Differential analysis of genomics count data with edge*

Lior Pachter*[1]

[1]Division of Biology and Biological Engineering & Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, USA

Abstract

The edgeR Bioconductor package is one of the most widely used tools for differential expression analysis of count-based genomics data. Despite its popularity, the R-only implementation limits its integration with the Python-centric ecosystem that has become dominant in single-cell genomics. We present edgePython, a Python port of edgeR 4.8.2 that extends the framework with a negative binomial–gamma mixed model for multi-subject single-cell analysis and empirical Bayes shrinkage of cell-level dispersion.

Introduction

The edgeR Bioconductor (Huber et al., 2015) package (Robinson et al., 2010) has been one of the most influential tools in genomics. Together with DESeq2 (Love et al., 2014), it established the negative binomial framework as the standard for count-based differential expression analysis. Since its introduction, edgeR and its underlying statistical ideas have shaped how the field thinks about count data: the negative binomial model for biological variability (Robinson and Smyth, 2007a), TMM normalization (Robinson and Oshlack, 2010), empirical Bayes shrinkage of dispersion estimates (Robinson and Smyth, 2007b), generalized linear models for multifactor experiments (McCarthy et al., 2012), quasi-likelihood F-tests (Lun et al., 2016), and gene set testing methods including camera (Wu and Smyth, 2012), roast and fry (Chen et al., 2016). A central theme running through this body of work is the idea, originating in the limma package (Smyth, 2004; Ritchie et al., 2015), that borrowing information across genes via empirical Bayes moderation yields more stable and powerful inference, especially in small-sample settings.

Despite the development of rigorous modeling of technical noise in edgeR, the method was "gene-based", meaning that the counts analyzed were aggregated over exons or gene bodies. In (Trapnell et al., 2010), the importance of isoform-level analysis was emphasized and a count-based method for isoform differential analysis was introduced in sleuth (Pimentel et al., 2017). The sleuth method relied on estimates of inferential uncertainty, an additional source of noise, to account for the ambiguity of read mapping to isoforms of genes. An adaptation of this approach was incorporated into edgeR in (Baldoni et al., 2024), which led to a method comparable to sleuth and addressed a major flaw of edgeR.

While edgeR now addresses quantification uncertainty, there is another important missing piece that is essential for current genomics applications, namely an approach to single-cell differential analysis. Specifically, single-cell genomics data require assessment not only of technical noise between samples, but also modeling of variability within cell-types. A number of methods have been developed that address this challenge (Finak et al., 2015; Crowell et al., 2020), however such approaches have not been incorporated in edgeR.

To address this shortcoming, we extend the empirical Bayes framework to single-cell data. Multi-subject single-cell differential expression requires a mixed model that accounts for both cell-level overdispersion and subject-level variation. We implement a negative binomial–gamma mixed model following the NEBULA-LN approach (He et al., 2021), and then apply edgeR's empirical Bayes machinery to shrink the per-gene cell-level dispersion estimates toward an abundance-dependent prior. We implement the extension via the same `squeezeVar` function that underlies edgeR's quasi-likelihood workflow, but our method is new: neither edgeR nor NEBULA currently perform empirical Bayes moderation of the within-cell dispersion in the mixed model setting.

Rather than working in R, we implement our method in Python, thus providing bidirectional conversion with the popular Python AnnData, which has become a standard for single-cell analysis. AnnData is part of the Scanpy framework (Wolf et al., 2018) and the broader scverse ecosystem (Virshup et al., 2023), and these tools have established AnnData as the standard data structure for single-cell analysis. Most new methods for dimensionality reduction, trajectory inference, and cell type annotation are implemented in Python (Zappia et al., 2018). This creates a practical barrier for researchers who wish to use edgeR's statistical methods within Python-based workflows: they must either export data to R and import results back, or rely on fragile inter-language bridges that are difficult to maintain and debug. We therefore performed a broad port of edgeR to Python,

*Address correspondence to lpachter@caltech.edu

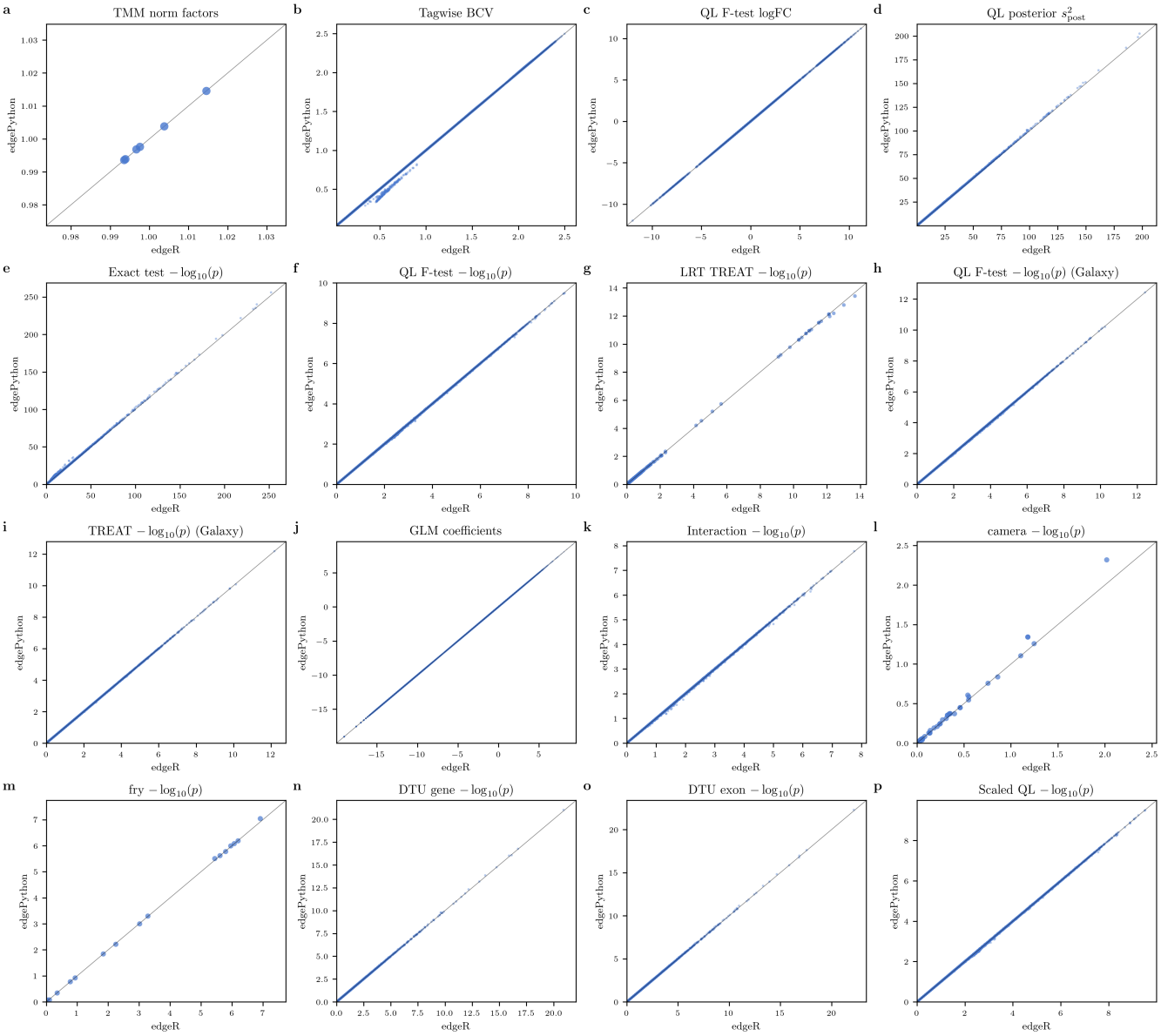

Figure 1: **Validation of edgePython.** Each panel shows a scatter plot comparing outputs from identical analyses run in R (edgeR, x-axis) and Python (edgePython, y-axis). Row 1 (a–d): core pipeline components (normalization, dispersion, effect sizes, empirical Bayes shrinkage). Row 2 (e–h): hypothesis testing frameworks (exact test, QL F-test, LRT TREAT, and the GSE60450 multi-factor QL F-test). Row 3 (i–l): multi-factor analyses and gene set testing (TREAT, GLM coefficients, interaction F-test, camera). Row 4 (m–p): additional gene set and specialized analyses (fry, DTU gene-level, DTU exon-level, scaled analysis). **(a)** TMM normalization factors for the HOXA1 dataset (6 samples). **(b)** Tagwise biological coefficient of variation (BCV $= \sqrt{\phi}$) for 44,514 transcripts. **(c)** QL F-test log-fold-changes, validating concordance of effect sizes. **(d)** QL posterior variances (s^2_{post}) after empirical Bayes shrinkage. **(e)** Exact test p-values ($-\log_{10}$ scale) for HOXA1 knockdown vs. scrambled. **(f)** QL F-test p-values on the same dataset. **(g)** LRT TREAT p-values pooled across four log-fold-change thresholds. **(h)** QL F-test p-values for the basal pregnant-vs-lactating contrast in the GSE60450 multi-factor design (15,804 genes). **(i)** TREAT p-values with log-fold-change threshold 0.58 on the same contrast. **(j)** GLM coefficient estimates (all six coefficients pooled) from the GSE60450 factorial design. **(k)** Multi-degree-of-freedom interaction F-test p-values from the same design. **(l)** camera competitive gene set test p-values pooled across six configurations. **(m)** fry gene set test p-values. **(n)** Differential transcript usage (DTU) gene-level Simes p-values for 9,697 genes. **(o)** DTU exon-level p-values for 40,958 individual exon tests. **(p)** Scaled analysis QL p-values for 44,811 transcripts, incorporating quantification uncertainty via overdispersion scaling. In all panels the diagonal line indicates perfect agreement.

which we name edgePython, including numerous routines that edgeR relies on written in C[1].

Our port covers all major edgeR functions, including normalization, dispersion estimation, GLM fitting, all four hypothesis testing frameworks (exact test, likelihood ratio test, quasi-likelihood F-test, and treat), as well as gene set testing (camera, fry, roast, mroast, romer). The implementation uses Python dictionaries to mirror edgeR's S3 list-based objects, and operates on NumPy arrays as well as SciPy sparse matrices. We validate the port by running identical analyses in R and Python and comparing outputs across the full workflow. This should facilitate the incorporation of the edge (empirical analysis of digital expression data) techniques in the computational single-cell genomics ecosystem.

Results

Python port

The edgePython program exports 86 public functions and classes organized into 24 modules covering the major workflows of edgeR 4.8.2: data structures (`DGEList`, `DGEGLM`, `DGELRT`), normalization (TMM, TMMwsp, RLE, upper quartile), dispersion estimation (common, trended, tagwise, and GLM-based), generalized linear model fitting (negative binomial and quasi-likelihood), all four hypothesis testing frameworks (exact test, likelihood ratio test, quasi-likelihood F-test, and TREAT), gene set testing (camera, fry, roast, mroast, romer, goana, kegga), as well as differential splicing and bisulfite sequencing analysis (Chen et al., 2014). The implementation uses Python dictionaries to mirror edgeR's S3 list-based objects and operates on NumPy (Harris et al., 2020) arrays and SciPy (Virtanen et al., 2020) sparse matrices. We validated edgePython with 87 unit tests across 15 test files comprising 4,344 lines of test code. These tests cover every major component of edgeR and include direct numerical comparisons against reference outputs generated by running the same analyses in R. Each test verifies that the edgePython matches the edgeR result within a relative tolerance of 10^{-3} or better.

To demonstrate end-to-end concordance on real data, we first applied both edgeR and edgePython to the HOXA1 knockdown dataset (Trapnell et al., 2010) that consists of six RNA-seq libraries (three scrambled controls, three HOXA1 knockdowns) quantified at the transcript level with kallisto (Bray et al., 2016; Melsted et al., 2021; Sullivan et al., 2025), yielding 207,175 transcripts of which 44,514 passed expression filtering. TMM normalization factors computed by both implementations agree to machine precision (1a). Tagwise biological coefficients of variation (BCV $= \sqrt{\phi}$) estimated by the full dispersion workflow, including common, trended, and tagwise shrinkage via weighted likelihood empirical Bayes, show near-perfect concordance across all transcripts (1b). A small streak of 178 low-expression transcripts (0.4% of the total) shows slightly lower BCV in Python than in R; these genes have nearly flat likelihoods, making the grid-based tagwise dispersion estimate sensitive to minor numerical differences in the interpolation step, but the discrepancy has no practical effect on downstream testing. Resolving this required tracing into edgeR's compiled C source code, where the `maximizeInterpolant` function finds the mode of a spline-interpolated log-likelihood surface via Forsythe–Malcolm–Moler cubic spline fitting followed by root-finding on the spline derivative. An initial naïve grid-search translation was improved by reimplementing the exact analytical approach, matching R's output to machine precision. The effect sizes also agree: QL F-test log-fold-changes fall on the identity line (1c), and the posterior variances s^2_{post} from the empirical Bayes quasi-likelihood moderation also match closely (1d). The exact test p-values for the knockdown effect (1e) and the quasi-likelihood F-test p-values from the GLM pipeline (1f) likewise fall on the identity line. Likelihood ratio test TREAT p-values, pooled across four log-fold-change thresholds, also agree (1g).

We next applied both tools to the GSE60450 mouse mammary gland dataset, which serves as the basis for a widely used Galaxy RNA-seq differential expression tutorial (Chen et al., 2016). This experiment has a considerably more complex design: 12 samples spanning six groups (basal and luminal cells at the virgin, pregnant, and lactating stages), with a full factorial design matrix. After filtering, 15,804 genes were retained. Quasi-likelihood F-test p-values for the basal pregnant-versus-lactating contrast (1h) and TREAT p-values with a log-fold-change threshold of 0.58 (1i) show excellent agreement between the two implementations. All six GLM coefficient estimates from the factorial design, including intercept, cell type, two status contrasts, and two interaction terms, match closely when pooled across genes (1j). The multi-degree-of-freedom interaction F-test, which tests a two-parameter hypothesis, also shows tight concordance (1k). Competitive gene set testing via camera (1l) and self-contained testing via fry (1m) also reproduce across all tested configurations. Two camera points for one gene set deviate modestly; this set contains several genes with very strong differential expression and very low dispersions, making the parametric test statistic sensitive to the same grid-interpolation differences noted above. The rank-based camera variant matches to four decimal places. Finally, we validated two additional analysis modes: differential transcript usage (DTU) at both the gene level (9,697 genes, 1n) and the exon level (40,958 individual tests, 1o), as well as the scaled analysis workflow that incorporates quantification uncertainty via overdispersion scaling for 44,811 transcripts (1p). Across the entire set of comparisons, the maximum observed relative difference in any p-value is below 10^{-3}.

[1] An analysis of the edgeR code reveals that 39% of the package is written in C, a fraction large enough that edgeR should be called edgeRC.

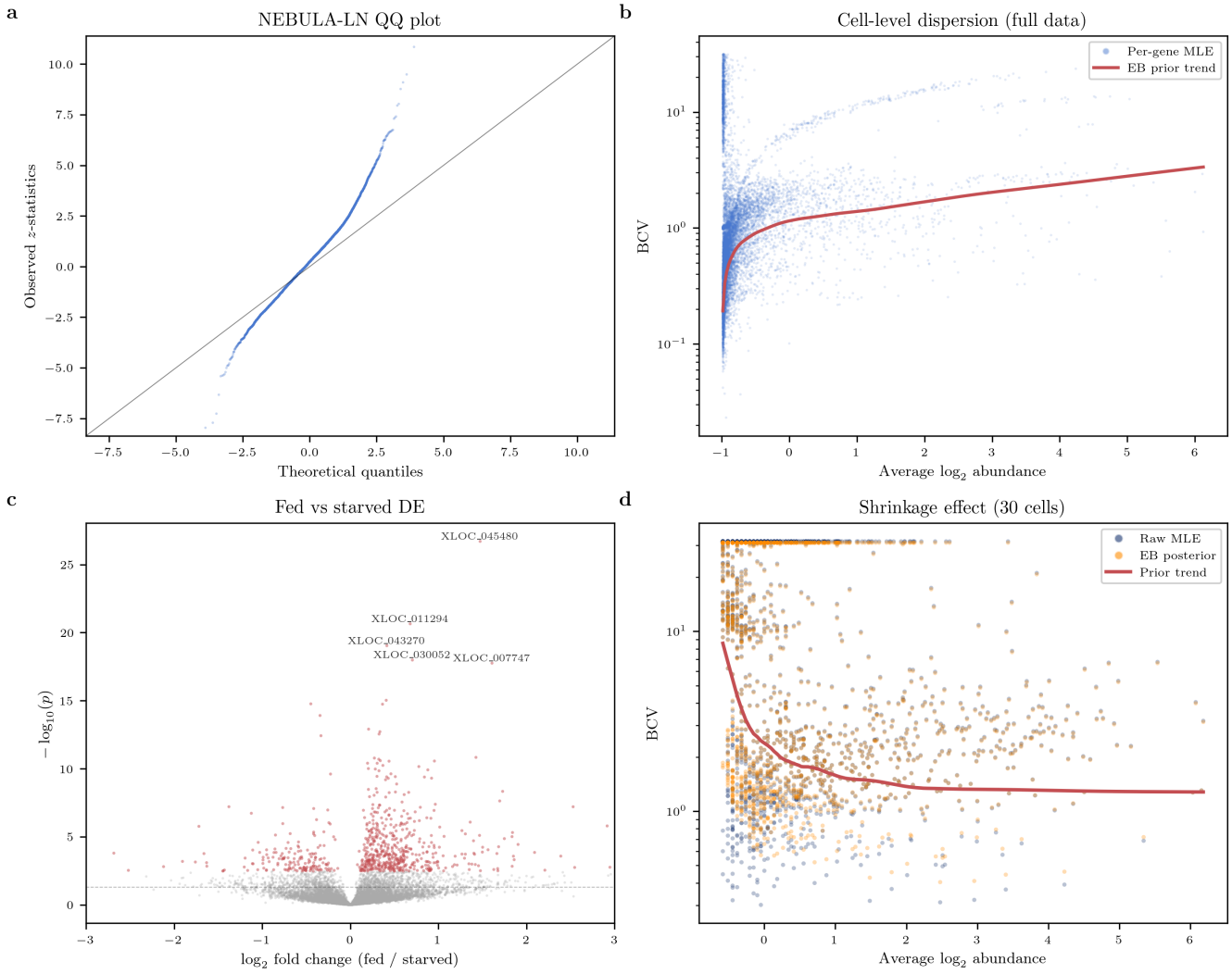

Figure 2: **Single-cell negative binomial mixed model with empirical Bayes shrinkage. (a)** Verification of the NEBULA-LN port: $-\log_{10}(p)$ from R NEBULA (x-axis) vs. edgePython (y-axis) on the same data. **(b)** Cell-level BCV ($\sqrt{\phi}$) as a function of average $\log_2$ abundance for 10,377 genes in *Clytia hemisphaerica* gastrodigestive cells (2,564 cells, 10 organisms). The red curve is the empirical Bayes prior trend. **(c)** Volcano plot of fed vs. starved differential expression. Colored points are significant at FDR < 0.05; key genes are labeled. **(d)** Effect of empirical Bayes shrinkage on a subsampled dataset (30 cells). Blue: raw MLE dispersions; orange: posterior estimates after shrinkage. The prior trend (red curve) stabilizes the noisy MLEs.

Negative binomial mixed model with empirical Bayes shrinkage

Multi-subject single-cell RNA-seq experiments present a statistical challenge not addressed by standard edgeR: the data contain both cell-level overdispersion and subject-level biological variation. Treating individual cells as independent replicates ignores the hierarchical structure and leads to inflated false positive rates (Crowell et al., 2020); a mixed model is needed.

We implement the NEBULA-LN negative binomial–gamma mixed model (He et al., 2021). Let $g = 1, \ldots, G$ index genes, $j = 1, \ldots, J$ index subjects, and $i = 1, \ldots, n_j$ index cells within subject j. The model is

$$Y_{gij} \mid b_{gj} \sim \mathrm{NB}\big(\mu_{gij}\, e^{b_{gj}},\ \phi_g\big),$$
$$b_{gj} \sim N(0,\ \sigma_g^2),$$

Here, Y_{gij} is the observed count for gene g in cell i from subject j. The conditional mean is $\mu_{gij} e^{b_{gj}}$, and we use the edgeR/NEBULA negative binomial parameterization

$$\mathrm{Var}(Y_{gij} \mid b_{gj}) = \mu_{gij} e^{b_{gj}} + \phi_g \big(\mu_{gij} e^{b_{gj}}\big)^2.$$

The log-mean model is

$$\log \mu_{gij} = \mathbf{x}_i^\top \boldsymbol{\beta}_g + \log N_{ij},$$

where $\mathbf{x}_i$ is the fixed-effect covariate vector for cell i (intercept and design terms), $\boldsymbol{\beta}_g$ is the corresponding gene-specific coefficient vector, and N_{ij} is the known library-size offset. The random effect b_{gj} is a gene-specific subject effect with variance σ_g^2, and ϕ_g is the gene-specific cell-level overdispersion parameter. TMM normalization (Robinson and Oshlack, 2010) is computed at the pseudobulk (subject) level and expanded to per-cell offsets. Parameters are estimated by a two-stage procedure: L-BFGS-B for initial values followed by penalized maximum likelihood with Newton–Raphson refinement, using a Laplace approximation to integrate out the random effects. The core likelihood and gradient functions are compiled with Numba (Lam et al., 2015) for performance.

The key methodological contribution beyond the NEBULA port is empirical Bayes shrinkage of the cell-level dispersion. The per-gene maximum likelihood estimates $\hat{\phi}_g$ are noisy, especially when the number of cells is modest. We apply the `squeezeVar` function, which is the same function that underlies edgeR's quasi-likelihood workflow, to borrow strength across genes: the raw $\hat{\phi}_g$ values are shrunk toward an abundance-dependent prior trend, yielding posterior estimates $\tilde{\phi}_g$ that are more stable. This approach is conceptually identical to the moderation of residual variances in limma (Smyth, 2004), but applied here to cell-level dispersion in the mixed model setting. Neither edgeR nor NEBULA currently performs this shrinkage step.

We verified our NEBULA-LN implementation by comparing the Python output against R's NEBULA package on the same dataset (2a), confirming that coefficient estimates, standard errors, and p-values agree. We then applied the full workflow to *Clytia hemisphaerica* (jellyfish) single-cell RNA-seq data from Chari et al. (2021): 2,564 gastrodigestive cells from 10 organisms (5 fed, 5 starved), with 10,377 genes passing expression filters. 2b shows the per-gene BCV ($\sqrt{\phi_g}$) as a function of average $\log_2$ abundance; the red curve is the empirical Bayes prior trend estimated by `squeezeVar`.

Differential expression analysis between the fed and starved conditions identified 689 genes with significant changes at FDR < 0.05 (2c). Of these, 524 are upregulated and 165 are downregulated in fed animals, a 3.2:1 ratio consistent with the expectation that gastrodigestive cells actively upregulate digestive and metabolic gene programs upon feeding. Several of the top differentially expressed genes were not highlighted in the original Chari et al. (2021) analysis, which used a different statistical framework, suggesting that the mixed model approach with empirical Bayes shrinkage can reveal additional biological signal.

To demonstrate the practical value of the shrinkage, we subsampled the data to 30 cells (2d). With fewer cells the per-gene MLE dispersions become markedly noisier, and the empirical Bayes posterior estimates are pulled substantially toward the prior trend. This regularization stabilizes inference in the small-sample regime, where it matters most.

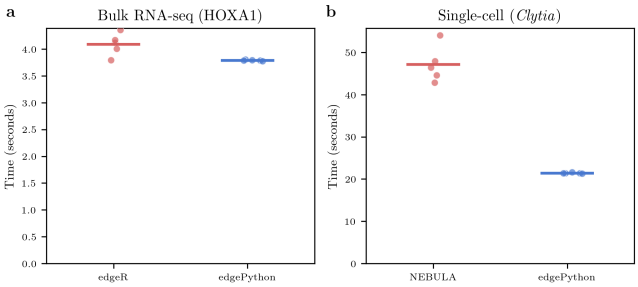

Figure 3: **Runtime comparison.** Elapsed time for the full differential expression pipeline in R (edgeR/NEBULA) and Python (edgePython) on the HOXA1 bulk dataset and the *Clytia* single-cell dataset.

Discussion

We have presented edgePython, a broad Python port of edgeR 4.8.2 that reproduces the original R implementation across normalization, dispersion estimation, GLM fitting, hypothesis testing, and gene set analysis (1), and extends the framework with a negative binomial–gamma mixed model for multi-subject single-cell differential expression (2). The port of edgeR and its extension are non-trivial. Notably, there has been a previous attempt to port edgeR (edgePy), but the project was ultimately unsuccessful and was aborted with the repository formally archived in January 2025. A project to port DESeq2 (Muzellec et al., 2023) remains active, but the port is not exact, nor does it support all DESeq2 (Love et al., 2014) functions. Although inMoose (Colange et al., 2025) provides a more exact port of edgeR and DESeq2, it is not complete, with three quarters of edgeR's functionality unavailable despite years of work on the project.

The performance of edgePython at runtime for bulk RNA-seq analysis is comparable to that of edgeR (3). For the single-cell mixed model, edgePython's Numba-compiled likelihood and gradient evaluations are substantially faster than R's NEBULA package. This speed advantage grows with the size of the dataset, making the Python implementation practical for large-scale single-cell atlases.

The edgePython program was developed with the assistance of Claude, a large language model. The faithful translation of a complex statistical package, including numerical optimization routines, special functions, and extensive C code, demonstrates that LLMs can serve as effective tools for cross-language software porting. This suggests that translating edgeR to additional languages is now straightforward: for example, a Rust implementation could integrate with the emerging Rust ecosystem for single-cell genomics or a CUDA implementation could leverage GPUs. The wildcard in "edge*" reflects this

idea: the asterisk stands for any target language, given that the barrier to porting has been dramatically reduced by modern language models.

To further reduce the barrier to use, edgePython includes a Model Context Protocol (MCP) server that exposes the full analysis pipeline from data loading through normalization, dispersion estimation, model fitting, hypothesis testing, and visualization. These functions are exposed as tool calls accessible to AI agents, enabling fully automated differential expression analyses driven by natural language instructions. We have verified that agents can successfully run complete bulk and single-cell workflows through this interface.

Our edgePython implementation also provides I/O capabilities beyond those available in edgeR. In particular, kallisto HDF5 output can be imported directly via `catch_kallisto_h5()`, which reads bootstrap samples and estimates per-transcript overdispersion. Although edgeR has supported kallisto bootstrap input via `catchKallisto`, it has not supported HDF5 input, which has been a major barrier for kallisto users. Bidirectional conversion between edgePython's DGEList and AnnData (`to_anndata()`) or Seurat (`seurat_to_pb()`) objects integrates edgePython into the scverse (Virshup et al., 2023) and Seurat ecosystems (Hao et al., 2024).

Several limitations should be noted: `processAmplicons`, a specialized function for CRISPR screen barcode processing, is not ported. This function reads directly from FASTQ files, and we chose to maintain a clean separation in which edgePython always starts from a count matrix, leaving FASTQ processing to dedicated Python tools. The pathway analysis functions `goana` and `kegga`, which in R depend on organism-specific Bioconductor annotation databases, instead delegate to the g:Profiler REST API (Raudvere et al., 2019), providing a language-agnostic interface with broad organism coverage. The NEBULA-HL (higher-order Laplace) method is not implemented; only the NEBULA-LN approximation is currently available. NEBULA-HL uses second-order corrections to the Laplace approximation to improve the estimation of the subject-level random effects variance, primarily benefiting settings with few subjects or large variance components. However, the empirical Bayes shrinkage of cell-level dispersion that we introduce already stabilizes inference in precisely these small-sample regimes, reducing the marginal benefit of the higher-order correction. The `voomLmFit` function (Law et al., 2014), which transforms counts to log-CPM and applies limma's linear modeling framework with precision weights, was also not ported. This function depends on substantial limma infrastructure (`lmFit`, `eBayes`, `duplicateCorrelation`, `arrayWeights`) that is not required by the negative binomial GLM pipeline. For standard differential expression analyses the quasi-likelihood workflow produces equivalent results, but `voomLmFit` offers support for within-subject blocking via `duplicateCorrelation` that has no current edgePython equivalent. Porting the necessary limma functions is straightforward and can be performed for a future release. Future releases should also correct several bugs that were identified in the edgeR software, but that were left intact so that the first release of edgePython could be as faithful as possible to edgeR.

Finally, it is worth noting a conceptual limitation of count-based differential expression methods in general. edgePython, like edgeR, operates on a single count matrix, yet for single-cell RNA-seq experiments it is now possible and preferable to quantify both spliced and unspliced RNA, yielding two coupled count matrices whose joint distribution is governed by the underlying transcriptional dynamics (Gorin et al., 2023). Methods such as Monod (Gorin et al., 2025) fit biophysically motivated stochastic models to these multimodal counts, enabling inference about transcriptional burst kinetics and regulatory state that a single-matrix framework cannot access. However, there remains a large corpus of legacy datasets for which only a single count matrix is available, and many current assays, including combinatorial indexing protocols, spatial transcriptomics, and CRISPR screens, inherently produce a single layer of counts. For these settings, a well-validated count-based differential expression tool integrated into the Python ecosystem fills a practical need, and edgePython is designed to serve this role.

Methods

Software porting

The edgePython program was developed by translating the R source code of edgeR 4.8.2 (Chen et al., 2025) (Bioconductor release 3.20) and its dependencies in the limma, statmod, and locfit packages into Python. The porting was performed with the help of Claude (Anthropic), using the Opus 4.5 and Opus 4.6 models. For each function, the R implementation, including C source when applicable, was provided to the model along with the corresponding R documentation and test cases. Claude produced an initial Python translation that was then validated against the R reference outputs. Iterative debugging resolved numerical discrepancies, typically arising from differences in optimization defaults, special function implementations, or floating-point edge cases. Performance-critical inner loops, particularly in the NEBULA-LN likelihood evaluation and the weighted LOWESS fitting used by the quasi-likelihood pipeline, were compiled with Numba `@njit`. The complete port comprises 15,318 lines of Python across 24 modules, validated by 87 unit tests (4,344 lines of test code) that compare against R reference outputs.

Datasets

HOXA1 knockdown. Six human RNA-seq libraries (three scrambled controls, three HOXA1 knockdowns)

from Trapnell et al. (2010) were quantified at the transcript level with kallisto (Bray et al., 2016; Melsted et al., 2021; Sullivan et al., 2025), yielding 207,175 transcripts. After filtering with `filterByExpr` (default parameters: minimum count 10, minimum total count 15), 44,514 transcripts were retained for analysis. This dataset was used for all panels in 1 labeled "HOXA1" as well as the bulk runtime benchmarks in 3.

GSE60450 mouse mammary gland. Twelve mouse mammary gland RNA-seq libraries spanning a 2×3 factorial design (basal and luminal cells at the virgin, pregnant, and lactating stages; two replicates per group) were obtained from GEO accession GSE60450 (Fu et al., 2015). Gene-level counts for 27,179 genes were filtered with `filterByExpr` to 15,804 genes. This dataset was used for the mouse mammary panels in 1 (panels h–k) and the GLM coefficient comparison (panel j).

***Clytia hemisphaerica* single-cell.** Single-cell RNA-seq data from Chari et al. (2021) were obtained from CaltechDATA (accession mm6y6-g4569). The complete data set contains 13,673 cells and 28,514 genes. We subset to 2,564 gastrodigestive cells from 10 organisms (5 fed, 5 starved). Genes were filtered by the NEBULA-LN defaults: minimum mean counts per cell > 0.005 and expression in ≥ 5 cells, yielding 10,377 genes for analysis.

Bulk RNA-seq analysis

For the HOXA1 dataset, both edgeR and edgePython were run with identical pipelines: TMM normalization, estimation of common, trended, and tagwise dispersions via `estimateDisp`, and hypothesis testing via exact test, quasi-likelihood F-test, likelihood ratio test with TREAT (at log-fold-change thresholds of 1.0, 1.2, and 1.5), camera, and fry. For the GSE60450 dataset, a full factorial design matrix was fitted using `glmQLFit` with robust empirical Bayes moderation. Contrasts were tested via `glmQLFTest` (single- and multi-degree-of-freedom) and `glmTreat` (log-fold-change threshold 0.58). Differential transcript usage analysis used `diffSpliceDGE` with exon-level F-tests and gene-level Simes aggregation. The scaled analysis incorporated per-transcript overdispersion estimates from kallisto bootstrap samples.

Single-cell analysis

The NEBULA-LN mixed model was fitted using `glm_sc_fit` with TMM normalization computed at the pseudobulk (subject) level and expanded to per-cell offsets. Parameters were estimated by L-BFGS-B initialization followed by penalized maximum likelihood with Newton–Raphson refinement. Dispersion bounds were set to $\phi \in [10^{-4}, 1000]$ and $\sigma^2 \in [10^{-4}, 10]$. Empirical Bayes shrinkage of cell-level dispersion was performed via `shrink_sc_disp` with robust estimation, using the `squeezeVar` framework with an abundance-dependent prior trend. In 2b, genes whose MLE dispersion hit the optimizer upper bound ($\phi \geq 999$) were excluded from the plot, as these represent genes too lowly expressed for reliable dispersion estimation.

Runtime benchmarks

All benchmarks were run on an Apple M1 processor with 16 GB RAM. Each workflow was timed for five replicates. The bulk benchmark comprised the full edgeR pipeline (TMM normalization, dispersion estimation, QL F-test) on the HOXA1 dataset. The single-cell benchmark comprised NEBULA-LN fitting on the *Clytia* gastrodigestive subset. R benchmarks used edgeR 4.8.2 and NEBULA 1.5.3. Python benchmarks used edgePython with NumPy 1.22, SciPy 1.11, and Numba 0.58.

Manuscript

The manuscript was prepared as a PDF/UA-2-compliant PDF using LuaLaTeX with tagged-PDF support via `\DocumentMetadata` and the PDF management/tagging framework in modern TeX Live. Conformance was validated with veraPDF (version 1.28) using the PDF/UA-2 profile.

Code availability

The edgePython code is available at https://github.com/pachterlab/edgePython. It can be installed with `pip install edgepython`.

Acknowledgments

The project resulted from a request by Sina Booeshaghi for a Python implementation of edgeR. Thanks to Sina Booeshaghi for also providing invaluable suggestions on how to use Claude effectively and for providing important feedback throughout the project. The work was supported in part by the Caltech Bioinformatics Resource Center.

Reflection

This project was completed in one week with Claude Opus 4.5 and Opus 4.6 performing the coding. As someone who had not coded seriously for more than 20 years, I found Claude's efficiency and accuracy enabling in a way that would have been inconceivable a few years ago. At this point, it is reasonable to predict that what took a week for this project may soon take less than a day. This raises the question of what the role of preprints, conference proceedings, and journal publications will be in the scientific enterprise of the future (Zeilberger, 1993).

References

Pedro L Baldoni, Yunshun Chen, Soroor Hediyeh-zadeh, Yang Liao, Xueyi Dong, Matthew E Ritchie, Wei Shi, and Gordon K Smyth. Dividing out quantification uncertainty allows efficient assessment of differential transcript expression with edgeR. *Nucleic Acids Research*, 52(3):e13–e13, 2024. ISSN 1362-4962. doi: 10.1093/nar/gkad1167. URL http://dx.doi.org/10.1093/nar/gkad1167.

Nicolas L. Bray, Harold Pimentel, Páll Melsted, and Lior Pachter. Near-optimal probabilistic RNA-seq quantification. *Nature Biotechnology*, 34(5):525–527, 2016. ISSN 1546-1696. doi: 10.1038/nbt.3519. URL https://doi.org/10.1038/nbt.3519.

Tara Chari, Brandon Weissbourd, Jase Gehring, Anna Ferraioli, Lucas Leclère, Makenna Herl, Fan Gao, Sandra Chevalier, Richard R. Copley, Evelyn Houliston, et al. Whole-animal multiplexed single-cell RNA-seq reveals transcriptional shifts across *Clytia* medusa cell types. *Science Advances*, 7(48), 2021. ISSN 2375-2548. doi: 10.1126/sciadv.abh1683. URL http://dx.doi.org/10.1126/sciadv.abh1683.

Yunshun Chen, Aaron T. L. Lun, and Gordon K. Smyth. *Differential Expression Analysis of Complex RNA-seq Experiments Using edgeR*, pages 51–74. Springer International Publishing, 2014. ISBN 9783319072128. doi: 10.1007/978-3-319-07212-8_3. URL http://dx.doi.org/10.1007/978-3-319-07212-8_3.

Yunshun Chen, Aaron T. L. Lun, and Gordon K. Smyth. From reads to genes to pathways: differential expression analysis of RNA-Seq experiments using Rsubread and the edgeR quasi-likelihood pipeline. *F1000Research*, 5:1438, 2016. ISSN 2046-1402. doi: 10.12688/f1000research.8987.2. URL http://dx.doi.org/10.12688/f1000research.8987.2.

Yunshun Chen, Lizhong Chen, Aaron T. L. Lun, Pedro L. Baldoni, and Gordon K. Smyth. edgeR v4: powerful differential analysis of sequencing data with expanded functionality and improved support for small counts and larger datasets. *Nucleic Acids Research*, 53(2):gkaf018, 2025. ISSN 1362-4962. doi: 10.1093/nar/gkaf018. URL https://doi.org/10.1093/nar/gkaf018.

Maximilien Colange, Guillaume Appé, Léa Meunier, Solène Weill, Akpéli Nordor, and Abdelkader Behdenna. Differential expression analysis with inmoose, the integrated multi-omic open-source environment in Python. *BMC Bioinformatics*, 26(1):160, 2025. ISSN 1471-2105. doi: 10.1186/s12859-025-06180-7. URL https://doi.org/10.1186/s12859-025-06180-7.

Helena L. Crowell, Charlotte Soneson, Pierre-Luc Germain, Daniela Calini, Ludovic Collin, Catarina Raposo, Dheeraj Malhotra, and Mark D. Robinson. muscat detects subpopulation-specific state transitions from multi-sample multi-condition single-cell transcriptomics data. *Nature Communications*, 11:6077, 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-19894-4. URL http://dx.doi.org/10.1038/s41467-020-19894-4.

Greg Finak, Andrew McDavid, Masanao Yajima, Jingyuan Deng, Vivian Gersuk, Alex K. Shalek, Chloe K. Slichter, Hannah W. Miller, M. Juliana McElrath, Martin Prlic, et al. MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome Biology*, 16:278, 2015. ISSN 1474-760X. doi: 10.1186/s13059-015-0844-5. URL http://dx.doi.org/10.1186/s13059-015-0844-5.

Nai Yang Fu, Anne C. Rios, Bhupinder Pal, Rina Soetanto, Aaron T. L. Lun, Kevin Liu, Tamara Beck, Sarah A. Best, François Vaillant, Philippe Bouillet, et al. EGF-mediated induction of Mcl-1 at the switch to lactation is essential for alveolar cell survival. *Nature Cell Biology*, 17(4):365–375, 2015. ISSN 1476-4679. doi: 10.1038/ncb3117. URL https://doi.org/10.1038/ncb3117.

Gennady Gorin, John J. Vastola, and Lior Pachter. Studying stochastic systems biology of the cell with single-cell genomics data. *Cell Systems*, 14(10):822–843, 2023. ISSN 2405-4712. doi: 10.1016/j.cels.2023.08.004. URL https://doi.org/10.1016/j.cels.2023.08.004.

Gennady Gorin, Tara Chari, Maria Carilli, John J. Vastola, and Lior Pachter. Monod: model-based discovery and integration through fitting stochastic transcriptional dynamics to single-cell sequencing data. *Nature Methods*, 22(11):2286–2300, 2025. ISSN 1548-7105. doi: 10.1038/s41592-025-02832-x. URL https://doi.org/10.1038/s41592-025-02832-x.

Yuhan Hao, Tim Stuart, Madeline H. Kowalski, Saket Choudhary, Paul Hoffman, Austin Hartman, Avi Srivastava, Gesmira Mez, Efthymia Clancy, Efthymia Papalexi, et al. Dictionary learning for integrative, multimodal and scalable single-cell analysis. *Nature Biotechnology*, 42(2):293–304, 2024. ISSN 1546-1696. doi: 10.1038/s41587-023-01767-y. URL https://doi.org/10.1038/s41587-023-01767-y.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, et al. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020. ISSN 1476-4687. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

Liang He, Jose Davila-Velderrain, Tomokazu S. Sumida, David A. Hafler, Manolis Kellis, and Alexan-

der M. Kulminski. NEBULA is a fast negative binomial mixed model for differential or co-expression analysis of large-scale multi-subject single-cell data. *Communications Biology*, 4(1), 2021. ISSN 2399-3642. doi: 10.1038/s42003-021-02146-6. URL http://dx.doi.org/10.1038/s42003-021-02146-6.

Wolfgang Huber, Vincent J. Carey, Robert Gentleman, Simon Anders, Marc Carlson, Benilton S. Carvalho, Hector Corrada Bravo, Sean Davis, Laurent Gatto, Thomas Girke, et al. Orchestrating high-throughput genomic analysis with Bioconductor. *Nature Methods*, 12(2):115–121, 2015. ISSN 1548-7105. doi: 10.1038/nmeth.3252. URL https://doi.org/10.1038/nmeth.3252.

Siu Kwan Lam, Antoine Pitrou, and Stanley Seibert. Numba: a LLVM-based Python JIT compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, pages 1–6. ACM, 2015. doi: 10.1145/2833157.2833162. URL https://doi.org/10.1145/2833157.2833162.

Charity W. Law, Yunshun Chen, Wei Shi, and Gordon K. Smyth. voom: precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biology*, 15(2):R29, 2014. ISSN 1474-760X. doi: 10.1186/gb-2014-15-2-r29. URL https://doi.org/10.1186/gb-2014-15-2-r29.

Michael I. Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, 15(12):550, 2014. ISSN 1474-760X. doi: 10.1186/s13059-014-0550-8. URL https://doi.org/10.1186/s13059-014-0550-8.

Aaron T. L. Lun, Yunshun Chen, and Gordon K. Smyth. *It's DE-licious: A Recipe for Differential Expression Analyses of RNA-seq Experiments Using Quasi-Likelihood Methods in edgeR*, pages 391–416. Springer New York, 2016. ISBN 9781493935789. doi: 10.1007/978-1-4939-3578-9_19. URL http://dx.doi.org/10.1007/978-1-4939-3578-9_19.

Davis J. McCarthy, Yunshun Chen, and Gordon K. Smyth. Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation. *Nucleic Acids Research*, 40(10):4288–4297, 2012. ISSN 0305-1048. doi: 10.1093/nar/gks042. URL http://dx.doi.org/10.1093/nar/gks042.

Páll Melsted, A. Sina Booeshaghi, Lauren Liu, Fan Gao, Lambda Lu, Kyung Hoi Joseph Min, Eduardo da Veiga Beltrame, Kristján Eldjárn Hjörleifsson, Jase Gehring, and Lior Pachter. Modular, efficient and constant-memory single-cell RNA-seq preprocessing. *Nature Biotechnology*, 39(7):813–818, 2021. ISSN 1546-1696. doi: 10.1038/s41587-021-00870-2. URL https://doi.org/10.1038/s41587-021-00870-2.

Boris Muzellec, Maria Teleńczuk, Vincent Cabeli, and Mathieu Andreux. PyDESeq2: a python package for bulk RNA-seq differential expression analysis. *Bioinformatics*, 39(9):btad547, 2023. ISSN 1367-4803. doi: 10.1093/bioinformatics/btad547. URL https://doi.org/10.1093/bioinformatics/btad547.

Harold Pimentel, Nicolas L Bray, Suzette Puente, Páll Melsted, and Lior Pachter. Differential analysis of RNA-seq incorporating quantification uncertainty. *Nature Methods*, 14(7):687–690, 2017. ISSN 1548-7105. doi: 10.1038/nmeth.4324. URL https://doi.org/10.1038/nmeth.4324.

Uku Raudvere, Liis Kolberg, Ivan Kuzmin, Tambet Arak, Priit Adler, Hedi Peterson, and Jaak Vilo. g:Profiler: a web server for functional enrichment analysis and conversions of gene lists (2019 update). *Nucleic Acids Research*, 47(W1):W191–W198, 2019. ISSN 1362-4962. doi: 10.1093/nar/gkz369. URL https://doi.org/10.1093/nar/gkz369.

Matthew E. Ritchie, Belinda Phipson, Di Wu, Yifang Hu, Charity W. Law, Wei Shi, and Gordon K. Smyth. limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Research*, 43(7):e47–e47, 2015. ISSN 0305-1048. doi: 10.1093/nar/gkv007. URL http://dx.doi.org/10.1093/nar/gkv007.

M. D. Robinson and G. K. Smyth. Small-sample estimation of negative binomial dispersion, with applications to sage data. *Biostatistics*, 9(2):321–332, 2007a. ISSN 1468-4357. doi: 10.1093/biostatistics/kxm030. URL http://dx.doi.org/10.1093/biostatistics/kxm030.

Mark D Robinson and Alicia Oshlack. A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology*, 11(3), 2010. ISSN 1474-760X. doi: 10.1186/gb-2010-11-3-r25. URL http://dx.doi.org/10.1186/gb-2010-11-3-r25.

Mark D. Robinson and Gordon K. Smyth. Moderated statistical tests for assessing differences in tag abundance. *Bioinformatics*, 23(21):2881–2887, 2007b. ISSN 1367-4803. doi: 10.1093/bioinformatics/btm453. URL http://dx.doi.org/10.1093/bioinformatics/btm453.

Mark D. Robinson, Davis J. McCarthy, and Gordon K. Smyth. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140, 2010. ISSN 1367-4803. doi: 10.1093/bioinformatics/btp616. URL http://dx.doi.org/10.1093/bioinformatics/btp616.

Gordon K Smyth. Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, 3(1):1–25, 2004. ISSN

1544-6115. doi: 10.2202/1544-6115.1027. URL http://dx.doi.org/10.2202/1544-6115.1027.

Delaney K. Sullivan, Kyung Hoi Joseph Min, Kristján Eldjárn Hjörleifsson, Laura Luebbert, Guillaume Holley, Lambda Moses, Johan Gustafsson, Nicolas L. Bray, Harold Pimentel, A. Sina Booeshaghi, et al. kallisto, bustools and kb-python for quantifying bulk, single-cell and single-nucleus RNA-seq. *Nature Protocols*, 20(2):587–607, 2025. ISSN 1750-2799. doi: 10.1038/s41596-024-01057-0. URL https://doi.org/10.1038/s41596-024-01057-0.

Cole Trapnell, Brian A Williams, Geo Pertea, Ali Mortazavi, Gordon Kwan, Marijke J van Baren, Steven L Salzberg, Barbara J Wold, and Lior Pachter. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature Biotechnology*, 28(5): 511–515, 2010. ISSN 1546-1696. doi: 10.1038/nbt.1621. URL https://doi.org/10.1038/nbt.1621.

Isaac Virshup, Danila Bredikhin, Lukas Heumos, Giovanni Palla, Gregor Sturm, Adam Gayoso, Ilia Kats, Mikaela Koutrouli, Philipp Angerer, Volker Bergen, et al. The scverse project provides a computational ecosystem for single-cell omics data analysis. *Nature Biotechnology*, 41(5):604–606, 2023. ISSN 1546-1696. doi: 10.1038/s41587-023-01733-8. URL http://dx.doi.org/10.1038/s41587-023-01733-8.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272, 2020. ISSN 1548-7105. doi: 10.1038/s41592-019-0686-2. URL https://doi.org/10.1038/s41592-019-0686-2.

F. Alexander Wolf, Philipp Angerer, and Fabian J. Theis. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biology*, 19(1), 2018. ISSN 1474-760X. doi: 10.1186/s13059-017-1382-0. URL http://dx.doi.org/10.1186/s13059-017-1382-0.

Di Wu and Gordon K. Smyth. Camera: a competitive gene set test accounting for inter-gene correlation. *Nucleic Acids Research*, 40(17):e133, 2012. ISSN 1362-4962. doi: 10.1093/nar/gks461. URL https://doi.org/10.1093/nar/gks461.

Luke Zappia, Belinda Phipson, and Alicia Oshlack. Exploring the single-cell RNA-seq analysis landscape with the scRNA-tools database. *PLOS Computational Biology*, 14(6):e1006245, 2018. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1006245. URL https://doi.org/10.1371/journal.pcbi.1006245.

Doron Zeilberger. Theorems for a price: Tomorrow's semi-rigorous mathematical culture. *The Mathematical Intelligencer*, 15(4):11–14, 1993. doi: 10.1007/BF03024696. URL https://doi.org/10.1007/BF03024696.