PinAPL-Py: a web-service for the analysis of CRISPR-Cas9 Screens

Philipp N. Spahn[1,2], Tyler Bath[4], Ryan J. Weiss[3], Jihoon Kim[4], Jeffrey D. Esko[3], Nathan E. Lewis[1,2,*] & Olivier Harismendy[4,5*]

[1]School of Medicine, Department of Pediatrics, [2]Novo Nordisk Foundation Center for Biosustainability at UCSD, [3]Glycobiology Research and Training Center, Department of Cellular and Molecular Medicine, [4]Division of Biomedical Informatics, [5]Moores Cancer Center, Department of Medicine, University of California San Diego, La Jolla, CA

*To whom correspondence should be addressed.

Abstract

Summary: Large-scale genetic screens using CRISPR/Cas9 technology have emerged as a major tool for functional genomics. With its increased popularity, experimental biologists frequently acquire large sequencing datasets for which they often do not have an easy analysis option. While a few bioinformatic tools are available, their installation and use typically require bioinformatic expertise. To make sequencing data analysis more accessible to a wide range of scientists, we developed a Platform-independent Analysis of Pooled Screens using Python (PinAPL-Py), which we present as an intuitive web-service. PinAPL-Py implements state-of-the-art tools and statistical models, assembled in a comprehensive workflow covering alignment, quality control, enrichment/depletion analysis and gene ranking. The workflow supports multiple libraries, and offers different analysis options for read count normalization or gene-ranking methods. Other technical parameters can be easily adjusted to allow greater flexibility and customization. Thus PinAPL-Py provides high-quality data analysis in an easily accessible service.

Availability and implementation: PinAPL-Py is freely accessible at pinapl-py.ucsd.edu with instructions, documentation and test datasets. Experienced users can run PinAPL-Py on their local machine using the Docker image (oncogx/pinaplpy_docker). Documentation can be found on GitHub at https://github.com/LewisLabUCSD/PinAPL-Py.

Supplementary information: Supplementary information for this article is available online.

Contact: oharismendy@ucsd.edu, nlewisres@ucsd.edu

Introduction

Genetic screens using pooled CRISPR/Cas9 libraries are functional genomics tools that are becoming increasingly popular throughout the life sciences to find novel molecular mechanisms and understand complex cellular systems (Opdam *et al.*, 2017). Despite the availability of a few bioinformatic solutions (Li *et al.*, 2014; Hart and Moffat, 2016; Winter *et al.*, 2015), the sequencing analysis remains challenging for most laboratories since installation and execution can require a bioinformatic expert. Here we introduce PinAPL-Py, a comprehensive analysis workflow, optimized for transparency and user-friendly operation. PinAPL-Py is run on a web-server through an intuitive interface and, thus, makes no requirements to either the user's skill or computer platform. This facilitates standardized, reproducible data analysis that can be carried out directly by the scientists conducting the experiments.

Description

User input: To start a PinAPL-Py run, the user enters a project name and an email address to receive start and completion notifications. The sequence read files (.fastq.gz) are then efficiently uploaded using on-the- fly compression by web-workers technology (Kim *et al.*, 2014). Next, the user assigns read files to the experimental condition (treatment or control/untreated). The provision of at least one control sample is required. In contrast to other tools, PinAPL-Py allows the analysis of multiple treatment types at once (such as different drugs or time-points), provided they share the same controls. Next, the user selects the sgRNA library screened. PinAPL-Py provides support for the most common mouse and human genome-wide CRISPR knock-out library designs such as GeCKO (Sanjana *et al.*, 2014), Brie, and Brunello (Doench *et al.*, 2016) Use of custom libraries is possible after the user uploads a spreadsheet, specifying gene IDs and sgRNA sequences and provides a few additional parameters, such as 5'- and 3'- sgRNA adapter sequences. Finally, the user can adjust configuration settings, such as alternative methods for read count normalization, gene ranking, or various technical parameters for individual steps of the workflow.

Analysis workflow:

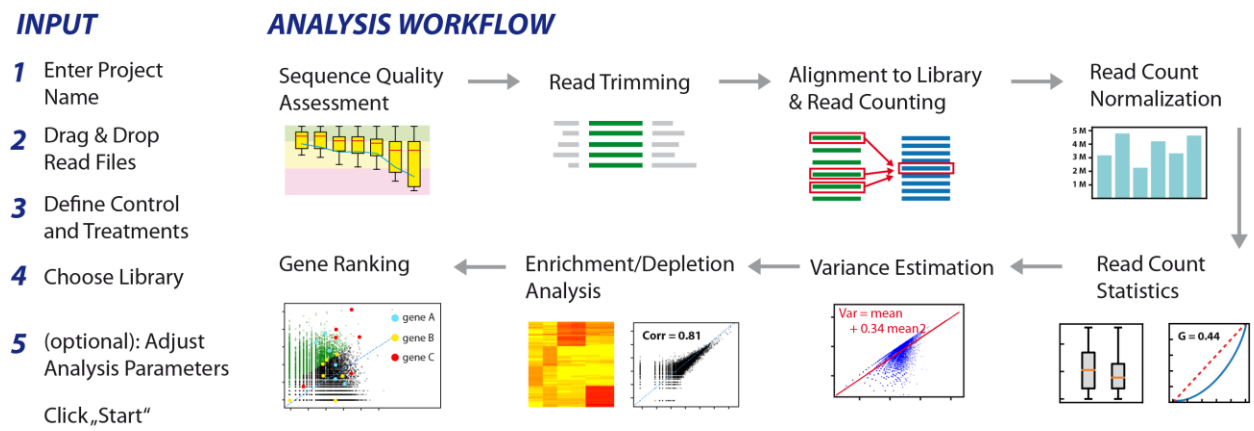

Fig. 1: The PinAPL-Py workflow. Sequence data input is done in 5 easy steps. Output from each analysis step is displayed on separate tabs and returned to the user as .xlsx, .txt, .png, and .svg files.

PinAPL-Py implements well-established methods to run its analysis workflow (Fig. 1). Sequence quality assessment is carried out using fastqc (http://www.bioinformatics.babraham.ac.uk/projects/fastqc/), providing plots of sequence content, sequencing quality and read depth. Reads are then processed with the cutadapt tool to clip sequence adapters (Martin, 2011) and aligned to the library using Bowtie2 (Langmead and Salzberg, 2012). Read counts per sgRNA and per gene are normalized using either counts per million, total, or size-factor normalization (Anders and Huber, 2010). Enrichment/depletion of single sgRNAs is determined by comparing the (normalized) read counts from a treatment sample to the estimated counts distribution from the control samples, based on a negative binomial model (Cameron and Trivedi, 2013). Analysis with only a single control sample is supported, but for statistical inference, at least two control samples need to be present. Based on the enrichment or depletion of individual sgRNAs, gene ranking using either the ES-, STARS, or αRRA metric (as introduced by MAGeCK) is carried out (Doench *et al.*, 2016; Li *et al.*, 2014; Subramanian *et al.*, 2005). The results are displayed through multiple tabs, allowing selection for different samples and steps of the workflow. A single archive file can be downloaded containing Excel tables, text files, high-resolution images to enable downstream analysis and processing. Running PinAPL-Py and MAGeCK on identical datasets proved to yield very similar results with respect to sgRNA enrichment, fold change and gene ranking (Suppl. Fig. 1). For method details we refer to the supplemental text.

Author contribution

Workflow development: PS. Experimental data: RW/PS. Webservice implementation: TB/JK. Supervision (Experimental data): JE. Supervision (Workflow development): NL/OH. Manuscript: PS/NL/OH. The authors declare no conflict of interest.

Funding

This work was supported by generous funding from the Novo Nordisk Foundation provided to the Center for Biosustainability (NNF16CC0021858), and grants from NIGMS (R35 GM119850 & P50 GM085764), NCI (R21 CA199292 & R21 CA177519), DOD (OC140179) and NHLBI (U54 HL108460 and U24 HL126127).

References

Anders,S. and Huber,W. (2010) Differential expression analysis for sequence count data. *Genome Biol.*, **11**, R106.

Cameron,A. and Trivedi,K. (2013) Regression Analysis of Count Data 2nd ed. Cambridge University Press.

Dai,Z. *et al.* (2014) shRNA-seq data analysis with edgeR. *F1000Research*, **3**, 95.

Doench,J.G. *et al.* (2016) Optimized sgRNA design to maximize activity and minimize off-target effects of CRISPR-Cas9. *Nat. Biotechnol.*, **34**, 184–191.

Hart,T. and Moffat,J. (2016) BAGEL: A computational framework for identifying essential genes from pooled library screens. *BMC Bioinformatics*, **17**, 33068.

Kim,J. *et al.* (2014) MAGI: A Node.js web service for fast microRNA-Seq analysis in a GPU infrastructure. *Bioinformatics*, **30**, 2826–2827.

Langmead,B. and Salzberg,S.L. (2012) Fast gapped-read alignment with Bowtie 2. *Nat. Methods*, **9**, 357–359.

Li,W. *et al.* (2014) MAGeCK enables robust identification of essential genes from genome-scale CRISPR/Cas9 knockout screens. *Genome Biol.*, 1–12.

Martin,M. (2011) Cutadapt Removes Adapter Sequences From High-Throughput Sequencing Reads. *EMBnet.journal*, **17**, 10–12.

Opdam,S. *et al.* (2017) A Systematic Evaluation of Methods for Tailoring Genome-Scale Metabolic Models. *Cell Syst.*, **In press**.

Sanjana,N.E. *et al.* (2014) Improved vectors and genome-wide libraries for CRISPR screening. *Nat. Methods*, **11**, 783–784.

Subramanian,A. *et al.* (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *PNAS*, **102**, 15545–50.

Winter,J. *et al.* (2015) CaRpools: An R package for exploratory data analysis and documentation of pooled CRISPR/Cas9 screens. *Bioinformatics*, **32**, 632–634.