The Journal of Open Source Software

# Enhancing short-read sequencing simulation: Updates to NEAT

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

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

**DOI:** 10.21105/joss.09056

**Software**

- Review
- Repository
- Archive

**Editor:** Charlotte Soneson
**Reviewers:**

- @erik-whiting
- @bricoletc
- @chenyenchung

**Submitted:** 08 June 2025
**Published:** 04 May 2026



## Summary

While the field of genomics has advanced significantly with high-throughput sequencing technologies, challenges related to the availability, complexity, and variability of data introduce difficulty when developing and validating computational tools. Simulated short-read sequencing datasets provide researchers with reproducible, verified data to test algorithms and benchmark software. Simulations also avoid the limitations of working with real data, such as the cost of genomic sequencing, processing time, accessibility, and protection of privacy for human datasets. Ideally, simulated datasets mimic the properties of real sequencing data, which is necessary to evaluate the accuracy and robustness of downstream alignment, variant-calling, and analysis pipelines. Sequencing parameters such as ploidy, repeat structure, and genome complexity vary across species, and it is important for simulated reads to reflect these species-specific characteristics. The NExt-generation sequencing Analysis Toolkit (NEAT) is an open-source Python package that creates simulated sequencing datasets, originally released in 2016. The current release, v4.4, introduces increased speed, accuracy, and usability.

## Statement of need

Developing and validating methods for read alignment, variant calling, and other analyses requires genomic data with known ground truth. Varying the parameters of the sequencing process allows us to test analysis pipelines across different coverages, read lengths, ploidies, and more. NEAT addresses this need as an open-source Python package that allows users to customize their sequencing data (Stephens et al., 2016).

NEAT is designed to simulate short reads from sequencing platforms with machine-specific custom error models and can account for single-base substitutions, insertions, deletions, and larger structural variants. Unlike simulators that rely on fixed statistical profiles, NEAT learns empirical mutation and sequencing statistics from real data. NEAT models realistic, benchmarking-ready sequencing data, providing outputs in common bioinformatics file formats, including FASTQ, binary alignment map (BAM), and variant call format (VCF) files.

## Software design

NEAT has changed significantly since the release of version 2.0 in 2016. The codebase was updated to Python 3 to take advantage of the latest libraries and then was optimized for speed and accuracy alongside several new features. A summary of changes to NEAT is provided in **Table 1**.

**Table 1. Methodological, software robustness, and user experience updates in NEAT 4.4**

| No. | Feature Name | Prior Implementation (2.0) | Updated Implementation (4.4) |
|---|---|---|---|
| **1** | GC Bias Computation | Used custom model of GC bias | Removed, pending further investigation |
| **2** | Read Generation | Sliding-window approach | Coordinate-based read selection |
| **3** | Variant Input | Partial data loss from input variants | Preserves all input data |
| **4** | Variant Modeling | Two variant types supported | Framework to expand variant types |
| **5** | Automated Testing | No formal testing framework | Unit tests and automated continuous integration |
| **6** | Configuration Files | Command line interface only | Structured configuration files for reproducibility |
| **7** | Detailed Logging | Minimal error logging | Extensive logs to recreate runs and describe errors |
| **8** | Friendly Installation | Clone repository and install | pip installable |
| **9** | Parallelization | Single-threaded | Multi-threaded |
| **10** | Refactored Unit Testing | None | Added for all major functions |

Below, we summarize methodological changes (**Table 1**) present in NEAT 4.4:

- [**1**] Guanine-cytosine (GC) bias refers to sequencing machine bias in GC-rich or GC-poor regions, causing uneven coverage results in real data (Benjamini & Speed, 2012; Ross et al., 2013). Evidence suggests GC bias may arise from library preparation and amplification rather than from sequencing (Benjamini & Speed, 2012). This is an area of ongoing research and development for NEAT.
- [**2**] NEAT 4.4's read generation algorithm eliminates artificial gaps in the output and now facilitates parallelization.
- [**3**] Variants can be incorporated into NEAT-generated reads via user-inputted VCF files. NEAT 2.0 reads only minimal data of the variant, but this has been expanded in NEAT 4.4.
- [**4**] NEAT 4.4 uses the same variants as NEAT 2.0, but the code has been updated to allow for additional variant types in future releases.

Documentation of software robustness, user experience, parallelization performance benchmarks, and more future improvements [**5–10**] is available on the project's online repository.

## State of the field

Even as long-read sequencing advances, short-read, bulk sequencing remains prominent due to its comparatively low cost and high throughput. Simulating short-read datasets can replicate sequencing pipelines used in a wide variety of research settings. Investigations of read simulators have analyzed use cases across whole genomes, exomes, and metagenomes (Escalona et al., 2016; M. Zhao et al., 2017) and whether empirical error-profile learning improves realism (Alosaimi et al., 2020; Milhaven & Pfeifer, 2023; Schmeing & Robinson, 2021). While many short-read simulators have appeared in these studies (ART, CuReSim, DWGSIM, GemSIM, InSilicoSeq, Mason, NEAT, pIRS, ReSeq, SInC, and wgsim (Caboche et al., 2014; Gourlé et al., 2019; Holtgrewe, 2010; Homer, 2010; Hu et al., 2012; Huang et al., 2012; Li, 2011; McElroy et al., 2012; Pattnaik et al., 2014; Schmeing & Robinson, 2021)), only a subset were found to produce sequencing data with explicit ground truth suitable for benchmarking (Alosaimi et al., 2020), including NEAT (Milhaven & Pfeifer, 2023).

Additionally, in their benchmark of twenty DNA read simulators that use reference genomes and produce FASTQ files, Alosaimi et al. (2020) found that NEAT's suite of features compares favorably to other tools. Although NEAT achieved the second-highest mapping sensitivity and precision on a human chromosome 22 test set, NEAT's runtimes were the second-longest (Alosaimi et al., 2020). Milhaven and Pfeifer also noted NEAT's realism in sequencing—but also its slow simulation runtimes (Milhaven & Pfeifer, 2023). NEAT is noteworthy for its ability to combine mutation and sequencing models in a single framework and accept user-specified mutation models (Stephens et al., 2016). The latest version of NEAT maintains these strengths and addresses some of these weaknesses with multi-threading and algorithmic updates, as outlined above.

## Research impact statement

NEAT is notable for producing ground-truth BAM and VCF outputs suitable for systematic benchmarking pipelines using custom mutation and sequencing models (Alosaimi et al., 2020; Milhaven & Pfeifer, 2023; Schmeing & Robinson, 2021; Stephens et al., 2016). Researchers have been using NEAT since its initial release—from assisting benchmarks in the sequencing of the human Y chromosome (Rhie et al., 2023) to evaluating other bioinformatics tools (Lefouili & Nam, 2022; S. Zhao et al., 2020). NEAT supports method development across a range of bioinformatics tasks, including optimization of high-throughput variant-calling workflows (Ahmed et al., 2019; Kendig et al., 2019), feasibility studies of exome sequencing (Ruiz-Schultz et al., 2021), pan-genome mapping (Jandrasits et al., 2019), Bayesian approaches for resolving ambiguously mapped reads (Shah & Ruthenburg, 2021), and ultra-sensitive multi-sample variant calling (Delhomme et al., 2020).

The updates described here (NEAT 4.4) highlight NEAT's practicality for future work in the field. The source code for NEAT is freely available on GitHub (Stephens et al., 2016).

## 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 Varenya Jain, Meredith Pudlewski, Karen H. Xiong, and other contributors for their work on updating NEAT.

Portions of this project were funded by the National Center for Supercomputing Applications' Students Pushing Innovation (SPIN) program and the Illinois Computes project through the University of Illinois Urbana-Champaign.

## References


Ahmed, A. E., Heldenbrand, J., Asmann, Y., Fadlelmola, F. M., Katz, D. S., Kendig, K., Kendzior, M. C., Li, T., Ren, Y., Rodriguez, E., Weber, M. R., Wozniak, J. M., Zermeno, J., & Mainzer, L. S. (2019). Managing genomic variant calling workflows with Swift/T. *PLOS ONE*, *14*(7), e0211608. https://doi.org/10.1371/journal.pone.0211608

Alosaimi, S., Bandiang, A., van Biljon, N., Awany, D., Thami, P. K., Tchamga, M. S. S., Kiran, A., Messaoud, O., Hassan, R. I. M., Mugo, J., Ahmed, A., Bope, C. D., Allali, I., Mazandu, G. K., Mulder, N. J., & Chimusa, E. R. (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

Benjamini, Y., & Speed, T. P. (2012). Summarizing and correcting the GC content bias in high-throughput sequencing. *Nucleic Acids Research*, *40*(10), e72. https://doi.org/10.1093/nar/gks001

Caboche, S., Audebert, C., Lemoine, Y., & Hot, D. (2014). Comparison of mapping algorithms used in high-throughput sequencing: Application to Ion Torrent data. *BMC Genomics*, *15*, 264. https://doi.org/10.1186/1471-2164-15-264

Delhomme, T. M., Avogbe, P. H., Gabriel, A. A. G., Alcala, N., Leblay, N., Voegele, C., Vallée, M., Chopard, P., Chabrier, A., Abedi-Ardekani, B., Gaborieau, V., Holcatova, I., Janout, V., Foretová, L., Milosavljevic, S., Zaridze, D., Mukeriya, A., Brambilla, E., Brennan, P., … Foll, M. (2020). Needlestack: An ultra-sensitive variant caller for multi-sample next generation sequencing data. *NAR Genomics and Bioinformatics*, *2*(2), lqaa021. https://doi.org/10.1093/nargab/lqaa021

Escalona, M., Rocha, S., & Posada, D. (2016). A comparison of tools for the simulation of genomic next-generation sequencing data. *Nature Reviews Genetics*, *17*(8), 459–469. https://doi.org/10.1038/nrg.2016.57

Gourlé, H., Karlsson-Lindsjö, O., Hayer, J., & Bongcam-Rudloff, E. (2019). Simulating Illumina metagenomic data with InSilicoSeq. *Bioinformatics*, *35*(3), 521–522. https://doi.org/10.1093/bioinformatics/bty630

Holtgrewe, M. (2010). *Mason – A read simulator for second generation sequencing data* (B-10-06). Freie Universität Berlin, Fachbereich Mathematik und Informatik; Technical Report TR-B-10-06. https://publications.imp.fu-berlin.de/962/2/mason201009.pdf

Homer, N. (2010). *DWGSIM: Whole genome simulator for next-generation sequencing*. GitHub repository. https://github.com/nh13/DWGSIM

Hu, X., Yuan, J., Shi, Y., Lu, J., Liu, B., Li, Z., Chen, Y., Mu, D., Zhang, H., Li, N., Yue, Z., Bai, F., Li, H., & Fan, W. (2012). pIRS: Profile-based Illumina paired-end reads simulator. *Bioinformatics*, *28*(11), 1533–1535. https://doi.org/10.1093/bioinformatics/bts187

Huang, W., Li, L., Myers, J. R., & Marth, G. T. (2012). ART: A next-generation sequencing read simulator. *Bioinformatics*, *28*(4), 593–594. https://doi.org/10.1093/bioinformatics/btr708

Jandrasits, C., Kröger, S., Haas, W., & Renard, B. Y. (2019). Computational pan-genome mapping and pairwise SNP-distance improve detection of *Mycobacterium tuberculosis* transmission clusters. *PLOS Computational Biology*, *15*(12), e1007527. https://doi.org/10.1371/journal.pcbi.1007527

Kendig, K. I., Baheti, S., Bockol, M. A., Drucker, T. M., Hart, S. N., Heldenbrand, J. R., Hernaez, M., Hudson, M. E., Kalmbach, M. T., Klee, E. W., Mattson, N. R., Ross, C. A., Taschuk, M., Wieben, E. D., Wiepert, M., Wildman, D. E., & Mainzer, L. S. (2019). Sentieon DNASeq variant calling workflow demonstrates strong computational performance and accuracy. *Frontiers in Genetics*, *10*, 736. https://doi.org/10.3389/fgene.2019.00736

Lefouili, M., & Nam, K. (2022). The evaluation of Bcftools mpileup and GATK HaplotypeCaller for variant calling in non-human species. *Scientific Reports*, *12*, 11331. https://doi.org/10.1038/s41598-022-15563-2

Li, H. (2011). *wgsim-Read simulator for next generation sequencing*. GitHub repository. https://github.com/lh3/wgsim

McElroy, K. E., Luciani, F., & Thomas, T. (2012). GemSIM: General, error-model based simulator of next-generation sequencing data. *BMC Genomics*, *13*, 74. https://doi.org/10.1186/1471-2164-13-74

Milhaven, M., & Pfeifer, S. P. (2023). Performance evaluation of six popular short-read simulators. *Heredity*, *130*, 55–63. https://doi.org/10.1038/s41437-022-00577-3

Pattnaik, S., Gupta, S., Rao, A. A., & Panda, B. (2014). SInC: An accurate and fast error-model based simulator for SNPs, Indels and CNVs coupled with a read generator for short-read sequence data. *BMC Bioinformatics*, *15*, 40. https://doi.org/10.1186/1471-2105-15-40

Rhie, A., Nurk, S., Cechova, M., Hoyt, S. J., Taylor, D. J., Altemose, N., Hook, P. W., Koren, S., Rautiainen, M., Alexandrov, I. A., Allen, J., Asri, M., Bzikadze, A. V., Chen, N.-C., Chin, C.-S., Diekhans, M., Flicek, P., Formenti, G., Fungtammasan, A., … Phillippy, A. M. (2023). The complete sequence of a human Y chromosome. *Nature*, *621*(7978), 344–354. https://doi.org/10.1038/s41586-023-06457-y

Ross, M. G., Russ, C., Costello, M., Hollinger, A., Lennon, N. J., Hegarty, R., Nusbaum, C., & Jaffe, D. B. (2013). Characterizing and measuring bias in sequence data. *Genome Biology*, *14*(5), R51. https://doi.org/10.1186/gb-2013-14-5-r51

Ruiz-Schultz, N., Sant, D., Norcross, S., Dansithong, W., Hart, K., Asay, B., Little, J., Chung, K., Oakeson, K. F., Young, E. L., Eilbeck, K., & Rohrwasser, A. (2021). Methods and feasibility study for exome sequencing as a universal second-tier test in newborn screening. *Genetics in Medicine*, *23*(4), 767–776. https://doi.org/10.1038/s41436-020-01058-w

Schmeing, S., & Robinson, M. D. (2021). ReSeq simulates realistic Illumina high-throughput sequencing data. *Genome Biology*, *22*, 67. https://doi.org/10.1186/s13059-021-02265-7

Shah, R. N., & Ruthenburg, A. J. (2021). Sequence deeper without sequencing more: Bayesian resolution of ambiguously mapped reads. *PLOS Computational Biology*, *17*(4), e1008926. https://doi.org/10.1371/journal.pcbi.1008926

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), e0167047. https://doi.org/10.1371/journal.pone.0167047

Zhao, M., Liu, D., & Qu, H. (2017). Systematic review of next-generation sequencing simulators: Computational tools, features and perspectives. *Briefings in Functional Genomics*, *16*(3), 121–128. https://doi.org/10.1093/bfgp/elw012

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*, 20222. https://doi.org/10.1038/s41598-020-77218-4