data doppelgänger and doppelgänger effect

When building a machine learning model, the training and testing data sets should be independently derived. **Data doppelgängers** occur when independently derived training set and validation set are very similar to each other, causing models to perform well regardless of how they are trained.

When a classifier falsely performs well because of the presence of data doppelgängers, we say that there is an observed **doppelgänger effect**. The existence of data doppelgängers may not guarantee a doppelgänger effect. However, they will still produce undesirable inflationary effects on ML once doppelgänger effect exist. Therefore, we are looking for a way to manage them.

I think doppelgänger effects are not unique to biomedical data, they are **common in biology**. In biology, structure decides function. Therefore, it's very possible for different molecules having similar function to have similar structure and composition, which easily forms data doppelgänger. 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. For example, antibodies are proteins having specific function. They are all composed by two light chains and two heavy chains with constant region and variable region. Antibodies from the same class share the same constant region, only have different sequence in variable region. Therefore, antibodies are very similar in both sequences and structures. Models established based on antibody data usually suffers from data doppelgänger. Others, like QSAR models which assume that structurally similar molecules have similar activities are also confound by data doppelgänger[1].To make matters worse, variations are also inevitable in biology, I think molecules before and after variation also bring our problems about data doppelgänger.

understanding of confounding effects

If we don't care about the existence of doppelgängers and building prediction models directly, maybe in most instances, models will perform well. But they are still poorly trained models. Because a well-trained model would theoretically perform well even there are some variations on structure or sequence or other things, whereas a poorly trained model would fail to identify the true biological activity. Because the existence of data doppelgängers prevent models from learning the nature of the data.

In addition, I think we can also understand doppelgänger effect in the aspect of feature extraction. Nowadays, we have dozens of ways to transform biological sequences into numeric value, like AAC, CTDT, DDE..., and most of these are based on the calculation of amino acid/nucleic acid properties, for example, AAC method calculates the frequency of occurrence of each amino acid/nucleic acid, TPC method takes the small fragments with fixed length inside the amino acid sequence as the research object, and calculates the frequency of occurrence of these small fragments. Frequency has already been small fraction, therefore, if data doppelgänger occurs, the differences between data will become much slighter. And leads to a confounding result.

Therefore, 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.

research status and difficulties

1. data doppelgängers are abundant and uncharacterized.
2. it's still uncommon to check whether the sample training–evaluation pairs are independent

and/or dissimilar.

3. data doppelgängers and their effects are poorly documented and not well understood.
4. existing methods are not generalizable or robust enough.

how to find data doppelgänger

1. use **ordination methods** or **embedding methods** coupled with scatterplots to see how samples are distributed in reduced-dimensional space.

drawback: It's unfeasible because data doppelgängers are not necessarily distinguishable in reduced-dimensional space.

2. use **dupChecker** to identify duplicate samples by comparing the MD5 finger prints of their CEL files[2]. Identical MD5 fingerprints would suggest that samples are duplicates, which are essentially replicates and, therefore, be indicative of leakage issues.

drawback: dupChecker does not detect true data doppelgängers that are independently derived samples that are similar by chance.

3. use **pairwise Pearson's correlation coefficient** (PPCC) to capture relations between sample pairs of different data sets[3]. An anomalously high PPCC value indicates that a pair of samples constitutes PPCC data doppelgängers.

drawback: 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.

4. use **CD-HIT** to cluster and compare protein or nucleotide sequences. It find out the longest sequences as the representative sequences, and then compare the remaining sequences with them. When the alignment between a sequence and a representative sequence reaches to the self-set similarity threshold, this sequence will be classified to this representative sequence's cluster.

drawback: after finding a qualified representative sequence for the first time, this sequence will not be compared with other representative sequences, which may lead to unreliable results.[4]

examples of finding data doppelgängers and showing their confounding effects

example1. PPCC and kNN model

Use the renal cell carcinoma (RCC) proteomics data of Guo et al.9 taken from the NetProt software library[5]. And they are divided into 3 categories:

1. negative cases: from different classes, can't be doppelgängers,
2. valid cases: from different patients' same class, may be doppelgängers,
3. negative cases: from same patients' same class, belong to leakage.

After that, calculate PPCC between samples of different datasets. And group sample pairs by similarity of patient/class. Then calculate the maximum PPCC of negative sample pairs. PPCC data doppelgängers are defined as valid sample pairs with PPCC values greater than all negative sample pairs.

The result show that there is a high proportion of PPCC data doppelgängers. Half of the samples are PPCC data doppelgängers with at least one other sample.

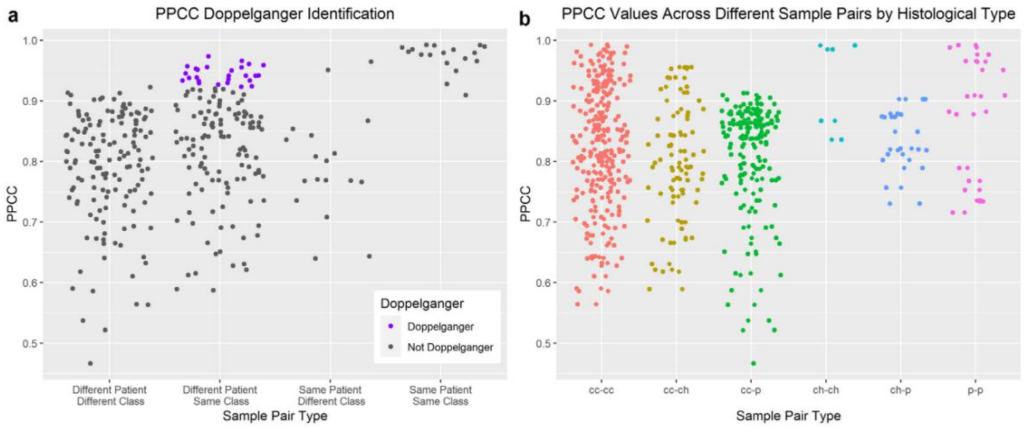


FIGURE 1

(a) Distribution of pairwise Pearson’s correlation coefficients across different sample pairs.
(b) Distribution of PPCC values of different sample pairs by their histological types.

We can see from picture 1b that PPCC values for same tissue pairs remain high overall, suggesting high correlations between samples, even if they come from different patients. PPCCs are also extremely high when we consider replicates from the same sample or tissue. PPCC distributions are assuredly lower if we compare different tissue pairs in which a class effect must also exist. They are all according with common sense, therefore, suggest that PPCC has meaningful discrimination value.

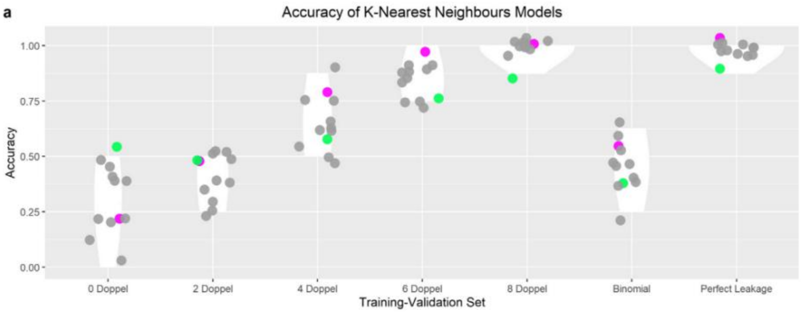


FIGURE 2

using k-nearest neighbor (kNN) models in different data set.

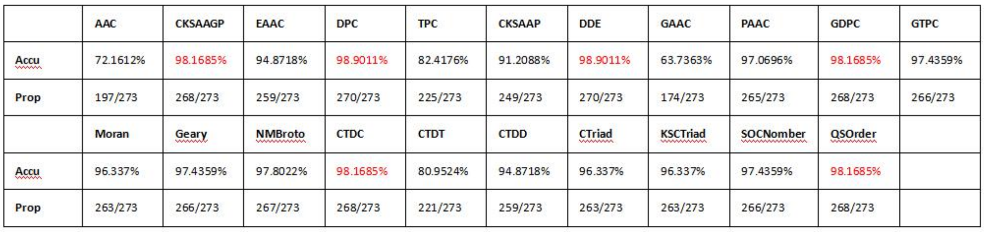
We can see from the picture that PPCC data doppelgängers act as functional doppelgängers (confounds ML outcomes), producing inflationary effects similar to data leakage.

Moreover, the presence of PPCC data doppelgängers in both training and validation data inflates ML performance, even if the features are randomly selected and the models should perform poorly.

In addition, 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.

example2. cdhit and svm model

Data doppelgängers also showed up in one of my project named “prediction of antibodies neutralising SARS-CoV2 based on antibody sequence”.

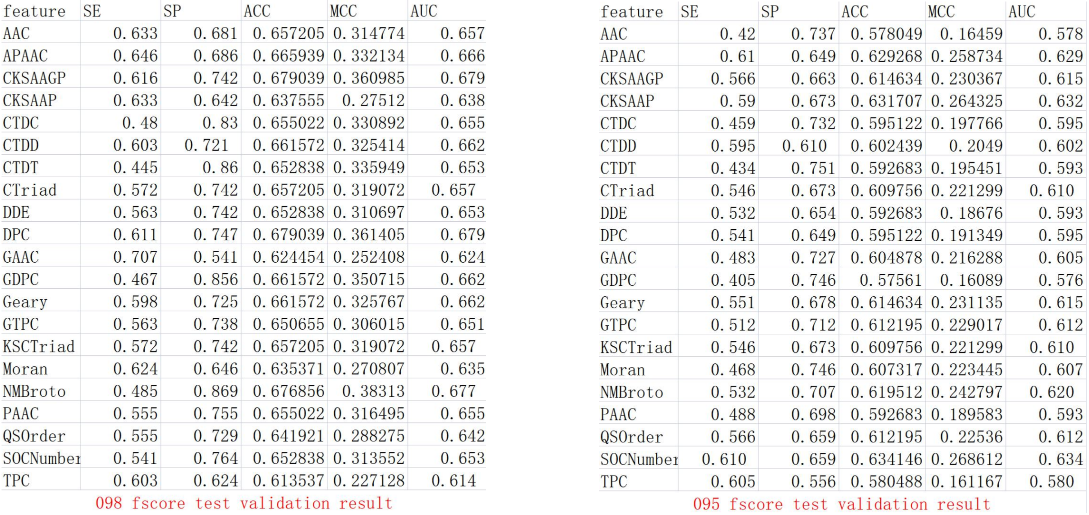


	AAC	CKSAAGP	EAAC	DPC	TPC	CKSAAP	DDE	GAAC	PAAC	GDPC	GTPC
Accu	72.1612%	98.1685%	94.8718%	98.9011%	82.4176%	91.2088%	98.9011%	63.7363%	97.0696%	98.1685%	97.4359%
Prop	197/273	268/273	259/273	270/273	225/273	249/273	270/273	174/273	265/273	268/273	266/273
	Moran	Geary	NMBroto	CTDC	CTDT	CTDD	CTriad	KSCTriad	SOCNomber	QSOrder	
Accu	96.337%	97.4359%	97.8022%	98.1685%	80.9524%	94.8718%	96.337%	96.337%	97.4359%	98.1685%	
Prop	263/273	266/273	267/273	268/273	221/273	259/273	263/273	263/273	266/273	268/273	

FIGURE 3

test validation result(threshold: 1)

Figure 3 is the prediction results of validation set without deleting similar data. The prediction accuracy is high no matter what feature extraction method was used, some even reached to 0.98.



feature	SE	SP	ACC	MCC	AUC
AAC	0.633	0.681	0.657205	0.314774	0.657
APAAC	0.646	0.686	0.665939	0.332134	0.666
CKSAAGP	0.616	0.742	0.679039	0.360985	0.679
CKSAAP	0.633	0.642	0.637555	0.27512	0.638
CTDC	0.48	0.83	0.655022	0.330892	0.655
CTDD	0.603	0.721	0.661572	0.325414	0.662
CTDT	0.445	0.86	0.652838	0.335949	0.653
CTriad	0.572	0.742	0.657205	0.319072	0.657
DDE	0.563	0.742	0.652838	0.310697	0.653
DPC	0.611	0.747	0.679039	0.361405	0.679
GAAC	0.707	0.541	0.624454	0.252408	0.624
GDPC	0.467	0.856	0.661572	0.350715	0.662
Geary	0.598	0.725	0.661572	0.325767	0.662
GTPC	0.563	0.738	0.650655	0.306015	0.651
KSCTriad	0.572	0.742	0.657205	0.319072	0.657
Moran	0.624	0.646	0.635371	0.270807	0.635
NMBroto	0.485	0.869	0.676856	0.38313	0.677
PAAC	0.555	0.755	0.655022	0.316495	0.655
QSOrder	0.555	0.729	0.641921	0.288275	0.642
SOCNumber	0.541	0.764	0.652838	0.313552	0.653
TPC	0.603	0.624	0.613537	0.227128	0.614

098 fscore test validation result

feature	SE	SP	ACC	MCC	AUC
AAC	0.42	0.737	0.578049	0.16459	0.578
APAAC	0.61	0.649	0.629268	0.258734	0.629
CKSAAGP	0.566	0.663	0.614634	0.230367	0.615
CKSAAP	0.59	0.673	0.631707	0.264325	0.632
CTDC	0.459	0.732	0.595122	0.197766	0.595
CTDD	0.595	0.610	0.602439	0.2049	0.602
CTDT	0.434	0.751	0.592683	0.195451	0.593
CTriad	0.546	0.673	0.609756	0.221299	0.610
DDE	0.532	0.654	0.592683	0.18676	0.593
DPC	0.541	0.649	0.595122	0.191349	0.595
GAAC	0.483	0.727	0.604878	0.216288	0.605
GDPC	0.405	0.746	0.57561	0.16089	0.576
Geary	0.551	0.678	0.614634	0.231135	0.615
GTPC	0.512	0.712	0.612195	0.229017	0.612
KSCTriad	0.546	0.673	0.609756	0.221299	0.610
Moran	0.468	0.746	0.607317	0.223445	0.607
NMBroto	0.532	0.707	0.619512	0.242797	0.620
PAAC	0.488	0.698	0.592683	0.189583	0.593
QSOrder	0.566	0.659	0.612195	0.22536	0.612
SOCNumber	0.610	0.659	0.634146	0.268612	0.634
TPC	0.605	0.556	0.580488	0.161167	0.580

095 fscore test validation result

FIGURE 4

test validation result(threshold: 0.98 (left), 0.95 (right))

Figure 4 is the prediction results of validation set after using CD-HIT to delete redundant data. The performance of models decreased significantly. The highest accuracy only reached to 0.67 when threshold was 0.98, while the highest accuracy only reached about 0.63 when threshold was 0.95. We can also know that the accuracy is directly proportional to the removal rate of data doppelgängers: The higher the threshold, the less data doppelgängers are removed, and the higher the accuracy.

How to manage data doppelgängers

1. Put all PPCC data doppelgängers in the training set or testing set can eliminate doppelgänger effects.

drawback: generalize a knowledge-lack model or end up with winner-takes-all scenarios.

2. more comprehensive and rigorous assessment strategies, based on the particular context of the data being analyzed[6].

drawback: it predicates on the existence of prior knowledge and good quality contextual/benchmarking data.

3. use tools like doppelgangR, CD-HIT to remove data doppelgängers[4,7,8].

drawback: when data sets are small and have a high proportion of PPCC data doppelgängers, this will reduce the data to an unusable size.

4. perform careful cross-checks using meta-data as a guide.

5. perform data stratification. Instead of evaluating model performance on whole test data, we can stratify data into strata of different similarities.

6. perform extremely robust independent validation checks involving as many data sets as possible (divergent validation)[9].

citation

[1]. D. Paul, G. Sanap, S. Shenoy, D. Kalyane, K. Kalia, R.K. Tekade, Artifificial intelligence in drug discovery and development, Drug Discov Today 26 (2021) 80–93.

[2]. 18 Q. Sheng, Y. Shyr, X. Chen, DupChecker: a bioconductor package for checking highthroughput genomic data redundancy in metaanalysis, BMC Bioinform 15 (2014) 323. 19

[3]. L.Waldron, M. Riester, M. Ramos, G. Parmigiani, M. Birrer, The Doppelgänger effect: hidden duplicates in databases of transcriptome profifiles, J Natl Cancer Inst 108 (2016) djw146.

[4]. Weizhong Li, Adam Godzik, Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences, Bioinformatics, Volume 22, Issue 13, 1 July 2006, Pages 1658–1659, https://doi.org/10.1093/bioinformatics/btl158

[5]. W.W.B. Goh, L. Wong, NetProt: Complex-based feature selection, J Proteome Res 16 (2017) 3102– 3112.

[6]. F. Cao, M.J. Fullwood, Inflflated performance measures in enhancer–promoter interaction-prediction methods, Nat Genet 51 (2019) 1196–1198.

[7]. 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.

[8]. 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.

[9]. 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.

[10]. Li R , Lwb C , Wwbgd E . How doppelgnger effects in biomedical data confound machine learning. 2021.