nature reviews genetics

https://doi.org/10.1038/s41576-023-00679-6

Review article

Check for updates

Challenges and best practices in omics benchmarking

Thomas G. Brooks[1], Nicholas F. Lahens[1], Antonijo Mrčela[1] & Gregory R. Grant[1,2]

Abstract

Technological advances enabling massively parallel measurement of biological features — such as microarrays, high-throughput sequencing and mass spectrometry — have ushered in the omics era, now in its third decade. The resulting complex landscape of analytical methods has naturally fostered the growth of an omics benchmarking industry. Benchmarking refers to the process of objectively comparing and evaluating the performance of different computational or analytical techniques when processing and analysing large-scale biological data sets, such as transcriptomics, proteomics and metabolomics. With thousands of omics benchmarking studies published over the past 25 years, the field has matured to the point where the foundations of benchmarking have been established and well described. However, generating meaningful benchmarking data and properly evaluating performance in this complex domain remains challenging. In this Review, we highlight some common oversights and pitfalls in omics benchmarking. We also establish a methodology to bring the issues that can be addressed into focus and to be transparent about those that cannot: this takes the form of a spreadsheet template of guidelines for comprehensive reporting, intended to accompany publications. In addition, a survey of recent developments in benchmarking is provided as well as specific guidance for commonly encountered difficulties.

Sections

[1]Institute for Translational Medicine and Therapeutics, University of Pennsylvania, Philadelphia, PA, USA. [2]Department of Genetics, University of Pennsylvania, Philadelphia, PA, USA. e-mail: ggrant@pennmedicine.upenn.edu

Introduction

Omics technologies generate vast amounts of high-dimensional data that can be analysed using various computational methods and algorithms to extract meaningful biological information. The accuracy, sensitivity, specificity and efficiency of these analytical methods can vary substantially depending on the specific questions being asked and the input data. By performing rigorous benchmarking — comparing the performance of different methods of analysis — researchers can ensure that their analysis is based on the most vetted methods available, thus improving accuracy and replicability. Benchmarking has several aims, including: validating the accuracy and reliability of new computational methods and tools prior to their use in the real world; identifying the most appropriate methods for specific analytical tasks; clarifying the trade-offs between different methods, such as computational efficiency versus accuracy, to enable the most suitable approach to be chosen; and tracking progress and identifying areas for improvements to drive further methodological developments. Besides comparative analyses, benchmarking might also establish a bar that any method should reasonably need to clear to be viable for use in practice.

Over the past 25 years, the practice of benchmarking has been steadily maturing, with the publication of thousands of omics benchmarking studies. At this point, the basic tenets of benchmarking are established and well described[1–3]. Moreover, domain-specific benchmarking guidelines have been developed, such as for RNA secondary structure prediction[4], for microbiomics and metagenomics[5,6] and for variant calling[7]. There are benchmarks of simulators for benchmarking[8–11] and software with which to organize benchmarking[12–17]. However, challenges associated with benchmarking still exist and some challenges are more difficult to solve than others.

Consider three examples of analyses that are increasingly more difficult to benchmark: RNA sequencing (RNA-seq) alignment, bulk RNA-seq differential expression, and single-cell RNA-seq analyses. Benchmarking alignment workflows is relatively straightforward, as one can simply obtain sequencing reads from known genomic locations (usually through simulation) and assess how closely tools can determine the correct alignments. By contrast, a benchmark to assess differential expression from bulk RNA-seq is considerably more complicated because it requires realistic variation between samples, including biological variation and dependence in gene expression. As this step in the RNA-seq workflow is downstream of sequence alignment, it naturally inherits the difficulties associated with alignment benchmarking as well. Finally, benchmarks of some single-cell RNA-seq analyses must account for multiple cell types, populations of cells of each type within an organism, and variation in cell populations across individuals. Each additional step makes it more challenging to generate meaningful data, to evaluate results, and even to specify what exactly is the 'ground truth' situation.

Despite the importance of omics methods to modern biology, researchers routinely face the necessity of performing benchmarking studies that cannot achieve desired benchmarking ideals owing to their inherent limitations. On top of these unavoidable limitations, a literature review of benchmarking indicates that studies are often compromised by avoidable oversights (Box 1), which could be identified by standardized reporting and subsequently corrected. In this Review, we expand on the established guidelines of benchmarking by recommending structured reporting of all critical decisions to avoid the major pitfalls often encountered and to be explicit about all judgement calls and simplifications.

Specifically, we provide a structured form for reporting eleven types of critical factors common to most omics benchmarking studies (Supplementary File 1). Updates and domain-specific templates will be maintained on GitHub, where community contributions are encouraged. These reports are intended to aid benchmarkers, reviewers and readers. We believe that such reporting will not only document the inherent and unavoidable limitations of a benchmark in order to improve its interpretation but will also improve the benchmark by highlighting avoidable mistakes so that they can be corrected. In addition, we give specific guidance on navigating the challenging decisions reported in these forms. Last, we highlight recent developments, including benchmarking of new methodologies and the increasing number of available benchmarking platforms and software.

Ten steps

Benchmarking analyses can be described in ten steps (Box 2). These steps have been described previously and are well established[1]. Here, an emphasis is placed on developing standards of reporting, with the aim of revealing common issues, which can then be addressed, if possible, or documented in a transparent manner. It is not essential to follow the ten steps in strict numerical order, and it will usually be necessary to revisit earlier steps when issues arise during later stages. We address the various difficulties associated with each step below.

Although it is desirable, it is not realistic to systematize benchmarking. For example, there is no established method for selecting which implementations (referred to as 'tools' from here onwards) of a given method to include. Rationales for the choice of tool vary widely, and two papers benchmarking the same problem will invariably address different sets of tools. Although it is difficult to develop and achieve community adoption of systematic methods in benchmarking, considerable clarity would be achieved by introducing standards of transparency, including comprehensive reporting of choices and documentation of rationales. We provide a thorough template as a proposed standard for these reports (Fig. 1a and Supplementary File 1).

Throughout this Review, we use a recurring example of the typical RNA-seq pipeline, which begins with raw data (generated from wet bench work and sequencing), followed by read alignment, quantification, normalization and differential expression analysis. This intentionally does not represent the cutting-edge limits of complexity in omics analysis pipelines, but it gives a familiar example that illustrates many widely relevant general issues. Benchmarking on more recently developed assays presents greater challenges that still rely on these principles; Box 3 is a summary of the latest benchmarking developments.

Step 1: Scope

The first benchmarking challenge is to balance broadness of scope and feasibility of implementation. For example, it is desirable to determine broad recommendations for peak calling from chromatin immunoprecipitation followed by sequencing (ChIP–seq), but transcription factor binding versus histone modifications[18] are sufficiently different applications of ChIP–seq that benchmarking one would not generalize well to the other. These could be combined into one publication, but each would require their own benchmarking study, addressing each of the ten steps separately.

The scope of a benchmark study is also restricted with regard to factors such as species, study designs and sample sizes. For example, the results of benchmarking on mouse data cannot be assumed to generalize to plants or bacteria and would preferably need be justified to extend even to other mammals. Ideally, any generalization should be supported by multiple data sets sufficient to provide convincing

evidence for why the results can be extended. Another example is that differential expression across very large numbers of replicates, such as in biobanks, calls for a dedicated benchmark study, because results from smaller sample sizes are not likely to generalize[19].

Because most omics methods involve multiple steps in a pipeline, benchmarkers often must consider whether to benchmark one specific step or several pipeline steps jointly, or even entire pipelines (Fig. 1b–d). In addition, they must identify the downstream goals of the benchmarking analysis. Some upstream steps, such as alignment, are common to many pipelines and so it may make sense to evaluate that step in isolation. On the other hand, normalization of RNA-seq quantifications is invariably tied to the downstream application, typically differential expression; it therefore only makes sense to evaluate normalization in terms of its impact on differentiation expression. Thus, it is necessary to identify any final downstream goals that will come into play in the evaluation step (Step 8) and upstream steps that will come into play in generating benchmark data (Step 6). This context should be part of systematic reporting and we have included tables to this end as part of the report template (Supplementary File 1).

Step 2: Tools

Benchmarkers must next delineate exactly which tools and specific versions are to be included. This should start with a search of the literature or community-maintained lists of tools[20,21], such as the single-cell RNA-tools database for single-cell RNA-seq[22]. Once an exhaustive list has been assembled, each tool should be evaluated for inclusion. Given that interpretation also depends on which tools were excluded, a table should be maintained that documents a justification for each excluded method. Reviewers can then judge whether methods were excluded for poor reasons. A suitable justification for exclusion might be that they are poorly documented, the software is un-runnable, or that the relevance is questionable. However, relevant popular software should be included regardless. An example of a poor criterion is to include only methods implemented in Python[23]. If criteria are conflicting, then benchmarkers must make judgement calls and document their decisions. Benchmarkers should try to reach out to the authors of the various tools to clarify issues with documentation or running the software. This does not necessarily introduce bias, but it factors in the availability of support, which itself is an important property of performance in actual use. We refer the reader to a previous article for further guidance[1].

A final consideration is the neutrality of the benchmarkers with respect to the assessed tools. When benchmarkers are themselves developers of one or more of the tools, then benchmarks have been shown to be less objective[24,25]. If benchmarkers are not neutral, then an alternative approach is to solicit input from the developers of all tools to ensure that all are used optimally and to minimize bias. Any vested interest in the tools being evaluated should be clearly reported.

Step 3: Ground truth

The usual approach to benchmarking is to obtain data for which the truth is known and can be meaningfully compared to the results produced (either directly or downstream) by the tools. In this step, one must try to identify exactly what it is that will be assessed for accuracy. This could be the true values of metrics or other inferred biological features such as alignments, molecular modifications or the true/false status of hypotheses.

The ground truth must be defined in the proper context for the benchmarking analysis at hand. For example, if the statistical analysis is testing hypotheses at the population level (such as for a differential expression analysis), then the ground truth must also be defined at the population level. This highlights the fact that benchmarking such tests requires using repeated full experiments to assess the variability of the evaluation metrics (Step 8), which is not always carried out[26,27].

At times, there might be multiple valid choices of ground truth for a specific problem. For example, when benchmarking differential expression methods, the ground truth may be either the true differentially expressed gene list or the true effect sizes (fold change in gene expression). Moreover, ground truth cannot always mirror the output of the tool; for example, tools that produce *P* values have ground truths that are the true/false statuses of the hypotheses, not the *P* values themselves.

In certain cases, there may be no way to obtain a ground truth, but benchmarkers should still consider what form it would take if available. In Step 8, we discuss what can be done in the case where the ground truth is unknown.

Box 1

Common pitfalls in benchmarking

Avoidable errors in benchmarks that we have observed include:

- Using data sets that poorly represent actual experiments (such as RNA universal reference samples) (Box 4)
- Using technical replicates as if they were true biological replicates (Box 4)
- Assuming independence, for example, between different genes or between different genomic loci
- Wrong or overly simplified parametric assumptions (such as normality)
- Generating or assessing data using the same model as one or more of the evaluated tools, thus biasing towards those tools
- Using training data from an evaluated tool as benchmarking data
- Over-generalizing results without including sufficient data sets to evaluate performance in that domain
- Criticizing non-problematic properties, such as assessing *P* value distributions for uniformity when one-sided tests that lack uniformity will still have valid *P* values
- Comparing to a 'gold standard' data set that was generated using one of the tools
- Presenting results as comprehensive despite critical omissions (such as not evaluating an important tool or use case)
- Overinterpretation or misinterpretation of correlation or other evaluation metrics
- Declaring a winner even if all the tools evaluated failed to meet the standards necessary for practical application
- Wrong choice of data model (such as assuming that differential expression happens at the level of one data set and not modelling population-level truth)
- Focusing on intermediate results when downstream results are what researchers depend upon (such as only assessing quantification accuracy when practical applications require differential expression results)

Step 4: Accuracy of modelling

Once the relevant tools have been identified (Step 2), one must identify the properties of the benchmarking data that are relevant. For example, benchmarking alignment requires realistic reads that cover the panoply of relevant alignments. In RNA-seq alignment, it is not necessary to factor realistic quantified gene-expression levels into simulated data because no tools make use of that information. In fact, it might be detrimental to have perfectly realistic data in this application because it might obscure important edge cases. Continuing the example, benchmarking alignment requires realistic reads, whereas benchmarking quantification requires realistic samples, and benchmarking differential expression requires realistic populations of samples. For differential expression analysis, this could include factors ranging from read sampling noise, biases from library preparation that may vary from library to library, and the actual biological variation in true gene expression between replicates and across treatment groups. Depending on the experimental setup, these factors vary in importance (for example, when there is lower biological variation in cell lines than in laboratory animal studies and higher variation in human studies). After data are acquired, they should be compared on these criteria to real-world data to assess their accuracy (Step 6).

Computational tools often make simplifying assumptions about biological data. For example, the BLAST[28] sequence aligner assumes that positions in the sequences are identically and independently distributed from their neighbours. Another common aspect of informatics tools in general is to make parametric assumptions, such as normality. Assumptions are often made necessary by the complexity of the task being benchmarked; however, using benchmark data that employ the same assumptions as the tools can cause bias and they must be avoided unless rigorously justified. Best practice should therefore require using data as realistic and complete as possible while clearly documenting all simplifying assumptions of the benchmark data as well as the tools.

Step 5: Results stratification

The next step is to identify aspects of the model that vary in practice and that probably affect performance of the tools, which we refer to as stratification parameters. Examples of stratification parameters include the number of replicates, sequencing depth, experimental error rates, effect sizes and evolutionary distance. Any such parameters should be evaluated by varying them and stratifying the results. As an example, when benchmarking differential expression, it should at minimum cover the most common study designs consisting of three to eight replicates per group. One advantage of simulated data is that it often allows for much greater control over these parameters[29]. Benchmarkers should document any relevant parameters that are omitted, as well as the range of values used for the parameters that are addressed. As it is generally not feasible to vary factors in all possible combinations, benchmark studies should at least vary them individually to assess their isolated impact.

Step 6: Data

The easiest mistake to make in benchmarking is to generate misleading or meaningless benchmarking data (Box 4). Benchmarking data will be used as input to the tools along with any corresponding ground-truth values to be used for evaluation. As such, it should reflect the established ground truth (Step 3), have realistic properties (Step 4) and allow variation of the relevant stratification parameters (Step 5). Step 6 can therefore demand the majority of the benchmarker's effort.

Box 2

Ten steps of benchmarking

Brief descriptions of the ten steps of a typical benchmarking study in scienceare provided below. Clarifying details and related issues are discussed in the main text.

1. **Scope:** Delineate exactly what problem is being evaluated, including which stages of any pipelines are to be benchmarked and whether to include the impact on the outputs of any downstream (dependent) analyses.
2. **Tools:** Determine the specific tools (typically software packages) that will be evaluated. This may include combinations of tools or entire pipelines.
3. **Ground truth:** Determine the form of reference or true data that will be compared to the output of the tools. In cases where no ground truth is available, this step can sometimes be circumnavigated.
4. **Accuracy of modelling:** Identify which properties of real-world experiments are important to capture during benchmarking. Also identify any properties that will be knowingly unrealistic in the benchmark data.
5. **Results stratification:** Identify any relevant factors that should be varied to assess their impact on performance.
6. **Data:** Create, find or combine data to form the benchmarking data set, which is to be used as input to all the tools during evaluation along with any reference ground-truth values as identified in Step 3. Ensure that the benchmarking data set conforms to the properties identified in Step 4 and generate multiple data sets, varying the stratification parameters identified in Step 5.
7. **Execution:** Apply each tool to the benchmarking data sets created in Step 6 and collect the results. Repeat with varying configurations for each tool if necessary.
8. **Evaluation:** Measure tool performance by one or more evaluation metrics that compare the outputs generated in Step 7 to the data set's ground truth. If no ground truth is available, compare using other performance criteria, across replications, or with other, independent assays.
9. **Interpretation:** Synthesize the results of the evaluation, generating recommendations to users of optimal tools, pipelines, configurations or study designs. Identify future directions for tool improvement.
10. **Maintenance:** Share data and software. Enable easy future reproduction. Update to reflect new tools, versions or other developments in the field.

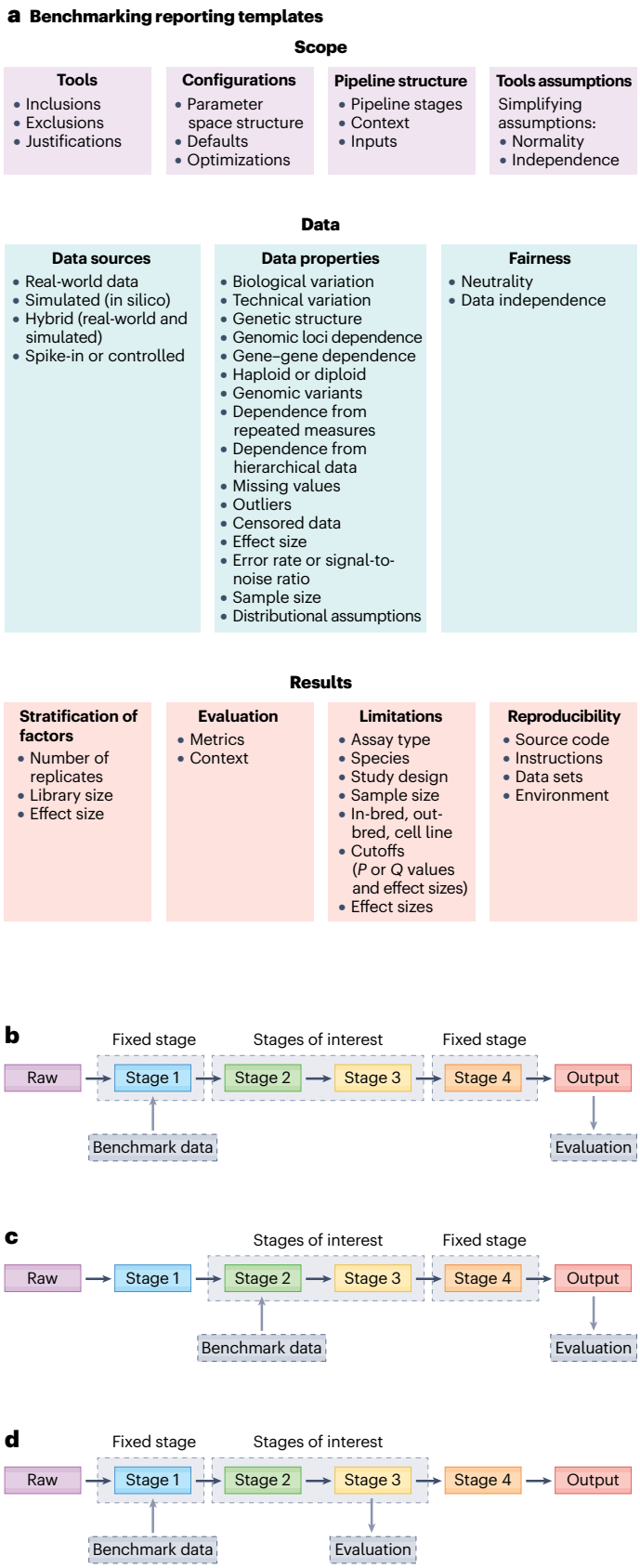

Fig. 1 | Overview of benchmarking report template and benchmarking a pipeline. a, Summary of the report template components, which are organized into three general categories: scope, data and results. Each of the 11 items requires the benchmarker to provide specific information detailing relevant lists (tools, parameters and so on) and explicit documentation of judgement calls and simplifying assumptions. This is designed to reveal common pitfalls that can then be addressed where possible and transparently documented otherwise. **b**–**d**, Typical omics computational methods consist of a pipeline of several stages, although benchmark studies may investigate an abbreviated or modified pipeline (dashed grey boxes). Benchmarkers must identify which stages of the pipeline will be directly benchmarked. All other stages will then either be omitted from the benchmark or will be fixed to have just one or a few configurations. Consider the following schemes for pipeline benchmarking. **b**, Evaluate final output from the entire pipeline. The choice of procedure for fixed stages influences the results of the benchmark analysis. **c**, Evaluate final output from a partial pipeline. Stages that have been omitted still need to be considered as sources of error or bias in bench experiments that must be reflected in the benchmark data for a meaningful result. **d**, Evaluate output from an intermediate pipeline stage. Choosing to evaluate results directly from an intermediate stage allows precise identification of where mistakes arise and could generalize to multiple downstream stages, but evaluation of a downstream goal reflects the impact on bench experiments better.

Types of benchmarking data

Approaches to generating benchmark data generally fall into four categories[1,2] (Box 5):

1. Real-world data (possibly with independent validation)
2. Simulated data (with well defined ground truth)
3. Semi-synthetic data (modified real data, or integration of real and simulated data)
4. Controlled experimental data (such as ground truth established by spike-ins, which refers to adding known quantities of biomolecules of interest into real samples)

It is often best to adopt multiple benchmarking approaches. Moreover, for a given data type, a detailed review should be performed of previous benchmarking efforts. Rarely will you be the first to generate benchmark data of a given type and benchmarkers should leverage the experience of predecessors.

Pipeline structure

Most omics methods are part of a greater analysis pipeline. Benchmarkers may evaluate entire pipelines, in which case their input data would need to resemble raw data. However, benchmark studies typically evaluate just one step in a pipeline and must choose from which upstream step of the pipeline to generate benchmarking data, keeping in mind the scope identified in Step 1. For example, an RNA-seq differential expression benchmark study could simulate sequencing reads for input into the alignment stage or could simulate a spreadsheet of quantified read counts directly.

Generating benchmarking data for the most upstream pipeline stage captures artefacts of upstream stages that might not be represented in data generated directly for the stage of primary interest. For example, starting with a spreadsheet of quantified RNA-seq read counts does not account for any alignment artefacts that could affect quantification. Generating raw reads inherently captures such effects. However, this introduces the necessity of processing data through earlier stages of the pipeline, which will necessitate choosing algorithms for those stages. Ultimately, this could affect the stages of interest, introducing yet another dimension of options that needs to be considered.

Box 3

Frontiers of benchmarking

Emerging trends in modern omics have increased both the challenges and opportunities in benchmarking. Three increasingly pervasive methodologies in omics are single-cell omics, spatial omics and multi-omics, along with combinations of these techniques. Each presents an increase in the complexity of benchmarking compared to earlier methods. One challenge is simply the explosion of tools and objectives that has followed these developments, although improved community aggregation of these methods[22] and standards for data and processing[122–124] do help to manage such complexity. More fundamentally, it is challenging to generate gold-standard real data reference samples for these data types[125], yet realistic data simulation is complex and benchmarking results are sensitive to the choices made in simulation[8]. In single-cell RNA-sequencing, for example, the so-called 'zero-inflation controversy' — whether to view zeros as true biological signals from unexpressed or low-expressed genes, or as missing data that need to be corrected — remains hotly debated and the choice influences simulator design and benchmarking results[126,127]. Other single-cell assays, such as in proteomics and whole-genome amplification, also suffer from increased missing data compared to their bulk equivalents[128,129], complicating the generation and evaluation of simulated data. The increased complexity of data types has necessitated sophisticated simulation of factors, including distinct cell types and their connectivity[130] and realistic trajectories of cells[131].

Other challenges include the large-batch effects that integrative methods must overcome; also, the decision of what counts as a batch effect versus true biological difference is context-dependent and complicated, further challenging benchmarkers[132].

Despite these increasingly growing hurdles, benchmarkers have continued to innovate and have demonstrated that data complexity also presents increased opportunities for evaluating method performance. For example, multi-modal single-cell data (such as data sets that include both gene expression and histone-modification assays from each cell) have been used to evaluate methodology on a single mode using the other mode as a reference, via neighbour similarity[133–135]. Similarly, integrated spatial and single-cell transcriptomics has been used to benchmark performance of ceaell interaction methodology by incorporating known constraints on the length of cell–cell interactions[136], and single-cell data has been used to generate realistic spatial transcriptomic data[132].

A particularly impressive effort is the Open Problems in Single-Cell Analysis project, which aims to formalize problems into community challenges that promise to become the community benchmarking standards for their topics. Another consortium, SpaceTx[137], has begun to systematize the benchmarking of spatial transcriptomics in brain tissues[138]. These projects collectively show how the benchmarking community is growing to meet new challenges.

By contrast, generating benchmarking data directly for later pipeline stages generally requires fewer computational resources, meaning that a greater number of iterations can be run to produce a more complete picture. Later stages may also be easier to parameterize in terms of directly relevant biological variables, such as population mean and variance. However, generating data for later steps will require simplifications and can easily lead to oversimplification, resulting in unreliable conclusions.

The extra effort required to generate multiple benchmark data sets of varying data types or at multiple pipeline stages might therefore be justified. Ultimately, judgement calls are unavoidable and should be transparently documented (Supplementary File 1).

Benchmarking data bias

Biases in benchmarking data can affect the reliability of results. Common biases are therefore compiled in Supplementary File 1. One bias, described in Step 4, arises when tools and benchmarking data rely on the same assumptions[30]. Additionally, it is important to ensure that the benchmark data in hand were not used for the development of any of the evaluated tools. Data sets well suited to benchmarking are also popular among tool developers and may end up being used as training data or during tool parameter optimization and fine-tuning of algorithms. Benchmarks using these data as reference could then favour tools that used those same data sets during development, and the benchmark results might not reflect performance in an independent data set[31,32]. Therefore, benchmarkers must be selective when using established data sets.

Subtler problems can arise when benchmark data are indirectly related to the training data. For example, RNA secondary structure prediction benchmark studies have often used cross-validation techniques[33]. However, RNA sequences from the same family were included in both training and validation data sets, and since secondary structure is often conserved, measured performance can be overstated compared to actual performance in novel RNA families.

Conversely, benchmarkers should avoid generating data with the aid of any of the tools under evaluation, to avoid the benchmark analysis being biased towards that tool. However, generating semi-synthetic data may unavoidably require invoking some of the tools under evaluation. One possible solution would be to control for this bias by employing all tools equally.

Variance within benchmark data sets

If the analysis being benchmarked is statistical in nature (for example, any tool that outputs *P* values), then it is important that variation captured in the benchmark data properly reflects variation present in real-world data sets[34]. Examples of too little variation include using only technical replicates when benchmarking differential expression tools (Box 4), or generating multiple samples by subsampling reads of a single data set. By contrast, an example of too much variation would be performing differential expression analysis on two or more dramatically different conditions, such as mixtures of human, yeast and fly samples[35], or comparing entirely different tissues that would be expected to have too high a level of differentially expressed genes for typical normalization procedures[36]. Instead, the benchmarker should

model their data off typical studies that use these tools, beginning by generating a list of relevant published experiments. Finally, additional relevant factors that must be considered are individual genomic variation and the diploid nature of mammalian genomes[37]. In some cases, benchmarking data may, however, justifiably benefit from deviating in targeted ways from real data (Step 4).

Dependence within samples

Omics studies are multivariate, with many values measured on the same biological sample. As mentioned above, benchmarkers must be vigilant about identifying potential biases before using benchmarking data with independent values within samples. Some differential expression methods[38,39], for example, make empirical Bayes assumptions that pool information across genes. It is therefore not appropriate to assess these methods without including dependence of gene expression within each subject. Even more subtly, the Benjamini–Hochberg false-discovery rate method, which is widely used to correct for multiple testing[40], requires that certain dependency assumptions hold[41]. Therefore, any tool evaluating false-discovery rate Q values is likely to be implicitly operating under this assumption. Benchmarkers must justify any decision to not capture dependencies known to exist in real data.

One reason for not including dependence is simply that benchmarkers may be unaware of how to include it. In addition to existing resources[42,43], we provide some guidance on this problem in the Supplementary Methods.

Evaluating benchmark data for realism

When not using real-world data, it is necessary to compare them to real data sets to assess their realism. For example, simulated single-cell[44] data with an unusually low or high number of zero counts may form a poor basis for comparing cell-type clustering methods. Therefore, a simulated single-cell transcriptomics data set should be checked to make sure zeros occur at a similar rate as in real data sets, as determined by standardized tooling where available. A non-exhaustive list of other single-cell RNA-seq features to asses includes: library sizes; normalization factors; Spearman correlation between cells and between genes; and the fraction of cells with zero reads, for each gene[45]. More generally, benchmarkers can check for the realistic distribution of parameters such as means, variances and the mean–variance relationship[46]. Benchmark papers should present factors like these plotted versus real data sets, typically by scatter plot or by graphing the distributions of both real and synthetic values. If the benchmark data diverge substantially from real data by any metrics, benchmarkers should document and justify why the discrepancy is unlikely to affect the conclusions. There are several tables in the report form (Supplemental File 1) intended to alert benchmarkers to such discrepancies and to document them.

Step 7: Execution

The use of a given tool invariably involves choices regarding configuration parameters, which may have default or recommended values. As most users use default configurations, evaluating tools with their default settings is important to establish results that reflect common use. However, this typically provides a very incomplete picture. If non-default configurations turn out to be optimal under certain situations, then benchmarking studies can inform not only which are the best tools but can also give guidance on optimal configurations[47]. The choice of which parameters to explore should be made judiciously, considering whether the parameter is likely to make a large impact, whether it is recommended to be adjusted by the tool's documentation, and by contacting the tool's authors for recommendations.

Informatics tools typically have many parameters, such that the parameter space can have many dimensions, and there may be a high cost to evaluating even one configuration of the parameters. If the parameter space is not too large, it could potentially be explored by a grid search, whereby each parameter is discretized sufficiently for all combinations of parameters to be evaluated. Otherwise, we recommend drawing inspiration from the machine learning community, which faces similar problems in hyper-parameter optimization. See ref. 48 for a comprehensive review, the relevant points of which are summarized below.

Bayesian optimization[49] is a family of approaches that efficiently build a model of performance as a function of the parameters and in some forms can be used for arbitrary parameter types (such as discrete and continuous data). Genetic algorithms[50] and particle swarm optimization[51] instead work by optimizing a population of parameter settings. To further cut down on resource use, multi-fidelity optimization methods can tailor the amount of runtime or data used to evaluate each parameter setting according to how likely it is predicted to perform well. In the simplest setting, this would entail running methods on stripped-down data sets to quickly assess more values before validating the best options on full-sized data sets. More advanced approaches, such as Hyperband[52], dynamically allocate resources to promising

Box 4

Case study of UHR data

The Stratagene Universal Human Reference (UHR) sample[139] is usually used to benchmark RNA-sequencing methods. This sample comprises a combination of ten human cancer cell lines and was created in 2004 for use in the reference channel for two-colour microarrays[140]. The rationale for combining ten cell lines was to produce a signal from as many genes as possible. As such, the UHR was never intended to represent a real sample for benchmarking; it is radically different from anything encountered in practice and tumorigenesis is well known to massively scramble transcriptomes, making cancer transcriptomics a special case that is difficult to extrapolate to any other scenario. A similar reference sample, consisting of a combination of ten brain tissues, is provided by ThermoFisher. Many publications have used these reference samples to benchmark differential expression by sequencing each sample several times to generate two groups of replicates[139,141–145]. However, this uses samples with highly artificial transcriptomes from radically different tissues, which does not reflect any real-world use case. Furthermore, in such a study design the only variation is technical and all genes are expressed independently from each other. Such data cannot substitute for data with realistic biological variation and inter-gene correlations, rendering any benchmark guidance derived from them unreliable. Yet, despite these issues, this strategy has been frequently used in benchmarking studies. The requirement to catalogue all sources of variation and dependence in real data and to document whether those dependencies were captured, or simplified, and why, would serve to reveal these shortcomings and allow them to be corrected before publication.

settings, using strategies inspired by reinforcement learning. However, these may be difficult to implement or limited by data availability in omics benchmarking.

Moreover, benchmarkers should consider the over-fitting problem: that their identified optimal configurations may not be optimal in other situations. In all cases, it is best practice to validate performance on data sets distinct from the ones used during parameter optimization. It is worth noting that parameter optimization is best performed on data sets that have low performance with default parameters, as these have the largest room for improvement. A subtle place that non-neutrality can sneak into is where benchmarkers have differing levels of experience with each tool, and so must be careful not to bias results towards tools for which they may be better at optimizing the parameters[1].

Box 5

Benchmarking data types

Benchmarking data can be categorized into four main types.

Real-world data
In some cases, tool performance can be evaluated on real-world data straight from a true representative experiment, in which the ground truth is determined by other 'gold standard' validation assays. Typically, the gold-standard assay will be slower and/or more expensive. For example, the PCR is often taken as a gold-standard assay for microarrays or RNA sequencing[146–148], and Sanger sequencing is accepted as the gold standard for Illumina sequencing[149,150].

Simulated data
Simulated data are generated de novo with a statistical model, with predetermined ground truth and configurable parameters. This typically involves making parametric assumptions about the nature of the data, but also provides the greatest level of control over all known aspects of the data. It is, however, easy to oversimplify when simulating data (such as independent normally distributed expression data). The time required to plan and implement a realistic simulation will usually be considerable, but in domains such as RNA sequencing, mature off-the-shelf simulators are available[75,151–153].

Semi-synthetic methods
Semi-synthetic data is generated using both real-world and simulated aspects. This includes a spectrum of data-generation methods with varying amounts of simulated and real content, including permutation or down-sampling of real data sets, the use of real data sets to determine parameters for simulation, and in silico spike-ins or spike-outs on top of real-world background data.

Spike-in or controlled experimental data
Spike-in refers to adding known quantities of biomolecules of interest into real samples.

These data types are generated using the experimental (bench) pipeline but with some aspects controlled to create known effects. Sometimes 'plasmode' is used to refer to methods that could fall under semi-synthetic methods or controlled experimental data[154].

Step 8: Evaluation

In the evaluation step, one compares the outputs of the tools to the ground truth. At this point, one should have already established the downstream goals identified in Step 1, if any, and whether to evaluate the direct outputs of the benchmarked pipeline steps or the downstream results of these steps, or both for completeness (Fig. 1b–d).

Evaluation should be of tool performance in terms of the needs and expectations of users, as opposed to more theoretical aspects such as model fit. A tool might be known to be a weak theoretical model of the data, but the benchmarker's job is first to assess whether that affects its performance. A simple, unrealistic model could outperform models with superior theoretical foundations, perhaps owing to overfitting or a lower power of the more complex model. Therefore, all evaluation criteria should be directly relevant to end users. We emphasize that it remains critical that the benchmark data are as realistic as possible (Step 6) to rigorously test all these aspects. Once deficiencies in performance have been identified, then it might be logical to further identify weaknesses in the model to prioritize continued development of the tool. However, this is not strictly necessary for a benchmarking study to be informative to the user; it is simply value added for the developer.

Evaluation metrics

The choice of evaluation metric depends upon the output of the tools being compared and is a critical and delicate decision. We discuss common situations below and in Table 1.

Assessing quantitative values. Quantitative outputs can be directly compared to known true values of the benchmark data. A popular metric for comparison is correlation (either Spearman or Pearson) between the estimated and the true values. Correlation, however, has well known limitations[53] and can be very misleading, so it is generally preferable to use a metric that incorporates an absolute measure of the error. Depending on the context, choices include the mean absolute deviation from the truth (MAD, defined as the mean of |estimated − true|) or a normalized form as the mean absolute relative deviation (MARD, defined as the mean of |estimated − true|/|estimated + true|). Scatter plots of estimated versus true values are also helpful, as are MA plots (also known as Bland–Altman plots)[54], which show the log ratio versus the mean of the estimated and true values. These can be used to compare the similarity of two estimates when no true values are known.

Assessing *P* values. Many tools perform statistical tests of null hypotheses and report *P* values. These are used to reject a set of non-null hypotheses with a controlled type I error rate. Therefore, any tool reporting *P* values should be assessed for whether it properly controls the type I error rate, in which case it is termed conservative or super-uniform[55,56]. It is critical to focus evaluation on values below and up to the *P* value cutoffs considered meaningful in practice (Supplementary Methods). Next, the tool should be assessed for the type II error rate by computing its power: the ability to reject the null when given non-null data. This can be computed from simulation at a particular effect size, but it can also be qualitatively assessed by running all tools on real-world data (even without known ground truth). If tools do not have conservative *P* values, then care must be taken in assessing power[34], and we recommend computing power at a type 1 error rate using the observed true error rate rather than the reported (inaccurate) rate. This provides a fair power comparison for methods regardless of their success in controlling *P* values. We detail how to assess *P* value

distributions for conservativeness and power in the Supplementary Methods and suggest some existing libraries[57]. Benchmarkers should also consider alternatives to directly evaluating *P* values, such as receiver operating curves[58], which use only the ordering of *P* values without their exact values.

Although evaluating the uniformity of *P* value distributions has uses outside benchmarking, such as diagnosing model fit or estimating the total number of non-null tests[59], we recommend against comparing methods by this criterion during benchmarking. Evaluating *P* value distributions is valid for considerations of conservativeness and power but non-uniformity under the null in and of itself does not necessarily render a method invalid. Non-uniformity can arise in many situations that are not necessarily problematic, including corrections for multiple hypotheses[60], composite null hypotheses (which includes one-sided tests)[61], and when analysing discrete data. Therefore, judging tools by (non-)uniformity under the null could incorrectly penalize ones that have applied valid procedures such as these. Instead, evaluating by power and conservativeness ensures that any problematic deviations from uniformity are identified without unjustified penalties.

False-discovery-rate *Q* values are never expected to follow a uniform distribution and should not be assessed as if they were *P* values. Instead, compare the reported *Q* value to the actual fraction of positive results that were false positives (Supplementary Methods).

Assessing genomic regions. Many omics tools identify regions of the genome, such as peak callers for ChIP–seq or assay for transposase-accessible chromatin with sequencing (ATAC-seq) or de novo annotation tools for transcripts. The simplest metrics check overlap with true regions (such as complete containment of one true region in the inferred region, or vice versa, or having at least one base in common). Alternatively, exact matches can be required. The appropriate measure depends on the downstream goals. For ChIP–seq, it is unrealistic to expect exact matches of peaks because the true regions are insufficiently well defined. A less strict metric requiring overlap of at least one base would be more appropriate. In contrast, annotation of transcribed regions demands higher resolution because metrics need to assess the exact accuracy of splice sites. There exist tools to aid in classifying annotation accuracy[62].

Benchmarking from real experimental data

Although simulated or controlled experiments with a known truth are the most straightforward way to assess the performance of a tool, there is substantial room to directly use real experimental data that lacks a known truth. Instead, one can search for surrogates of accuracy. Consider ChIP–seq peak calling; benchmarks have compared called peaks to a wide range of alternative sources of information[31]. These include manual inspection[63], enrichment for known binding motifs[64], expressed genes from RNA-seq (for H3K36me3 peaks)[65], promoter regions of expressed genes (for H3K4me3 peaks)[65], genome-wide association study single-nucleotide polymorphisms of an associated phenotype[66], and quantitative PCR validation[67–70].

Another approach is to find the ground truth at another level of data analysis. For example, when benchmarking single-cell assays, it is often impossible to obtain ground-truth values on the same set of cells assayed owing to the destructive nature of the experiments. However, it may be possible to obtain ground-truth values on a similar population of cells. Although that does not provide cell-level ground-truth values, instead properties of the distribution can be assayed for correctness[71]. Moreover, benchmarks may check for conservation of results across species[72–74], which is often indicative that a tool has identified biologically relevant results.

Last, most tools can be assessed for consistency across replicated experiments. However, tools may produce consistent false positives. Nonetheless, consistency can be one aspect of a comprehensive benchmark analysis.

Table 1 | Evaluation metrics

Value type	Examples	Evaluation metrics: comparing to truth
Continuous	Abundance, expression	Mean absolute deviation; correlation (Spearman, Pearson)
Binary	Null hypothesis rejection	AUROC[112]; true positives; false positives; *F*-metric; Matthews correlation coefficient[113]
Categorical	Diploid genotyping	Confusion matrices; Cramer's V[114]
Genomic region	Peak calling, annotation	Number of bases overlap; any overlap at all; exact matches of end point
Sequence or motif	Motif calling	Shannon entropy; Hamming distance; Levenshtein distance, BLAST score
P value	Differential expression, GWAS	Conservative on nulls; power on non-nulls
Clusters	Cell-type clustering	Rand index[115]; mutual information; V-measure; Fowlkes–Mallows[116]
Ranked lists	ChIP–seq peak list	Irreproducible discovery rate[69]; sequential rank agreement[117]

AUROC, area under the receiver operating characteristic; BLAST, basic local alignment search tool; ChIP–seq, chromatin immunoprecipitation followed by sequencing; GWAS, genome-wide association study.

Assessing specific stratification parameters

Considering the stratification parameters identified in Step 5, benchmarkers should then stratify results by the values of these parameters. The goal here is to determine whether subcases or specific situations have substantially different performance. In such cases, distinct recommendations can be formulated for users of different situations and the failure modes of tools can be identified.

Step 9: Interpretation

Benchmarkers would like to provide universal guidelines based on benchmarking observations; however, in omics almost every use case is a special case, and it is not feasible for one benchmark to assess all the ways the tools in question will be applied. Although it is tempting to overgeneralize results, conclusions must reflect the limitations in scope from Step 1, the realities of the benchmarking data in terms of factors such as species and study designs, and the evaluation criteria. Readers should be assumed to be sophisticated users who can evaluate nuances and make special considerations for how the results apply to their case, at least to a reasonable level. For example, it is common to evaluate methods only at specific *Q* value thresholds, often limited to 0.05 or 0.01. However, in contrast to *P* values, *Q* value cutoffs used

in practice are often higher than 0.05, so the benchmark may have little relevance to how the tool is used in practice. Requiring authors to document and justify such decisions enables readers to efficiently identify such relevant limitations.

Benchmarkers should be prepared to conclude that all tools perform approximately equally well, that no tool performs adequately at the measured task, or that the benchmarking results give no clear guidance. For example, it had arguably been the case for RNA-seq isoform level structure inference that none of the available methods were viable, at least through 2015 (ref. 75). In a crowded field, identifying a singular winner may be less important than identifying the clear underperformers. Moreover, benchmark studies have uses beyond providing actionable recommendations, such as aiding tool developers by identifying weaknesses and providing future directions for the field.

Perhaps the most challenging aspect of benchmarking arises when there are conflicting results, such as when two critical evaluation metrics rank tools in different orderings or when one tool outperforms in specificity but underperforms in sensitivity. Again, this calls for presenting all the results so that readers can consider all the factors for their specific needs (reviewed in more depth elsewhere[1]).

Step 10: Maintenance

Although the sharing of code and input data is increasingly common, a recent review found that benchmarks were lacking in extensibility and interoperability[76]. Benchmarking studies often become outdated in rapidly evolving fields. Therefore, benchmarking is never done, but instead requires ongoing efforts and, crucially, necessitates regular updates from the benchmarking community as new methods or data become available. Turning benchmarking into regularly scheduled competitions has driven methods development via projects such as CASP[77]. This reinforces the impact that all benchmarking efforts can have if they are made fully reusable and kept up to date.

It is critical that benchmarkers facilitate reuse by creating reproducible and maintainable benchmark studies. The use of Docker[78,79] or similar containerization technology is often recommended for this purpose; however, it remains easy to fail to achieve a reproducible or maintainable benchmark analysis even with containerization. We provide some advice regarding containerization specific to benchmarking in Supplementary Methods Note 3.

As links to private websites can go offline[80,81], shared data and code is best done through public data repositories. However, many public data repositories such as the Gene Expression Omnibus[82,83] accept only experimental, rather than simulated, data. Instead, generic repositories, which include Zenodo or figshare, can be used.

Benchmarking frameworks

There is growing appreciation that benchmarking needs centralized platforms to enable coordinated and continuous updates of benchmarking studies. Several groups have taken up this challenge, and such platforms are being adopted[84–87], although their use remains quite limited[76]. We review here the state of these systems and other benchmarking software (Table 2). The most ambitious frameworks provide centralized infrastructures for running benchmarks with continuously updated results and allow submission of new or updated tools or data sets. Notable examples of these are OpenEBench[16,17] and Omnibenchmark[15]. Although such systems bring benchmarks closer to the ideal of a continuously updated, reproducible benchmark, we speculate that these platforms have not gained more traction with benchmarkers because the benefits do not go to benchmarkers themselves. Instead, the benefits they provide are primarily for the broader community, but it is the benchmarker who decides whether to use these platforms. The adoption of practices that provide benefits only to others is not unprecedented, however; think of the sharing of data and code, at least when incentivized by community expectations and journal requirements. Another important area for future development is the organization and presentation of benchmarking results on these platforms, given that continuously updated benchmarking efforts cannot depend upon traditional, static publications to make the results available. These platforms must make sure to highlight not only performance but also the important context documenting the

Table 2 | Benchmarking frameworks and libraries

Framework	Scope	Features
OpenEBench[16,17]	General	Continuous updates, containerization, platform, community contributions
Omnibenchmark[15]	General	Continuous updates, containerization, provenance tracking, platform, community contributions
SummarizedBenchmark[12]	General	Pipelining, organization and structure
pipeComp[13]	General	Pipelining, organization and structure
DSC[14]	General	Pipelining, organization and structure
dynverse[94]	scRNA-seq trajectory inference	Data sets, evaluation, visualization, web app interface
iCOBRA[57]	Feature ranking, binary classification	Data sets, evaluation, visualization, web app interface
compcodeR[92]	RNA-seq differential expression	Simulation, evaluation, visualization
BDTcomparator[118]	Binary classification	Evaluation
rnaseqcomp[93]	RNA-seq quantification	Data sets, evaluation, community contributions
LEMMI[89]	Metagenomics	Data sets, evaluation, pipeline, continuous, community contributions, containerization
LEMORTHO[90]	Orthologue delineation	Data sets, evaluation, pipeline, continuous, community contributions, containerizations
IBRAP[119]	scRNA-seq	Data sets, evaluation, web app interface
scRNA-IBT[120]	scRNA-seq imputation	Containerization, pipeline, plugins
Spotless[121]	Cell-type deconvolution	Pipeline, containerization, simulations
RNAontheBENCH[26]	RNA-seq	Data sets, evaluation
SQANTI[62]	Transcript annotation	Evaluation

RNA-seq, RNA sequencing; scRNA-seq, single-cell RNA-sequencing.

why and how of the benchmark (which can be achieved by including the report template provided in Supplementary File 1) so that readers can make informed decisions.

Another area in which benchmarks could improve is to have more systematic reuse of components of benchmarking and better organized benchmark software[76]. There are software frameworks[12,13] that aim to provide organization and standardization for general benchmarking without providing infrastructure to run the benchmarks. There are also many libraries and frameworks covering more specific benchmarking tasks, typically benchmarking of specific topics[58,88–94]. One example is dynverse, for benchmarking single-cell trajectory inference. This provides an extensive collection of data sets and evaluation metrics for this domain and has seen repeated use as a means of benchmarking new methods[95–98]. Another success story is iCOBRA[57], which performs evaluations of methods that produce *P* values or other scores on a list of features and has been widely used[99–105]. By restricting their scope, these projects can address some of the most challenging aspects of benchmarking: generating meaningful data sets and evaluations. This, paradoxically, may mean that their limited scope has driven adoption by providing more direct benefits to the benchmarkers relative to more general frameworks.

Conclusions and future perspectives

High-quality benchmarking is critically important for accurate data analysis but produces several challenges. For example, simulating a data type for a particular analysis requires a deep understanding of the data type and nimble modelling skills[29]. To achieve meaningful benchmarking of RNA-seq differential expression analysis through simulation, for example, one must simulate a population of transcriptomes, ideally from a population of diploid genomes across multiple subpopulations. Moreover, the various effects and properties of the population parameters must be adjustable to enable their impact on the analysis to be assessed. These are increasingly challenging tasks in statistical and computational modelling as ever more complex omics methods continue to be developed. For example, there are now multi-omics benchmarks[106–111] that have had to combine the difficulties of multiple data types (Box 3). Therefore, there is a strong and increasing need for more support and more groups dedicated to methods evaluation, who are not themselves methods developers and who do not have other conflicts of interest that might influence their results.

A cultural shift is also required to support the ongoing maintenance and repetition of benchmarking studies. Long-term benchmarking efforts require deceptively large investments of time and resources to sustain and are often undervalued because they are perceived to lack 'novelty', rendering these essential tasks less appealing to researchers. As a solution, the community could promote this critical maintenance work: journals could accept short-form notes and letters reporting regular benchmarking updates, and funding agencies and hiring committees could recognize ongoing maintenance efforts with funding mechanisms for the maintenance and updating of benchmarks. Furthermore, a new generation of benchmarking platforms, such as OpenEBench[16,17] and Omnibench[15], are working to lower the costs of such maintenance.

An overarching theme in omics benchmarking is that it is not possible to formalize guidance that applies generally. Every study is a special case whose analysis requires a deep understanding of a given data type at multiple levels. As such, one should proceed with caution when conducting a comprehensive benchmarking study, given that it could influence the choices of methodology for a whole research community.

The field of benchmarking in modern biology is not in its infancy. We have reviewed the literature to identify important lessons and critical areas for improvement moving forward. There is an ongoing need to continually raise the bar in benchmarking and to improve the value and interpretability of our studies. Foremost among our recommendations is to establish standards that include the comprehensive documentation of informed decisions and judgement calls. We have developed what should be taken as a first attempt at such reporting standards, which includes (in eleven sections) a comprehensive list of potentially relevant tools with notes for each one on why it was or was not included, and comprehensive lists of all sources of dependence in real data and notes for each one on how it was accounted for, or a justification for why and how it was simplified. These lists will reveal avoidable shortcomings in benchmarking studies so that these can be corrected prior to publication and will also help readers to understand the applicability of the study. This report format could be refined into specific and comprehensive reports for particular benchmarking domains. For example, a differential expression benchmarking report would include tables of known properties of real-world data, such as dependence between genes across biological replicates. Benchmarkers can then simply indicate whether the factors were captured in the model without having to determine them anew each time, thus mitigating author biases or oversights. The establishment of these standards would follow a period of community feedback. The introduction of such systematic reporting in benchmarking could have a profound effect on the effective translation of benchmarking studies into everyday practice. The inclusion of such reports could ultimately become a requirement enforced by journals.

Published online: 12 January 2024

References

1. Weber, L. M. et al. Essential guidelines for computational method benchmarking. *Genome Biol.* **20**, 125 (2019).
   This landmark paper describes the fundamental tenets of omics benchmarking in biology, for those intending to perform benchmarking studies or to study the literature in search of guidance.
2. Mangul, S. et al. Systematic benchmarking of omics computational tools. *Nat. Commun.* **10**, 1393 (2019).
   This landmark paper describes the fundamentals of benchmarking, with a focus on the big picture rather than the particulars of data generation.
3. Aniba, M. R., Poch, O. & Thompson, J. D. Issues in bioinformatics benchmarking: the case study of multiple sequence alignment. *Nucleic Acids Res.* **38**, 7353–7363 (2010).
4. Mathews, D. H. How to benchmark RNA secondary structure prediction accuracy. *Methods* **162–163**, 60–67 (2019).
5. Bokulich, N. A., Ziemski, M., Robeson, M. S. & Kaehler, B. D. Measuring the microbiome: best practices for developing and benchmarking microbiomics methods. *Comput. Struct. Biotechnol. J.* **18**, 4048–4062 (2020).
6. Meyer, F. et al. Tutorial: assessing metagenomics software with the CAMI benchmarking toolkit. *Nat. Protoc.* **16**, 1785–1801 (2021).
7. Olson, N. D. et al. Variant calling and benchmarking in an era of complete human genome sequences. *Nat. Rev. Genet.* **24**, 464–483 (2023).
8. Crowell, H. L., Morillo Leonardo, S. X., Soneson, C. & Robinson, M. D. The shaky foundations of simulating single-cell RNA sequencing data. *Genome Biol.* **24**, 62 (2023).
9. Escalona, M., Rocha, S. & Posada, D. A comparison of tools for the simulation of genomic next-generation sequencing data. *Nat. Rev. Genet.* **17**, 459–469 (2016).
10. Milhaven, M. & Pfeifer, S. P. Performance evaluation of six popular short-read simulators. *Heredity* **130**, 55–63 (2023).
11. Shakola, F., Palejev, D. & Ivanov, I. A framework for comparison and assessment of synthetic RNA-seq data. *Genes* **13**, 2362 (2022).
12. Kimes, P. K. & Reyes, A. Reproducible and replicable comparisons using SummarizedBenchmark. *Bioinformatics* **35**, 137–139 (2018).
13. Germain, P.-L., Sonrel, A. & Robinson, M. D. pipeComp, a general framework for the evaluation of computational pipelines, reveals performant single cell RNA-seq preprocessing tools. *Genome Biol.* **21**, 227 (2020).
14. Stephens, M. DSC: dynamic statistical comparisons. *GitHub* https://stephenslab.github.io/dsc-wiki/overview.html (2023).
15. Robinson, M. Omnibenchmark: open and continuous community benchmarking. *Omnibenchmark* https://omnibenchmark.org (2023).

16. Capella-Gutierrez, S. et al. Lessons learned: recommendations for establishing critical periodic scientific benchmarking. Preprint at *bioRxiv* https://doi.org/10.1101/181677 (2017).
17. de Pico, E. M., Gelpi, J. L. & Capella-Gutiérrez, S. FAIRsoft — a practical implementation of FAIR principles for research software. Preprint at *bioRxiv* https://doi.org/10.1101/2022.05.04.490563 (2022).
18. Nakato, R. & Sakata, T. Methods for ChIP-seq analysis: a practical workflow and advanced applications. *Methods* **187**, 44–53 (2021).
19. Li, Y., Ge, X., Peng, F., Li, W. & Li, J. J. Exaggerated false positives by popular differential expression methods when analyzing human population samples. *Genome Biol.* **23**, 79 (2022).
20. Ison, J. et al. Tools and data services registry: a community effort to document bioinformatics resources. *Nucleic Acids Res.* **44**, D38–D47 (2015).
21. Wikipedia. List of bioinformatics software. *Wikipedia* https://en.wikipedia.org/wiki/List_of_bioinformatics_software (2022).
22. Zappia, L. & Theis, F. J. Over 1000 tools reveal trends in the single-cell RNA-seq analysis landscape. *Genome Biol.* **22**, 301 (2021).
23. Koch, F. C., Sutton, G. J., Voineagu, I. & Vafaee, F. Supervised application of internal validation measures to benchmark dimensionality reduction methods in scRNA-seq data. *Brief. Bioinform.* **22**, bbab304 (2021).
24. Norel, R., Rice, J. J. & Stolovitzky, G. The self-assessment trap: can we all be better than average? *Mol. Syst. Biol.* **7**, 537 (2011).
This paper reviews the reported performances of new methods and calls for increased use of multiple evaluation metrics and publication of novel methods even when they do not improve performance above prior works.
25. Buchka, S., Hapfelmeier, A., Gardner, P. P., Wilson, R. & Boulesteix, A. L. On the optimistic performance evaluation of newly introduced bioinformatic methods. *Genome Biol.* **22**, 152 (2021).
This review compares the initial performance claims of published methods to later benchmarking of the same methods, highlighting the need for independent benchmarking.
26. Germain, P.-L. et al. RNAontheBENCH: computational and empirical resources for benchmarking RNAseq quantification and differential expression methods. *Nucleic Acids Res.* **44**, 5054–5067 (2016).
27. Holik, A. Z. et al. RNA-seq mixology: designing realistic control experiments to compare protocols and analysis methods. *Nucleic Acids Res.* **45**, e30 (2017).
This paper demonstrates the importance of including both technical and biological variation in benchmark data, as well as one approach for including realistic biological variation when evaluating RNA-seq.
28. Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. Basic local alignment search tool. *J. Mol. Biol.* **215**, 403–410 (1990).
29. Sandve, G. K. & Greiff, V. Access to ground truth at unconstrained size makes simulated data as indispensable as experimental data for bioinformatics methods development and benchmarking. *Bioinformatics* **38**, 4994–4996 (2022).
This paper argues for nearly always including simulated data in methods evaluation in order to go beyond the limitations of experimental data with regard to factors such as sample size, knowledge of ground truth and explicit presentation of assumptions.
30. Maza, E., Frasse, P., Senin, P., Bouzayen, M. & Zouine, M. Comparison of normalization methods for differential gene expression analysis in RNA-seq experiments: a matter of relative size of studied transcriptomes. *Commun. Integr. Biol.* **6**, e25849 (2013).
31. Szalkowski, A. M. & Schmid, C. D. Rapid innovation in ChIP-seq peak-calling algorithms is outdistancing benchmarking efforts. *Brief. Bioinform.* **12**, 626–633 (2011).
32. Jelizarow, M., Guillemot, V., Tenenhaus, A., Strimmer, K. & Boulesteix, A. L. Over-optimism in bioinformatics: an illustration. *Bioinformatics* **26**, 1990–1998 (2010).
This paper emphasizes the importance of evaluating methods on 'fresh' validation data sets that were not used for tuning the method under evaluation.
33. Szikszai, M., Wise, M., Datta, A., Ward, M. & Mathews, D. H. Deep learning models for RNA secondary structure prediction (probably) do not generalize across families. *Bioinformatics* **38**, 3892–3899 (2022).
34. Mehta, T., Tanik, M. & Allison, D. B. Towards sound epistemological foundations of statistical methods for high-dimensional biology. *Nat. Genet.* **36**, 943–947 (2004).
35. Lin, M. H. et al. Benchmarking differential expression, imputation and quantification methods for proteomics data. *Brief. Bioinform.* **23**, bbac138 (2022).
36. Robinson, M. D. & Oshlack, A. A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biol.* **11**, R25 (2010).
37. Lahens, N. F. et al. CAMPAREE: a robust and configurable RNA expression simulator. *BMC Genomics* **22**, 692 (2021).
38. Love, M. I., Huber, W. & Anders, S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.* **15**, 550 (2014).
39. Law, C. W., Chen, Y., Shi, W. & Smyth, G. K. voom: precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biol.* **15**, R29 (2014).
40. Korthauer, K. et al. A practical guide to methods controlling false discoveries in computational biology. *Genome Biol.* **20**, 118 (2019).
41. Benjamini, Y. & Yekutieli, D. The control of the false discovery rate in multiple testing under dependency. *Ann. Stat.* **29**, 1165–1188 (2001).
42. Burton, A., Altman, D. G., Royston, P. & Holder, R. L. The design of simulation studies in medical statistics. *Stat. Med.* **25**, 4279–4292 (2006).
43. Madsen, L. & Birkes, D. Simulating dependent discrete data. *J. Stat. Comput. Simul.* **83**, 677–691 (2013).
44. Soneson, C. & Robinson, M. D. Towards unified quality verification of synthetic count data with countsimQC. *Bioinformatics* **34**, 691–692 (2017).
45. Cao, Y., Yang, P. & Yang, J. Y. H. A benchmark study of simulation methods for single-cell RNA sequencing data. *Nat. Commun.* **12**, 6911 (2021).
A benchmark of 12 single-cell RNA-seq simulation methods, including an exhaustive evaluation of simulation quality by comparison to real data sets.
46. Warton, D. I. & Hui, F. K. C. The central role of mean–variance relationships in the analysis of multivariate abundance data: a response to Roberts (2017). *Methods Ecol. Evol.* **8**, 1408–1414 (2017).
47. Baruzzo, G. et al. Simulation-based comprehensive benchmarking of RNA-seq aligners. *Nat. Methods* **14**, 135–139 (2017).
48. Yang, L. & Shami, A. On hyperparameter optimization of machine learning algorithms: theory and practice. *Neurocomputing* **415**, 295–316 (2020).
This paper reviews the common techniques used for parameter optimization in machine learning, some of which can be used in omics benchmarking for optimizing parameters of the assessed tools.
49. Bischl, B. et al. Hyperparameter optimization: foundations, algorithms, best practices, and open challenges. *WIREs Data Min. Knowl. Discov.* **13**, e1484 (2023).
50. Lessmann, S., Stahlbock, R. & Crone, S. F. in *Proc. Int. Conf. Artificial Intelligence* 74–82 (ICAI, 2005).
51. Lorenzo, P. R., Nalepa, J., Kawulok, M., Ramos, L. S. & Pastor, J. R. in *Proc. Genetic Evolutionary Computation Conf.* 481–488 (ACM, 2017).
52. Eggensperger, K., Hutter, F., Hoos, H. & Leyton-Brown, K. in *Proc. AAAI Conf. Artificial Intelligence* (AAAI, 2015).
53. Anscombe, F. J. Graphs in statistical analysis. *Am. Stat.* **27**, 17–21 (1973).
This classic paper shows, with a now well-known example, the shortcomings of summary statistics such as mean and correlation.
54. Bland, J. M. & Altman, D. G. Statistical methods for assessing agreement between two methods of clinical measurement. *Lancet* **1**, 307–310 (1986).
55. Chen, X. & Sarkar, S. K. On Benjamini–Hochberg procedure applied to mid *p*-values. *J. Stat. Plan. Infer.* **205**, 34–45 (2020).
56. Lyu, P., Li, Y., Wen, X. & Cao, H. JUMP: replicability analysis of high-throughput experiments with applications to spatial transcriptomic studies. *Bioinformatics* **39**, btad366 (2023).
57. Soneson, C. & Robinson, M. D. iCOBRA: open, reproducible, standardized and live method benchmarking. *Nat. Methods* **13**, 283 (2016).
A widely useful library for benchmarking that performs comparisons of methods that produce ranked lists of features, particularly *P* values but also numerical rankings.
58. Sing, T., Sander, O., Beerenwinkel, N. & Lengauer, T. ROCR: visualizing classifier performance in R. *Bioinformatics* **21**, 3940–3941 (2005).
59. Breheny, P., Stromberg, A. & Lambert, J. *p*-value histograms: inference and diagnostics. *High Throughput* **7**, 23 (2018).
60. VanderWeele, T. J. & Mathur, M. B. Some desirable properties of the Bonferroni correction: is the Bonferroni correction really so bad. *Am. J. Epidemiol.* **188**, 617–618 (2019).
61. Bayarri, M. J. & Berger, J. O. *P* values for composite null models. *J. Am. Stat. Assoc.* **95**, 1127–1142 (2000).
62. Tardaguila, M. et al. SQANTI: extensive characterization of long-read transcript sequences for quality control in full-length transcriptome identification and quantification. *Genome Res.* **28**, 396–411 (2018).
63. Rye, M. B., Sætrom, P. & Drabløs, F. A manually curated ChIP-seq benchmark demonstrates room for improvement in current peak-finder programs. *Nucleic Acids Res.* **39**, e25 (2010).
64. Wilbanks, E. G. & Facciotti, M. T. Evaluation of algorithm performance in ChIP-seq peak detection. *PLoS ONE* **5**, e11471 (2010).
65. Thomas, R., Thomas, S., Holloway, A. K. & Pollard, K. S. Features that define the best ChIP-seq peak calling algorithms. *Brief. Bioinform.* **18**, 441–450 (2016).
66. de Boer, B. A. et al. OccuPeak: ChIP-seq peak calling based on internal background modelling. *PLoS ONE* **9**, e99844 (2014).
67. Johnson, D. S., Mortazavi, A., Myers, R. M. & Wold, B. Genome-wide mapping of in vivo protein–DNA interactions. *Science* **316**, 1497–1502 (2007).
68. Laajala, T. D. et al. A practical comparison of methods for detecting transcription factor binding sites in ChIP-seq experiments. *BMC Genomics* **10**, 618 (2009).
69. Li, Q., Brown, J. B., Huang, H. & Bickel, P. J. Measuring reproducibility of high-throughput experiments. *Ann. Appl. Stat.* **5**, 1752–1779 (2011).
This paper defines the widely used irreproducible discovery rate, which measures the consistency of rankings of features to evaluate consistency across independent biological samples.
70. Nakato, R. & Shirahige, K. Recent advances in ChIP-seq analysis: from quality management to whole-genome annotation. *Brief. Bioinform.* **18**, 279–290 (2016).
71. Huang, M. et al. SAVER: gene expression recovery for single-cell RNA sequencing. *Nat. Methods* **15**, 539–542 (2018).
72. Laloum, D. & Robinson-Rechavi, M. Methods detecting rhythmic gene expression are biologically relevant only for strong signal. *PLoS Comput. Biol.* **16**, e1007666 (2020).
73. Kryuchkova-Mostacci, N. & Robinson-Rechavi, M. A benchmark of gene expression tissue-specificity metrics. *Brief. Bioinform.* **18**, 205–214 (2016).
74. Manni, M., Berkeley, M. R., Seppey, M. & Zdobnov, E. M. BUSCO: assessing genomic data quality and beyond. *Curr. Protoc.* **1**, e323 (2021).

75. Hayer, K. E., Pizarro, A., Lahens, N. F., Hogenesch, J. B. & Grant, G. R. Benchmark analysis of algorithms for determining and quantifying full-length mRNA splice forms from RNA-seq data. *Bioinformatics* **31**, 3938–3945 (2015).
76. Sonrel, A. et al. Meta-analysis of (single-cell method) benchmarks reveals the need for extensibility and interoperability. *Genome Biol.* **24**, 119 (2023).
This paper extensively reviews recent single-cell analysis method benchmarking papers and quantifies the need for documented, reproducible and extensible benchmarking.
77. Kryshtafovych, A., Schwede, T., Topf, M., Fidelis, K. & Moult, J. Critical assessment of methods of protein structure prediction (CASP) — round XIV. *Proteins* **89**, 1607–1617 (2021).
An important example of competition-style benchmarking, in which regularly scheduled independent, blind assessment of protein structure prediction methods is performed using novel, experimentally determined proteins as reference.
78. Merkel, D. Docker: lightweight Linux containers for consistent development and deployment. *Linux J.* **2014**, 2 (2014).
79. Kadri, S., Sboner, A., Sigaras, A. & Roy, S. Containers in bioinformatics: applications, practical considerations, and best practices in molecular pathology. *J. Mol. Diagn.* **24**, 442–454 (2022).
80. Audoux, J. et al. SimBA: a methodology and tools for evaluating the performance of RNA-seq bioinformatic pipelines. *BMC Bioinformatics* **18**, 428 (2017).
81. Bansal, S. & Parmar, S. Decay of URLs citation: a case study of current science. *Libr. Philos. Pract.* https://digitalcommons.unl.edu/libphilprac/3582 (2020).
82. Edgar, R., Domrachev, M. & Lash, A. E. Gene expression omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res.* **30**, 207–210 (2002).
83. Barrett, T. et al. NCBI GEO: archive for functional genomics data sets–update. *Nucleic Acids Res.* **41**, D991–D995 (2013).
84. Altenhoff, A. M. et al. The Quest for Orthologs benchmark service and consensus calls in 2020. *Nucleic Acids Res.* **48**, W538–W545 (2020).
85. Conte, A. D. et al. Critical assessment of protein intrinsic disorder prediction (CAID) — results of round 2. *Proteins* **91**, 1925–1934 (2023).
86. Bryce-Smith, S. et al. Extensible benchmarking of methods that identify and quantify polyadenylation sites from RNA-seq data. *RNA* **29**, 1839–1855 (2023).
87. Nevers, Y. et al. The Quest for Orthologs orthology benchmark service in 2022. *Nucleic Acids Res.* **50**, W623–W632 (2022).
88. Pratapa, A., Jalihal, A. P., Law, J. N., Bharadwaj, A. & Murali, T. M. Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nat. Methods* **17**, 147–154 (2020).
89. Seppey, M., Manni, M. & Zdobnov, E. M. LEMMI: a continuous benchmarking platform for metagenomics classifiers. *Genome Res.* **30**, 1208–1216 (2020).
90. Kuznetsov, D. et al. OrthoDB v11: annotation of orthologs in the widest sampling of organismal diversity. *Nucleic Acids Res.* **51**, D445–D451 (2022).
91. Perscheid, C. Comprior: facilitating the implementation and automated benchmarking of prior knowledge-based feature selection approaches on gene expression data sets. *BMC Bioinformatics* **22**, 401 (2021).
92. Soneson, C. compcodeR — an R package for benchmarking differential expression methods for RNA-seq data. *Bioinformatics* **30**, 2517–2518 (2014).
93. Teng, M. et al. A benchmark for RNA-seq quantification pipelines. *Genome Biol.* **17**, 74 (2016).
94. Saelens, W., Cannoodt, R., Todorov, H. & Saeys, Y. A comparison of single-cell trajectory inference methods. *Nat. Biotechnol.* **37**, 547–554 (2019).
95. Smolander, J., Junttila, S. & Elo, L. L. Cell-connectivity-guided trajectory inference from single-cell data. *Bioinformatics* **39**, btad515 (2023).
96. Wang, C. X., Zhang, L. & Wang, B. One cell at a time (OCAT): a unified framework to integrate and analyze single-cell RNA-seq data. *Genome Biol.* **23**, 102 (2022).
97. Van den Berge, K. et al. Trajectory-based differential expression analysis for single-cell sequencing data. *Nat. Commun.* **11**, 1201 (2020).
98. Li, R. & Quon, G. scBFA: modeling detection patterns to mitigate technical noise in large-scale single-cell genomics data. *Genome Biol.* **20**, 193 (2019).
99. Zhu, A., Ibrahim, J. G. & Love, M. I. Heavy-tailed prior distributions for sequence count data: removing the noise and preserving large differences. *Bioinformatics* **35**, 2084–2092 (2018).
100. Choudhary, S. & Satija, R. Comparison and evaluation of statistical error models for scRNA-seq. *Genome Biol.* **23**, 27 (2022).
101. Spies, D., Renz, P. F., Beyer, T. A. & Ciaudo, C. Comparative analysis of differential gene expression tools for RNA sequencing time course data. *Brief. Bioinform.* **20**, 288–298 (2017).
102. Zhu, A., Srivastava, A., Ibrahim, J. G., Patro, R. & Love, M. I. Nonparametric expression analysis using inferential replicate counts. *Nucleic Acids Res.* **47**, e105 (2019).
103. Gilis, J., Vitting-Seerup, K., Van den Berge, K. & Clement, L. satuRn: scalable analysis of differential transcript usage for bulk and single-cell RNA-sequencing applications. *F1000Res* **10**, 374 (2021).
104. Wu, E. Y. et al. SEESAW: detecting isoform-level allelic imbalance accounting for inferential uncertainty. *Genome Biol.* **24**, 165 (2023).
105. He, Z., Pan, Y., Shao, F. & Wang, H. Identifying differentially expressed genes of zero inflated single cell RNA sequencing data using mixed model score tests. *Front. Genet.* **12**, 616686 (2021).
106. Li, Y., Mansmann, U., Du, S. & Hornung, R. Benchmark study of feature selection strategies for multi-omics data. *BMC Bioinformatics* **23**, 412 (2022).
107. Herrmann, M., Probst, P., Hornung, R., Jurinovic, V. & Boulesteix, A.-L. Large-scale benchmark study of survival prediction methods using multi-omics data. *Brief. Bioinform.* **22**, bbaa167 (2020).
108. Leng, D. et al. A benchmark study of deep learning-based multi-omics data fusion methods for cancer. *Genome Biol.* **23**, 171 (2022).
109. Cantini, L. et al. Benchmarking joint multi-omics dimensionality reduction approaches for the study of cancer. *Nat. Commun.* **12**, 124 (2021).
110. Pierre-Jean, M., Deleuze, J.-F., Le Floch, E. & Mauger, F. Clustering and variable selection evaluation of 13 unsupervised methods for multi-omics data integration. *Brief. Bioinform.* **21**, 2011–2030 (2020).
111. Rappoport, N. & Shamir, R. Multi-omic and multi-view clustering algorithms: review and cancer benchmark. *Nucleic Acids Res.* **46**, 10546–10562 (2018).
112. Fawcett, T. An introduction to ROC analysis. *Pattern Recogn. Lett.* **27**, 861–874 (2006).
This paper provides advice on applying the widely used receiver operating characteristic (ROC) curve, including pitfalls in interpretation when using the ROC to compare method performance.
113. Matthews, B. W. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim. Biophys. Acta* **405**, 442–451 (1975).
114. Cramér, H. *Mathematical Methods of Statistics* 282 (Princeton Univ. Press, 1946).
115. Rand, W. M. Objective criteria for the evaluation of clustering methods. *J. Am. Stat. Assoc.* **66**, 846–850 (1971).
116. Fowlkes, E. B. & Mallows, C. L. A method for comparing two hierarchical clusterings. *J. Am. Stat. Assoc.* **78**, 553–569 (1983).
117. Ekstrom, C. T., Gerds, T. A. & Jensen, A. K. Sequential rank agreement methods for comparison of ranked lists. *Biostatistics* **20**, 582–598 (2019).
118. Fijorek, K., Fijorek, D., Wisniowska, B. & Polak, S. BDTcomparator: a program for comparing binary classifiers. *Bioinformatics* **27**, 3439–3440 (2011).
119. Knight, C. H. et al. IBRAP: integrated benchmarking single-cell RNA-sequencing analytical pipeline. *Brief. Bioinform.* **24**, bbad061 (2023).
120. Tantasatityanon, P. & Wichadakul, D. in *Proc. 15th Int. Conf. Computer Modeling Simulation* 84–91 (ACM, 2023).
121. Sang-aram, C., Browaeys, R., Seurinck, R. & Saeys, Y. Spotless: a reproducible pipeline for benchmarking cell type deconvolution in spatial transcriptomics. *eLife* **12**, RP88431 (2023).
122. Virshup, I. et al. The scverse project provides a computational ecosystem for single-cell omics data analysis. *Nat. Biotechnol.* **41**, 604–606 (2023).
123. Huber, W. et al. Orchestrating high-throughput genomic analysis with Bioconductor. *Nat. Methods* **12**, 115–121 (2015).
124. Amezquita, R. A. et al. Orchestrating single-cell analysis with Bioconductor. *Nat. Methods* **17**, 137–145 (2020).
125. Lähnemann, D. et al. Eleven grand challenges in single-cell data science. *Genome Biol.* **21**, 31 (2020).
126. Jiang, R., Sun, T., Song, D. & Li, J. J. Statistics or biology: the zero-inflation controversy about scRNA-seq data. *Genome Biol.* **23**, 31 (2022).
127. Silverman, J. D., Roche, K., Mukherjee, S. & David, L. A. Naught all zeros in sequence count data are the same. *Comput. Struct. Biotechnol. J.* **18**, 2789–2798 (2020).
128. Gatto, L. et al. Initial recommendations for performing, benchmarking and reporting single-cell proteomics experiments. *Nat. Methods* **20**, 375–386 (2023).
129. Valecha, M. & Posada, D. Somatic variant calling from single-cell DNA sequencing data. *Comput. Struct. Biotechnol. J.* **20**, 2978–2985 (2022).
130. Baker, E. A. G., Schapiro, D., Dumitrascu, B., Vickovic, S. & Regev, A. In silico tissue generation and power analysis for spatial omics. *Nat. Methods* **20**, 424–431 (2023).
131. Zappia, L., Phipson, B. & Oshlack, A. Splatter: simulation of single-cell RNA sequencing data. *Genome Biol.* **18**, 174 (2017).
132. Li, B. et al. Benchmarking spatial and single-cell transcriptomics integration methods for transcript distribution prediction and cell type deconvolution. *Nat. Methods* **19**, 662–670 (2022).
133. Raimundo, F., Prompsy, P., Vert, J.-P. & Vallot, C. A benchmark of computational pipelines for single-cell histone modification data. *Genome Biol.* **24**, 143 (2023).
134. Yuan, H. & Kelley, D. R. scBasset: sequence-based modeling of single-cell ATAC-seq using convolutional neural networks. *Nat. Methods* **19**, 1088–1096 (2022).
135. Ashuach, T., Reidenbach, D. A., Gayoso, A. & Yosef, N. PeakVI: a deep generative model for single-cell chromatin accessibility analysis. *Cell Rep. Methods* **2**, 100182 (2022).
136. Liu, Z., Sun, D. & Wang, C. Evaluation of cell–cell interaction methods by integrating single-cell RNA sequencing data with spatial information. *Genome Biol.* **23**, 218 (2022).
137. Long, B., Miller, J. & the SpaceTx Consortium. SpaceTx: a roadmap for benchmarking spatial transcriptomics exploration of the brain. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2301.08436 (2023).
138. Zhang, Y. et al. Reference-based cell type matching of in situ image-based spatial transcriptomics data on primary visual cortex of mouse brain. *Sci. Rep.* **13**, 9567 (2023).
139. Bullard, J. H., Purdom, E., Hansen, K. D. & Dudoit, S. Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. *BMC Bioinformatics* **11**, 94 (2010).
140. Yoshihara, K. et al. The landscape and therapeutic relevance of cancer-associated transcript fusions. *Oncogene* **34**, 4845–4854 (2015).
141. Lataretu, M. & Hölzer, M. RNAflow: an effective and simple RNA-seq differential gene expression pipeline using Nextflow. *Genes* **11**, 1487 (2020).
142. Tarazona, S., Garcia-Alcalde, F., Dopazo, J., Ferrer, A. & Conesa, A. Differential expression in RNA-seq: a matter of depth. *Genome Res.* **21**, 2213–2223 (2011).

143. Costa-Silva, J., Domingues, D. & Lopes, F. M. RNA-seq differential expression analysis: an extended review and a software tool. *PLoS ONE* **12**, e0190152 (2017).
144. Yang, E. W., Girke, T. & Jiang, T. Differential gene expression analysis using coexpression and RNA-seq data. *Bioinformatics* **29**, 2153–2161 (2013).
145. Zhang, Z. H. et al. A comparative study of techniques for differential expression analysis on RNA-seq data. *PLoS ONE* **9**, e103207 (2014).
146. Rajkumar, A. P. et al. Experimental validation of methods for differential gene expression analysis and sample pooling in RNA-seq. *BMC Genomics* **16**, 548 (2015).
147. Das, A., Das, D. & Panda, A. C. Validation of circular RNAs by PCR. *Methods Mol. Biol.* **2392**, 103–114 (2022).
148. Rai, M. F., Tycksen, E. D., Sandell, L. J. & Brophy, R. H. Advantages of RNA-seq compared to RNA microarrays for transcriptome profiling of anterior cruciate ligament tears. *J. Orthop. Res.* **36**, 484–497 (2018).
149. Beck, T. F., Mullikin, J. C., Program, N. C. S. & Biesecker, L. G. Systematic evaluation of Sanger validation of next-generation sequencing variants. *Clin. Chem.* **62**, 647–654 (2016).
150. Zheng, J. et al. A comprehensive assessment of next-generation sequencing variants validation using a secondary technology. *Mol. Genet. Genom. Med.* **7**, e00748 (2019).
151. Li, B. & Dewey, C. N. RSEM: accurate transcript quantification from RNA-seq data with or without a reference genome. *BMC Bioinformatics* **12**, 323 (2011).
152. Griebel, T. et al. Modelling and simulating generic RNA-seq experiments with the flux simulator. *Nucleic Acids Res.* **40**, 10073–10083 (2012).
153. Frazee, A. C., Jaffe, A. E., Langmead, B. & Leek, J. T. Polyester: simulating RNA-seq datasets with differential transcript expression. *Bioinformatics* **31**, 2778–2784 (2015).
154. Franklin, J. M., Schneeweiss, S., Polinski, J. M. & Rassen, J. A. Plasmode simulation for the evaluation of pharmacoepidemiologic methods in complex healthcare databases. *Comput. Stat. Data Anal.* **72**, 219–226 (2014).

Acknowledgements

The authors thank the reviewers and editors for their helpful comments and suggestions. This work was funded by the National Center for Advancing Translational Sciences (grant 5UL1TR000003).

Author contributions

The authors contributed equally to all aspects of the article.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41576-023-00679-6.

Peer review information *Nature Reviews Genetics* thanks Yvan Saeys and the other, anonymous, reviewer for their contribution to the peer review of this work.

Related links

Open Problems in Single-Cell Analysis: https://openproblems.bio
SpaceTx: https://spacetx.github.io
Updates and domain-specific templates: https://github.com/itmat/OmicsBenchmarkReport