The Journal of Open Source Software

Enhancing Next-Generation Sequencing Simulation: Updates to NEAT

Joshua M. Allen[1*], **Keshav R. Gandhi**[1,2*¶], **and Christina E. Fliege**[1]

1 National Center for Supercomputing Applications, Genomics Group, Urbana, IL, USA, 61801, **2** University of Illinois at Chicago, Chicago, IL, USA, 60607 ¶ Corresponding author * These authors contributed equally.

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Charlotte Soneson

Reviewers:

- @erik-whiting
- @bricoletc

Submitted: 08 June 2025
Published: unpublished

Summary

While the field of genomics has advanced significantly with the advent of high-throughput sequencing technologies, challenges related to the availability, complexity, and variability of this data can introduce difficulty to the development and validation of computational tools. Simulated short-read sequencing datasets provide researchers a way to get reproducible, verified data to test algorithms and benchmark software. Simulations also avoid the limitations of working with real data, including the cost of genomic sequencing, time to process sequencing data, and protection of privacy. Ideally, these datasets mimic the properties of real sequencing datasets—from introducing specific patterns of sequencing errors to modeling localized regions of mutations.

Statement of Need

The NExt-generation sequencing Analysis Toolkit (NEAT) is an open-source Python package that creates simulated next-generation sequencing datasets. NEAT's simulations account for a wide range of sequencing parameters (e.g., DNA read fragment length, sequencing error rates, mutation frequencies, etc.) and allow users to customize their sequencing data (Stephens et al., 2016). Since the original release of NEAT in 2016, most scripts have been greatly modified, and NEAT is currently on version 4.2. The code has undergone significant ongoing changes since 2020. Upgrading to Python 3 brought NEAT up to modern coding standards and allowed it to use standard Python libraries in order to streamline the code and improve its maintainability. The toolkit is optimized for both speed and accuracy, and new features have been implemented. A summary of algorithmic changes is provided in **Table 1**.

NEAT can integrate seamlessly with existing bioinformatics workflows, providing outputs in several common file formats. The toolkit's ability to simulate gold-standard synthetic datasets with ground truth annotations is useful for testing bioinformatics pipelines. Uses of NEAT continue to be prominently featured—from scientists who have comprehensively sequenced the human Y chromosome (Rhie et al., 2023) to researchers who use NEAT to evaluate and validate the performance of other high-profile bioinformatics tools (Lefouili & Nam, 2022; Zhao et al., 2020). Earlier versions of NEAT have also demonstrated utility when benchmarked in comparison to similar tools (Alosaimi et al., 2020).

Table 2 describes recent changes to NEAT's user experience. The source code for both original and updated versions of NEAT is freely available on GitHub (Stephens et al., 2016).

Tables

Table 1. Algorithmic Improvements and Methodological Changes

#	Feature Name	Prior Implementation (2.0)	Updated Implementation (4.2)
1	**BAM File Generation**	File generation was tightly integrated with all NEAT processes	BAM creation was isolated from core functions
2	**GC Bias Computation**	Used a custom script for GC bias calculation	Feature deprecated
3	**Ploidy Simulation**	Limited to diploid organisms in practice	Supports all ploidy levels
4	**Read Generation**	Sliding-window approach to generate reads	A new form of coordinate-based read selection
5	**Read Quality Modeling**	Markov-based model	Binning method with an option to also implement a revised Markov-based model
6	**Variant Insertion**	Issues with inserted variants (loss of genotype data)	Preserves genotype data in the final VCF file
7	**Variant Handling**	Limited introduction of new variant types	A modular design with generic variant handling and the separation of insertions and deletions

The creation of simulated Binary Alignment Map (BAM) files (**1**) in NEAT 2.0 was tightly integrated with all NEAT functions. The new update isolates BAM creation, improving runtime and modularity. Guanine-cytosine (GC) bias computation (**2**) was removed due to redundancy, and its removal reduced runtime. Ploidy simulation (**3**) has been extended to improve accurate simulation of tumor genomes and polyploid organisms (e.g., plants), and ploidy inputs greater than two and fractional ploidies are now handled. Previously, NEAT 2.0's read generation (**4**) algorithm introduced read gaps (~50 base pairs) due to its sliding-window approach. The updated coordinate-based selection eliminates these gaps. Modeling of sequencing quality scores for each nucleotide base (**5**) was updated by incorporating a revised Markov model alongside a binning method. We accurately account for a tapering effect that reduces sequencing quality scores along a simulated sequence's edges. Variant insertion (**6**) was updated to preserve genotype data in the final simulated Variant Call Format (VCF) file, improving accuracy and giving users greater control over the insertion of variants. Finally, variant handling (**7**) has been modularized to support structural and copy number variants, increasing flexibility and ensuring future extensibility for handling more complex variants.

Table 2. Improvements in User Experience

#	Feature Name	Prior Implementation (2.0)	Updated Implementation (4.2)
1	**Automated Testing**	No formal testing framework	Implemented continuous integration with GitHub-based automated tests
2	**Configuration Files**	Required explicit command-line flags	Introduced structured configuration files
3	**Friendly Installation**	Not installable as a package	Fully modular and pip-installable via Poetry
4	**Refactored Unit Testing**	Not originally present	Rewritten with testable, discrete functions

Our new continuous integration pipeline (**1**) detects bugs early, streamlining development and enhancing error detection (e.g., handling of multiple genomic file formats as inputs and outputs). Configuration files in NEAT 4.2 (**2**) and package installation (**3**) facilitate user friendliness and portability. NEAT 4.2 features testable, discrete functions (**4**) that allows users to debug more easily. NEAT 4.2's read simulator is also parallelized, facilitating faster runtimes and ease of use. Quality-of-life development continues into the present.

Acknowledgements

We thank the original creators of NEAT: Zachary D. Stephens, Matthew E. Hudson, Liudmila S. Mainzer, Morgan Taschuk, Matthew R. Weber, and Ravishankar K. Iyer.

We also thank Raghid Alhamzy, Yash Wasnik, Varenya Jain, and Karen H. Xiong.

References

Alosaimi, S., Bandiang, A., Biljon, N. van, & others. (2020). A broad survey of DNA sequence data simulation tools. *Briefings in Functional Genomics*, *19*(1), 49–59. https://doi.org/10.1093/bfgp/elz033

Lefouili, M., & Nam, K. (2022). The evaluation of bcftools mpileup and GATK HaplotypeCaller for variant calling in non-human species. *Scientific Reports*, *12*(1). https://doi.org/10.1038/s41598-022-15563-2

Rhie, A., Nurk, S., Cechova, M., & others. (2023). The complete sequence of a human y chromosome. *Nature*, *621*(7978), 344–354. https://doi.org/10.1038/s41586-023-06457-y

Stephens, Z. D., Hudson, M. E., Mainzer, L. S., Taschuk, M., Weber, M. R., & Iyer, R. K. (2016). Simulating next-generation sequencing datasets from empirical mutation and sequencing models. *PLOS ONE*, *11*(11). https://doi.org/10.1371/journal.pone.0167047

Zhao, S., Agafonov, O., Azab, A., Stokowy, T., & Hovig, E. (2020). Accuracy and efficiency of germline variant calling pipelines for human genome data. *Scientific Reports*, *10*(1). https://doi.org/10.1038/s41598-020-77218-4