Genome Biology

SHORT REPORT Open Access

Commonly used software tools produce conflicting and overly-optimistic AUPRC values

Wenyu Chen[1†], Chen Miao[1†], Zhenghao Zhang[2], Cathy Sin-Hang Fung[1], Ran Wang[1], Yizhen Chen[2], Yan Qian[3], Lixin Cheng[4], Kevin Y. Yip[2,5*], Stephen Kwok-Wing Tsui[1,6*] and Qin Cao[1,6,7*]

†Wenyu Chen and Chen Miao are co-first authors.

*Correspondence:
kyip@sbpdiscovery.org;
kwtsui@cuhk.edu.hk;
qcao@cuhk.edu.hk

[1] School of Biomedical Sciences, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong SAR, China
[2] Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong SAR, China
[3] The First Affiliated Hospital, Sun Yat-sen University, Guangzhou, China
[4] Shenzhen People's Hospital, First Affiliated Hospital of Southern University of Science and Technology, Second Clinical Medical College of Jinan University, Shenzhen, China
[5] Sanford Burnham Prebys Medical Discovery Institute, La Jolla, CA, USA
[6] Hong Kong Bioinformatics Centre, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong SAR, China
[7] Shenzhen Research Institute, The Chinese University of Hong Kong, Shenzhen, China

Abstract

The precision-recall curve (PRC) and the area under the precision-recall curve (AUPRC) are useful for quantifying classification performance. They are commonly used in situations with imbalanced classes, such as cancer diagnosis and cell type annotation. We evaluate 10 popular tools for plotting PRC and computing AUPRC, which were collectively used in more than 3000 published studies. We find the AUPRC values computed by the tools rank classifiers differently and some tools produce overly-optimistic results.

Introduction

Many problems in computational biology can be formulated as binary classification, in which the goal is to infer whether an entity (e.g., a cell) belongs to a target class (e.g., a cell type). Accuracy, precision, sensitivity (i.e., recall), specificity, and F1 score (Additional file 1: Fig. S1) are some of the measures commonly used to quantify classification performance, but they all require a threshold of the classification score to assign every entity to either the target class or not. The receiver operating characteristic (ROC) and precision-recall curve (PRC) avoid this problem by considering multiple thresholds [1], which allows detailed examination of the trade-off between identifying entities of the target class and wrongly including entities not of this class. It is common to summarize these curves by the area under them (AUROC and AUPRC, respectively), which is a value between 0 and 1, with a larger value corresponding to better classification performance.

When the different classes have imbalanced sizes (e.g., the target cell type has few cells), AUPRC is a more sensitive measure than AUROC [1–4], especially when there are errors among the top predictions (Additional file 1: Fig. S2). As a result, AUPRC has been used in a variety of applications, such as reconstructing biological networks [5], identifying cancer genes [6] and essential genes [7], determining protein binding sites [8], imputing sparse experimental data [9], and predicting patient treatment response [10]. AUPRC has also been extensively used as a performance measure in benchmarking

© The Author(s) 2024. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/. The Creative Commons Public Domain Dedication waiver (http://creativecommons.org/publicdomain/zero/1.0/) applies to the data made available in this article, unless otherwise stated in a credit line to the data.

studies, such as the ones for comparing methods for analyzing differential gene expression [11], identifying gene regulatory interactions [12], and inferring cell-cell communications [13] from single-cell RNA sequencing data.

Given the importance of PRC and AUPRC, we analyzed commonly used software tools and found that they produce contrasting results, some of which are overly-optimistic.

Results

Basics

For each entity, a classifier outputs a score to indicate how likely it belongs to the target (i.e., "positive") class. Depending on the classifier, the score can be discrete (e.g., random forest) or continuous (e.g., artificial neural network). Using a threshold t, the classification scores can be turned into binary predictions by considering all entities with a score $\geq t$ as belonging to the positive class and all other entities as not. When these predictions are compared to the actual classes of the entities, precision is defined as the proportion of entities predicted to be positive that are actually positive, while recall is defined as the proportion of actually positive entities that are predicted to be positive (Additional file 1: Fig. S1).

The PRC is a curve that shows how precision changes with recall. In the most common way to produce the PRC, each unique classification score observed is used as a threshold to compute a pair of precision and recall values, which forms an anchor point on the PRC. Adjacent anchor points are then connected to produce the PRC.

When no two entities have the same score (Fig. 1a), it is common to connect adjacent anchor points directly by a straight line [14–19] (Fig. 1b). Another method uses an expectation formula, which we will explain below, to connect discrete points by piecewise linear lines [20] (Fig. 1c). The third method is to use the same expectation formula to produce a continuous curve between adjacent anchor points [17, 21] (Fig. 1d). A fourth method that has gained popularity, known as Average Precision (AP), connects adjacent anchor points by step curves [15, 19, 22, 23] (Fig. 1e). In all four cases, PRC estimates a function of precision in terms of recall based on the observed classification scores of the entities, and AUPRC estimates the integral of this function using trapezoids (in the direct straight line case), interpolation lines/curves (in the expectation cases), or rectangles (in the AP case).

When there are ties with multiple entities having the same score, which happens more easily with classifiers that produce discrete scores, these entities together define only one anchor point (Fig. 1f). There are again four common methods for connecting such an anchor point to the previous anchor point, which correspond to the four methods for connecting anchor points when there are no ties (details in Additional file 1: Supplementary Text). The first method is to connect the two anchor points by a straight line [15, 18, 19] (Fig. 1g). This method is known to easily produce overly-optimistic AUPRC values [2, 24], which we will explain below. The second method is to interpolate additional points between the two anchor points using a non-linear function and then connect the points by straight lines [14, 17, 20] (Fig. 1h). The interpolated points appear at their expected coordinates under the assumption that all possible orders of the entities with the same score have equal probability. The third method uses the same interpolation formula as the second method but instead of creating a finite number of interpolated points, it connects the two anchor points by a a

Anchor point	Classification score	Actual class	TP	FP	Recall	Precision
A	0.9	1	1	0	0.5	1
B	0.8	0	1	1	0.5	0.5
C	0.7	1	2	1	1	0.667
D	0.6	0	2	2	1	0.5
E	0.5	0	2	3	1	0.4

f

Anchor point	Classification score	Actual class	TP	FP	Recall	Precision
A	0.7	1	2	1	0.333	0.667
	0.7	0				
	0.7	1				
B	0.5	1	5	5	0.833	0.5
	0.5	1				
	0.5	0				
	0.5	0				
	0.5	1				
	0.5	0				
	0.5	0				
C	0.4	0	5	7	0.833	0.417
	0.4	0				
D	0.3	1	6	7	1	0.462

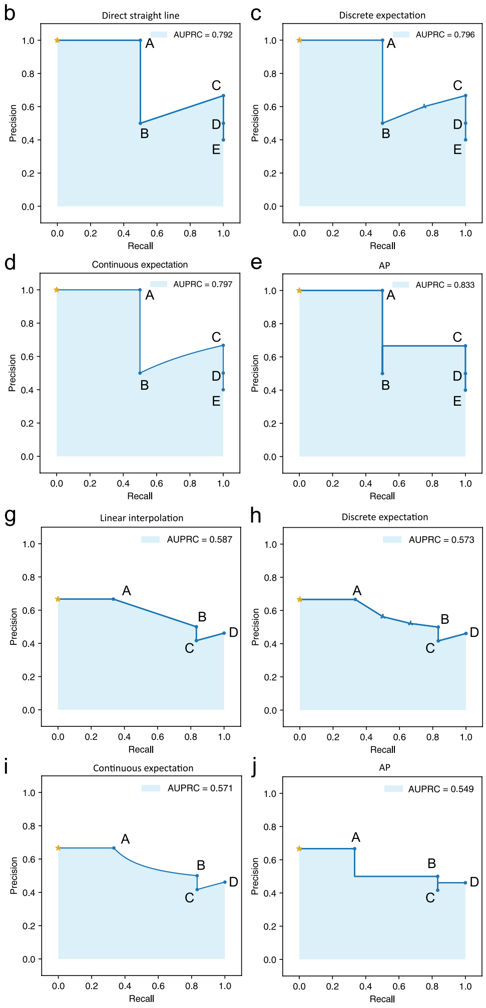


Fig. 1 Different methods for connecting adjacent anchor points on the PRC. **a** An illustrative data set with no two entities receiving the same classification score. **b**–**e** Different methods for connecting adjacent anchor points when there are no ties in classification scores, namely **b** direct straight line, **c** discrete expectation, **d** continuous expectation, and **e** AP. **f** An illustrative data set with different entities receiving the same classification score. Each group of entities with the same classification score defines a single anchor point (A, B, C, and D, from 3, 7, 2, and 1 entities, respectively). **g**–**j** Different methods for connecting anchor point B to its previous anchor point, A, namely **g** linear interpolation, **h** discrete expectation, **i** continuous expectation, and **j** AP. In **c** and **h**, *tp* is set to 0.5 and 1 in Formula 1, respectively (Additional file 1: Supplementary Text)

continuous curve [17, 21] (Fig. 1i). Finally, the fourth method comes naturally from the AP approach, which uses step curves to connect the anchor points [15, 19, 22, 23] (Fig. 1j).

Using the four methods to connect anchor points when there are no ties and the four methods when there are ties can lead to very different AUPRC values (Fig. 1, Additional file 1: Fig. S3 and Supplementary Text).

Conceptual and implementation issues of some popular software tools

We analyzed 10 tools commonly used to produce PRC and AUPRC (Additional file 1: Table S1). Based on citations and keywords, we estimated that these tools have been used in more than 3000 published studies in total (Methods).

The 10 tools use different methods to connect anchor points on the PRC and therefore they can produce different AUPRC values (Table 1, Additional file 1: Fig. S4–S7 and Supplementary Text). As a comparison, all 10 tools can also compute AUROC, and we found most of them to produce identical values (Additional file 1: Supplementary Text).

We found five conceptual issues with some of these tools when computing AUPRC values (Table 1):

① Using the linear interpolation method to handle ties, which can produce overly-optimistic AUPRC values [2, 24]. When interpolating between two anchor points, linear interpolation produces higher AUPRC than the other three methods under conditions that can easily happen in real situations (Additional file 1: Supplementary Text)

② Always using (0, 1) as the starting point of the PRC (procedurally produced or conceptually derived, same for ③ and ⑤ below), which is inconsistent with the concepts behind the AP and non-linear expectation methods when the first anchor point with a non-zero recall does not have a precision of one (Additional file 1: Supplementary Text)

③ Not producing a complete PRC that covers the full range of recall values from zero to one

④ Ordering entities with the same classification score by their order in the input and then handling them as if they have distinct classification scores

⑤ Not putting all anchor points on the PRC

These issues can lead to overly-optimistic AUPRC values or change the order of two AUPRC values (Additional file 1: Supplementary Text and Fig. S8-S13).

Some of these tools also produce a visualization of the PRC. We found three types of issues with these visualizations (Table 1):

Table 1 Methods used by the different software tools to connect anchor points and issues found in their calculation of AUPRC and construction of the PRC. For tools that can connect anchor points in multiple ways, we show each of them in a separate row. The AUPRC and PRC issues are defined in the text and detailed in Additional file 1: Supplementary Text. "—" means no issues found. [a] PerfMeas orders entities with the same classification score by their order in the input and then defines anchor points as if there are no ties. [b] The source code of TorchEval states that it uses Riemann integral to compute AUPRC, which is equivalent to AP

Tool	**Anchor point connection**		**AUPRC issues**	**PRC issues**
	Without ties	**With ties**		
ROCR	Direct straight line	Discrete expectation	②⑤⚠	Ⅱ
Weka	AP	AP	—	Ⅱ
scikit-learn	Direct straight line	Linear interpolation	①②	(No visualization)
	AP	AP	—	Ⅲ
PerfMeas	Direct straight line	Direct straight line[a]	③④	Ⅰ
PRROC	Direct straight line	Discrete expectation	—	⚠
	Continuous expectation	Continuous expectation	—	—
TensorFlow	Continuous expectation	Continuous expectation	⚠	(No visualization)
precrec	Discrete expectation	Discrete expectation	—	—
TorchEval	AP[b]	AP	—	(No visualization)
MLeval	Direct straight line	Linear interpolation	①③	Ⅰ
yardstick	Direct straight line	Linear interpolation	①②	Ⅰ
	AP	AP	—	(No visualization)

I. Producing a visualization of PRC that has the same issue(s) as in the calculation of AUPRC

II. Producing a PRC visualization that does not always start the curve at a point with zero recall

III. Producing a PRC visualization that always starts at (0, 1)

Finally, we also found some programming bugs and noticed that some tools require special attention for correct usage (both marked by ⚠ in Table 1).

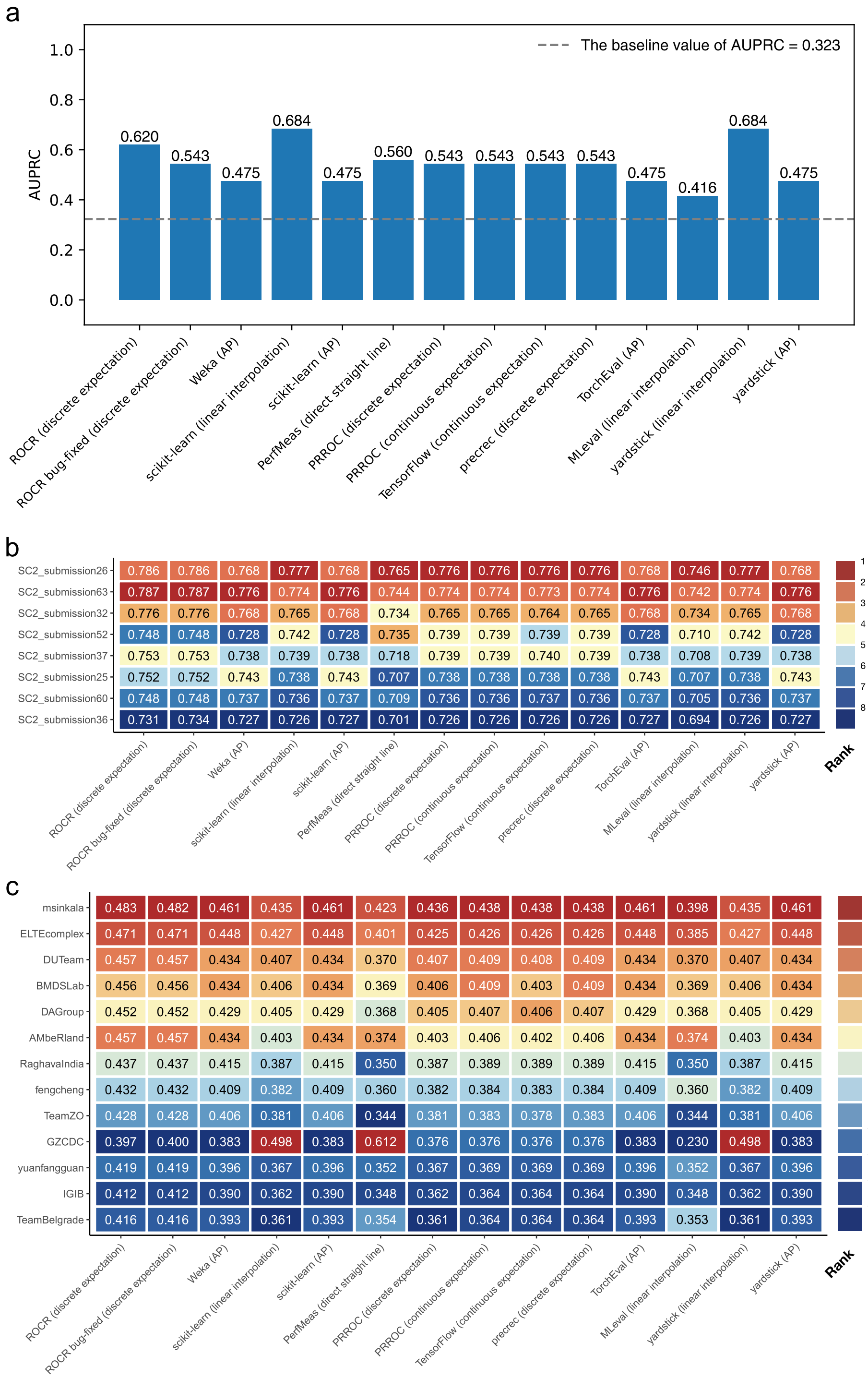
Inconsistent AUPRC values and contrasting classifier ranks produced by the popular tools

To see how the use of different methods by the 10 tools and their other issues affect PRC analysis in practice, we applied them to evaluate classifiers in four realistic scenarios.

In the first scenario, we analyzed data from a COVID-19 study [25] in which patient blood samples were subjected to Cellular Indexing of Transcriptomes and Epitopes by Sequencing (CITE-seq) assays [26]. We constructed a classifier for predicting CD4$^+$ T cells, which groups the cells based on their transcriptome data alone and assigns a single cell type label to each group. Using cell type labels defined by the original authors as reference, which were obtained using both antibody-derived tags (ADTs) and transcriptome data, we computed the AUPRC of the classifier. Figure 2a shows that the 10 tools produced 6 different AUPRC values, ranging from 0.416 to 0.684. In line with the conceptual discussions above, the AP method generally produced the smallest AUPRC values while the linear interpolation method generally produced the largest, although individual issues of the tools created additional variations of the AUPRC values computed.

In the second scenario, we compared the performance of different classifiers that predict whether a patient has the ulcerative colitis (UC) subtype of inflammatory bowel disease (IBD) or does not have IBD, based on metagenomic data (processed taxonomy-based profile) [27]. The predictions made by these classifiers were submitted to the sbv IMPROVER Metagenomics Diagnosis for IBD Challenge. Their performance was determined by comparing against diagnosis of these patients based on clinical, endoscopic, and histological criteria. Figure 2b shows that based on the AUPRC values computed, the 10 tools ranked the classifiers differently. For example, among the top 8 submissions with the highest performance, the classifier in submission 26 was ranked first in 8 cases, sole second place in 2 cases, and tied second place with another classifier in 4 cases (Fig. 2b and Additional file 1: Fig. S14). We observed similar rank flips when considering the top 30 submissions (Additional file 1: Fig. S15 and S16).

In the third scenario, we compared the performance of different classifiers in identifying preterm prelabor rupture of the membranes (PPROM) cases from normal pregnancy in the DREAM Preterm Birth Prediction Challenge [28]. Based on the AUPRC values produced by the 10 tools, the 13 participating teams were ranked very differently (Fig. 2c and Additional file 1: Fig. S17). For example, Team "GZCDC" was ranked first (i.e., highest) in 3 cases, tenth in 4 cases, and thirteenth (i.e., lowest) in 7 cases. In addition to differences in the ranks, some of the AUPRC values themselves are also very different. For example, the AUPRC values computed by PerfMeas and MLeval have a Pearson correlation of -0.759.



	ROCR (discrete expectation)	ROCR bug-fixed (discrete expectation)	Weka (AP)	scikit-learn (linear interpolation)	scikit-learn (AP)	PerfMeas (direct straight line)	PRROC (discrete expectation)	PRROC (continuous expectation)	TensorFlow (continuous expectation)	precrec (discrete expectation)	TorchEval (AP)	MLeval (linear interpolation)	yardstick (linear interpolation)	yardstick (AP)
SC2_submission26	0.786	0.786	0.768	0.777	0.768	0.765	0.776	0.776	0.776	0.776	0.768	0.746	0.777	0.768
SC2_submission63	0.787	0.787	0.776	0.774	0.776	0.744	0.774	0.774	0.773	0.774	0.776	0.742	0.774	0.776
SC2_submission32	0.776	0.776	0.768	0.765	0.768	0.734	0.765	0.765	0.764	0.765	0.768	0.734	0.765	0.768
SC2_submission52	0.748	0.748	0.728	0.742	0.728	0.735	0.739	0.739	0.739	0.739	0.728	0.710	0.742	0.728
SC2_submission37	0.753	0.753	0.738	0.739	0.738	0.718	0.739	0.739	0.740	0.739	0.738	0.708	0.739	0.738
SC2_submission25	0.752	0.752	0.743	0.738	0.743	0.707	0.738	0.738	0.738	0.738	0.743	0.707	0.738	0.743
SC2_submission60	0.748	0.748	0.737	0.736	0.737	0.709	0.736	0.736	0.737	0.736	0.737	0.705	0.736	0.737
SC2_submission36	0.731	0.734	0.727	0.726	0.727	0.701	0.726	0.726	0.726	0.726	0.727	0.694	0.726	0.727

	ROCR (discrete expectation)	ROCR bug-fixed (discrete expectation)	Weka (AP)	scikit-learn (linear interpolation)	scikit-learn (AP)	PerfMeas (direct straight line)	PRROC (discrete expectation)	PRROC (continuous expectation)	TensorFlow (continuous expectation)	precrec (discrete expectation)	TorchEval (AP)	MLeval (linear interpolation)	yardstick (linear interpolation)	yardstick (AP)
msinkala	0.483	0.482	0.461	0.435	0.461	0.423	0.436	0.438	0.438	0.438	0.461	0.398	0.435	0.461
ELTEcomplex	0.471	0.471	0.448	0.427	0.448	0.401	0.425	0.426	0.426	0.426	0.448	0.385	0.427	0.448
DUTeam	0.457	0.457	0.434	0.407	0.434	0.370	0.407	0.409	0.408	0.409	0.434	0.370	0.407	0.434
BMDSLab	0.456	0.456	0.434	0.406	0.434	0.369	0.406	0.409	0.403	0.409	0.434	0.369	0.406	0.434
DAGroup	0.452	0.452	0.429	0.405	0.429	0.368	0.405	0.407	0.406	0.407	0.429	0.368	0.405	0.429
AMbeRland	0.457	0.457	0.434	0.403	0.434	0.374	0.403	0.406	0.402	0.406	0.434	0.374	0.403	0.434
RaghavaIndia	0.437	0.437	0.415	0.387	0.415	0.350	0.387	0.389	0.389	0.389	0.415	0.350	0.387	0.415
fengcheng	0.432	0.432	0.409	0.382	0.409	0.360	0.382	0.384	0.383	0.384	0.409	0.360	0.382	0.409
TeamZO	0.428	0.428	0.406	0.381	0.406	0.344	0.381	0.383	0.378	0.383	0.406	0.344	0.381	0.406
GZCDC	0.397	0.400	0.383	0.498	0.383	0.612	0.376	0.376	0.376	0.376	0.383	0.230	0.498	0.383
yuanfangguan	0.419	0.419	0.396	0.367	0.396	0.352	0.367	0.369	0.369	0.369	0.396	0.352	0.367	0.396
IGIB	0.412	0.412	0.390	0.362	0.390	0.348	0.362	0.364	0.364	0.364	0.390	0.348	0.362	0.390
TeamBelgrade	0.416	0.416	0.393	0.361	0.393	0.354	0.361	0.364	0.364	0.364	0.393	0.353	0.361	0.393

Fig. 2 The AUPRC values computed by the 10 tools in several realistic scenarios. **a** Predicting CD4^{+} T cells from single-cell transcriptomic data. **b** Predicting inflammatory bowel disease cases that belong to the ulcerative colitis subtype in the sbv IMPROVER Metagenomics Diagnosis for Inflammatory Bowel Disease Challenge. Only the top 8 submissions according to PRROC (discrete expectation) AUPRC values are included. **c** Predicting cases with preterm prelabor rupture of membranes in the DREAM Preterm Birth Prediction Challenge. In **b** and **c**, each entry shows the AUPRC value and the background color indicates its rank among the competitors

In the fourth scenario, we compared 29 classifiers that predicted target genes of transcription factors in the DREAM5 challenge [29]. Again, some classifiers received very different ranking based on the AUPRC values computed by the different tools (Additional file 1: Fig. S18 and S19). For example, the classifier named "Other4" was ranked second based on the AUPRC values computed by PerfMeas but it was ranked twenty-fifth based on the AUPRC values computed by MLeval. In general, tools that use the discrete expectation, continuous expectation, and AP methods are in good agreements in this scenario, but they differ substantially from tools that use the linear interpolation method.

Conclusions

Due to their highly technical nature, it is easy to overlook the inconsistencies and issues of the software tools used for producing PRC and AUPRC. Some possible consequences include reporting overly-optimistic AUPRC, ranking classifiers differently by different tools, and introducing biases to the evaluation process, such as inflating the AUPRC of classifiers that produce discrete scores.

To address the problems, it is crucial to use tools that are free of the bugs described and avoid using the linear interpolation method (Table 1). It is also necessary to state clearly in manuscripts both the tool used (with its version number) and the underlying methods implemented by the tool for producing PRC and AUPRC. Whenever feasible, the adoption of multiple tools that implement different methods (e.g., one based on continuous expectation and one based on AP) is recommended, with comprehensive reporting of all their results.

Methods

Information about the tools

In this study, we included 12 tools commonly used for PRC and ROC analyses (Additional file 1: Table S1). For each tool, we analyzed the latest stable version of it as of August 15, 2023. Because TorchEval had not released a stable version, we analyzed the latest version of it, version 0.0.6. Among the 12 tools, ten can compute both AUROC and AUPRC, while the remaining two can only compute AUROC. We focused on these 10 tools in the study of PRC and AUPRC. Some tools provide multiple methods for computing AUROC/AUPRC.

For tools with an associated publication, we obtained its citation count from Google Scholar. If a tool has multiple associated publications, we selected the one with the largest number of citations. As a result, the citation counts we report in Additional file 1: Table S1 are underestimates if different publications associated with the same tool are not always cited together.

The Comprehensive R Archive Network (CRAN) packages PerfMeas and MLeval did not have an associated formal publication but only release notes. In each of these cases, we used the package name as keyword to search on Google Scholar and then manually checked the publications returned to determine the number of publications that cited these packages.

The CRAN package yardstick also did not have an associated formal publication. However, we were not able to use the same strategy as PerMeas and MLeval to determine the number of publications that cited the yardstick package since “yardstick” is an English word and the search returned too many publications to be verified manually. Therefore, we only counted the number of publications that cited yardstick’s release note, which is likely an underestimate of the number of publications that cited yardstick.

All citation counts were collected on October 9, 2023.

For tools with an associated formal publication, based on our collected lists of publications citing the tools, we further estimated the number of times the tools were actually used in the studies by performing keyword-based filtering. Specifically, if the main text or figure captions of a publication contains either one of the keywords “AUC” and “AUROC,” we assumed that the tool was used in that published study to perform ROC analysis. In the case of PRC, we performed filtering in two different ways and reported both sets of results in Additional file 1: Table S1. In the first way, we assumed a tool was used in a published study if the main text or figure captions of the publication contains any one of the following keywords: “AUPR,” “AU-PR,” “AUPRC,” “AU-PRC,” “AUCPR,” “AUC-PR,” “PRAUC,” “PR-AUC,” “area under the precision recall,” and “area under precision recall.” In the second way, we assumed a tool was used in a published study if the main text or figure captions of the publication contains both “area under” and “precision recall.”

For the CRAN packages PerfMeas and MLeval, we estimated the number of published studies that actually used them by searching Google Scholar using the above three keyword sets each with the package name appended. We found that for all the publications we considered as using the packages in this way, they were also on our lists of publications that cite these packages. We used the same strategy to identify published studies that used the CRAN package yardstick. We found that some of these publications were not on our original list of publications that cite yardstick, and therefore we added them to the list and updated the citation count accordingly.

TorchEval was officially embedded into PyTorch in 2022. Due to its short history, among the publications that cite the PyTorch publication, we could not find any of them that used the TorchEval library.

Data collection and processing

We used four realistic scenarios to illustrate the issues of the AUPRC calculations.

In the first scenario, we downloaded CITE-seq data produced from COVID-19 patient blood samples by the COVID-19 Multi-Omic Blood ATlas (COMBAT) consortium [25]. We downloaded the data from Zenodo [30] and used the data in the “COMBAT-CITESeq-DATA” archive in this study. We then used a standard procedure to cluster the cells based on the transcriptome data and identified CD4$^+$ T cells. Specifically, we extracted the raw count matrix of the transcriptome data and ADT features (“X” object) and the annotation data frame (“obs” object) from the H5AD file. We dropped all ADT features (features with names starting with “AB-”) and put the transcriptome data along with the annotation data frame into Seurat (version 4.1.1). We then log-normalized the transcriptome data (method “NormalizeData(),” default parameters), identified highly-variable genes (method “FindVariableFeatures(),” number of variable genes set to 10,000), scaled the data (method “ScaleData(),” default parameters), performed principal component analysis (method “RunPCA(),” number of principal components set to 50), constructed the shared/k-nearest neighbor (SNN/kNN) graph (method "FindNeighbours()," default parameters), and performed Louvain clustering of the cells (method "FindClusters()," default parameters). We then extracted the clustering labels generated and concatenated them with cell type, major subtype, and minor subtype annotations provided by the original authors, which were manually curated using both ADT and transcriptome information.

Our procedure produced 29 clusters, which contained 836,148 cells in total. To mimic a classifier that predicts CD4$^+$ T cells using the transcriptome data alone, we selected one cluster and "predicted" all cells in it as CD4$^+$ T cells and all cells in the other 28 clusters as not, based on which we computed an AUPRC value by comparing these "predictions" with the original authors' annotations. We repeated this process for each of the 29 clusters in turn, and chose the one that gave the highest AUPRC as the final cluster of predicted CD4$^+$ T cells.

For the second scenario, we obtained the data set used in the sbv IMPROVER (Systems Biology Verification combined with Industrial Methodology for PROcess VErification in Research) challenge on inflammatory bowel disease diagnosis based on metagenomics data [27]. The challenge involved 12 different tasks, and we focused on the task of identifying UC samples from non-IBD samples using the processed taxonomy-based profile as features. The data set contained 32 UC samples and 42 non-IBD samples, and therefore the baseline AUPRC was $\frac{32}{32+42} = 0.432$. There were 60 submissions in total, which used a variety of classifiers. We obtained the classification scores in the submissions from Supplementary Information 4 of the original publication [27]. When we extracted the classification scores of each submission, we put the actual positive entities before the actual negative entities. This ordering did not affect the AUPRC calculations of most tools except those of PerfMeas, which depend on the input order of the entities with the same classification score.

To see how the different tools rank the top submissions, we first computed the AUPRC of each submission using PRROC (option that uses the discrete expectation method to handle ties) since we did not find any issues with its AUPRC calculations (Table 1). We then analyzed the AUPRC values produced by the 10 tools based on either the top 8 (Fig. 2b and Additional file 1: Fig. S14) or top 30 (Additional file 1: Fig. S15 and S16) submissions.

For the third scenario, we downloaded the data set used in the Dialogue on Reverse Engineering Assessment and Methods (DREAM) Preterm Birth Prediction Challenge [28] from https://www.synapse.org/#!Synapse:syn22127152. We collected the classification scores, from the object "prpile" in each team's RData file, and the actual classes produced based on clinical evidence, from "anoSC2_v21_withkey.RData" (https://www.synapse.org/#!Synapse:syn22127343). The challenge contained 7 scenarios, each of which had 2 binary classification tasks. For each scenario, 10 different partitioning of the data into training and testing sets were provided. We focused on the task of identifying PPROM cases from the controls under the D2 scenario defined by the challenge. For this task, the baseline AUPRC value averaged across the 10 testing sets was 0.386. There were 13 participating teams in total. For each team, we extracted its classification scores and placed the actual positive entities before the actual negative entities. For submissions that contained negative classification scores, we re-scaled all the scores to the range between 0 and 1 without changing their order since TensorFlow expects all classification scores to be between zero and one (Additional file 1: Supplementary Text). Finally, for each team, we computed its AUPRC using each of the 10 testing sets and reported their average. We note that the results we obtained by using PRROC (option that uses the continuous expectation method to handle ties) were identical to those reported by the challenge organizer.

For the fourth scenario, we obtained the data set used in the DREAM5 challenge on reconstructing transcription factor-target networks based on gene expression data [29]. The challenge included multiple networks and we focused on the E. coli in silico Network 1, which has a structure that corresponds to the real E. coli transcriptional regulatory network [29]. We obtained the data from Supplementary Data of the original publication [29]. There were 29 submissions in total. For each submission, we extracted the classification scores of the predicted node pairs (each pair involves one potential transcription factor and one gene it potentially regulates) from Supplementary Data 3 and compared them with the actual classes (positive if the transcription factor actually regulates the gene; negative if not) in the gold-standard network from Supplementary Data 1. Both the submissions and the gold-standard were not required to include all node pairs. To handle this, we excluded all node pairs in a submission that were not included in the gold-standard (because we could not judge whether they are actual positives or actual negatives), and assigned a classification score of 0 to all node pairs in the gold-standard that were not included in a submission (because the submission did not give a classification score to them). The gold-standard contained 4012 interacting node pairs and 274,380 non-interacting node pairs, and therefore the baseline AUPRC value was $\frac{4012}{4012+274380} = 0.014$.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13059-024-03266-y.

Additional file 1. Supplementary text, figures and tables.

Additional file 2. Review history.

Review history

The review history is available as Additional file 2.

Peer review information

Andrew Cosgrove was the primary editor of this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.

Authors' contributions

KYY, SKWT, and QC conceived and supervised the project. WC, CM, KYY, and QC designed the computational experiments and data analyses. WC, CM, and YC surveyed and collected the tools. WC, CM, ZZ, CSHF, and RW prepared the data. WC and CM conducted the computational experiments. WC, CM, ZZ, and RW performed the data analyses. All the authors interpreted the results. WC, CM, KYY, and QC wrote the manuscript.

Funding

QC is supported by National Natural Science Foundation of China under Award Number 32100515. YQ is supported by Guangdong Basic and Applied Basic Research Foundation under Award Number 2021A1515110759 and Natural Science Foundation of Guangdong Province of China under Award Number 2023A1515011187. LC is supported by National Natural Science Foundation of China under Award Number 32370711 and Shenzhen Medical Research Fund under Award Number A2303033. KYY is supported by National Cancer Institute of the National Institutes of Health under Award Number P30CA030199, National Institute on Aging of the National Institutes of Health under Award Numbers U54AG079758 and R01AG085498, and internal grants of Sanford Burnham Prebys Medical Discovery Institute. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

Availability of data and materials

Our code is written in Java, Python and R. The reproducible code and all the data used in this paper are available at GitHub [31] and Zenodo [32].

Declarations

Ethics approval and consent to participate

No ethical approval was required for this study. All utilized public datasets were generated by other organizations that obtained ethical approval.

Competing interests

The authors declare that they have no competing interests.

Received: 7 February 2024 Accepted: 30 April 2024
Published online: 13 May 2024

References

1. Saito T, Rehmsmeier M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLoS ONE. 2015;10(3):e0118432.
2. Davis J, Goadrich M. The relationship between precision-recall and ROC curves. In: Proceedings of the 23rd International Conference on Machine Learning. New York: Association for Computing Machinery; 2006. p. 233–40.
3. He H, Garcia EA. Learning from imbalanced data. IEEE Trans Knowl Data Eng. 2009;21(9):1263–84.
4. Lichtnwalter R, Chawla NV. Link prediction: fair and effective evaluation. In: 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. New York: IEEE; 2012. p. 376–83.
5. Cao Q, Anyansi C, Hu X, Xu L, Xiong L, Tang W, et al. Reconstruction of enhancer-target networks in 935 samples of human primary cells, tissues and cell lines. Nat Genet. 2017;49(10):1428–36.
6. Schulte-Sasse R, Budach S, Hnisz D, Marsico A. Integration of multiomics data with graph convolutional networks to identify new cancer genes and their associated molecular mechanisms. Nat Mach Intell. 2021;3(6):513–26.
7. Hong C, Cao Q, Zhang Z, Tsui SKW, Yip KY. Reusability report: Capturing properties of biological objects and their relationships using graph neural networks. Nat Mach Intell. 2022;4(3):222–6.
8. Sielemann J, Wulf D, Schmidt R, Bräutigam A. Local DNA shape is a general principle of transcription factor binding specificity in Arabidopsis thaliana. Nat Commun. 2021;12(1):6549.
9. Li Z, Kuppe C, Ziegler S, Cheng M, Kabgani N, Menzel S, et al. Chromatin-accessibility estimation from single-cell ATAC-Seq data with scOpen. Nat Commun. 2021;12(1):6386.
10. Chowell D, Yoo SK, Valero C, Pastore A, Krishna C, Lee M, et al. Improved prediction of immune checkpoint blockade efficacy across multiple cancer types. Nat Biotechnol. 2022;40(4):499–506.
11. Dal Molin A, Baruzzo G, Di Camillo B. Single-cell RNA-sequencing: assessment of differential expression analysis methods. Front Genet. 2017;8:62.
12. Pratapa A, Jalihal AP, Law JN, Bharadwaj A, Murali T. Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. Nat Methods. 2020;17(2):147–54.
13. Dimitrov D, Türei D, Garrido-Rodriguez M, Burmedi PL, Nagai JS, Boys C, et al. Comparison of methods and resources for cell-cell communication inference from single-cell RNA-Seq data. Nat Commun. 2022;13(1):3224.
14. Sing T, Sander O, Beerenwinkel N, Lengauer T. ROCR: visualizing classifier performance in R. Bioinformatics. 2005;21(20):3940–1.
15. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: machine learning in Python. J Mach Learn Res. 2011;12:2825–30.
16. Valentini G, Re M. PerfMeas: performance measures for ranking and classification tasks. R package. 2014. https://cran.r-project.org/web/packages/PerfMeas.
17. Grau J, Grosse I, Keilwagen J. PRROC: computing and visualizing precision-recall and receiver operating characteristic curves in R. Bioinformatics. 2015;31(15):2595–7.
18. John C. MLeval: machine learning model evaluation. R package. 2020. https://cran.r-project.org/web/packages/MLeval.
19. Kuhn M, Vaughan D. yardstick: tidy characterizations of model performance. R package. 2021. https://cran.r-project.org/web/packages/yardstick.
20. Saito T, Rehmsmeier M. Precrec: fast and accurate precision-recall and ROC curve calculations in R. Bioinformatics. 2017;33(1):145–7.
21. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, et al. TensorFlow: large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:160304467. 2016. https://arxiv.org/abs/1603.04467.
22. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA data mining software: an update. ACM SIGKDD Explor Newsl. 2009;11(1):10–8.
23. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: an imperative style, high-performance deep learning library. Adv Neural Inf Process Syst. 2019;32:8026–37.
24. Flach P, Kull M. Precision-recall-gain curves: PR analysis done right. Adv Neural Inf Process Syst. 2015;28:838–46.
25. COvid-19 Multi-omics Blood ATlas (COMBAT) Consortium. A blood atlas of COVID-19 defines hallmarks of disease severity and specificity. Cell. 2022;185(5):916–938.e58.
26. Stoeckius M, Hafemeister C, Stephenson W, Houck-Loomis B, Chattopadhyay PK, Swerdlow H, et al. Simultaneous epitope and transcriptome measurement in single cells. Nat Methods. 2017;14(9):865–8.

27. Khachatryan L, Xiang Y, Ivanov A, Glaab E, Graham G, Granata I, et al. Results and lessons learned from the sbv IMPROVER metagenomics diagnostics for inflammatory bowel disease challenge. Sci Rep. 2023;13:6303.
28. Tarca AL, Pataki BÁ, Romero R, Sirota M, Guan Y, Kutum R, et al. Crowdsourcing assessment of maternal blood multi-omics for predicting gestational age and preterm birth. Cell Rep Med. 2021;2(6):100323.
29. Marbach D, Costello JC, Küffner R, Vega NM, Prill RJ, Camacho DM, et al. Wisdom of crowds for robust gene network inference. Nat Methods. 2012;9(8):796–804.
30. COvid-19 Multi-omics Blood ATlas (COMBAT) Consortium. A blood atlas of COVID-19 defines hallmarks of disease severity and specificity: Associated data. Zenodo. 2022. https://doi.org/10.5281/zenodo.6120249.
31. Chen W, Miao C, Zhang Z, Fung CSH, Wang R, Chen Y, et al. Commonly used software tools produce conflicting and overly-optimistic AUPRC values. GitHub. 2024. https://github.com/wychencuhk/AUPRC_project.
32. Chen W, Miao C, Zhang Z, Fung CSH, Wang R, Chen Y, et al. Commonly used software tools produce conflicting and overly-optimistic AUPRC values. Zenodo. 2024. https://doi.org/10.5281/zenodo.11076192.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.