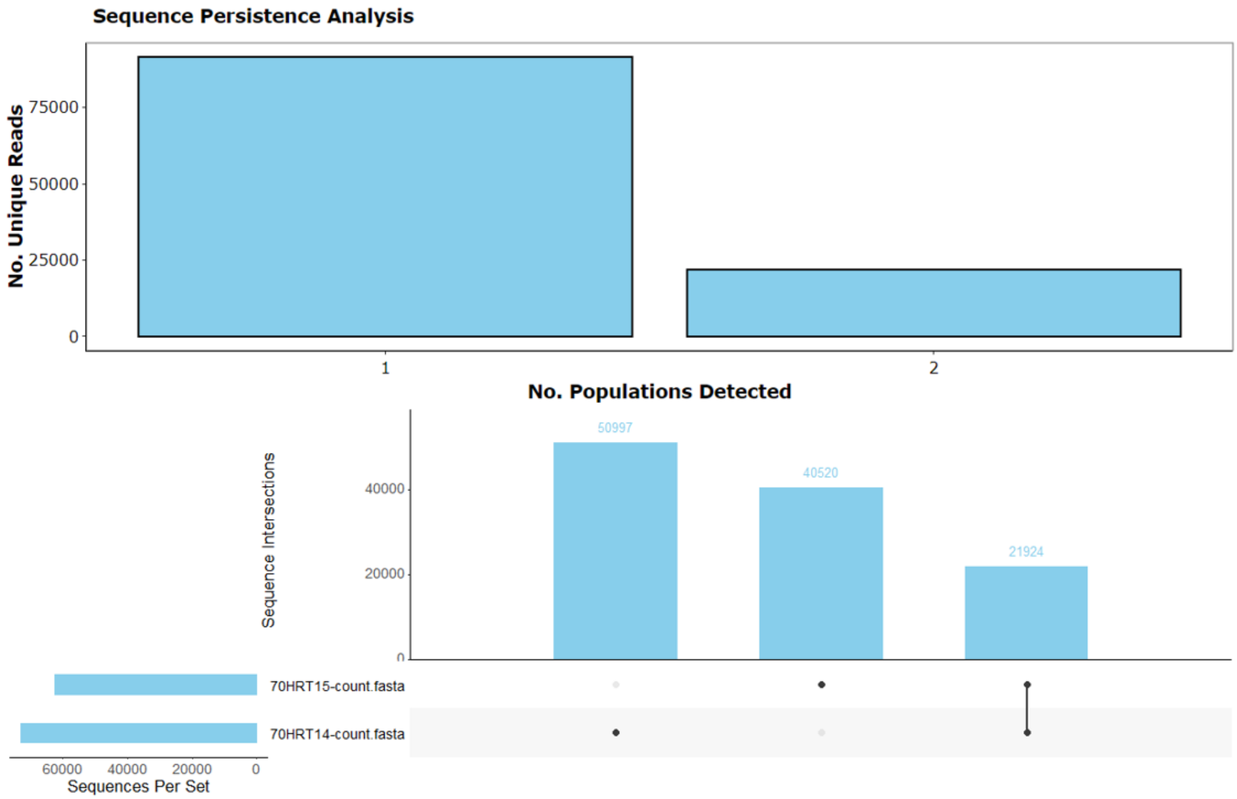


Figure 32: FASTAptameR-Data_Merge plots after merging 70HRT14 and 70HRT15.

3.11 FASTAptameR-Sequence_Enrich

3.11.1 Description

The degree to which a given sequence enriches (or depletes) during a selection - or along different branches of a divergent evolution - is a powerful indicator of the relative fitness of that sequence. Enrichment analysis can be applied to the population as a whole or to subsets of closely related sequences (within-cluster).

FASTAptameR-Sequence_Enrich calculates the enrichment (or depletion) of each sequence in one population relative to one other population. The module takes 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* and *.b*, which depends on the order in which the files are selected by the dropdown menu. Additional columns include enrichment scores (`E = RPM2 / RPM1`) and the base-two logarithm of the Enrichment ("log2(E)" = `log2(Enrichment)`). This output can be downloaded as a CSV.

All modules directly connected to FASTAptameR-Sequence_Enrich are shown in **Figure 33**, and a screenshot of the module interface is shown in **Figure 34**.

3.11.2 Usage

The input FASTA files must be chosen with the file browser. The dropdown menu immediately under the file input browser is used to select the order of files for the analysis. 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, such as when a sequence enriches from below the detection limit to above the detection limit.

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 without the filters, simply 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 output data table that does not include missing sequences is shown in **Figure 35**.

3.11.3 Plotting

This module can generate four types of interactive plots (**Figure 36, 37**): $\log_2(Enrichment)$ histograms (**Figure 36A**), RPM scatter plots (**Figure 36B**), ratio average (RA) plots (**Figure 36C**), and a cluster box plot in the case when **clustered** FASTA files are provided as input (**Figure 37**).

The spread of the $\log_2(Enrichment)$ histogram (**Figure 36A**) relative to a vertical line at `x = 0` can indicate the overall magnitude 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 and displacement of the RPM scatter plot (**Figure 36B**) with respect to the diagonal line at `y = x` can also indicate the magnitudes of enrichment and possible directionality. Finally, the RA plot (**Figure 36C**) is used to show the relationship between the average log-RPM and $\log_2(Enrichment)$ for each sequence. Note that missing sequences (those with `RPM = 0` in one round) are treated as having `RPM = 0.1` for the sake of calculating their log2 values.

The cluster box plot (**Figure 37**) shows the distribution of enrichment values for sequences after first grouping by cluster. The 25th and 75th quartiles are respectively represented by the bottom and top of each box. The line in the middle of the box represents the median. Whiskers are at most `1.5 * IQR`, and any points beyond that are shown as outliers. The red marker indicates where the seed sequence of the cluster falls. Individual points that are well above or well below the median represent species that are enriching or

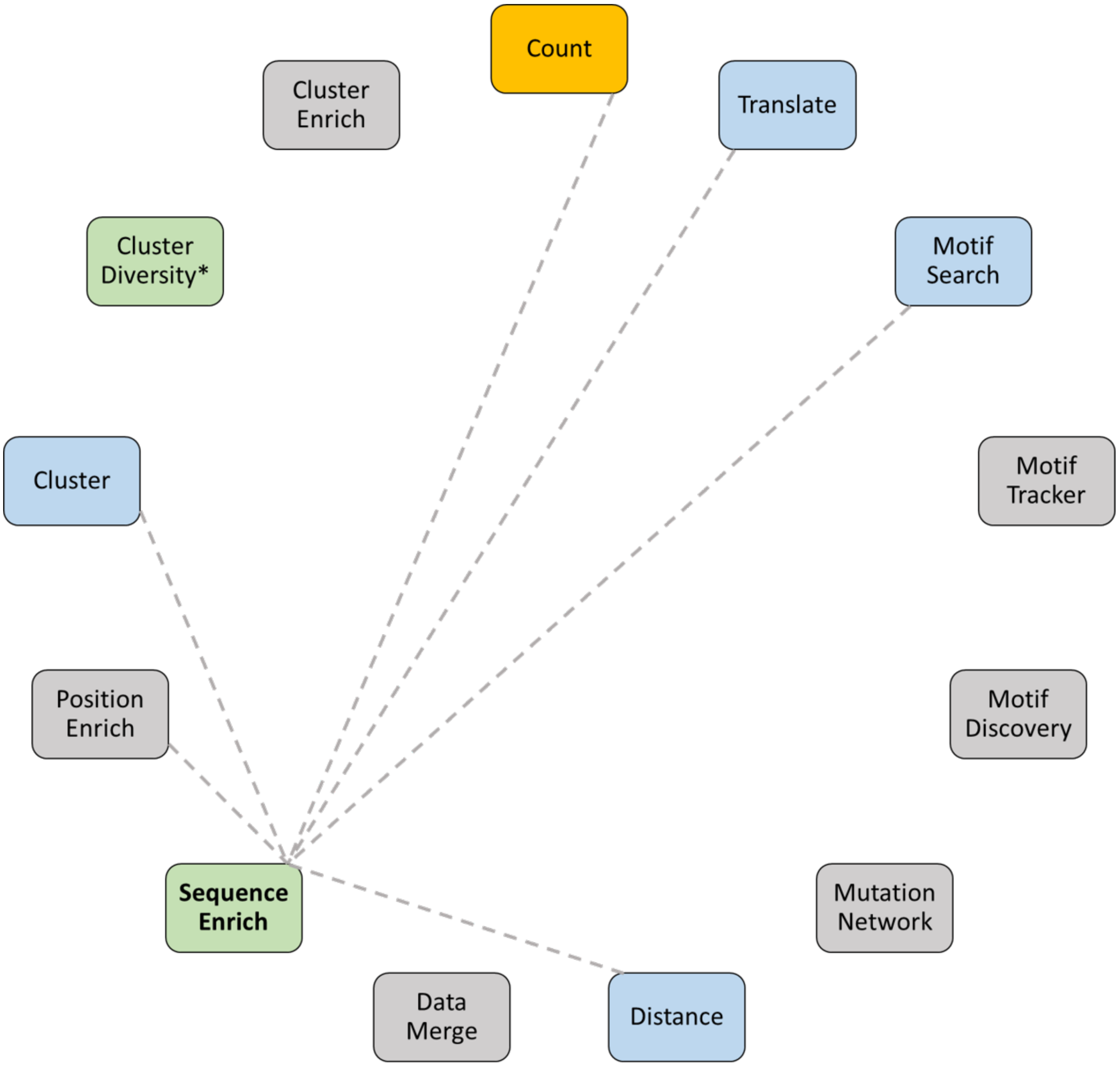
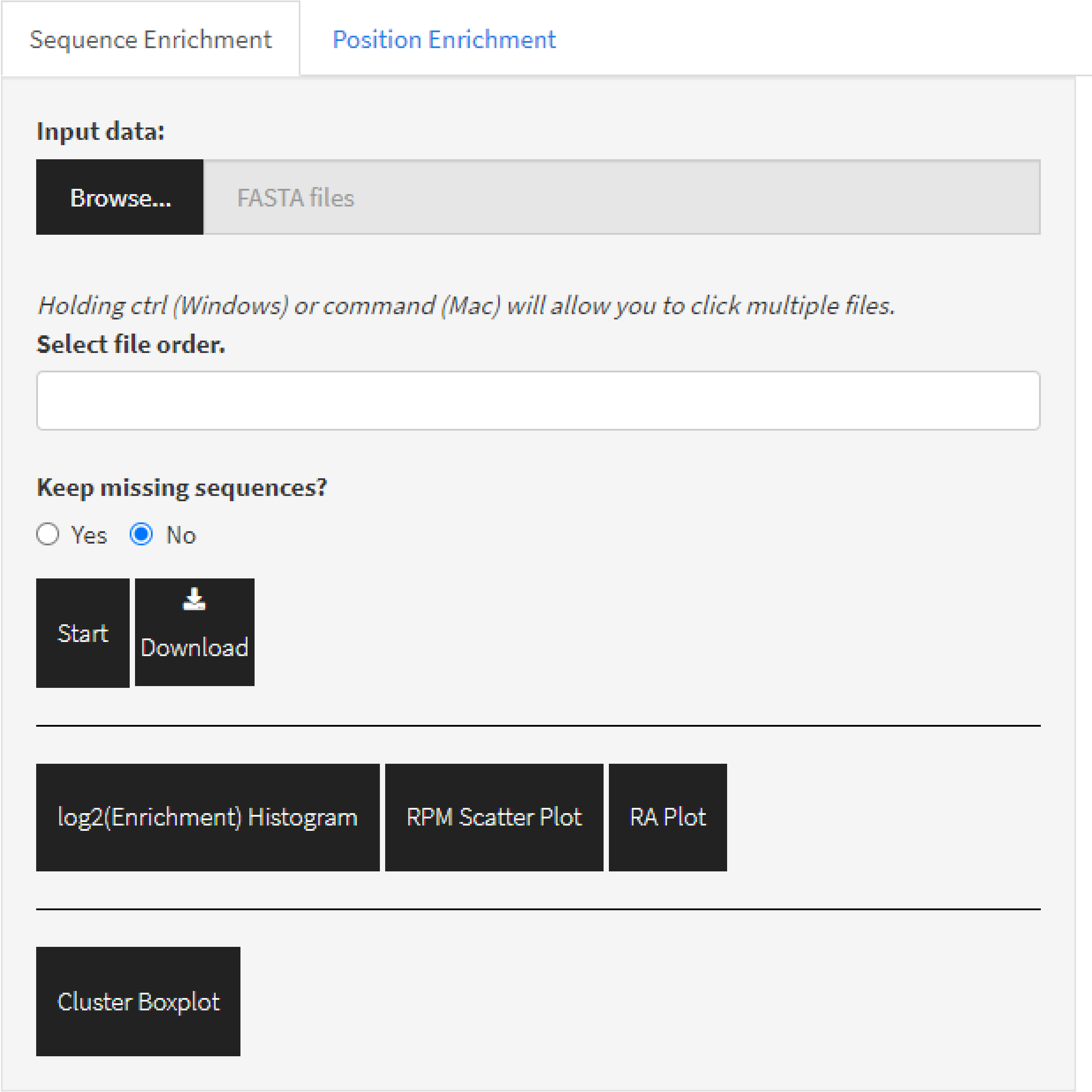


Figure 33: All modules connected to FASTAptameR-Sequence_Enrich.



Input data:

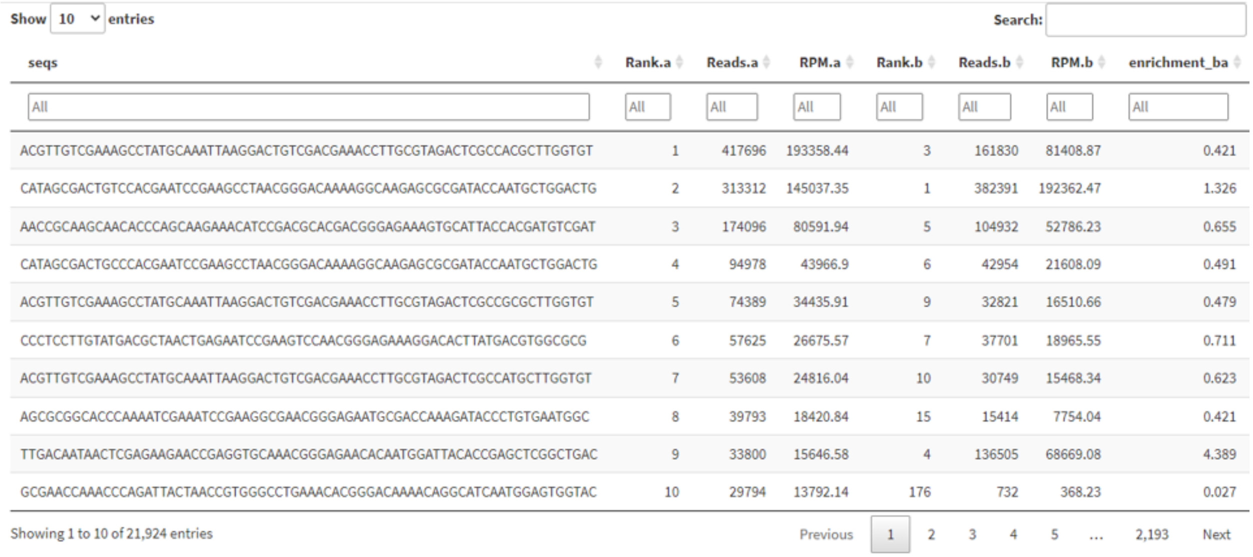
Holding ctrl (Windows) or command (Mac) will allow you to click multiple files.

Select file order.

Keep missing sequences?

○ Yes ● No

Figure 34: Screenshot of FASTAptameR-Sequence_Enrich.



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

Showing 1 to 10 of 21,924 entries

Figure 35: FASTAptameR-Sequence_Enrich Output.

depleting relative to the cluster as a whole. Both types of outliers can be highly informative; species with strongly advantageous variations may be emerging as future dominant species for that cluster, while species with strongly disadvantageous variations can illuminate critical portions of the biomolecule.

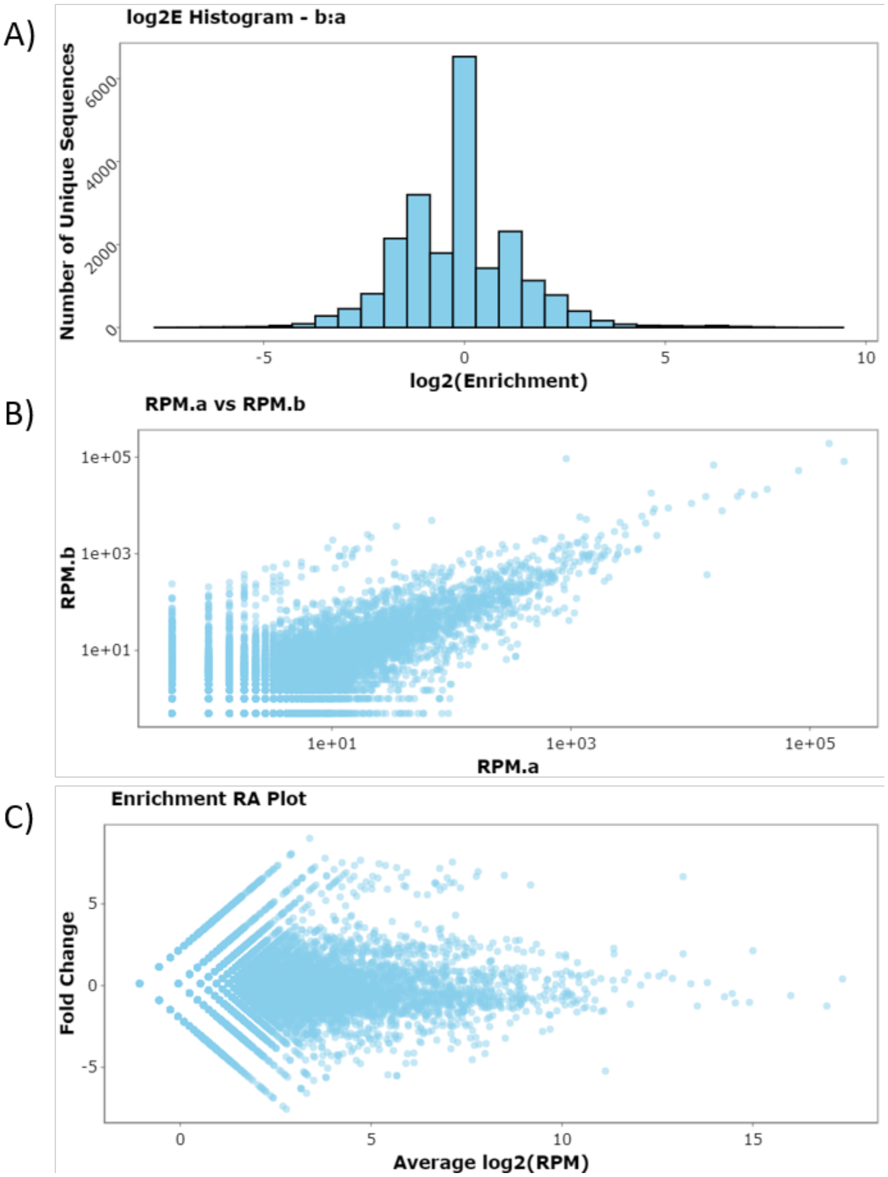
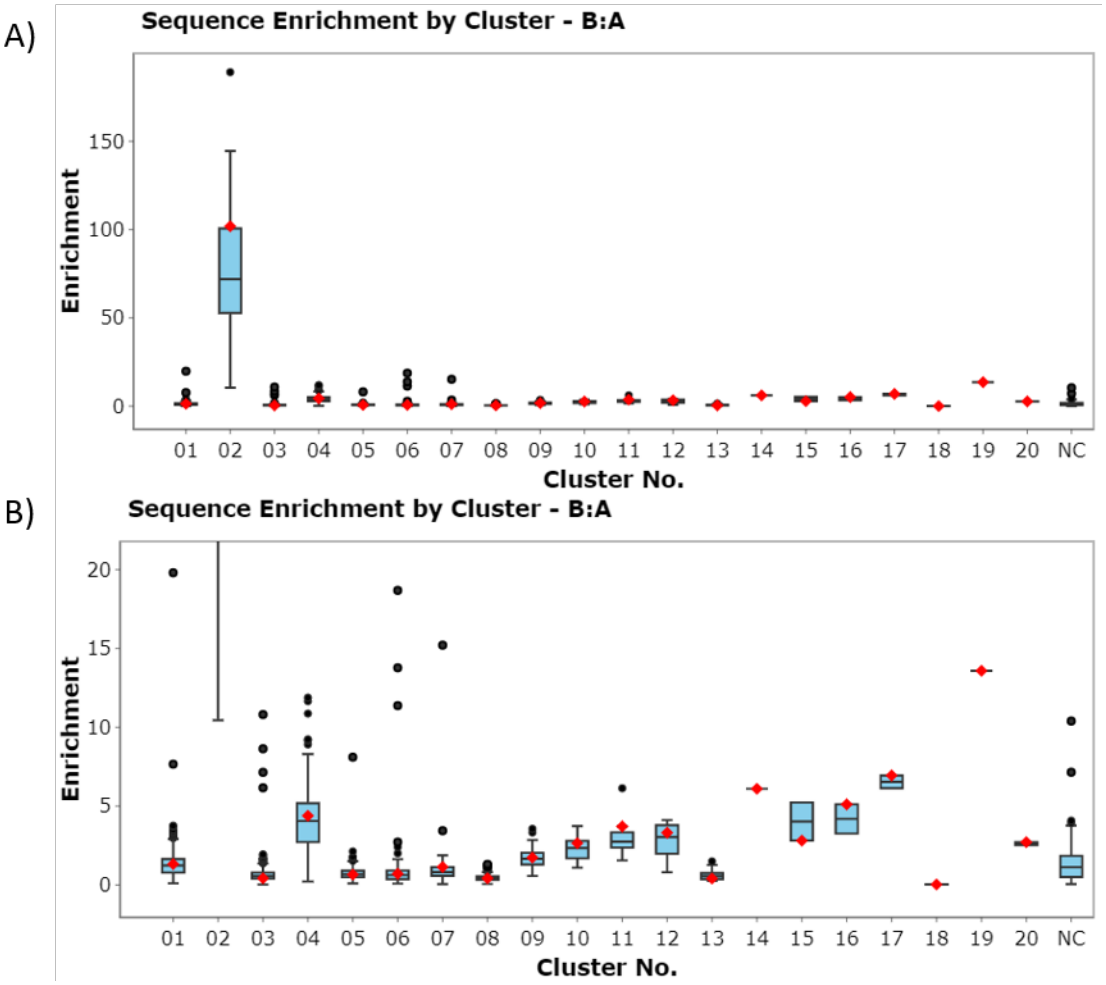


Figure 36: FASTAptameR-Sequence_Enrich Plots. A) The histogram shows the distribution of fold-change between rounds. B) The RPM scatter plot shows the RPM of sequences across two rounds. C) The RA plot has fold change on the y-axis (`R` for Ratio) and average log2(RPM) on the x-axis (`A` for average). A small cloud of enriching sequences is visible in each of panel C (diagonal above the main distribution) and in panel D (horizontal above the main distribution).


A)

Sequence Enrichment by Cluster - B:A

B)

Sequence Enrichment by Cluster - B:A

Figure 37: Cluster Box Plots of Sequence Enrichment. A) Enrichment distribution for the top 20 clusters. B) The same plot after zooming in on the y-axis.

3.12 FASTAptameR-Position_Enrich

3.12.1 Description

The plasticity of each position within a query sequence, as defined by its ability to tolerate substitutions, is related to the importance of that position and establishing the functional biomolecule. Sequences with mutations at critical positions are expected to deplete relative to those with mutations in more neutral locations. For aptamer selections, the functional core is often flanked by unimportant sequences that can be safely trimmed off once they are identified. However, identifying the functional boundaries experimentally can be labor intensive.

The FASTAptameR-Position_Enrich module calculates the average enrichment (or depletion) at each position for each species that does not match the corresponding residue in the user-defined reference sequence at that position. 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. Low 'enrichment' values imply low tolerance for substitution at those positions, which is typically interpreted as implying that those positions are important for function. Given the algorithm that computes these position-specific enrichment values, 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). Sequences with lengths different than the reference sequence will be omitted from these calculations.

This module accepts a CSV from the previous module (FASTAptameR-Sequence_Enrich), though it exclusively operates on the `enrichment_ba` column. Thus, the output CSV file from an enrichment analysis of >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 values for each position. The second is a heat map that shows the average enrichment per position per residue.

All modules directly connected to FASTAptameR-Position_Enrich are shown in **Figure 38**, and a screenshot of the module interface is shown in **Figure 39**.

3.12.2 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 alphabet to be altered to include nonstandard nucleotides or amino acids. 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.*, `U,F` will replace `U` with `F` 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 for the plot). The final set of radio buttons determines whether the algorithm searches for nucleotide or amino acid residues. The next three text boxes allow the user to set the "low", "middle", and "high" colors for the plots.

The text input at the bottom of the UI allows the user to enter comma-separated breakpoints for which average enrichments will be calculated. For example, if the user enters `1,20,50` and the reference sequence is 70 nucleotides, then position-specific enrichments will be averaged for positions [1,20), [20, 50), and [50, 70).

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

3.12.3 Plotting

FASTAptameR-Position_Enrich generates two types of plots (**Figure 40**): 1) average enrichment bar plot and 2) average enrichment heat map. When the mouse hovers over a bar in either plot, the position and

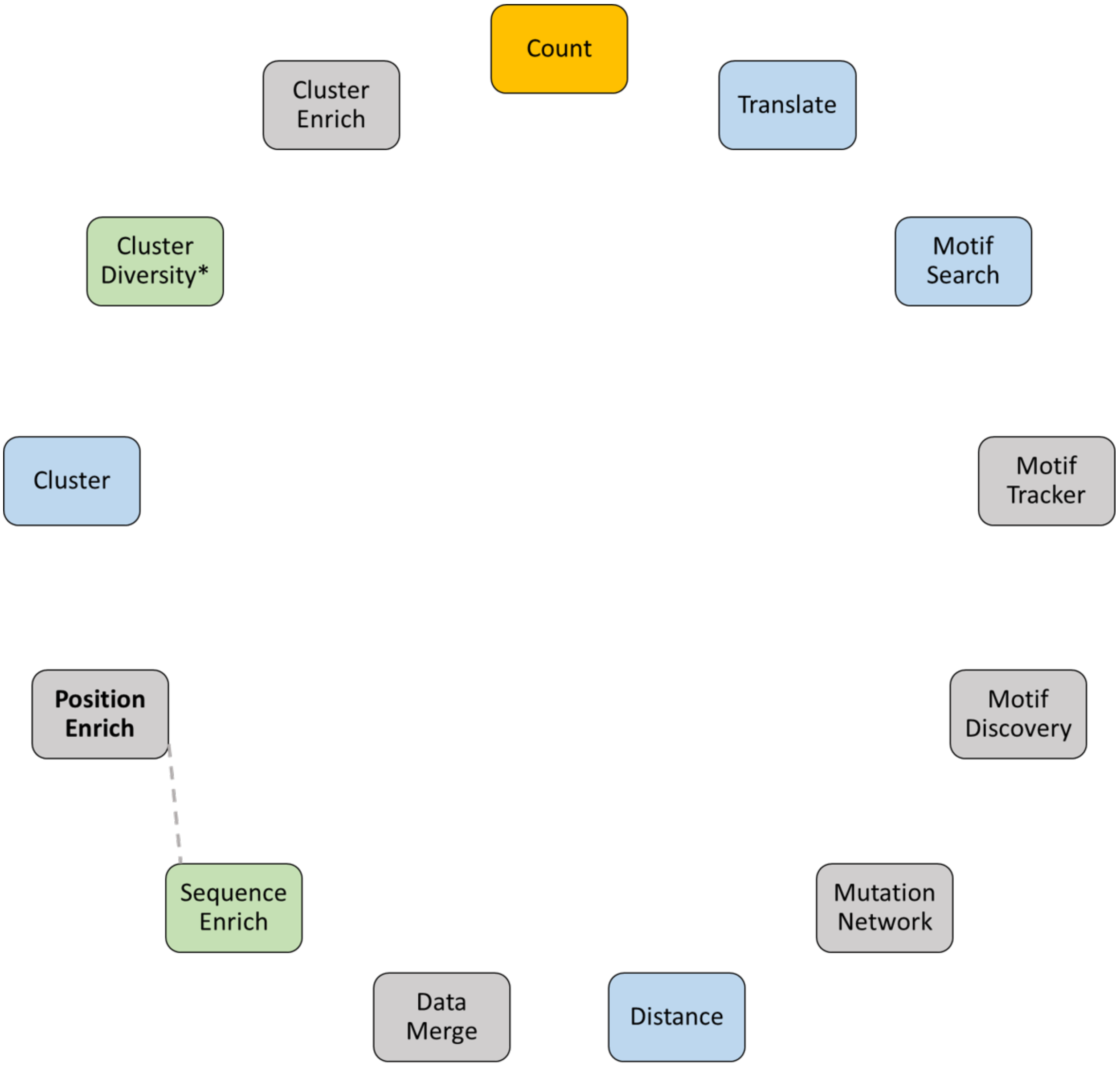
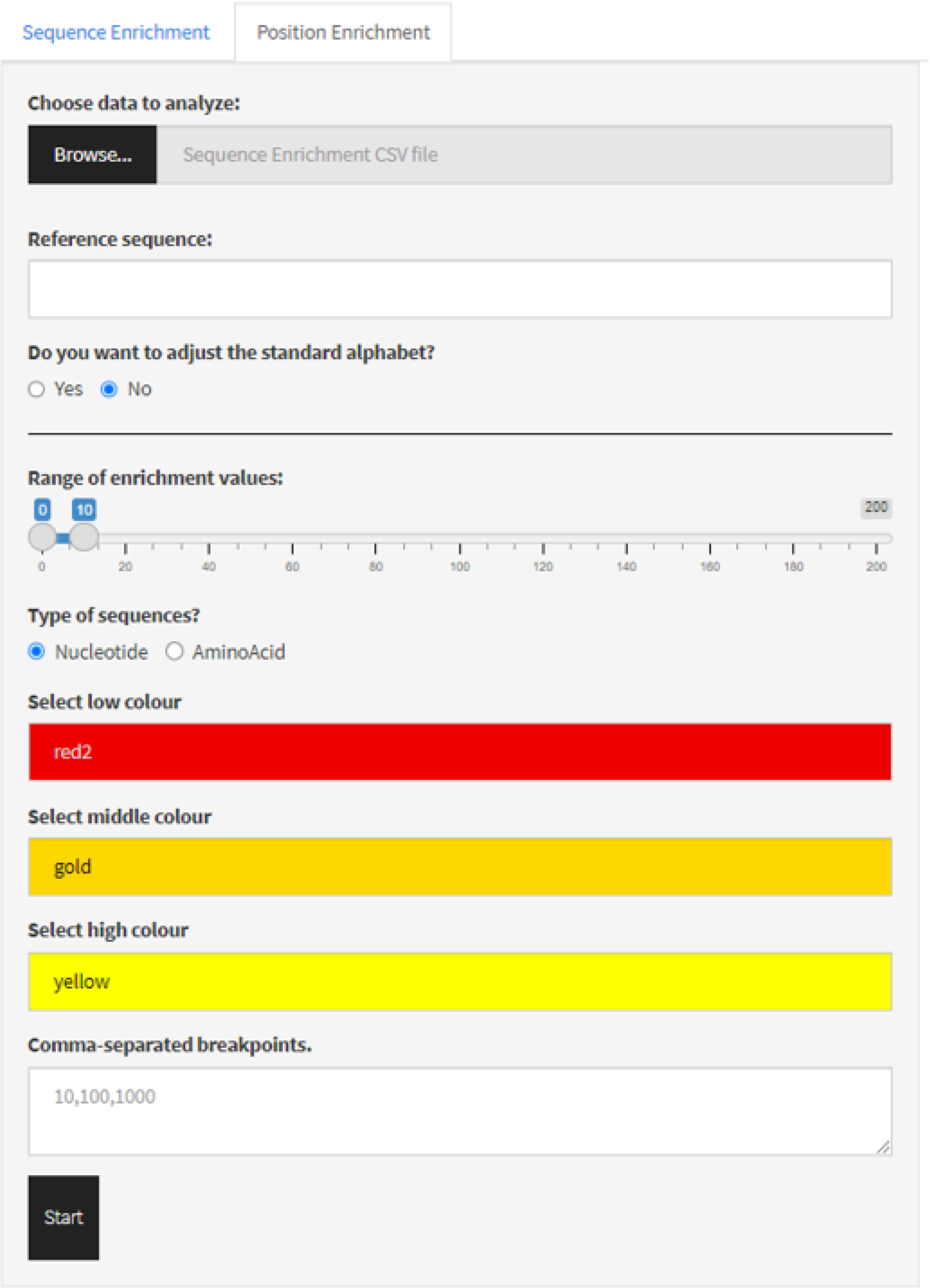


Figure 38: All modules connected to FASTAptameR-Position_Enrich.



Choose data to analyze:

Browse... Sequence Enrichment CSV file

Reference sequence:

Do you want to adjust the standard alphabet?

○ Yes ● No

Range of enrichment values:

0 10 200

Type of sequences?

● Nucleotide ○ AminoAcid

Select low colour

red2

Select middle colour

gold

Select high colour

yellow

Comma-separated breakpoints.

10,100,1000

Figure 39: Screenshot of FASTAptameR-Position_Enrich. Note that the three color choices are used to create a gradient, and the midpoint is set at the 65th quantile.

enrichment value is returned. The bar plot (**Figure 40A**) shows the reference sequence on the x-axis and the average enrichment on the y-axis. The heat map (**Figure 40B**) shows the reference sequence on the x-axis, possible residues on the y-axis, and average enrichment in the color axis.

The motif of interest for this example is the family 1 pseudoknot (F1Pk), which is defined as `UCCG[n*]CGGGAnAAAA` where `n` is any nucleotide and `[n*]` is any number (typically 4-10) of any nucleotide (C. Tuerk, Macdougal, and Gold 1992; Ditzler MA 2013). The F1Pk is shown between the dashed lines in **Figures 40A,B**. **Figure 40C** shows the experimentally determined secondary structure of this motif. The workflow to generate these plots is given below:

1. Count 70HRT14.fastq and 70HRT15.fastq with FASTAptameR-Count.
2. Generate top five clusters for each counted population with FASTAptameR-Cluster. Include all sequences (min. reads > 0) and use LED = 7.
3. Use cluster 2 from 70HRT14 and cluster 1 from 70HRT15 in FASTAptameR-Sequence_Enrich Representative sequences from each population contain the F1Pk motif.
4. Use the output CSV in FASTAptameR-Position_Enrich, specifying the most abundant sequence in 70HRT15 as the reference sequence.

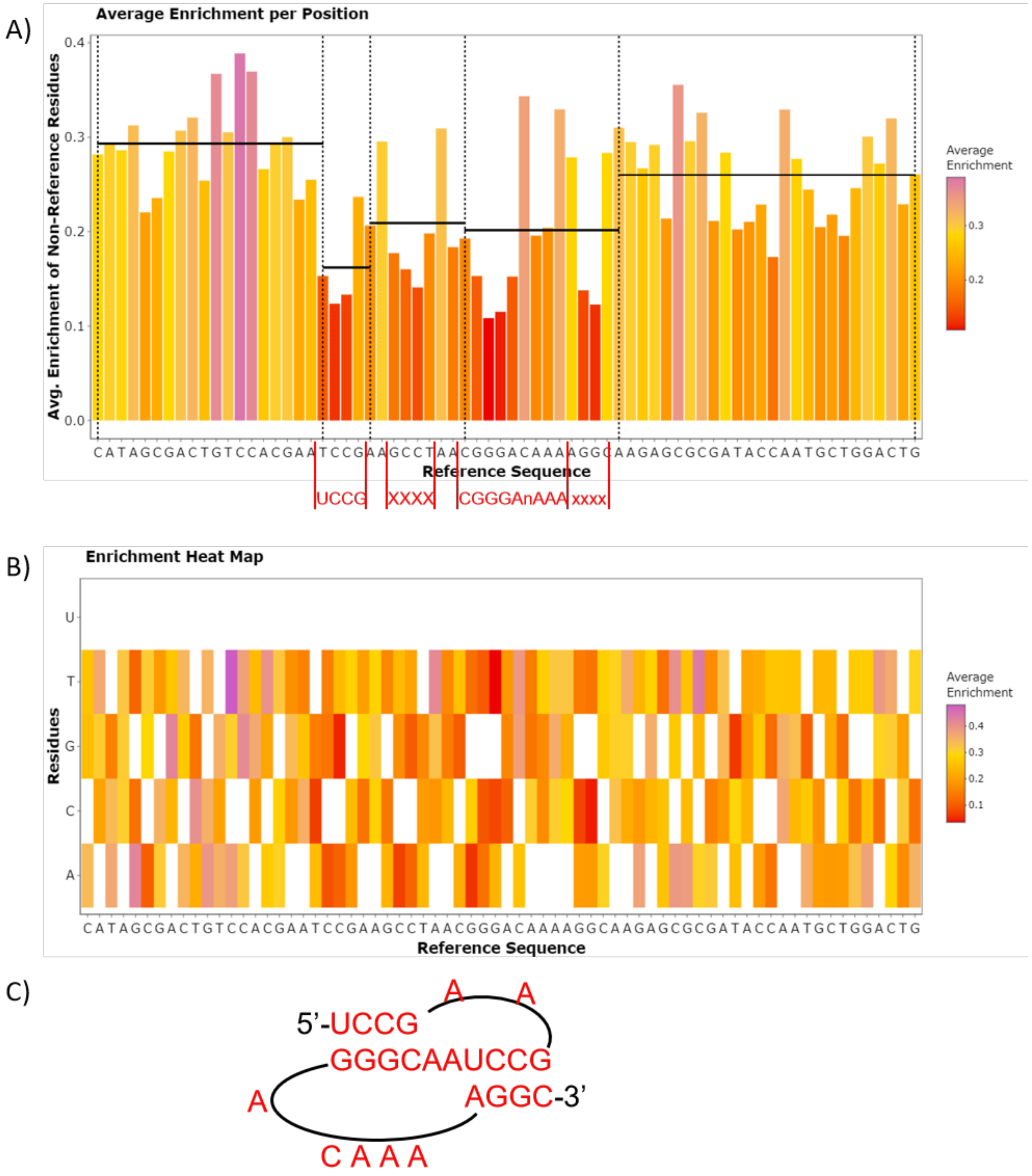


Figure 40: FASTAptameR-Position_Enrich Plots. A) The heat map shows the user-defined reference sequence on the x-axis and all possible residues on the y-axis (nucleotides for this use case). Colors depict average enrichment of non-reference residues. B) The bar plot shows the user-defined reference sequence on the x-axis and average enrichment of non-reference residues on the y-axis. Horizontal dashed lines indicate average enrichment values (left-inclusive) within the ranges of positions 1-19, 20-42, and 43-70. Positions ~20-43 have low enrichment scores, suggesting the importance of nucleotides within this region, which contains the F1Pk motif. C) The experimentally determined structure the F1Pk module within the most abundant sequence of 70HRT15.

3.13 FASTAptameR-Cluster

3.13.1 Description

Combinatorial selections often produce clusters of very closely related sequences as a result of divergent evolution (accumulation of point mutations, neutral drift), or convergent evolution (independent seeding of the population with closely related sequences, as with low-level mutatgenesis of a seed sequence or dense oversampling at a limited number of positions). Grouping sequences into clusters can greatly simplify analysis of population structure and evolution outcomes/dynamics. Sequence clusters often behave as quasi-species, sampling local sequence space and evolving in similar fashion in response to local fitness landscapes. The FASTAptameR-Cluster module defines sequence clusters within the population, while the other modules in this platform - FASTAptameR-Cluster_Diversity, FASTAptameR-Cluster_Enrich, and the box plot feature of FASTAptameR-Sequence_Enrich - allow the user to explore different aspects of the quasi-species nature of the clusters.

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, up to a user-defined number of output files.

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.

All modules directly connected to FASTAptameR-Cluster are shown in **Figure 41**, and a screenshot of the module interface is shown in **Figure 42**.

3.13.2 Usage

The input FASTA file must be chosen with the file browser. The first slider bar (**Figure 42A**) sets the minimum number of reads a sequence must have for it to be included within a cluster (`DEFAULT = 10`). Sequences with the chosen number or fewer reads are removed prior to clustering. Thus, setting this filter to values >1 can significantly shorten runtime, especially for relatively complex data sets. The second slider bar (**Figure 42B**) sets the maximum Levenshtein edit distance to consider between a seed sequence and all other sequences (`DEFAULT = 7`). Users may wish to run FASTAptameR-Distance to guide threshold LED values to use in establishing cluster definitions (see **Figure 28**) The third slider bar (**Figure 42C**) sets the total number of desired clusters (`DEFAULT = 20` to limit runtime during exploratory runs). Note, any remaining sequences will be grouped as `NC` (“not clustered”).

The first set of radio buttons (**Figure 42D**) 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 second set of radio buttons (**Figure 42E**) indicate whether each cluster should be written to a different FASTA file (`DEFAULT = No`). If `No`, then all clusters are grouped together and can be 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 that if this option is `Yes`, then a directory path **must be copied or typed** into the corresponding text box (**Figure 42F**) if this option is `Yes`. A sample directory path could be `C:/Users/Kramer/Desktop/Data/directory/`, though this will depend on your system. **Please note** that this requires backslashes (`/`) and that forward slashes (`\`) will cause errors. Also note that the path should end with a backslash.

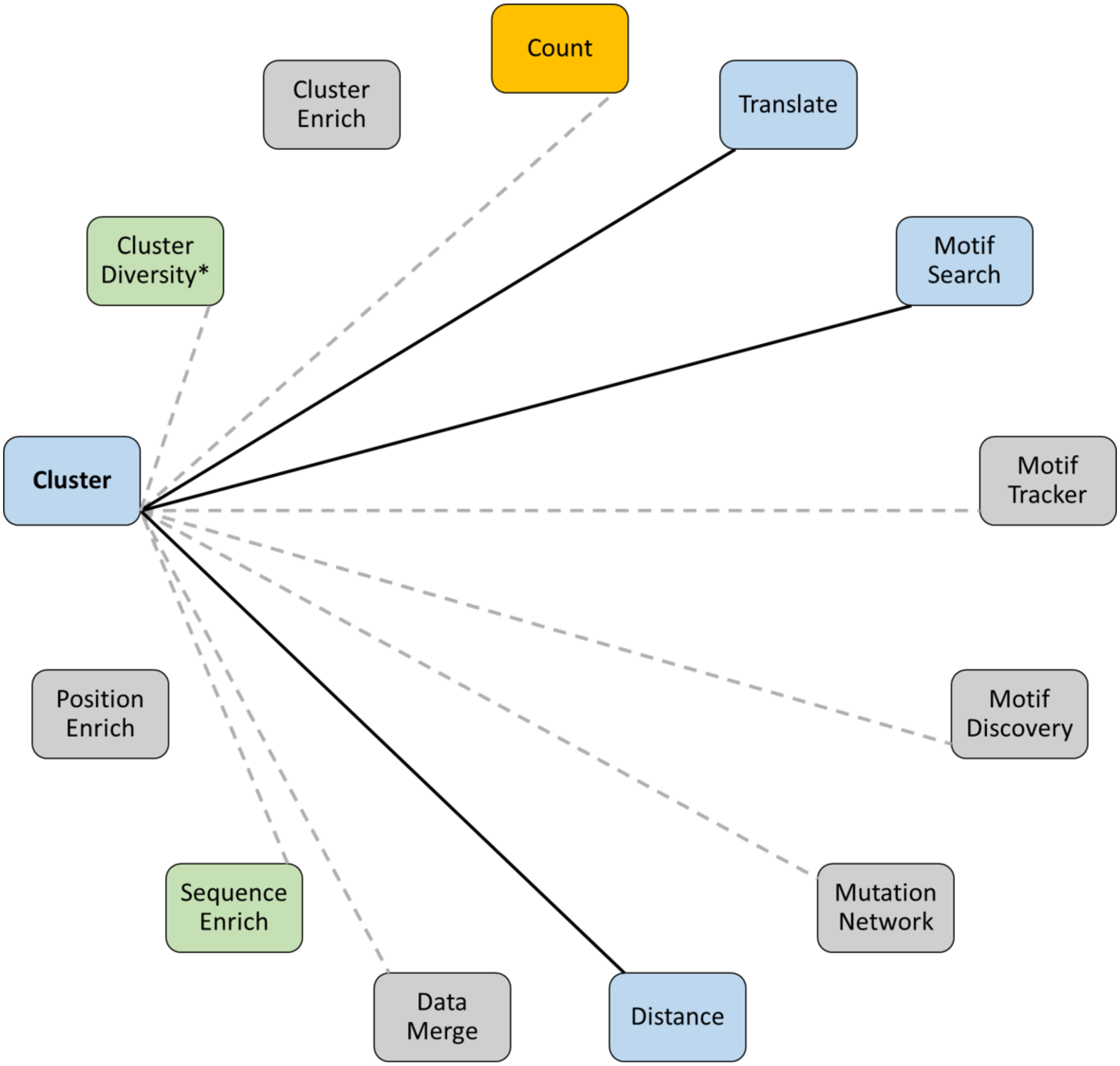
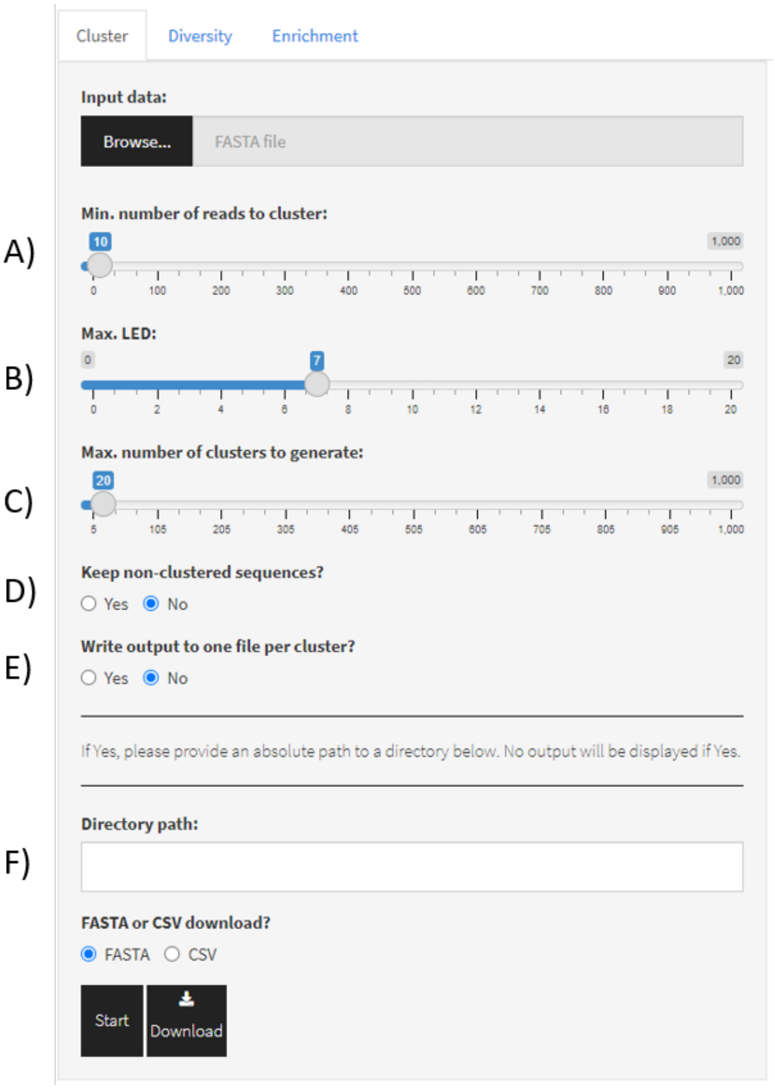


Figure 41: All modules connected to FASTAptameR-Cluster.



Input data:

Min. number of reads to cluster: 10 (range 0 – 1,000)

Max. LED: 7 (range 0 – 20)

Max. number of clusters to generate: 20 (range 5 – 1,000)

Keep non-clustered sequences?

○ Yes ● No

Write output to one file per cluster?

○ Yes ● No

If Yes, please provide an absolute path to a directory below. No output will be displayed if Yes.

Directory path:

FASTA or CSV download?

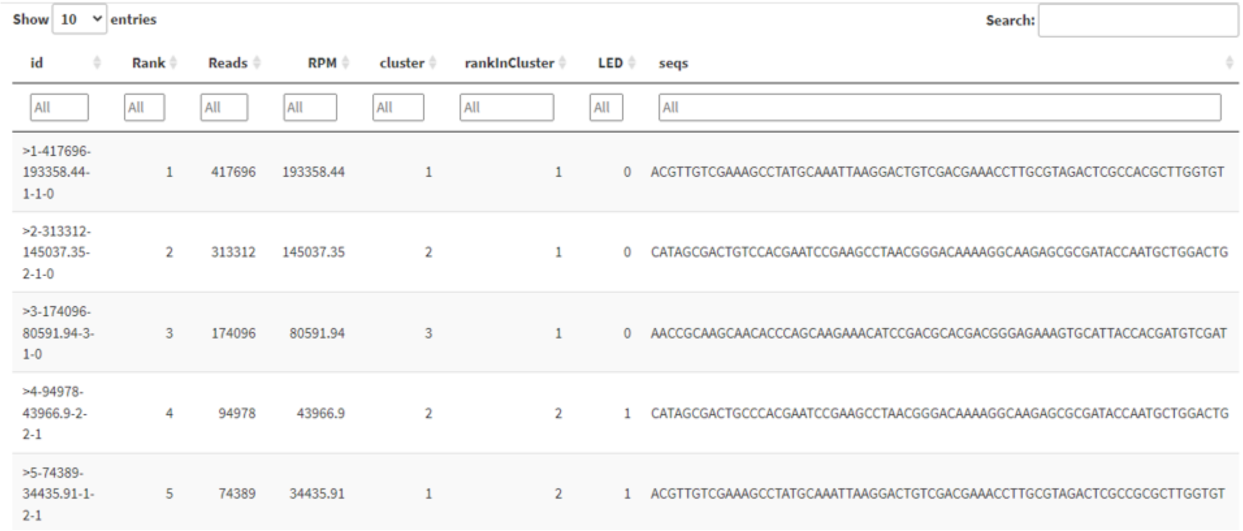
● FASTA ○ CSV

Figure 42: Screenshot of FASTAptameR-Cluster.

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 **Figure 43**.



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 43: FASTAptameR-Cluster Output.

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

3.14 FASTAptameR-Cluster_Diversity

3.14.1 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.

All modules directly connected to FASTAptameR-Cluster_Diversity are shown in **Figure 44**, and a screenshot of the module interface is shown in **Figure 45**.

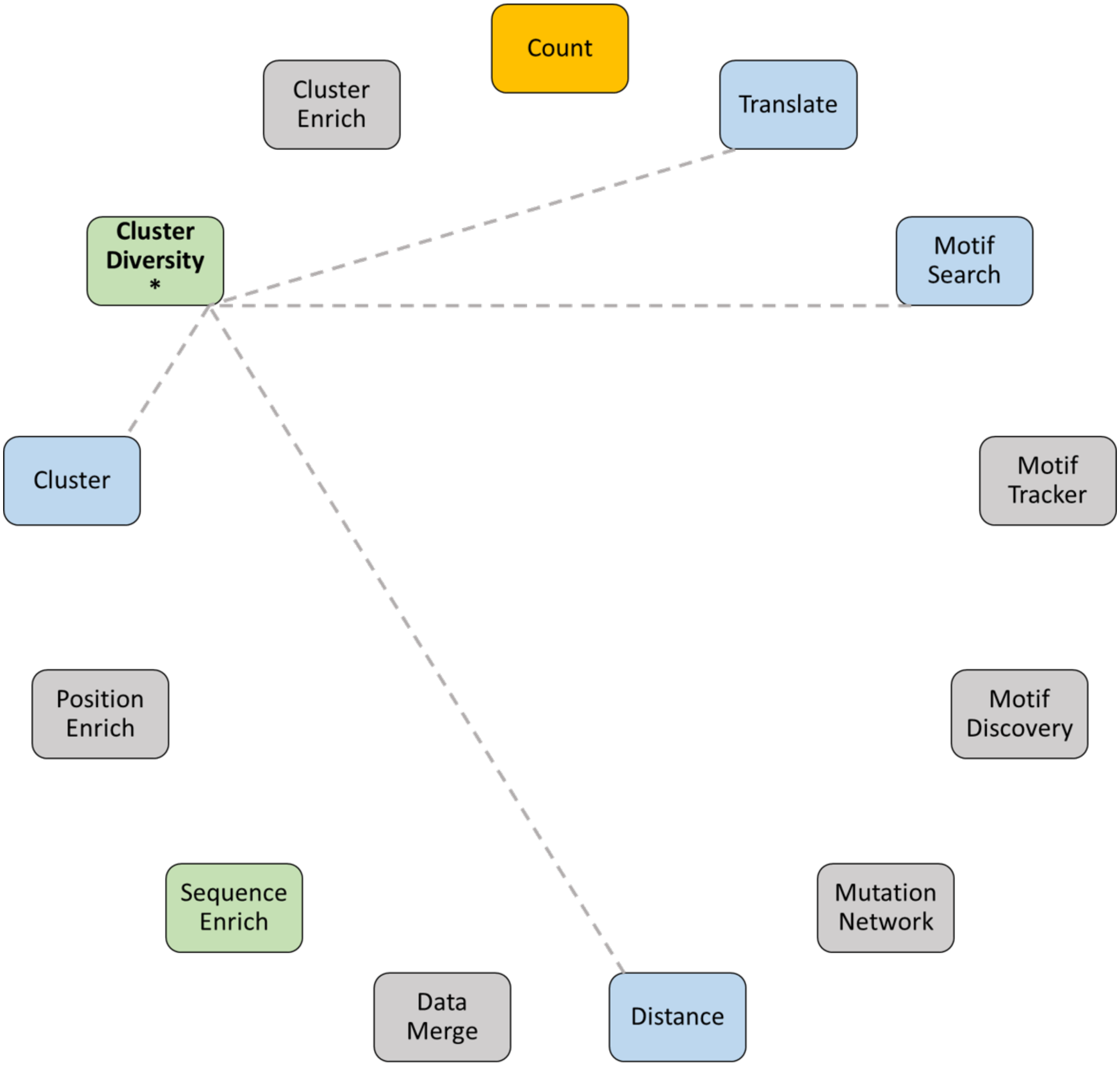
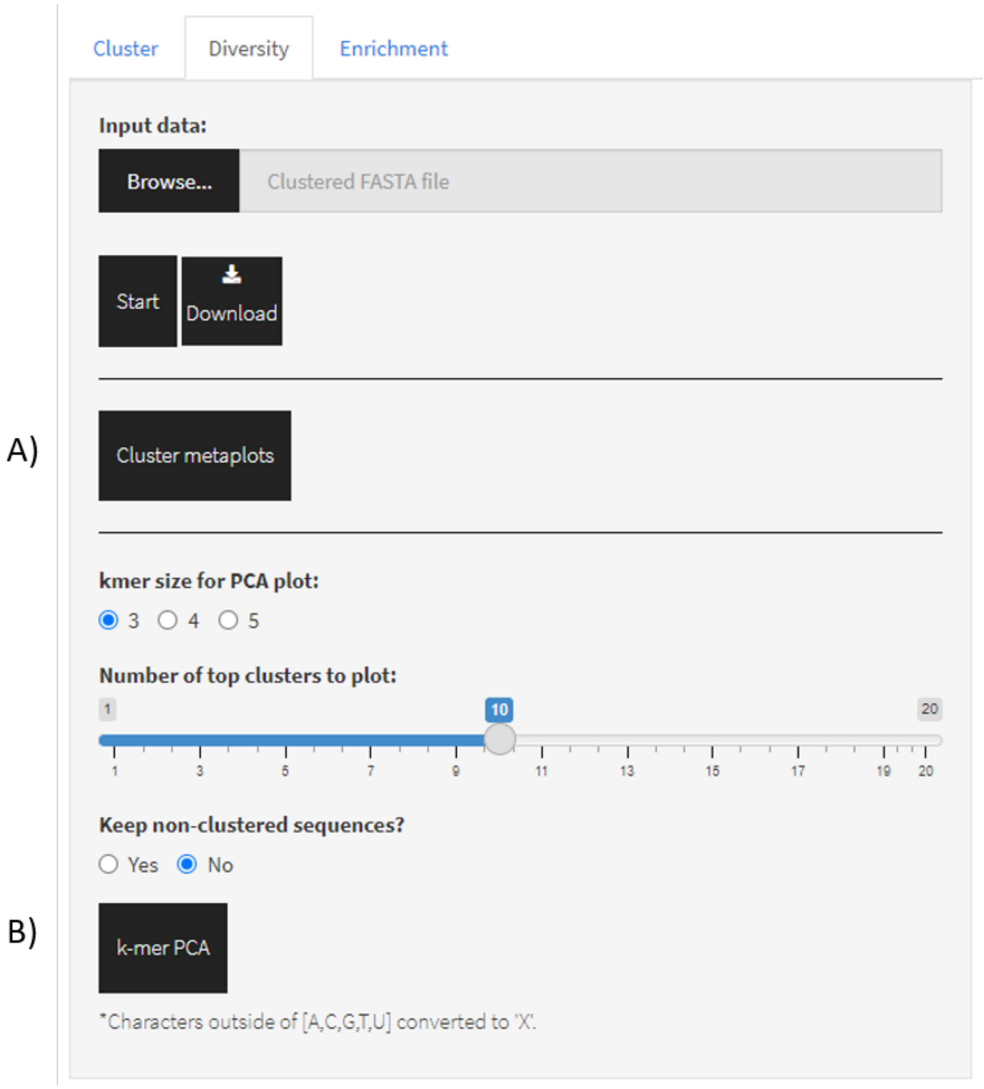


Figure 44: All modules connected to FASTAptameR-Cluster_Diversity.



Input data:

Clustered FASTA file

kmer size for PCA plot:

● 3 ○ 4 ○ 5

Number of top clusters to plot:

1 10 20

1 3 5 7 9 11 13 15 17 19 20

Keep non-clustered sequences?

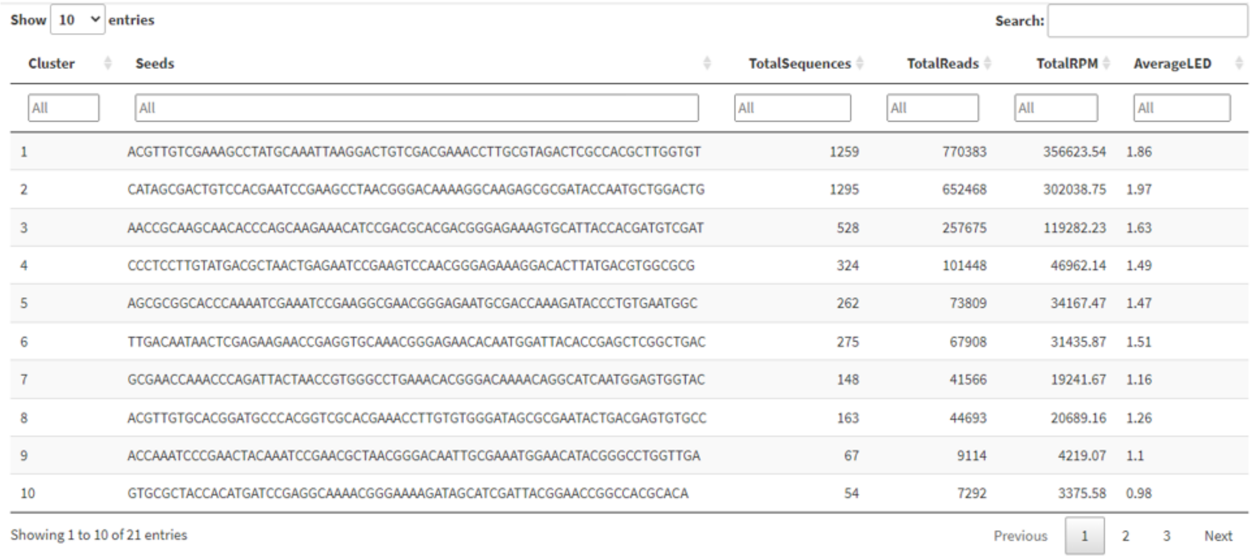
○ Yes ● No

*Characters outside of [A,C,G,T,U] converted to 'X'.

Figure 45: Screenshot of FASTAptameR-Cluster_Diversity.

3.14.2 Usage

The input FASTA file (clustered FASTA file from FASTAptameR-Cluster) 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 will 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 **Figure 46**.



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

Figure 46: FASTAptameR-Cluster_Diversity Output.

3.14.3 Plotting

This module can generate metaplots of the analyzed data. These line plots correspond to the number of unique sequences per cluster, total reads per cluster, and average LED to seed sequence per cluster (button shown in **Figure 45A**, output plots shown in **Figure 47A**).

This module is also able to analyze clusters by converting all sequences into k-mer vectors (see below) and rendering an interactive 2D PCA plot, colored by cluster (button shown in **Figure 45B**, output shown in **Figure 47B**). The value of `k` can be chosen with the first set of radio buttons (`DEFAULT = 3`). The slider bar indicates how many of the top clusters should be plotted (max = 21 clusters due to graphics limitations). The second set of radio buttons 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.

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 alter any set of sequences with characters outside of `[A, C, G, T/U]` by converting any other character to `X`.

3.14.4 A side note on how k-mer clustering works and what it means

In the k-mer method, sequence relatedness is measure by the degree to which sequences share a set of short sequence elements ('k-mers'), typically of length 3, 4, or 5. For example, if k=3, there are 64 possible k-mers: 'AAA', 'AAC', 'AAG', 'AAT', 'ACA', 'ACC', . . . , 'TTA', 'TTC', 'TTG', 'TTT.' *These triplet sequences constitute the axes of a 64-dimensional hyperspace.*

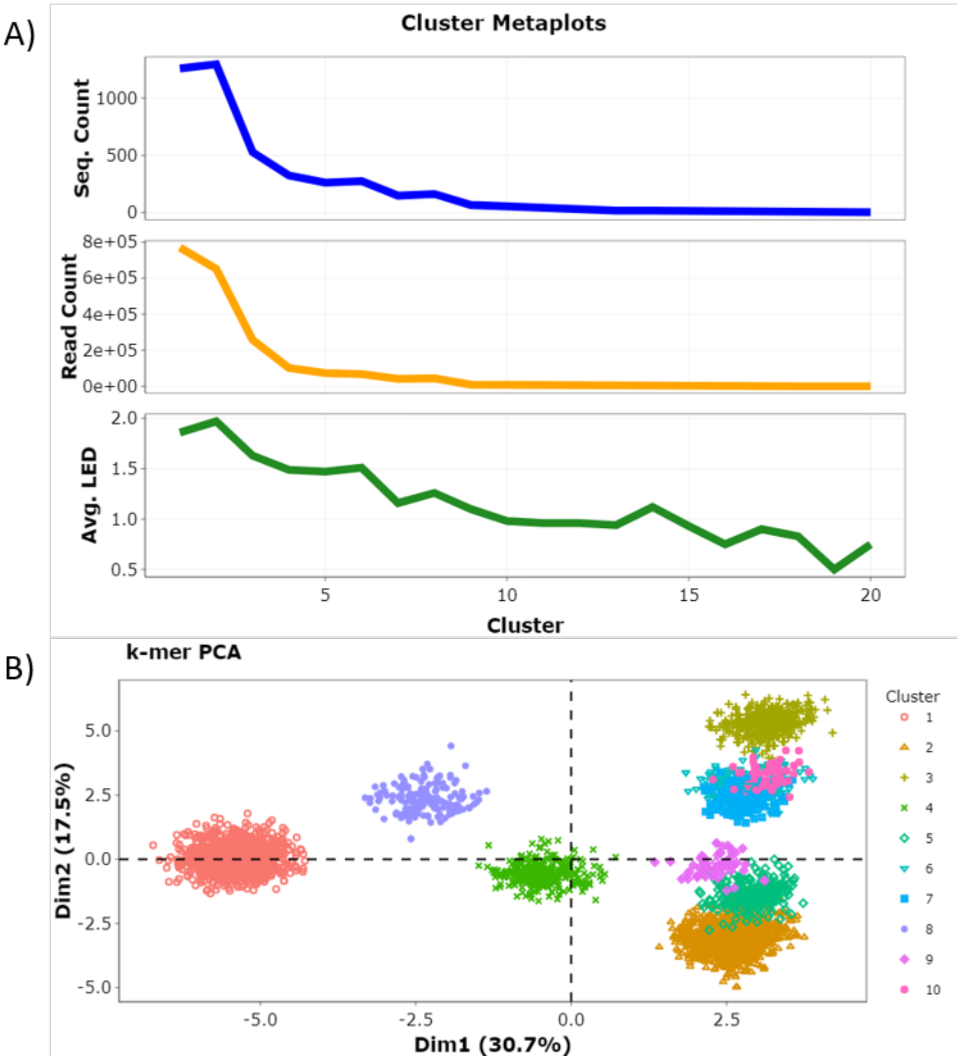


Figure 47: Cluster plots. A) Cluster metaplots depict number of unique sequences, total number of reads, and average LED to cluster seed per cluster. B) The k-mer PCA plot can qualitatively suggest how well the cluster algorithm performed (as indicated by cohesive grouping of each cluster within the plot) and identifies clusters that are especially distinct from (well-separated) or similar to (close or overlapping) the others.

For each sequence, count the number of occurrences for each k-mer. *These counts constitute the values associated with the axes for that sequence.* For example, the sequence 'AAAAGT' has 2 copies of 'AAA', 1 copy of 'AAG', 1 copy of 'AGT', and 0 copies of all other triplets. Every sequence can therefore be defined as a single point in 64-dimensional space, with the (x, y, z, . . .) values corresponding to the respective k-mer counts.

More generally, every sequence can therefore be defined as a single point (or 'vector') in A^k**-dimensional space**, where **A** is the number of letters in the polymer alphabet [standard values = 4 for nucleic acids and 20 for proteins, but other values are also possible], and k is the chosen k-mer value.

The DISTANCE between two sequences is then simply the Euclidean distance calculated from one point to another in that hyperdimensional space.

Don't get thrown off by the word 'hyperdimensional'. It's really easy to see the pattern and then generalize from there. Consider going from 2D to 3D. The distance between two points in 2D is just the Pythagorean formula that you've known for many years:

$d = \sqrt{}([x_2 - x_1]^2 + [y_2 - y_1]^2)$.

Applying this to 3D, you simply add the z-dimension:

$d = \sqrt{}([x_2 - x_1]^2 + [y_2 - y_1]^2 + [z_2 - z_1]^2)$.

As you add more dimensions, you just add more variables.

But what do these distances mean?

The biologically inclined among us are used to thinking of mutational or evolutionary distance, which is essentially the number of mutational events that are thought to have occurred since the two species diverged. k-mer distance is not mutational distance, but the two kinds of 'distance' are related, in that the k-mer analysis gives longer distances for more divergent sequences, so long as "k" is large enough to provide the necessary resolution.

3.15 FASTAptameR-Cluster_Enrich

3.15.1 Description

Considering all the members of a cluster together (rather than as independently evolving species) can help the user spot large-scale trends, in addition to adding statistical weight to enrichment analyses. For example, a cluster with a large number of low-abundance functional variants may outperform another cluster with few variants that are each at higher abundances. The strong performance of the first cluster might be missed in an analysis that looks only at individual species. A counterpoint to this collective approach is provided in the cluster box plot feature of FASTAptameR-Sequence_Enrich (see **Figure 37**).

FASTAptameR-Cluster_Enrich calculates the enrichment (or depletion) of each cluster in one population relative to other populations. Thus, this module is conceptually identical to FASTAptameR-Sequence_Enrich but is applied to clusters rather than individual sequences. The module accepts two or three *Cluster_Diversity* 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 tables can be downloaded as CSV files. The first CSV file provides summary statistics for each cluster in each population. The second CSV file contains enrichment scores.

All modules directly connected to FASTAptameR-Cluster_Enrich are shown in **Figure 48**, and a screenshot of the module interface is shown in **Figure 49**.

3.15.2 Usage

The input CSV files must be chosen with the file browser. The `Start` button begins the enrichment calculation. The results will be displayed as two data tables on the right side of the screen. The first table summarizes each cluster, and the second table provides enrichment values. The `Download` button opens a file browser prior to downloading the output as a CSV file.

A sample output data table is shown in **Figure 50**.

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

3.15.3 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 (**Figure 51**). Although only two populations were used here for illustration, this tool can be especially useful for observing the rise and fall of multiple clusters over the course of multiple rounds of selection.

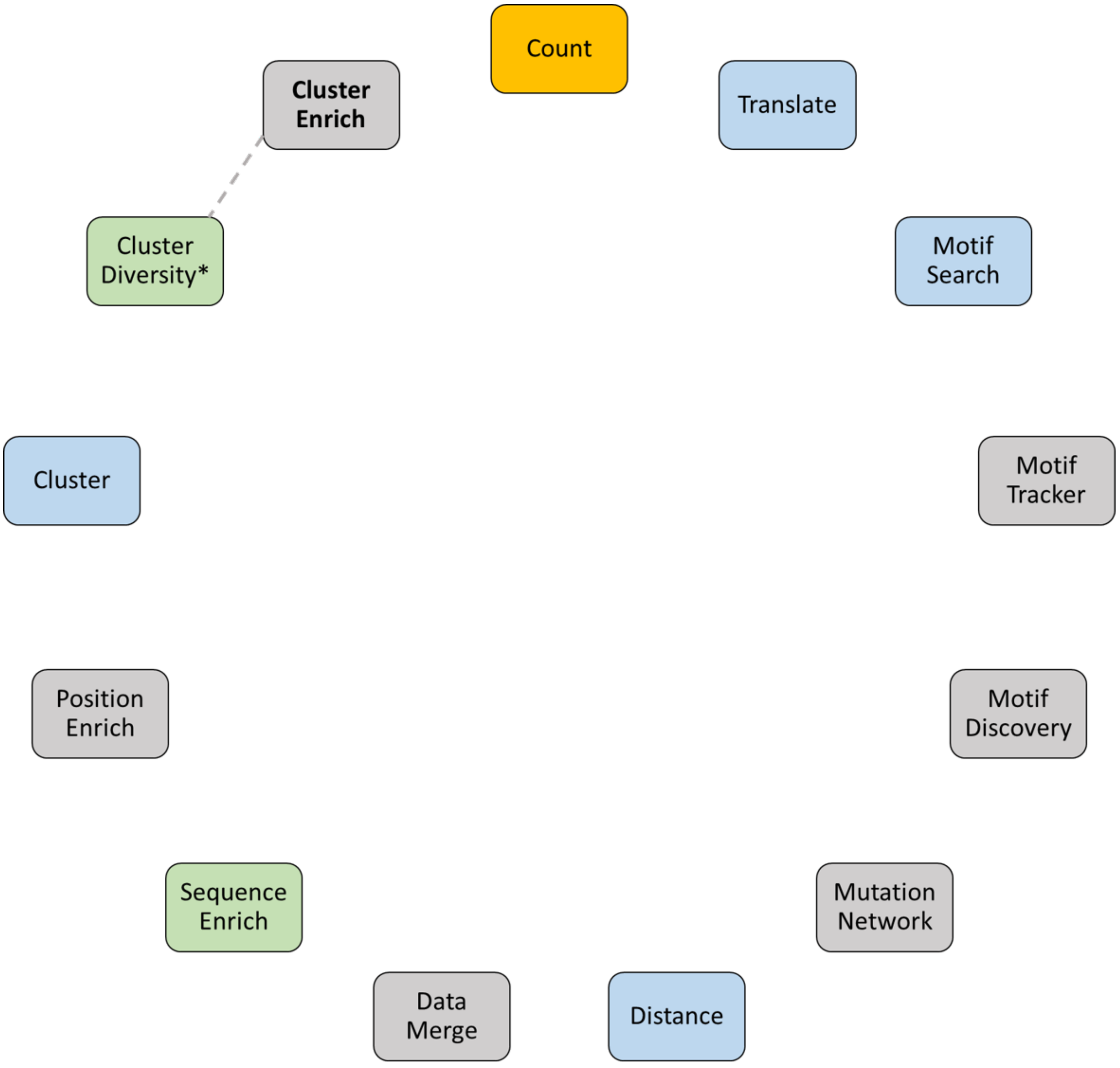
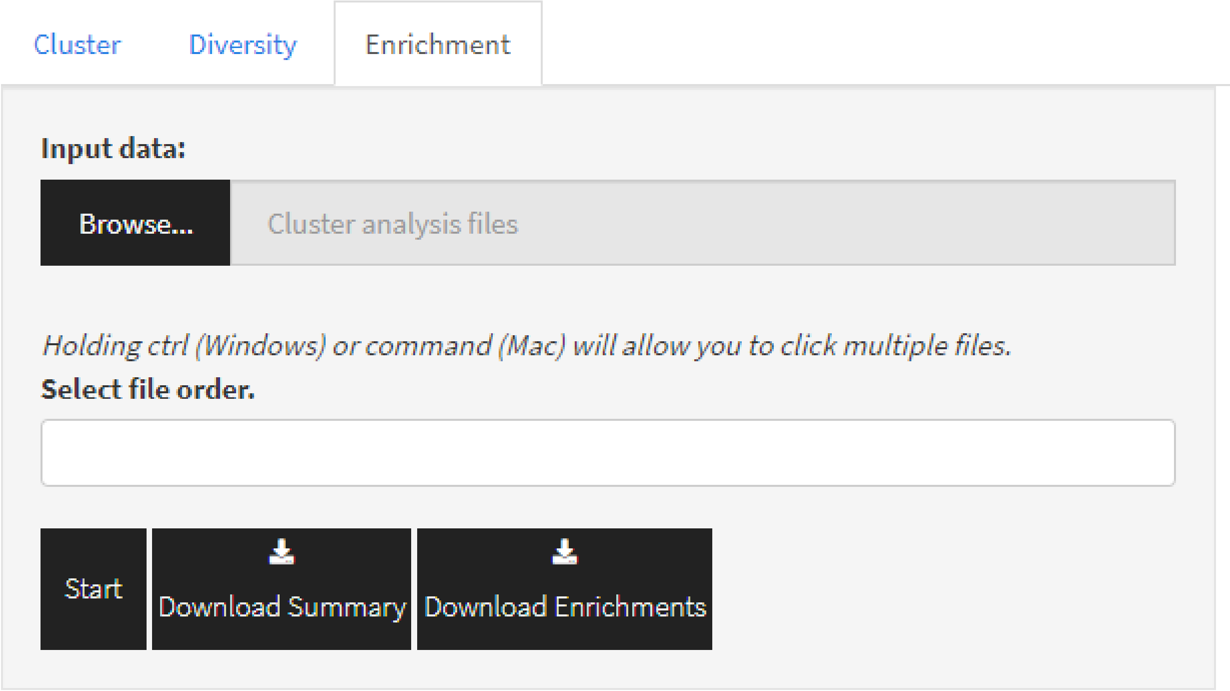


Figure 48: All modules connected to FASTAptameR-Cluster_Enrich.

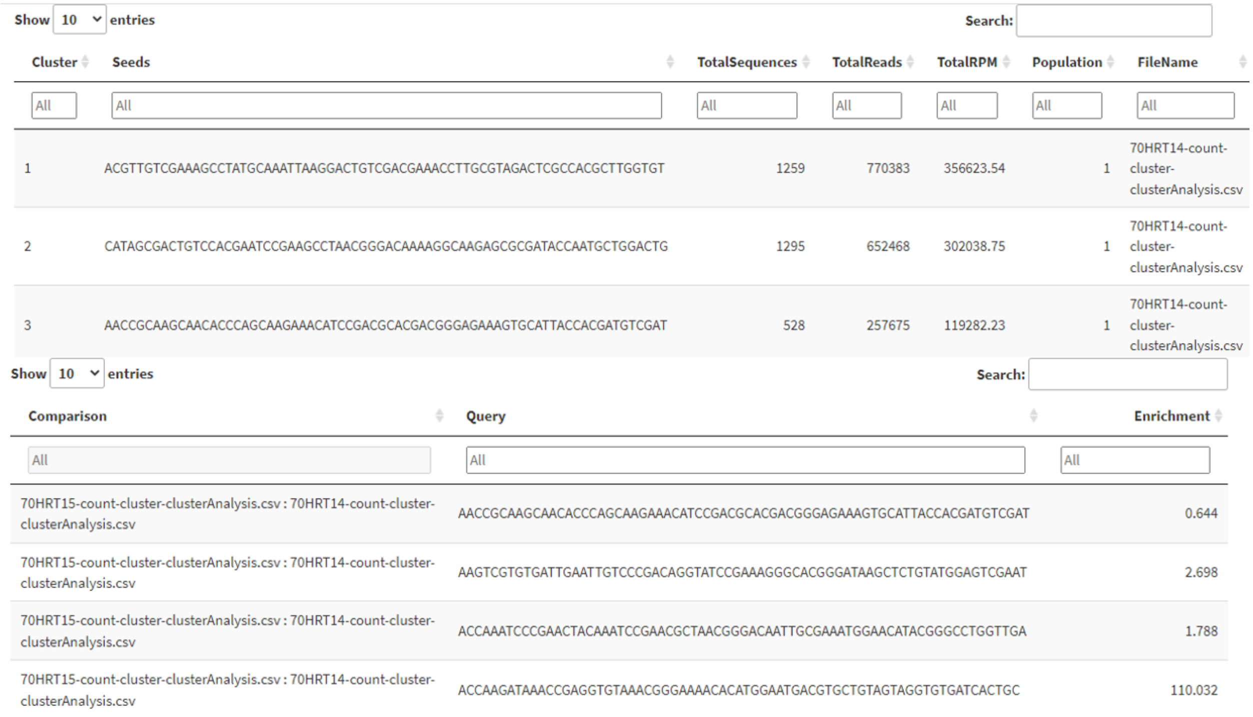


Input data:

Holding ctrl (Windows) or command (Mac) will allow you to click multiple files.

Select file order.

Figure 49: Screenshot of FASTAptameR-Cluster_Enrich.



Cluster	Seeds	TotalSequences	TotalReads	TotalRPM	Population	FileName
All	All	All	All	All	All	All
1	ACGTTGTCGAAAGCCTATGCAAATTAAGGACTGTCGACGAAACCTTGCGTAGACTCGCCACGCTTGGTGT	1259	770383	356623.54	1	70HRT14-count-cluster-clusterAnalysis.csv
2	CATAGCGACTGTCCACGAATCCGAAGCCTAACGGGACAAAAGGCAAGAGCGCGATACCAATGCTGGACTG	1295	652468	302038.75	1	70HRT14-count-cluster-clusterAnalysis.csv
3	AACCGCAAGCAACACCCAGCAAGAAACATCCGACGCACGACGGGAGAAAGTGCATTACCACGATGTCGAT	528	257675	119282.23	1	70HRT14-count-cluster-clusterAnalysis.csv

Comparison	Query	Enrichment
All	All	All
70HRT15-count-cluster-clusterAnalysis.csv : 70HRT14-count-cluster-clusterAnalysis.csv	AACCGCAAGCAACACCCAGCAAGAAACATCCGACGCACGACGGGAGAAAGTGCATTACCACGATGTCGAT	0.644
70HRT15-count-cluster-clusterAnalysis.csv : 70HRT14-count-cluster-clusterAnalysis.csv	AAGTCGTGTGATTGAATTGTCCCGACAGGTATCCGAAAGGGCACGGGATAAGCTCTGTATGGAGTCGAAT	2.698
70HRT15-count-cluster-clusterAnalysis.csv : 70HRT14-count-cluster-clusterAnalysis.csv	ACCAAATCCCGAACTACAAATCCGAACGCTAACGGGACAATTGCGAAATGGAACATACGGGCCTGGTTGA	1.788
70HRT15-count-cluster-clusterAnalysis.csv : 70HRT14-count-cluster-clusterAnalysis.csv	ACCAAGATAAACCGAGGTGTAAACGGGAAAACACATGGAATGACGTGCTGTAGTAGGTGTGATCACTGC	110.032

Figure 50: FASTAptameR-Cluster_Enrich Output.

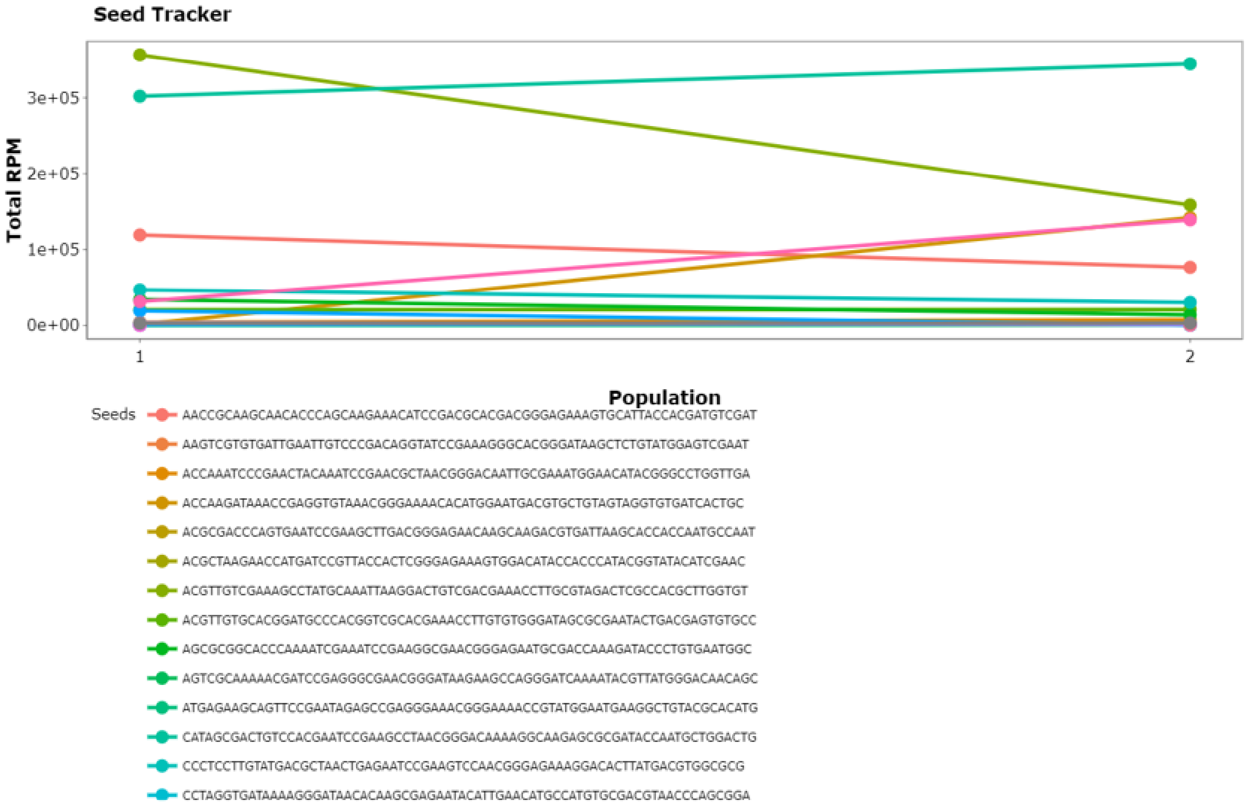


Figure 51: Cluster Seed Tracker Line Plot.

4 Known issues

- **File upload speed**: The speed at which files are uploaded for larger files (especially those going into FASTAptameR-Count) can be quite slow due to a number of issues. If you have large files that need processing, we recommend using the local version of FASTAptameR 2.0.
- **Web server greys out and becomes unresponsive**: We are aware of this issue and are looking into how to fix it. In the meantime, we have found that refreshing the page, reopening it, or switching browsers can temporarily solve this issue. Alternatively, you could use the local version of FASTAptameR 2.0.

5 Version history

- v2.0.0 (29 Apr 2022; simultaneous with manuscript submission to bioRxiv and MTNA): The initial release of FASTAptameR 2.0
- v2.0.1 (28 Jun 2022): Corrected how the server checks the file inputs for modules that allow multiple inputs (*e.g.*, FASTAptameR-Sequence_Enrich) such that the code is compatible with newer versions of R
- v2.0.2 (08 Jul 2022): Altered the visualization for the cluster metaplots in FASTAptameR-Cluster_Diversity such that all axes are clearly visible (previous plot was created with plotly, whereas the current plots are created with a combination of ggplot2 and plotly)
- v2.1.0 (29 Jul 2022; simultaneuos with manuscript resubmission to MTNA):
  - Added functionality to:
    - generate reverse complements of sequences in FASTAptameR-Count (via the Biostrings package)
    - find over-represented D/RNA strings in FASTAptameR-Motif_Discovery (via the FSBC tool); includes visualization of results
    - find mutational intermediates in FASTAptameR-Mutation_Network (via the cppRouting package)
    - merge FASTA files with union, intersection, or left joins in FASTAptameR-Data_Merge; includes visualization of results
  - removed bug in FASTAptameR-Translate that set each translation to the standard genetic code, regardless of user input

References

Alam, Khalid K., Jonathan L. Chang, Margaret J. Lange, Phuong D. M. Nguyen, Andrew W. Sawyer, and Donald H. Burke. 2018. "Poly-Target Selection Identifies Broad-Spectrum RNA Aptamers." *Molecular Therapy - Nucleic Acids* 13: 605–19. https://doi.org/10.1016/j.omtn.2018.10.010.

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.

Bailey, Timothy L., James Johnson, Charles E. Grant, and William S. Noble. 2015. "The MEME Suite." *Nucleic Acids Res.* 43 (W1): W39–49. https://doi.org/10.1093/nar/gkv416.

Bolger, Anthony M., Marc Lohse, and Bjoern Usadel. 2014. "Trimmomatic: a flexible trimmer for Illumina sequence data." *Bioinformatics* 30 (15): 2114–20. https://doi.org/10.1093/bioinformatics/btu170.

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.

Chen, Shifu, Yanqing Zhou, Yaru Chen, and Jia Gu. 2018. "fastp: an ultra-fast all-in-one FASTQ preprocessor." *Bioinformatics* 34 (17): i884–90. https://doi.org/10.1093/bioinformatics/bty560.

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.

Gasteiger, E. 2003. "ExPASy: The Proteomics Server for in-Depth Protein Knowledge and Analysis." *Nucleic Acids Research* 31 (13): 3784–88. https://doi.org/10.1093/nar/gkg563.

Kato, Shintaro, Takayoshi Ono, Hirotaka Minagawa, Katsunori Horii, Ikuo Shiratori, Iwao Waga, Koichi Ito, and Takafumi Aoki. 2020. "FSBC: Fast String-Based Clustering for HT-SELEX Data." *BMC Bioinformatics* 21 (1). https://doi.org/10.1186/s12859-020-03607-1.

Kramer, Skyler T., Paige R. Gruenke, Khalid K. Alam, Dong Xu, and Donald H. Burke. 2022. "FASTAptameR 2.0: A Web Tool for Combinatorial Sequence Selections." *bioRxiv.* https://doi.org/10.1101/2022.04.27.489774.

Martin, Marcel. 2011. "Cutadapt Removes Adapter Sequences from High-Throughput Sequencing Reads." *EMBnet.journal* 17 (1): 10. https://doi.org/10.14806/ej.17.1.200.

Nawrocki, Eric P., and Sean R. Eddy. 2013. "Infernal 1.1: 100-fold faster RNA homology searches." *Bioinformatics* 29 (22): 2933–35. https://doi.org/10.1093/bioinformatics/btt509.

Tuerk, C., S. Macdougal, and L. Gold. 1992. "RNA Pseudoknots That Inhibit Human Immunodeficiency Virus Type 1 Reverse Transcriptase." *Proceedings of the National Academy of Sciences* 89 (15): 6988–92. https://doi.org/10.1073/pnas.89.15.6988.

Tuerk, Craig, and Larry Gold. 1990. "Systematic Evolution of Ligands by Exponential Enrichment: RNA Ligands to Bacteriophage T4 DNA Polymerase" 249 (4968): 505–10. https://doi.org/10.1126/science.2200121.

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.