BIOINFORMATICS APPLICATIONS NOTE Vol. 26 no. 19 2010, pages 2472–2473
doi:10.1093/bioinformatics/btq463

Gene expression

Advance Access publication August 11, 2010

BDVal: reproducible large-scale predictive model development and validation in high-throughput datasets

Kevin C. Dorff[1], Nyasha Chambwe[1,2], Marko Srdanovic[1] and Fabien Campagne[1,2,*]

[1]Department of Physiology and Biophysics and [2]Institute for Computational Biomedicine, Weill Medical College of Cornell University, New York, NY, USA

Associate Editor: Olga Troyanskaya

ABSTRACT

Summary: High-throughput data can be used in conjunction with clinical information to develop predictive models. Automating the process of developing, evaluating and testing such predictive models on different datasets would minimize operator errors and facilitate the comparison of different modeling approaches on the same dataset. Complete automation would also yield unambiguous documentation of the process followed to develop each model. We present the BDVal suite of programs that fully automate the construction of predictive classification models from high-throughput data and generate detailed reports about the model construction process. We have used BDVal to construct models from microarray and proteomics data, as well as from DNA-methylation datasets. The programs are designed for scalability and support the construction of thousands of alternative models from a given dataset and prediction task.

Availability and Implementation: The BDVal programs are implemented in Java, provided under the GNU General Public License and freely available at http://bdval.campagnelab.org

Contact: fac2003@med.cornell.edu

Received on May 25, 2010; revised on August 5, 2010; accepted on August 6, 2010

1 INTRODUCTION

The technical ability to assay levels of molecules in biological material has evolved dramatically in the last decade. Various technologies now make it possible to assay a large number of features in individual patients. These data can be used to train models to predict information of clinical interest (such as the response of a patient to a given treatment), or a specific biological attribute of a sample. Since it is not clear *a priori* which modeling approach will yield a competitive model for a given problem, several modeling approaches are often compared on the same dataset to yield estimates of future performance for each model that they produce. Performance estimation and model selection is often done with cross-validation (Stone, 1974). Several authors have noted that complete cross-validation must be used to avoid optimistically biasing the estimates of performance that result from the evaluation (Quackenbush, 2004). In complete cross-validation, the feature selection approach is used within each fold of cross-validation to select features used to train the model. Performance estimates can also be biased when choosing a model with the parameters that maximize performance estimated from the training set (Varma, and Simon, 2006). Tibshirani and Tibshirani (2009) have recently described a practical and scalable approach to estimate such a parameter selection bias.

In this application note, we present the BDVal suite of programs, a set of software tools designed to automate the process of developing and evaluating the performance of predictive classification models in high-throughput datasets. BDVal supports several modeling approaches, complete cross-validation and can estimate the Tibshirani and Tibshirani (2009) parameter selection bias.

2 METHODS

Abstractions: BDVal provides useful abstractions to help organize the process of developing models from high-throughput data. These include configurable assay platforms, dataset and endpoint abstractions, as well as reproducible validation plans. The online user manual provides a detailed introduction to these concepts. We provide a brief overview here.

Dataset configuration: the platform abstraction maps assay identifiers (such as probeset identifiers for microarray data) to gene identifiers. Several classification problems (or classification endpoints) can be defined for the same dataset. BDVal can be configured to develop models with a consistent set of approaches across different datasets and endpoints.

Reproducible validation plans: the split plan abstraction makes it possible to reproducibly split a dataset into consistent subsets of samples. Each split is given an identifier, a split index and repeat index. The identifier describes how the samples in the split will be used. For instance, calling a split 'training' indicates that the samples are to be used to train a classifier. The split index ranges from 1 to the number of splits of the training set (10-fold cross-validation will result in 10 splits). The repeat index indicates which splits belong to specific random repeats in an evaluation plan (50 repeats of 10-fold cross-validation will result in 500 splits, with repeat indices from 1 to 50).

Feature aggregation: the program supports aggregating a subset of features. Averaging and projection to principal components are supported for arbitrary subsets of features.

Feature pre-filtering: features can be pre-filtered by providing a gene list. The gene list format requires either assay identifiers or gene identifiers. Pre-filtering a dataset with a gene lists eliminates the features that do not map to the genes or assay identifiers in the gene list without having to create and configure a different version of the dataset. This feature works seamlessly with feature aggregation schemes so that it is possible to filter with aggregated feature identifiers.

Feature scaling: scaling transforms each feature independently of the others to make its scale comparable to that of other features. BDVal implements several scaling schemes. Statistics needed to scale each feature are determined by observing the training samples and recorded in models written to disk to make it possible to scale the test samples consistently.

Feature selection: BDVal implements several feature selection methods. Univariate feature selection methods include fold-change, two-tailed *t*-test

*To whom correspondence should be addressed.

and ranking features by Kendall's tau statistic. Multivariate methods include selection of features by decreasing feature weight of a trained linear support vector machine (SVM-weight method), an implementation of recursive feature elimination (Guyon *et al*., 2002) and selection by genetic algorithm optimization of a cross-validated performance measure (Xiong *et al*., 2001).

Classification: BDVal can train Support Vector Machine classifiers with libSVM, or arbitrary Weka classifiers. Weka classifiers that we have tested include Random Forest, LogitBoost, Logistic Regression or the Naïve Bayes classifier.

Self-contained models: models produced by BDVal are saved to disk in a zip file, with all the information required to apply the model to new data samples. BDVal prediction results include both a decision value (numerical score produced by the classifier, such as the probability that the predicted label is correct) and a human readable unambiguous label. The model also includes the threshold of the decision function. Self-contained models eliminate many sources of possible human errors that can occur when trained models are applied to new data (e.g. label swaps, incorrect prediction threshold, inconsistent scaling parameters). Furthermore, BDVal will produce an error condition when a new sample is missing features that the model was trained with. This situation can arise when the training and validation sets were assayed with platforms that are mostly compatible, but distinct (i.e. different versions of the same microarray chip may have more or less probesets).

Sequence programs: the BDVal sequence files define small programs written in a script-like computer language. Sequence programs make it possible to execute a sequence of operations on a dataset. When BDVal executes a validation plan, the sequence program is repetitively run for each unique combination of split and random repeat indices defined in a split plan. Typical sequences perform feature selection, subsequently train the model with the subset of features (writing the model to disk), and reload the model to predict samples in the test set. Test set predictions are written to disk and associated with split index, repeat index, split identifier and model identifier. Sequence programs implement reproducible model development protocols.

Model identifiers: models are assigned a unique identifier derived from hashing a subset of the command line arguments provided to the sequence program that produced the model. The identifier is a six-letter code that is used to track individual models during development and validation. The code is written to prediction files and statistics output files to make it possible to join tables of information at the model level.

Performance Measures: BDVal supports the following performance measures: Matthews Correlation Coefficients, Area under the ROC curve, Sensitivity, Specificity, Accuracy and their associated standard deviations. When scanning series of parameters for a given classification algorithm, BDVal estimates the Tibshirani and Tibshirani parameter selection bias adjustment. Performance measures can be estimated from test-set prediction files recorded during cross-validation. This decouples performance estimation from the computationally intensive cross-validation model development steps. Performance can also be estimated from prediction files that result from predicting sample labels in datasets other than those used for training. In this case, BDVal supports estimating standard deviation of the performance measures by sampling the validation set with replacement.

Parallel processing: BDVal can run in parallel on a multi-core shared memory computer. In this case, each split of a validation plan is run in parallel, up to the number of threads available. This technique scales for small to medium scale projects (approximately building up to 1000 models). For large-scale projects, we have developed Sun/Oracle Grid Engine scripts to dispatch model evaluation jobs to a grid of computers and collate the results.

3 CONCLUSIONS

The BDVal suite of programs makes it possible to automate most steps of the model development and performance evaluation process, extending the recommendations of Kostka and Spang (2008) from pre-processing steps to the entire model-development process. Such automation helps ensure reproducibility of model development efforts and produces detailed model documentation. We have extensively tested BDVal while developing tens of thousands of predictive models in the MAQC-II high-throughput clinical datasets (MAQC 2010). We anticipate that the BDVal programs and process abstractions implemented in these tools can be helpful when developing predictive models from various types of bioinformatics datasets.

ACKNOWLEDGEMENTS

We thank the members of the MAQC-II consortium for stimulating exchanges.

Funding: National Institutes of Health, Institutional Clinical and Translational Science Award UL1-RR024996; Resources from the David A. Cofrin Center for Biomedical Information at Weill Cornell (to F.C.).

Conflict of Interest: none declared.

REFERENCES

Guyon,I. *et al*. (2002) Gene selection for cancer classification using Support Vector Machines. *Mach. Learn.*, **46**, 389–422.

Kostka,D. and Spang,R. (2008) Microarray based diagnosis profits from better documentation of gene expression signatures. *PLoS Comput. Biol.*, **4**, e22.

Quackenbush,J. (2004) Meeting the Challenges of Functional Genomics: From the Laboratory to the Clinic. *Preclinica*, **2**, 313–316.

Stone,M. (1974) Cross-Validatory Choice and Assessment of Statistical Predictions. *J. R. Stat. Soc. Ser. B (Methodological)*, **36**, 111.

Tibshirani,R.J. and Tibshirani,R. (2009) A bias correction for the minimum error rate in cross-validation. *Appl. Stat.*, **822**, 822–829.

Varma,S. and Simon,R. (2006) Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, **7**, 91.

Xiong,M. *et al*. (2001) Biomarker identification by feature wrappers. *Genome Res.*, **11**, 1878–1887.

MAQC. (2010) The MAQC-II Project: A comprehensive study of common practices for the development and validation of microarray-based predictive models. The MicroArray Quality Control. *Nat. Biotechnol.*, **28**, 827–838.