Title: Towards reproducible, transparent, and systematic benchmarking of omics computational tools

Serghei Mangul^{1\$}, Lana S. Martin^{1\$}, Brian Hill¹, Angela Ka-Mei Lam¹, Margaret Distler³, Alex Zelikovsky⁴, Eleazar Eskin^{1,2} and Jonathan Flint³

¹Department of Computer Science, University of California Los Angeles, 580 Portola Plaza, Los Angeles, CA 90095, USA

²Department of Human Genetics, University of California Los Angeles, 695 Charles E. Young Drive South, Los Angeles, CA 90095, USA

³Center for Neurobehavioral Genetics, Semel Institute for Neuroscience and Human Behaviour, University of California-Los Angeles, CA 90095, USA

⁴Department of Computer Science, Georgia State University, Atlanta, GA 30303, USA

\$ - These authors contributed equally to the paper

Abstract

Rapid technological advances, such as the development of next-generation sequencing, are driving the need for comprehensive computational tools to analyze the wealth of generated genomic data. Systematic benchmarking has been successful in many research disciplines to help non-computational researchers evaluate the accuracy and applicability of new computational tools. Adopting a standardized benchmarking practice and following established principles for the design of new benchmarking studies could help researchers using omics data to better leverage technological innovation. In this Review, we propose step-by-step

instructions for increasing reusability, transparency, and reproducibility of benchmarking studies by using containerization, common data representation, open data, and systematic parameter description.

Introduction

Almost every biological discipline now uses genomic data; while geneticists move into an era where the ability to interrogate genome sequences from millions of individuals is imminent, cell biologists are taking the first steps to extract genomic information from tens of thousands of cells¹. By 2013, the European Bioinformatics Institute (EBI) was storing two petabytes (PB) of genome sequence data in their archive (ENA) — a number that has since doubled every year². The size of current biological data sets is so large that life scientists have become dependent on software tools for both data analysis and data visualization. Between 1994 and 2014, a bioinformatics software program was the most cited paper for half of the years³. Scientists' dependence on software tools comes at a cost: computational inefficiencies can result in non-trivial computational loads, substantially reducing time to complete analyses. Furthermore, even small errors can have profound consequences, potentially producing both false positive and false negative results.

Many new algorithms^{4,5} have been developed to accommodate the flood of genomic data, but the bioinformatics software development community currently lacks a systematic procedure for assessing their performances³. Without a standardized comparison, potential software users

lack an adequate guide for selecting tools that best suit their data. Researchers with limited computational backgrounds may lack sufficient contextual knowledge to weigh the advantages of adopting a new tool, which promises specific gains, against discarding an existing tool with proven capability. Unsystematic assessment of new algorithms creates a gap between tool developers and biomedical researchers, the end users of the developed tool.

The developer–researcher gap can be addressed with benchmarking studies, which inform the research community about the most appropriate tools for specific analytical tasks and data types^{6–8}. A benchmarking study consists of a robust and comprehensive evaluation of the abilities of existing algorithms to solve a particular computational biology problem. These studies use gold standard datasets as a ground truth and well-defined scoring metrics to assess the performance and accuracy of each tool when applied to a variety of analytical tasks and data types. Gold standard data are obtained through the most accurate experimental procedures available and represent a thoroughly tested benchmark; often the experimental procedures used to establish gold standard data carry too high a cost to be routinely applied in biomedical research. For example, Sanger sequencing is a highly accurate DNA sequencing technology that is limited in applicability by its high cost. By contrast, massively parallel DNA sequencing, also known as next-generation sequencing (NGS), is a popular technology that is widely available due to recent cost reductions of over two orders of magnitude⁹. The interpretation of NGS data requires sophisticated computational algorithms to overcome limitations posed by the length of resulting short reads and sequencing errors. Gold standard data obtained by Sanger sequencing used in conjunction with raw data obtained by NGS

enables explicit comparisons of existing computational algorithms, which allow biomedical researchers who may otherwise be unfamiliar with software development procedures to choose the ideal computational tool for a study.

In this Review, we summarize established principles for guiding the design of new benchmarking studies. We also discuss different strategies to optimize benchmarking, including crowdsourcing and challenge-based benchmarking^{10–12}. We also identify and discuss aspects of challenge-based benchmarking relevant to tests performed by individual research groups.

Problem: unsystematic comparison of tools

At present, assessment of a newly-published algorithm is typically performed by the researchers who develop the tool. An unsystematic assessment practice can lead to biases in published results, a phenomenon referred to as the “self-assessment trap.” Many computational laboratories use simulated data, as they lack adequate resources to generate or access gold standard experimental data when self-assessing a newly-developed tool. Using simulated data to estimate the performance of a tool is common practice yet poses several limitations. First, the models under which the simulated data are generated can differentially bias the outcomes of algorithms. For example, the algorithm itself could be trained on simulated data prior to running. Second, simulated data cannot capture true experimental variability and will always be less complex than real data¹³. Finally, not all simulated data are

validated with real-world data, and many methods used to simulate data have yet to be validated by the research community¹⁴.

Solution: systematic benchmarking

Systematic benchmarking based on gold standard data would inform the biomedical research community of the strengths and weaknesses associated with specific analytical tools in computational biology¹⁵. A benchmarking study first runs regular biotechnological protocols to produce the raw data, which serve as the input for the computational tools (Figure 1). Results obtained by running computational tools are compared against the gold standard data; comparison of these results with the gold standard allows researchers to use statistical and performance metrics to explicitly differentiate among existing computational algorithms in a standardized way. Types of technologies available for use in preparation of gold standard data are described in Section 'Defining gold standards.'. Evaluation methods and criteria that may be used to select the methods with the best performances are described in Section 'Selection of method with best performance.'. Ultimately, the generated data and robust scoring methodologies produced by benchmarking studies would be a valuable resource when shared with the research community.

Figure 1. Study design for benchmarking omics computational tools. In order to evaluate the accuracy of benchmarked computational tools, results obtained by running the computational tools are compared against the gold standard data (ground truth). First, the biological samples are probed by regular biotechnological protocols (i.a). Raw data generated by these protocols serve as the input for the computational tools (i.b-c). Results obtained by running computational tools are the final output of the biotechnological pipeline (i.d). Gold standard data, which are produced by the benchmarking procedure and are based on technological protocol, expert manual evaluation, synthetic mock community, curated databases, or are computationally simulated (ii.a). Types of technologies available for use in the preparation of gold standard data are described in Section ‘Defining gold standards.’ Some of the techniques used to generate the gold standard data produce raw data, which needs to be analyzed; other techniques directly produce the gold standard data (ii.c). Gold standard data obtained by or in conjunction with the raw data generated by regular biotechnological protocols enables researchers to use statistical metrics (iii.a) and metrics accessing the computational cost and speed to run the benchmarked computational tools (iii.b) to draw explicit, standardized comparison of existing computational algorithms. Methods with the best performances are located on Pareto frontier and are identified as Pareto efficient methods(iv). A method is considered to be Pareto efficient if no other benchmarked method improves the score of one of the evaluation metrics without degrading the score of another evaluation metric. Evaluation methods and criteria for selecting the methods with the best performances are described in Section ‘Selection of method with best performance’.

Although the primary goal of a benchmarking study is to produce a robust assessment of existing algorithms, the data generated by benchmarking studies can also be a valuable resource for the research community ¹⁶. Benchmarking data ultimately include gold standard data (Figure 1.ii) , raw data (Figure 1.i.b), and data generated by benchmarking tools (Figure 1.i.d). Access to data generated by benchmarking tools can easily improve the precision of newly developed tools by comparing a new method to tools indexed in the benchmarking study. Results of the benchmarking study can either be downloaded and the analysis locally run, or researchers can upload their own results and obtain a comparison through an online interface ^{17,18}. In both cases, benchmarking data allows researchers to easily compare newly developed tools against existing tools without installing and running third-party software — often an extremely complicated and time-consuming process, especially when the software lacks detailed documentation ¹⁷.

Despite the many advantages of reusing benchmarking data, there is a risk of overtraining the developed software to produce the best results on the particular gold standard dataset. This process is known as overfitting, which can cause the software to produce unreliable results on future datasets. Kanitz et al. ¹⁷ and Altenhoff et. al. ¹⁸ have implemented an online interface that evaluates the results of newly developed algorithms. However, mere access to the algorithm evaluation provides no realistic guarantee against algorithm overtraining. One potential approach to avoiding overfitting is to split benchmarking data into training data and test data, where training data is publically available and test data is reserved for evaluating the performance of new algorithms. Identical performances of the new algorithm based on training

data and test data will demonstrate minimal, if any, algorithm overfitting. Nevertheless, it is extremely important to continuously extend and update gold standards by incorporating novel benchmarks into training and test datasets. This approach avoids overtraining and is capable of meeting newly demanding modifications in usage and technology.

Additionally, benchmarking studies have the potential to combine the strengths of individual tools for particular usage or technology specifics into a more accurate consensus tool. For example, Aghaeepour et al.¹⁹ showed that the accuracy of cell population identification from flow cytometry data can be improved by combining predictions from individual computational algorithms.

Challenges to effective benchmarking

A review of 26 benchmarking efforts published across 10 relevant peer-reviewed journals from 2011 to 2017 reveals several aspects of software development unique to computational biology. Challenges inherent to computational biology pose a challenge to the establishment of an effective benchmarking platform (Table 1).

Individual versus competition-based benchmarking.

Our survey included 18 benchmarking studies performed by a single research group and seven challenge-based studies. Single research groups conduct individual benchmarking studies of relevant computational problems, whereas challenge-based studies take the form of an organized competition in which participants compete to solve the problems. One study included in our survey is driven by a hybrid approach that features both benchmarking types.

Challenge-based competitions aim to solve a fundamental research problem in a short period of time by building a scientific community around the topic. First, organizers provide participants with a training data set to develop novel methodologies. Next, participants apply developed methodologies to real data and submit results to a centralized hub, where the evaluation is performed²⁰. Challenge-based benchmarking was pioneered by Critical Assessment of protein Structure Prediction (CASP)²¹, the first community-wide contest held in 1994 to assess protein structure prediction methods. The inaugural challenge-based benchmarking event was followed by Critical Assessment of Massive Data Analysis (CAMDA, <http://www.camda.info/>), the first community-wide experiment in genomics, transcriptomics, metabolomics, and other omics domains^{21–23}. Since then, community-driven benchmarking efforts have been recognized as effective tools that are capable of enabling the evaluation of novel or existing computational methods²³. The limitations and challenges of competition-based benchmarking have been reviewed elsewhere^{10,23–25}.

A standardized benchmarking approach could use crowdsourcing motivated by competition to develop and run analytical algorithms on proposed data. A competitive crowdsourcing approach has been successfully applied across various domains of computational biology¹⁰. However, this approach may fail to account for potentially useful tools whose developers did not participate in the competition. In addition, the crowdsourcing approach requires organization of an active working group, and may not be suitable for independent groups planning to perform extensive benchmarking studies. Further, individual benchmarking only evaluates published tools, whereas competition-based studies also include novel methods that have not yet been published.

A wealth of new tools.

An overwhelmingly large number of software tools are currently available, and an increasing number of applications are released each month. For example, over 200 computational tools have been developed for variant analysis of next-generation genome sequencing data³³. Independent benchmarking teams would need to invest substantial effort in systematically assessing the accuracy of the extraordinary volume of new analytical methods. Tools are often excluded from a benchmarking study if they lack comprehensive documentation, require a complicated installation process, or are impossible to install and run in a reasonable amount of time¹⁷. Other benchmarking studies focus only on well-known or frequently used computational tools^{26–28}. Surveyed benchmarking studies evaluated an average of 11 computational tools designed to solve a targeted problem. Benchmarking studies performed by

one research team evaluated on average 18 computational tools, ranging from three to 32 tools. Competition-based studies evaluated on average 37 computational tools, ranging from 12 to 121 tools (Table 1).

Defining gold standards.

There is little consensus among researchers about what constitutes a gold standard experimental data set for each particular application. For example, what is the minimum number of samples that should be included in a benchmarking study? What are adequate levels of coverage and/or fidelity of data? Should there be molecular confirmation of data? These fundamental questions are presently unresolved; systematic benchmarking studies can provide the data and tools to support informed dialogue exploring these inquiries.

Producing gold-standard data can be a complicated and laborious process²⁹.

If possible, a trusted technology needs to be applied to obtain the gold standard benchmark (e.g. Sanger sequencing) (Table 2). Trusted technology often is not available and, in such cases, alternative technology is instead applied. An alternative technology is similar to trusted technology in that it requires minimal or no computational inference, a feature that allows the tools to avoid biases introduced during computational processing of data. In many cases, the accuracy of alternative technologies may be inadequate. The produced gold standards should be applied with caution in such cases. For example, qPCR — widely considered the gold standard for gene expression profiling — shows strong deviations of ~5-10% across various

targets³⁰. Alternatively, developers can apply an integration and arbitration approach⁷, which integrates results from multiple ordinary experimental procedures and generates a consensus that serves as a gold standard. For example, the Genome in the Bottle Consortium successfully generated a gold standard reference genome that includes a set of single-nucleotide polymorphisms and small indels by integrating and arbitrating across five sequencing technologies, seven read mappers, and three variant callers⁷. Despite the ability of this approach to reduce the number of false positives compared to each individual technology, disagreement between used technologies can result in incompleteness of the gold standard.

Due to the complexity of the biological system, it is impossible to obtain the ground truth in many applications. In these cases, instead of obtaining the golden standard, one can design a mock community (often referred as a synthetic mock community) by combining *in vitro* titrated proportions of community elements. The most popular mock communities are prepared as mixtures of known microbial organisms^{31,32}. When microbial organisms are closely related with similar sequences, such as intra-host RNA-virus populations, one should challenge computational methods with various frequency profiles and include closely related pairs^{19,32-34}. Despite the numerous advantages of the mock community gold standards, one should keep in mind that they are still artificial. Indeed, mock communities are usually comprised of a small number of members when compared to real communities.

In some cases, the output of the technology used to produce the gold standard can be manually evaluated. For example, a trained pathologist can manually evaluate a histological

image to determine the infiltration levels of lymphocytes in a tumor. This process allowed the pathologist to assign a tumor-infiltrating lymphocyte score ³⁵. Despite the robustness of the manually-produced output, the manual procedure cannot scale across multiple samples and lacks formal procedure, thereby limiting comparison of results produced by different experts.

Large curated databases promise to deliver a highly accurate set of genes, genes variants, and other genomic elements in the style of an Encyclopedia. Building such databases is a tremendous effort using a combination of computational analysis, manual annotation, and experimental validation that are exercised across multiple institutions and agencies. One example of such databases is GENCODE, database of highly accurate gene features that appear across the entire human genome ³⁶. Such databases often can serve as a gold standard, with the assumption that elements from the database cover the elements from the sample of study. However, this assumption is not always true due to incompleteness of such databases; the fact that some elements present in the sample are missing in the database may limit our ability to define true positives and false negatives. Despite those limitations, large curated databases grant a high level of sensitivity by allowing comparison of the number of elements in the sample and the database.

The benchmarked problem may, at times, represent a subproblem within a multi-step pipeline intended to solve a complex biological problem. For example, the scaffolding problem is a part of the assembly pipeline that starts with the assembly of reads into contigs and ends with the output of a scaffold, a set of chains each consisting of ordered oriented contigs bearing

estimated gaps between neighbors. The input contigs may be misassembled or may repeat each other; therefore, a “real” benchmark requires curation in order to produce a valid ground truth ^{37, 38}.

Lastly, the golden standard can be simulated on the computer. Such data is often referred to as simulated or *in silico* gold standard data. Despite the ability to generate such data at no cost, simulated data need only be used as a complementary to the real experimental gold standard data. Simulated data cannot be used as a gold standard as it fails to capture true experimental variability and will always be less complex than real data ¹³. Methods designed to simulate experimental data are constantly in development ¹⁴. Multiple attempts to improve the quality of simulated data have been executed; for example, efforts have been made to incorporate real and simulated data in one comprehensive data set. Ewing et al. ³⁹ proposed to introduce simulated cancer mutations into real sequencing data to analogously improve the quality of the sequencing data. Using real sequencing data rather than simulated data, this approach maintains the true properties of the sequencing reads.

Area of application	Study	Type of the study	Parameter optimization	Summary	Supporting documentation	Raw input data	Gold standard data	Data sharing	Evaluation metrics	Computational costs	Number of tools
Error correction	40	Reg.	No	Yes	No	Real	SIMUL	Part.	TPR, SPC	ExTIME, RAM	7
Flow cytometry analysis	19	Comp.	No	Yes	Yes	Real	EXPERT	Fully	F1	None	14
Genome assembly	41	Comp.	NA	Yes	Yes	Real	ALTECH	Fully	ACC, CORR	None	21
	37	Reg.	No	Yes	Yes	Both	SOFT WARE	Part.	None	CPU, RAM	10
Microbiome analysis	27,37	Reg.	No	Yes	Yes	Sim.	SIMUL	No	PPV, TPR, SPC, MCC	ExTIME	14
	8	Comp.	NA	Yes	Yes	Sim.	SIMUL	Fully	PPV, TPR.	None	6+9+10
	34	Reg.	No	Yes	Yes	Both	MOCK	Part.	PPV, TPR, F1, AUPR	ExTIME, RAM	11
Protein function prediction	42	Comp.	NA	Yes	No	Real	ALTECH	Part.	PPV, TPR, F1, ACC	None	54
	43	Comp.	NA	No	No	DB	DB	Part.	PPV, TPR, F1	None	121
	18	Reg.	Yes	Yes	No	DB	DB	Part.	PPV; TPR	None	15
Read alignment	44	Reg.	Yes	Yes	Yes	Both	SIMUL	Fully	TPR, ACC	ExTIME, CPU, RAM	9
	45	Comp.	NA	No	Yes	Both	SIMUL	Fully	PPV, TPR, F1, CORR.	None	12

	6	Reg.	Yes	Yes	Yes	Sim.	SIMUL	Part.	PPV, TPR	ExTIME, CPU, RAM	14
RNA-Seq analysis	6,46	Reg.	No	Yes	Part.	Real	DB	Fully	None	None	4
	47	Hybrid	NA	Yes	Part.	Real	ALTECH	Part.	PPV, TPR	None	14
	26	Reg.	Yes	Yes	Yes	Real	ALTECH	Part.	TPR,FPR, AUC	None	3
	30	Reg.	Yes	No	Yes	Real	ALTECH	Fully	ACC, CORR	None	6
	48	Reg.	Yes	Yes	Part.	Real	ALTECH	Fully	CORR	None	5
	17	Reg.	No	Yes	Yes	Both	ALTECH	Fully	CORR	ExTIME, CPU, RAM	11
	49	Reg.	No	No	No	Both	ALTECH	Part.	PPV; TPR; CORR	None	7
	50	Reg.	No	Yes	Part.	Real	ALTECH	Fully	TPR; SPC; FDR	None	7
Sequence Alignment	28	Reg.	No	No	No	DB	DB	Part.	None	None	8
Genome assembly	41	Comp.	NA	Yes	Yes	Real	ALTECH	Fully	ACC; CORR	None	21
Variant analysis	51	Reg.	No	Yes	No	Both	SIMUL	No	PPV;TPR	None	32
	39	Comp.	NA	No	No	Sim.	SIMUL	Part.	PPV; TPR; F1	None	14
	52	Reg.	Yes	Yes	Yes	Both	I&A	Part.	PPV,TPR	None	19

Table 1. Benchmarking studies published from 2011 to 2017. Surveyed benchmarking studies are grouped according to their area of application (indicated in column ‘Area of Application’). We documented benchmarking studies of different types (‘Type of the study’), such as those performed by a single group (‘Reg.’), a competition-based approach (‘Comp.’), and a hybrid approach (combining elements of ‘Regular’ and ‘Competition-based’). In competition-based benchmarking studies, parameter optimization (‘Parameter Optimization’) is performed by each team and is not mandatory (marked here as “N/A”). We documented whether benchmarking studies summarized the benchmarked algorithm’s features (‘Summary’). We recorded whether commands to install and run benchmarked tools were shared (‘Supporting documentation’). Types of raw input data (‘Raw input data’) and gold standard data (‘Gold standard data’) were documented across benchmarking study. When a benchmarking study uses computationally simulated data, we marked the study as ‘SIMUL’; when raw data was experimentally generated in the wet-lab, we marked the study as ‘Real’. When the study used both simulated and real data, we marked the study as ‘Both’. Gold standard data types included data that was computationally simulated (marked as ‘Sim’), manually evaluated by Experts (marked as ‘EXPERT’), prepared by alternative technology (‘marked as ALTECH’), prepared as Curated software input (marked as ‘SOFTWARE’), prepared as mock community (marked as ‘MOCK’), prepared from curated databases (marked as ‘DB’), and prepared using an Integration and Arbitration approach (marked as ‘I&A’). More details about the characteristics of techniques to prepare gold standard datasets are provided in Table 2. We documented whether the benchmarking data are shared publicly (‘Data sharing’) . We consider the benchmarking data to be shared ‘Fully’ if the gold standard data, raw input data, and the raw

output of each benchmarked tools are shared. When any one or more of those datasets is not shared publicly, we documented the study as partially ('Part.'). We recorded statistical measures derived from confusion matrix to evaluate the performance of these tools ('Evaluation metrics') and computational resources required to run the benchmarked tools ('Computational costs'). When the benchmarking study used none of the statistical measures from the confusion matrix, the study was marked as 'None'. We recorded three measures of computational costs: Execution time (marked as 'ExTIME'), CPU time (marked as 'CPU'), and maximum amount of RAM required to run the tool (marked as 'RAM'). We also recorded the number of tools benchmarked by each study ('Number of tools').

Technique	Advantages	Limitations
Trusted technology	<ul style="list-style-type: none"> • High accuracy • Direct, usually no computational inference is required 	<ul style="list-style-type: none"> • Carry high cost • Does not scale
Alternative technology	<ul style="list-style-type: none"> • Direct, usually no computational inference is required 	<ul style="list-style-type: none"> • Not necessarily more accurate
Integration and arbitration	<ul style="list-style-type: none"> • Using a consensus between the technologies allow to reduce number of false positives compared to each individual technology 	<ul style="list-style-type: none"> • Disagreement between used technologies results in incompleteness of the gold standard
Mock communities	<ul style="list-style-type: none"> • Ground truth is fully known, because raw data is generated from prepared gold standard 	<ul style="list-style-type: none"> • Small number of items (e.g. microbial species) compared to reality • Designed community is artificial

Expert manual evaluation	<ul style="list-style-type: none"> • Most suitable for specialist understanding 	<ul style="list-style-type: none"> • Does not scale • Lack formal procedure, limiting comparison of results produced by different experts
Curated databases	<ul style="list-style-type: none"> • Allows to access sensitivity, by comparing the number of elements in the sample and the database 	<ul style="list-style-type: none"> • Incompleteness of curated databases results in limited ability to define true positives and false negatives
Curated software input	<ul style="list-style-type: none"> • Ground truth is fully known, because raw data is generated from prepared gold standard 	<ul style="list-style-type: none"> • Does not validate on real inputs which usually contain errors
Simulated gold standard	<ul style="list-style-type: none"> • Ground truth is fully known, because raw data is generated from prepared gold standard • Cost-free generation of multiple gold standards 	<ul style="list-style-type: none"> • Technology is simulated, and cannot capture true experimental variability and will always be less complex than real data • Gold standard data is artificial

Table 2. Advantages and limitations of various techniques to prepare the gold standard data.

Our survey of benchmarking tools reveals a large variety of techniques currently used to prepare gold standard data. According to our survey, the most popular technique (10 out of 26 surveyed benchmarking studies) for preparing gold standard data is to use various alternative technologies (Table 1). The second most popular technique (8 surveyed studies) is to computationally simulate the gold standard. As previously mentioned, simulated data can

never fully capture true experimental variability and should be used complementary to real gold standard data. Other techniques include manual preparation of gold standard data by experts¹⁹ and use of available databases and literature references^{18,28,43,46}.

Selecting default parameters versus parameter optimization.

The process for evaluating a software tool is usually a non-binary decision and involves choosing specific parameters and input pre-processing. Parameter optimization is a computationally intensive task that requires running multiple rounds of the same tool with different combinations of parameters. Fifty-percent of the surveyed regular benchmarking studies performed parameter optimization; other studies use tools with default parameters (Table 1). For example, tuning parameters of RNA-Seq aligners consistently increases the number of correctly mapped reads by an average of 10% across all 14 state-of-the-art aligners⁶. Even a small number of parameters results in an unfeasibly enormous number of all possible parameter combinations for each benchmarked tool. Various heuristics are used to narrow the search space; usually narrowing is performed by prioritizing parameters for optimization. A benchmarking study from⁶ recommends identifying parameters that have the most effect on the quality of a tool and then optimize this effect over several combinations of the parameters. For example, the most influential parameters for RNA-Seq aligners are the number of allowed mismatches and seed length⁶. In competition-based benchmarking studies, parameter optimization is optional; such benchmarking studies rely on the expertise of the tool's developer to choose optimal parameters.

Availability of benchmarking data and supporting documentation.

All the data generated by a benchmarking study offer substantial value to the software development and research community — yet these data are often not shared (Table 1). Factors contributing to the current low rate of data and code sharing with newly developed methods include an absence of journal policies requiring the public sharing of these resources and infrastructural challenges to sharing large data generated by the benchmarking studies⁵³. Ideally, a benchmarking study should make publicly available all benchmarking data and code necessary to process data and reproduce results¹⁹. While the vast majority of surveyed benchmarking studies are widely disseminated benchmarking data, only 42% of the studies surveyed completely shared benchmarking data (Table 1). Most studies adopted the ‘shared upon request’ model, which is a less reliable and less reproducible method of data dissemination as it relies on individual authors’ availability to perpetually share data^{53,54}.

Only half of the surveyed benchmarking projects shared commands crucially important for properly running benchmarked tools (Table 1). Under ideal circumstances, the commands and parameters required to optimally run and install the benchmarked tool would be publically available, allowing maximum computational reproducibility of the benchmarking study⁵⁵. Providing supporting documentation is particularly important for benchmarked tools that have complicated installation processes or that require prior installation of dependencies to be easily adopted by the community^{37,55}. However, many peer-reviewed journals that publish benchmarking studies do not require the sharing of benchmarking data and supporting documentation. To date, 70% of the surveyed benchmarking studies do not to publicly

disseminate the raw output data of a benchmarked algorithms. Further, 50% of surveyed benchmarking studies do not to share supporting documentation on installing and running the benchmark tools (Table 1). Ideally, supporting documentation needs to be shared as an easy-to-use interface instead of main text and/or supplementary materials, both of which make it easier for researchers to adopt the method recommended by the benchmarking study^{8,56}.

Selection of method with best performance

Statistical measures to evaluate the benchmarked methods are derived from the confusion matrix. According to the confusion matrix, all predictions are classified as true positives, false positives (also known as type I error), false negatives (also known as type II error), and true negative, which are the number of correct predictions (hits), false predictions, missed predictions, and correct rejections, respectively. The most common measure is precision (also known as positive predictive value); that is, the fraction of positive predictions made by a test that are true, and sensitivity (also known as true positive rate or recall); that is, the fraction of known positives that are correctly predicted by a test. The two factors are often combined into an F-score measure (also known as F1 Score or F-measure), a harmonic mean of precision and recall with a high f-score indicating a good method. If true negatives are defined, one can calculate the specificity (also known as true negative rate), accuracy, and Matthews correlation coefficient (MCC). When the benchmarked method predicts the relative frequencies, one can use correlation between true and predicted relative frequencies. When the benchmarked method produces a non-uniform distribution of frequencies, with few items accounting for

majority of frequencies, correlation cannot accurately account for rare items since correlation is completely dominated by the major players. To avoid this, one can partition the items in several frequency intervals (usually four quartiles) and separately report correlation for each interval. An alternative metric treating all frequencies equally can be represented by the median percent error (MPE), which is the computed median of absolute percent errors by which predicted frequencies differ from the true frequencies^{57; 58}. We observe a variety of statistical measures used to evaluate performance of benchmarked methods. The most popular ones were true positive rate and positive predictive value, used by 17 and 13 studies, respectively (Table 1). Our survey also documented three benchmarking studies that use none of the standard statistical measures derived from a confusion matrix.

Important underrated evaluation metrics carry computational costs, currently defined as the runtime and amount of computational resources required to run the tool. The most accurate software tool cannot be widely useful if the required infrastructure is unaffordable or if the program's running time is extremely long. A tool's computational resources are typically measured by the maximum amount of RAM required to run the method. The running time is usually measured in CPU time. Since execution time may vary across different servers, standard procedure recommends to report specifications of the server and number of processors used. While not easy to measure, factors critical for the wide adoption of a tool by the scientific community include ease of installing and running the tool as well as availability of options and developers' timely response to users' requests. In our survey, the vast majority of

benchmarking studies failed to report computational resources and time required to run the evaluated tools.

A benchmarking study often cannot identify a single winner according to all evaluation metrics. Instead, identifying multiple methods with excellent performance under different evaluation criteria is a valid outcome ⁴¹. In this case, a set of Pareto efficient benchmarked methods forming a Pareto frontier is identified. A method is Pareto efficient if no other benchmarked method improves one of the evaluation metrics without degrading another evaluation metric. For example, the fastest method with a satisfactory level of accuracy and the slowest method with the highest level of accuracy are both identified as Pareto efficient; in such cases, both methods would be reported as winners (Figure 1).

Continuous benchmarking.

We note that benchmarking studies can only evaluate the performance of the methods implemented in a current release of the software. New releases of the method can potentially differ in accuracy and running time, an observation that suggests a community-wide need for a permanent benchmarking effort ⁵⁹.

Community adoption.

It is important that the community accepts the need for benchmarking and adopts it as a standard. However, challenges to establishing an effective benchmarking platform, compounded by a lack of incentive, may limit the number of researchers willing to conduct benchmarking studies. Scientific journals can promote the attractiveness of benchmarking by allocating to researchers a special track in each issue for benchmarking papers. Researchers at universities can more efficiently manage the workflow of benchmarking by teaching undergraduate students to perform ‘low level’ tasks, such as installing and running software tools^{59,60}.

Box 1. Principles for best benchmarking practices

Our survey results point to several specific opportunities for establishing a systematic benchmarking platform. Here we present eight principles to guide researchers in designing a benchmarking study. The majority of surveyed benchmarking studies address only a few of these principles. These guidelines emphasize reproducibility and enable more effective utilization of resources and data. Below we propose step-by-step instructions for conducting benchmarking in the most comprehensive and reproducible way.

Compile a comprehensive list of tools to be benchmarked. Identify the list of software tools that are most suitable for the analytical task and data types of the benchmarking project. One approach is to perform a PubMed (<https://www.ncbi.nlm.nih.gov/pubmed/>) search for relevant

articles. Once relevant publications are identified, complete the compiled list by reviewing references of each identified publication. We advise compiling the most comprehensive list of tools available. A tool may be impossible to install and run in a reasonable time; in order to save the effort of other researchers, we suggest documenting such cases in log files. Instructions on how to create log file are provided in Supplementary Text 1.

We note that the number of methods can be extremely large, with up to 200 tools currently available for some domains of computational biology⁵¹. Given an overwhelming number of relevant tools, researchers can undertake challenge-based benchmarking and invite the developers of the bioinformatics tools to participate in the challenge by running their tools. Alternatively, researchers may review all published tools and select the most popular algorithms for the benchmarking study based on the number of citations and/or reputation of the journal. However, selecting the best tool *a priori* is risky, as tool popularity, journal impact, and citation frequency do not necessarily imply a particular algorithm is optimal^{3,6,61}.

Prepare and describe benchmarking data. Compose a table summarizing the benchmarking data. Explain protocols used for preparing the raw and gold standard data sets. Describe any potential limitations of the data; for example, can the benchmarking data bias the performance of any specific type of algorithm? Share methods of benchmarking data preparation, complete provenance and, if preferable, code for gathering and cleaning data. Preparing a bias-free benchmarking data set can be an extremely complicated and laborious process, and is discussed elsewhere^{39,62,63}.

Select evaluation metrics. Metrics for evaluating the accuracy of the software tools need to be carefully determined and packed in the form of scripts. Such scripts can be used by the community to evaluate the performance of any new algorithms. Often, comparison of the results obtained by computational tools with a gold standard is not a straightforward task. For example, comparing variant calls may be confounded by different representations of insertions, deletions and multiple nucleotide polymorphisms — particularly in complex regions of the genome ⁶⁴. Benchmarking studies need to consider and document nuances in data representation.

Parameter optimization. Deciding how a method is applied to a data set usually involves choosing parameters and input pre-processing. Parameter optimization is often best understood by the method developers. In the case of competition-based benchmarking, the participants will determine the optimal parameters for the tool they run. In the case of non-competition-based benchmarking, authors need access to all useful combinations of parameters in order to identify combinations of parameters that perform best for a given algorithm.

Summarize algorithm features and share the commands to install and run tools. Prepare a table outlining the benchmarked algorithm's features, the underlying algorithm, the dependencies of the software, and the journal the algorithm was published in. For example, features supported by benchmarked mapping tools were summarized in a table by Hatem and

Ayat et al.⁴⁴. The template of a table outlining the benchmarked algorithms is provided in Supplementary table S1. Consider verifying table contents with the relevant tool developers. Summarizing algorithm features is often a straightforward task, but 27% of surveyed benchmarking studies fail to provide such a summary (Table 1). Finally, providing additional, detailed instructions for how to install and run the benchmarked tools produces a valuable resource for the research community. For example, when a large number of dependencies are required to run the tools, this may represent a significant barrier for potential users. In case where many dependencies are required to run the tool, it is preferable to share the benchmarked tool in the form of a computable environment such as virtual machine images or containers, for example, Docker (<https://www.docker.com/>)⁵⁶, an easy-to-use interface that packages software with all its dependencies and parameters, enabling flexibility and portability of the software tool across platforms and operating systems. To the best of our knowledge, only a single benchmarking study (CAMI)⁸ wrapped tools as portable containers. Prior to running the software, contact the software developers to ensure the correctness of chosen commands and parameters.

Define a universal format (if necessary). When the output of each tool is different, develop and share a script with the community to generate a universal format. Even though data types and formats in many fields of computational biology are rapidly changing, benchmarking studies should take a lead in standardizing data types and formats, which can then be adopted by software developers. For example, the QfO consortium has defined common file formats for orthology inference methods¹⁸.

Provide a flexible interface to download the data. Without easy access, the community cannot independently verify benchmarking data. Share an easy-to-use interface that can be used to download the input raw data and the gold standard data, to maximize data reusability. In addition to the input and gold standard data, we recommend sharing the raw output data of each benchmarked tool, so that the end user can apply their own evaluation metrics. Scripts available via such interfaces can also be used to reproduce the results and figures of benchmarking studies, ultimately increasing their transparency and computational reproducibility of benchmarking studies ³⁷. For example, an interface was implemented by The Critical Assessment of Metagenome Interpretation (CAMI) ^{8,37} and is available via GitHub (<https://github.com/dkoslicki/CAMIProfilinAnalysis>). Ideally, the repositories storing code and data should be permanently archived (for example, via Zenodo <https://zenodo.org/>) to prevent stored materials from being changed or removed.

Many principles, including running all published tools, and defining and calculating performance metrics in a robust way, can be time-consuming to perform correctly and require significant care in design and execution ⁶⁵.

---End of Box 1.

Conclusions

Computational biology is rapidly advancing thanks to the many new tools developed and published each month. However, we cannot easily expand upon inconsistent benchmarking efforts that are not well-documented and are not widely disseminated to the academic community. A systematic benchmarking practice would help biomedical researchers leverage this technological expansion to optimize their projects. Although the community is interested in systematic benchmarking of previously published methods, running algorithms developed by other researchers is a challenging task for tools with many dependencies and limited documentation. The extraordinary volume of new analytical methods that are published each month compounds the challenge of accurately testing each tool. These challenges discourage the research community from performing systematic benchmarking studies of computational biology methods. Nonetheless, a consolidated approach that implements principles used by other communities is feasible. Proposed principles will make computational biology benchmarking studies more sustainable and reproducible, ultimately increasing the transparency of biomedical data and results.

References

1. Macosko, E. Z. *et al.* Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets. *Cell* **161**, 1202–1214 (2015).
2. Marx, V. The big challenges of big data. *Nature* **498**, 255–260 (2013).
3. Wren, J. D. Bioinformatics programs are 31-fold over-represented among the highest impact scientific papers of the past two decades. *Bioinformatics* **32**, 2686–2691 (2016).
4. Nagarajan, N. & Pop, M. Sequence assembly demystified. *Nat. Rev. Genet.* **14**, 157–167 (2013).
5. Hackl, H., Charoentong, P., Finotello, F. & Trajanoski, Z. Computational genomics tools for dissecting tumour-immune cell interactions. *Nat. Rev. Genet.* **17**, 441–458 (2016).
6. Baruzzo, G. *et al.* Simulation-based comprehensive benchmarking of RNA-seq aligners. *Nat. Methods* **14**, 135–139 (2017).
7. Zook, J. M. *et al.* Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls. *Nat. Biotechnol.* **32**, 246–251 (2014).
8. Sczyrba, A. *et al.* Critical Assessment of Metagenome Interpretation-a benchmark of metagenomics software. *Nat. Methods* **14**, 1063–1071 (2017).
9. Shendure, J. & Ji, H. Next-generation DNA sequencing. *Nat. Biotechnol.* **26**, 1135–1145 (2008).
10. Boutros, P. C., Margolin, A. A., Stuart, J. M., Califano, A. & Stolovitzky, G. Toward better benchmarking: challenge-based methods assessment in cancer genomics. *Genome Biol.* **15**, 462 (2014).

11. Costello, J. C. & Stolovitzky, G. Seeking the wisdom of crowds through challenge-based competitions in biomedical research. *Clin. Pharmacol. Ther.* **93**, 396–398 (2013).
12. Meyer, P. *et al.* Verification of systems biology research in the age of collaborative competition. *Nat. Biotechnol.* **29**, 811–815 (2011).
13. Maier, R., Zimmer, R. & Küffner, R. A Turing test for artificial expression data. *Bioinformatics* **29**, 2603–2609 (2013).
14. 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).
15. 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).
16. Sansone, S.-A., Cruse, P. & Thorley, M. High-quality science requires high-quality open data infrastructure. *Sci Data* **5**, 180027 (2018).
17. Kanitz, A. *et al.* Comparative assessment of methods for the computational inference of transcript isoform abundance from RNA-seq data. *Genome Biol.* **16**, (2015).
18. Altenhoff, A. M. *et al.* Standardized benchmarking in the quest for orthologs. *Nat. Methods* **13**, 425–430 (2016).
19. Aghaeepour, N. *et al.* Critical assessment of automated flow cytometry data analysis techniques. *Nat. Methods* **10**, 228–238 (2013).
20. Marbach, D. *et al.* Wisdom of crowds for robust gene network inference. *Nat. Methods* **9**, 796–804 (2012).
21. Moult, J., Pedersen, J. T., Judson, R. & Fidelis, K. A large-scale experiment to assess protein structure prediction methods. *Proteins* **23**, ii–v (1995).

22. Johnson, K. & Lin, S. Call to work together on microarray data analysis. *Nature* **411**, 885 (2001).
23. Tilstone, C. Vital statistics. *Nature* **424**, 610–612 (2003).
24. Capella-Gutierrez, S. *et al.* Lessons Learned: Recommendations for Establishing Critical Periodic Scientific Benchmarking. (2017). doi:10.1101/181677
25. Saez-Rodriguez, J. *et al.* Crowdsourcing biomedical research: leveraging communities as innovation engines. *Nat. Rev. Genet.* **17**, 470–486 (2016).
26. Zhang, Z. H. *et al.* A comparative study of techniques for differential expression analysis on RNA-Seq data. *PLoS One* **9**, e103207 (2014).
27. Lindgreen, S., Adair, K. L. & Gardner, P. P. An evaluation of the accuracy and speed of metagenome analysis tools. *Sci. Rep.* **6**, 19233 (2016).
28. Thompson, J. D., Linard, B., Lecompte, O. & Poch, O. A comprehensive benchmark study of multiple sequence alignment methods: current challenges and future perspectives. *PLoS One* **6**, e18093 (2011).
29. Mason, C. E., Afshinnekoo, E., Tighe, S., Wu, S. & Levy, S. International Standards for Genomes, Transcriptomes, and Metagenomes. *J. Biomol. Tech.* **28**, 8–18 (2017).
30. SEQC/MAQC-III Consortium. A comprehensive assessment of RNA-seq accuracy, reproducibility and information content by the Sequencing Quality Control Consortium. *Nat. Biotechnol.* **32**, 903–914 (2014).
31. Human Microbiome Project Consortium. A framework for human microbiome research. *Nature* **486**, 215–221 (2012).
32. Artyomenko, A. *et al.* Long Single-Molecule Reads Can Resolve the Complexity of the

- Influenza Virus Composed of Rare, Closely Related Mutant Variants. *J. Comput. Biol.* **24**, 558–570 (2017).
33. Giallonardo, F. D. *et al.* Full-length haplotype reconstruction to infer the structure of heterogeneous virus populations. *Nucleic Acids Res.* **42**, e115 (2014).
 34. McIntyre, A. B. R. *et al.* Comprehensive benchmarking and ensemble approaches for metagenomic classifiers. *Genome Biol.* **18**, 182 (2017).
 35. Smyrk, T. C., Watson, P., Kaul, K. & Lynch, H. T. Tumor-infiltrating lymphocytes are a marker for microsatellite instability in colorectal carcinoma. *Cancer* **91**, 2417–2422 (2001).
 36. Derrien, T. *et al.* The GENCODE v7 catalog of human long noncoding RNAs: analysis of their gene structure, evolution, and expression. *Genome Res.* **22**, 1775–1789 (2012).
 37. Hunt, M., Newbold, C., Berriman, M. & Otto, T. D. A comprehensive evaluation of assembly scaffolding tools. *Genome Biol.* **15**, R42 (2014).
 38. Mandric, I., Knyazev, S. & Zelikovsky, A. Repeat-aware evaluation of scaffolding tools. *Bioinformatics* (2018). doi:10.1093/bioinformatics/bty131
 39. Ewing, A. D. *et al.* Combining tumor genome simulation with crowdsourcing to benchmark somatic single-nucleotide-variant detection. *Nat. Methods* **12**, 623–630 (2015).
 40. Yang, X., Chockalingam, S. P. & Aluru, S. A survey of error-correction methods for next-generation sequencing. *Brief. Bioinform.* **14**, 56–66 (2013).
 41. Bradnam, K. R. *et al.* Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species. *Gigascience* **2**, 10 (2013).
 42. Radivojac, P. *et al.* A large-scale evaluation of computational protein function prediction. *Nat. Methods* **10**, 221–227 (2013).

43. Jiang, Y. *et al.* An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biol.* **17**, 184 (2016).
44. Hatem, A., Bozdağ, D., Toland, A. E. & Çatalyürek, Ü. V. Benchmarking short sequence mapping tools. *BMC Bioinformatics* **14**, 184 (2013).
45. Earl, D. *et al.* Alignathon: a competitive assessment of whole-genome alignment methods. *Genome Res.* **24**, 2077–2089 (2014).
46. Łabaj, P. P. *et al.* Characterization and improvement of RNA-Seq precision in quantitative transcript expression profiling. *Bioinformatics* **27**, i383–91 (2011).
47. Steijger, T. *et al.* Assessment of transcript reconstruction methods for RNA-seq. *Nat. Methods* **10**, 1177–1184 (2013).
48. Li, S. *et al.* Detecting and correcting systematic variation in large-scale RNA sequencing data. *Nat. Biotechnol.* **32**, 888–895 (2014).
49. 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).
50. Łabaj, P. P. & Kreil, D. P. Sensitivity, specificity, and reproducibility of RNA-Seq differential expression calls. *Biol. Direct* **11**, 66 (2016).
51. Pabinger, S. *et al.* A survey of tools for variant analysis of next-generation genome sequencing data. *Brief. Bioinform.* **15**, 256–278 (2014).
52. Bohnert, R., Vivas, S. & Jansen, G. Comprehensive benchmarking of SNV callers for highly admixed tumor data. *PLoS One* **12**, e0186175 (2017).
53. Stodden, V., Seiler, J. & Ma, Z. An empirical analysis of journal policy effectiveness for

- computational reproducibility. *Proc. Natl. Acad. Sci. U. S. A.* **115**, 2584–2589 (2018).
54. Langille, M. G. I., Ravel, J. & Florian Fricke, W. ‘Available upon request’: not good enough for microbiome data! *Microbiome* **6**, (2018).
 55. Baker, M. 1,500 scientists lift the lid on reproducibility. *Nature* **533**, 452–454 (2016).
 56. Leipzig, J. A review of bioinformatic pipeline frameworks. *Brief. Bioinform.* **18**, 530–536 (2017).
 57. Nicolae, M., Mangul, S., Măndoiu, I. I. & Zelikovsky, A. Estimation of alternative splicing isoform frequencies from RNA-Seq data. *Algorithms Mol. Biol.* **6**, 9 (2011).
 58. Li, B. & Dewey, C. N. RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinformatics* **12**, 323 (2011).
 59. Puton, T., Kozłowski, L. P., Rother, K. M. & Bujnicki, J. M. CompRNA: a server for continuous benchmarking of automated methods for RNA secondary structure prediction. *Nucleic Acids Res.* **41**, 4307–4323 (2013).
 60. Mangul, S., Martin, L. S. & Eskin, E. Involving undergraduates in genomics research to narrow the education–research gap. *Nat. Biotechnol.* **36**, 369–371 (2018).
 61. Gardner, P. P. *et al.* A meta-analysis of bioinformatics software benchmarks reveals that publication-bias unduly influences software accuracy. (2016). doi:10.1101/092205
 62. Alberti, C. *et al.* An Evaluation Framework for Lossy Compression of Genome Sequencing Quality Values. *Proc Data Compress Conf* **2016**, 221–230 (2016).
 63. Costello, J. C. *et al.* A community effort to assess and improve drug sensitivity prediction algorithms. *Nat. Biotechnol.* **32**, 1202–1212 (2014).
 64. Cleary, J. G. *et al.* Comparing Variant Call Files for Performance Benchmarking of Next-

Generation Sequencing Variant Calling Pipelines. (2015). doi:10.1101/023754

65. Zheng, S. Benchmarking: contexts and details matter. *Genome Biol.* **18**, 129 (2017).

Glossary

Algorithms in computational biology are designed to model biological and/or ecological processes so that scientists can identify meaningful patterns in real or simulated data from fields such as biology, chemistry, genomics, and evolution.

Computational biology is a rapidly growing interdisciplinary field that applies computer science, mathematics, and statistics to problems in biology.

Gold standard is a data set that serves as a ground truth to estimate the accuracy of the benchmarking algorithms.

Parameter optimization (or tuning) is a process of choosing a set of input parameters for the benchmarked method that leads to best performance. Usually parameter optimization is a computationally intensive process that requires running the tool many times with multiple combinations of parameters.

Portable containers (or virtual machine images) are tools to containerize software with all relevant dependencies and parameters. Portable containers enable flexibility and portability across platforms and operating systems.

Supplementary Materials

[provided as a separate file]

Table S1. Summary of error correction algorithm features and publication details.

Supplementary Text 1.

An example of a log file for a software tool installation and running

In the log file includes the dependencies and documents the process of installing the software tools and corresponding dependencies. Including: Any errors that occurred while installing dependencies and the command's used to overcome these installation problems. The log file documents the type of files that is needed to input into the tools and the format of the output file.

This is the possible structure of the log file:

- Input
- Output
- Dependencies
- Commands used to Install the tool
- Commands used to run the tool
- Reason the tool is impossible to install. This should include the exact error message and document steps (if any) which were performed to resolve the problem. In case the software developers were contacted, their suggestions should be listed here.

Supplementary Text 2

List of names of the studies with corresponding citation used in Table 1.

- A survey of error-correction methods for next-generation sequencing ⁴⁰
- Critical assessment of automated flow cytometry data analysis techniques ¹⁹
- Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species ⁴¹
- A comprehensive evaluation of assembly scaffolding tools ³⁷
- An evaluation of the accuracy and speed of metagenome analysis tools ^{27,37}
- Critical Assessment of Metagenome Interpretation—a benchmark of metagenomics software ⁸
- Comprehensive benchmarking and ensemble approaches for metagenomic classifiers ³⁴
- A large-scale evaluation of computational protein function prediction ⁴²
- An expanded evaluation of protein function prediction methods shows an improvement in accuracy ⁴³
- Standardized benchmarking in the quest for orthologs ¹⁸
- Benchmarking short sequence mapping tools ⁴⁴
- Alignathon: a competitive assessment of whole-genome alignment methods ⁴⁵
- Simulation-based comprehensive benchmarking of RNA-seq aligners ⁶
- Characterization and improvement of RNA-Seq precision in quantitative transcript expression profiling ^{6,46}
- Assessment of transcript reconstruction methods for RNA-seq ⁴⁷

- A Comparative Study of Techniques for Differential Expression Analysis on RNA-Seq Data

26

- A comprehensive assessment of RNA-seq accuracy reproducibility and information content by the Sequencing Quality Control Consortium³⁰
- Detecting and correcting systematic variation in large-scale RNA sequencing data⁴⁸
- Comparative assessment of methods for the computational inference of transcript isoform abundance from RNA-seq data¹⁷
- Benchmark analysis of algorithms for determining and quantifying full-length mRNA splice forms from RNA-seq data⁴⁹
- Sensitivity, specificity, and reproducibility of RNA-Seq differential expression calls⁵⁰
- A Comprehensive Benchmark Study of Multiple Sequence Alignment Methods: Current Challenges and Future Perspectives²⁸
- Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species⁴¹
- A survey of tools for variant analysis of next-generation genome sequencing data⁵¹
- Combining tumor genome simulation with crowdsourcing to benchmark somatic single-nucleotide-variant detection³⁹
- Comprehensive benchmarking of SNV callers for highly admixed tumor data⁵²