Information Systems 93 (2020) 101557

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/is

DMAKit: A user-friendly web platform for bringing state-of-the-art data analysis techniques to non-specific users

David Medina-Ortiz [a,b], Sebastián Contreras [a,c], Cristofer Quiroz [d], Juan A. Asenjo [a,b], Álvaro Olivera-Nappa [a,b,*]

[a] Centre for Biotechnology and Bioengineering - CeBiB, Universidad de Chile, Beauchef 851, 8370448 Santiago, Chile
[b] Departamento de Ingeniería Química, Biotecnología y Materiales, Universidad de Chile, Beauchef 851, 8370448 Santiago, Chile
[c] Laboratory for Rheology and Fluid Dynamics, Department of Mining Engineering, Universidad de Chile, Beauchef 850, 8370448 Santiago, Chile
[d] Facultad de Ingeniería, Universidad Autonóma de Chile, Cinco Pte. 1670, Talca, Chile

ARTICLE INFO

Article history:
Received 4 May 2020
Received in revised form 12 May 2020
Accepted 13 May 2020
Available online 16 May 2020
Recommended by Dennis Shasha

Keywords:
Machine learning
Data mining
Pattern recognition
Statistics
User-friendly web platform

ABSTRACT

Tremendous advances in different areas of knowledge are producing vast volumes of data, a quantity so large that it has made necessary the development of new computational algorithms. Among the algorithms developed, we find Machine Learning models and specific data mining techniques that might be useful for all areas of knowledge. The use of computational tools for data analysis is increasingly required, given the need to extract meaningful information from such large volumes of data. However, there are no free access libraries, modules, or web services that comprise a vast array of analytical techniques in a user-friendly environment for non-specific users. Those that exist raise high usability barriers for those untrained in the field as they usually have specific installation requirements and require in-depth programming knowledge, or may result expensive. As an alternative, we have developed DMAKit, a user-friendly web platform powered by DMAKit-lib, a new library implemented in Python, which facilitates the analysis of data of different kind and origins. Our tool implements a wide array of state-of-the-art data mining and pattern recognition techniques, allowing the user to quickly implement classification, prediction or clustering models, statistical evaluation, and feature analysis of different attributes in diverse datasets without requiring any specific programming knowledge. DMAKit is especially useful for users who have large volumes of data to be analyzed but do not have the informatics, mathematical, or statistical knowledge to implement models. We expect this platform to provide a way to extract information and analyze patterns through data mining techniques for anyone interested in applying them with no specific knowledge required. Particularly, we present several cases of study in the areas of biology, biotechnology, and biomedicine, where we highlight the applicability of our tool to ease the labor of non-specialist users to apply data analysis and pattern recognition techniques. DMAKit is available for non-commercial use as an open-access library, licensed under the GNU General Public License, version GPL 3.0. The web platform is publicly available at https://pesb2.cl/dmakitWeb. Demonstrative and tutorial videos for the web platform are available in https://pesb2.cl/dmakittutorials/. Complete urls for relevant content are listed in the Data Availability section.

1. Introduction

Immense and extensive advances in different areas of knowledge are producing vast volumes of data. Although data do not have high relevance by itself, the questions that we can answer by using it certainly do. Usually, users are interested in finding correlations in the data, designing and implementing models to classify or to predict variables that are experimentally difficult, costly, or ethically incorrect to obtain, or extracting information from the data already available. As a result, the Knowledge Discovery in Databases (KDD) process [1], mathematical modeling techniques for complex systems, and data mining and Machine Learning (ML) were born [2]. These techniques have reached a peak and contributed significantly to understanding complex processes in the so-called era of Big Data [3]. However, most of them remain obscure for a non-specialist user, since their use requires knowledge about scientific programming, algorithm optimization, statistics, a correct interpretation of performance metrics, among other challenging issues, thus limiting the access to data science. Besides, applying such techniques in tangential

* Corresponding author at: Centre for Biotechnology and Bioengineering - CeBiB, Universidad de Chile, Beauchef 851, 8370448 Santiago, Chile.
E-mail address: aolivera@ing.uchile.cl (Á. Olivera-Nappa).

https://doi.org/10.1016/j.is.2020.101557
0306-4379/

fields of study would grant great benefits for those enthusiastic researchers keen to take their chances. In the area of biomedicine, where the applications of computational techniques for data analysis are very promising [4–6], researchers call for a new era in the application of Machine Learning [7]. Taking advantage of the -nowadays- easy generation of data, the incorporation of information will be a key feature for success [8], keeping always in mind the challenges that generating useful data, especially in the area of medicine, would imply [9]. Some examples of the applications of ML may be found in studies related to diabetes research [10], cancer diagnosis and treatment [11,12] and decision support in critical care [13], among others.

Software such as WEKA [14], KEEL [15], STATISTICA [16], Torch7 [17], Dlib-ml [18], and others, have been proposed as a support to explore the area of data mining and statistical analysis. However, they do not comprise the totality of state-of-the-art analyses and algorithms. On the other hand, Python libraries such as Open Source Clustering Software [19], Scikit-learn [20] and Pandas [21], R packages such as CARET [22], nnet [23] and Matlab toolboxes, such as the Machine Learning and Neural Network toolboxes, are alternatives for implementing different data mining algorithms or exploring predictive or classification models in a simple way. Notwithstanding their utility, using these tools often requires high programming skills, to install different components, or may seem unfriendly to non-specific users. Moreover, the commercial ones, which might include friendlier interfaces than their open-access counterparts, may result too expensive for low-frequency users.

In this work, we present DMAKit, Data Mining Analysis Kit, a user-friendly web platform, powered by DMAKit-lib, a Python-based commands library, which comprises a vast array of state-of-the-art algorithms and techniques, aiming to allow any user to perform multiple data mining analyses for different types of datasets. This tool was developed to facilitate statistical evaluation, pattern search employing unsupervised learning algorithms, training of models through supervised learning, feature analysis, dimensionality reduction, and linear regressions for users with a non-specialist background. Furthermore, DMAKit is an open-source library under the GNU OpenGL 3.0 license, enabled for non-commercial use, and the web platform is publicly available in pesb2.cl/dmakitWeb. Both its simplicity and accessibility were designed to turn DMAKit into a significant contribution to the field, a fresh, user-friendly, readily accessible, powerful, and free alternative to existing tools and libraries developed for the analysis of data through mining and pattern recognition techniques.

2. Methodology

2.1. Behind the tool: Informatic description

Web programming We designed the DMAKit web platform following the Model-View-Controller (MVC) design pattern and a client–server-based architecture. In general, it is possible to separate the main components of the tool into two: front-end and back-end. The first is mainly based on HTML5, HTML tagging language for visualization, CSS3 for aesthetic improvement, through the Bootstrap 3 framework, and improving the user experience through JavaScript (JS). Besides, various JS plug-ins, such as D3Js, DataTables, Highcharts, among others, were used to add some visualization-related functionalities to the platform, associating this component with the MVC Pattern View component. The back-end mainly represents the controller and model concepts of the MVC pattern. Regarding the controller, it is possible to divide it into two: the client–server communication manager, which is implemented based on Ajax and JQuery, and the request manager, which facilitates the execution of the server actions, obtains and processes the responses returning them to the client, implemented under PHP 7. In both processes, we used the JavaScript Object Notation (JSON) as the data exchange format. Finally, as MVC pattern model we take the DMAKit Python library (DMAKit-Lib), which allows implementing all the modules, features, and functionalities that are available in the tool.

DMAKit-Lib implementation DMAKit-Lib was designed under the Object-Oriented Programming paradigm [24], advantageous to generate the encapsulation and modularity necessary for this type of tool. Its implementation relies on a set of modules written in Python version 2.7. All modules for generating supervised and unsupervised learning models use the Scikit-learn library [20]. Dataset management is performed using the Pandas library [21] and graphics rely on Matplotlib [25] and Seaborn library. Finally, scripts have been generated to allow easy installation of the modules using Disutils Python module.

Server Configuration A server with a Debian 8 operating system hosts the platform, and MySql manages its persistent storage. Finally, task execution processes are queued through the SLURM [26] system, thus optimizing the resources available on the server.

2.2. Selection of case studies

Each of the case studies presented in this article was selected in order to demonstrate the usability and applicability of DMAKit in different fields of biotechnology, protein engineering, and bioinformatics studies. Thus, the datasets were obtained from databases reported in the literature, simulated, or provided by members of our laboratory. The origin of each of them is explained in detail in each case study.

2.3. Data pre-processing in DMAKit

DMAKit selectively implements different dataset processing strategies, depending on which type of model or analysis the user wants to implement. Here we will describe the general data processing procedure.

Null data processing DMAKit receives datasets in which some attributes of the examples may have values of type NA or Null. In order to correctly execute the algorithms or some other type of analysis, DMAKit has a validator of datasets, which allows eliminating from the sample the examples that present values of type NA or Null. Besides, if there are purely numerical attributes and there are examples with categorical values, they are also eliminated.

Categorical attributes coding If the analysis involves the training of classification models or pattern search using clustering algorithms, a transformation of categorical features into continuous values is performed. For this, each categorical feature with N categories is transformed into N new binary features, if the total number of new features does not surpass a default threshold of 1.2 times the number of original features; if it is higher, then each categorical variable is transformed into an integer variable with values from 0 to N-1.

Standardization of datasets The user can alter the default threshold at will. In addition, the user can select whether data should be standardized. DMAKit has three types of data normalization:

(a) Normal scale allows standardization based on the values of the mean and the standard deviation of the distribution
(b) Min–max scaler normalizes according to the maximum and minimum values in the distribution
(c) log and log-normal scalers apply a logarithmic transformation to the data; if there are negative values when applying the transformation they are ignored by default, but they may also be set as any real constant.

2.4. Licensing and how to access DMAKit web tool

Both the computational tool and the DMAKit-Lib library are licensed under the GNU General Public License, version GPL 3.0, which ensures all the advantages and characteristics of Free Software. The source code and sample datasets are available in the GitHub repository https://github.com/dMedinaO/dmakitWeb. To use our tool, the reader should visit pesb2.cl/dmakitWeb/, where users can log-in. In order to use the modules implemented in DMAKit, it is imperative to have an account (which is entirely free). To create it, users must complete a basic form with information, like their email address and institution, and pick a username and password. The requested data is only for statistical purposes, and all the input data is securely stored in the own user workspace. Each workspace is private and independent for each user, ensuring the confidentiality of the results and data arranged to work with DMAKit.

3. Overview of the web-tool

3.1. DMAKit modules

DMAKit has five main modules, which allow the user to (a) evaluate feature relevance using dimensionality reduction techniques, (b) developing statistical analysis of the data, (c) searching for patterns employing unsupervised learning algorithms (d) training classification or regression models through supervised learning algorithms, and (e) linear regression models for predictive analysis (see Fig. 1).

3.1.1. Feature analysis

The feature analysis module allows the evaluation of relations between different descriptive attributes of the dataset, based on a correlation matrix analysis and mutual information techniques. It also implements different dimensionality reduction algorithms based on linear models, such as Principal Component Analysis (PCA) and its variants, Kernel PCA and Incremental PCA. Additionally, it allows the user to evaluate the relevance of attributes in the training of supervised learning models using the Random Forest algorithm, both for classification and prediction of continuous variables. It is possible to use different attributes as a response to evaluate the relevance for the prediction of the remaining features, and the importance they have when developing classification or regression models.

3.1.2. Statistical analysis

Different statistical analyses can be performed on both continuous and discrete type attributes using DMAKit. Among the various features implemented we find distribution analysis, dispersion and frequency evaluation, and visualization of continuous variables using categorical variables, such as scatter plot matrix (SPLOM) and parallel coordinates plots. Both SPLOM and parallel coordinates plots are relevant when analyzing visual patterns, the separation between attributes, or identifying features that will predominate and lead to a better classification in supervised learning models. In addition, statistical tests for different applications are included, such as tests to evaluate the normality of the distribution of the data (Shapiro–Wilk and Kolmogorov–Smirnov tests), a test to compare two distributions (Mann–Whitney test) and tests to evaluate distribution correlations (Pearson's coefficient, Spearman's rank and Kendall's τ tests).

3.1.3. Pattern recognition

We implemented several unsupervised learning algorithms for finding patterns and clusters in datasets, namely k-Means, Birch, DBScan, Mean Shift, Affinity Propagation and Agglomerative and Hierarchical Algorithms. Each clustering model is evaluated by the Calinski–Harabasz index [27] and the silhouette coefficient [28]. The user may generate visualizations of the distribution of elements by cluster to evaluate class imbalance, or to export the input dataset, including the labels associated with the group that was assigned by the different algorithms. Finally, DMAKit generates a summary file with the algorithm, the parameters used for its configuration, the number of groups obtained, and the values of the corresponding evaluation metrics.

3.1.4. Predictive models

We have implemented different supervised learning algorithms in DMAKit for use both in classification and regression models. The validation method for each model can be chosen between k-cross-validation or Leave-One-Out methods. k-cross-validation methods are recommended when each class represents at least a 10% of the sample, with k values ranging from 5 to 10. Otherwise, the use of Leave-One-Out is suggested, notwithstanding its higher computational cost. DMAKit also reports different performance measures according to the model type. For classification models, Precision, Recall, Accuracy, and F1 score, whereas for continuous variable prediction, the Pearson's coefficient, Spearman's rank, Kendall's τ, and R score are reported by the system as performance measures. Distance-based algorithms are also available in this module, such as k-Nearest Neighbor, kernel transformations, and evaluation of dividing hyperplanes using Support Vector Machine (SVM) and νSVM. Feature evaluation methods are also included, such as Decision Trees, assembly, and feature exploration methods such as Random Forest, Bagging, Gradient Boosting and AdaBoost; methods based on probability such as Naïve Bayes; and neural networks such as Multi-Layer Perceptron.

3.1.5. Linear regressions

Different ways of finding a data fit to a linear model have been implemented, including Ordinal Least Square, Ridge Regression, Lasso, Elastic Net, Lars Lasso, and Bayesian Regression, and they stand for numerical and categorical tasks. Like in the predictive models, this module also has implemented different ways of measuring the performance of the model. Finally, the tool reports these measurements and generates a scatter plot of control versus predicted values, and residues histograms.

3.1.6. Models exploration

The model exploration tool in DMAKit allows the simultaneous automatic execution of several algorithms with different parameter combinations applied to the same dataset, in order to evaluate the performance of the resulting models. This option is enabled for supervised learning models, both for prediction and classification tasks, and the clustering through unsupervised learning algorithms. We developed and tested a new methodology included in DMAKit to select and study model performance, generating histograms for each performance measure. A ranking of the best models per measure is proposed, as well as a statistical summary per measure for all executions in the process. The main

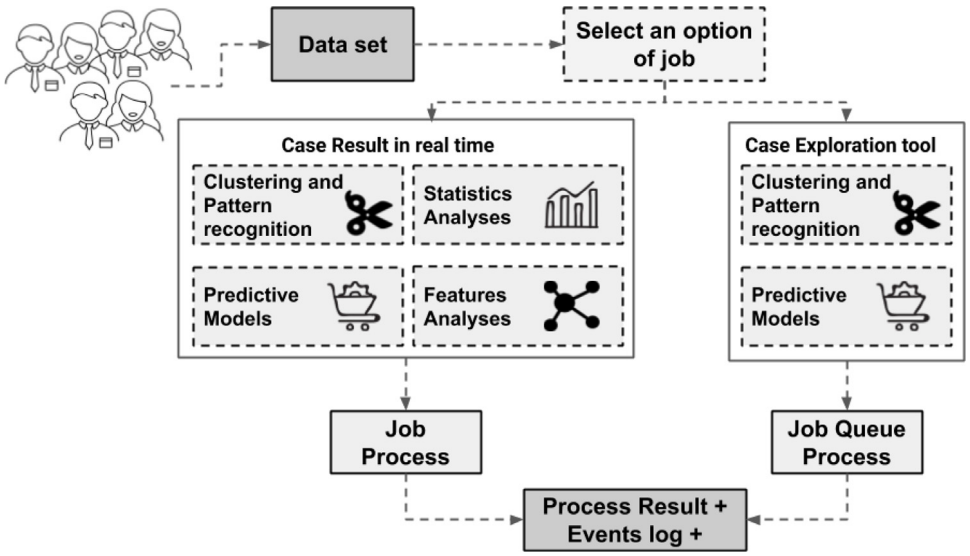

Fig. 1. Graphical abstract of DMAKit's modules and workflow.

advantage of our approach is the ease of evaluating different algorithms for the same dataset, obtaining not only models resulting from different algorithms but those that result from different parameter selection of the same algorithm. The distribution-based visualization and selection allow the user to assess the general panorama of the best models, in the context of the studied dataset and, when developing classification and prediction systems, it may be used to select between different algorithms or parameter combinations in a learning system, aiming to improve its overall performance. Matlab, within its Machine Learning Toolbox, has a tool for the exploration of classification models. However, the number of models to be tested, which typically does not exceed 20, does not cover different parameter values and only uses metrics associated with the model accuracy, not allowing to optimize the model based on precision or specificity or to study the effect of parameter changes on a single algorithm.

3.2. Usability

Applying the different modules detailed above, DMAKit may offer the users a new perspective to analyze and visualize their data, new ways to cast predictions, or design their experimental planning through ML techniques, among other possibilities; the only limit is creativity. Among the different tasks and aims that we may reach using DMAKit, the user may find:

- Recognition of useful patterns and grouping among the data, the meaning of which is left to be clarified by the researcher. For instance, clustering of samples, clustering of experimental results, finding distinct groups within a population, association between response groups and control variable levels, among others.
- Experimental planning optimization. After performing a preliminary factorial search of a multi-variable objective, the user might be concerned about which are the most important parameters to be modified for maximizing the amplitude of the control. DMAKit can provide that answer, through its Feature Analysis module.
- Creation of machine learning predictive models, a useful alternative to traditional mathematical models, especially when there is no further information regarding the mechanisms that generate the targeted response, when the mathematical modeling of the system is too complicated, the co-dependency of the attributes is intricate, or the number of observations is limited. These non-parametric models offer a handy way to predict new examples and optimize experimental planning, creating relevant models for a wide spectrum of data origins.
- Qualitatively learning from the system by analyzing the decision-making path of the Decision Tree models trained. Some empirical relations between attributes may appear, which is especially handy to understand the variables that underlie cluster separation.

4. Discussion

4.1. Testing and recommendations of use

The different existing modules in DMAKit were tested using the UCI Machine Learning publicly available datasets for training and implementation of classification, regression, and clustering models [29]. Additionally, we generated different datasets with biological, clinical, and biotechnological relevance to highlight the usability of our tool. All the input datasets must have CSV format, with a header for each column. If there were examples with Null/NA values, these would be removed. Each one of the datasets used in the Case Studies, workflows, and expected results are available in the tool's GitHub repository.

In case DMAKit is used to design and implement classification models, we recommend to evaluate class imbalance in the first instance using the statistical analysis module. This module includes within its functionalities means to analyze the frequency of categories within an attribute of the categorical type. If there were a category with a very high proportion of the sample, the classifier would tend to classify new examples in such category, causing false-positive type errors. Similarly, when implementing prediction models for numerical attributes, it is recommended to analyze its distribution. If the data do not follow a uniform distribution within the analyzed range, the generated model tends to over-fit the range, which concentrates most of the data, misleading the prediction of new examples within that range. In order to avoid that situation, we recommend standardizing the response variable. However, which strategy to follow to achieve such standardization relies on the experience of the user. Among the possibilities offered by DMAKit, we may find min–max scaler, max-abs scaler, and quantile transformation, among others. In case the dataset presents any categorical attribute, DMAKit will

encode its values. However, if there were only one category for an attribute, it would be tagged as non-informative, and we suggest to remove it.

Before using of the model exploration tool, the user should keep in mind that the application of different algorithms, the use of different datasets, as well as the number of cases and attributes that compose them, are directly related to the computation time involved in the exploration. For a dataset of 250 cases and five attributes, the exploration stage of clustering methods varies between 5 and 15 min. However, for classification models, the computation time is in the range of 120 to 180 min, directly related to the number of iterations involved in the process. For clustering models, considering only about 150 iterations, while for classification or regression methods, at least 2000 iterations are considered. Parallelizing the code to minimize the computational cost is part of our future work.

Regarding the limitations of our web-tool, the user should consider the following:

- Maximum file size: 1 GB.
- Maximum number of examples per dataset: 20 000.
- Maximum number of features per dataset: 2000.
- Maximum waiting time for results: 4 days.

4.2. Comparison with other tools

Table 1 shows a summary of the features and functionalities of DMAKit, quantitatively comparing our tool with other libraries and tools available for similar purposes. The different aspects considered for the comparison are listed below.

Licenses and access DMAKit is licensed under open GPL 3.0, which allows public access to the tool, for non-commercial use. Having the advantages of Free Software is one of the most relevant points when comparing the DMAKit web tool with software such as Matlab and Statistica since both involve obtaining a license for its use.

Machine learning and data mining The supervised learning modules for the development of classification, regression and clusterization models were compared with existing tools presented in the literature that have similar features to DMAKit. To do this, Matlab Machine Learning Toolbox and the related R packages in the area were used. For evaluation, and because of its popularity among available datasets for the development and training of models, the iris dataset [30] was used.

Results are summarized in Table 1.

For classification and regression models, the Decision Trees algorithm was used, with its default parameters and $k = 10$ for cross validation, while for clustering models, the k-Means algorithm was selected. The accuracy obtained for classification models in all cases was similar, with an average of 96%. R score values presented slight differences, although minimal to establish comparisons. The same trend was observed when evaluating clustering models, except for WEKA, which does not provide a useful performance metric for comparison. The fact that the results obtained by DMAKit are similar to currently used tools corroborates the correct implementation of its methods. However, these results may be affected by user decisions regarding data pre-processing, coding of categorical variables and use of different dataset normalization techniques, differences between performance metrics are not significant for this case of study. Additionally, another distinguishing feature of our tool is to allow us to export the predictive models that are developed, allowing us to obtain predictions for new examples without the need to train the datasets again.

Table 1
Comparison of different data mining tools and programming libraries, applying the same dataset and the same algorithms in different data analysis. For classification and regression models, Decision Trees were used, and k-Means was used for clustering.

Tool or library	Classifier (Accuracy)	Clustering (Silhouette score)	Regression (R score)
DMAKit	96.7%	0.48	0.98
WEKA	96.0%	0.48	0.96
Matlab machine Learning toolbox	96.7%	0.49	0.94
R packages	96.5%	0.47	0.95

In-algorithm parametric swipe: algorithms exploration module One of the significant advantages that DMAKit has is its Machine Learning algorithms exploration module, available for classification, prediction, and clustering tasks. In this module, our tool performs a scan on the training parameters of algorithm models, obtaining different models, and their respective performance measures. Once the scan is complete, we select those models that correspond to the upper outliers of the distributions of performance measures that result from joining all the individual models. Although Matlab (Machine Learning Toolbox), KEEL, and STATISTICA, among others, present a similar module, they only explore the algorithm itself, without generating changes in the parametric combinations that give rise to the models. Another point to note is that only DMAKit presents this exploration module for unsupervised learning algorithms, mainly related to pattern identification tasks.

Persistent storage DMAKit allows the user to generate persistent storage of the input datasets, the results, and the generated models, and to manage them through the dashboard implemented in the web tool. All the stored data are of private access and, thanks to the user management system and personal work area implemented in the tool, are only available to the owner user.

An in-depth comparison of DMAKit with other tools with matching criteria (web interfaces and machine learning packages) was performed. We compared both free-to-use and licensed tools, as summarized in the supplementary file `ComparingDMAKit.xlsx`

4.3. Deployment scalability to cloud applications

The DMAKit deployment scheme was inspired in an IaaS–PaaS hybrid system. In the first, our local host provides us with virtual machines to exclusively execute DMAKit applications and tasks. Therefore, scalability would be reachable by increasing bandwidth, number of cores, amount of RAM, and storage in the server. A concurrency evaluation was carried out through stress tests, simulating a maximum number of 500 users connected simultaneously. The deployment scheme, in this context, allows a vertical scaling by default, since a better user experience would be possible by acquiring and implementing more or better hardware resources in the event of an increased demand.

The PaaS deployment scheme inspires the hosting of all modules and tasks related to the execution of actions of our platform. The architecture is mainly hosted on the NLHPC (National Laboratory for High Performance Computing) computer systems, which automatically implements scaling measures if needed to ensure the availability of the systems and execution of queued jobs. The deployment of visualization instances for results and user-platform interaction interfaces followed the IaaS architecture. Persistent storage and job results were deployed as PaaS, accessible through an API-REST service.

5. Cases studies

5.1. Use of DMAKit in biotechnological and clinical datasets

We developed DMAKit for the analysis of datasets applying data mining and pattern recognition techniques, without requiring any specific knowledge on informatics. The tool can be used on any dataset in the correct input format, independently of its origins and with different objectives. Which of the implemented modules is appropriate for each particular case depends only on the knowledge about the provided dataset and aimed results. In order to illustrate the many functionalities presented by the tool, we present two cases of study on the application of the different modules implemented in DMAKit for analyzing clinical and biotechnological datasets.

5.1.1. Study of single point mutations in proteins

The analysis of single point mutations in proteins is one of the most relevant areas of interest in the field of protein engineering. The main interest in the study of mutations is due to the possibility of correlating energy changes in the protein ($\Delta\Delta G$) and clinical relevance of the mutation; if $\Delta\Delta G$ is greater than 2 kCal/mol, the mutation is likely to imply an adverse change in the patient's health status. The above is because, structurally speaking, the protein perceives more significant restrictions for the residue, which causes interactions to be lost, which turns out to be harmful if said residue belongs to prostatic, regulatory or allosteric sites, among other possibilities [31]. It is a typical procedure to perform a feature analysis and a preliminary assessment of the correlation between the attributes as a first step for the development of prediction or classification models. To illustrate the significant advantages that DMAKit offers for the analysis of datasets and the development of classification or regression models, a dataset of single point mutations reported in the protein Human Growth Hormone bound to a single receptor, (Protein Data Bank Code 1A22) [32] was analyzed. The dataset contemplated a total of 132 examples and 13 attributes that described structural components and thermodynamic information associated with each mutation, obtained by using the SDM software [33]. We selected two attributes in this dataset as response variables: the protein stability given the mutation and the difference of free energy ($\Delta\Delta G$) between the wild-type and mutated residues. Using DMAKit, we developed classification models for stability assessment and regression models for the evaluation of $\Delta\Delta G$. Table 1 (Supplementary Materials) offers a complete description of the attributes. Mutations for 1A22 were extracted from different studies reported in the literature [34–43].

We used the DMAKit Statistical Analysis module to study the $\Delta\Delta G$ values, evaluating their distribution by drawing histograms and applying the Shapiro test. With this analysis, we determined that $\Delta\Delta G$ values have a Gaussian distribution, given the value of the Shapiro statistic (0.9).

DMAKit Feature Analysis module was used to evaluate the relationships between the descriptors and their relevance when developing classification or regression models. Dimensionality reduction techniques were also applied to determine how individual attributes contribute to the total variance through a Principal Components Analysis (PCA). PCA indicated that five components were explaining 80% of the variance, afterwards obtaining the same result via Random Forest. Structural and thermodynamic information presented a greater relevance when describing the mutation in terms of residues and positions in the sequence. Fig. 2 presents the relevance of the descriptors in the dataset.

We employed the DMAKit Supervised Learning module to train classification and regression models. The input dataset was preprocessed by coding categorical variables using One Hot Encoder and standardized using the Z-score. No issues were detected when analyzing class imbalance or the presence of outliers in the $\Delta\Delta G$ distribution. Different models were trained, both for classification (stability) and regression ($\Delta\Delta G$), applying cross-validation with $k = 10$ as an over-fit avoidance method.

At the training stage of classification models, DMAKit reports the confusion matrix, the performance measures, the learning curve, and the evaluation of the predictor's specificity and sensitivity for the different classes. The best model developed for classification of protein stability before the substitution of residues resulted from Bernoulli Naïve Bayes, a probability-driven algorithm. The metrics obtained contemplated a Precision value of 87.26%, Accuracy of 81.9%, Recall of 0.83 and an F1 score of 0.84, showing that it is a highly efficient model, in particular regarding precision, as it has a low probability of making type I errors (false positives). Fig. 3 presents the confusion matrix reported by DMAKit for the classification model and the specificity and sensitivity of the model for each class.

Regarding regression models for estimating free energy differences between mutant and wild-type protein, the most suitable models resulted from Decision Trees (DTs). The best model presented an R score of 0.916 and a Pearson coefficient of 0.980, implying that the model will be efficient when handling new cases. Fig. 4 is a scatter plot of predicted versus real values for changes in free energy caused by the substitution of amino acids. Significant correlations are observed, in line with the high R score and Pearson's coefficient.

The use of DMAKit allowed the analysis of datasets related to the stability of a protein and to the differences in free energy caused by substitutions of residues, identifying the attributes with greater relevance, and influencing the general variance by applying the feature analysis module. It also allowed the development and validation of classification and regression models for the evaluation of mutations in the 1A22 protein through supervised learning modules.

5.1.2. Clustering of linear peptide sequences

As mentioned in the previous sections, DMAKit also has a module for the unsupervised learning-driven pattern recognition and clustering models training. To test its functionality, we assessed its ability to find patterns in linear sequences of peptides, a complex challenge in protein engineering. We worked with 111 linear amino acid sequences, aiming to obtain a subset of elements that would represent the entire peptide dataset accurately.

We coded each linear sequence into a vector, where the i'th component represented the frequency of appearance of the $i-'$th amino acid, and the 21 entry had the length of the sequence. DMAKit tested different unsupervised learning algorithms to generate clusters and assessed the performance of the models obtained using their silhouette coefficient and Calinski–Harabasz index. As the previous step resulted in many models, we obtained distributions of the performance of each metric, reporting as the selected models those which stood as upper outliers in such distributions. For this case of study, the best models resulted from agglomerative or hierarchical algorithms, using Euclidean distances. Finally, we obtained 49 representative clusters with a silhouette coefficient of 0.5 and a Calinski–Harabasz index of 4.8. The distribution of elements in each group of the cluster was not uniform, given that there are clusters that concentrate elements.

The results obtained by applying the clustering module and the model exploration tool highlight the usability of DMAKit to develop clustering models, based on unsupervised learning algorithms, and the advantages of using exploratory tools when evaluating algorithms and combinations of parameters.

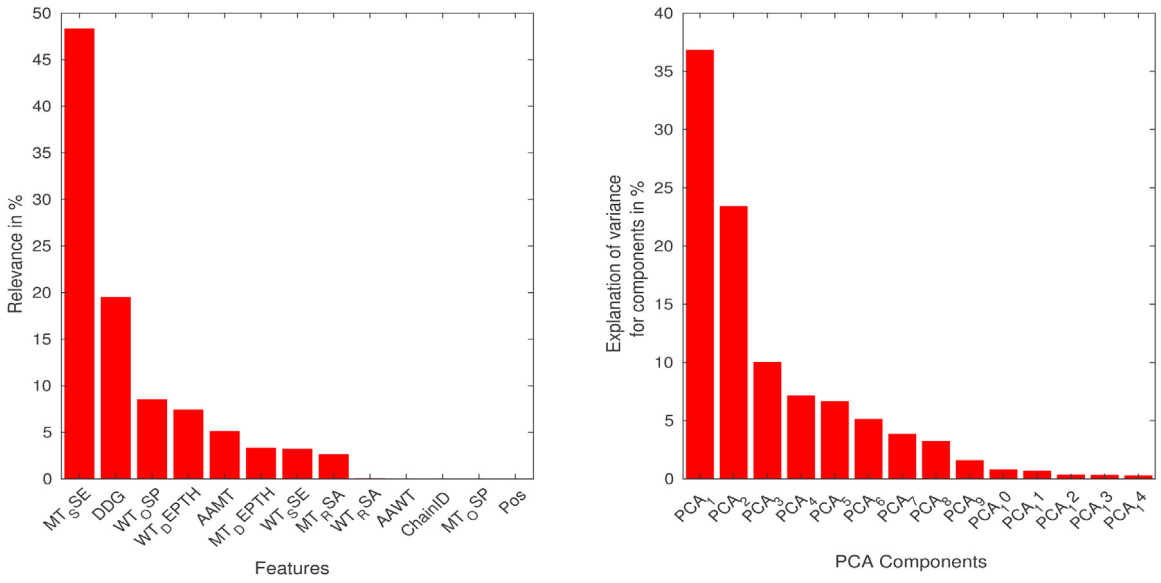

Fig. 2. Feature relevance according to Random Forest-based models (left), and a principal components analysis PCA (right). Attributes MT_SSE, $\Delta\Delta G$ and WT_OSP together have a joint relevance close to 75%, which is confirmed by PCA, where the three first components represent 70% of the total variance.

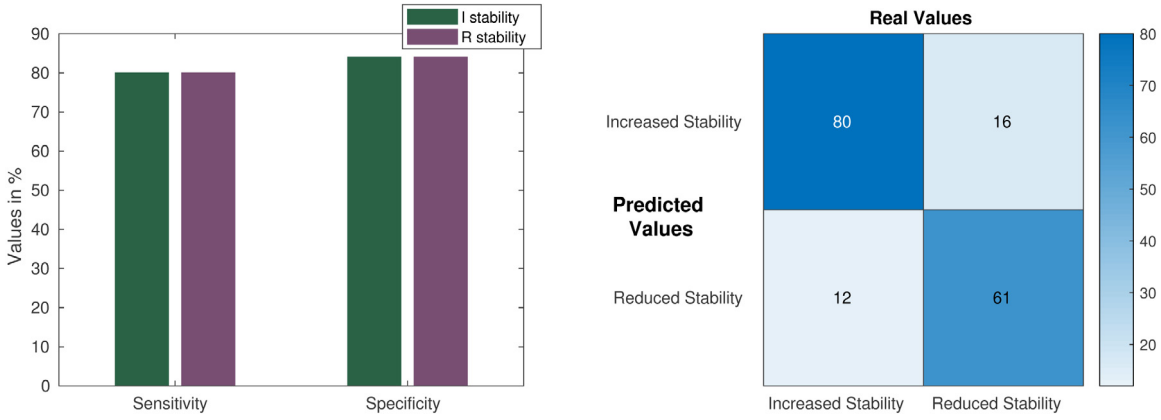

Fig. 3. Specificity and sensitivity graph of mode (left), and confusion matrix (right) reported for the protein stability classification model given residual substitutions, trained with Bernoulli Naïve Bayes algorithm and cross-validation with $k = 10$. There is a slight difference when classifying examples that increase stability (I stability) and those that reduce stability (R stability), but not enough to suspect bias either in the data or the prediction process.

Table 2
User evaluation of DMAKit in a 1–5 scoring scale.

Concept	Average score	Variability (STD)
Ease of task execution	4,6	0,49
Task execution time	4,9	0,36
Clarity of results	4,1	0,67
How well did the results meet your expectations?	4,8	0,43
How likely are you to recommend DMAKit?	4,8	0,40

5.2. User evaluation feedback

To assess user experience with DMAKit, we recorded the testimonials of a group of researchers from different backgrounds, who did not have relevant experience implementing Machine Learning models or knew DMAKit previously. The background of each one covers the areas of Molecular Biology, Biotechnology, Bioinformatics, Neuroscience, Chemistry, Physical Chemistry, Protein Engineering, Education, Computer Science, and Mathematics.

The users remarked the high usability and ease of use of our tool, and the overall evaluation of DMAKit was positive (results presented in Table 2). More advanced users compared DMAKit models with results obtained using other methods and tools, checked the robustness of DMAKit results on the same dataset and predicted experimental results contained in a test subset of their own. In all these tests, they were able to confirm the correctness of DMAKit models and checked that DMAKit models and classification are comparable to those obtained using state-of-the-art methods and software. Besides all the positive feedback and validation, several opportunities for improvement were pointed out, which we will include in future versions of our tool. Some of them relate to data visualization, specific analyses, and background information, in order to propose DMAKit not only as a tool for obtaining models but also for learning about Machine Learning, Statistics, and scientific programming. All testimonials are available in Section 4 of the Supplementary Materials.

6. Conclusions

We designed and implemented DMAKit, a user-friendly web platform based on Python programming language for non-

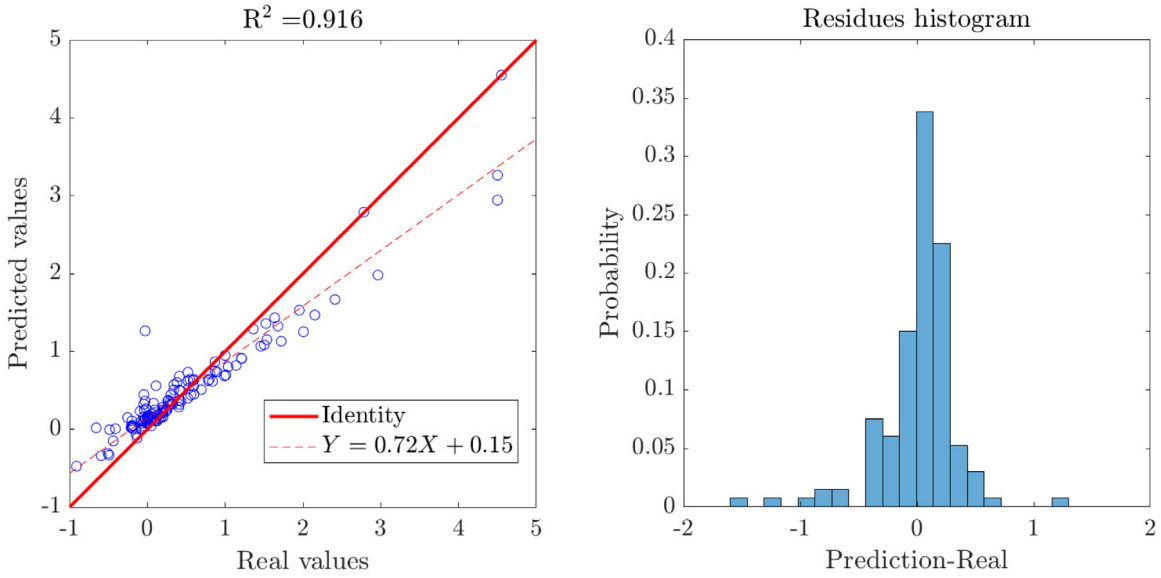

Fig. 4. Scatter plot of model prediction vs. real $\Delta\Delta G$ values (left), and scatter plot of error model (right). A clear linear behavior is observed with an R score of 0.916, highlighting the high quality and precision of the model when estimating new $\Delta\Delta G$ values caused by mutational changes.

commercial use, which allows non-specialist users to apply data mining and pattern recognition techniques to different kinds of datasets. Remarkably, DMAKit was thought and programmed to serve researchers from the biology, biotechnology and biomedical communities allowing them to generate results based on different types of analyses. DMAKit includes modules for statistical analysis, feature relevance evaluation, dimensionality reduction techniques, pattern search through unsupervised learning algorithms, and training of classification and prediction models, as well as an exploratory model module. We have demonstrated the applicability and power of this tool when analyzing different datasets with a wide array of state-of-the-art ML and statistical approaches, with an essential advantage over presently existing alternatives. Features such as exporting and using predictive models, algorithm exploration tools and job combinations, as well as extra elements, such as not requiring any computer resource other than just access to an internet connection, and the creation of dataset repositories, models and results of data analysis, also demonstrate large advantages of DMAKit with respect to those currently available. For these reasons, we expect DMAKit to become a significant contribution for those scientists who have large volumes of data to be analyzed but do not have the informatics, mathematical, statistical knowledge, or time, to implement models. Future work on DMAKit comprises the development of modules for the analysis of signals and frequencies, modeling through graph techniques and the benefits that these entail, and analysis of images for the extraction and recognition of recurrent patterns. Artificial neural networks will undoubtedly be a valuable tool to be incorporated into DMAKit in a future version

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

DMAKit web platform is publicly available at https://pesb2.cl/dmakitWeb. The source code for DMAKit-lib and sample datasets used for the different cases of study presented in this manuscript are available in the GitHub repository https://github.com/dMedinaO/dmakitWeb. Demonstrative and tutorial videos for the web platform are available in https://pesb2.cl/dmakittutorials/, or directly in https://www.youtube.com/playlist?list=PL8E6de1zuTIi_fxYi9a38hFeC72UgpCHq.

Acknowledgments

This research has been financed mainly by the Centre for Biotechnology and Bioengineering - CeBiB (PIA project FB0001, Conicyt, Chile). Powered@NLHPC: This research was partially supported by the supercomputing infrastructure of the National Laboratory for High-Performance Computing, NLHPC (ECM-02), Chile. DM-O gratefully acknowledges Conicyt, Chile, for PhD fellowship 21181435. SC gratefully acknowledges support from the Chilean National Agency for Research and Development through ANID PIA Grant AFB180004.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.is.2020.101557.

References

[1] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, The kdd process for extracting useful knowledge from volumes of data, Commun. ACM 39 (1996) 27–35.
[2] O. Maimon, L. Rokach, Data Mining and Knowledge Discovery Handbook, 2005.
[3] A. McAfee, E. Brynjolfsson, T.H. Davenport, D. Patil, D. Barton, Big data: the management revolution, Harv. Bus. Rev. 90 (2012) 60–68.
[4] F.F. Costa, Big data in biomedicine, Drug Discov. Today 19 (2014) 433–440.
[5] C.S. Greene, J. Tan, M. Ung, J.H. Moore, C. Cheng, Big data bioinformatics, J. Cell. Physiol. 229 (2014) 1896–1900.
[6] C.H. Lee, H.J. Yoon, Medical big data: promise and challenges, Kidney Res. Clin. Pract. 36 (3) (2017).
[7] D.M. Camacho, K.M. Collins, R.K. Powers, J.C. Costello, J.J. Collins, Next-generation machine learning for biological networks, Cell 173 (2018) 1581–1592.
[8] K.Y. Michael, J. Ma, J. Fisher, J.F. Kreisberg, B.J. Raphael, T. Ideker, Visible machine learning for biomedicine, Cell 173 (2018) 1562–1565.
[9] C. Auffray, R. Balling, I. Barroso, L. Bencze, M. Benson, J. Bergeron, E. Bernal-Delgado, N. Blomberg, C. Bock, A. Conesa, et al., Making sense of big data in health research: towards an eu action plan, Genome Med. 8 (71) (2016).
[10] I. Kavakiotis, O. Tsave, A. Salifoglou, N. Maglaveras, I. Vlahavas, I. Chouvarda, Machine learning and data mining methods in diabetes research, Comput. Struct. Biotechnol. J. 15 (2017) 104–116.

[11] K. Kourou, T.P. Exarchos, K.P. Exarchos, M.V. Karamouzis, D.I. Fotiadis, Machine learning applications in cancer prognosis and prediction, Comput. Struct. Biotechnol. J. 13 (2015) 8–17.
[12] I.V. Hinkson, T.M. Davidsen, J.D. Klemm, I. Chandramouliswaran, A.R. Kerlavage, W.A. Kibbe, A comprehensive infrastructure for big data in cancer research: accelerating cancer research and precision medicine, Front. Cell Dev. Biol. 5 (83) (2017).
[13] A.E. Johnson, M.M. Ghassemi, S. Nemati, K.E. Niehaus, D.A. Clifton, G.D. Clifford, Machine learning and decision support in critical care, Proc. IEEE Inst. Electr. Electron. Eng. 104 (2016) 444.
[14] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The weka data mining software: an update, ACM SIGKDD Explor. Newsl. 11 (2009) 10–18.
[15] J. Alcalá-Fdez, L. Sánchez, S. Garcia, M.J. del Jesus, S. Ventura, J.M. Garrell, J. Otero, C. Romero, J. Bacardit, V.M. Rivas, et al., Keel: a software tool to assess evolutionary algorithms for data mining problems, Soft Comput. 13 (2009) 307–318.
[16] J.M. Hilbe, Statistica 7: an overview, Amer. Statist. 61 (2007) 91–94.
[17] R. Collobert, K. Kavukcuoglu, C. Farabet, et al., Torch7: A matlab-like environment for machine learning, in: BigLearn, NIPS workshop, Granada, 2011, p. 10.
[18] A. Vedaldi, K. Lenc, Matconvnet: Convolutional neural networks for matlab, in: Proceedings of the 23rd ACM International Conference on Multimedia, ACM, 2015, pp. 689–692.
[19] M.J. De Hoon, S. Imoto, J. Nolan, S. Miyano, Open source clustering software, Bioinformatics 20 (2004) 1453–1454.
[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[21] W. McKinney, Data structures for statistical computing in python, in: van der Walt, J. Millman (Eds.), Proceedings of the 9th Python in Science Conference, 2010, pp. 51–56.
[22] M. Kuhn, Caret: classification and regression training, in: Astrophysics Source Code Library, 2015.
[23] B. Ripley, W. Venables, M.B. Ripley, Package 'nnet', in: R Package Version, Vol. 7, 2016, pp. 3–12.
[24] P. Wegner, Concepts and paradigms of object-oriented programming, ACM SIGPLAN Oops Messenger 1 (1990) 7–87.
[25] J.D. Hunter, Matplotlib: A 2d graphics environment, Comput. Sci. Eng. 9 (2007) 90–95.
[26] A.B. Yoo, M.A. Jette, M. Grondona, Slurm: Simple linux utility for resource management, in: Workshop on Job Scheduling Strategies for Parallel Processing, Springer, 2003, pp. 44–60.
[27] U. Maulik, S. Bandyopadhyay, Performance evaluation of some clustering algorithms and validity indices, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 1650–1654.
[28] P.J. Rousseeuw, Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, J. Comput. Appl. Math. 20 (1987) 53–65.
[29] D. Dua, C. Graff, UCI machine learning repository, 2017.
[30] R.A. Fisher, The use of multiple measurements in taxonomic problems, Ann. Eugen. 7 (1936) 179–188.
[31] C.L. Worth, R. Preissner, T.L. Blundell, Sdm—a server for predicting effects of mutations on protein stability and malfunction, Nucleic Acids Res. 39 (2011b) W215–W222.
[32] T. Clackson, M.H. Ultsch, J.A. Wells, A.M. de Vos, Structural and functional analysis of the 1: 1 growth hormone: receptor complex reveals the molecular basis for receptor affinity, J. Mol. Biol. 277 (1998) 1111–1128.
[33] C.L. Worth, R. Preissner, T.L. Blundell, SDM-a server for predicting effects of mutations on protein stability and malfunction, Nucleic Acids Res. 39 (2011a) W215–W222.
[34] E. Capriotti, P. Fariselli, I. Rossi, R. Casadio, A three-state prediction of single point mutations on protein stability changes, BMC Bioinformatics 9 (S6) (2008).
[35] A.J. Bordner, R.A. Abagyan, Large-scale prediction of protein geometry and stability changes for arbitrary single point mutations, Proteins: Struct. Funct. Bioinform. 57 (2004) 400–413.
[36] G. Wainreb, L. Wolf, H. Ashkenazy, Y. Dehouck, N. Ben-Tal, Protein stability: a single recorded mutation aids in predicting the effects of other mutations in the same amino acid site, Bioinformatics 27 (2011) 3286–3292.
[37] Y. Peng, L. Sun, Z. Jia, L. Li, E. Alexov, Predicting protein–DNA binding free energy change upon missense mutations using modified MM/PBSA approach: SAMPDI webserver, Bioinformatics 34 (2017) 779–786.
[38] I. Getov, M. Petukh, E. Alexov, Saafec: Predicting the effect of single point mutations on protein folding free energy using a knowledge-modified mm/pbsa approach, Int. J. Mol. Sci. (17) (2016).
[39] M. Petukh, L. Dai, E. Alexov, Saambe: Webserver to predict the charge of binding free energy caused by amino acids mutations, Int. J. Mol. Sci. (17) (2016).
[40] Z. Zhang, L. Wang, Y. Gao, J. Zhang, M. Zhenirovskyy, E. Alexov, Predicting folding free energy changes upon single point mutations, Bioinformatics 28 (2012) 664–671.
[41] F. Ancien, F. Pucci, M. Godfroid, M. Rooman, Prediction and interpretation of deleterious coding variants in terms of protein structural stability, Sci. Rep. 8 (4480) (2018).
[42] A. Broom, Z. Jacobi, K. Trainor, E.M. Meiering, Computational tools help improve protein stability but with a solubility tradeoff, J. Biol. Chem. 292 (2017) 14349–14361.
[43] L. Quan, Q. Lv, Y. Zhang, STRUM: structure-based prediction of protein stability changes upon single-point mutation, Bioinformatics 32 (2016) 2936–2946.