Applicant Name: Zhang Wan

Date: 2021/11/28

Tick the appropriate box:

- ☐ Citizen
- ☐ Permanent Resident
- ☑ Student Pass
- ☐ Others, please specify: _____

Instructions to Applicants

Please read these submission instructions carefully. Submissions not following the below instructions will be rejected.

1. Load your answers for Q1 – 3 in **Github repository**.
2. Put all results/write-up into Github, organize your Github project into folders.
3. One folder for each question.
4. For your submission, ensure that you give ***one and only one link*** to your Github project and give this link explicitly in your response email to the school administrator.

   **You should ensure that Github links embedded in your documents are identifiable. Make sure you send your links clearly and check with the school Administrator if the links are received well.*

   Please indicate the questions you attempted.

Questions	Yes/No
1	Yes
2	Yes
3	Yes

5. Indicate your Github link here and in the email to school Administrator.

Github Link	https://github.com/ZhangWan-byte/NTU_BioDS

6. Submit this document (pdf) in Github

Declaration

1. I hereby declare the entirety of the assessment was completed by myself.
2. All online materials and resources used in the completion of this assessment are cited.
3. I agree not to share the questions with others or post the questions on public forums

Signature: Zhang Wan

Name: Zhang Wan

Programming Interview

Question 1:

Read the following paper
https://www.biorxiv.org/content/10.1101/2021.07.08.451443v1.abstract and write a report limited to 3 pages, including citations, figures etc. Reports beyond 3 pages will be rejected automatically.

Implement a simpler version of the method using the MNIST data set for regression on digit 0 and digit 7. Each bag consists of 100 images with a fraction x of digit 0 and 1-x of digit 7. Then train neural network on regression using the neural network architecture specified in the given paper.

Report and code is graded based on:

1. Clarity
2. Show of understanding in the biological and cancer domain knowledge
3. Show of understanding in the machine learning technology
4. Generate results on the MNIST toy data set – graphs and plots to show that your code is working
5. Put a version of your source code in github, code is graded based on:
    a. Good code design
    b. Good coding habits
    c. Correctness

Question 2: Enzyme Kinetics

Enzymes are catalysts that help convert molecules that we will call substrates into other molecules that we will products. They themselves are not changed by the reaction. Within cells, enzymes are typically proteins. They can speed up biological reactions, sometimes by up to millions of times. They are also regulated by a very complex set of positive and negative feedback systems. Computational biologists are painstakingly mapping out this complex set of reactions. In this problem, we will model and simulate a simplified enzyme reaction.

An enzyme *E* converts the substrate *S* into the product *P* through a two-step process. First, *E* forms a complex with *S* to form an intermediate species *ES* in a reversible manner at the forward rate *k1* and reverse rate *k2*. The intermediate *ES* then breaks down into the product *P* at a rate *k3*, thereby releasing *E*. Schematically, we write

$$E + S \underset{k_2}{\overset{k_1}{\rightleftharpoons}} ES \overset{k_3}{\rightarrow} E + P$$

8.1. Using the law of mass action, write down four equations for the rate of changes of the four species, *E*, *S*, *ES*, and *P*.

8.2. Write a code to numerically solve these four equations using the fourth-order Runge-Kutta method. For this exercise, assume that the initial concentration of *E* is 1 μM, the initial concentration of *S* is 10 μM, and the initial concentrations of *ES* and *P* are both 0. The rate constants are: *k1*=100/μM/min, *k2*=600/min, *k3*=150/min.

8.3. We define the velocity, *V*, of the enzymatic reaction to be the rate of change of the product *P*. Plot the velocity *V* as a function of the concentration of the substrate *S*. You should find that, when the concentrations of *S* are small, the velocity *V* increases approximately linearly. At large concentrations of *S*, however, the velocity *V* saturates to a maximum value, *Vm*. Find this value *Vm* from your plot.

Question 3:

Read the below paper and write a report limited to 5 pages, including citations, figures etc. Reports beyond 3 pages will be rejected automatically.

The report should contain whether you think doppelganger effects are unique to biomedical data, and how you think it can be avoided in the practice and development of machine learning models for health and medical science.

Extra points are awarded if [1] you can find interesting examples in other data types e.g. imaging, gene sequencing, metabonomics. [2] Demonstrate clear understanding on how these doppelganger effects emerge from a quantitative angle, and [3] propose interesting and useful ways of avoiding or checking for doppelganger effects.

.

ELSEVIER

Feature

How doppelgänger effects in biomedical data confound machine learning

Li Rong Wang [a], Limsoon Wong [b,c], Wilson Wen Bin Goh [d,e,*]

Machine learning (ML) models have been increasingly adopted in drug development for faster identification of potential targets. Cross-validation techniques are commonly used to evaluate these models. However, the reliability of such validation methods can be affected by the presence of data doppelgängers. Data doppelgängers occur when independently derived data are very similar to each other, causing models to perform well regardless of how they are trained (i.e., the doppelgänger effect). Despite the abundance of data doppelgängers in biomedical data and their inflationary effects, they remain uncharacterized. We show their prevalence in biomedical data, demonstrate how doppelgängers arise, and provide proof of their confounding effects. To mitigate the doppelgänger effect, we recommend identifying data doppelgängers before the training-validation split.

Keywords: Computational biology; Data science; Doppelgänger effect; Machine learning

Introduction

ML models have been increasingly used in drug discovery to speed up drug development. ML increases the efficiency of drug discovery in a multitude of ways: ML models can shortlist better drug candidates (targets) faster, reducing time spent on discovery and testing. They can also identify existing US Food and Drug Administration (FDA)-approved drugs for the treatment of other diseases (drug repurposing), dramatically decreasing the cost of drug development.[1] Both methods have shown promise in recent years. A new anticancer drug candidate, EXS21546, was discovered by Exscientia's 'Centaur Chemist' artificial intelligence (AI) platform after 8 months and is currently undergoing clinical testing (NCT04727138).[2] Several ML-identified drugs and drug combinations for coronavirus 2019 (COVID-19) treatment have also advanced into clinical trials, such as the drug combination of melatonin and toremifene, identified by network-based approaches[3] (NCT04531748) and baricitinib, identified by BenevolentAI's knowledge graph[4] (NCT04373044 and NCT04401579).

Classification models based on ML and AI also increase efficacy in drug development. Classifiers (trained models) have been used for the prediction of new drug–disease interactions[5,6] and possible adverse drug reactions.[7] Given the expensive drug-testing process, it is important that these classifiers are properly trained and tested to identify suitable drug candidates.

It is well established in ML that, when assessing the performance of a classifier, the training and test data sets should be independently derived. However, independently derived training and test sets could still yield unreliable validation results. For example, models trained and validated on data doppelgängers (where training and validation sets are highly similar because of chance or otherwise) might perform well regardless of the quality of training.[8] When a classifier falsely performs well because of the presence of data

https://doi.org/10.1016/j.drudis.2021.10.017

Please cite this article in press as: L.R. Wang et al., Drug Discovery Today (2021), https://doi.org/10.1016/j.drudis.2021.10.017

doppelgängers, we say that there is an observed doppelgänger effect. Earlier, we stated that data doppelgängers are when samples appear similar across their measurements. However, this may not guarantee a doppelgänger effect. Thus, data doppelgängers that also generate a doppelgänger effect (confounding ML outcomes) are termed functional doppelgängers. Despite several documented examples of data doppelgängers (see 'Abundance of data doppelgängers in biological data'), it remains uncommon to check whether the sample training–evaluation pairs are independent and/or dissimilar. Furthermore, data doppelgängers and their accompanying downstream analytical effects (doppelgänger effects) are poorly documented and not well understood. Here, we would like to understand better the level of similarity between suspected functional doppelgängers and the acceptable proportion of functional doppelgängers in the validation set. Although several proposed methods of identifying data doppelgängers exist (see 'Ameliorating data doppelgängers'), most methods are not generalizable or robust enough. Hence, it is imperative to investigate the nature of data doppelgängers and propose improved methods for doppelgänger identification. Using a renal cell carcinoma benchmark data set with appropriately designed controls,[9] we illustrate here the prevalence of functional doppelgängers given this biomedical data, the implications of data doppelgängers on ML, and ways to mitigate the doppelgänger effect.

Abundance of data doppelgängers in biological data

Data doppelgängers have been observed in modern bioinformatics. In one notable case, Cao and Fullwood performed a detailed evaluation of existing chromatin interaction prediction systems.[10] Their work revealed that the performance of these systems has been overstated because of problems in assessment methodologies when these systems were reported. In particular, these systems were evaluated on test sets that shared a high degree of similarity to training sets. The presence of data doppelgängers was also observed by Goh and Wong, whereby certain validation data were guaranteed a good performance given a particular training data, even if the selected features were random.[11] Data doppelgängers are also present in established fields of bioinformatics: in protein function prediction, proteins with similar sequences are inferred to be descended from the same ancestor protein and thereby inherit the function of that ancestor (i.e., the two proteins are presumed to be similar in function). This naïve application of abductive reasoning is true in most cases (cases of data doppelgängers), giving us a false impression of highly accurate predictions. However, on greater inspection, we realize that this approach would be unable to correctly predict functions for proteins with less similar sequences but similar functions, such as twilight-zone homologs[12] and enzymes that are dissimilar in sequence overall but with similar active site residues.[13] A similar example exists in drug discovery: quantitative structure–activity relationship (QSAR) models are classification and regression ML models trained to predict the biological activities of molecules from their structural properties.[14] QSAR models assume that structurally similar molecules have similar activities. In most instances, this assumption is true (cases of data doppelgängers). Sorting similar molecules with similar activities into both training and validation sets (by chance during time-split validation or random test set selection)[15] confounds model validation because poorly trained models (trained on uninformative structural properties) might still perform well on these molecules.[16] We can only differentiate poorly trained models from their well-trained counterparts by testing their performance on similar molecules with different activities (SAR paradox). If the paradox is the result of small variations in structure that substantially impact binding affinity,[17] a well-trained model would theoretically still perform well on these instances given that they are trained on informative structural properties and, hence, able to detect these small variations, whereas a poorly trained model would fail to identify the true biological activity. Although the biomedical data science community appears to be increasingly aware of such data doppelgänger problems, it is surprising that procedures for eliminating or minimizing similarity between test and training data still do not constitute standard practice before classifier evaluation.

Identification of data doppelgängers

Given the potential of doppelgänger effects to confound, it is crucial to be able to identify the presence of data doppelgängers between training and validation sets before validation. One logical approach to data doppelgänger identification would be to use ordination methods (e.g., principal component analysis) or embedding methods (e.g., t-SNE), coupled with scatterplots, to see how samples are distributed in reduced-dimensional space. However, we found such method to be unfeasible because data doppelgängers are not necessarily distinguishable in reduced-dimensional space (Fig. S1 in the supplemental information online).

Earlier studies working on similar problems have also proposed measures for identifying data doppelgängers. One method, dupChecker, identifies duplicate samples by comparing the MD5 fingerprints of their CEL files.[18] Identical MD5 fingerprints would suggest that samples are duplicates (essentially replicates and, therefore, indicative of leakage issues). Thus, dupChecker does not detect true data doppelgängers that are independently derived samples that are similar by chance. Another measure, the pairwise Pearson's correlation coefficient (PPCC), captures relations between sample pairs of different data sets.[19] An anomalously high PPCC value indicates that a pair of samples constitutes PPCC data doppelgängers (note that it is impossible to determine which one between the pair is the original). Although reasonable and intuitive, the prime limitation of the original PPCC paper was that it never conclusively made a link between PPCC data doppelgängers and their ability to confound ML tasks (i.e., having a functional effect and, therefore, acting as functional doppelgängers). We also realized during reanalysis of their data that their reported doppelgängers were in fact the result of leakage (between sample replicates) and, therefore, do not constitute true data doppelgängers. However, the basic design of PPCC as a quantitation measure is reasonable methodologically. Thus, we use this for identifying potential functional doppelgängers (from PPCC data dop-

Please cite this article in press as: L.R. Wang et al., Drug Discovery Today (2021), https://doi.org/10.1016/j.drudis.2021.10.017

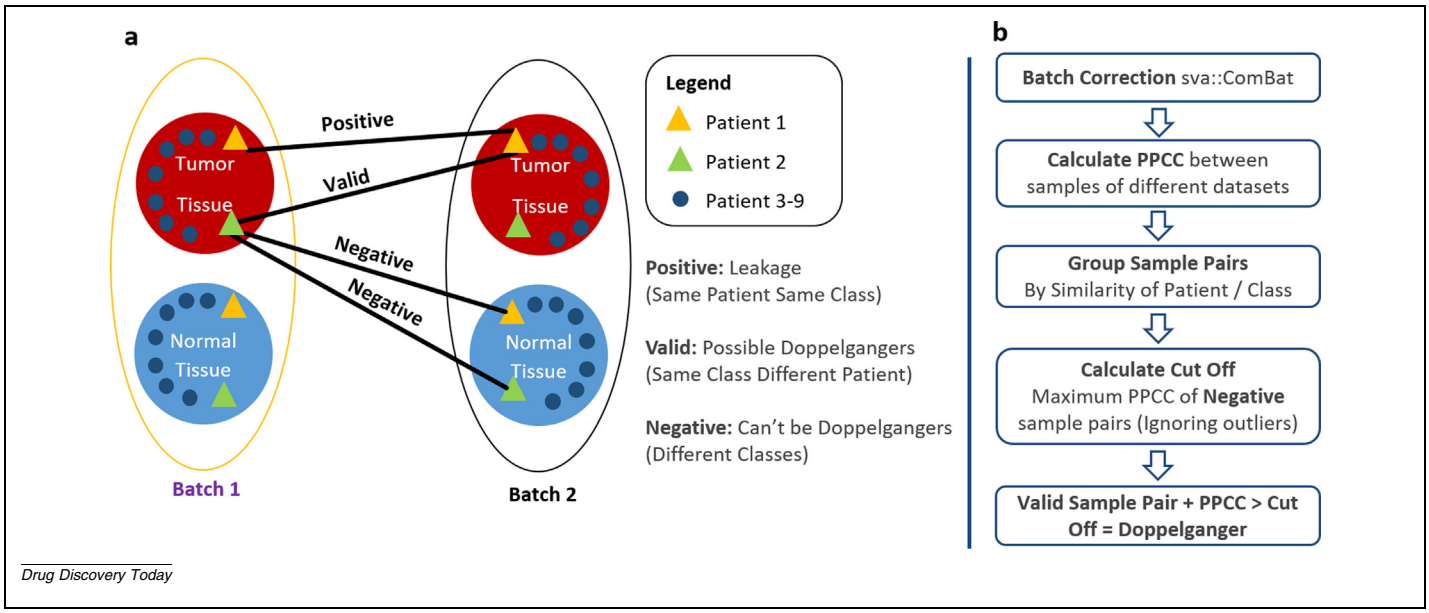


FIGURE 1

Diagram illustrating the pairwise Pearson's correlation coefficient (PPCC) data doppelgänger identification method. (a) Naming convention for different types of sample pair based on the similarities of their patient and class. (b) Process of PPCC data doppelgänger identification. PPCC data doppelgängers are defined as valid sample pairs with PPCC values greater than all negative sample pairs.

pelgängers) from constructed benchmark scenarios.

To construct benchmark scenarios, we used the renal cell carcinoma (RCC) proteomics data of Guo et al.[9] taken from the NetProt software library[20] (see supplementary methods in the supplemental information online). RCC was chosen for its utility in constructing clear-cut scenarios: (i) negative cases, in which doppelgängers are nonpermissible by constructing samples pairs of different class labels; (ii) valid cases, in which doppelgängers are permissible by constructing sample pairs assigned to the same class label but from different samples. These effects can then be compared against positive cases (pairs constructed by taking technical replicates arising from the same sample; these constitute obvious leakage issues and, therefore, are not considered doppelgängers) (Fig. 1a). We simulate these scenarios across the two batches of the RCC data set (see supplementary methods in the supplemental information online).

We identified PPCC data doppelgängers based on the PPCC distribution of the valid scenario against the negative and positive scenarios. Surprisingly, we observed a high proportion of PPCC data doppelgängers (half of the samples are PPCC data doppelgängers with at least one other sample; Fig. 2c). PPCC distributions on the valid scenario exist as a wide continuum, without obvious breaks. This suggests that using outlier detection methods (as recommended in the original PPCC paper) will not be sensitive enough. It also suggests that data doppelgängers exist naturally as part of the similarity spectrum between samples (and are not spectacular anomalies). As for why this happens, we cannot say for sure whether this is a problem because of PPCC itself or because the transcriptional profile of genes is, for the most part, positively correlated.[21] We checked PPCC distributions between same and different tissue pairs (Fig. 2b). PPCC values for same tissue pairs remain high overall, suggesting high correlations between samples, even if they come from different patients. This is not surprising, given how many genes share common regulators. However, PPCC distributions are assuredly lower if we compare different tissue pairs in which a class effect must also exist. By contrast, PPCCs are also extremely high when we consider replicates from the same sample or tissue. These evaluations suggest that PPCC has meaningful discrimination value.

Confounding effects of PPCC data doppelgängers

After identifying PPCC data doppelgängers in RCC, we explored their effects on validation accuracy across different randomly trained classifiers (a trained classifier is an ML model that has 'learnt' from training data). This would determine whether PPCC data doppelgängers act as functional doppelgängers, having an obvious inflationary effect on ML performances (see supplementary methods in the supplemental information online).

We noted that the presence of PPCC data doppelgängers in both training and validation data inflates ML performance, even if the features are randomly selected (and, therefore, meaningless; in other words, the models should perform poorly during validation). This finding is consistently reproducible on different sets of training and validation data (Fig. 3) and on different ML models. Moreover, the more doppelgänger pairs represented in both training and validation sets, the more inflated the ML performance. This points toward a dosage-based relationship between the number of PPCC data doppelgängers and the magnitude of the doppelgänger effect. When the validation accuracy for all properly trained models

Please cite this article in press as: L.R. Wang et al., Drug Discovery Today (2021), https://doi.org/10.1016/j.drudis.2021.10.017

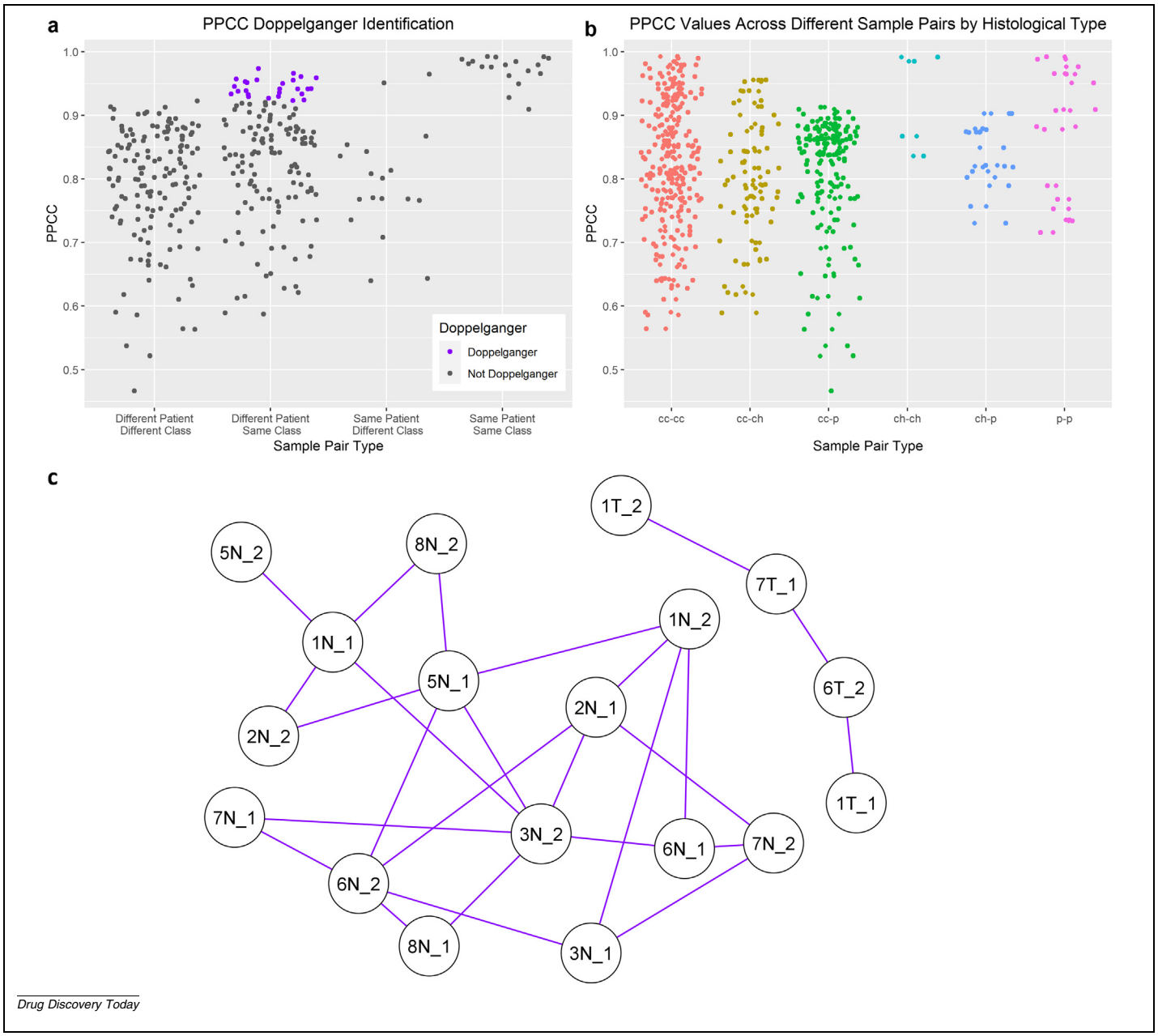
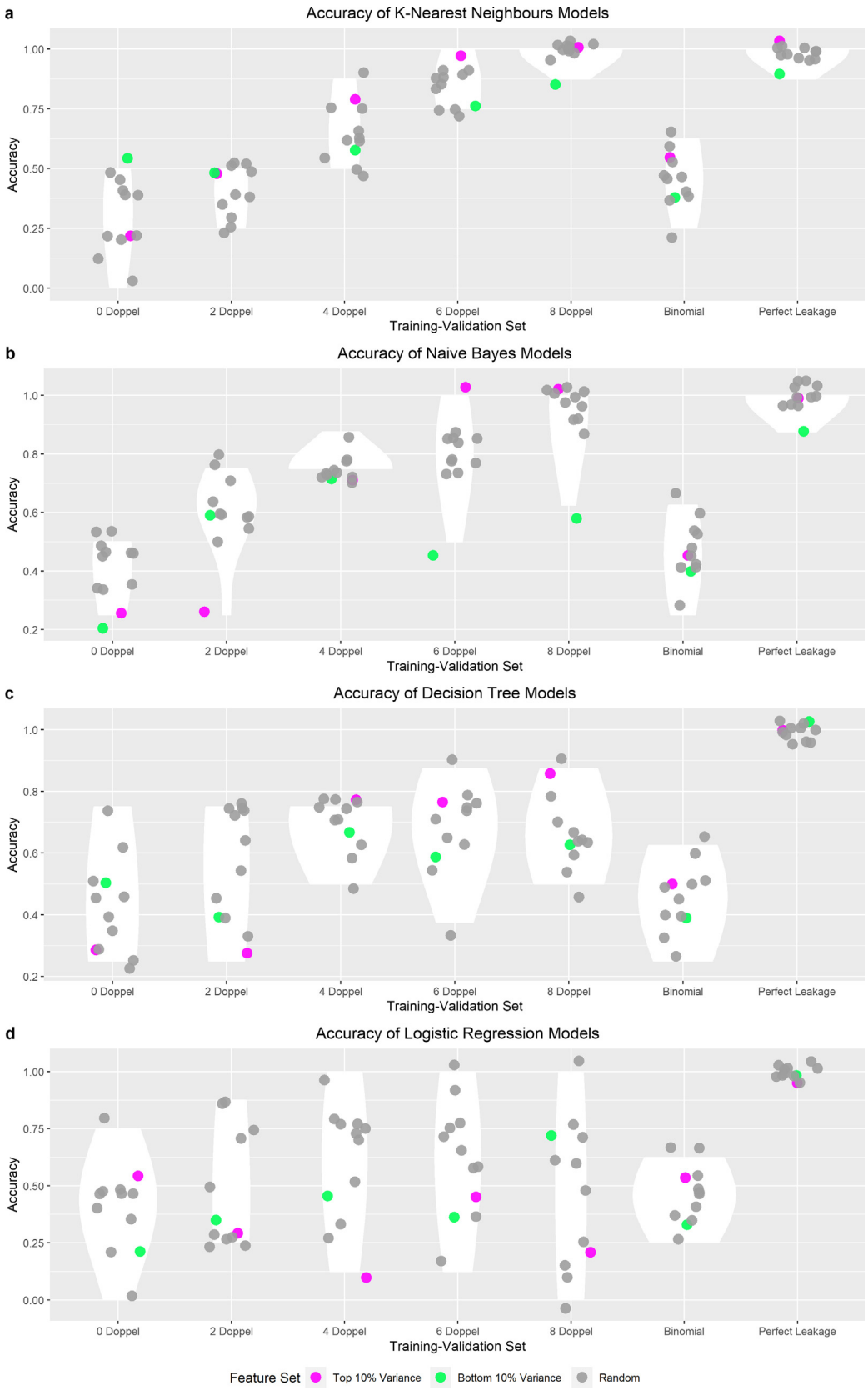


FIGURE 2

(a) Distribution of pairwise Pearson's correlation coefficients (PPCCs) across different sample pairs. The X-axis indicates the types of sample pair grouped by the similarities of their patient and class. The Y-axis indicates the PPCC (i.e., Pearson's correlation coefficient between two samples). The 26 PPCC data doppelgängers are labeled in purple. (b) Distribution of PPCC values of different sample pairs by their histological types. X-axis indicates the types of sample pairs grouped by histological type pairs. Clear cell renal cell carcinoma (RCC) is indicated by cc, chromophobe RCC by ch, and papillary RCC by p. Y-axis indicates the PPCC. (c) 26 PPCC data doppelgängers visualised as a graph. Each node represents a different sample, the first number indicates the patient number, the following letter represents the class (N, normal; T, tumor), the number after the hyphen represents the batch of the sample (e.g., '5N_2' indicates the second replicate of a normal sample from the fifth patient). The presence of an edge between each node/sample means that the two samples are PPCC data doppelgängers. There are 18 nodes in this graph (i.e., 18 samples out of a total of 36 samples are doppelgängers with at least one other sample).

(with 'Top 10% Variance' feature set) on 'Doppel 4' (training-validation set) were stratified into PPCC data doppelgängers and non-PPCC data doppelgängers strata, all ML models showed higher performance on PPCC data doppelgängers than on non-PPCC data doppelgängers (Table S2 in the supplemental information online). This result mirrors our earlier point regarding protein sequence function predictions: where there are many similar examples (many data doppelgängers), good accuracy is easily obtained without in fact assuring generalizability to less-similar examples. However, where there are few similar examples (few data doppelgängers), gaps in the model are revealed and, thus, the model tends to underperform.

Please cite this article in press as: L.R. Wang et al., Drug Discovery Today (2021), https://doi.org/10.1016/j.drudis.2021.10.017



Drug Discovery Today

Please cite this article in press as: L.R. Wang et al., Drug Discovery Today (2021), https://doi.org/10.1016/j.drudis.2021.10.017

This result confirms that PPCC data doppelgängers (based on pairwise correlations) act as functional doppelgängers (confounds ML outcomes), producing inflationary effects similar to data leakage. The similarities between doppelgänger effects and leakage are evident in our experiment using k-nearest neighbor (kNN) models in which the training-validation set with eight doppelgängers in validation showed an identical accuracy distribution to the training-validation set with perfect leakage (Fig. 3a). However, not all models are equally affected: kNN and naïve bayes models have a clearer linear relationship between performance inflation and doppelgänger dosage compared with decision tree and logistic regression models.

By placing all doppelgängers in the training set, accuracies drop to ~0.5, which is the expected accuracy of a model trained on random signatures. Obviously, when all PPCC data doppelgängers are placed together in the training set, the doppelgänger effect is eliminated. This provides a possible way of avoiding the doppelgänger effect. However, constraining the PPCC data doppelgängers to either the training or validation set are suboptimal solutions. In the former, when the size of training set is fixed (thus, each data doppelganger that gets included causes a less similar sample to be excluded from the training set), it leads to models that might not generalize well because the model lacks knowledge. In the latter, you might end up with spectacular winner-takes-all scenarios (the doppelgängers will all either be predicted correctly or wrongly.)

Ameliorating data doppelgängers

Data doppelgängers produce undesirable inflationary effects on ML. This raises the question of how doppelgänger effects could be managed. In the previous section, we argued that enforced colocation of doppelgängers in either training or validation sets are suboptimal solutions. Cao and Fullwood called for more comprehensive and rigorous assessment strategies, based on the particular context of the data being analysed.[10] This could be achieved by splitting training and test data based on individual chromosomes (instead of considering all chromosomes together), as well as using different cell types to generate the training–evaluation pair, thus establishing a good practice/standard in the field. However, this is difficult to do practically because it predicates on the existence of prior knowledge and good quality contextual/benchmarking data.

In studies in which the PPCC outlier detection package, doppelgangR (see 'Identification of data doppelgängers') was used for the identification of doppelgängers, PPCC data doppelgängers could be removed to mitigate their effects.[22,23] However, this approach does not work on small data sets with a high proportion of PPCC data doppelgängers, such as RCC, because the removal of PPCC data doppelgängers would reduce the data to an unusable size (as for what the prevalence of data doppelgängers across all instances of biomedical data is, we consider this would be a consortium level effort, which might be necessary given how the effect of doppelgängers poses a clear obstacle to good-quality biomedical ML models).

Ergo, we also attempted to alleviate doppelgänger effect with methods that would not lead to a significant reduction in sample size or require a high amount of contextual data, although our attempts have met with failure thus far. For example, we attempted data trimming by removing variables contributing strongly toward data doppelgängers effects (Fig. S2 in the supplemental information online). However, we observed no change in the inflationary effects of the PPCC data doppelgängers after the removal of correlated variables. This observation hints at the extreme complexity of the doppelgänger effect, given that the reason for high correlations between sample pairs cannot simply be explained by a subset of highly correlated variables. We are now looking toward novel feature engineering and normalization approaches to address this is, which could prove more successful.

Recommendations

Although removing data doppelgängers from data directly has proven elusive, we still need to guard against doppelgänger effects.

Our first recommendation is to perform careful cross-checks using meta-data as a guide. Here, we used the meta-data in RCC for constructing negative and positive cases. This allowed us to anticipate PPCC score ranges for scenarios in which doppelgängers cannot exist (different class; negative cases) and where leakage exists (same-patient and same-class based on replicates; positive cases). The plausible data doppelgängers that warrant concern are samples arising from same class but different patients. With this information from the meta-data, we are able to identify potential doppelgängers and assort them all into either training or validation sets, effectively preventing doppelgänger effects, and allowing a relatively more objective evaluation of ML performance. In a similar vein, technical replicates arising from the same sample should also be dealt with similarly. This recommendation parallels guidelines in bioinformatics that suggest, when ML models are trained on

FIGURE 3

The prediction performance of different machine learning (ML) models on pairs of training-validation sets with varying numbers of pairwise Pearson's correlation coefficient (PPCC) data doppelgängers in the validation set. ML models assessed includes: k-nearest neighbor (kNN) (a), naïve bayes (b), decision tree (c), and logistic regression (d) models. X-axis indicates the type of validation set: 'i Doppel' refers to a validation set with *i* number of PPCC data doppelgängers in the training set (where $i = 0, 2, 4, 6,$ and 8), 'Binomial' refers to the accuracies generated by 12 (number of feature sets) binomial distributions with $N = 8$ (because there are eight samples in the validation set) and $P = 0.5$ (probability of guessing the correct label for each validation sample) (negative control), 'Perfect Leakage' refers to a validation set with eight duplicates with the training set (positive control). Y-axis indicates the accuracy of ML models on a validation set of eight samples with the lowest accuracy being 0 and the highest accuracy being 1. 'Top 10% Variance' refers to the feature set comprising proteins of the highest variance (i.e., top 10% among the total number of proteins in the data set). 'Bottom 10% Variance' refers to the feature set comprising proteins of the lowest variance at 10% of the total number of proteins in the data set, 'Random' refers to the feature set comprising randomly select proteins at 10% of the total number of proteins in the data set.

Please cite this article in press as: L.R. Wang et al., Drug Discovery Today (2021), https://doi.org/10.1016/j.drudis.2021.10.017

data derived from biological sequences, researchers should ensure that training and test samples are not duplicates or samples of high similarity.[24]

Our second recommendation is to perform data stratification. Instead of evaluating model performance on whole test data, we can stratify data into strata of different similarities (e.g., PPCC data doppelgängers and non-PPCC data doppelgängers, and evaluate model performance on each stratum separately). Assuming each stratum coincides with a known proportion of real-world population, we are still able to appreciate the real-world performance of the classifier by considering the real-world prevalence of a stratum when interpreting the performance at that stratum. More importantly, strata with poor model performance pinpoint gaps in the classifier. In RCC, the non-PPCC doppelgängers used in stratified performance assessment also happen to be papillary RCC samples. Given that the proportion of kidney cancer cells of each tissue is known (papillary RCC comprises 10% of kidney cancer cells),[25] the poor performance of the classifier on papillary RCC would indicate that this 10% of kidney cancer cell samples is an area of weakness for our classifier, which would require further improvements.[21]

Our third recommendation is to perform extremely robust independent validation checks involving as many data sets as possible (divergent validation).[8] Although not a direct hedge against data doppelgängers, divergent validation techniques can inform on the objectivity of the classifier. It also informs on the generalizability of the model (in terms of real-world usage) despite the possible presence of data doppelgängers in the training set.

Future research could explore other methods of functional doppelgänger identification that do not rely heavily on metadata. In such approaches, we could identify functional doppelgängers directly. For example, we might look for subsets of a validation set that are predicted correctly regardless of the ML method used. These subsets are potential functional doppelgängers of the training set (sample pairs between training and validation sets that inflate model accuracies regardless of how we train the model). Further pairing this approach with PPCC subsequently may allow us to discern the doppelgänger partners of test set samples in the training set (or, conversely, the interesting question of whether dissimilar sets, or nondata doppelgängers, can also act as functional doppelgängers). During model evaluation, these subsets should be avoided because they act as functional doppelgängers, and give little insight into the relative performance of different models.

Concluding remarks

ML model performance is usually assessed by testing the accuracy of the model on validation data. This approach toward model validation is only valid if validation data are independent from the training data. However, this assumption is usually assumed to be true with no prior checks. This widely held assumption might not hold true in the presence of doppelgänger effects. We find that doppelgängers are fairly common in our test data, and that it has a direct inflationary effect on ML accuracy. This, in turn, reduces the usefulness of ML for phenotype analysis and subsequent identification of potential drug leads. We also noted that the extent of this inflationary effect varies depending on two main factors: the similarity of functional doppelgängers and the proportion of functional doppelgängers in the validation set. Unfortunately, doppelgänger effects are not easy to resolve analytically. Therefore, to avoid performance inflation, it is important to check for potential doppelgängers in data before assortment in training and validation data.

Author contributions

L.R.W. implemented analyses and wrote the manuscript. L.W. provided critical feedback. W.W.B.G. supervised and co-wrote the manuscript.

Acknowledgments

This research/project is supported by the National Research Foundation, Singapore under its Industry Alignment Fund – Prepositioning (IAF-PP) Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore. W.W.B.G. also acknowledges support from a Ministry of Education (MOE), Singapore Tier 1 grant (Grant No. RG35/20).

Declaration of interests

The authors declare no competing interests.

Appendix A. Supplementary material

Supplementary data to this article can be found online at https://doi.org/10.1016/j.drudis.2021.10.017.

References

1 Y. Zhou, F. Wang, J. Tang, R. Nussinov, F. Cheng, Artificial intelligence in COVID-19 drug repurposing, Lancet Digital Health 2 (2020) e667–e676.
2 Savage N. Tapping into the drug discovery potential of AI. Nature. Published online May 27, 2021. Accessed 2 September 2021. https://doi.org/10.1038/d43747-021-00045-7.
3 F. Cheng, S. Rao, R. Mehra, COVID-19 treatment: combining anti-inflammatory and antiviral therapeutics using a network-based approach, Cleve Clin J Med (2021), https://doi.org/10.3949/ccjm.87a.ccc037 [Published online June 30, 2021].
4 P. Richardson, I. Griffin, C. Tucker, D. Smith, O. Oechsle, A. Phelan, et al., Baricitinib as potential treatment for 2019-nCoV acute respiratory disease, Lancet 395 (2020) e30.
5 J.-Y. Shi, X.-Q. Shang, K. Gao, S.-W. Zhang, S.-M. Yiu, An integrated local classification model of predicting drug-drug interactions via Dempster-Shafer theory of evidence, Sci Rep 8 (2018) 1–11.
6 M. Oh, J. Ahn, Y. Yoon, A network-based classification model for deriving novel drug-disease associations and assessing their molecular actions, PLoS ONE 9 (2014) e111668.
7 Y. Hwang, M. Oh, G. Jang, T. Lee, C. Park, J. Ahn, et al., Identifying the common genetic networks of ADR (adverse drug reaction) clusters and developing an ADR classification model, Mol Biosyst 13 (2017) 1788–1796.
8 S.Y. Ho, K. Phua, L. Wong, W.W.B. Goh, Extensions of the external validation for checking learned model interpretability and generalizability, Patterns 1 (2020) 100129.
9 T. Guo, P. Kouvonen, C.C. Koh, L.C. Gillet, W.E. Wolski, H.L. Röst, et al., Rapid mass spectrometric conversion of tissue biopsy samples into permanent quantitative digital proteome maps, Nat Med 21 (2015) 407–413.
10 F. Cao, M.J. Fullwood, Inflated performance measures in enhancer–promoter interaction-prediction methods, Nat Genet 51 (2019) 1196–1198.
11 W.W.B. Goh, L. Wong, Turning straw into gold: building robustness into gene signature inference, Drug Discov Today 24 (2019) 31–36.
12 M.N. Wass, M.J. Sternberg, ConFunc: functional annotation in the twilight zone, Bioinformatics 24 (2008) 798–806.
13 I. Friedberg, Automated protein function prediction—the genomic challenge, Brief Bioinform 7 (2006) 225–242.
14 D. Paul, G. Sanap, S. Shenoy, D. Kalyane, K. Kalia, R.K. Tekade, Artificial intelligence in drug discovery and development, Drug Discov Today 26 (2021) 80–93.
15 E.N. Muratov, J. Bajorath, R.P. Sheridan, I.V. Tetko, D. Filimonov, V. Poroikov, et al., QSAR without borders, Chem Soc Rev 49 (2020) 3525–3564.
16 A. Cherkasov, E.N. Muratov, D. Fourches, A. Varnek, I.I. Baskin, M. Cronin, et al., QSAR modeling: where have you been? Where are you going to?, J Med Chem 57 (2014) 4977–5010
17 Q. Chen, L. Wu, W. Liu, L. Xing, X. Fan, Enhanced QSAR model performance by integrating structural and gene expression information, Molecules 18 (2013) 10789–107801.

Please cite this article in press as: L.R. Wang et al., Drug Discovery Today (2021), https://doi.org/10.1016/j.drudis.2021.10.017

18 Q. Sheng, Y. Shyr, X. Chen, DupChecker: a bioconductor package for checking high-throughput genomic data redundancy in meta-analysis, BMC Bioinform 15 (2014) 323.
19 L. Waldron, M. Riester, M. Ramos, G. Parmigiani, M. Birrer, The Doppelgänger effect: hidden duplicates in databases of transcriptome profiles, J Natl Cancer Inst 108 (2016) djw146.
20 W.W.B. Goh, L. Wong, NetProt: Complex-based feature selection, J Proteome Res 16 (2017) 3102–3112.
21 D. Venet, J.E. Dumont, V. Detours, Most random gene expression signatures are significantly associated with breast cancer outcome, PLoS Comput Biol 7 (2011) e1002240.
22 K. Lakiotaki, N. Vorniotakis, M. Tsagris, G. Georgakopoulos, I. Tsamardinos, BioDataome: a collection of uniformly preprocessed and automatically annotated datasets for data-driven biology, Database 2018 (2018) bay011.
23 S. Ma, S. Ogino, P. Parsana, R. Nishihara, Z. Qian, J. Shen, et al., Continuity of transcriptomes among colorectal cancer subtypes based on meta-analysis, Genome Biol 19 (2018) 1–14.
24 Bioinformatics. Instructions to Authors (Machine learning). https://academic.oup.com/bioinformatics/pages/instructions_for_authors#General%20Policies. [Accessed October 21, 2021].
25 V.F. Muglia, A. Prando, Renal cell carcinoma: histological classification and correlation with imaging findings, Radiol Bras 48 (2015) 166–174.

Li Rong Wang [a], **Limsoon Wong** [b,c], **Wilson Wen Bin Goh** [d,e,*]

[a] School of Computer Science and Engineering, Nanyang Technological University, Singapore
[b] Department of Computer Science, National University of Singapore, Singapore
[c] Department of Pathology, National University of Singapore, Singapore
[d] Lee Kong Chian School of Medicine, Nanyang Technological University, Singapore
[e] School of Biological Sciences, Nanyang Technological University, Singapore

* *Corresponding author at: Lee Kong Chian School of Medicine, Nanyang Technological University, Singapore.*

Please cite this article in press as: L.R. Wang et al., Drug Discovery Today (2021), https://doi.org/10.1016/j.drudis.2021.10.017