

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/342547853>

Recommendations for machine learning validation in biology

Preprint · June 2020

CITATIONS

0

READS

4,842

8 authors, including:



Ian Walsh

Bioprocessing Technology Institute

71 PUBLICATIONS 3,097 CITATIONS

SEE PROFILE



Tiina Titma

Tallinn University of Technology

18 PUBLICATIONS 1,326 CITATIONS

SEE PROFILE



Fotis E. Psomopoulos

The Centre for Research and Technology, Hellas

155 PUBLICATIONS 1,631 CITATIONS

SEE PROFILE



Silvio Tosatto

University of Padova

302 PUBLICATIONS 21,620 CITATIONS

SEE PROFILE

Recommendations for machine learning validation in biology

Ian Walsh^{1,*}, Dmytro Fishman^{2,*}, Dario Garcia-Gasulla³, Tiina Titma⁴, The ELIXIR Machine Learning focus group[#], Jen Harrow^{5,+}, Fotis E. Psomopoulos^{6,+} & Silvio C.E. Tosatto^{7,+}

¹ Bioprocessing Technology Institute, Agency for Science, Technology and Research, Singapore, ² Institute of Computer Science, University of Tartu, Estonia, ³ Barcelona Supercomputing Center (BSC), Barcelona, Spain, ⁴ School of Information Technologies, Tallinn University of Technology, Estonia, ⁵ ELIXIR HUB, South building, Wellcome Genome Campus, Hinxton, Cambridge, UK, ⁶ Institute of Applied Biosciences, ⁷ Centre for Research and Technology Hellas, Thessaloniki, Greece, ⁷ Dept. of Biomedical Sciences, University of Padua, Padua, Italy.

^{*}contributed equally

[#]see list of co-authors at the end of the manuscript

⁺corresponding authors

Abstract

Modern biology frequently relies on machine learning to provide predictions and improve decision processes. There have been recent calls for more scrutiny on machine learning performance and possible limitations. Here we present a set of community-wide recommendations aiming to help establish standards of machine learning validation in biology. Adopting a structured methods description for machine learning based on DOME (data, optimization, model, evaluation) will allow both reviewers and readers to better understand and assess the performance and limitations of a method or outcome. The recommendations are complemented by a machine learning summary table which can be easily included in the supplementary material of published papers.

Introduction

With the steep decline in the cost of high-throughput technologies, large amounts of biological data are being generated and made accessible to researchers. Analyzing this highly complex and voluminous data is not trivial, while the use of classical statistics is not enough to explore their full potential. Machine learning (ML) has thus been brought into the spotlight as a very useful approach to understand cellular¹, genomic², proteomic³, post-translational⁴, metabolic⁵ and drug discovery data⁶ with the potential to result in ground-breaking medical applications^{7,8} (Supp. Figure 1). This is clearly reflected in the corresponding growth of ML publications (Figure 1), reporting a wide range of modelling techniques in biology. However, this sharp increase in publications inherently requires a corresponding increase in the number and depth of experts' reviews that can offer critical assessment⁹ and improve reproducibility¹⁰. Consequently, there are ML applications published and used today that contain substantial flaws often producing true prediction performances worse than claimed or even worse than random estimates^{11,12}.

A complete set of standard guidelines and best practices is still a matter of debate. However, some concerns that arise in ML are universally agreed. For example, in a sample of over 400 papers submitted at major AI conferences only 6% shared their code, one third shared their data, and around half shared pseudocode¹³. This lack of availability is resulting in a reproducibility crisis as critical components of ML models are not shared with the community^{13–15}. Perhaps, the biggest reason for the lack of sharing is the vast commercial value of useful ML models¹⁶ and their potential impact. Data characteristics such as train/test set independence, size, distribution and quality are often under-reported¹⁷. Recent deep learning ML approaches are suited to extract knowledge from large volumes of data but depend on underlying

data quality¹⁸. Conversely, being able to generalise from small training sets is also receiving interest recently¹⁹. Data quality may be affected by unbalance in terms of classes, poor quality or noisy observations, or may not be released to the public. Furthermore, biological data often offers only a partial view of the problem at hand, which can easily introduce unrecognized biases on. It is therefore necessary to perform rigorous quality checks before using ML. The interpretability of predictions also depends on the ML models being used, with a prevalence of back box results²⁰. Equally important is a reliable evaluation of the model on data which is completely unrelated to the data used in training in order to assess the final performance. Without a correct benchmark, ML models simply cannot be used for biological inference with guarantees. In practice, implementing ML models that can generalize well on unseen biological data is a challenging task that requires both significant proficiency in ML and biology¹⁷.

A recent high-profile citation abuse case involving a scientist publishing ML methods has highlighted how the review process can be easily distorted at the moment for personal benefit²¹. Additionally, just last year another article in Nature Reviews Molecular Cell Biology highlighted the need for stricter standards in ML²², arguing for the adoption of on-submission checklists⁹ as a first step towards improving publication standards. In this direction, a universal checklist for ML on biological data is presented in Table 1 outlining four broad topics with recommended actions. We have focused on Data, Optimization, Model and Evaluation (DOME) as we found most errors in ML protocols occur in these. To remedy this situation through a community-driven consensus, we propose a set of minimum information insights to be provided for ML papers, allowing reviewers and readers to assess the quality and reliability of the proposed methods more faithfully. Our recommendations are made primarily for the case of supervised learning, as this is the most common type of ML approach used, but these can be easily extended to other fields of ML, like unsupervised, semi-supervised learning, transfer learning and reinforcement learning.

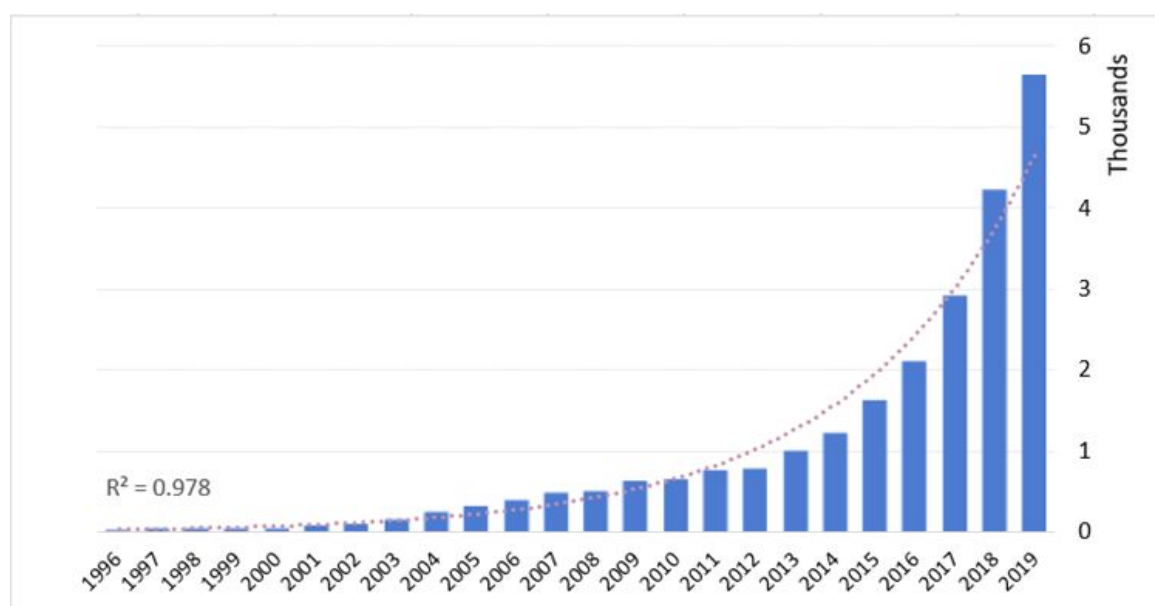


Figure 1. Exponential increase of ML publications in biology. The number of ML publications per year is based on Web of Science from 1996 onwards using the “topic” category for “machine learning” in combination with each of the following terms: “biolog*”, “medicine”, “genom*”, “prote*”, “cell*”, “post translational”, “metabolic” and “clinical”.

Broad topic	Major concerns	Consequences	Recommendation(s)/requirement(s)
Data	<ul style="list-style-type: none"> - <u>Appropriate partitioning, dependence</u> between train and test data, <u>sufficient data size & quality</u> - <u>Class imbalance</u> - No access to data 	<ul style="list-style-type: none"> • Unreliable or biased performance evaluation. Data not representative of domain application. • Cannot check data credibility. 	<p>Independence of optimization (training) and evaluation (testing) sets. (requirement)</p> <p>Especially important for meta algorithms, where independence of multiple training sets must be shown to be independent from the evaluation (testing) sets. Data size & distribution is representative of the domain. (requirement)</p> <p>Release data preferably using appropriate long-term repositories, including exact splits (requirement)</p> <p>For data produced by different experimental runs the batch effect must be tested, by treating each set of experiments as not independent data <i>(recommendation)</i></p>
Optimization	<ul style="list-style-type: none"> - <u>Overfitting, underfitting & Illegal parameter tuning</u> - Imprecise parameters and protocols given. 	<ul style="list-style-type: none"> • Over/under optimistic performance reported. Model noise & miss relevant relationships. 	<p>Evaluation datasets should not be used for either feature selection or parameter tuning. (requirement)</p> <p>Appropriate metrics to prove no over/under fitting, i.e. comparison of training and testing error. (requirement)</p> <p>Release definitions of all algorithmic hyper-parameters, parameters and optimization protocol (requirement)</p> <p>Include explicit model validation techniques, such as N-fold Cross validation <i>(recommendation)</i>.</p>
Model	<ul style="list-style-type: none"> - Models are <u>blackboxes</u> but <u>transparency</u> is required for certain problem - No access to: resulting <u>source code</u>, trained <u>models & data</u> - <u>Execution time</u> is impractical 	<ul style="list-style-type: none"> • Models output some score which cannot be traced back to data. Perhaps, a <u>transparent</u> model is a better solution • Cannot: cross compare methods, reproducibility, & check data credibility. • Model takes too much time to produce results 	<p>Blackbox models are interesting but certain problems need interpretable solutions (e.g. diagnostics). Blackbox vs. transparent should be decided before optimization. <i>(recommendation)</i>.</p> <p>Hybrid blackbox and transparent models might be a balance. <i>(recommendation)</i>.</p> <p>Release of: documented source code + models + executable + UI/webserver + software containers. <i>(recommendation)</i></p> <p>Report execution time averaged across many repeats. If computationally tough compare to similar methods <i>(recommendation)</i></p>
Evaluation	<ul style="list-style-type: none"> - <u>Performance measures</u> not <u>adequate</u> - <u>No comparisons</u> to <u>baselines</u> or <u>other methods</u> 	<ul style="list-style-type: none"> • Biased performance measures reported. • The method is falsely claimed as 	<p>Compare with public methods & simple models (baselines). (requirement)</p> <p>Adoption of community validated measures for evaluation. (requirement)</p>

	- Highly <u>variable performance</u> .	state-of-the-art. <ul style="list-style-type: none"> • Then, unpredictable performance in production. 	Comparison of related methods and alternatives on the same dataset, ablation study for measuring the impact of components. (<i>recommendation</i>) Data distribution checked for good domain representativeness. Confidence intervals/error intervals to gauge prediction robustness. (<i>recommendation</i>)
--	--	---	--

Table 1. ML in the Biology: concerns, the consequences they impart and recommendations/requirements (recommendations in *italics* and requirements in **bold**). Key terms underlined.

Development of the recommendations

The recommendations outlined below were initially formulated through the ELIXIR ML focus group after the publication of a comment calling for the establishment of standards for ML in biology²². ELIXIR, initially established in 2014, is now a mature intergovernmental European infrastructure for biological data and represents over 220 research organizations in 22 countries across many aspects of bioinformatics²³. Over 700 national experts participate in the development and operation of national services that contribute to data access, integration, training and analysis for the research community. Over 50 of these experts involved in the field of ML have established an ELIXIR ML focus group (<https://elixir-europe.org/focus-groups/machine-learning>) which held a number of meetings to develop and refine the recommendations based on a broad consensus among them.

Scope of the recommendations

The recommendations cover four separate aspects covering the major areas of ML according to the “DOME” acronym: data, optimization, model and evaluation. In the following, we will address each of these aspects separately.

1. Data

Machine learning models analyse experimental biological data by extracting patterns. The extracted patterns can then be used to give biological insights on similar data that were not previously seen by the model. The degree to which a model retains its performance on new data is called *generalization* power. Building ML models that generalize well is the main challenge of these methodologies, otherwise the trained models cannot be reused. Preprocessing data properly, and using it in a knowledgeable manner is the only way to obtain good generalization. Some basic concerns to consider are:

- training, test and validation datasets are partially or completely overlapping. This includes both sequence/structure similarities and a set of experiments obtained in different conditions or batches (e.g. for next-generation sequencing data).
- training dataset is too small to capture the full complexity of the underlying distribution.
- validation and test datasets are too small to provide a stable estimate of the model’s generalization power.
- training, validation and test sets are not representative of the problem domain due to e.g. presence of high noise levels, imbalanced classes, large chunks of similar (redundant) data points
- data is not released to the public

State-of-the-art ML models are often capable of memorizing all the variation in the given data. Such models when evaluated on data that they were exposed to during training would create an illusion of mastering the task at hand. However, when tested on an independent set of data, the performance would seem less impressive, suggesting low generalization power of the model. In order to tackle this problem, initial data is divided randomly into non-overlapping unequal parts. The larger set (usually about 70-80%) is called training data and is used to shape the model, helping it achieve the highest possible performance in the given task via optimization (see the next section). The data split, termed the test set, is invariably smaller and it is left out from training. It is used only when it is time to estimate the true generalization power of the final model. The resulting generalization score is considered realistic only if the test set was used once. Overlapping of train/test data splits is particularly troublesome in the case of protein and DNA sequences that make up a large fraction of biological data and can be similar by evolutionary homology. Thus, special methods such as redundancy reduction techniques can be used to mitigate this problem for sequences^{9,16}. Furthermore, to ensure unbiased evaluation, when researchers want to train multiple models or select the best hyper-parameters, the training data is divided further to form a validation set. Different models and hyper-parameters are then trained on the remaining part of the training data, and evaluated on the validation set. Next, the best model is selected and evaluated on the test data, producing the final performance result. Despite seeming bias-proof this scenario still leaves room for unwanted unfairness. Randomly choosing a single validation set may favour some models, while opposing others. The cross-validation or bootstrapping techniques which choose a new validation set multiple times from the available training data, is thus considered a preferred solution²⁴.

The size of all three subsets of data, namely training, validation and test sets is of great importance. A too small training set may prevent the model from reaching its full potential. While having too small test and validation sets, may result in unstable performance estimates. When data is limited (which often can be the case in biology and medicine), a fine balance between the size of train and validation/test splits is required. Having said this, the problem of training ML models that can generalize well in so called small training data, usually requires special models and algorithms^{19,25}.

Another important indicator to keep in mind when preparing data for the ML model is the degree to which training data truly represents the domain of the problem, related to a representative sample of a population in statistics²⁶. Namely, if training data is significantly different from the data that the model will encounter in the future, for example, due to the presence of excessive amounts of noise or skewed distributions of input and output, the model's generalization power will be limited. In the case where input features and output class distributions significantly differ from what is found in the literature (or more importantly in nature), it is often necessary to make sure that training, validation and test sets are pre-processed/sampled to be representative. This can be again achieved by adding or removing data points that create the skew. To this end, thorough exploration analysis as well as quality control need to be performed to make sure that data is trustworthy.

Lastly, it is important to make as much data available for the public as possible. Having open access to the data used for experiments including precise data splits would ensure better reproducibility of published research and as a result will improve the overall quality of published ML papers. If datasets are not already available at any public repository, authors should be encouraged to find the most appropriate one, e.g. ELIXIR Deposition databases or Zenodo, to guarantee the long-term availability of such data.

There are several strategies that can be employed to mitigate the issues described above in order to ensure correct data organization:

1. Create two distinct sets of data: training and test sets (requirement). If the training process implicitly requires a validation set (e.g. multiple models are optimized; hyper-parameters are optimized), an additional validation set is needed. Alternatively, consider applying a cross-validation algorithm.
2. Report size of the data as well as the size of resulting training and test (and if applicable validation) sets. Search in the literature and report data sizes used to train similar ML methods.
3. Plot the distributions of the data points (e.g. number of patients vs number of controls, age of patients, protein structure types) in the train and test sets to indicate the type of data the model can handle. Make sure, given the problem domain, there is good representation in both sets. This is a highly recommended strategy, as it can clearly show whether the data size and distribution is representative of the respective problem domain, e.g., methods exist that address imbalanced classes^{27,28}.
4. Perform quality control and report indicators that can help to assess the quality of the dataset (e.g. resolution of images or sequence annotation source etc.).
5. It is a requirement to release the train/test/validation splits exactly as described. If cross-validation is performed the exact folds should be released as described.

2. Optimization

Optimization, also known as model training, refers to the process of changing values that constitute the model (parameters and hyper-parameters of the model) in a way that maximizes the model's ability to solve a given problem. In this section, we will focus on problems associated with a poor choice of optimization strategy. Such problems may include:

- selecting a too powerful model, that may over-fit (known as high-variance²⁹)
- selecting a too simple model, that may under-fit (known as high-bias²⁹)
- parameters are optimized and/or features are selected on hold-out data used to evaluate performance only. This may be particularly hard to spot for meta-predictors.
- parameters, hyper-parameters and optimization protocol are not specified and/or files supporting their specification are not open-access or follow a standard widely adopted by the community (if any).

Since model parameters are openly optimized to achieve the best possible performance on the training data, it is common for models to perform better on training sets than on validation or test sets. With more complex and hence powerful algorithms the gap between performance on training and on hold out data grows. In ML, this gap is indicative of over-fitting. This is particularly difficult to spot in meta-algorithms, such as boosting or majority voting for an ensemble of classifiers, where it is not immediately clear if the hold-out set is independent of multiple training sets. Over-fitting usually happens due to the capacity of more complex models to learn arbitrary patterns in the training data, including random noise, sampling artifacts or distributions of small data not representative of the underlying domain. A model that has suffered severe over-fitting will show an extraordinary performance on training data, while performing poorly on unseen data, rendering it useless for real-life applications. On the other side of the spectrum is a problem of under-fitting. Under-fitting occurs when very simple models capable of capturing only straightforward dependencies between features are applied to data of a more complex nature (e.g logistic regression or linear classifiers are considered simple models). The number of input features (f) and feature encoding schemes are also a factor for optimization since (1) good feature

representations make for good optimization and (2) because large f often implies a large number of parameters and in turn algorithm complexity. Thus, algorithms for feature selection³⁰ can be employed to reduce the chances of over-fitting. However, feature selection comes with its own concerns, the main one being illegal selection on non-training data which may lead to an overestimation of performance. Finally, the release of files showing the exact specification of the optimization protocol and the type of parameters/hyper-parameters are a vital characteristic of the final algorithm. For instance, neural networks can have a differing number of neurons with different activation functions and weight updating settings; support vector machines can have varying margin hyperplane sizes, input kernels and margin computations; and random forests can have a variable number of trees of varied depth.

Some key recommendations that may help to detect and mitigate problems with optimization:

- When building models, estimate and compare performances on both training and validation sets (large difference usually suggests over-fitting, while low overall performance on both points towards under-fitting).
- If your model suffers from under-fitting, choose a more complex ML algorithm (e.g. with more parameters). If you diagnosed over-fitting, either select a simpler algorithm, use more data or data augmentation or reduce your model's complexity (e.g. in neural networks move from deep to shallow neural networks or apply regularization). Note that choosing between different algorithms must be done on training or validation data in order to not introduce bias.
- If over-fitting persists, attempt to increase the size or quality, reduce the number of input features of the training data and re-train.
- It is a requirement that exact details of optimized parameters/hyper-parameters and optimization protocols are described.
- It is a requirement that the protocol to reduce input features to the 'best' ones and parameter tuning is done on training data only. For meta-algorithms, this must be performed clearly if all train sets are available.
- It is good practice to release models (i.e. saved model files) at the initial and final training point. These files can include regular parameters and hyper-parameter configurations, optimization protocols and selected features.

3. Model

Good overall performance of the trained model and its ability to generalize well to unseen data are important factors that undoubtedly affect the applicability of any proposed ML research. However, a few other important aspects related to ML models must be kept in mind. These include the following fallacies:

- Employing unexplainable models (blackbox) for areas where interpretability is required.
- The various components of a model (source code, model files, parameter configurations, executables) are not made available to the public.
- Computational requirements to execute the trained models (generate prediction based on new data) are impractical.

ML models differ significantly in their ability to explain learned patterns back to humans. Hence, two main classes of ML models are recognized: transparent and blackbox³¹. For example, one may easily interpret a prediction made by linear regression or a decision tree; however, a model based on neural

networks or support vector machines can be effectively considered a blackbox. This by itself is not an issue - a blackbox model can be very effective in a particular context (e.g. image recognition). On the other hand certain problems and particular studies require model transparency in order to claim knowledge breakthrough. Interpretability is particularly relevant in areas of discovery such as drug design^{6,32} and diagnostics³³ where inferring causal relationships from observational data is of utmost importance. Many approaches were proposed that aim at opening the blackboxes^{34,35}. However, despite growing interest in explaining blackbox models, a recent perspective has proposed to enable transparency by employing interpretable models in the first place²⁰. Conversely, opinions on the adoption of blackbox models under specific circumstances have been voiced³⁶. Depending on the value of the outcome, blackbox models may help to remove subjective factors from controversial decision-making, and offer the potential to inspire and guide human inquiry.

For a newly reported, state-of-the-art performing model, restricting open-access of its underlying components such as source code, trained model files that include hyper-parameter configurations, and a means to run the algorithm is a major obstacle for furthering scientific goals in ML. The consequences that arise here are the inability to reproduce the method, build improved versions, benchmark its performances or even use it for biological insight. Finally, the model's speed of execution is also another factor especially in biologically domains that require large-scale predictions to make hypotheses.

Some key recommendations towards addressing these issues are the following:

1. There should always be a clear selection process between the choice of blackbox vs transparent models. Blackbox models are interesting but certain challenges and approaches require interpretable solutions (such as within a diagnostic context).
 2. It is highly recommended to give special attention to models that offer mechanistic insights on the internal data, over blackbox models. This consequently implies that thinking of ways to interpret predictions needs to be planned ahead.
 3. It is highly recommended to include a performance comparison between blackbox and transparent models, in order to gauge the tradeoff (e.g. between the choice of neural networks vs decision trees).
 4. It is recommended that documented source code should be released unless it is undergoing commercialization in which case statements in 'conflict of interest' sections of articles should be made.
 5. It is recommended that executables, web-servers, virtual machines and software container instances that can run the ML algorithm are released so that they can be used for cross comparison of methods. The algorithms run time on standard PC CPUs should also be reported for a typical set of data points.
-

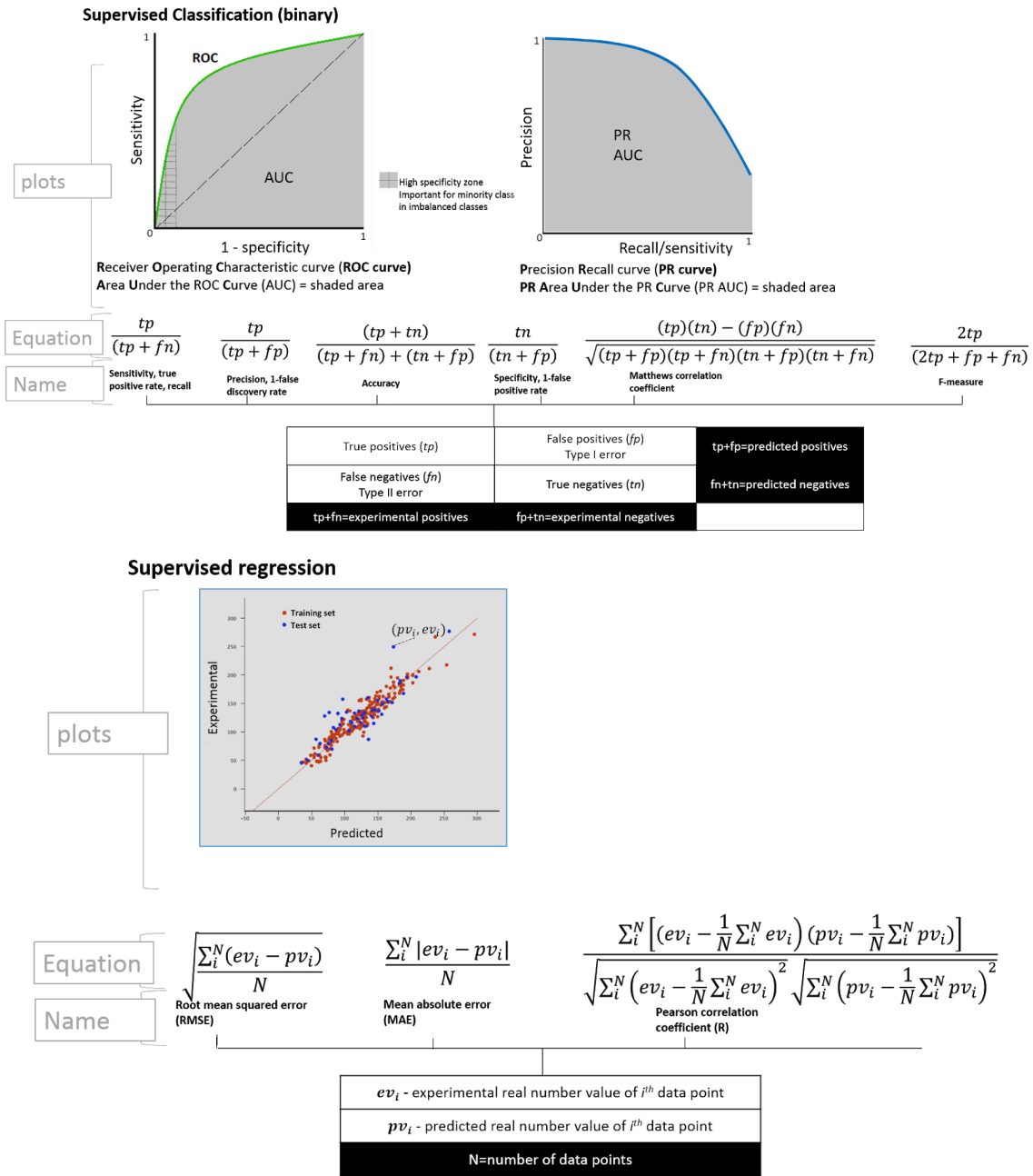


Figure 2. (Top) Classification metrics. For binary classification, true positives (tp), false positives (fp), false negatives (fn) and true negatives (tn) together form the confusion matrix. As all classification measures can be calculated from combinations of these four basic values, the confusion matrix should be provided as a core metric. Several measures (shown as equations) and plots should be used to evaluate the ML methods. If the article becomes too cluttered, many of the metrics can be moved to supplementary material. For descriptions on how to attempt these metrics to multi-class problems see ³⁷. **(Bottom) Regression metrics.** ML regression attempts to produce predicted values (pv) matching experimental values (ev). Metrics (shown as equations) attempt to capture the difference in various ways. Alternatively, a plot can provide a visual way to represent the differences. It is advisable to report all in any ML work.

4. Evaluation

In the implementation of a robust and trustworthy ML method, a comprehensive data description, a correct optimization protocol, and a clearly defined (and open-access) model are critical first steps; however, equally important is a valid assessment methodology for any final model. Here are a few possible risks related to model assessment and evaluation:

- the selected performance measures are not adequate or comprehensive for the problem at hand
- reported performances are highly unstable.
- obtained performances are not compared to similar studies, methods and community-agreed datasets; obtained performances are not compared to simpler baseline methods.

Performance metrics are quantifiable indicators of a model's ability to solve the given task. For assessing different ML problems (e.g. classification or regression) dozens of metrics are available³⁷. Sometimes it is challenging to select the right measure from the exhaustive list, while on other occasions researchers may be tempted to use specific metrics which may provide a more favourable account of their model performance (i.e. cherry-picking). In either way, the resulting estimate may be neither valid nor useful. Currently, it is up to the peer-reviewer to check if reasonable metrics were employed. But as the choice of metric tends to be domain-specific, reviewers should not be expected to always know the most suitable ones. In this work, we propose a consensus-based list of metrics that should be applied for classification and regression types of ML, Figure 2. Simultaneously, confidence intervals or variance/standard deviations should be associated with each metric to show prediction stability.

Once performance metrics are decided, methods published in the same biological domain must be cross-compared using appropriate statistical tests (e.g. Student's t-test). Additionally, to prevent the release of ML methods that appear sophisticated but perform no better than simpler algorithms, baselines should be compared to the 'sophisticated' method and proven to be statistically inferior (e.g. logistic regression vs. small NN vs. deep NN).

Some key recommendations towards this direction are:

- Performance metrics, selected for assessing model performance should be clearly reported and the choice should be justified. Figure 2 shows a list of reasonable performance measures for classification and regression tasks.
- A case when the training data is imbalanced (i.e. either positive or negative class is prevalent) requires a distinct evaluation strategy. For example, ROC analysis at high specificity for the minority class; see Figure 2.
- The procedure that was used to evaluate the performance of the trained model should also be discussed in detail as relevant also for other methods being compared.
- When reporting final ML performance always report on unseen data (either validation set or test set). Training performance can be reported for self-consistency checks but not as a performance metric.
- When performing method comparisons, these should be made against methods that are publicly available, baseline methods, or community-wide evaluation frameworks (e.g. CASP/CAFA) or combinations thereof.
- When reporting the model performance, it is important to include confidence intervals using statistical mean and variance calculations to ensure stable results.
- It is recommended that raw evaluation tables, equations, visualization, statistical tests and statistical testing code should be released to the community.

Box 1: How to structure a Materials and Methods section

Here we suggest a list of questions that must be asked about each section to ensure high quality of ML analysis.

- **Data:** *(this section is to be repeated separately for each dataset)*
 - *Provenance:* What is the source of the data (database, publication, direct experiment)? How many data points are available in total for the positive (N_{pos}) and negative (N_{neg}) cases? Has the dataset been previously used by other papers and/or is it recognized by the community?
 - *Data splits:* How many data points are in the training and test sets? Was a separate validation set used, and if yes, how large was it? Is the distribution of data types in the training and test sets different? Is the distribution of data types in both training and test sets plotted?
 - *Redundancy between data splits:* How were the sets split? Are the training and test sets independent? How was this enforced (e.g. redundancy reduction to less than X% pairwise identity)? How does the distribution compare to previously published ML datasets?
 - *Availability of data:* Is the data, including the data splits used, released in a public forum? If yes, where (e.g. supporting material, URL)?
- **Optimization:** *(this section is to be repeated separately for each trained model)*
 - *Algorithm:* What is the ML algorithm class used? Is the ML algorithm new? If yes, why is it not published in a ML journal, and why was it chosen over better known alternatives?
 - *Meta-predictions:* Does the model use data from other ML algorithms as input? If yes, which ones? Is it completely clear that training data of initial predictors and meta-predictor is independent of test data for the meta-predictor?
 - *Data encoding:* How was the data encoded and pre-processed for the ML algorithm?
 - *Parameters:* How many parameters (p) are used in the model? How was p selected?
 - *Features:* How many features (f) are used as input? Was feature selection performed? If yes, was it performed using the training set only?
 - *Fitting:* Is the number of parameters (p) reasonable for the number of training points ($N_{\text{pos}} + N_{\text{neg}}$) and/or is the number of features (f) large (e.g. is $p \gg (N_{\text{pos}} + N_{\text{neg}})$ and/or $f > 100$)? If yes, how was over-fitting ruled out? Conversely, if $N_{\text{pos}} + N_{\text{neg}}$ seems very much larger than p and/or f is small (e.g. $N_{\text{pos}} + N_{\text{neg}} \gg p$ and/or $f < 5$) how was under-fitting ruled out?
 - *Availability of configuration:* Are the hyper-parameter configurations, model files and optimization parameters available? If yes, where (e.g. URL)?
- **Model:** *(this section is to be repeated separately for each trained model)*
 - *Interpretability:* Is the model blackbox, if so did you compare performance to transparent models? Is the model transparent, if so did you compare performance to blackbox models? Can you explain clearly why your model is transparent/interpretable?
 - *Execution time:* How much real-time does a single representative prediction require on a standard machine? (e.g. seconds on a desktop PC or high-performance computing cluster)
 - *Availability of software:* Is the source code released? Is a method to run the algorithm such as executable, web server, virtual machine or container instance released? If yes, where (e.g. URL)?
- **Evaluation:**
 - *Evaluation method:* How was the method evaluated? (E.g. cross-validation, independent dataset, novel experiments)

- *Performance measures*: Which performance metrics are reported? Is this set representative?
- *Comparison*: Was a comparison to publicly available methods performed on benchmark datasets? Was a comparison to simpler baselines performed?
- *Confidence*: Do the performance metrics have confidence intervals? Are the results statistically significant to claim that the method is superior to others and baselines?
- *Availability of evaluation*: Are the raw evaluation files (e.g. assignments for comparison and baselines, statistical code, confusion matrices) available? If yes, where (e.g. URL)?

The above description is shown in table format in the Supplementary Material.

Conclusion

The objective of our recommendations is to increase the transparency and reproducibility of ML methods for the reader, the reviewer, the experimentalist and the wider community. Although we refer to ML, our recommendations should also be applied to any statistical and empirical methods. We recognize that these recommendations are not exhaustive and should be viewed as a consensus-based first iteration of a continuously evolving system of community self-review. One of the most pressing issues is to agree to a standardized data structure to describe the most relevant features of the ML methods being presented. As a first step to address this issue, we recommend including a “ML summary table” as described here in future ML studies. We recommend including the following sentence in the methods section of all papers: “To increase reproducibility of the machine learning method of this study, the machine learning summary table (Table X) is included in the supporting information as per consensus guidelines (with reference to this manuscript).” We also recommend that training and testing sets be made available for reanalysis to interested parties by including them as supporting information to the manuscript.

The development of a standardized approach for reporting ML methods has major advantages in increasing the quality of publishing ML methods. First, the disparity in manuscripts of reporting key elements of the ML method can make reviewing and assessing the ML method challenging. Second, certain key statistics and metrics that may affect the validity of the publication’s conclusions are sometimes not mentioned at all. Third, there are unexplored opportunities associated with meta-analyses of ML datasets. Access to large sets of raw data can enhance both the comparison between methods and facilitate the development of better performing methods, while reducing unnecessary repetition of data generation. We believe that our recommendations to include a “machine learning summary table” and the availability of raw data as described above will greatly benefit the ML community and improve its standing with the intended users of these methods.

The ELIXIR Machine Learning focus group

Emidio Capriotti (ORCID: 0000-0002-2323-0963)

Department of Pharmacy and Biotechnology, University of Bologna, Bologna (Italy)

Rita Casadio (ORCID: 0002-7462-7039)

Biocomputing Group, University of Bologna, Italy; IBIOM-CNR, Italy

Salvador Capella-Gutierrez (ORCID: 0000-0002-0309-604X)

INB Coordination Unit, Life Science Department. Barcelona Supercomputing Center (BSC), Barcelona, Spain

Davide Cirillo (ORCID: 0000-0003-4982-4716)

Life Science Department. Barcelona Supercomputing Center (BSC), Barcelona, Spain

Alexandros C. Dimopoulos (ORCID: 0000-0002-4602-2040)

Institute for Fundamental Biomedical Science, Biomedical Sciences Research Center "Alexander Fleming", Athens, Greece

Victoria Dominguez Del Angel (ORCID: 0000-0002-5514-6651)

Centre National de Recherche Scientifique, University Paris-Saclay, IFB, France

Joaquin Dopazo (ORCID: 0000-0003-3318-120X)

Clinical Bioinformatics Area, Fundación Progreso y Salud, Sevilla, Spain

Piero Fariselli (ORCID: 0000-0003-1811-4762)

Department of Medical Sciences, University of Turin, Turin, Italy

José M^a Fernández (ORCID: 0000-0002-4806-5140)

INB Coordination Unit, Life Sciences Department, Barcelona Supercomputing Center (BSC), Barcelona, Spain

Dmytro Fishman (ORCID: 0000-0002-4644-8893)

Institute of Computer Science, University of Tartu, Estonia

Dario Garcia-Gasulla (ORCID: 0000-0001-6732-5641)

Barcelona Supercomputing Center (BSC), Barcelona, Spain

Jen Harrow (ORCID:0000-0003-0338-3070)

ELIXIR HUB, South building, Wellcome Genome Campus, Hinxton, Cambridge, UK.

Florian Huber (ORCID: 0000-0002-3535-9406)

Netherlands eScience Center, Amsterdam, the Netherlands.

Anna Kreshuk (ORCID:0000-0003-1334-6388)

EMBL Heidelberg

Pier Luigi Martelli (ORCID: 0000-0002-0274-5669)

Biocomputing Group, University of Bologna, Italy

Arcadi Navarro (ORCID: 0000-0003-2162-8246)

Institute of Evolutionary Biology (Department of Experimental and Health Sciences, CSIC-Universitat Pompeu Fabra), Barcelona, Spain

Catalan Institution of Research and Advanced Studies (ICREA), Barcelona, Spain

CRG, Centre for Genomic Regulation, Barcelona Institute of Science and Technology (BIST), Barcelona, Spain

Marco Necci (ORCID: 0000-0001-9377-482X)

Dept. of Biomedical Sciences, University of Padua, Padua, Italy.

Pilib Ó Broin (ORCID: 0000-0002-6702-8564)

School of Mathematics, Statistics & Applied Mathematics, National University of Ireland Galway, Ireland

Janet Piñero (ORCID: 0000-0003-1244-7654)

Research Programme on Biomedical Informatics (GRIB), Hospital del Mar Medical Research Institute (IMIM),
Department of Experimental and Health Sciences, Pompeu Fabra University (UPF), Barcelona, Spain

Damiano Piovesan (ORCID: 0000-0001-8210-2390)

Dept. of Biomedical Sciences, University of Padua, Padua, Italy.

Gianluca Pollastri (ORCID: 0000-0002-5825-4949)

School of Computer Science, University College Dublin, Ireland

Fotis E. Psomopoulos (ORCID: 0000-0002-0222-4273)

Institute of Applied Biosciences, Centre for Research and Technology Hellas, Thessaloniki, Greece

Martin Reczko (ORCID: 0000-0002-0005-8718)

Institute for Fundamental Biomedical Science, Biomedical Sciences Research Center "Alexander Fleming",
Athens, Greece

Francesco Ronzano (ORCID: 0000-0001-5037-9061)

Research Programme on Biomedical Informatics (GRIB), Hospital del Mar Medical Research Institute (IMIM),
Department of Experimental and Health Sciences, Pompeu Fabra University (UPF), Barcelona, Spain

Venkata Satagopam (ORCID: 0000-0002-6532-5880)

Luxembourg Centre For Systems Biomedicine (LCSB), University of Luxembourg and ELIXIR-Luxembourg
Node

Castrense Savojardo (ORCID: 0000-0002-7359-0633)

Biocomputing Group, University of Bologna, Italy

Vojtech Spiwok (ORCID: 0000-0001-8108-2033)

Department of Biochemistry and Microbiology, University of Chemistry and Technology, Prague, ELIXIR-Czech
Republic

Marco Antonio Tangaro (ORCID: 0000-0003-3923-2266)

Institute of Biomembranes, Bioenergetics and Molecular Biotechnologies, National Research Council (CNR), Bari,
Italy

Giacomo Tartari

Institute of Biomembranes, Bioenergetics and Molecular Biotechnologies, National Research Council (CNR), Bari,
Italy

Tiina Titma (ORCID: 0000-0002-4935-8914)

School of Information Technologies, Tallinn University of Technology, Estonia

Silvio C. E. Tosatto (ORCID: 0000-0003-4525-7793)

Dept. of Biomedical Sciences, University of Padua, Padua, Italy.

Alfonso Valencia (ORCID:0000-0002-8937-6789)

Catalan Institution of Research and Advanced Studies (ICREA), Barcelona, Spain
Life Science Department. Barcelona Supercomputing Center (BSC), Barcelona, Spain

Ian Walsh (ORCID: 0000-0003-3994-5522)

Bioprocessing Technology Institute, Agency for Science, Technology and Research, Singapore

Federico Zambelli (ORCID: 0000-0003-3487-4331)

Dept. of Biosciences, University of Milan, Milan, Italy

Author contributions

IW, DF, JH, FP and SCET guided the development, writing and final edits. All members of the ELIXIR machine learning focus group contributed to the discussions leading to the recommendations and writing of the manuscript.

Competing interests

The authors declare no competing interests.

Acknowledgements

The work of the ML focus group was funded by ELIXIR, the Research infrastructure for life-science data. The authors are grateful to Xenia Perez Sitja for help with supplementary figure 1.

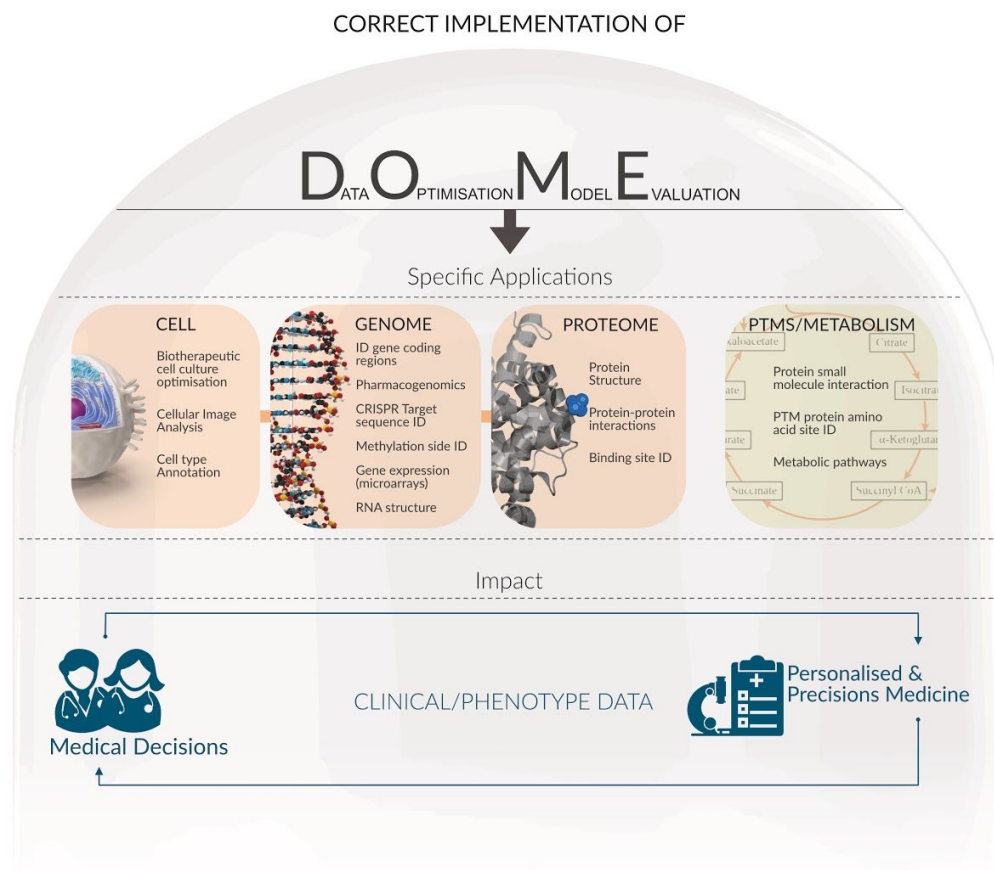
References

1. Baron, C. S. *et al.* Cell Type Purification by Single-Cell Transcriptome-Trained Sorting. *Cell* **179**, 527-542.e19 (2019).
2. Libbrecht, M. W. & Noble, W. S. Machine learning applications in genetics and genomics. *Nat. Rev. Genet.* **16**, 321–332 (2015).
3. Radivojac, P. *et al.* A large-scale evaluation of computational protein function prediction. *Nat. Methods* **10**, 221–227 (2013).
4. Franciosa, G., Martinez-Val, A. & Olsen, J. V. Deciphering the human phosphoproteome. *Nat. Biotechnol.* 1–2 (2020) doi:10.1038/s41587-020-0441-3.
5. Yang, J. H. *et al.* A White-Box Machine Learning Approach for Revealing Antibiotic Mechanisms of Action. *Cell* **177**, 1649-1661.e9 (2019).
6. Vamathevan, J. *et al.* Applications of machine learning in drug discovery and development. *Nat. Rev. Drug Discov.* **18**, 463–477 (2019).
7. Rajkomar, A., Dean, J. & Kohane, I. Machine Learning in Medicine. *N. Engl. J. Med.* **380**, 1347–1358 (2019).
8. Ascent of machine learning in medicine. *Nat. Mater.* **18**, 407 (2019).

9. Walsh, I., Pollastri, G. & Tosatto, S. C. E. Correct machine learning on protein sequences: a peer-reviewing perspective. *Brief. Bioinform.* **17**, 831–840 (2016).
10. Bishop, D. Rein in the four horsemen of irreproducibility. *Nature* **568**, 435 (2019).
11. Schwartz, D. Prediction of lysine post-translational modifications using bioinformatic tools. *Essays Biochem.* **52**, 165–177 (2012).
12. Piovesan, D. *et al.* Assessing predictors for new post translational modification sites: a case study on hydroxylation. *bioRxiv* (2020).
13. Hutson, M. Artificial intelligence faces reproducibility crisis. *Science* **359**, 725–726 (2018).
14. Ong, Y.-S. & Gupta, A. AIR5: Five Pillars of Artificial Intelligence Research. *IEEE Trans. Emerg. Top. Comput. Intell.* **3**, 411–415 (2019).
15. Haibe-Kains, B. *et al.* The importance of transparency and reproducibility in artificial intelligence research. *ArXiv200300898 Stat* (2020).
16. Columbus, L. Roundup Of Machine Learning Forecasts And Market Estimates For 2019. *Forbes* <https://www.forbes.com/sites/louiscolombus/2019/03/27/roundup-of-machine-learning-forecasts-and-market-estimates-2019/>.
17. Littmann, M. *et al.* Validity of machine learning in biology and medicine increased through collaborations across fields of expertise. *Nat. Mach. Intell.* 1–7 (2020).
18. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *nature* **521**, 436–444 (2015).
19. Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K. & Wierstra, D. Matching Networks for One Shot Learning. *ArXiv160604080 Cs Stat* (2017).
20. Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.* **1**, 206–215 (2019).
21. Noorden, R. V. Highly cited researcher banned from journal board for citation abuse. *Nature* **578**, 200–201 (2020).
22. Jones, D. T. Setting the standards for machine learning in biology. *Nat. Rev. Mol. Cell Biol.* **20**, 659–660 (2019).
23. ELIXIR: Providing a Sustainable Infrastructure for Federated Access to Life Science Data at European Scale. *submitted*.

24. Kohavi, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. in vol. 14 1137–1145 (Montreal, Canada, 1995).
25. Pan, S. J. & Yang, Q. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* **22**, 1345–1359 (2010).
26. Campbell, R. C. *Statistics for biologists*. (Cambridge University Press, 1989).
27. Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **16**, 321–357 (2002).
28. He, H., Bai, Y., Garcia, E. A. & Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. in 1322–1328 (IEEE, 2008).
29. Mehta, P. *et al.* A high-bias, low-variance introduction to machine learning for physicists. *Phys. Rep.* (2019).
30. Guyon, I. & Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.* **3**, 1157–1182 (2003).
31. Arrieta, A. B. *et al.* Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* **58**, 82–115 (2020).
32. Richardson, P. *et al.* Baricitinib as potential treatment for 2019-nCoV acute respiratory disease. *The Lancet* **395**, e30–e31 (2020).
33. He, J. *et al.* The practical implementation of artificial intelligence technologies in medicine. *Nat. Med.* **25**, 30–36 (2019).
34. Guidotti, R. *et al.* A survey of methods for explaining black box models. *ACM Comput. Surv. CSUR* **51**, 1–42 (2018).
35. Adadi, A. & Berrada, M. Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI). *IEEE Access* **6**, 52138–52160 (2018).
36. Holm, E. A. In defense of the black box. *Science* **364**, 26–27 (2019).
37. Baldi, P., Brunak, S., Chauvin, Y., Andersen, C. A. & Nielsen, H. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinforma. Oxf. Engl.* **16**, 412–424 (2000).

Supplementary Material



Supplementary Figure 1. The four major topics when developing machine learning: data, optimization, model and evaluation (DOME). These can be used for many specific applications, some of which are shown for cellular, genomics, proteomics, post-translational modifications (PTMs) and metabolomic analysis. The impact of correct machine learning goes all the way to medical decisions and personalised medicine.

Machine learning summary table:

Data	Provenance	Source of data, data points (positive, N_{pos} / negative, N_{neg}). Used by previous papers and/or community.
	Dataset splits	Size of N_{pos} and N_{neg} of training set, validation set (if present), test set. Distribution of N_{pos} and N_{neg} across sets.
<u>(section to be repeated for each dataset)</u>	Redundancy between data splits	Independence between sets. Strategy used to make examples representative (e.g. eliminating data points more similar than X%). Comparison relative to other datasets.
	Availability of data	Yes/no for datasets. If yes: Supporting information, website URL.
Optimization	Algorithm	ML class (e.g. neural network, random forest, SVM). If novel approach, reason is it not previously published.
<u>(section to be repeated for each trained model)</u>	Meta-predictions	Yes/No. If yes: how other methods are used and whether the datasets are clearly independent.
	Data encoding	How input data is transformed (e.g. global features, sliding window on sequence).
	Parameters	Number of ML model parameters (p), e.g. tunable weights in neural networks. Protocol used to select p .
	Features	Number of ML input features (f), i.e. encoding of data points. In case of feature selection: Protocol used, indicating whether it was performed on training data only.
	Fitting	Justification for excluding over- (if $p \gg N_{pos, train} + N_{neg, train}$ or $f > 100$) and under-fitting (if $p \ll N_{pos, train} + N_{neg, train}$ or $f < 5$).
	Availability of configuration	Yes/no for hyper-parameter configuration and model files. If yes: Supporting information, website URL.
Model	Interpretability	Black box/transparent and justification. Explanation whether the model was compared to the other class.
<u>(section to be repeated for each trained model)</u>	Execution time	CPU time of single representative execution on standard hardware (e.g. seconds on desktop PC).
	Availability of software	Source code repository (e.g. GitHub), software container, website URL.
Evaluation	Evaluation method	Cross-validation, independent dataset or novel experiments.
	Performance measures	Accuracy, sensitivity, specificity, etc.
	Comparison	Name of other methods and, if available, definition of baselines compared to. Justification of representativeness.
	Confidence	Confidence intervals and statistical significance. Justification for claiming performance differences.

	Availability of evaluation	Yes/no for raw evaluation files (e.g. assignments for comparison and baselines, confusion matrices). If yes: Supporting information, website URL.
--	----------------------------	---

Example tables for author reference:

The following is an example for a primary ML summary table built from (Walsh et al., Bioinformatics 2012).

Data: X-ray disorder	Provenance	<i>Protein Data Bank (PDB) X-ray structures until May 2008 (training) and from May 2008 until September 2010 (test). 3,813 proteins total. $N_{pos} = 44,433$ residues. $N_{neg} = 710,207$ residues. Not previously used.</i>
	Dataset splits	<i>$N_{pos,train} = 37,495$. $N_{neg,train} = 622,625$. $N_{pos,test} = 6,938$. $N_{neg,test} = 87,582$ residues. No separate validation set. 5.68% positives on training set. 7.34% positives on test set.</i>
	Redundancy between data splits	<i>Maximum pairwise identity within and between training and test set is 25% enforced with UniqueProt tool.</i>
	Availability of data	<i>Yes, URL: http://protein.bio.unipd.it/espritz/</i>
Data: DisProt disorder	Provenance	<i>DisProt version 3.7 (January 2008) for training set, DisProt version 5.7 for test set. 536 proteins total. $N_{pos} = 63,841$ residues. $N_{neg} = 164,682$ residues. Not previously used.</i>
	Dataset splits	<i>$N_{pos,train} = 56,414$. $N_{neg,train} = 163,010$. $N_{pos,test} = 7,427$. $N_{neg,test} = 1,672$ residues. No separate validation set. 25.71% positives on training set. 41.04% positives on test set.</i>
	Redundancy between data splits	<i>Maximum pairwise identity within and between training and test set is 40% enforced with UniqueProt tool. Less stringent threshold used to maintain adequate dataset size.</i>
	Availability of data	<i>Yes, URL: http://protein.bio.unipd.it/espritz/</i>
Data: NMR disorder	Provenance	<i>Protein Data Bank (PDB) NMR structures until May 2008 (training) and from May 2008 until September 2010 (test) analyzed using the Mobi software. 2,858 proteins total. $N_{pos} = 40,368$ residues. $N_{neg} = 192,170$ residues. Not previously used.</i>
	Dataset splits	<i>$N_{pos,train} = 29,263$. $N_{neg,train} = 143,891$. $N_{pos,test} = 11,105$. $N_{neg,test} = 48,279$ residues. No separate validation set. 16.9% positives on training set. 18.7% positives on test set.</i>
	Redundancy between data splits	<i>Maximum pairwise identity within and between training and test set is 25% enforced with UniqueProt tool.</i>
	Availability of data	<i>Yes, URL: http://protein.bio.unipd.it/espritz/</i>
Optimization	Algorithm	<i>BRNN (Bi-directional recurrent neural network) with ensemble averaging.</i>

	Meta-predictions	No.
	Data encoding	<i>Sliding window of length 23 residues on input sequence with “one hot” encoding (i.e. 20 inputs per residue).</i>
	Parameters	<i>ESpritz $p = 4,647$ to $5,886$ depending on model used. No optimization.</i>
	Features	<i>ESpritz $f = 460$ for sliding window of 23 residues with 20 inputs per residue. No feature selection.</i>
	Fitting	<i>The number of training examples is between 30 and 100 times p, suggesting neither over- nor under-fitting.</i>
	Availability of configuration	No.
Model	Interpretability	<i>Black box, as correlation between input and output is masked. No attempt was made to make the model transparent.</i>
	Execution time	<i>ESpritzS ca. 1 sec / protein, ESpritzP ca. 1,500 sec / protein on a single Intel Xeon core.</i>
	Availability of software	<i>Web server, URL: http://protein.bio.unipd.it/espritz/ Linux executable, URL: http://protein.bio.unipd.it/download/</i>
Evaluation	Evaluation method	<i>Independent datasets.</i>
	Performance measures	<i>Accuracy, sensitivity, specificity, selectivity, F-measure, MCC, AUC are standard. $S_w = \text{Sens} + \text{Spec} - 1$.</i>
	Comparison	<i>Disopred, MULTICOM, DisEMBL, IUpred, PONDR-FIT, Spritz, CSpritz. Wide range of popular predictors used for comparison.</i>
	Confidence	<i>Bootstrapping was used to estimate statistical significance as in CASP-8 (Noivirt-Brik et al, Proteins 2009). 80% of target proteins were randomly selected 1000 times, and the standard error of the scores was calculated (i.e. $1.96 \times \text{standard_error}$ gives 95% confidence around mean for normal distributions).</i>
	Availability of evaluation	No.

The following is an example for meta-predictor ML summary built from (Necci et al., Bioinformatics 2017).

Data	Provenance	<i>Protein Data Bank (PDB). X-ray structures missing residues. $N_{\text{pos}} = 339,603$ residues. $N_{\text{neg}} = 6,168,717$ residues. Previously used in (Walsh et al., Bioinformatics 2015) as an independent benchmark set.</i>
	Dataset splits	<i>training set: N/A $N_{\text{pos,test}} = 339,603$ residues. $N_{\text{neg,test}} = 6,168,717$ residues. No validation set. 5.22% positives on test set.</i>
	Redundancy between data splits	<i>Not applicable.</i>

	Availability of data	Yes, URL: http://protein.bio.unipd.it/mobidblite/
Optimization	Algorithm	Majority-based consensus classification based on 8 primary ML methods and post-processing.
	Meta-predictions	Yes, predictor output is a binary prediction computed from the consensus of other methods; Independence of training sets of other methods with test set of meta-predictor was not tested since datasets from other methods were not available.
	Data encoding	Label-wise average of 8 binary predictions.
	Parameters	$p = 3$ (Consensus score threshold, expansion-erosion window, length threshold). No optimization.
	Features	Not applicable.
	Fitting	Single input ML methods are used with default parameters. Optimization is a simple majority.
	Availability of configuration	Not applicable.
Model	Interpretability	Transparent, in so far as meta-prediction is concerned. Consensus and post processing over other methods predictions (which are mostly black boxes). No attempt was made to make the meta-prediction a black box.
	Execution time	ca. 1 second per representative on a desktop PC.
	Availability of software	Yes, URL: http://protein.bio.unipd.it/mobidblite/
Evaluation	Evaluation method	Independent dataset
	Performance measures	Balanced Accuracy, Precision, Sensitivity, Specificity, F1, MCC.
	Comparison	DisEmbl-465, DisEmbl-HL, ESpritz Disprot, ESpritz NMR, ESpritz Xray, Globplot, IUPred long, IUPred short, VSL2b. Chosen methods are the methods from which the meta prediction is obtained.
	Confidence	Not calculated.
	Availability of evaluation	No.