FASTAptamer: A Bioinformatic Toolkit for High-throughput Sequence Analysis of Combinatorial Selections

Khalid K Alam¹, Jonathan L Chang²,³ and Donald H Burke¹,²

High-throughput sequence (HTS) analysis of combinatorial selection populations accelerates lead discovery and optimization and offers dynamic insight into selection processes. An underlying principle is that selection enriches high-fitness sequences as a fraction of the population, whereas low-fitness sequences are depleted. HTS analysis readily provides the requisite numerical information by tracking the evolutionary trajectory of individual sequences in response to selection pressures. Unlike genomic data, for which a number of software solutions exist, user-friendly tools are not readily available for the combinatorial selections field, leading many users to create custom software. FASTAptamer was designed to address the sequence-level analysis needs of the field. The open source FASTAptamer toolkit counts, normalizes and ranks read counts in a FASTQ file, compares populations for sequence distribution, generates clusters of sequence families, calculates fold-enrichment of sequences throughout the course of a selection and searches for degenerate sequence motifs. While originally designed for aptamer selections, FASTAptamer can be applied to any selection strategy that can utilize next-generation DNA sequencing, such as ribozyme or deoxyribozyme selections, in vivo mutagenesis and various surface display technologies (peptide, antibody fragment, mRNA, etc.). FASTAptamer software, sample data and a user’s guide are available for download at http://burkelab.missouri.edu/FASTAptamer.html.

Molecular Therapy—Nucleic Acids (2015) 4, e230; doi:10.1038/mtna.2015.4; published online 3 March 2015

Subject Category: Bioinformatics, Aptamers, Ribozymes and DNAzymes

Introduction

Combinatorial selection technologies identify novel ligands, catalysts, and other functional biomolecules from large randomized libraries. The selected biomolecules find application as reagents for diagnostics¹ and therapeutics,² and as research tools to probe sequence-structure or sequence-function relationships,³ fitness landscapes,⁴-⁶ the origins of life,⁷ and other questions.⁸ Many technology platforms exist for carrying out combinatorial selections, such as in vitro evolution, various surface display techniques and biological selections, each of which samples diverse nucleic acid-encoded sequence space for desired phenotypes. The defining step in each strategy is to partition or preferentially amplify molecules of high-fitness from those of low-fitness. This evolutionary process is typically iterated for several rounds that include enzymatic or biological amplification of surviving molecules. Each iteration shifts genotypic and phenotypic frequencies within the population to favor those molecules that best survive the selection process, until a highly enriched library emerges.

Analysis typically begins by sequencing the functional nucleic acids (for aptamer or (deoxy)ribozyme selections) or the genes that encode the selected amino acid sequences (for phage display and in vivo selections). It is common to clone the output of the final selection round and to sequence a number of plasmids using chain-terminating Sanger sequencing. This low-throughput sequencing method has identified functional biomolecules and provided a low-resolution snapshot of the selected populations at the end of the process. However, the steadily decreasing cost and increasing availability of next-generation HTS technologies provide opportunities to increase sampling depth dramatically and to extract high-resolution sequence information from multiple selection rounds, thereby making it possible to understand the dynamics of enrichment as they occurred.⁹,¹⁰ Furthermore, monitoring the evolutionary trajectory of individual sequences throughout the course of a selection provides insight that can facilitate earlier discovery of candidate molecules and minimize the number of selection rounds performed,¹¹,¹² thereby limiting the loss of high-affinity molecules by preserving library diversity⁴,¹³ and reducing biases associated with biological amplification, nontarget binding, and cloning.¹⁴-¹⁷

Nevertheless, analysis of the data remains an obstacle to many practitioners of combinatorial selections. Although a few software tools have been published to analyze HTS data from combinatorial selections,¹⁸-²² these have not yet been widely adopted. Among workflows and tools that are shared, it is often the case that they are available only upon request, that they require a high level of computational expertise to implement, or that they are built around software primarily designed to answer specific experimental questions. For most other studies, informatics pipelines are unpublished custom scripts that lack the transparency and standardized workflow that has become essential in the era of Big Data.

¹Department of Biochemistry, University of Missouri, Columbia, Missouri, USA; ²Department of Molecular Microbiology and Immunology, University of Missouri, Columbia, Missouri, USA; ³Current Address: School of Medicine, University of Missouri, Columbia, Missouri, USA. Correspondence: Donald H Burke, Department of Biochemistry, University of Missouri, Columbia, Missouri, USA, E-mail: burkedh@missouri.edu

Keywords: aptamer; bioinformatics; high-throughput sequence; next-generation sequencing; phage display; ribozyme; sequencing; SELEX; software

Received 31 August 2014; accepted 8 January 2015; published online 3 March 2015. doi:10.1038/mtna.2015.4
Taken together, these factors create an unnecessarily high barrier to the broad implementation of HTS and bioinformatic technologies by the combinatorial selections field.

To address these issues, we have developed the FASTAptamer toolkit as an open source collection of scripts that seamlessly perform many of the first-stage, sequence-level tasks that are common to all combinatorial selections, independent of the technology employed in the selection. The initial release of the toolkit (version 1.0) processes FASTQ formatted sequencing data, counts sequence frequency, ranks and sorts by abundance, calculates fold-enrichment (change in genotypic frequency across populations), clusters sequences based on a user-defined Levenshtein edit distance and enables searches for co-occurring nucleotide sequence motifs using degenerate nomenclature. Although the toolkit was initially designed for analysis of aptamer populations, the information it provides is equally applicable to the analysis of DNA-encoded sequence data from selection of nucleic acid and protein catalysts, surface display technologies (such as phage display) and selections for biological activity from mutant libraries. The software thereby fills a critical need for a stand-alone, readily-accessible, and easy-to-use informatics platform for HTS analysis of combinatorial selection outcomes, and it obviates the requirement for each lab to adapt or develop its own custom informatics pipeline to perform these tasks.

**Results**

**Overview**

FASTAptamer is a collection of modular Perl scripts written to be compatible across all UNIX-like systems (including Linux and Mac OS X, which both include Perl interpreters) and has no external dependencies. The software can also be used on Microsoft Windows platforms provided that a Perl interpreter is installed. (see Materials and Methods). Each script is designed as a module to address one of several informatic tasks that are standard for the combinatorial selections field. The toolkit can be used by nonbioinformaticians for user-friendly, stand-alone analysis. For advanced users, the plug-and-play nature of the toolkit, open source licensing, and standard file formats preserve the option of integrating the output into customized workflows and existing sequence analysis pipelines as well as further development of the toolkit. FASTAptamer makes use of the widely supported FASTA format to ensure maximal interoperability with existing software. The algorithms of FASTAptamer exploit the format’s description line (beginning with “>”) to identify each sequence uniquely with its corresponding data (such as rank within population, read number, and cluster information). The sequence line gives the actual entries using standard single-letter abbreviations for amino acid and nucleotide information. The analytical pipeline therefore begins with [FASTAptamer-Count](#), which assigns unique identifying information to the description line of each sequence entry for downstream analysis. From [FASTAptamer-Count](#), analysis can then continue within the toolkit to compare populations, to generate clusters of sequence families, to calculate fold-enrichment in one population relative to another, to search for sequence motifs (including degenerate or ambiguous motifs), or to proceed directly to other software tools.

The utility of the FASTAptamer toolkit was evaluated by analyzing two populations of RNA aptamers selected to bind HIV-1 Reverse Transcriptase (RT). The first population (70HRT,) was selected through 14 cycles of selection by nitrocellulose filter retention and native gel shift; this population has been described previously and thus served as a control for the analysis. The 70HRT population was subjected to one additional cycle of selection under slightly modified conditions to generate the second population (70HRT,) and both populations underwent HTS.

Raw FASTQ sequence files were preprocessed as described in Materials and Methods prior to use with the FASTAptamer toolkit. In general, the extent to which data files are processed after the sequencing run varies and often includes demultiplexing, merging paired-end reads, and filtering for high quality reads, all of which can be done with readily-available and widely-used software. For combinatorial selection libraries, these files are often additionally trimmed to remove fixed flanking regions prior to sequence analysis. An advantage of trimming before applying a quality filter is that it prevents unnecessary exclusion of sequences for which ambiguous base calls lie only within the constant regions. Phage display, protein mutagenesis selection, and other selections based on functional amino acid sequences can be analyzed by FASTAptamer either as nucleic acid sequence files or after translating to amino acid sequence (D. Salamando and M. Johnson, unpublished data). For the present study, a total of 2,160,216 sequences were obtained for 70HRT and 1,987,867 for 70HRT after pre-processing.

**FASTAptamer-Count**

The first steps in analyzing sequencing data from selections—whether by HTS or low-throughput sequencing—are the simple acts of counting the number of times each sequence is sampled from a population (alternately referred to as read count, sequence frequency, multiplicity, or copy number) and then ranking and sorting the sequences by abundance. Ordering in this manner allows for the downstream analysis of a subset of sequences based on multiplicity, rather than on a random capture of sequences by low-throughput sequencing methods. Normalizing the read count to the total number of reads in the population allows full data sets to be compared among multiple populations, even when the total numbers of reads varies between samples and sequencing runs.

For the 70HRT and 70HRT populations, these tasks were accomplished using [FASTAptamer-Count](#) (command: fastaptamer_count), the module that serves as the entry point into the FASTAptamer toolkit. [FASTAptamer-Count](#) enumerates the abundance of each identical sequence within a population file and ranks the sequence entries by decreasing abundance. Each preprocessed, FASTQ formatted input file is nondestructively output as a FASTA formatted plain-text file in which the sequences are sorted as nonredundant entries. The FASTA identifier line that precedes each sequence entry in the output file contains the rank, reads and normalized reads per million (RPM) of the respective sequence separated by dashes, values that are later used throughout the toolkit. Thus, for the 70HRT population, the most abundant
sequence (Rank = 1) was sampled 417,696 times, which corresponds to 193,358 RPM (Figure 1).

**FASTAptamer-Count** generates additional output in the form of a displayed summary report that includes the number of unique sequences and the total number of sequences, which can be used to determine library complexity and enrichment, as well as input and output file names and execution time. The FASTA formatted output of **FASTApptamer-Count** can be applied as input into downstream applications, including other **FASTAptamer** scripts or any software that accepts FASTA-formatted files as input. For highly-oversampled libraries (such as highly-enriched libraries from later selection cycles or libraries derived from low-complexity initial libraries), **FASTAptamer-Count** has the additional benefit of shrinking large file sizes to more manageable sizes for sharing and archiving. In the case of the 70HRT14 population, **FASTAptamer-Count** reduced the trimmed and filtered, 808.8 MB input FASTQ data file to a mere 6.2 MB output file in a matter of seconds using a standard laptop computer.

**Figure 1** **FASTAptamer-Count** generates a nonredundant FASTA-formatted output file that preserves sequence abundance metrics and remains compatible with downstream applications. A preprocessed FASTQ-formatted file serves as input into **FASTAptamer-Count** (top, left). Sequences are counted for abundance, normalized to total number of reads per million (RPM) and ranked according to decreasing abundance. Output from **FASTAptamer-Count** is a FASTA-formatted, plain-text file in which the identifier line is used to preserve sequence rank, reads and RPM information separated by dashes (middle). Output from **FASTAptamer-Count** serves as the input (yellow arrows) for downstream **FASTAptamer** modules or for other software capable of manipulating FASTA input (bottom).

**FASTAptamer-Compare**

Pairwise comparison of sequence distributions between two populations provides a snapshot of reproducibility between replicate selections or replicate sampling of selection, and it reveals changes in genotypic frequencies that result from cycle-to-cycle enrichment, from application of a subtractive or negative selection step, or from otherwise changing the selection conditions. **FASTAptamer-Compare** populates a single, tab-delimited, plain-text file listing each sequence that is present in both populations along with the corresponding RPM values from each population in adjacent columns (Figure 2a). From this output, generation of XY scatter plots using standard spreadsheet software provides a visual snapshot of how genotypic frequencies in one population compare with those in another. When the two replicate datasets for sequencing of the 70HRT14 population were processed through **FASTAptamer Compare** (command: fastaptamer_compare) and plotted against each other, the vast majority of the sequences clustered very closely to the diagonal (Figure 2b), as expected from unbiased replicate sampling of the same population. In contrast, plotting RPM values for the two sequential cycles of selection shows a large group of sequences that are much more loosely clustered around the diagonal and a second group that is shifted notably upward (Figure 2c), as expected from a population that continues to evolve.

Aggregate changes in genotypic distributions are often represented as histograms of log2 values of the ratio of normalized reads: log2(RPMy/RPMx). These values were calculated for the two 70HRT14 replicates and between the 70HRT14 and 70HRT15 populations using **FASTAptamer-Compare**. The script creates 100 bins spanning log2 values from −5 to +5 in intervals of 0.1 and populates each bin based on the number of calculated log2 values that fall within each bin interval. Values that fall outside of this range populate two additional bins on either end of the distribution (< −5 or > 5). The relatively high number of bins allows users the flexibility of reducing the number of bins easily without having to recalculate all values and facilitates the generation of a sequence distribution histogram, useful for comparing the distribution of sequences in two populations. As expected for a single population sequenced in replicate, the majority of sequence...
log2 values fall within −2 to +2 (Figure 2d), whereas the distribution was wider for a population that has undergone one additional round of selection as sequences were enriched or depleted (Figure 2e).

The default of FASTAptamer-Compare is to output information only for sequences that are present in both populations; however, users can invoke a command line option to include data for sequences that are present in only one population (Table 1). If this option is invoked, log2 values will not be calculated for those sequences to avoid divide-by-zero errors, and those sequences will be sent to output along with their corresponding nonzero RPM values. A summary report, including the input and output files and execution time, is displayed on screen upon completion of FASTAptamer-Compare.

### FASTAptamer-Cluster

Populations from combinatorial selections usually contain clusters of closely-related sequences. For selections in which the randomized regions are small relative to the library size, all possible variants can be sampled in the initial library (e.g., 1015 strands, each with 20 randomized nucleotides, yielding 420 ≈ 1012 possible sequences, or 1,000 copies of each on average). In such cases, the clusters reflect convergent mapping of sequence space onto the fitness landscape. For
long random regions and for small starting libraries, the probability becomes vanishingly small that two closely-related sequences would be independently sampled in the initial library. In such cases, clusters of closely-related sequences represent divergence from the initial seed sequences as the result of errors during polymerase chain reaction or other amplification steps. The accumulation of point mutants allows for the local exploration of sequence space by generating variants that were likely not present in the initial library, especially when intentional mutagenesis strategies are employed, such as doped-resynthesis of a seed sequence or low-fidelity amplification. In all of these cases, evolutionary trajectories within individual clusters and the aggregate trajectories of whole clusters are both informative. 

FASTAptamer-Cluster (command: fastaptamer_cluster) was used to parse the 70HRT\textsubscript{14} and 70HRT\textsubscript{15} populations (separately) into clusters of closely-related sequences, using the corresponding FASTAptamer-Count output files for each population. The script accomplishes this by considering Levenshtein edit distance (L), which is the total number of insertions, deletions, or substitutions required to transform one character string into another. In contrast to Hamming edit distance, which is utilized by SEWAL\textsuperscript{18} and AptaCluster\textsuperscript{22} and requires strings of equal length, Levenshtein edit distance can be calculated for sequences of unequal length that arise due to nucleotide insertions or deletions ("indels"). In the 70HRT\textsubscript{14} and 70HRT\textsubscript{15} libraries indels accounted for more than 36% of all unique sequences (40,898/113,441), etc.

### Table 1 Summary of FASTAptamer scripts and command line flags

| Script             | Function                                                                 | Command line flags | Interpretation                                                                 |
|--------------------|--------------------------------------------------------------------------|--------------------|--------------------------------------------------------------------------------|
| fastaptamer_count  | Determines abundance of each sequence, normalizes value to total reads per million, ranks and sorts by decreasing abundance. | -i Input file (.FASTQ)* | -o Output file (.FASTA)* -h Help screen -q Quiet mode—suppresses summary report |
| fastaptamer_compare| Calculates log\(_2\) values of RPM y/x, generates table containing RPM for each sequence in both files, generates and fills values for histogram of sequence distribution. | -x Input file 1 (.FASTA from FASTAptamer-Count)* -y Input file 2 (.FASTA from FASTAptamer-Count)* -o Output file (.TSV)* -h Help screen -a Output all sequences -q Quiet mode—suppresses summary report |
| fastaptamer_cluster| Generates sequence clusters based on a user-defined Levenshtein edit distance. | -i Input file (.FASTA from FASTAptamer-Count)* -o Output file (.FASTA)* -h Help screen -d Edit Distance* -f Read filter -q Quiet mode—suppresses summary report |
| fastaptamer_enrich | Calculates fold-enrichment values for each sequence in 2 or 3 populations. | -x Input file 1 (.FASTA from FASTAptamer-Count or FASTAptamer-Cluster)* -y Input file 2 (.FASTA from FASTAptamer-Count or FASTAptamer-Cluster)* -z Input file 3 (optional -.FASTA from FASTAptamer-Count or FASTAptamer-Cluster) -o Output file (.TSV)* -h Help screen -f RPM threshold filter -q Quiet mode—suppresses summary report |
| fastaptamer_search  | Degenerately searches for multiple sequence patterns across several files. | -i Input files(s) (.FASTA from FASTAptamer-Count, FASTAptamer-Cluster or other)* -o Output file (.FASTA) -p Pattern(s)* -help Help screen -highlight Highlight matched motifs -q Quiet mode—suppresses summary report |

*Required.
unclustered sequence then serves as the seed sequence for assembling the second cluster from the remaining sequences (Cluster #2), followed by the next most abundant unclustered sequence (Cluster #3), and so on. This process is iterated until every sequence is clustered.

Output from FASTAptamer-Cluster is a FASTA formatted file that appends new information to the identifier line for each sequence. In addition to rank, RPM and total reads information originally generated by FASTAptamer-Count, each sequence in the FASTAptamer-Cluster output file is additionally and uniquely identified by its cluster number, rank within that cluster and the Levenshtein edit distance required to transform it into the seed sequence. As with FASTAptamer-Count, values are separated by dashes. For example, the most abundant sequence in the 70HRT14 population is also a member of the first cluster to be populated (Cluster #1); it is the most abundant sequence within that cluster (rank within cluster = 1); and 0 edits are required to transform it into itself (L = 0). The second and third most abundant sequences in this cluster each lie at an edit distance of L = 1 from the seed sequence, and they are the fifth and seventh most abundant sequences within the population as a whole (Figure 3a). Part of the power of FASTAptamer is that analytics such as these are immediately accessible for any sequence within its identifier line, and they can serve to prioritize a given sequence or cluster for further analysis.

FASTAptamer-Cluster also generates a summary report listing each cluster generated and the number of unique sequences belonging to that cluster in terms of total reads and RPM. Plotting those values for the 70HRT14 population reveals the large number of closely-related sequence variants present within each of the top hundred or so clusters (Figure 3b). An elimination filter, similar to the “true-selected sequence” criteria employed by Thiel et al., can be invoked at the command line to prevent the clustering of sequences below a user-defined read(s) threshold (Table 1). This both improves execution time and reduces the size of the dataset by excluding singletons and low-copy reads that, in aggregate, dominate highly diverse libraries. Invoking this filter reduced clustering time for the 70HRT14 and 70HRT15 populations to minutes, rather than hours, on a standard desktop computer (Figure 3c).

For the 70HRT14 and 70HRT15 libraries, both of which carry 70 initially-random nucleotides, we found that a user-defined edit distance of seven, entered on the command line prior to program execution, captured essentially all members of each cluster, as there were very few members of any cluster separated from the seed sequence by an edit distance of more than five. This value also prevented independent clusters from overlapping, given the near-zero probability of two sequences matching at 63 of 70 positions by chance. Although intentional mutagenesis will increase the likelihood that sequences accumulate >10% divergence from the parent, we have found that allowing too much variation for aptamer libraries increases the sampling of unrelated folds that are structurally unrelated to the seed sequence.

**FASTAptamer-Enrich**

While sequence abundance alone can serve as useful criteria for identifying candidate molecules, the fold-enrichment
Table 2: Sample tab-delimited output from FASTAptamer-Enrich, sorted to show the most highly enriched sequences from 70HRT15

| Sequence | Length | Reads x | RPM x | Rank (x) | Reads y | RPM y | Rank (y) | Cluster (y) | Enrichment (y/x) |
|----------|--------|---------|------|---------|---------|-------|---------|-------------|-----------------|
| ACCAGAAAA... | 69     | 22,894  | 1    | 0.46    | 254     | 475   | 238.95  | 2           | 36              | 519.46          |
| ACCAGAAAA... | 69     | 22,894  | 1    | 0.46    | 429     | 246   | 123.75  | 2           | 60              | 269.02          |
| ACCAGATGAAA... | 69     | 22,894  | 1    | 0.46    | 437     | 241   | 121.24  | 2           | 64              | 263.57          |
| TCAAGAAAA... | 69     | 22,894  | 1    | 0.46    | 451     | 230   | 115.7   | 2           | 66              | 251.52          |
| ACCAGAAAA... | 69     | 14,649  | 2    | 0.93    | 283     | 412   | 207.26  | 2           | 41              | 222.86          |

Populations x and y correspond to 70HRT14 and 70HRT15 respectively.
FASTAptamer: A Bioinformatic Toolkit for Selections

Figure 4 FASTAptamer-Search allows for degenerate motif searching. (a) Output from FASTAptamer-Cluster was input into FASTAptamer-Search. The focused search output can then be used in downstream applications, such as multiple sequence alignment software or comparative sequence analysis tools for secondary structure prediction. (b) Searching the Round 15 population for the F1Pκ sequence motif using the patterns “UCCG” and “CGGGANAA” returns a list of matching sequences, some of which are seen here. Matched patterns (in bold) are enclosed with parentheses using FASTAptamer-Search’s highlight option for increased visibility.

useful for preprocessing and manipulating HTS data, the vast majority are geared towards genomic data analysis. The act of counting, ranking, and sorting a single selection population through the main set of Galaxy tools requires several discrete text-manipulation steps rather than the single-step processing offered by each module of FASTAptamer. While no public Galaxy server currently exists specifically for the combinatiorial selections field, local or private installations can make use of the FASTAptamer toolkit by installing the toolkit from the Galaxy ToolShed. The field will benefit from increased accessibility, decreased external dependencies and a universally compatible and open source software platform to build upon. FASTAptamer is designed for nonbioinformaticians with ease-of-use, accessibility and compatibility in mind, in line with accepted best practices and guidelines in developing bioinformatics software. The scripts that comprise the toolkit are capable of functioning independently, allowing for modular integration into existing pipelines or in structuring new workflows for more experienced users. In addition to universal platform compatibility (i.e., ready to use on virtually all UNIX-like systems, including Mac OS X), the output from FASTAptamer is designed to remain compatible with existing software by providing only tab-delimited, plain-text files or FASTA formatted files, for which numerous computational tools can be employed, such as multiple sequence alignment or structure prediction software. FASTAptamer-Search offers basic analysis tools for comparing populations and facilitates the generation of XY-scatter plots and histograms of sequence distribution. FASTAptamer-Cluster groups closely-related sequences into sequence families and provides information on each sequence’s rank within the cluster and its degree of relatedness to the seed sequence. FASTAptamer-Enrich simultaneously provide information on multiplicity, rank, clusters and enrichment, aiding informed decisions for candidate screening. Finally, FASTAptamer-Search provides basic functionality for sequence motif searching.

The FASTAptamer toolkit introduces a novel mechanism for generating clusters of closely-related sequences. In contrast to the fitness landscape constructed by Jimenez et al. to determine evolutionary pathways and the network analysis performed by Thiel et al. to establish the interconnectedness of all sequences in a population, FASTAptamer-Cluster utilizes edit distance to facilitate the candidate identification process for postselection screening. FASTAptamer-Cluster determines whether each sequence in a population is closely related to a highly-abundant “seed” sequence, rather than determining its relationship to the entire population. Therefore, each sequence in a population can only belong to a single cluster and candidate molecules can be identified either for being representative of a cluster, or for outperforming members of their cluster. FASTAptamer-Cluster is implemented using only core Perl modules and uses an exhaustive approach to clustering, rather than the heuristic, computationally-efficient algorithms used by SEWAL and AptaCluster. This tradeoff results in lengthier execution times but allows for consideration of indels. Larger and more diverse populations (such as those from early selection rounds) can require an impractical amount of time to cluster, as execution time is a function of the number of unique sequences in a population, overall diversity of sequences, length of random region and edit distance criteria. To address this limitation, FASTAptamer-Cluster allows users to implement a threshold filter based on a sequence’s total reads to exclude those sequences not highly sampled. Future implementation in a lower-level language or utilizing nonstandard Perl modules offers the potential of quicker processing, although at the cost of universal compatibility.

The FASTAptamer toolkit is intended to provide practitioners of combinatiorial selection a simple, easy-to-implement toolkit to facilitate operations that are essential to all classes of selections, such as counting, comparing, calculating fold-enrichment and searching. We encourage others to build upon the open source platform. Opportunities for future improvements include expanded statistical analysis within existing scripts and additional tools for analysis of amino acid sequences and for intracluster enrichment comparisons. Long-term goals for FASTAptamer include improvements to the algorithm and execution speed (particularly for FASTAptamer-Cluster) and a graphical user interface or Galaxy based web server to eliminate the requirement for command line operation. FASTAptamer is available to download at http://burkelab.missouri.edu/fastaaptamer.html and is also available on GitHub and through the Galaxy ToolShed.

Materials and methods

In vitro selection. A previously reported 70N RNA aptamer library, enriched through 14 rounds for aptamers that bind RT from HIV-1 strain BH10 (Group M: Subtype B), was subjected to an additional round of in vitro selection against RT from the closely-related HXB2 strain, with slight modifications. In short, 200 pmol of the transcribed library was resuspended in 100 μl binding buffer (150 mmol/l KCl, 10 mmol/l MgCl₂, 50
mmol/l Tris-HCl, pH 7.5), heated to 95 °C and cooled on ice. The RNA library was incubated with a thin strip of nitrocellulose (~3 x 20 mm) for several minutes and removed to subtract nonspecific binders. 40 pmol of RT was then added, and the mixture was incubated on ice for an additional 20 minutes. Separately, a nitrocellulose filter was prewet with 2 ml binding buffer. Immediately before application of incubated RNA:RT complex, the nitrocellulose filter was washed again with 1 ml of binding buffer. RNA:RT complex was then applied to filter under vacuum and washed with 1 ml of binding buffer. Suction continued for 10 minutes to allow the filter to dry. The filter was then removed and incubated in extraction buffer (10 mmol/l EDTA, 50 mmol/l NaCl, 8 mol/l urea). RNA was recovered by phenol/chloroform extraction and ethanol precipitation. Recovered RNA was reverse transcribed using ImProm-II Reverse Transcriptase (Promega, Madison, WI) and polymerase chain reaction amplified for HTS.

High-throughput sequencing and data preprocessing pipeline. Libraries were prepared for sequencing as previously described using a series of polymerase chain reaction steps to add Illumina adapters and sequencing indices for multiplexing. Sequencing was performed on an Illumina HiSeq2000 (University of Missouri DNA Core Facility). Data preprocessing was performed using cutadapt to trim 5' and 3' constant regions from sequences and to discard any uncut sequences or sequences not between 67 and 73 nucleotides long after trimming. Trimmed sequences were then filtered for high-quality reads by eliminating entire sequences if a single position had a Phred quality score less than 20 using FASTQ quality filter from a locally installed copy of the FASTX-Toolkit (http://hannonlab.cshl.edu/fastx_toolkit/). Both the 70HRT14 and 70HRT15 libraries were sequenced in duplicate and later merged using the UNIX program “cat.”

Implementation. The FASTAptamer toolkit is written in Perl 5 with no external dependencies and is designed to operate on any modern Unix-like system (including Linux and Mac OS X). The FASTAptamer toolkit has been tested using Mac OS X 10.5+, CentOS Linux 5.4 (Lewis Cluster—University of Missouri Bioinformatics Consortium) and Debian GNU/Linux 7.0. FASTAptamer can also run on a Microsoft Windows platform provided that a Perl interpreter has been installed, such as ActiveState Perl or Strawberry Perl. Installation and use of the FASTAptamer toolkit assumes only a basic working knowledge of command line operation. A tutorial using the sample data and with examples of properly formatted commands, including the use of command line options, is included in the user's guide that accompanies the software.

The scripts that comprise the toolkit make use of Perl regular expressions (pattern-matching strings) to identify FASTAQ/FASTA entries or sequence motifs (for FASTAptamer-Search) in input files. Many of the scripts function by creating a Perl variable for each population, known as a hash, which associates a unique key (the sequence) with a value. For FASTAptamer-Count, each unique sequence generates a corresponding key whose value is incremented with each occurrence of the sequence in the input file. The hash variables in FASTAptamer-Count are sorted into arrays by count values, in order of decreasing abundance, and then sent to output with a ranking. FASTAptamer-Count utilizes standard competition ranking, where tied values are given the same rank and leave a gap in the rankings (e.g., if two values are tied for the second-most abundance the rankings would be scored as 1, 2, 2, 4).

FASTAptamer-Compare and FASTAptamer-Enrich utilize the hash variable in a similar manner to FASTAptamer-Count, in that each sequence serves as a key, but for FASTAptamer-Compare and FASTAptamer-Enrich, the paired hash value stores a string of information (such as rank, reads, and RPM). An algorithm in both scripts iterates through the list of keys for the population z (for three input files in FASTAptamer-Enrich) and/or y hash and tests for the presence of the same key in the other hash variable(s). For sequences present in another population, the matching hash key and values are sent to output and removed from both hashes. In FASTAptamer-Compare, invoking the command to include all sequences results in the emptying of hashes to output for which no match in the sequence key was found.

FASTAptamer-Cluster utilizes a regular expression designed to match the two-line output generated by FASTAptamer-Count, where each pattern match creates a new entry in a Perl array variable. Several subroutines are included in the algorithm to iterate through all sequence entries, set up a distance matrix comparing two strings and calculate the Levenshtein edit distance. For each cluster, the first entry in the array (the “seed sequence”), which contains the entry of the most abundant sequence, is split into two parts containing the sequence metrics in the description line and the sequence itself. The next entry in the array is then similarly split, and both sequence strings are sent to a subroutine to calculate the degree of similarity. For sequences that have an edit distance equal to or less than the user-defined distance, the sequence entry is sent to output with cluster data appended to the description line. For sequences that require additional transformations to meet the threshold, the sequence is redirected to a separate array to be processed in the next cluster. This process is iterated until every sequence has been compared to the seed sequence and until all sequences have been clustered. Clustering performance was determined using a desktop computer running Debian GNU/Linux 7.0 and powered by an AMD Phenom X3 8750 processor with 4 GB of RAM.

FASTAptamer supports the International Union of Pure and Applied Chemistry and International Union of Biochemistry and Molecular Biology (IUPAC-IUBMB) sequence nomenclature for amino acid and nucleotide information.

Availability and license. The FASTAptamer toolkit, user’s guide, and sample data can be downloaded from http://burkelab.missouri.edu/fastapamer.html, and has been deposited on GitHub (http://github.com/FASTAptamer) for archival, version control, and open source development. Additionally, FASTAptamer is available in the Galaxy ToolShed under the “Combinatorial Selections” category. FASTAptamer is distributed under a GNU General Public License version 3.0.

Acknowledgments. Funding was provided by a graduate research assistantship through the University of Missouri Department of Biochemistry to K.K.A. and through grants...
Molecular Therapy—Nucleic Acids

1. Wang, AZ and Farokhzad, OC (2014). Current progress of aptamer-based molecular imaging. J Nucl Med 55: 353–356.

2. Sundaram, P, Kurniawan, H, Byrne, ME and Wower, J (2013). Therapeutic RNA aptamers in clinical trials. Eur J Pharm Sci 48: 259–271.

3. Fowler, DM, Araya, CL, Fleishman, SJ, Kellogg, EH, Stephany, JJ, Baker, D et al. (2010). High-resolution mapping of protein sequence-function relationships. Nat Methods 7: 741–746.

4. Pitt, JN and Ferré-D’Amare, AR (2010). Rapid construction of empirical RNA fitness landscapes. Science 330: 378–379.

5. Ameta, S, Winz, ML, Previ, C and Jascho, A (2014). Next-generation sequencing reveals how RNA catalysts evolve from random space. Nucleic Acids Res 42: 1303–1310.

6. Jiménez, JI, Xuvi-Brunet, R, Campbell, GW, Turk-MacLeod, R and Chen, IA (2013). Comprehensive experimental fitness landscape and evolutionary network for small RNA. Proc Natl Acad Sci USA 110: 14981–14989.

7. Vaidya, N, Manapat, ML, Chen, IA, Xuvi-Brunet, R, Hayden, EJ and Lehman, N (2012). Spontaneous network formation among cooperative RNA replicators. Nature 491: 72–77.

8. Jolma, A, Kovivc, T, Toivanen, J, Cheng, L, Wei, G, Enge, M et al. (2010). Multiplexed massively parallel SELEX for characterization of human transcription factor binding specificities. Genome Res 20: 867–873.

9. Cho, M, Xiao, Y, Nie, J, Stewart, R, Eisenstein, M, Chambers, J et al. (2013). Quantitative selection and parallel characterization of aptamers. Proc Natl Acad Sci USA 110: 18460–18465.

10. Thiel, WH, Bair, T, Peek, AS, Liu, X, Dassie, J and Stockdale, KR et al. (2012). Delivery of chemo-sensitizing siRNAs to HER2+-breast cancer cells using RNA aptamers. Nucleic Acids Res 40: 6319–6337.

11. Thiel, WH, Bair, T, Peek, AS, Liu, X, Dassie, J, Stockdale, KR et al. (2012). Rapid identification of cell-specific, internalizing RNA aptamers with bioinformatics analyses of a cell-based aptamer selection. PLoS One 7: e43836.

12. Szeto, K, Latulippe, DR, Ozar, A, Pagano, JM, White, BS, Shalloway, D et al. (2013). RAPID-SELEX for RNA aptamers. PLoS One 8: e52687.

13. Held, DM, Greathouse, ST, Agrawal, A and Burke, DH (2003). Evolutionary landscapes for the acquisition of new ligand recognition by RNA aptamers. J Mol Evol 57: 299–308.

14. Tuerk, C, MacDougall, S and Gold, L (1992). RNA pseudoknots that inhibit human immunodeficiency virus type 1 reverse transcriptase. Proc Natl Acad Sci USA 89: 6898–6902.

15. Dias-Neto, E, Nunes, DN, Giordano, RJ, Sun, J, Botz, GH, Yang, K et al. (2009). Next-generation phage display: integrating and comparing available molecular tools to enable cost-effective high-throughput analysis. PLoS One 4: e8338.

16. Girardin, B, Riemer, C, Hardison, RC, Burhans, R, Elnitski, L, Shah, P et al. (2008). Galaxy: a platform for interactive large-scale genome analysis. Genome Res 18: 1451–1455.

17. Goeds, J, Nekrutken, A and Taylor, J, Galaxy Team (2010). Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. Genome Biol 11: R66.

18. Kheradpour, D, Von Kuster, G, Bouvier, E, Baker, D, Aftan, E, Stoler, N et al. (2010). Galaxy Team. (2010). Dissemination of scientific software with Galaxy ToolShed. Genome Biol 11: 403.

19. Code share. (2014) Nature 514: 536.

20. Pric, A and Procter, JB (2012). Ten simple rules for the open development of scientific software. PLoS Comput Biol 8: e1002802.

21. Seemann, T (2013). Ten recommendations for creating usable bioinformatics command line software. Gigascience 2: 15.

22. Sandve, GK, Nekrutken, A, Taylor, J and Hovig, E (2013). Ten simple rules for reproducible computational research. PLoS Comput Biol 9: e1003286.

23. Lepretrost, Filipa, V, Barbosa, VC, Franco, CL, Perez-Perez, Y and Carvalho, PC (2014). On best practices in the development of bioinformatics software. Front Genet 5: 199.

24. Martin, M (2011). Cutadapt removes adapter sequences from high-throughput sequencing reads. EMBnet J 17: 10–12.

25. Cock, PJ, Fields, CJ, Goto, N, Heuer, ML and Rice, PM (2010). The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. Nucleic Acids Res 38: 1767–1771.