Automated Feature Selection and Classification for High-Dimensional Biomedical Data

Tammo P.A. Beishuizen
Eindhoven University of Technology
Joaquin Vanschoren
Eindhoven University of Technology
Peter A.J. Hilbers
Eindhoven University of Technology
Dragan Bošnački (d.bosnacki@tue.nl)
Eindhoven University of Technology

Research Article

Keywords: automated machine learning, feature selection, biomedical data

Posted Date: October 12th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-563410/v1

RESEARCH

Automated Feature Selection and Classification for High-Dimensional Biomedical Data

Tammo P.A. Beishuizen*, Joaquin Vanschoren, Peter A.J. Hilbers and Dragan Bošnački

*Correspondence:
tim.beishuizen@gmail.com
Eindhoven University of Technology, Eindhoven, Netherlands
Full list of author information is available at the end of the article

Abstract

Background Automated machine learning aims to automate the building of accurate predictive models, including the creation of complex data preprocessing pipelines. Although successful in many fields, they struggle to produce good results on biomedical datasets, especially given the high dimensionality of the data.

Result In this paper, we explore the automation of feature selection in these scenarios. We analyze which feature selection techniques are ideally included in an automated system determine how to efficiently find the ones that best fit a given dataset, integrate this into an existing AutoML tool (TPOT), and evaluate it on four very different yet representative types of biomedical data: microarray, mass spectrometry, clinical and survey datasets. We focus on feature selection rather than latent feature generation since we often want to explain the model predictions in terms of the intrinsic features of the data.

Conclusion Our experiments show that for none of these datasets we need more than 200 features to accurately explain the output. Additional features did not increase the quality significantly. We also find that the automated machine learning results are significantly improved after adding additional feature selection methods and prior knowledge on how to select and tune them.

Keywords: automated machine learning; feature selection; biomedical data

1 Introduction

Although biomedical datasets often contain a large number of features, only a subset of these features are usually relevant for the modeling and analysis in a particular application. Models that explain the output based on only the most relevant features are desirable because of computational efficiency as well as the model interpretability. Therefore, there lies a clear benefit in reducing the number of features by removing the irrelevant ones. There exists extensive work in feature selection [1, 2] which has resulted in multiple selection methods [3, 4, 5, 6] and evaluations of their performance [7, 8].

In this work, we aim to automate the task of feature selection, especially for high-dimensional datasets, which are prevalent in biomedical studies but less well covered by currently used automated machine learning (AutoML) methods.

We evaluate a wide range of feature selection methods to make an informed decision on which methods should be incorporated in automated machine learning

frameworks. In particular, we consider three classes of methods: filter methods, wrapper methods, and embedded methods, and integrate them as additions to *TPOT*, an automated machine learning tool, to explore the benefits of these new components. As part of this integration, we introduce a quality metric which, apart from the accuracy of the resulting models, also takes into account the number of preserved features to balance accuracy vs interpretability of the resulting models. Different combinations of feature selection methods and meta-parameters are tested according to this metric. To evaluate the effectiveness of the resulting framework four datasets were used.

In the remainder of this paper, we first cover related work and the contributions of this paper in Section 2. Section 3 provides the necessary background. Section 4 describes our method for adapting TPOT to work better on high-dimensional data. Section 5 covers our experimental setup, including details of the datasets under study. Section 6 interprets the results of our experiments and Section 7 concludes the work with a discussion.

2 Related work

A good overview of basic feature selection techniques can be found in [9]. Numerous algorithms have been proposed for feature selection [3, 4, 5, 6, 10, 11] and various performance evaluation tests have been developed [1, 7, 8, 12, 13]. Most of these focus on datasets with much smaller numbers of features (no more than 1,000) than is commonly encountered in biomedical datasets. Other feature selection research focuses on specific types of feature selection, e.g. wrapper methods [14, 15, 16] or embedded methods [17]. A select number of articles focuses on bigger feature set sizes. Chen et al. [18] considered larger general datasets and used support vector machines for feature selection. Xing et al. [19] focused on feature selection for a microarray dataset of 7000 instances and showed that feature selection is beneficial. Other work looked at 35 datasets with a range of features between 37 and 50,000, trying to identify the best performing feature selection types. [20]. They used five advanced, less generally applicable algorithms, whereas the ones applied in our work are more general and intuitive. Some other examples of automated machine learning approaches that search for pipelines and perform dimensionality reduction are [21, 22, 23].

Georges et al.[24] compared feature selection techniques with regard to the similarity in feature subset outcome. They also considered feature selection based on dataset reproducibility. The tested techniques are rather specific and as such less applicable for an extension for an automated machine learning tool.

Pes et al. [25] focused on feature selection for imbalanced classes. They conclude that feature selection contributed quite significantly to the efficiency of the algorithms, but no algorithm clearly outperformed the other ones. This work also highlights the need for feature selection algorithms in automated machine learning, specifically for imbalanced datasets.

Panicker et al. [26] investigated the quality of feature selection algorithms and their influence on the classification quality. The outcome was that feature selection is beneficial for classification, however there is no clear best feature selection algorithm.

The main characteristics of our work on feature selection for biomedical data are the following:

- To balance predictive accuracy and model interpretability, we aim to find feature subsets that are as small as possible while keeping the performance as high as possible. This was seldomly a goal in previous works [27]. To this end, we developed a new metric, FS_score, that incorporates both the feature subset size and predictive performance. The utility of this score is evaluated on different case studies.
- We focus on high-dimensional biomedical datasets with more than 1000 features, where feature selection has an especially large impact. This is not often done, whereas these types of datasets are becoming increasingly prevalent.
- We leverage very general feature selection algorithms, and as a result we expect that non-expert users can more easily understand and trust the techniques. Moreover, the considered methods are also quite diverse in their properties, and our analysis shows the benefits and drawbacks of different types of methods.
- Automation of feature selection techniques is added to the framework. Automated machine learning is a very promising approach to extracting information out of a dataset. Properly implementing feature selection in automated machine learning opens it up for high dimensional datasets.

3 Background

Here we give a brief overview of the considered feature selection methods. Also we discuss the concept of automated machine learning as a possible technique to be added to the framework, in combination with a tool that implements automated machine learning.

3.1 Feature Selection Methods

Feature selection is a way to perform dimensionality reduction. In feature selection a subset of features is chosen to represent the complete sample space [28]. Several techniques are available to choose a representative subset and the effectiveness of these techniques has been tested in numerous works [12, 29]. These techniques can be grouped accordingly in three different categories: *filter methods*, *wrapper methods* and *embedded methods* [9]. A thorough explanation of these three categories can be found in Saeys et al.[9].

3.2 Automated Machine Learning: Tree-based Pipeline Optimization Tool

Tree-based Pipeline Optimization Tool (TPOT) is a *Python* based tool that implements automated machine learning (autoML). Using genetic programming and tree structures pipelines are formed and tried out to find the best solution for a particular dataset. The pipeline backbones consist of preprocessing and machine learning algorithms from *scikit-learn*, but also several additional algorithms are present (e.g. a hot encoding algorithm).

Considering the capabilities from *TPOT* to cope with challenges in biomedical data, several methods are available. It has several different normalisation/standardisation scalers (StandardScaler, RobustScaler, MinMaxScaler) to tackle feature heterogeneity between different datasets and errors. It also has some feature selection

operators to deal with errors (VarianceThreshold, SelectKBest, SelectPercentile). Wrapper methods however are not included, which may perform better in some situations. In place of missing values it imputes the median before starting the evolutionary algorithm. Non-numeric data needs to be converted manually.

A wide variety of algorithms are present in TPOT, as well as numerous hyper parameter values for those algorithms. This is both an advantage as well as a disadvantage since it is very likely that many - if not most - combinations of algorithms and hyper parameters are not a useful outcome. TPOT randomly chooses mutations, but possibly better algorithms could be selected based on some guiding heuristics. For example meta-features could be used to improve the algorithm selection.

4 Methods

4.1 Feature Selection Quality

To estimate the quality of the feature selection, multiple machine learning algorithms are explored [30]. Classification of the datasets is done with several machine learning algorithms and validation and tests scores show how well the data can be classified after feature selection. The quality will be described with the accuracy of the machine learning algorithms: the number of correct classifications divided by the total number of classifications. In the evaluation five different machine learning classifiers are used from *scikit-learn*: logistic regression, decision trees, nearest neighbour, support vector machines and Naive Bayes. Better feature selection algorithms have relatively high accuracy, as they are better at preserving the right features. To capture possible overfitting, both a validation and a test score is computed for the accuracy. For every experiment a training set and a test set is created, making the test set 25% of the complete dataset. A validation score is computed by using the "leave one out" technique on the training set and a test score is computed by testing the classification score of the test set. Since the samples are not evenly distributed over the classes also the precision, recall and F1 scores are also computed to find potential bias in the result.

In some cases it could be convenient that the quality metrics of the machine learning algorithm takes into account the feature subset size. The standard metrics, like accuracy, in those cases are not sufficient enough to capture this aspect. Therefore, we introduce a new modified quality metrics *FS_score* (Equation 1). In *FS_score* a modified version of the original score, in this work being accuracy, is adjusted by multiplying it by a factor dependent on the number of features. This factor consists of a constant β, a value in range $[0, 1]$ which can be chosen to express the influence extent of the number of features. In practice $\beta < 0.025$, so the reduction does not have too much influence on the score.

$$\text{FS\_score} = \text{score} \times 10^{-\beta \cdot \#\text{features}} \tag{1}$$

The factor #features is the absolute future number rather than a relative one (for example a percentage of features). The reason for using an absolute count can be found in the goal of the data analysis. In data analysis the number of relevant features that can be quite limited. Useful results need relations between the input

Figure 1: An example of the impact of the correction factor on the score, in this case accuracy. The shown correction factor uses $\beta = 0.005$. On the x-axis the number of features is shown and on the y-axis the value for the original accuracy, the correction factor and the *FS_score*.

and the output, and these relations need to be as simple as possible, which among others, is reflected in the smaller number of features. If the number of features is relative to the total number of features, the input size can change significantly. Take for example the Micro-organisms and the RSCTC datasets provided for this study. A 2% of the total features would be 26 and 1,100 features for these datasets respectively. Intuitively, the relations between 26 features and the output should be easier to grasp than the relations involving 1100 features.

To put the impact of the *FS_score* in perspective, an example figure is made how the outcome is computed (Figure 1). This figure shows the impact on the performance of a method when using a certain number of features. After using a correction factor, an optimum is created for which the number of features is important, the old optimal score did not include the number of features. The example (with $\beta = 0.005$) seems to give a good indication of the desired outcome. For every 10 features, the *FS_score* is reduced by about 10%, which meant for the example filter method that for 50 features an optimum was reached (Figure 1). Because of this empirically found desired trade-off, $\beta = 0.005$ will be chosen in this project when using *FS_score*.

4.2 TPOT feature selection integration

As already mentioned, TPOT is an effective tool to find the best machine learning pipeline for a certain dataset. Two restrictions hinder optimization regarding feature selection:

1 *Lack of warm start*
TPOT has a vast array of machine learning and preprocessing algorithms to find the best possible pipeline. Due to the number of possibilities being very high, a lot of time may be wasted due to searching in wrong directions. For feature selection, a pre-defined selection of pipelines (also known as a warm start) would improve efficiency.

2 *Feature selection possibilities*
Several filter and embedded methods are present (Subsection 3.1). All of these select percentages of feature selection, though. Still a large number of features can be present in the result after using percentages. On top of that, no wrapper methods are available, either.

To resolve these optimization restrictions, two corresponding additions are made to *TPOT*.

Focused feature selection An option is added to always start the original population with a feature selection algorithm in the feature selection pipeline. Due to this start the expected search for a good feature selection method is bypassed immediately, which should result in more optimized final pipeline.

Table 1: The experiment details for testing the non-trivial changes in *TPOT*. This experiment is rerun 5 times.

Experiment factors	Detailed values	Remarks
Datasets	Micro-organisms Arcene RSCTC Psoriasis	large number of features (Subsection 5.1)
Performance measurement	One type: *FS_score* - $\beta = 0.005$	Addition of correction factor (Equation 1)
TPOT input parameters	One set of input values: - max. optimization time = 120 min - max. alg. evaluation time = 10 min - pop. size = 12 - train size = 0.9	Explanation of input values: - The time one run should take (2 hours) - The evaluation time of one pipeline - The number of pipelines in one generation - The number of samples used for training
Pipeline selection	Regular selection Feature selection focused	Possible obligatory addition of a feature selection algorithm
Change in feature selection algorithm set	Regular feature selection algorithm set New feature selection algorithm set	A change between several basic feature selection algorithms to feature selection algorithms designed for at most 200 features preservation

Alternative feature selection algorithm set An option is added to use an alternative feature selection algorithms set. This set consists of filter, wrapper and embedded methods and the hyper-parameters are predefined to use an upper bound of 200 features. This upper bound was chosen based on the aforementioned FS_score, as the factor in this score already has a big impact: a factor 0.2 on the final score.

4.3 Implementation

All methods and additions to TPOT are implemented in *Python* version 3.6 with *Anaconda* version 3 environment. The code is available at `https://github.com/TimBeishuizen/cBioF`.

5 Experimental Setup

5.1 Datasets

Four datasets were used as a case study for the feature selection algorithms. Two sets are microarray datasets that are used for research on psoriasis [31, 32, 33, 34] and cancer [35]. The two other sets are mass spectrometry datasets, used for research on cancer [36] and micro organisms [37]. All four datasets have a large number of features varying from 1,300 to 54,675 features with a number of samples varying from 200 to 580 (Table 2). All of these datasets are related to classification since tests are done for different test subject groups. Therefore the emphasis on feature selection algorithms for classification. Considering the large number of features, it can be expected that many of them would be irrelevant or redundant and as such lend themselves to feature selection.

5.1.1 Micro-array Datasets

Micro-array data contains information on expression levels of a large number of genes [38]. These expression levels are usually used for a genomics level research, such as genome mapping, transcription factor activity and pathogen identifications.

Micro-array data are known to present challenges in data quality and need normalisation [39]. The thousands of features that are present in the data also indicate

Table 2: A schematic overview of the four datasets.

Dataset focus	Data type	Features	Samples	Classes	Remarks
Psoriasis	Micro-array	54675	580	3	- Derived from five different datasets [31, 32, 33, 34] –
Cancer	Micro-array	54675	383	9	- Used in a data mining challenge [35]
Cancer	Mass Spectrometry	10000	200	2	- Created for the NIPS conference [36] - Several probe features are present –
Micro-organisms	Mass Spectrometry	1300	571	20	- Originates from a micro organisms study [37]

that selection is very important, so that the irrelevant genes can be ruled out. Aside from that, size may also be an issue. Due to the size of the data not all analyses will perform optimally in both time and quality. We use two microarray datasets:

- *Psoriasis microarray dataset*
  This dataset is comprised of five different datasets consisting of 54,675 features, all corresponding to gene expression [31, 32, 33, 34]. Samples were collected from three different test subject groups: affected skin from test subjects suffering from psoriasis (214 samples), unaffected skin from test subjects suffering from psoriasis (209 samples) and skin from healthy test subjects (85 samples). Combining these three sample types gives 508 samples. Since the data comes from five different experiments, the data is normalized for every experiment.
- *Arcene: Cancer microarray dataset*
  This dataset, called Arcene dataset, is used in a challenge focusing on classification problems with a low number of samples, but a large number of features [35]. It has the same number of features as the Psoriasis dataset, 54675, corresponding to gene expression. It also has 383 samples corresponding to nine different test subject groups. The challenge did not provide labels for the test subject groups. Also these groups differ in size, one group corresponding to 150 samples and the others varying from 16 to 47 samples.

5.1.2 Mass Spectrometry Datasets

Mass spectrometry data contains information on proteins and peptides [40, 41]. Analysis of this information is used in studies about proteins, e.g., proteomics [42]. Mass spectrometry is a technique that can be used to find how much of a certain protein is present [43].

Challenges in the analysis of mass spectrometry datasets are mainly found in the way this data is produced. The expression is given for several proteins and this expression can differ significantly between proteins. Therefore some type of normalisation is useful. Aside from that, in this technique usually a large number of proteins is tested at the same time. Therefore feature selection can be helpful to remove irrelevant features. We use two mass spectrometry datasets:

- *RSCTC: Cancer mass spectrometry dataset*
  This dataset was created in the context of a classification problem to distinguish cancer patterns from normal patterns [36]. It is created for the 'Neural

Table 3: The four meta-parameters with their possible values in the first experiment.

Variable	Description	Values
Dataset	The datasets used (subsection 5.1)	Psoriasis RSCTC Arcene Micro-Organisms
Ranking method	The method used for ranking the features [9]	– T-test (*SciPy*) Mutual Information (*scikit-learn*)
Feature preservation values	The fixed size of the feature subset after feature selection	– 1, 5, 10, 25, 50, 75, 100, 150, 250, 500, 1,000
classification method	The machine learning algorithms used for selection (F1-score computed) (subsection 4.1)	– Naive Bayes Logistic Regression Support Vector Machine Decision Tree Nearest Neighbours

Information Processing Systems' conference by merging three mass spectrometry datasets. It consists of 10,000 features corresponding to either spectra of the mass spectrometry or probe variables without any predictive power. Samples from two groups are taken from patients with ovarian or prostate cancer and from control patients. No labels are given to the groups, however, it is known that one of the groups has 88 samples and the other 112 samples, combined in a total of 200 samples.

- *Micro organisms mass spectrometry dataset*
  This dataset is created to back up a proposed method for routinely performing direct mass spectrometry based bacterial species identification [37]. It consists of 1,300 features corresponding to different spectra of the mass spectrometry data and 20 test subject groups corresponding to Gram positive and negative bacterial species. Gram classification is a result of a Gram stain test [44]. The groups differ in size varying from 11 to 60 samples, making a total of 571 samples.

5.2 Evaluation of the Basic Filter Method in Combination with Classification Methods

In the first set of evaluation experiments we fix the feature selection algorithm template and evaluate it in combination with different classification methods. The basic filter method algorithm selecting the top n features [9] is used as a feature selection method. The changing meta variables are the dataset, the ranking method, the feature preservation values and the classification methods (Table 3). The range of feature preservation values chosen to reflect both the ability to show impact of separate features (more impact from fewer features) and the relevance of keeping that number of features (irrelevant feature selection when having more than 1,000 features). All of this together results in a total of $4(\text{datasets}) \times 2(\text{ranking methods}) \times 11(\text{top } n \text{ features}) \times 5(\text{accuracy computation methods}) = 440$ experiments. These experiments are visualized in eight plots, one plot for every combination of dataset and classification method. These plots then show the change in quality for different number of preserved features.

5.3 Feature Selection Algorithms Evaluation

The second set of experiments fixes the classification method in order to compare the feature selection methods. The logistic regression is chosen as the classification

Table 4: The methods that are evaluated in the second experiment setup.

Type	Method	Parameters
Filter methods	Basic Filter Methods [9]	- Rank: T-test, Mutual Information - Thresholds: 50, 100, 150 features
Wrapper methods	Forward selection [9]	- Order: Random, Mutual Information - Evaluation: Naive Bayes - Alpha: 0.01, 0.001 –
	PTA	- Order: Random, Mutual Information - Evaluation: Naive Bayes - [l, r] = [20, 10], [5, 2] - Alpha: 0.01, 0.001 –
	Floating search	- Order: Random, Mutual Information - Evaluation: Naive Bayes - Alpha: 0.01, 0.001
Embedded methods	Forward selection [9]	- Machine Learning: SVM, RandomForests - Threshold: 50, 100, 150 features

method and used with the four datasets. This is done because the logistic regression gave the most consistent result of the five machine learning algorithms, showing the smallest variation across the datasets. The average performance for all datasets is also computed for clarification purposes. An overview of feature selection methods is made (Table 4). Spectra are made with the results of the experiment that show the performance of all different combinations, showing the accuracy, precision, recall and F1-score. On top of that the computation time is computed and shown per algorithm, as well, in a separate bar chart. At last a combination of the earlier proposed *FS_score* and the computation time is shown as well for $\beta = 0.005$ to show the relation between computation time on one hand and the *FS_score* on the other hand. Other feature selection methods, such as the backwards elimination sequential method, the simulated annealing stochastic search method and the embedded backwards elimination method [9] are not evaluated. These feature selection methods are omitted because of the poor scalability with regards to computation time and therefore unfit for datasets with this many features.

5.4 TPOT Integration Evaluation

The final experiment evaluates whether our additions to TPOT improve its utility on biomedical problems. This experiment consists of multiple runs of *TPOT* and all of these steps are also shown in an explanatory table (Table 1). All four datasets are tested (Subsection 5.1) and the accuracy is changed to the feature sensitive *FS_score* with $\beta = 0.005$ (Equation 1), as previously discussed (Subsection 4.1). For testing *TPOT* an optimization time of 120 minutes (two hours) was chosen as a reasonable time constraint to run each experiment 5 times, an algorithm was not allowed to run for longer than ten minutes, a population size of 12 was chosen to not be too selective at the start, and a training set size within TPOT of 0.90 which is a general training set size when not many samples are present. Furthermore tests were done for pipeline selection both with and without focused feature selection and for both with and without alternative feature selection algorithms set, as these additions must be tested for their quality. This gives a total of 4(datasets) $\times$ 2(pipeline selection) $\times$ 2(feature selection set) $\times$ 5(experiment reruns) = 16 different experiments.

Figure 2: The average validation F1-scores shown per dataset and rank. T and MI refer to t-test/ANOVA and mutual information, respectively.

Figure 3: The F1 spectrum for the average dataset. The x-axis shows the average number of features that are preserved and the y-axis shows the F1 score of logistic regression. The legend indicates the algorithms and their corresponding shapes, as well as the chosen parameters with their matching colours. Abbreviations in legend: Mutual Information (MI), Pick l-Take Away r (PTA), Machine Learning algorithm (ML), Support Vector Machine (svm), random forest (rf)

6 Results

In this section we analyze the results or all the proposed experiments.

6.1 Feature Selection Exploration Results

The results in Figure 2 show a clear difference in score quality across different datasets. Only for the Arcene dataset a significant difference was observed between using Mutual Information and T-test/ANOVA. Therefore it seems that the ranking method type has less influence on the measurement quality. One interesting aspect was that methods using Mutual Information had a longer computation time than methods using the T-test/ANOVA.

6.2 Feature Selection Algorithms Evaluation Results

Figure 3 shows that all wrapper algorithms preserved less than 65 features for these settings, whereas the performance seems to average around 75%. The filter and embedded methods performed worse, with an overall lower performance score than the wrapper methods, even when more features were present. No immediate conclusions can be drawn from only the filter and embedded algorithm results.

When only looking at the wrapper algorithms, some other observations could be done as well. Ordering the features before using a wrapper method structurally gave a better result than using a random ordering. Also a threshold of $\alpha = 0.001$ usually resulted in more features and in a higher scores in comparison with a threshold of $\alpha = 0.01$. Comparing the algorithms, the floating search with ordering did best in performance, whereas the other algorithms are performing more similarly.

6.3 Evaluation of the TPOT Feature Selection Integration Results

The results of this experiment set are summarized in Table 5. The optimization process is also recorded and shown in a plot for the four datasets (Figure 4). The five experiment reruns are averaged to one complete result.

Figure 4: The optimization process for the different *TPOT* algorithms for the micro-organisms dataset.

Table 5: The performance of the final pipeline for the different types of *TPOT* and the mentioned dataset. The five reruns are averaged into this one result.

Algorithm	selection availability	regular selection		always feature selection	
		regular algorithms	feature selection algorithms	regular algorithms	feature selection algorithms
Datasets	Micro-organisms	0.45	0.71	0.64	0.59
	Arcene	0.46	0.69	0.62	0.69
	RSCTC	0.00	0.22	0.17	0.44
	Psoriasis	0.00	0.24	0.16	0.54

The results show that after two hours of running *TPOT*, regular *TPOT* always performs worst. According to the processes the high feature selection algorithm set shows much bigger stepwise improvements, accompanied with fewer smaller stepwise improvements. This shows the earlier observed trade-off between computation time and feature selection quality for these specific feature selection algorithms. It takes longer to compute those algorithms, but it also gives a much better result.

The choice for the best *TPOT* algorithm is not conclusive. For high feature selection, all three new possibilities show improved results. The *TPOT* variant that is feature selection focused and has the new feature selection algorithm set performs the best for datasets with a very large number of features (RSCTC and Psoriasis), but the trade-off in computation time seems to hurt the performance for smaller datasets. The algorithm that is only feature selection focused has a steady performance for datasets with a lower number of features, but this performance does not seem to be better than the algorithm with only the new feature selection algorithm set.

7 Discussion

There are several works focusing on the possibility of feature selection, both from a data analytical [7] as well as a biomedical [1, 2, 8] point of view. They are usually putting emphasis on datasets with a low number of features for which feature selection is less relevant. Hence, in that regard this paper complements the existing work since we handle datasets that have a significantly higher number of at least 1,000 features. The relevance of large datasets with a large number of features will only increase over time due to new and improved techniques of data acquisition and storage. We believe that our work is a good start towards the challenge of tackling feature selection on such big datasets.

From the various feature selection methods that were discussed, several were not tested due to computation time constraints. The wrapper methods backwards elimination and simulated annealing and the embedded backwards elimination all were too computationally intensive to become relevant for the research. In datasets with fewer features, these methods may be showing better results and could be possible candidates for feature selection.

After evaluation of the results in the first experiment setup, it could be concluded that there was not a big difference between using T-test/ANOVA or Mutual Information as a ranking method. Both gave similar results, with the exception of one dataset.

Also a conclusion can be drawn from looking at the accuracy with the number of features preserved. After a threshold of 200 features, additional features did not raise

the validation score as much as the first 200 features did. This indicated a second rule of thumb, that at least 200 features should be preserved after using a filter method. This absolute number was not intuitive and much lower than expected. More research is needed on these preserved 200 features and why no other features were needed to predict the output. Cluster analyses for example may show insights in this phenomenon.

After evaluation of all three kinds of methods -filter, wrapper and embedded methods - wrapper methods were significantly better at selecting a smaller fraction of features while preserving a similar test score. Our intuitive expectation was that wrapper methods would be the best in efficiency and embedded methods were expected to outperform both filter methods and wrapper methods when looking at both quality and computation time. This was not the case, however, as embedded methods had a near identical quality to filter methods and performed much worse than wrapper methods. The outcome indicated that wrapper methods are significantly different in results and filter and embedded methods being nearly identical.

Since, unlike filter methods, wrapper methods take dependencies between features into account, they kept these dependencies at a minimum. This makes wrapper methods more suitable in handling multicollinearity and are obviously more useful than filter methods. If dependencies between features themselves are important, wrapper methods might not be the best choice. Further research can be done to investigate this. A downside of the wrapper methods however was that they took much more computation time than the filter and embedded methods. Therefore if it does not matter when features have dependencies with each other, a filter method should be preferred.

There was a difference in quality within the wrapper methods. The forward selection and PTA all showed promising results and therefore should be considered for the framework. Floating search showed the best results, however took significantly longer in computation time. Within the wrapper methods, an ordering beforehand showed improved results. Backward elimination wrapper algorithms, stochastic search algorithms and embedded backwards elimination algorithms all were significantly worse in computation time than the other wrapper algorithms and therefore should not be considered in these cases.

A recommendation for the threshold in wrapper methods is not trivial. A higher threshold of $\alpha = 0.01$ gave a smaller feature subset at the cost of a lower classification score. If a smaller subset is desired or more influential features are needed, a bigger threshold should be chosen, whereas it will be smaller when a higher quality of the feature subset is desired.

Both additions to *TPOT* were beneficial according to the experiment. When the initial number of features was very high (50.000), a combination of initial bias towards feature selection and using feature selection algorithms focusing on preservation of only 200 features performed best. For lower numbers of features, only adding one of the improvements was better, as computation time became the limiting factor when using both improvements.

Acknowledgements
Not applicable

Funding
Not applicable

Abbreviations
Not applicable

Availability of data and materials
All methods and additions to TPOT are implemented in *Python* version 3.6 with *Anaconda* version 3 environment. The code is available at https://github.com/TimBeishuizen/cBioF.

Ethics approval and consent to participate
Not applicable

Competing interests
The authors declare that they have no competing interests.

Consent for publication
Not applicable

Authors' contributions
TB implemented the tools, performed the simulations and wrote the major part of the text. DB and JV participated in the designing of the project, writing of the text and developing of the algorithms. PH participated in the designing of the project and writing of the text.

Author details
Eindhoven University of Technology, Eindhoven, Netherlands.

References

1. Baumgartner, R., Somorjai, R.L.: Data complexity assessment in undersampled classification of high-dimensional biomedical data. Pattern Recognition Letters **27**(12), 1383–1389 (2006)
2. Welthagen, W., Shellie, R.A., Spranger, J., Ristow, M., Zimmermann, R., Fiehn, O.: Comprehensive two-dimensional gas chromatography–time-of-flight mass spectrometry (gc× gc-tof) for high resolution metabolomics: biomarker discovery on spleen tissue extracts of obese nzo compared to lean c57bl/6 mice. Metabolomics **1**(1), 65–73 (2005)
3. Lim, I.S., de Heras Ciechomski, P., Sarni, S., Thalmann, D.: Planar arrangement of high-dimensional biomedical data sets by isomap coordinates. In: Computer-Based Medical Systems, 2003. Proceedings. 16th IEEE Symposium, pp. 50–55 (2003). IEEE
4. Peng, Y., Wu, Z., Jiang, J.: A novel feature selection approach for biomedical data classification. Journal of Biomedical Informatics **43**(1), 15–23 (2010)
5. Biesiada, J., Duch, W.: Feature selection for high-dimensional data—a pearson redundancy based filter. In: Computer Recognition Systems 2, pp. 242–249. Springer, - (2007)
6. Ding, C., Peng, H.: Minimum redundancy feature selection from microarray gene expression data. Journal of bioinformatics and computational biology **3**(02), 185–205 (2005)
7. Catal, C., Diri, B.: Investigating the effect of dataset size, metrics sets, and feature selection techniques on software fault prediction problem. Information Sciences **179**(8), 1040–1058 (2009)
8. Liu, H., Li, J., Wong, L.: A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns. Genome informatics **13**, 51–60 (2002)
9. Saeys, Y., Inza, I., Larrañaga, P.: A review of feature selection techniques in bioinformatics. bioinformatics **23**(19), 2507–2517 (2007)
10. Donoho, D., Jin, J.: Higher criticism thresholding: Optimal feature selection when useful features are rare and weak. Proceedings of the National Academy of Sciences **105**(39), 14790–14795 (2008)
11. Senawi, A., Wei, H.-L., Billings, S.A.: A new maximum relevance-minimum multicollinearity (mrmmc) method for feature selection and ranking. Pattern Recognition **67**, 47–61 (2017). doi:10.1016/j.patcog.2017.01.026
12. Catal, C., Diri, B.: Investigating the effect of dataset size, metrics sets, and feature selection techniques on software fault prediction problem. Information Sciences **179**(8), 1040–1058 (2009). doi:10.1016/j.ins.2008.12.001
13. Huang, Y.-J., Chan, D.-Y., Cheng, D.-C., Ho, Y.-J., Tsai, P.-P., Shen, W.-C., Chen, R.-F.: Automated feature set selection and its application to mcc identification in digital mammograms for breast cancer detection. Sensors **13**(4), 4855–4875 (2013)
14. Tsamardinos, I., Borboudakis, G., Katsogridakis, P., Pratikakis, P., Christophides, V.: Massively-parallel feature selection for big data. arXiv preprint arXiv:1708.07178 (2017)
15. El Akadi, A., Amine, A., El Ouardighi, A., Aboutajdine, D.: A new gene selection approach based on minimum redundancy-maximum relevance (mrmr) and genetic algorithm (ga). In: Computer Systems and Applications, 2009. AICCSA 2009. IEEE/ACS International Conference On, pp. 69–75 (2009). IEEE
16. Radovic, M., Ghalwash, M., Filipovic, N., Obradovic, Z.: Minimum redundancy maximum relevance feature selection approach for temporal gene expression data. BMC bioinformatics **18**(1), 9 (2017)
17. Jong, K., Marchiori, E., Sebag, M., Van Der Vaart, A.: Feature selection in proteomic pattern data with support vector machines. In: Computational Intelligence in Bioinformatics and Computational Biology, 2004. CIBCB'04. Proceedings of the 2004 IEEE Symposium On, pp. 41–48 (2004). IEEE
18. Chen, Y.-W., Lin, C.-J.: Combining svms with various feature selection strategies. In: Feature Extraction, pp. 315–324. Springer, - (2006)
19. Xing, E.P., Jordan, M.I., Karp, R.M., *et al.*: Feature selection for high-dimensional genomic microarray data. In: ICML, vol. 1, pp. 601–608 (2001). Citeseer

20. Song, Q., Ni, J., Wang, G.: A fast clustering-based feature subset selection algorithm for high-dimensional data. IEEE transactions on knowledge and data engineering **25**(1), 1–14 (2013)
21. Yang, C., Fan, J., Wu, Z., Udell, M.: Efficient AutoML Pipeline Search with Matrix and Tensor Factorization (2020). 2006.04216
22. Drori, I., Liu, L., Nian, Y., Koorathota, S.C., Li, J.S., Moretti, A.K., Freire, J., Udell, M.: AutoML using Metadata Language Embeddings (2019). 1910.03698
23. Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., Hutter, F.: Efficient and robust automated machine learning. In: Advances in Neural Information Processing Systems, pp. 2962–2970 (2015)
24. Georges, N., Mhiri, I., Rekik, I., Initiative, A.D.N., *et al.*: Identifying the best data-driven feature selection method for boosting reproducibility in classification tasks. Pattern Recognition **101**, 107183 (2020)
25. Pes, B.: Learning from high-dimensional biomedical datasets: the issue of class imbalance. IEEE Access **8**, 13527–13540 (2020)
26. Panicker, S.S., Gayathri, P.: Feature selection algorithms in medical data classification: A brief survey and experimentation. ICDSMLA 2019, 831–841 (2020)
27. Prados, J., Kalousis, A., Sanchez, J.-C., Allard, L., Carrette, O., Hilario, M.: Mining mass spectra for diagnosis and biomarker discovery of cerebral accidents. Proteomics **4**(8), 2320–2332 (2004)
28. Guyon, I., Elisseeff, A.: An Introduction to Feature Extraction, pp. 1–25. Springer, Berlin, Heidelberg (2006). doi:10.1007/978-3-540-35488-8_1. https://doi.org/10.1007/978-3-540-35488-8_1
29. Molina, L.C., Belanche, L., Nebot, À.: Feature selection algorithms: A survey and experimental evaluation. In: Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference On, pp. 306–313 (2002). IEEE
30. Hall, M.A., Smith, L.A.: Practical feature subset selection for machine learning (1998)
31. Nair, R.P., Duffin, K.C., Helms, C., Ding, J., Stuart, P.E., Goldgar, D., Gudjonsson, J.E., Li, Y., Tejasvi, T., Feng, B.-J., *et al.*: Genome-wide scan reveals association of psoriasis with il-23 and nf-κb pathways. Nature genetics **41**(2), 199–204 (2009)
32. Suárez-Farinas, M., Li, K., Fuentes-Duculan, J., Hayden, K., Brodmerkel, C., Krueger, J.G.: Expanding the psoriasis disease profile: interrogation of the skin and serum of patients with moderate-to-severe psoriasis. Journal of Investigative Dermatology **132**(11), 2552–2564 (2012)
33. Bigler, J., Rand, H.A., Kerkof, K., Timour, M., Russell, C.B.: Cross-study homogeneity of psoriasis gene expression in skin across a large expression range. PLoS One **8**(1), 52242 (2013)
34. Yao, Y., Richman, L., Morehouse, C., De Los Reyes, M., Higgs, B.W., Boutrin, A., White, B., Coyle, A., Krueger, J., Kiener, P.A., *et al.*: Type i interferon: potential therapeutic target for psoriasis? PloS one **3**(7), 2737 (2008)
35. Wojnarski, M., Janusz, A., Nguyen, H.S., Bazan, J., Luo, C., Chen, Z., Hu, F., Wang, G., Guan, L., Luo, H., *et al.*: Rsctc'2010 discovery challenge: Mining dna microarray data for medical diagnosis and treatment. In: International Conference on Rough Sets and Current Trends in Computing, pp. 4–19 (2010). Springer
36. Guyon, I., Gunn, S., Ben-Hur, A., Dror, G.: Result analysis of the nips 2003 feature selection challenge. In: Saul, L.K., Weiss, Y., Bottou, L. (eds.) Advances in Neural Information Processing Systems 17, pp. 545–552. MIT Press, - (2005). http://papers.nips.cc/paper/2728-result-analysis-of-the-nips-2003-feature-selection-challenge.pdf
37. Mahé, P., Arsac, M., Chatellier, S., Monnin, V., Perrot, N., Mailler, S., Girard, V., Ramjeet, M., Surre, J., Lacroix, B., van Belkum, A., Veyrieras, J.-B.: Automatic identification of mixed bacterial species fingerprints in a maldi-tof mass-spectrum. Bioinformatics **30**(9), 1280–1286 (2014). doi:10.1093/bioinformatics/btu022. /oup/backfile/content$_public/journal/bioinformatics/30/9/10.1093_bioinformatics_btu022/3/btu022.pdf$
38. Brazma, A., Hingamp, P., Quackenbush, J., Sherlock, G., Spellman, P., Stoeckert, C., Aach, J., Ansorge, W., Ball, C.A., Causton, H.C., *et al.*: Minimum information about a microarray experiment (miame)—toward standards for microarray data. Nature genetics **29**(4), 365 (2001)
39. Selvaraj, S., Natarajan, J.: Microarray data analysis and mining tools. Bioinformation **6**(3), 95 (2011)
40. Cottrell, J.S., London, U.: Probability-based protein identification by searching sequence databases using mass spectrometry data. electrophoresis **20**(18), 3551–3567 (1999)
41. Dettmer, K., Aronov, P.A., Hammock, B.D.: Mass spectrometry-based metabolomics. Mass spectrometry reviews **26**(1), 51–78 (2007)
42. Matthiesen, R., Jensen, O.N.: Analysis of mass spectrometry data in proteomics. In: Bioinformatics, pp. 105–122. Springer, Berlin, Heidelberg (2008)
43. Watson, J.T., Sparkman, O.D.: Introduction to Mass Spectrometry: Instrumentation, Applications, and Strategies for Data Interpretation. John Wiley & Sons, ??? (2007)
44. Madigan, M.T., Martinko, J.M., Parker, J., *et al.*: Brock Biology of Microorganisms vol. 13. Pearson, ??? (2017)

Figures

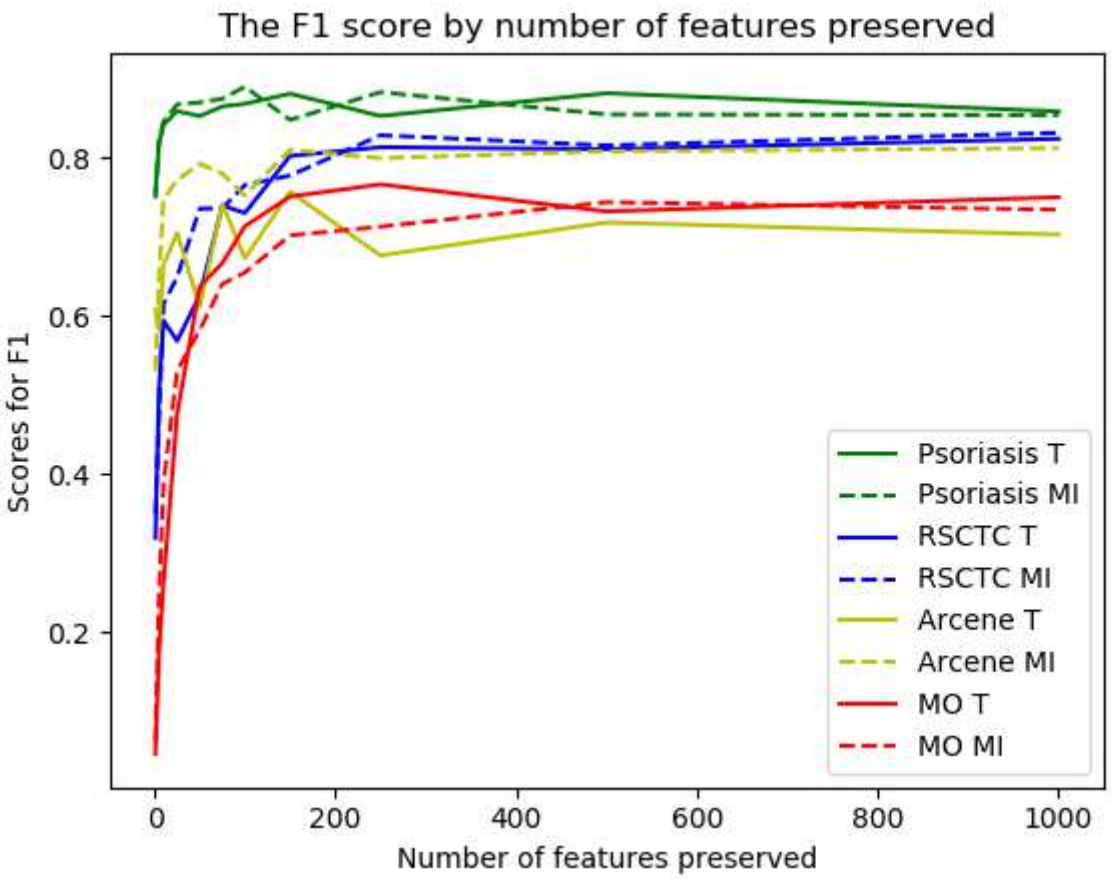

Figure 1

An example of the impact of the correction factor on the score, in this case accuracy. The shown correction factor uses β = 0:005. On the x-axis the number of features is shown and on the y-axis the value for the original accuracy, the correction factor and the FS_score.

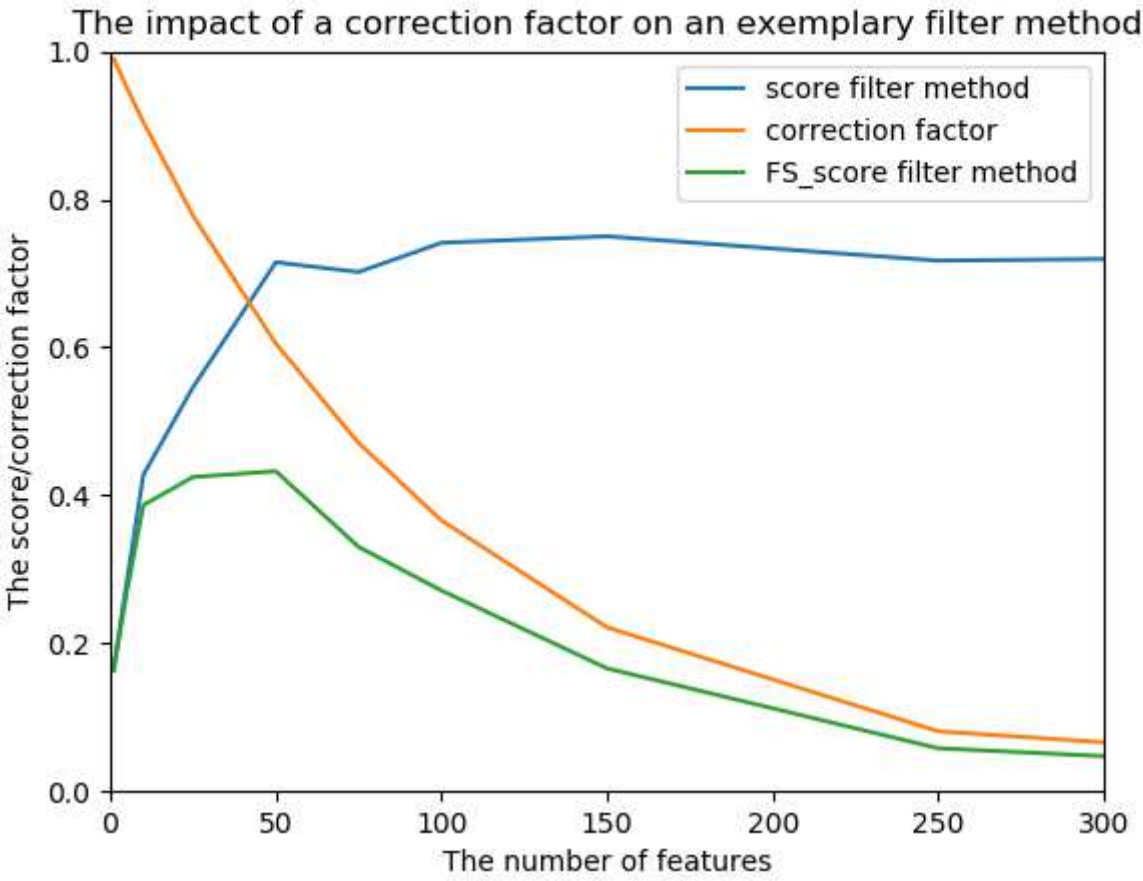

Figure 2

The average validation F1-scores shown per dataset and rank. T and MI refer to t-test/ANOVA and mutual information, respectively.

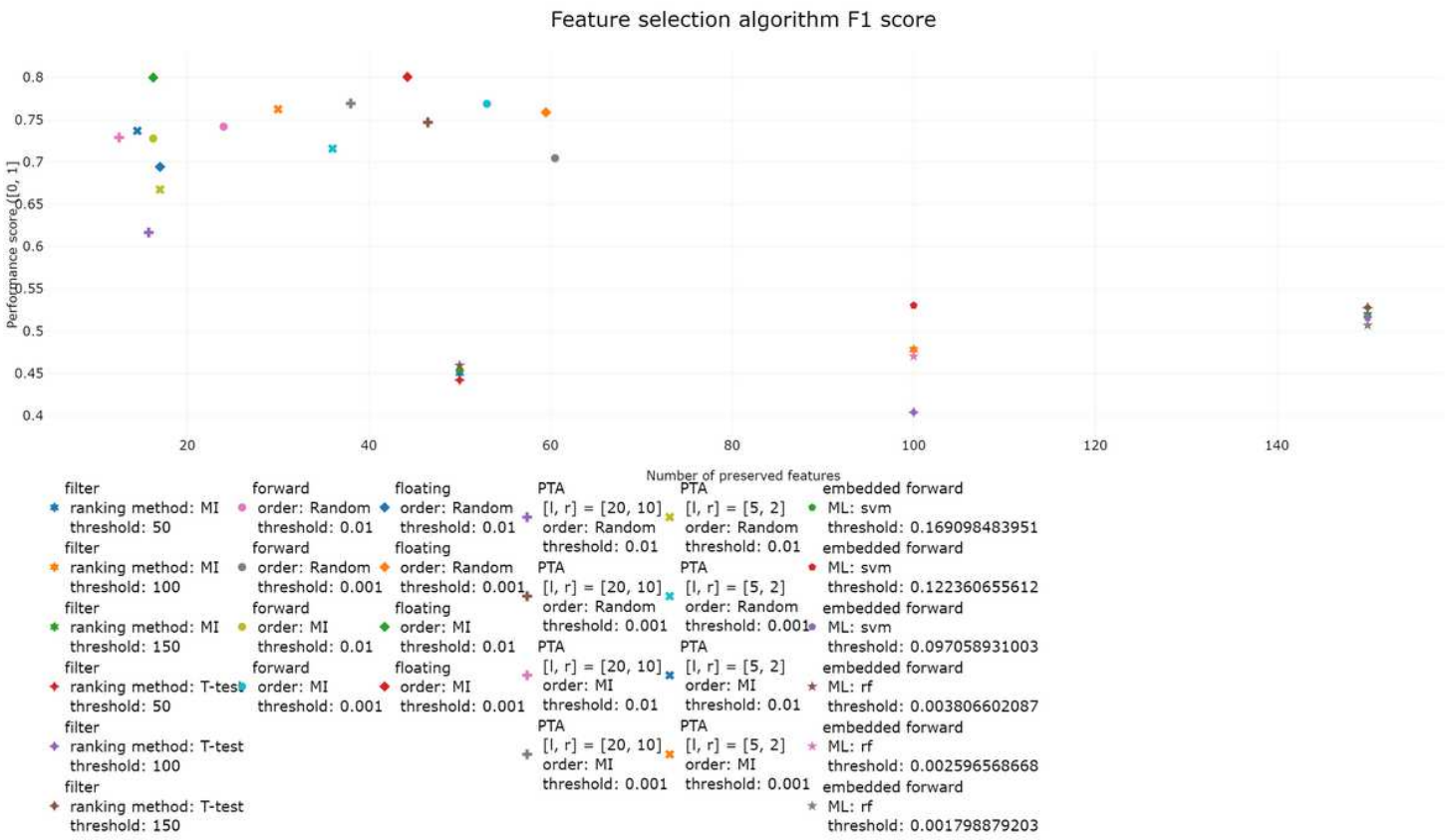

Figure 3

The F1 spectrum for the average dataset. The x-axis shows the average number of features that are preserved and the y-axis shows the F1 score of logistic regression. The legend indicates the algorithms and their corresponding shapes, as well as the chosen parameters with their matching colours. Abbreviations in legend: Mutual Information (MI), Pick l-Take Away r (PTA), Machine Learning algorithm (ML), Support Vector Machine (svm), random forest (rf)

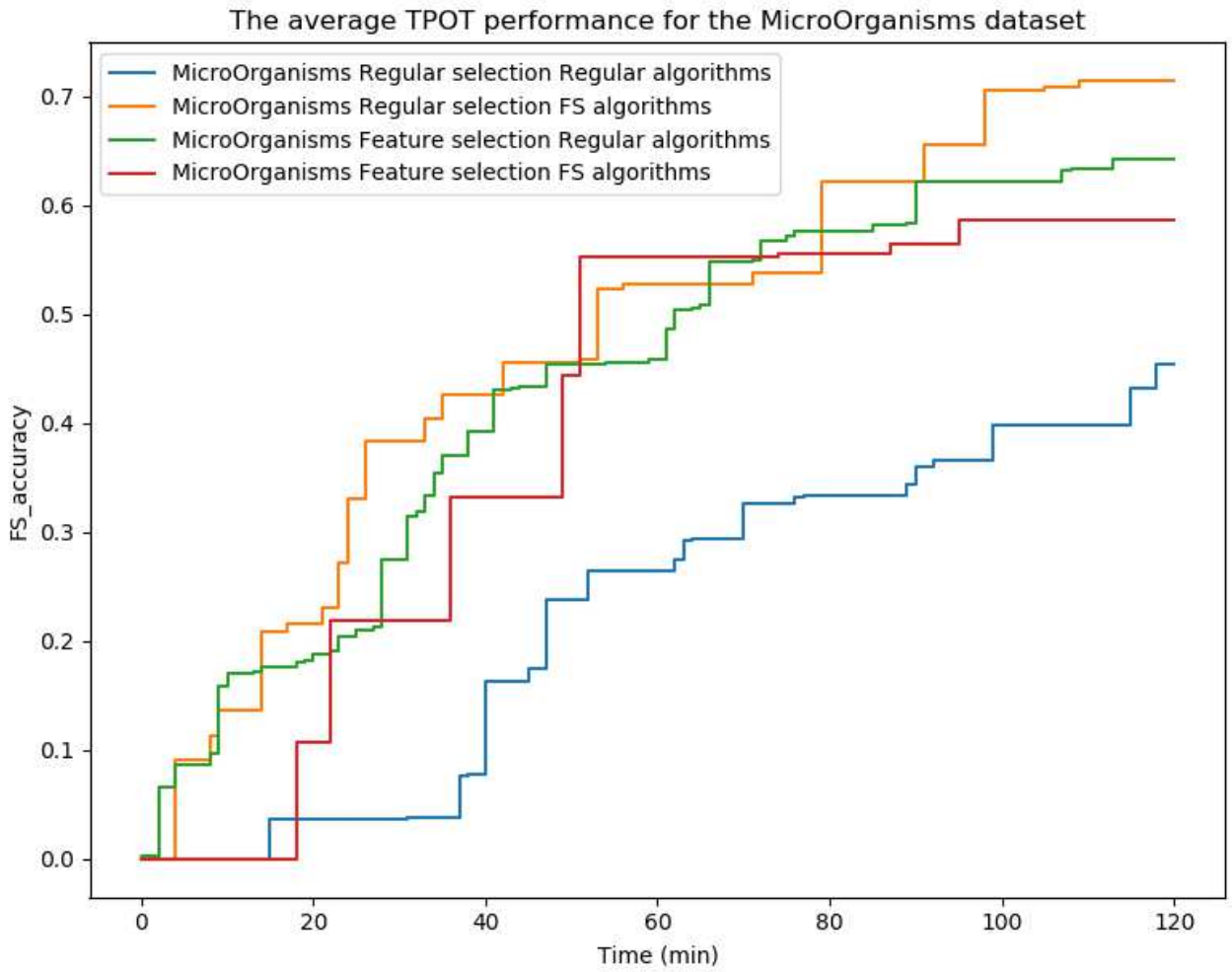

Figure 4

The optimization process for the different TPOT algorithms for the micro-organisms dataset.