The Journal of Open Source Software

Pycashier: cash in on DNA barcode tags

Daylin Morgan [1] **and Amy Brock** [1¶]

1 The University of Texas at Austin, United States of America ¶ Corresponding author

DOI: 10.21105/joss.06891

Software

- Review
- Repository
- Archive

Editor: Lorena Pantano
Reviewers:

- @adrientaudiere
- @LTLA

Submitted: 12 April 2024
Published: 31 October 2024

Summary

Pycashier is a tool designed to extract cellular DNA barcode tags from next generation sequencing data. These DNA barcode tags are heritable and stably integrated genetic markers useful for clonal tracking (Bhang et al., 2015) and lineage tracing (McKenna et al., 2016) within *in vitro* and *in vivo* cell-based disease models. These exogenous cell-based DNA barcodes, when amplified from genomic DNA and sequenced, can be used as a proxy for assessing changes in clonal abundance and better understanding population dynamics. Pycashier was originally developed for use with the ClonMapper Barcoding System (Al'Khafaji et al., 2018; Gardner et al., 2022, 2024)), which is comprised of random 20 nucleotide barcodes integrated as both a functional gRNA and expressed transcript. Pycashier has been previously utilized to interrogate tumor heterogeneity in barcoded cancer cell-line models (Gutierrez et al., 2021; Johnson et al., 2020), however, it is generalizable to similar DNA barcoding systems with known flanking regions and expected length.

Statement of need

DNA sequencing and cellular DNA barcoding specifically, have become more common as a modality for the characterization of clonal and lineage-specific subpopulations of cells. As researchers leverage these technologies, they'll require tools easy to setup and use to facilitate downstream biological analysis. DNA barcode sequencing suffers from several sources of noise that must be accounted for prior to statistical analysis. This noise can arise in typical Polymerase Chain Reaction (PCR) preparation (Kebschull & Zador, 2015; Potapov & Ong, 2017) or during read-out (Manley et al., 2016). Historically, the analysis of cellular DNA barcoding has relied on tailored computational workflows, such as TimeMachine (Emert et al., 2021), that are difficult to parameterize or extend to similarly designed cellular DNA barcoding systems. Recently, there has been the development of several NextFlow-based techniques, such as BARtab (Holze et al., 2024) and NextClone (Putri et al., 2023), In particular BARtab and it's associated post-processing library bartools, offer an end-to-end toolkit for barcoding analysis. As a more feature complete end-to-end toolkit BARtab differs from pycashier by including support for spatial transcriptomics data and reference-based processing of barcodes. Additionally, being based on NextFlow offers some advantages to these tools including sample-level parameterization and tighter control on system resources. However, experience using NextFlow may be uncommon for experimentalists. Pycashier aims to be simple to install and generalizable enough to be useful to the broader community while also providing a user friendly interface.

Implementation and Usage

Pycashier was intentionally designed to be simple-to-use for both computational and experimental biologists. It accomplishes this by leveraging purpose-built software for an opinionated DNA barcode processing pipeline. Pycashier has a command-line interface (CLI) implemented in python. Users have the option of installing pycashier with pip, conda/mamba/pixi (from conda-forge), or as a standalone Docker image which includes all necessary runtime dependencies for maximum reproducibility. Pycashier maintains outputs and logs of all steps for simple debugging and reuse across a project. The pycashier CLI has four subcommands to facilitate processing of DNA barcode sequencing data, extract, merge, scrna and receipt (Figure 1). Users can specify parameters either through command-line flags or through a configuration toml file.

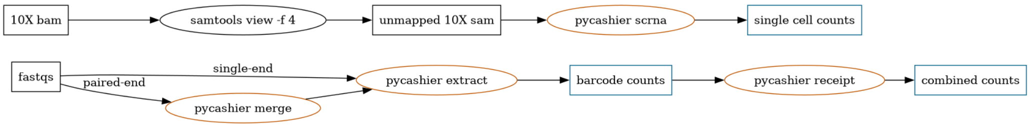

Figure 1: pycashier workflow

Pycashier is primarily used for generating counts of individual barcode sequences from targeted PCR amplifications of DNA-barcoded cells. Pycashier extracts these DNA barcode sequences without the use of any pre-defined list. This makes it amenable to systems in which sequences are not known ahead of time or randomly generated as in the case of ClonMapper. This is done with pycashier extract, which accepts a directory of fastq files directly from Illumina sequencing and generates a tsv of individual barcodes and counts for each input fastq. These sequences should be an expected length (specified with --length, by default 20), and flanked by *known regions* which are detectable in sequencing reads. These flanking regions can be specified either using CLI flags (as --upstream-adapter/--downstream-adapter) or within a user provided configuration file. First, filtering is performed with fastp (Chen et al., 2018) to remove low quality sequencing reads. Next, flanking sequences are used to extract a region of interest with cutadapt (Martin, 2011). Then, the list of identified sequences are corrected for errors introduced in either preparation or sequencing using a message passing clustering powered by starcode (Zorita et al., 2015). Finally, minimal count filtering is applied to remove any remaining noise from sequencing.

In addition to barcode extraction from targeted sequencing, pycashier facilitates barcode extraction from single-cell RNA-sequencing (scRNA-seq) in which cellular DNA barcodes are expressed as poly-adenylated transcripts. Specifically, it's compatible with data generated with the 10X Genomics 3' based single cell gene expression kit. In this case, the command pycashier scrna accepts sam files [1] derived from processed 10X data and generates a tsv with cell/UMI resolved barcode sequences, which can then be mapped directly back to the transcriptome of individual cells. To accomplish this, pycashier first extracts cell/UMI and read sequences from sam files using pysam (*Pysam-Developers/Pysam*, 2024). Next, sequences are individually extracted from fastq files again, using cutadapt with known flanking sequences (See Figure 1).

Pycashier provides two additional convenience commands: merge, to generate single read consensus sequences from paired-end sequencing, and reciept, to combine output tsv files from pycashier extract along with calculating some basic metrics across samples.

Documentation and further usage instructions for pycashier can be found at docs.brock-lab.com/pycashier.

Acknowledgments

We acknowledge Aziz Al'Khafaji for the development of the ClonMapper Barcoding System, which drove the development of this tool. The authors are grateful for support from NIH 1U01CA25354 (to A.B).

[1] CellRanger bam files can be converted to a sam with unmapped reads using samtools view -f 4.

References

Al'Khafaji, A. M., Deatherage, D., & Brock, A. (2018). Control of Lineage-Specific Gene Expression by Functionalized gRNA Barcodes. *ACS Synthetic Biology*, *7*(10), 2468–2474. https://doi.org/10.1021/acssynbio.8b00105

Bhang, H. C., Ruddy, D. A., Krishnamurthy Radhakrishna, V., Caushi, J. X., Zhao, R., Hims, M. M., Singh, A. P., Kao, I., Rakiec, D., Shaw, P., Balak, M., Raza, A., Ackley, E., Keen, N., Schlabach, M. R., Palmer, M., Leary, R. J., Chiang, D. Y., Sellers, W. R., … Stegmeier, F. (2015). Studying clonal dynamics in response to cancer therapy using high-complexity barcoding. *Nature Medicine*, *21*(5), 440–448. https://doi.org/10.1038/nm.3841

Chen, S., Zhou, Y., Chen, Y., & Gu, J. (2018). Fastp: An ultra-fast all-in-one FASTQ preprocessor. *Bioinformatics*, *34*(17), i884–i890. https://doi.org/10.1093/bioinformatics/bty560

Emert, B. L., Cote, C. J., Torre, E. A., Dardani, I. P., Jiang, C. L., Jain, N., Shaffer, S. M., & Raj, A. (2021). Variability within rare cell states enables multiple paths toward drug resistance. *Nature Biotechnology*, *39*(7), 865–876. https://doi.org/10.1038/s41587-021-00837-3

Gardner, A., Morgan, D., Al'Khafaji, A., & Brock, A. (2022). Functionalized Lineage Tracing for the Study and Manipulation of Heterogeneous Cell Populations. In A. Rasooly, H. Baker, & M. R. Ossandon (Eds.), *Biomedical Engineering Technologies: Volume 2* (pp. 109–131). Springer US. https://doi.org/10.1007/978-1-0716-1811-0_8

Gardner, A., Morgan, D., Al'Khafaji, A., & Brock, A. (2024). *ClonMapper Barcoding System*. https://docs.brocklab.com/clonmapper.

Gutierrez, C., Al'Khafaji, A. M., Brenner, E., Johnson, K. E., Gohil, S. H., Lin, Z., Knisbacher, B. A., Durrett, R. E., Li, S., Parvin, S., Biran, A., Zhang, W., Rassenti, L., Kipps, T. J., Livak, K. J., Neuberg, D., Letai, A., Getz, G., Wu, C. J., & Brock, A. (2021). Multifunctional barcoding with ClonMapper enables high-resolution study of clonal dynamics during tumor evolution and treatment. *Nature Cancer*, *2*(7), 758–772. https://doi.org/10.1038/s43018-021-00222-8

Holze, H., Talarmain, L., Fennell, K. A., Lam, E. Y., Dawson, M. A., & Vassiliadis, D. (2024). Analysis of synthetic cellular barcodes in the genome and transcriptome with BARtab and bartools. *Cell Reports Methods*, *4*(5). https://doi.org/10.1016/j.crmeth.2024.100763

Johnson, K. E., Howard, G. R., Morgan, D., Brenner, E. A., Gardner, A. L., Durrett, R. E., Mo, W., Al'Khafaji, A., Sontag, E. D., Jarrett, A. M., Yankeelov, T. E., & Brock, A. (2020). Integrating transcriptomics and bulk time course data into a mathematical framework to describe and predict therapeutic resistance in cancer. *Physical Biology*, *18*(1), 016001. https://doi.org/10.1088/1478-3975/abb09c

Kebschull, J. M., & Zador, A. M. (2015). Sources of PCR-induced distortions in high-throughput sequencing data sets. *Nucleic Acids Research*, *43*(21), e143. https://doi.org/10.1093/nar/gkv717

Manley, L. J., Ma, D., & Levine, S. S. (2016). Monitoring Error Rates In Illumina Sequencing. *Journal of Biomolecular Techniques: JBT*, *27*(4), 125–128. https://doi.org/10.7171/jbt.16-2704-002

Martin, M. (2011). Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet.journal*, *17*(1), 10–12. https://doi.org/10.14806/ej.17.1.200

McKenna, A., Findlay, G. M., Gagnon, J. A., Horwitz, M. S., Schier, A. F., & Shendure, J. (2016). Whole-organism lineage tracing by combinatorial and cumulative genome editing. *Science (New York, N.Y.)*, *353*(6298), aaf7907. https://doi.org/10.1126/science.aaf7907

Potapov, V., & Ong, J. L. (2017). Examining Sources of Error in PCR by Single-Molecule

Sequencing. *PLOS ONE*, *12*(1), e0169774. https://doi.org/10.1371/journal.pone.0169774

Putri, G. H., Pires, N., Davidson, N. M., Blyth, C., Al'Khafaji, A. M., Goel, S., & Phipson, B. (2023). *Extraction and quantification of lineage-tracing barcodes with NextClone and CloneDetective* (p. 2023.11.19.567755). bioRxiv. https://doi.org/10.1101/2023.11.19.567755

Pysam-developers/pysam. (2024). https://github.com/pysam-developers/pysam.

Zorita, E., Cuscó, P., & Filion, G. J. (2015). Starcode: Sequence clustering based on all-pairs search. *Bioinformatics*, *31*(12), 1913–1919. https://doi.org/10.1093/bioinformatics/btv053