bioRxiv preprint doi: https://doi.org/10.1101/092205; this version posted September 2, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

RESEARCH

Sustained software development, not number of citations or journal choice, is indicative of accurate bioinformatic software

Paul P. Gardner[1,2*], James M. Paterson[3], Stephanie McGimpsey[4], Fatemeh Ashari-Ghomi[5], Sinan U. Umu[6], Aleksandra Pawlik[7], Alex Gavryushkin[8] and Michael A. Black[1]

*Correspondence: paul.gardner@otago.ac.nz
[1]Department of Biochemistry,, University of Otago, Dunedin, New Zealand
Full list of author information is available at the end of the article

Abstract

Background: Computational biology provides widely used and powerful software tools for testing and making inferences about biological data. In the face of rapidly increasing volumes of data, heuristic methods that trade software speed for accuracy may be employed. We are have studied these trade-offs using the results of a large number of independent software benchmarks, and evaluated whether external factors are indicative of accurate software.

Method: We have extracted accuracy and speed ranks from independent benchmarks of different bioinformatic software tools, and evaluated whether the speed, author reputation, journal impact, recency and developer efforts are indicative of accuracy.

Results: We found that software speed, author reputation, journal impact, number of citations and age are all unreliable predictors of software accuracy. This is unfortunate because citations, author and journal reputation are frequently cited reasons for selecting software tools. However, GitHub-derived records and high version numbers show that the accurate bioinformatic software tools are generally the product of many improvements over time, often from multiple developers.

Discussion: We also find that the field of bioinformatics has a large excess of slow and inaccurate software tools, and this is consistent across many sub-disciplines. Meanwhile, there are few tools that are middle-of-road in terms of accuracy and speed trade-offs. We hypothesise that a form of publication-bias influences the publication and development of bioinformatic software. In other words, software that is intermediate in terms of both speed and accuracy may be difficult to publish - possibly due to author, editor and reviewer practices. This leaves an unfortunate hole in the literature as the ideal tools may fall into this gap. For example, high accuracy tools are not always useful if years of CPU time are required, while high speed is not useful if the results are also inaccurate.

Background

Computational biology software is widely used and has produced some of the most cited publications in the entire scientific corpus [1, 2, 3]. These highly-cited software tools include implementations of methods for sequence alignment and homology inference [4, 5, 6, 7], phylogenetic analysis [8, 9, 10, 11, 12], biomolecular structure analysis [13, 14, 15, 16, 17], and visualization and data collection [18, 19]. However, the popularity of a software tool does not necessarily mean that it is accurate or

bioRxiv preprint doi: https://doi.org/10.1101/092205; this version posted September 2, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

computationally efficient; instead usability, ease of installation, operating system support or other indirect factors may play a greater role in a software tool's popularity. Indeed, there have been several notable incidences where convenient, yet inaccurate software has caused considerable harm [20, 21, 22].

Progress in the biological sciences is increasingly limited by the ability to analyse large volumes of data, therefore the dependence of biologists on software is also increasing [23]. There is an increasing reliance on technological solutions for automating biological data generation (e.g. next-generation sequencing, mass-spectroscopy, cell-tracking and species tracking), therefore the biological sciences have become increasingly dependent upon software tools for processing large quantities of data [23]. As a consequence, the computational efficiency of data processing and analysis software is of great importance to decrease the energy, climate impact, and time costs of research [24]. Furthermore, as datasets become larger even small error rates can have major impacts on the number of false inferences [25].

The gold-standard for determining accuracy is for researchers independent of individual tool development to conduct benchmarking studies; these benchmarks can serve a useful role in reducing the over-optimistic reporting of software accuracy [26, 27, 28] and the self-assessment trap [29, 30]. Benchmarking typically involves the use a number of positive and negative control datasets, so that predictions from different software tools can be partitioned into true or false groups, allowing a variety of metrics to be used to evaluate performance [31, 32, 28]. The aim of these benchmarks is to robustly identify tools that make acceptable compromises in terms of balancing speed with discriminating true and false predictions, and are therefore suited for wide adoption by the community.

For common computational biology tasks, a proliferation of software-based solutions often exists [33, 34, 35]. While this is a good problem to have, and points to a diversity of options from which practical solutions can be selected, having many possible options creates a dilemma for users. In the absence of any recent gold-standard benchmarks, how should scientific software be selected? In the following we presume that the "biological accuracy" of predictions is the most desirable feature for a software tool. Biological accuracy is the degree to which predictions or measurements reflect the biological truths based on expert-derived curated datasets (see Methods for the mathematical definition used here).

A number of possible predictors of software quality are used by the community of computational biology software users [36, 37, 38]. Some accessible, quantifiable and frequently used proxies for identifying high quality software include: **1. Recency:** recently published software tools may have built upon the results of past work, or be an update to an existing tool. Therefore these may be more accurate and faster. **2. Wide adoption:** a software tool may be widely used because it is fast and accurate, or because it is well-supported and user-friendly. In fact,"large user base", "word-of-mouth", "wide-adoption", "personal recommendation", and "recommendation from a close colleague", were frequent responses to surveys of "how do scientists select software?" [36, 37, 38]. **3. Journal impact:** many believe that high profile journals are run by editors and reviewers who carefully select and curate the best manuscripts. Therefore, high impact journals may be more likely to select manuscripts describing good software [39]. **4. Author/group reputation:** the key

bioRxiv preprint doi: https://doi.org/10.1101/092205; this version posted September 2, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

to any project is the skills of the people involved, including maintaining a high collective intelligence [37, 40, 41]. As a consequence, an argument could be made that well respected and high-profile authors may write better software [42, 43]. **5. Speed:** software tools frequently trade accuracy for speed. For example, heuristic software such as the popular homology search tool, BLAST, compromises the mathematical guarantee of optimal solutions for more speed [4, 7]. Some researchers may naively interpret this fact as implying that slower software is likely to be more accurate. But speed may also be influenced by the programming language [44], and the level of hardware optimisation [45, 46]; however, the specific method of implementation generally has a greater impact (e.g., brute-force approaches versus rapid and sensitive pre-filtering [47, 48, 49]). **6. Effective software versioning:** With the wide adoption of public version-control systems like GitHub, quantifiable data on software development time and intensity indicators, such as the number of contributors to code, number of code changes and versions is now available [50, 51, 52].

In the following study, we explore factors that may be indicative of software accuracy. This, in our opinion, should be one of the prime reasons for selecting a software tool. We have mined the large and freely accessible PubMed database [53] for benchmarks of computational biology software, and manually extracted accuracy and speed rankings for 499 unique software tools. For each tool, we have collected measures that may be predictive of accuracy, and may be subjectively employed by the research community as a proxy for software quality. These include relative speed, relative age, the productivity and impact of the corresponding authors, journal impact, number of citations and GitHub activity.

Results

We have collected relative accuracy and speed ranks for 499 distinct software tools. This software has been developed for solving a broad cross-section of computational biology tasks. Each software tool was benchmarked in at least one of 69 publications that satisfy the Boulesteix criteria [54]. In brief, the Boulesteix criteria are: 1. the main focus of the article is a benchmark. 2. the authors are reasonably neutral. 3. the test data and evaluation criteria are sensible.

For each of the publications describing these tools, we have (where possible) collected the journal's H5-index (Google Scholar Metrics), the maximum H-index and corresponding M-indices [42] for the corresponding authors for each tool, and the number of times the publication(s) associated with a tool has been cited using Google Scholar (data collected over a 6 month period in late 2020). Note that citation metrics are not static and will change over time. In addition, where possible we also extract the version number, the number of commits, number of contributors total number "issues", the proportion of issues that remain open, the number of pull requests, and the number of times the code was forked from public GitHub repositories.

We have computed the Spearman's correlation coefficient for each pairwise combination of the mean normalised accuracy and speed ranks, with the year published, mean relative age (compared to software in the same benchmarks), journal H5 metrics, the total number of citations, the relative number of citations (compared to software in the same benchmarks) and the maximum H- and corresponding M-indices for the corresponding authors, version number, and the GitHub statistics

bioRxiv preprint doi: https://doi.org/10.1101/092205; this version posted September 2, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

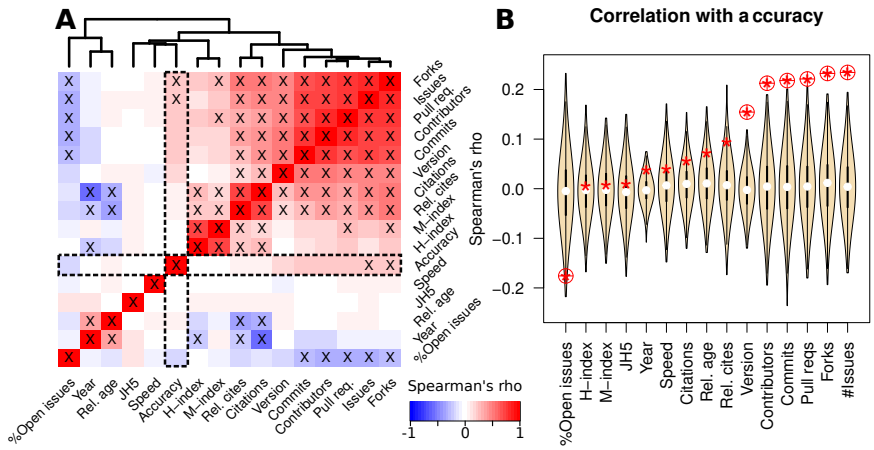

Figure 1: **A.** A heatmap indicating the relationships between different features of bioinformatic software tools. Spearman's rho is used to infer correlations between metrics such as citations based metrics, the year and relative age of publication, version number, GitHub derived activity measures, and the mean relative speed and accuracy rankings. Red colours indicate a positive correlation, blue colours indicate a negative correlation. Correlations with a P-value less than 0.05 (corrected for multiple-testing using the Benjamini-Hochberg method) are indicated with a 'X' symbol. The correlations with accuracy are illustrated in more detail in **B**, the relationship between speed and accuracy is shown in more detail in **Figure 2**. **B.** Violin plots of Spearman's correlations for permuted accuracy ranks and different software features. The unpermuted correlations are indicated with a red asterisk. For each benchmark, 1,000 permuted sets of accuracy and speed ranks were generated, and the ranks were normalised to lie between 0 and 1 (see Methods for details). Circled asterisks are significant (empirical P-value < 0.05, corrected for multiple-testing using the Benjamini-Hochberg method).

commits, contributors, pull requests, issues, % open issues and forks. The results are presented in Figure 1A&B, and Additional file 1: Figures S5&S6. We find significant associations between most of the citation-based metrics (journal H5, citations, relative citations, H-index and M-index). There is also a negative correlation between the year of publication, the relative age and many of the citation-based metrics.

Data on the number of updates to software tools from GitHub such as the version number, and numbers of contributors, commits, forks and issues was significantly correlated with software accuracy (respective Spearman's rhos = 0.15, 0.21, 0.22, 0.23, 0.23 and respective Benjamini & Hochberg corrected P-values = 5.4×10^{-4}, 1.1×10^{-3}, 7.9×10^{-4}, 3.4×10^{-4}, 3.2×10^{-4}, Additional file 1: Figures S6). The significance of these features was further confirmed with a permutation test (Figure 1B). These features were not correlated with speed however (see Figure 1A & Additional file 1: Figures S5&S6). We also found that reputation metrics such as citations, author and journal H-indices, and the age of tools were generally **not** correlated with either tool accuracy or speed (Figure 1A&B).

In order to gain a deeper understanding of the distribution of available bioinformatic software tools on a speed versus accuracy landscape, we ran a permutation

bioRxiv preprint doi: https://doi.org/10.1101/092205; this version posted September 2, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

test. The ranks extracted from each benchmark were randomly permuted, generating 1,000 randomized speed and accuracy ranks. In the cells of a 3×3 grid spanning the normalised speed and accuracy ranks we computed a Z-score for the observed number of tools in a cell, compared to the expected distributions generated by 1,000 randomized ranks. The results of this are shown in Figure 2. We identified 4 of 9 bins where there was a significant excess or dearth of tools. For example, there was an excess of "slow and inaccurate" software (Z=3.39, P-value=3.5×10^{-4}), with more moderate excess of "slow and accurate" and "fast and accurate" software (Z=2.49 & 1.7, P=6.3×10^{-3} & 0.04 respectively). We find that only the "fast and inaccurate" extreme class is at approximately the expected proportions based upon the permutation test (Figure 2B).

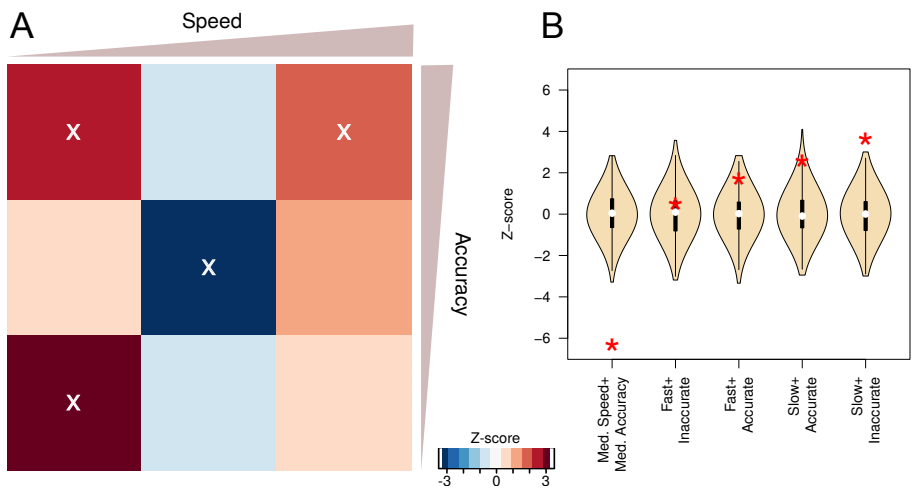

Figure 2: **A.** A heatmap indicating the relative paucity or abundance of software in the range of possible accuracy and speed rankings. Redder colours indicate an abundance of software tools in an accuracy and speed category, while bluer colours indicate scarcity of software in an accuracy and speed category. The abundance is quantified using a Z-score computation for each bin, this is derived from 1,000 random permutations of speed and accuracy ranks from each benchmark. Mean normalised ranks of accuracy and speed have been binned into 9 classes (a 3×3 grid) that range from comparatively slow and inaccurate to comparatively fast and accurate. Z-scores with a P-value less than 0.05 are indicated with a 'X'. **B.** The z-score distributions from the permutation tests (indicated with the wheat coloured violin plots) compared to the z-score for the observed values for each of the corner and middle square of the heatmap.

The largest difference between the observed and expected software ranks is the reduction in the number of software tools that are classed as intermediate in terms of both speed and accuracy based on permutation tests (see Methods for details, Figure 2). The middle cell of Figure 2A and left-most violin plot of Figure 2B highlight this extreme, (Z=-6.38, P-value=9.0×10^{-11}).

bioRxiv preprint doi: https://doi.org/10.1101/092205; this version posted September 2, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Conclusion

We have gathered data on the relative speeds and accuracies of 499 bioinformatic tools from 69 benchmarks published between 2005 and 2020. Our results provide significant support for the suggestion that there are major benefits to the long-term support of software development [55]. The finding of a strong relationship between the number of commits and code contributors to GitHub (i.e. software updates) and accuracy, highlights the benefits of long-term or at least intensive development.

Our study finds little evidence to support that impact-based metrics have any relationship with software quality, which is unfortunate, as these are frequently cited reasons for selecting software tools [38]. This implies that high citation rates for bioinformatic software [1, 2, 3] is more a reflection of other factors such as user-friendliness or the Matthew Effect [56, 57] other than accuracy. Specifically, software tools published early are more likely to appear in high impact journals due to their perceived novelty and need. Yet without sustained maintenance these may be outperformed by subsequent tools, yet early publications still accrue citations from users, and all subsequent software publications as tools need to be compared in order to publish. Subsequent tools are not perceived to be as novel, hence appear in "lower" tier journals, despite being more reliable. Hence, the "rich" early publishers get richer in terms of citations. Indeed, citation counts are mainly predictive of age (Figure 1A).

We found the lack of a correlation between software speed and accuracy surprising. The slower software tools are over-represented at both high and low levels of accuracy, with older tools enriched in this group (Figure 2 and Additional file 1: Figure S7). In addition, there is an large under-representation of software that has intermediate levels of both accuracy and speed. A possible explanation for this is that bioinformatic software tools are bound by a form of publication-bias [58, 59]. That is, the probability that a study being published is influenced by the results it contains [60]. The community of developers, reviewers and editors may be unwilling to publish software that is not highly ranked on speed or accuracy. If correct, this may have unfortunate consequences as these tools may nevertheless have further uses.

While we have taken pains to mitigate many issues with our analysis, nevertheless some limitations remain. For example, it has proven difficult to verify if the gap in medium accuracy and medium speed software is genuinely the result of publication bias, or due to additional factors that we have not taken in to account. In addition, all of the features we have used here are moving targets. For example, as software tools are refined, their relative accuracies and speeds will change, the citation metrics, ages, and version control derived measures also change over time. Here we report a snapshot of values from 2020. The benchmarks themselves may also introduce biases into the study. For example, there are issues with a potential lack of independence between benchmarks (e.g., shared datasets, metrics and tools), there are heterogeneous measures of accuracy and speed and often unclear processes for including different tools.

We propose that the full spectrum of software tool accuracies and speeds serves a useful purpose to the research community. Like negative results, if honestly reported this information, illustrates to the research community that certain approaches

bioRxiv preprint doi: https://doi.org/10.1101/092205; this version posted September 2, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

are not practical research avenues [61]. The current novelty-seeking practices of many publishers, editors, reviewers and authors of software tools therefore may be depriving our community of tools for building effective and productive workflows. Indeed, the drive for novelty may be an actively harmful criteria for the software development community, just as it is for reliable and reproducible research [62]. Novelty-criteria for publication may, in addition, discourage continual, incremental improvements in code post-publication in favour of splashy new tools that are likely to accrue more citations.

In addition we suggest that further efforts be made to encourage continual updates to software tools. To paraphrase some of the suggestions of Siepel (2019), these efforts may include more secure positions for developers, institutional promotion criteria include software maintenance, lower publication barriers for significant software updates, encourage further funding for software maintenance and improvement - not just new tools [55]. If these issues were recognised by research managers, funders and reviewers, then perhaps the future bioinformatic software tool landscape will be much improved.

The most reliable way to identify accurate software tools is through neutral software benchmarks [54]. We are hopeful that this, along with steps to reduce the publication-bias we have described, will reduce the over-optimistic and misleading reporting of tool accuracy [26, 27, 29].

Methods

In order to evaluate predictors of computational biology software accuracy, we mined the published literature, extracted data from articles, connected these with bibliometric databases, and tested for correlates with accuracy. We outline these steps in further detail below.

Criteria for inclusion: We are interested in using computational biology benchmarks that satisfy Boulesteix's (ALB) three criteria for a "neutral comparison study" [54]. Firstly, the main focus of the article is the comparison and **not** the introduction of a new tool as these can be biased [30]. Secondly, the authors should be reasonably neutral, which means that the authors should not generally have been involved in the development of the tools included in the benchmark. Thirdly, the test data and evaluation criteria should be sensible. This means that the test data should be independent of data that tools have been trained upon, and that the evaluation measures appropriately quantify correct and incorrect predictions. In addition, we excluded benchmarks with too few tools ≤ 3, or those where the results were inaccessible (no supplementary materials or poor figures).

Literature mining: We identified an initial list of 10 benchmark articles that satisfy the ALB-criteria. These were identified based upon previous knowledge of published articles and were supplemented with several literature searches (e.g., ["benchmark" AND "cputime"] was used to query both GoogleScholar and Pubmed [53, 63]). We used these articles to seed a machine-learning approach for identifying further candidate articles and to identify new search terms to include. This is outlined in Additional file 1: Figure S1.

For our machine-learning-based literature screening, we computed a score, $s(a)$, for each article that tells us the likelihood that it is a benchmark. In brief, our approaches uses 3 stages:

bioRxiv preprint doi: https://doi.org/10.1101/092205; this version posted September 2, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

1. Remove high frequency words from the title and abstract of candidate articles (e.g. 'the', 'and', 'of', 'to', 'a', ...)
2. Compute a log-odds score for the remaining words
3. Use a sum of log-odds scores to give a total score for candidate articles

For stage 1, we identified a list of high frequency (e.g. $f(\text{word}) > 1/10,000$) words by pooling the content of two control texts [64, 65].

For stage 2, in order to compute a log-odds score for bioinformatic words, we computed the frequency of words that were not removed by our high frequency filter in two different groups of articles: bioinformatics-background and bioinformatics-benchmark articles. The text from bioinformatics-background articles were drawn from the bioinformatics literature, but these were not necessarily associated with benchmark studies. For background text we used Pubmed [53, 63] to select 8,908 articles that contained the word "bioinformatics" in the title or abstract and were published between 2013 and 2015. We computed frequencies for each word by combining text from titles and abstracts for the background and training articles. A log-odds score was computed for each word using the following formula:

$$lo(word) = \log_2 \frac{f_{tr}(word) + \delta}{f_{bg}(word) + \delta}$$

Where δ was a pseudo-count added for each word ($\delta = 10^{-5}$, by default), $f_{bg}(word)$ and $f_{tr}(word)$ were the frequencies of a *word* in the background and training datasets respectively. Word frequencies were computed by counting the number of times a word appears in the pool of titles and abstracts, the counts were normalised by the total number of words in each set. Additional file 1: Figure S2 shows exemplar word scores.

Thirdly, we also collected a group of candidate benchmark articles by mining Pubmed for articles that were likely to be benchmarks of bioinformatic software, these match the terms: "((bioinformatics) AND (algorithms OR programs OR software)) AND (accuracy OR assessment OR benchmark OR comparison OR performance) AND (speed OR time)". Further terms used in this search were progressively added as relevant enriched terms were identified in later iterations. The final query is given in **Additional file 1**.

A score is computed for each candidate article by summing the log-odds scores for the words in title and abstract, i.e. $s(a) = \sum_i^N lo(w_i)$. The high scoring candidate articles are then manually evaluated against the ALB-criteria. Accuracy and speed ranks were extracted from the articles that met the criteria, and these were added to the set of training articles. The evaluated candidate articles that did not meet the ALB-criteria were incorporated into the set of background articles. This process was iterated and resulted in the identification of 69benchmark articles, containing 134 different benchmarks. Together these ranked 499 distinct software packages.

There is a potential for bias to have been introduced into this dataset. Some possible forms of bias include converging on a niche group of benchmark studies due to the literature mining technique that we have used. A further possibility is that benchmark studies themselves are biased, either including very high performing or very low performing software tools. To address each of these concerns we have

bioRxiv preprint doi: https://doi.org/10.1101/092205; this version posted September 2, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

attempted to be as comprehensive as possible in terms of benchmark inclusion, as well as including comprehensive benchmarks (i.e., studies that include all available software tools that address a specific biological problem).

Data extraction and processing: for each article that met the ALB-criteria and contained data on both the accuracy and speed from their tests, we extracted ranks for each tool. Until published datasets are made available in consistent, machine-readable formats this step is necessarily a manual process – ranks were extracted from a mixture of manuscript figures, tables and supplementary materials, each data source is documented in Additional file 2: Table S1. In addition, a variety of accuracy metrics are reported e.g. "accuracy", "AUROC", "F-measure", "Gain", "MCC", "N50", "PPV", "precision", "RMSD", "sensitivity", "TPR", "tree error", etc. Our analysis makes the necessarily pragmatic assumption that highly ranked tools on one accuracy metric will also be highly ranked on other accuracy metrics. Many articles contained multiple benchmarks, in these cases we recorded ranks from each of these, the provenance of which is stored with the accuracy metric and raw speed and accuracy rank data for each tool (Additional file 2: Table S1). In line with rank-based statistics, the cases where tools were tied were resolved by using a midpoint rank (e.g., if tools ranked 3 and 4 are tied, the rank 3.5 was used) [66]. Each rank extraction was independently verified by at least one other co-author to ensure both the provenance of the data could be established and that the ranks were correct. The ranks for each benchmark were then normalised to lie between 0 and 1 using the formula $1 - \frac{r-1}{n-1}$ where 'r' is a tool's rank and 'n' is the number of tools in the benchmark. For tools that were benchmarked multiple times with multiple metrics (e.g., BWA was evaluated in 6 different articles [67, 68, 69, 70, 71, 72]) a mean normalised rank was used to summarise the accuracy and speed performance. Or, more formally:

$$accuracy = \sum_{i=1..N} 1 - \frac{r_i^{accuracy} - 1}{n_i - 1},$$
$$speed = \sum_{i=1..N} 1 - \frac{r_i^{speed} - 1}{n_i - 1}$$

For each tool we identified the corresponding publications in GoogleScholar; the total number of citations was recorded, the corresponding authors were also identified, and if they had public GoogleScholar profiles, we extracted their H-index and calculated a M-index ($\frac{H-index}{y}$) where 'y' is the number of years since their first publication. The journal quality was estimated using the H5-index from GoogleScholar Metrics.

The year of publication was also recorded for each tool. "Relative age" and "relative citations" were also computed for each tool. For each benchmark, software was ranked by year of first publication (or number of citations), ranks were assigned and then normalised as described above. Tools ranked in multiple evaluations were then assigned a mean value for "relative age" and "relative citations".

The papers describing tools were checked for information on version numbers and links to GitHub. Google was also employed to identify GitHub repositories. When

bioRxiv preprint doi: https://doi.org/10.1101/092205; this version posted September 2, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

a repository was matched with a tool, the number of "commits" and number of "contributors" was collected, when details of version numbers were provided, these were also harvested. Version numbers are inconsistently used between groups, and may begin at either 0 or 1. To counter this issue we have added '1' to all versions less than '1', for example, version 0.31 become 1.31. In addition, multiple point releases may be used e.g. 'version 5.2.6', these have been mapped to the nearest decimal value '5.26'.

Statistical analysis: For each tool we manually collected up to 12 different statistics from GoogleScholar, GitHub and directly from literature describing tools (1. corresponding author's H-index, 2. corresponding author's M-index, 3. journal H5 index, 4. normalised accuracy rank, 5. normalised speed rank, 6. number of citations, 7. relative age, 8. relative number of citations, 9. year first published, 10. version 11. number of commits to GitHub, 12. number of contributors to GitHub). These were evaluated in a pairwise fashion to produce Figure 1 A&B, the R code used to generate these is given in a GitHub repository (linked below).

For each benchmark of three or more tools, we extracted the published accuracy and speed ranks. In order to identify whether there was an enrichment of certain accuracy and speed pairings we constructed a permutation test. The individual accuracy and speed ranks were reassigned to tools in a random fashion and each new accuracy and speed rank pairing was recorded. For each benchmark this procedure was repeated 1,000 times. These permuted rankings were normalised and compared to the real rankings to produce the 'X' points in Figure 1B and the heatmap and histograms in Figure 2. The heatmap in Figure 2 is based upon Z-scores ($Z = \frac{x-\bar{x}}{s}$). For each cell in a 3×3 grid a Z-score (and corresponding P-value is computed, either with the 'pnorm' distribution function in R (Figure 2A) or empirically (Figure 2B)) is computed to illustrate the abundance or lack of tools in a cell relative to the permuted data.

The distributions for each feature and permuted accuracy or speed ranks are shown in Additional file 1: Figures S3&S4. Scatter-plots for each pair of features is shown in Additional file 1: Figure S5. Plots showing the sample sizes for each tool, and feature are shown in Additional file 1: Figure S8, illustrates a power analysis to show what effect sizes we are likely to detect for our sample sizes.

Acknowledgements
The authors acknowledge the valued contribution of invaluable discussions with Anne-Laure Boulesteix, Shinichi Nakagawa, Suetonia Palmer and Jason Tylianakis. Murray Cox, Raquel Norel, Alexandros Stamatakis, Jens Stoye, Tandy Warnow, and Luis Pedro Coelho and three anonymous reviewers provided valuable feedback on drafts of the manuscript.
This work was largely conducted on the traditional territory of Kāi Tahu.

Funding
PPG is supported by a Rutherford Discovery Fellowship, administered by the Royal Society Te Apārangi, PPG, AG and MAB acknowledge support from a Data Science Programmes grant (UOAX1932).

Availability of data and materials
Raw datasets, software and documents are available under a CC-BY license:
`https://github.com/Gardner-BinfLab/speed-vs-accuracy-meta-analysis`

Competing interests
The authors declare that they have no competing interests.

Additional information
Additional file 1
Supplementary Figures S1-S8 and the neutral software benchmark reference list the accuracy and speed data is derived from [1-66].

bioRxiv preprint doi: https://doi.org/10.1101/092205; this version posted September 2, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Additional file 2
Supplementary Tables S1-S7.

Author details
[1]Department of Biochemistry,, University of Otago, Dunedin, New Zealand. [2]Biomolecular Interaction Centre, University of Canterbury, Christchurch, New Zealand. [3]Department of Civil and Natural Resources Engineering, University of Canterbury, Christchurch, New Zealand. [4]Parasites and Microbes, Wellcome Sanger Institute, Hinxton, UK. [5]Research Group for Genomic Epidemiology, Technical University of Denmark, Kongens Lyngby, Denmark. [6]Department of Research, Cancer Registry of Norway, Oslo, Norway. [7]Manaaki Whenua - Landcare Research, Lincoln, New Zealand. [8]Department of Computer Science, University of Otago, Dunedin, New Zealand.

References

1. Perez-Iratxeta, C., Andrade-Navarro, M.A., Wren, J.D.: Evolving research trends in bioinformatics. Brief. Bioinform. **8**(2), 88–95 (2007)
2. Van Noorden, R., Maher, B., Nuzzo, R.: The top 100 papers. Nature **514**(7524), 550–553 (2014)
3. Wren, J.D.: Bioinformatics programs are 31-fold over-represented among the highest impact scientific papers of the past two decades. Bioinformatics (2016)
4. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: Basic local alignment search tool. J. Mol. Biol. **215**(3), 403–410 (1990)
5. Thompson, J.D., Higgins, D.G., Gibson, T.J.: CLUSTAL w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acids Res. **22**(22), 4673–4680 (1994)
6. Thompson, J.D., Gibson, T.J., Plewniak, F., Jeanmougin, F., Higgins, D.G.: The CLUSTAL_X windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools. Nucleic Acids Res. **25**(24), 4876–4882 (1997)
7. Altschul, S.F., Madden, T.L., Schäffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J.: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. **25**(17), 3389–3402 (1997)
8. Felsenstein, J.: Confidence limits on phylogenies: An approach using the bootstrap. Evolution **39**(4), 783–791 (1985)
9. Saitou, N., Nei, M.: The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol. Biol. Evol. **4**(4), 406–425 (1987)
10. Posada, D., Crandall, K.A.: MODELTEST: testing the model of DNA substitution. Bioinformatics **14**(9), 817–818 (1998)
11. Ronquist, F., Huelsenbeck, J.P.: MrBayes 3: Bayesian phylogenetic inference under mixed models. Bioinformatics **19**(12), 1572–1574 (2003)
12. Tamura, K., Dudley, J., Nei, M., Kumar, S.: MEGA4: Molecular evolutionary genetics analysis (MEGA) software version 4.0. Mol. Biol. Evol. **24**(8), 1596–1599 (2007)
13. Sheldrick, G.M.: Phase annealing in SHELX-90: direct methods for larger structures. Acta Crystallogr. A **46**(6), 467–473 (1990)
14. Sheldrick, G.M.: A short history of SHELX. Acta Crystallogr. A **64**(Pt 1), 112–122 (2008)
15. Jones, T.A., Zou, J.Y., Cowan, S.W., Kjeldgaard, M.: Improved methods for building protein models in electron density maps and the location of errors in these models. Acta Crystallogr. A **47 (Pt 2)**, 110–119 (1991)
16. Laskowski, R.A., MacArthur, M.W., Moss, D.S., Thornton, J.M.: PROCHECK: a program to check the stereochemical quality of protein structures. J. Appl. Crystallogr. **26**(2), 283–291 (1993)
17. Otwinowski, Z., Minor, W.: [20] processing of x-ray diffraction data collected in oscillation mode. In: Methods in Enzymology vol. Volume 276, pp. 307–326. Academic Press, ??? (1997)
18. Kraulis, P.J.: MOLSCRIPT: a program to produce both detailed and schematic plots of protein structures. J. Appl. Crystallogr. **24**(5), 946–950 (1991)
19. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E.: The protein data bank. Nucleic Acids Res. **28**(1), 235–242 (2000)
20. Leveson, N.G., Turner, C.S.: An investigation of the therac-25 accidents. Computer **26**(7), 18–41 (1993)
21. Cummings, M., Britton, D.: Regulating safety-critical autonomous systems: past, present, and future perspectives. In: Living with Robots, pp. 119–140. Elsevier, ??? (2020)
22. Herkert, J., Borenstein, J., Miller, K.: The boeing 737 max: Lessons for engineering ethics. Sci Eng Ethics **26**(6), 2957–2974 (2020). doi:10.1007/s11948-020-00252-y
23. Marx, V.: Biology: The big challenges of big data. Nature **498**(7453), 255–260 (2013)
24. Gombiner, J.: Carbon footprinting the internet. Consilience-The Journal of Sustainable Development **5**(1) (2011)
25. Storey, J.D., Tibshirani, R.: Statistical significance for genomewide studies. Proc. Natl. Acad. Sci. U. S. A. **100**(16), 9440–9445 (2003)
26. Boulesteix, A.: Over-optimism in bioinformatics research. Bioinformatics **26**(3), 437–439 (2010)
27. Jelizarow, M., Guillemot, V., Tenenhaus, A., Strimmer, K., Boulesteix, A.: Over-optimism in bioinformatics: an illustration. Bioinformatics **26**(16), 1990–1998 (2010)
28. Weber, L.M., Saelens, W., Cannoodt, R., Soneson, C., Hapfelmeier, A., Gardner, P.P., Boulesteix, A.L., Saeys, Y., Robinson, M.D.: Essential guidelines for computational method benchmarking. Genome Biol **20**(1), 125 (2019). doi:10.1186/s13059-019-1738-8
29. Norel, R., Rice, J.J., Stolovitzky, G.: The self-assessment trap: can we all be better than average? Mol. Syst. Biol. **7**(1), 537 (2011)
30. 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**(1), 152 (2021). doi:10.1186/s13059-021-02365-4

bioRxiv preprint doi: https://doi.org/10.1101/092205; this version posted September 2, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

31. Egan, J.P.: Signal Detection Theory and ROC-analysis. Series in Cognition and Perception. Academic Press, New York, ??? (1975)
32. Hall, T., Beecham, S., Bowes, D., Gray, D., Counsell, S.: A systematic literature review on fault prediction performance in software engineering. IEEE Trans. Software Eng. **38**(6), 1276–1304 (2012)
33. Felsenstein, J.: Phylogeny programs. Internet address: http://evolution. gs. washington. edu/phylip/software. html (1995)
34. Altschul, S., Demchak, B., Durbin, R., Gentleman, R., Krzywinski, M., Li, H., Nekrutenko, A., Robinson, J., Rasband, W., Taylor, J., Trapnell, C.: The anatomy of successful computational biology software. Nat. Biotechnol. **31**(10), 894–897 (2013)
35. Henry, V.J., Bandrowski, A.E., Pepin, A., Gonzalez, B.J., Desfeux, A.: OMICtools: an informative directory for multi-omic data analysis. Database **2014** (2014)
36. Hannay, J.E., MacLeod, C., Singer, J., Langtangen, H.P., Pfahl, D., Wilson, G.: How do scientists develop and use scientific software? In: Proceedings of the 2009 ICSE Workshop on Software Engineering for Computational Science and Engineering. SECSE '09, pp. 1–8. IEEE Computer Society, Washington, DC, USA (2009)
37. Joppa, L.N., McInerny, G., Harper, R., Salido, L., Takeda, K., O'Hara, K., Gavaghan, D., Emmott, S.: Troubling trends in scientific software use. Science **340**(6134), 814–815 (2013)
38. Loman, N., Connor, T.: Bioinformatics infrastructure and training survey. Figshare (2015)
39. Garfield, E.: Citation indexes for science; a new dimension in documentation through association of ideas. Science **122**(3159), 108–111 (1955)
40. Woolley, A.W., Chabris, C.F., Pentland, A., Hashmi, N., Malone, T.W.: Evidence for a collective intelligence factor in the performance of human groups. Science **330**(6004), 686–688 (2010)
41. Cheruvelil, K.S., Soranno, P.A., Weathers, K.C., Hanson, P.C., Goring, S.J., Filstrup, C.T., Read, E.K.: Creating and maintaining high-performing collaborative research teams: the importance of diversity and interpersonal skills. Front. Ecol. Environ. **12**(1), 31–38 (2014)
42. Hirsch, J.E.: An index to quantify an individual's scientific research output. Proc. Natl. Acad. Sci. U. S. A. **102**(46), 16569–16572 (2005)
43. Bornmann, L., Mutz, R., Daniel, H.: Are there better indices for evaluation purposes than the h-index? a comparison of nine different variants of the h-index using data from biomedicine. J. Am. Soc. Inf. Sci. **59**(5), 830–837 (2008)
44. Fourment, M., Gillings, M.R.: A comparison of common programming languages used in bioinformatics. BMC Bioinformatics **9**, 82 (2008)
45. Farrar, M.: Striped Smith–Waterman speeds database searches six times over other SIMD implementations. Bioinformatics **23**(2), 156–161 (2007)
46. Dematté, L., Prandi, D.: GPU computing for systems biology. Brief. Bioinform. **11**(3), 323–333 (2010)
47. Schaeffer, J.: The history heuristic and alpha-beta search enhancements in practice. IEEE Trans. Pattern Anal. Mach. Intell. **11**(11), 1203–1212 (1989)
48. Papadimitriou, C.H.: Computational complexity. In: Encyclopedia of Computer Science, pp. 260–265. John Wiley and Sons Ltd., Chichester, UK (2003)
49. Leiserson, C.E., Thompson, N.C., Emer, J.S., Kuszmaul, B.C., Lampson, B.W., Sanchez, D., Schardl, T.B.: There's plenty of room at the top: What will drive computer performance after moore's law? Science **368**(6495) (2020)
50. Ray, B., Posnett, D., Filkov, V., Devanbu, P.: A large scale study of programming languages and code quality in github. In: Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering, pp. 155–165 (2014)
51. Dozmorov, M.G.: Github statistics as a measure of the impact of open-source bioinformatics software. Front Bioeng Biotechnol **6**, 198 (2018). doi:10.3389/fbioe.2018.00198
52. Mangul, S., Mosqueiro, T., Abdill, R.J., Duong, D., Mitchell, K., Sarwal, V., Hill, B., Brito, J., Littman, R.J., Statz, B., Lam, A.K., Dayama, G., Grieneisen, L., Martin, L.S., Flint, J., Eskin, E., Blekhman, R.: Challenges and recommendations to improve the installability and archival stability of omics computational tools. PLoS Biol **17**(6), 3000333 (2019). doi:10.1371/journal.pbio.3000333
53. Sayers, E.W., Barrett, T., Benson, D.A., Bolton, E., Bryant, S.H., Canese, K., Chetvernin, V., Church, D.M., Dicuccio, M., Federhen, S., Feolo, M., Geer, L.Y., Helmberg, W., Kapustin, Y., Landsman, D., Lipman, D.J., Lu, Z., Madden, T.L., Madej, T., Maglott, D.R., Marchler-Bauer, A., Miller, V., Mizrachi, I., Ostell, J., Panchenko, A., Pruitt, K.D., Schuler, G.D., Sequeira, E., Sherry, S.T., Shumway, M., Sirotkin, K., Slotta, D., Souvorov, A., Starchenko, G., Tatusova, T.A., Wagner, L., Wang, Y., John W, W., Yaschenko, E., Ye, J.: Database resources of the national center for biotechnology information. Nucleic Acids Res. **38**(Database issue), 5–16 (2010)
54. Boulesteix, A., Lauer, S., Eugster, M.J.A.: A plea for neutral comparison studies in computational sciences. PLoS One **8**(4), 61562 (2013)
55. Siepel, A.: Challenges in funding and developing genomic software: roots and remedies. Genome biology **20**(1), 1–14 (2019)
56. Larivière, V., Gingras, Y.: The impact factor's Matthew Effect: A natural experiment in bibliometrics. J. Am. Soc. Inf. Sci. **61**(2), 424–427 (2010)
57. Merton, R.K.: The Matthew Effect in Science. Science **159**(3810), 56–63 (1968)
58. Boulesteix, A., Stierle, V., Hapfelmeier, A.: Publication bias in methodological computational research. Cancer Inform. **14**(Suppl 5), 11–19 (2015)
59. Nissen, S.B., Magidson, T., Gross, K., Bergstrom, C.T.: Publication bias and the canonization of false facts. Elife **5**, 21451 (2016)
60. Sterling, T.D., Rosenbaum, W.L., Weinkam, J.J.: Publication decisions revisited: The effect of the outcome of statistical tests on the decision to publish and vice versa. The American Statistician **49**(1), 108–112 (1995)
61. Fanelli, D.: Negative results are disappearing from most disciplines and countries. Scientometrics **90**(3), 891–904 (2012)

bioRxiv preprint doi: https://doi.org/10.1101/092205; this version posted September 2, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

62. Brembs, B.: Reliable novelty: New should not trump true. PLoS Biology **17**(2), 3000117 (2019)
63. McEntyre, J., Lipman, D.: PubMed: bridging the information gap. CMAJ **164**(9), 1317–1319 (2001)
64. Carroll, L.: Alice's Adventures in Wonderland. Macmillan and Co., London (1865)
65. Tolkien, J.R.R.: The Hobbit, Or, There and Back Again. George Allen & Unwin, UK (1937)
66. Mann, H.B., Whitney, D.R.: On a test of whether one of two random variables is stochastically larger than the other. Ann. Math. Stat. **18**(1), 50–60 (1947)
67. Bao, S., Jiang, R., Kwan, W., Wang, B., Ma, X., Song, Y.: Evaluation of next-generation sequencing software in mapping and assembly. J. Hum. Genet. **56**(6), 406–414 (2011)
68. Caboche, S., Audebert, C., Lemoine, Y., Hot, D.: Comparison of mapping algorithms used in high-throughput sequencing: application to ion torrent data. BMC Genomics **15**, 264 (2014)
69. Hatem, A., Bozdağ, D., Toland, A.E., Çatalyürek, Ü.V.: Benchmarking short sequence mapping tools. BMC Bioinformatics **14**, 184 (2013)
70. Schbath, S., Martin, V., Zytnicki, M., Fayolle, J., Loux, V., Gibrat, J.: Mapping reads on a genomic sequence: an algorithmic overview and a practical comparative analysis. J. Comput. Biol. **19**(6), 796–813 (2012)
71. Ruffalo, M., LaFramboise, T., Koyutürk, M.: Comparative analysis of algorithms for next-generation sequencing read alignment. Bioinformatics **27**(20), 2790–2796 (2011)
72. Holtgrewe, M., Emde, A., Weese, D., Reinert, K.: A novel and well-defined benchmarking method for second generation read mapping. BMC Bioinformatics **12**, 210 (2011)