The Journal of Open Source Software

FantasticLamp: a genome graph based pipeline for calculating the efficacy of genomic edits

Casper J. H. Schutte[1], **Ian T. Fiddes**[2], **and Erik Garrison**[3¶]

1 Stellenbosch University, South Africa **2** Inscripta Inc **3** The University of Tennessee Health Science Center, United States ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

Accurately calculating the efficacy of genomic edits is crucial to understanding the performance of the editing techniques, in order to optimize methods and improve the success rate of the edits. Additionally, by understanding the success rate of genomic edits, researchers can identify any potential problems or limitations of the techniques and work to overcome them. This can help to improve the accuracy and precision of the editing methods, which is essential for many applications, such as creating genetically engineered cells for therapeutic purposes (Doudna & Charpentier, 2014), understanding gene functions (Liu et al., 2019), and studying genetic diversity in a population of cells(Hsu et al., 2014). FantasticLamp is an open source pipeline for calculating the efficacy of genomic edits performed on multiple populations of cells. It constructs a variation graph from the reference genome and edit template sequences that includes both edited and unedited sequences as paths, and combinations of edits as potential walks through the graph. It then maps reads from the edited populations onto this graph in order to calculate the coverage of the edited sequences compared to the unedited sequences. This pipeline aims to provide a quantitative measure of the success of each genomic edit without bias towards the reference or particular single editing constructs.

Statement of need

In the field of genome engineering, researchers often perform genomic edits in cells, such as CRISPR/Cas9, TALEN, and ZNF-based systems, to study gene functions and to create new cell lines (Gaj et al., 2013). However, these edits may not always be successful, and it may be challenging to identify and quantify the success rate of these edits in large-scale data sets (Güell et al., 2014),(Haasteren et al., 2020). Compared to linear alignment methods, genome graphs can provide a more accurate and comprehensive view (Garrison et al., 2018) of the relationships between sequences (Paten et al., 2017), and by using this method, one can identify and quantify the success rate of genomic edits. When assessing the success rate of genomic edits, a linear alignment method may introduce reference bias and overlook the complexity of edits, particularly when multiple nearby edits and edit state mixtures are present. Linear approaches that attempt to consider all possible edit states as individual sequences in order to accurately represent complexity can become overly complex themselves and prone to errors (Huang et al., 2013), (Mun et al., 2021). Alternatively, a genome graph approach that attempts the same thing can provide a more complete and accurate perspective of sequence relationships (Eggertsson et al., 2017), while avoiding reference bias and capturing all allele state mixtures present within the targeted edits, even in the case of overlapped edits. The cost of this increased accuracy and the ability to represent complexity is that using genome graphs is more computationally intensive than linear alignments methods (Rakocevic et al., 2019).

FantasticLamp is a pipeline that consists of a bash script that can be initiated from the command line. The bash script calls a Python script written for this pipeline and the following bioinformatics tools: vg, which constructs genome graphs to represent genetic variation and facilitates efficient variant analysis (Garrison et al., 2018); odgi, which optimizes the representation of sequence graphs for scalable genome analysis and visualization (Guarracino et al., 2022); minimap2, which rapidly aligns long sequencing reads to a reference sequence, enabling efficient variant calling and structural variation analysis (Li, 2018); and seqwish, which converts sequences and independently generated alignments between them into a variation graph (Garrison & Guarracino, 2023) – here used to generate the reference graph target for alignment. FantasticLamp was designed to calculate the coverage of genomic edits in multiple populations of cells, simultaneously. The pipeline can handle both paired-end and single-end reads.

Given a reference genome and reads sequenced from multiple edited populations, the pipeline uses a design library CSV file, which contains the intended edits and reference sequences at the intended edit sites, to construct a genome graph. The genome graph is made up of the reference genome, the intended edit sequences ("homology arms", or "edit homology arms"), and the reference sequences at the edit sites ("reference homology arms"). The construction of the graph involves taking the homology arms and reference homology arms and mapping them to the reference genome using minimap2 (Li, 2018). A variation graph is then induced by seqwish (Garrison & Guarracino, 2023) that represents the relationships between the reference genome and both groups of homology arms (reference and edit). The inclusion of the reference homology arms is what allows the pipeline to calculate the relative coverage (and thus the edit efficiency) by comparing the edit homology arm coverage to the reference homology arm coverage. Next, the graph is constructed using odgi (Guarracino et al., 2022) by first "chopping" the nodes into segments smaller than 256 base pairs and then sorting the graph. These steps are necessary in order for the following steps to work. The graph is converted into a format that is more efficient and finally indexed, both steps using vg (Garrison et al., 2018). The pipeline then maps reads from the edited populations to this graph using vg, creating a GAF (Gene Annotation Format) file representing the alignment. A Python script is called to parse the alignment file and calculate the coverage of the homology arms compared to the reference homology arms. The output of this pipeline is a coverage table, which displays the name of each intended edit sequence along with the homology arm coverage and reference homology arm coverage. This allows FantasticLamp to simultaneously quantify the efficacy of edits in multiple edited populations, which can be used to test novel editing methods as well as to verify current methods in experiments that involve genomic edits. By using reads aligned to a graph instead of linear alignment, FantasticLamp can avoid reference bias, making it a useful tool for researchers in the field of genome engineering and other related fields who wish to precisely quantify genome edit states.

As novel genome editing processes are developed in the near future, we posit that the need for software tools that can capture the complexity of the edits and the relationship between them will increase. We show a basic approach for quantifying complex genome editing results, which are difficult to reliably and simply evaluate using a single reference genome that may lead to biased estimates of edit states. The design library CSV file used in the creation of this pipeline was unique to a specific set of editing experiments performed at Inscripta Inc. However, the basic format is generic to other editing experiments that users may carry out in the future, with only minimal changes to the script to adjust for difference in formatting of the design library file. Future users will have to edit the bash script "find_coverage.sh" such that the correct columns containing the edit- and reference homology arms are extracted. While the pipeline was tested only on singleplex sequencing data, it is possible to perform analysis using pooled sequencing data, but data availability has limited our ability to thoroughly test this. In summary, FantasticLamp provides a basic demonstration of the principle of using a variation graph as a reference system to avoid bias when quantifying genomic edits, and stands as a prototype for future work in this space.

Acknowledgements

We are grateful to Inscripta Inc for providing the data from editing experiments that was used to create and test this pipeline. We thank Deanna Church for supporting our collaboration.

References

Doudna, J. A., & Charpentier, E. (2014). The new frontier of genome engineering with CRISPR-Cas9. *Science*, *346*(6213), 1258096.

Eggertsson, H. P., Jonsson, H., Kristmundsdottir, S., Hjartarson, E., Kehr, B., Masson, G., Zink, F., Hjorleifsson, K. E., Jonasdottir, A., Jonasdottir, A., & others. (2017). Graphtyper enables population-scale genotyping using pangenome graphs. *Nature Genetics*, *49*(11), 1654–1660.

Gaj, T., Gersbach, C. A., & Barbas III, C. F. (2013). ZFN, TALEN, and CRISPR/cas-based methods for genome engineering. *Trends in Biotechnology*, *31*(7), 397–405. https://doi.org/10.1016/j.tibtech.2013.04.004

Garrison, E., & Guarracino, A. (2023). Unbiased pangenome graphs. *Bioinformatics*, *39*(1), btac743.

Garrison, E., Sirén, J., Novak, A. M., Hickey, G., Eizenga, J. M., Dawson, E. T., Jones, W., Garg, S., Markello, C., Lin, M. F., & others. (2018). Variation graph toolkit improves read mapping by representing genetic variation in the reference. *Nature Biotechnology*, *36*(9), 875–879.

Guarracino, A., Heumos, S., Nahnsen, S., Prins, P., & Garrison, E. (2022). ODGI: Understanding pangenome graphs. *Bioinformatics*, *38*(13), 3319–3326.

Güell, M., Yang, L., & Church, G. M. (2014). Genome editing assessment using CRISPR genome analyzer (CRISPR-GA). *Bioinformatics*, *30*(20), 2968–2970.

Haasteren, J. van, Li, J., Scheideler, O. J., Murthy, N., & Schaffer, D. V. (2020). The delivery challenge: Fulfilling the promise of therapeutic genome editing. *Nature Biotechnology*, *38*(7), 845–855.

Hsu, P. D., Lander, E. S., & Zhang, F. (2014). Development and applications of CRISPR-Cas9 for genome engineering. *Cell*, *157*(6), 1262–1278.

Huang, L., Popic, V., & Batzoglou, S. (2013). Short read alignment with populations of genomes. *Bioinformatics*, *29*(13), i361–i370.

Li, H. (2018). Minimap2: Pairwise alignment for nucleotide sequences. *Bioinformatics*, *34*(18), 3094–3100.

Liu, J.-J., Orlova, N., Oakes, B. L., Ma, E., Spinner, H. B., Baney, K. L., Chuck, J., Tan, D., Knott, G. J., Harrington, L. B., & others. (2019). CasX enzymes comprise a distinct family of RNA-guided genome editors. *Nature*, *566*(7743), 218–223.

Mun, T., Chen, N.-C., & Langmead, B. (2021). LevioSAM: Fast lift-over of variant-aware reference alignments. *Bioinformatics*, *37*(22), 4243–4245.

Paten, B., Novak, A. M., Eizenga, J. M., & Garrison, E. (2017). Genome graphs and the evolution of genome inference. *Genome Research*, *27*(5), 665–676.

Rakocevic, G., Semenyuk, V., Lee, W.-P., Spencer, J., Browning, J., Johnson, I. J., Arsenijevic, V., Nadj, J., Ghose, K., Suciu, M. C., & others. (2019). Fast and accurate genomic analyses using genome graphs. *Nature Genetics*, *51*(2), 354–362.