

SOFTWARE

Open Access



# FlashFry: a fast and flexible tool for large-scale CRISPR target design

Aaron McKenna<sup>1\*</sup>  and Jay Shendure<sup>1,2</sup>

## Abstract

**Background:** Genome-wide knockout studies, noncoding deletion scans, and other large-scale studies require a simple and lightweight framework that can quickly discover and score thousands of candidate CRISPR guides targeting an arbitrary DNA sequence. While several CRISPR web applications exist, there is a need for a high-throughput tool to rapidly discover and process hundreds of thousands of CRISPR targets.

**Results:** Here, we introduce FlashFry, a fast and flexible command-line tool for characterizing large numbers of CRISPR target sequences. With FlashFry, users can specify an unconstrained number of mismatches to putative off-targets, richly annotate discovered sites, and tag potential guides with commonly used on-target and off-target scoring metrics. FlashFry runs at speeds comparable to commonly used genome-wide sequence aligners, and output is provided as an easy-to-manipulate text file.

**Conclusions:** FlashFry is a fast and convenient command-line tool to discover and score CRISPR targets within large DNA sequences.

**Keywords:** CRISPR, Guide library, Cas9, Cpf1, Off-target, On-target, Deletion scan

## Background

The CRISPR prokaryotic immune system has transformed genome engineering. As typically used, CRISPR proteins are directed to create double-stranded DNA breaks at location(s) in a genome matching a specified guide sequence [1]. These double-stranded breaks are commonly repaired by a non-homologous end joining (NHEJ) pathway, which can leave small insertions or deletions (indels) at the genomic target site. These site-specific, targeted indels can be used to perturb endogenous gene function [2], encode information [3], or characterize the function of genomic sequence [4–6].

Although CRISPR editing is specific [7], not all guides function with the same efficiency or specificity. For instance, double-stranded breaks can occur at genomic locations (“targets”) that are an imperfect match to the supplied guide sequence (termed “off-targets”). To reduce the chance of such unintended genome editing, guide sequences can be chosen that contain less overlap with all possible alternative targets in the genome. The importance

of specific differences in the guide sequence, the genomic location and chromatin environment of the target, and the method of guide delivery all appear to affect the distribution and rate of this off-target cutting [8].

To help users choose both specific and active guide sequences, the community has created a large number of CRISPR target selection tools, most of which are made available as web applications [8–10]. Such tools are convenient for researchers screening a small set of guides or scanning a single genomic locus like an exon. Unfortunately, these tools require batched queries for large sets, which makes it more challenging to scan loci or whole genomes for guides. Additionally, some guide screening tools rely upon genome-wide alignment tools to generate putative off-target lists for each guide. For practical reasons, these aligners are generally designed to quickly discover only the most similar sequences with a limited number of mismatches in comparison to the guide (typically  $\leq 3$ ), whereas experimental efforts have shown activity at off-target sequences containing upwards of six mismatches to the guide [11]. Some of these tools also miss a subset of potential off-targets altogether, regardless of the mismatch distance [12]. Lastly, end users may need to design guides to non-model organisms,

\* Correspondence: [aaronmck@uw.edu](mailto:aaronmck@uw.edu)

<sup>1</sup>Department of Genome Sciences, University of Washington, Seattle, WA, USA

Full list of author information is available at the end of the article



engineered DNA sequences, or genomic sequences that contain single nucleotide variants [13]. To address these issues, we have created FlashFry, a command-line tool for discovery and characterization of CRISPR guide sequences from arbitrary genomic regions.

## Implementation

### Database creation

FlashFry generates a block-compressed binary database of all potential target sequences within a given reference sequence. This FASTA-formatted sequence can be the genome of a canonical model organism, the transcriptome of a new species, or a custom-built reference that integrates strain or cell-line-specific variants. This database is generated for a specific CRISPR enzyme and protospacer adjacency motif (PAM) combination and can be constructed in approximately an hour on a standard computer (Table 1). In this database, sequences that contain the required CRISPR PAM sequence are organized into sorted prefix-bins (Fig. 1). This prefix length can be specified at runtime, with larger bins being automatically sub-indexed to reduce lookup times. The resulting data structure can then be quickly searched by comparing a target sequence's prefix against the bin's sequence. To save space and to further improve search times, target sequences and their number of occurrences within the genome are stored as a 64 bit-encoded value within each bin. Each target sequence is then followed by additional binary-encoded values for each position within the genome. These bins are compressed using the HTSJDK library [14], and an index is created for each bin's offset within this file.

### Search

Given the inherent inefficiencies of high-mismatch searches, FlashFry uses a filtering approach to find candidate off-targets. It does so by precomputing a traversal over target bins with less than  $k$  mismatches to each guide in the candidate (such filtering approaches are well reviewed in Navarro et al. [15]). Each target sequence is compared to the prefix of every bin, and only those bins with less than or equal to  $k$  mismatches are recorded as potential search locations. The list of visited bins then

aggregated for all targets, these bins are loaded sequentially, and the genomic target sites within are compared against each target with a matching prefix. When a large number of bins are to be searched (> 95% of all bins), which is common with large guide screens or with a high  $k$  mismatch threshold, FlashFry will instead search the full database to avoid the overhead cost of disk seeks.

To further reduce search times, FlashFry uses bit parallelism when determining mismatches between the binary-encoded target and candidate genomic matches [15]. Bit parallelism uses bit-shifting and optimized processor instructions to compare the binary representations of nucleotide strings in a small number of compute operations, avoiding a character by character string comparison. FlashFry is currently compatible with target sequences up to 24 bases in length, although it could be expanded to longer target sequences, as our bin storage approach allows recovery of the prefix of a target sequence (allowing target sequences of length 24 plus the prefix length). Lastly, off-target discovery is halted for candidate guides that have exceeded a user-defined number of off-target hits, saving compute time by eliminating poor candidates early from the putative target pool. This off-target limit can be set by the user, defaulting to 2000 off-targets.

We then compared the runtime and memory usage of FlashFry to the commonly used CRISPR characterization tools Cas-OFFinder and CRISPRseek [16, 17]. We also included BWA in this comparison, which is used as a backend search tool for some CRISPR web applications [8, 18]. For all tools, per-target search times decrease with the number of targets and the allowed number of mismatches (Fig. 2a). FlashFry and BWA run approximately two to three orders of magnitude faster than the conventional CRISPR target discovery tools. FlashFry outperforms BWA for larger target sets and at higher allowed mismatches between the candidate guide and its genomic targets, though at the cost of higher memory usage for very large target sets (Fig. 2a, b).

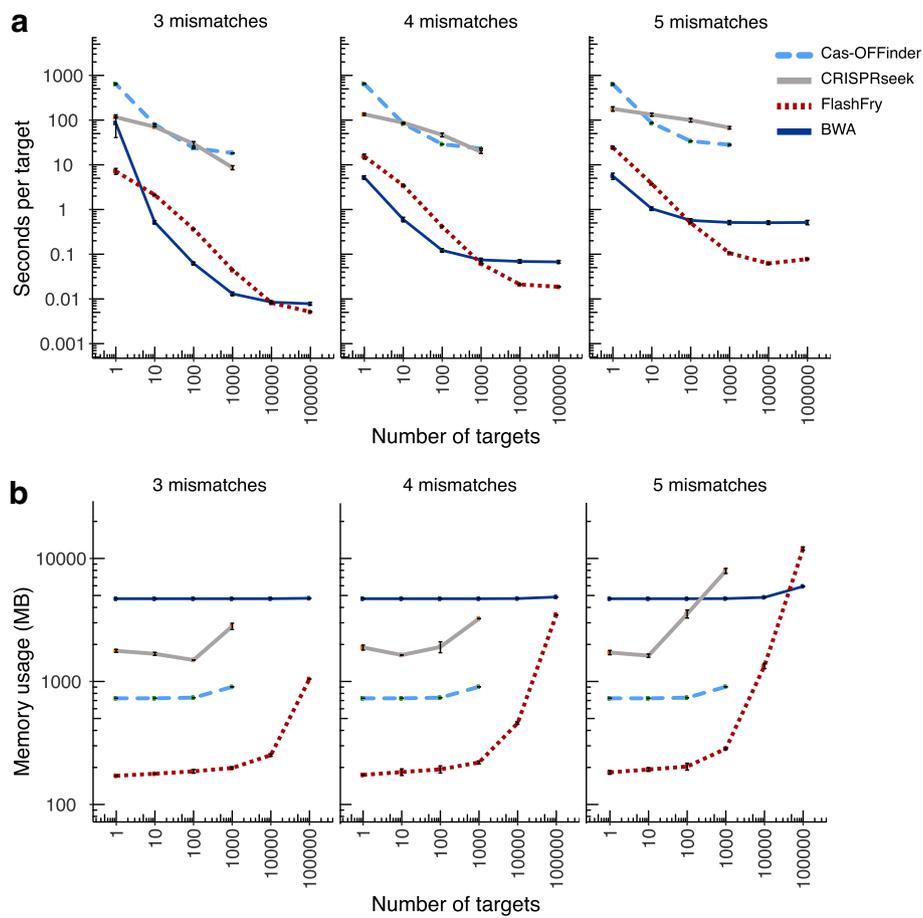
### Guide characterization and scoring

The goal of a typical biologist is to pick a subset of highly active and specific CRISPR guide sequences from

**Table 1** A sample of computational times (h:m:s) required to build a FlashFry database for versions of the *Caenorhabditis elegans*, human, mouse, and *Drosophila melanogaster* genomes for common CRISPR enzymes. All timing analyses were run with default FlashFry parameters on an Amazon r4.large instance, limited to 8 GB of memory limited and one CPU core

Genome	CRISPR/Cas9 (NGG)	CRISPR/Cas9 (NGG/NAG)	Cpf1 (TTTN)
<i>Caenorhabditis elegans</i> —235	0:1:28	0:2:55	0:2:13
Human—hg38	1:15:15	2:42:27	0:56:23
Mouse—mm10	1:02:13	2:21:11	0:43:13
<i>Drosophila melanogaster</i> —BDGP6	0:2:38	0:5:01	0:2:14





**Fig. 2** Comparison of the runtimes and memory usage of common CRISPR target discovery tools over an increasing number of targets and permitted mismatches. Five random CRISPR guide sets were run for each target-count (x-axis) and permitted mismatch level (y-axis). Plotted are the mean runtime with standard deviation bars for each set of replicates. **a** Running time per sequence for increasing numbers of target sites and **b** their corresponding memory usage. FlashFry benefits from aggregating all guide-to-genome comparisons in one pass of the database, matching BWA's performance at hundreds of targets for five mismatches, and thousands of targets at four mismatches. Only BWA and FlashFry were run for the 10,000 and 100,000 target searches

exons. These filtered candidate libraries can be easily processed into the format required by a wide variety of custom microarray manufacturers.

### Conclusions

The needs of genome-wide knockout studies, noncoding deletion scans, and other large-scale studies or method development projects are unfortunately not well-met by the abundant CRISPR web applications. FlashFry, an efficient and flexible toolset, fills this void and can be used to rapidly discover and characterize tens to hundreds of thousands of guides from an arbitrary sequence quickly and with a relatively low memory footprint. For method developers, we also expose a simple interface for implementing additional scoring schemes, given the sequence context of a target and its off-target hits within the genome. FlashFry has no system dependencies outside

of the JVM and avoids many of the configuration pitfalls and complexity of tools that rely on genome aligners to discover off-target sequences. Documentation, code, and tutorials are available on the FlashFry GitHub website.

### Methods

#### Software availability

FlashFry is written in Scala and bundled as a single stand-alone Jar file, easily run on any system with an installed Java virtual machine (JVM). The tool is freely licensed under version 3 of the GPL, and code, documentation, and tutorials are available on the GitHub page: <https://aaronmck.github.io/FlashFry/>

#### Database generation times

We timed FlashFry's database creation on an Amazon's Elastic Compute Cloud r4.large instance type using an

SSD filesystem. FlashFry was limited to one CPU with the ‘taskset -c 0’ command and 8GB of memory using the java “-Xmx8g” argument.

### Off-target tool comparisons

Runtime and memory comparisons were run on Amazon’s Elastic Compute Cloud using the r4.large instance type with an Intel Xeon CPU at 2.30 GHz. Each machine instance was setup using a Docker configuration file, which is provided within our GitHub codebase. Each tool was limited to a single processor with the “taskset” Linux command, and compute and memory usage were recorded with the “time -v” command. The full pipeline is available in the GitHub repository, along with the timing results of individual runs. Each guide-count and permitted mismatch (3,4, and 5) value was replicated five times using the human HG38 reference. Guide counts of 1, 10, 100, and 1000 were run for all tools. These random guides were generated using FlashFry’s “random” analysis module with the “—onlyUnidirectional” flag set to ensure a single candidate per sequence. Additionally 10,000 and 100,000 guide iterations were run for FlashFry and BWA, but were not run with Cas-OFFinder and CRISPRseek for practical reasons. Individual tool configurations are detailed below.

### FlashFry

FlashFry version 1.8.1 was run in discovery mode with java option “-Xmx8g” for 1–1000 target runs, and “-Xmx15g” for 10,000 and 100,000 target searches. Mismatches were set with the “-maxMismatch” command, and defaults were used for all remaining parameters.

### BWA

BWA runtime includes the initial alignment step (aln) and mapping to genomic coordinates (samse). BWA aln was run with parameters taken from Haeussler et al. [8]: `aln -o 0 -m 20000000 -n <mismatch_count> -k <mismatch_count> -N -l 20 <humanRef>`, using BWA commit tag e624290ad42f6c1deea87332337b08302faece48 from the following repository: <https://github.com/lh3/bwa>.

### Cas-OFFinder

A custom script was used to convert the random target FASTA file into the Cas-OFFinder input with the appropriate mismatch setting, using the 20(N)NGG search string. This conversion time was not towards Cas-OFFinder’s runtime or memory usage. Cas-OFFinder requires a custom Linux kernel driver supporting OpenCL to be installed on the machine, and our Docker instance pre-configures Intel’s OpenCL version 2017\_7.0.0.2568\_x64. Cas-OFFinder was then run using the CPU “C” option against the input file.

### CRISPRseek

We ran CRISPRseek using a custom R script and the “Rscript” command-line tool (the associated code is available in our GitHub repository). Timing data includes loading the relevant libraries and resources, executing the off-target search, and as it was impossible to separate discovery of off-targets and scoring, the scoring of guides.

### Cancer gene census calculations

All processing was done on a standard Amazon AWS r4.xlarge compute instance with an Intel Xeon CPU at 2.30 GHz. The cancer gene census (CGC) dataset, version 83 was downloaded from the CGC portal [20]. Intervals were generated using custom Scala scripts capturing the RefSeq exonic sequence of each gene using the model with the largest number of exons, adding 10 bases up and downstream. The corresponding genomic sequences were extracted from the human HG19 genome using Picard’s ExtractSequences (<https://broadinstitute.github.io/picard/>). Sites were then discovered and scored using FlashFry with a maximum of four mismatches to off-targets, and a maximum of 2000 off-target sequences per candidate. Off-target scoring was run with 30 GB of memory, taking 15,024.70 s. The Hsu et al. off-target scoring scheme [19] and the Moreno-Mateos and Vejnar et al. [10] on-target metric were run against the 426,560 sites, and an aggregate ranking was produced by supplying FlashFry the “rank” scoring option. The rank option will produce a rank-ordered assignment for each target based on the median rank of individual scores and will additionally use the Schulze method to rank the top 1000 targets [21]. Lastly, the best and second-best hits per individual exon and gene were calculated with Python and R scripts, available in the GitHub repository.

### Availability and requirements

Project name: FlashFry

Project home page: <http://aaronmck.github.io/FlashFry/>

Operating system(s): Platform independent

Programming language: Scala/Java/JVM

Other requirements: Java 1.8 or higher

License: GNU GPL v3

Any restrictions to use by non-academics: None

### Acknowledgements

We thank the members of the Shendure lab for their feedback, suggestions, and critical reading of the manuscript, especially Molly Gasperini, Vikram Agarwal, and Martin Kircher.

### Funding

This work was supported by grants from the Paul G. Allen Family Foundation (JS), an NIH Directors Pioneer Award (JS; DP1HG007811). AM was supported by a fellowship from the NIH/NHLBI (T32HL007312). JS is an investigator of the Howard Hughes Medical Institute.

### Availability of data and materials

All scripts, code, tools, and data are available on the FlashFry website: <http://aaronmck.github.io/FlashFry/>.

**Authors' contributions**

AM developed the method; AM and JS wrote the manuscript. Both authors read and approved the final manuscript.

**Ethics approval and consent to participate**

Not applicable.

**Consent for publication**

All authors have read the manuscript and approve of its publication.

**Competing interests**

The authors declare that they have no competing interests.

**Publisher's Note**

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Author details**

<sup>1</sup>Department of Genome Sciences, University of Washington, Seattle, WA, USA. <sup>2</sup>Howard Hughes Medical Institute, Seattle, WA, USA.

Received: 20 May 2018 Accepted: 22 June 2018

Published online: 05 July 2018

**References**

- Wright AV, Nuñez JK, Doudna JA. Biology and applications of CRISPR systems: harnessing nature's toolbox for genome engineering. *Cell*. 2016;164:29–44.
- Wang X, He L, Goggin S, Saadat A, Wang L, Claussnitzer M, Kellis M. High-resolution genome-wide functional dissection of transcriptional regulatory regions in human. *bioRxiv* 193136; <https://doi.org/10.1101/193136>.
- McKenna A, Findlay GM, Gagnon JA, Horwitz MS, Schier AF, Shendure J. Whole-organism lineage tracing by combinatorial and cumulative genome editing. *Science*. 2016;353:aaf7907.
- Gasperini M, Findlay GM, McKenna A, Milbank JH, Lee C, Zhang MD, et al. CRISPR/Cas9-mediated scanning for regulatory elements required for HPRT1 expression via thousands of large, programmed genomic deletions. *Am J Hum Genet*. 2017;101:192–205.
- Chen S, Sanjana NE, Zheng K, Shalem O, Lee K, Shi X, et al. Genome-wide CRISPR screen in a mouse model of tumor growth and metastasis. *Cell*. 2015;160:1246–60.
- Liu SJ, Horlbeck MA, Cho SW, Birk HS, Malatesta M, He D, et al. CRISPRi-based genome-scale identification of functional long noncoding RNA loci in human cells. *Science*. 2017;355:eaah7111.
- Doench JG, Hartenian E, Graham DB, Tothova Z, Hegde M, Smith I, et al. Rational design of highly active sgRNAs for Crispr-Cas9-mediated gene inactivation. *Nat Biotechnol*. 2014;32:1262–67.
- Haeussler M, Schönig K, Eckert H, Eschstruth A, Mianné J, Renaud J-B, et al. Evaluation of off-target and on-target scoring algorithms and integration into the guide RNA selection tool CRISPOR. *Genome Biol*. 2016;17:148.
- Labun K, Montague TG, Gagnon JA, Thyme SB, Valen E. CHOPCHOP v2: a web tool for the next generation of CRISPR genome engineering. *Nucleic Acids Res*. 2016;44:W272–6.
- Moreno-Mateos MA, Vejnar CE, Beaudoin J-D, Fernandez JP, Mis EK, Khokha MK, et al. CRISPRscan: designing highly efficient sgRNAs for CRISPR-Cas9 targeting in vivo. *Nat Methods*. 2015;12:982–8.
- Tsai SQ, Zheng Z, Nguyen NT, Liebers M, Topkar W, Thapar V, et al. GUIDE-seq enables genome-wide profiling of off-target cleavage by CRISPR-Cas nucleases. *Nat Biotechnol*. 2015;33:187–97.
- Doench JG, Fusi N, Sullender M, Hegde M, Vaimberg EW, Donovan KF, et al. Optimized sgRNA design to maximize activity and minimize off-target effects of CRISPR-Cas9. *Nat Biotechnol*. 2016;34:184–91.
- Canver MC, Lessard S, Pinello L, Wu Y, Ilboudo Y, Stern EN, et al. Variant-aware saturating mutagenesis using multiple Cas9 nucleases identifies regulatory elements at trait-associated loci. *Nat Genet*. 2017;49:625–34.
- HTSJDK: A Java API for high-throughput sequencing data (HTS) formats, version 2.8.1 [Computer software]. 2018. Retrieved from: <https://github.com/samtools/htsjdk>.
- Navarro G. A guided tour to approximate string matching. *ACM Computing Surveys ACM*. 2001;33:31–88.
- Zhu LJ, Holmes BR, Aronin N, Brodsky MH. CRISPRseek: a bioconductor package to identify target-specific guide RNAs for CRISPR-Cas9 genome-editing systems. de Crécy-Lagard V, editor. *PLoS One* 2014;9:e108424–e108427.
- Bae S, Park J, Kim JS. Cas-OFFinder: a fast and versatile algorithm that searches for potential off-target sites of Cas9 RNA-guided endonucleases. *Bioinformatics*. 2014;30:1473–5.
- Li H, Durbin R. Fast and accurate long-read alignment with Burrows-Wheeler transform. *Bioinformatics*. 2010;26:589–95.
- Hsu PD, Scott DA, Weinstein JA, Ran FA, Konermann S, Agarwala V, et al. DNA targeting specificity of RNA-guided Cas9 nucleases. *Nat Biotechnol*. 2013;31:827–32.
- Futreal PA, Coin L, Marshall M, Down T, Hubbard T, Wooster R, et al. A census of human cancer genes. *Nat Rev Cancer*. 2004;4:177–83.
- Markus S. A new monotonic and clone-independent single-winner election method. *Voting Matters*. 2003;17:9–19.

**Ready to submit your research? Choose BMC and benefit from:**

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

At BMC, research is always in progress.

Learn more [biomedcentral.com/submissions](https://biomedcentral.com/submissions)

