Hindawi Publishing Corporation
Computational and Mathematical Methods in Medicine
Volume 2016, Article ID 1782732, 8 pages
http://dx.doi.org/10.1155/2016/1782732

Research Article

Use of a Novel Grammatical Inference Approach in Classification of Amyloidogenic Hexapeptides

Wojciech Wieczorek[1] and Olgierd Unold[2]

[1]*Faculty of Computer Science and Materials Science, University of Silesia, Ulica Zytnia 12, 41-200 Sosnowiec, Poland*
[2]*Department of Computer Engineering, Faculty of Electronics, Wroclaw University of Science and Technology, Wybrzeże Wyspianskiego 27, 50-370 Wroclaw, Poland*

Correspondence should be addressed to Wojciech Wieczorek; wojciech.wieczorek@us.edu.pl

Received 22 October 2015; Accepted 17 February 2016

Academic Editor: Humberto González-Díaz

The present paper is a novel contribution to the field of bioinformatics by using grammatical inference in the analysis of data. We developed an algorithm for generating star-free regular expressions which turned out to be good recommendation tools, as they are characterized by a relatively high correlation coefficient between the observed and predicted binary classifications. The experiments have been performed for three datasets of amyloidogenic hexapeptides, and our results are compared with those obtained using the graph approaches, the current state-of-the-art methods in heuristic automata induction, and the support vector machine. The results showed the superior performance of the new grammatical inference algorithm on fixed-length amyloid datasets.

1. Introduction

Grammatical inference (GI) is an intensively studied area of research that sits at the intersection of several fields including formal languages, machine learning, language processing, and learnability theory. The main task of the field is about finding some unknown rule when given some elements: examples and counterexamples. This presentation of elements may be finite (in practice) or infinite (in theory). As this study will be especially focused on obtaining a regular expression from finite positive and negative data, the various models of incremental learning and their decidability questions have not been mentioned. The book by de la Higuera [1] can be of major help on such theoretical aspects of grammatical inference.

Here and subsequently $S = (S_+, S_-)$ stands for a sample where S_+ is the set of examples and S_- is the set of counterexamples over a fixed alphabet Σ. Our aim is to obtain a compact description of a finite language L satisfying all the following conditions: (i) $L \subset \Sigma^+$, (ii) $S_+ \subseteq L$, and (iii) $S_- \cap L = \emptyset$. We will consider a star-free regular expression (i.e., without the Kleene closure operator) as the compact description of a language L. It is worthy to emphasize that such a formulation of an induction problem is justified by intended applications in bioinformatics. A sample in biological or medical domains consists of positive and negative objects (mainly proteins) with certain properties, whereas a star-free regular expression may serve to predict new objects. The data explored by Tian et al. [2] and Maurer-Stroh et al. [3] are good illustrations. They consist of examples and counterexamples of amyloids, that is, proteins which have been associated with the pathology of more than 20 serious human diseases. In the experimental part of the present paper, we are going to undertake an examination of binary classification efficiency for selected real biological/medical data. By binary classification, we mean mapping a string to one out of two classes by means of induced regular expressions (regex). For classification, especially for two-class problems, a variety of measures has been proposed. Since our experiments lie in a (bio)medical context, the Matthews Correlation Coefficient is regarded as a primary score, as the goal of this whole process is to predict new strings that are likely to be positive.

There is a number of closely related works to our study. Angluin showed that the problem of inferring minimum-size regular expression satisfying (i), (ii), and (iii) remains NP-complete even if a regex is required to be star-free

(containing no "$*$" operations) [4]. In our previous work [5] similar bioinformatics datasets have been analyzed, but with different acceptors—directed acyclic word graphs. Some of classical automata learning algorithms like ECGI [6], k-RI [7], and k-TSSI [8] could be applied to the problem, but they do not make use of counterexamples. Many authors advocated the benefit of viewing the biological sequences as sentences derived from a formal grammar or automaton. As a good bibliographical starting point, see articles by Coste and Kerbellec [9], Sakakibara [10], and Searls [11]. In connection with this problem of data classification, it is worth remembering that there is a field of computer science that can be also involved, namely, machine learning (ML), which includes such methods as classification trees, clustering, the support vector machine [12], and rough sets [13]. All above-mentioned ML methods are aimed at compact description of input data, though in various ways. In view of our applications, they have, however, a drawback. The problem is that they are not suited for variable-length data.

In the present algorithm a star-free regular expression (SFRE) is achieved based on a learning sample containing the examples and counterexamples (these examples and counterexamples are also called positive and negative words). It is a two-phase procedure. In the first phase an initial graph is built in order to reveal possible substring interchanges. In the second phase all maximal cliques of the graph are yielded to build a SFRE. We have implemented our induction algorithm of a SFRE and started applying it to a real bioinformatics task, that is, classification of amyloidogenic hexapeptides. Amyloids are proteins capable of forming fibrils instead of the functional structure of a protein [14] and are responsible for a group of diseases called amyloidosis, such as Alzheimer's, Huntington's disease, and type II diabetes [15]. Furthermore, it is believed that short segments of proteins, like hexapeptides consisting of 6-residue fragments, can be responsible for amyloidogenic properties [16]. Since it is not possible to experimentally test all such sequences, several computational tools for predicting amyloid chains have emerged, inter alia, based on physicochemical properties [17] or using machine learning approach [18–21].

To test the performance of our SFRE approach, the following six additional programs have been used in experiments: the implementation of the Trakhtenbrot-Barzdin state merging algorithm, as described in [22]; the implementation of Rodney Price's Abbadingo winning idea of evidence-driven state merging [23]; a program based on the Rlb state merging algorithm [24]; ADIOS (for Automatic Distillation of Structure)—a context-free grammar learning system, which relies on a statistical method for pattern extraction and on structured generalization [25]; our previous approach with directed acyclic word graphs [5]; and, as an instance of ML methods, the support vector machine [26].

A rigorous statistical procedure has been applied to compare all the above methods in terms of a correlation between the observed and predicted binary classification (Matthews Correlation Coefficient, MCC). The proposed approach significantly outperforms both GI-based methods and ML algorithm on fixed-length amyloid datasets.

2. Materials and Methods

2.1. Datasets. The algorithm for generating star-free regular expressions SFRE has been tested over three recently published Hexpepset datasets, that is, Waltz [3], WALTZ-DB [27], and exPafig [5]. The first two databases consist of only experimentally asserted amyloid sequences. Note that the choice of experimental verified short peptides is very limited since very few data are available. The Waltz dataset has been published in 2010 and is composed of 116 hexapeptides known to induce amyloidosis (S_+) and by 161 hexapeptides that do not induce amyloidosis ($S\_$). The WALTZ-DB has been prepared by the same science team in the Switch Lab from KU Leuven and published in 2015. This dataset expands the Waltz set to total number of hexapeptides of 1089. According to Beerten et al. (2015), additional 720 hexapeptides were derived from 63 different proteins and combined with 89 peptides taken from the literature [27]. In the WALTZ-DB database, 244 hexapeptides are regarded as positive for amyloid formation (S_+) and 845 hexapeptides as negative for amyloid formation ($S\_$).

SFRE algorithm was also validated and trained on database (denoted by exPafig), which was computationally obtained with Pafig method [2], and then statistically processed [5]. exPafig consists of 150 amyloid positive hexapeptides (S_+) and 2259 negative hexapeptides ($S\_$). As seen, the database is strongly imbalanced.

2.2. An Algorithm for the Induction of a SFRE

2.2.1. Definitions

Definition 1. Σ will be a finite nonempty set, the *alphabet*. Σ^+ will denote the set of all nonempty strings over the alphabet Σ. If $s, t \in \Sigma^+$, the concatenation of s and t, written st, will denote the string formed by making a copy of s and following it by a copy of t. If $A, B \subseteq \Sigma^+$, then

$$AB = \{s \mid s = tu \text{ for some } t \in A,\ u \in B\}. \tag{1}$$

To simplify the representations for finite languages, we define the notion of star-free regular expressions over alphabet Σ as follows.

Definition 2. The set of *star-free regular expressions* (SFREs) over Σ will be the set of strings R such that

(1) $\emptyset \in R$ which represents the empty set;

(2) $\Sigma \subseteq R$; each element a of the alphabet represents language $\{a\}$;

(3) if r_A and r_B are SFREs representing languages A and B, respectively, then $(r_A + r_B) \in R$ and $(r_A r_B) \in R$ representing $A \cup B$, AB, respectively, where the symbols $(,), +$ are not in Σ.

We will freely omit unnecessary parentheses from SFREs assuming that concatenation has higher priority than union. If $r \in R$ represents language A, we will write $L(r) = A$.

Definition 3. A *sample* S over Σ will be an ordered pair $S = (S_+, S_-)$ where S_+, S_- are finite subsets of Σ^+ and $S_+ \cap S_- = \emptyset$. S_+ will be called the *positive part of* S, and S_- the *negative part of* S. A star-free regular expression r is *consistent* (or *compatible*) with a sample $S = (S_+, S_-)$ if and only if $S_+ \subseteq L(r)$ and $S_- \cap L(r) = \emptyset$.

Definition 4. A *graph* G is a finite nonempty set of objects called *vertexes* together with a (possibly empty) set of unordered pairs of distinct vertexes of G called *edges*. The vertex set of G is denoted by $V(G)$, while the edge set is denoted by $E(G)$. The edge $e = \{u, v\}$ is said to *join* the vertexes u and v. If $e = \{u, v\}$ is an edge of a graph G, then u and v are *adjacent vertexes*. In a graph G, a *clique* is a subset of the vertex set $C \subseteq V(G)$ such that every two vertexes in C are adjacent. By definition, a clique may be also composed of only one vertex. If a clique does not exist exclusively within the vertex set of a larger clique, then it is called a *maximal clique*.

Definition 5. Let Σ be an alphabet and let G be a graph. Suppose that every vertex in G is associated with an ordered pair of nonempty strings over Σ; that is, $V(G) = \{v_1, v_2, \ldots, v_n\}$, where $v_i = (u_i, w_i) \in \Sigma^+ \times \Sigma^+$ for $1 \leq i \leq n$. Let $C = \{v_{i_1}, v_{i_2}, \ldots, v_{i_m}\}$ be a clique in G. Then

$$r(C) = \left(u_{i_1} + u_{i_2} + \cdots + u_{i_m}\right)\left(w_{i_1} + w_{i_2} + \cdots + w_{i_m}\right) \quad (2)$$

is a star-free regular expression over Σ *induced by* C.

For the simplicity's sake, we also denote the set $L(u_{i_1} + \cdots + u_{i_m}) = \{u_{i_1}, \ldots, u_{i_m}\}$ by U and the set $L(w_{i_1} + \cdots + w_{i_m}) = \{w_{i_1}, \ldots, w_{i_m}\}$ by W in the context of C.

2.2.2. The Algorithm. In this section, we are going to show how to generate a SFRE compatible with a given sample. These expressions do not have many theoretical properties but have marvelous accomplishment in the analysis of some bioinformatics data in terms of classification quality.

Let $S = (S_+, S_-)$ be a sample over Σ in which every string is at least of length 2. Construct the graph G with vertex set

$$V(G) = \bigcup_{s \in S_+} \{(u, w) \mid s = uw,\ u, w \in \Sigma^+\} \quad (3)$$

and with edge set $E(G)$ given by

$$\{(u, w), (x, y)\} \in E(G) \Longleftrightarrow |u| = |x|,\ uy \notin S_-,\ xw \notin S_-. \quad (4)$$

Next, find a set of cliques $\mathscr{C} = \{C_1, C_2, \ldots, C_k\}$ in G such that $S_+ \subseteq \sum_{i=1}^{k} r(C_i)$. For this purpose one can take advantage of an algorithm proposed by Tomita et al. [28] for generating all maximal cliques. Although it takes $O(n3^{n/3})$ time in the worst case for an n-vertex graph, computational experiments described in Section 3 demonstrate that it runs very fast in practice (a few seconds for thousands of vertexes). Finally, return the union of SFREs induced by all maximal cliques $\mathscr{C}$; that is, $e = r(C_1) + r(C_2) + \cdots + r(C_k)$.

In order to reduce the computational complexity of the induction, instead of Tomita's algorithm, the ensuing randomized procedure could be applied. Consecutive cliques C_i with their catenations U_iW_i are determined until $S_+ \subseteq \bigcup_{i=1}^{k} U_iW_i$. The catenations emerge in the following manner. In step $i + 1$, a vertex $v_{s_1} = (u, w) \in V(G)$ for which $uw \notin \bigcup_{m=1}^{i} U_mW_m$ is chosen at random. Let $U_{i+1} = \{u\}$ and $W_{i+1} = \{w\}$. Then sets U_{i+1} and W_{i+1} are updated by adding words from the randomly chosen neighbor of v_{s_1}, say v_{s_2}, and subsequently by adding words from the randomly chosen neighbor v_{s_3} of $\{v_{s_1}, v_{s_2}\}$, and so forth. In the end, a maximal clique C_{i+1} is obtained for which $L(r(C_{i+1})) = U_{i+1}W_{i+1}$. Naturally, $e = r(C_1) + r(C_2) + \cdots + r(C_k)$ fulfills $S_+ \subseteq L(e)$, and the whole procedure runs in polynomial time with respect to the input size.

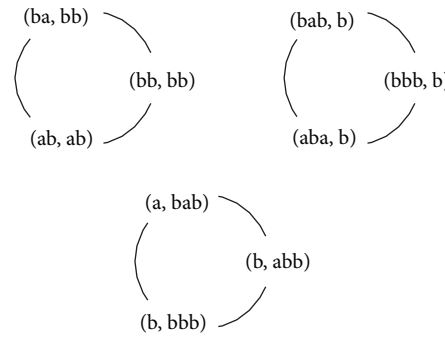

Figure 1: A graph G built from a sample S, according to definitions (3) and (4).

Here are some elementary properties of a resultant expression e and the complexity of the induction algorithm.

(i) $S_- \cap L(e) = \emptyset$ is implied from (4).

(ii) If all strings in a sample have equal length, let us say ℓ, then all strings from $L(e)$ also are of the same length ℓ.

(iii) Let $n = \sum_{s \in S} |s|$. A graph G, based on (3) and (4), may be constructed in $O(n^3)$ time. Determining a set of cliques $\mathscr{C}$ and corresponding regular expressions $r(C_1), r(C_2), \ldots, r(C_k)$ also takes no more than $O(n^3)$ time, assuming that the graph is represented by adjacency lists. Thus, the overall computational complexity is $O(n^3)$.

2.2.3. An Illustrative Run. Suppose $S = (\{\texttt{bbbb}, \texttt{babb}, \texttt{abab}\}, \{\texttt{bbba}, \texttt{baba}, \texttt{baaa}, \texttt{abaa}, \texttt{aaba}, \texttt{aaab}\})$ is a sample (one of possible explanations for the input is, each a follows at least one b). A constructed graph G is depicted in Figure 1. It has three maximal cliques and regardless of the method—either Tomita's or randomized algorithm was selected—all of them would be determined in this case. The final SFRE induced by the cliques is

$$e = (\texttt{ab} + \texttt{ba} + \texttt{bb})(\texttt{bb} + \texttt{ab}) + (\texttt{aba} + \texttt{bbb} + \texttt{bab})(\texttt{b}) + (\texttt{b} + \texttt{a})(\texttt{bab} + \texttt{bbb} + \texttt{abb}). \quad (5)$$

Among all words of length four over the alphabet $\{\texttt{a}, \texttt{b}\}$ it does not accept aaaa, baaa, abaa, bbaa, aaba, baba, abba,

bbba, aaab, but accepts baab, abab, bbab, aabb, babb, abbb, bbbb.

2.3. Validation with Other Methods. The SFRE classification quality over hexapeptides from three datasets was compared to three state-of-the-art tools for heuristic state merging DFA induction: the Trakhtenbrot-Barzdin state merging algorithm (denoted Traxbar) [22], Rodney Price's Abbadingo winning idea of evidence-driven state merging (Blue-fringe) [23], Rlb state merging algorithm (Rlb) [24], and a context-free grammar learning system ADIOS [25]. The compared set of methods was extended by our previous approach with directed acyclic word graphs (DAWG) [5] and the support vector machine with linear kernel function (SVM) [26].

Trakhtenbrot and Barzdin described an algorithm for constructing the smallest DFA consistent with a complete labeled training set [29]. The input to the algorithm is the prefix-tree acceptor which directly embodies the training set. This tree is collapsed into a smaller graph by merging all pairs of states that represent compatible mappings from string suffixes to labels. This algorithm for completely labeled trees has been generalized by Lang [22] to produce a (not necessarily minimum) machine consistent with a sparsely labeled tree (we used implementations from the archive http://abbadingo.cs.nuim.ie/dfa-algorithms.tar.gz for the Traxbar and for the two remaining state merging algorithms).

The second algorithm that starts with the prefix-tree acceptor for the training set and folds it up into a compact hypothesis by merging pairs of states is Blue-fringe. This program grows a connected set of red nodes that are known to be unique states, surrounded by a fringe of blue nodes that will either be merged with red nodes or be promoted to red status. Merges only occur between red nodes and blue nodes. Blue nodes are known to be the roots of trees, which greatly simplifies the code for correctly doing a merge. The only drawback of this approach is that the pool of possible merges is small, so occasionally the program has to do a low scoring merge.

The idea that lies behind the third algorithm, Rlb, is as follows. It dispenses with the red-blue restriction and is able to do merges in any order. However, to have a practical run time, only merges between nodes that lie within a distance "window" of the root on a breadth-first traversal of the hypothesis graph are considered. This introduction of a new parameter is a drawback to this program, as is the fact that its run time scales very badly with training string length. However, on suitable problems, it works better than the Blue-fringe algorithm. The detailed description of heuristics for evaluating and performing merges can be found in Lang's work [24].

ADIOS starts by loading the corpus (examples) onto a directed graph whose vertexes are all lexicon entries, augmented by two special symbols, begin and end. Each corpus sentence defines a separate path over the graph, starting at begin and ending at end, and is indexed by the order of its appearance in the corpus. Loading is followed by an iterative search for significant patterns, which are added to the lexicon as new units. The algorithm generates candidate patterns by traversing in each iteration a different search path, seeking subpaths that are shared by a significant number of partially aligned paths. The significant patterns are selected according to a context-sensitive probabilistic criterion defined in terms of local flow quantities in the graph. At the end of each iteration, the most significant pattern is added to the lexicon as a new unit, the subpaths it subsumes are merged into a new vertex, and the graph is rewired accordingly. The search for patterns and equivalence classes and their incorporation into the graph are repeated until no new significant patterns are found. The Java implementation of ADIOS made available to us by one of the authors was used in our experiments.

DAWG is a two-phase procedure. In the first phase, an initial directed graph is built in a way that resembles the construction of the minimal DFA, but nondeterminism is also allowed. In the second phase, the directed graph is extended in an iterative process by putting some additional labels onto the existing arcs. The order of putting new labels alters the results; hence a greedy heuristic has been proposed in order to obtain the words most consistent with a sample. We used the same implementation of DAWG as in our earlier work on classification of biological sequences [5].

SVM constructs a hyperplane or set of hyperplanes in a high-dimensional space, which can be used for classification, regression, or other tasks. A good separation is achieved by the hyperplane that has the largest distance to the nearest training data points of any class (so-called functional margin), since, in general, the larger the margin, the lower the generalization error of the classifier. In the experiments, we took advantage of scikit SVM, a machine learning Python library with default parameters [30].

2.4. Experiment Design and Statistical Analysis. To estimate the SFRE's and compared approaches' ability to classify unseen hexapeptides repeated stratified k-fold cross-validation (cv) strategy was used. Note that holdout method is the simplest kind of cross-validation, but multiple cv is thought to be more reliable than holdout due to its evaluation variance [31]. The simplest form of cross-validation is to split the data randomly into k mutually exclusive folds, building a model on all but one fold, and to evaluate the model on the skipped fold. The procedure is repeated k-times, each time evaluating the model on the next omitted fold. The overall assessment of the model is based on the mean of k-individual evaluations. Since the cv assessment would depend on the random assignment samples, a common practice is to stratify the folds themselves [32]. In a stratified variant of cv, the pseudorandom folds are generated in such a way that each fold contains approximately the same percentage of samples of each class as the whole set. Although the cv is considered as one of the most utilized validation methods, it is well known that cv-based estimators have high variance and nonzero bias [33–36]. It is therefore recommended to use a repeated cross-validation approach [37].

The main problem with (repeated) cv is that the training and test sets are not independent samples. Dietterich [31] found that comparing algorithms on the basis of repeated resampling of the same data can cause very high Type-I errors. It means that statistical hypothesis test, like the standard paired t-test, incorrectly rejects a true null hypothesis (so-called false positive). Note that cv can be viewed as a kind

of random subsampling. To correct the variance estimate of dependent samples, Nadeau and Bengio [38] proposed the following statistic of the *corrected resampled t-test*:

$$t_c = \frac{1/n \sum_{j=1}^{n} x_j}{\sqrt{\left(1/n + n_2/n_1\right) \widehat{\sigma}^2}}, \quad (6)$$

where x_j is the difference of the performance quality between two compared algorithms on j-run ($1 \le j \le n$). We assume that in each run n_1 samples are used for training and n_2 samples for testing. $\widehat{\sigma}^2$ stands for the variance of the n differences. This statistic obeys approximately Student's t-distribution with $n-1$ degrees of freedom. The only difference to the standard t-test is that the factor $1/n$ in the denominator is by the factor $1/n+n_2/n_1$. The corrected resampled t-test has the Type-I error close to the significance level and—opposite to the McNemar test and the 5×2 cv test—low Type-II error (i.e., the failure to reject a false null hypothesis). If we consider test based on r-times k-fold cv, the statistic

$$t_c = \frac{1/\left(k \cdot r\right) \sum_{i=1}^{k} \sum_{j=1}^{r} x_{ij}}{\sqrt{\left(1/\left(k \cdot r\right) + n_2/n_1\right) \widehat{\sigma}^2}} \quad (7)$$

has $k \cdot r - 1$ degrees of freedom and is called *corrected repeated k-fold cv test*. To detect performance differentiation of compared algorithms we use 10×10 cv scheme with 10 (instead of 99) degrees of freedom. This scheme was shown [39] to have excellent replicability. Note that, to perform multiple comparisons involving a control method (i.e., SFRE), we are supposed to control the family-wise error (FWER) [40, 41]. FWER is the probability of making Type-I error when testing many null hypotheses simultaneously. Several methods of relaxing the FWER have been proposed [42]. To keep the probability of rejecting any true null hypothesis small, in our experiments we applied Holm correction [43].

The predictive performance of algorithms was evaluated with the confusion matrix and some of the figures of merit associated with it. First, the following four scores were defined as tp, fp, fn, and tn, representing the numbers of true positives (correctly recognized amyloids), false positives (nonamyloids recognized as amyloids), false negatives (amyloids recognized as nonamyloids), and true negatives (correctly recognized nonamyloids), respectively. The following three figures of merit were considered here, since they are widely used.

The Sensitivity, also known as true positive rate, represents the percentage of correctly identified positive cases and is defined as

$$\text{Sensitivity} = \frac{\text{tp}}{\left(\text{tp} + \text{fn}\right)}. \quad (8)$$

Specificity, known is as true negative rate, represents the percentage of correctly identified negative cases and is calculated as

$$\text{Specificity} = \frac{\text{tn}}{\left(\text{tn} + \text{fp}\right)}. \quad (9)$$

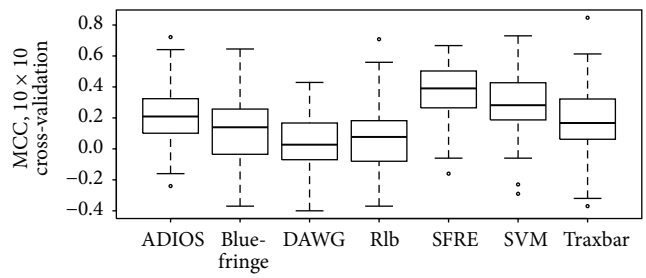

Figure 2: Performance comparison of ADIOS, Blue-fringe, DAWG, Rlb, SFRE, SVM, and Traxbar methods on Waltz database [3]. Boxplots represent the MCC values obtained from 10×10 cross-validation. The ratio of S_+/S_- is 116/161.

Table 1: p values for the comparison of the SFRE (the control algorithm) with the other methods on Waltz database. The initial level of confidence $\alpha = 0.05$ is adjusted by Holm procedure.

SFRE versus	Unadjusted p	Holm p
ADIOS	$1.872447e-04$	$3.744893e-04$
Blue-fringe	$5.341761e-09$	$2.136704e-08$
DAWG	$1.089587e-13$	$6.537519e-13$
RLB	$7.529027e-12$	$3.764514e-11$
SVM	$9.527442e-03$	$9.527442e-03$
Traxbar	$4.257174e-07$	$1.277152e-06$

Matthews Correlation Coefficient is defined as

$$\text{MCC} = \frac{\left(\text{tp} \cdot \text{tn} - \text{fp} \cdot \text{fn}\right)}{\sqrt{\left(\text{tp} + \text{fn}\right)\left(\text{tp} + \text{fp}\right)\left(\text{tn} + \text{fp}\right)\left(\text{tn} + \text{fn}\right)}}. \quad (10)$$

Note that several other scores derived from the confusion matrix can be used for estimating the prediction reliability. These three figures of merit, that is, Sensitivity, Specificity, and Matthews Correlation Coefficient, seem to be indispensable for the following reasons. Sensitivity and Specificity tend to be anticorrelated and monitor different aspects of the prediction process. Both of them may range from 0 to +1, where +1 means perfect prediction. Second, Matthews Correlation Coefficient [44] considers both the true positives and true negatives as successful predictions. MCC is always between −1 and +1. A value of −1 indicates total disagreement, 0 random prediction, and +1 perfect prediction. What is important in our case is, MCC is resistant to imbalanced dataset.

3. Result and Discussion

Figure 2 and Table 1, Figure 3 and Table 2, and Figure 4 and Table 3 summarize the performances of the SFRE algorithm and compared methods on Waltz, WALTZ-Db, and exPafig databases, respectively. The figures present boxplots representing the MCC values obtained from 10×10 cross-validation, whereas the tables give unadjusted and adjusted by Holm procedure p values for the comparison of the SFRE algorithm (the control method) with the remaining algorithms. Note that adjusted p for each method and each database is lower than desired level of a confidence α, 0.05, in

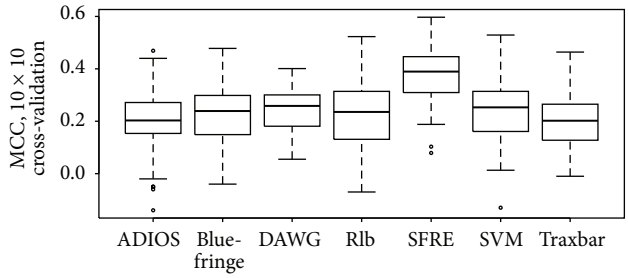

Figure 3: Performance comparison of ADIOS, Blue-fringe, DAWG, Rlb, SFRE, SVM, and Traxbar methods on WALTZ-DB database [27]. Boxplots represent the MCC values obtained from 10 × 10 cross-validation. The ratio of S_+/S_- is 240/836.

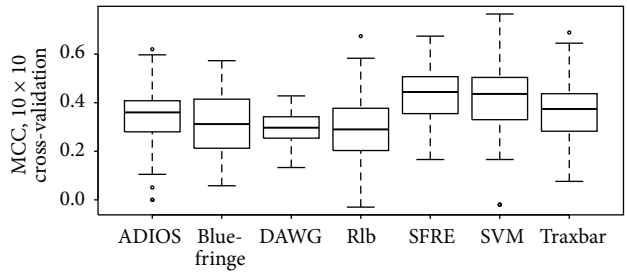

Figure 4: Performance comparison of ADIOS, Blue-fringe, DAWG, Rlb, SFRE, SVM, and Traxbar methods on exPafig database [5]. Boxplots represent the MCC values obtained from 10 × 10 cross-validation. The ratio of S_+/S_- is 150/2259.

Table 2: p values for the comparison of the SFRE (the control algorithm) with the other methods on WALTZ-DB database. The initial level of confidence $\alpha = 0.05$ is adjusted by Holm procedure.

SFRE versus	Unadjusted p	Holm p
ADIOS	$4.904483e-13$	$1.961793e-12$
Blue-fringe	$4.495071e-10$	$4.495071e-10$
DAWG	$4.838106e-14$	$2.419053e-13$
RLB	$3.326237e-11$	$6.652474e-11$
SVM	$1.703864e-11$	$5.111592e-11$
Traxbar	$9.161256e-16$	$5.496754e-15$

Table 3: p values for the comparison of the SFRE (the control algorithm) with the other methods on exPafig database. The initial level of confidence $\alpha = 0.05$ is adjusted by Holm procedure.

SFRE versus	unadjusted p	Holm p
ADIOS	$1.501499e-05$	$4.504496e-05$
Blue-fringe	$1.019785e-08$	$4.079139e-08$
DAWG	$4.319299e-12$	$2.591579e-11$
RLB	$7.401268e-11$	$3.700634e-10$
SVM	$3.295963e-02$	$3.295963e-02$
Traxbar	$1.368667e-04$	$2.737334e-04$

our experiments. These p values indicate that there are significant performance differences between SFRE algorithm and compared methods.

SFRE algorithm outperforms all other compared methods in terms of MCC over both experimentally asserted datasets, Waltz and WALTZ-DB, and computationally generated exPafig. It is worth noting that all p values except for comparing with SVM algorithm are lower than not only 0.05, but also the often used 0.01, hence confirming the superiority of the SFRE.

Comparative analysis of the three figures of merit (Sensitivity, Specificity, and Matthews Correlation Coefficient) is summarized in Table 4. These quantities are reported for seven compared predictors and three databases (Waltz, WALTZ-DB, and exPafig). Numerical results reported in Table 4 show that SFRE has the highest Average MCC (0.40) followed by SVM (0.31), ADIOS and Traxbar (0.25), Blue-fringe (0.22), and DAWG and Rlb (0.19). Furthermore, SFRE has the highest MCC score compared to the other predictors on each dataset (0.37, 0.38, and 0.44, resp.). Although the results of MCC score seem to be not high (at the level of 0.40), it should be noted that many of the amyloid predictors are reported to have similar or lower values [45]. It is also worth mentioning that all methods have gained the highest MCC values for the computationally generated exPafig dataset.

SFRE has a higher Specificity score than other methods except SVM in case of WALTZ-DB (0.95 to 0.98, resp.) and exPafig databases (both Spe of 1.00). These two predictors have a very good capacity at predicting nonamyloid hexapeptides, with Spe higher than 0.90 for each database. The counterpart is their poor Sensitivity. Concerning Sen score, DAWG, our earlier proposal, has the highest value on each database (0.90, 0.81, and 0.73, resp.). SFRE algorithm showed a low Sensitivity for each tested dataset (0.30, 0.33, and 0.25, resp.).

The evaluation of SFRE on three amyloidogenic hexapeptide datasets revealed its accuracy to predict nonamyloid segments. We showed that the new grammatical inference algorithm gives the best Matthews Correlation Coefficient in comparison to six other methods, including support vector machine.

4. Conclusions

In the present paper, the way in which regex induction may support predicting new hexapeptides has been revealed. We, therefore, studied the following problem: given a sample $S = (S_+, S_-)$, find a "general" star-free regular expression e such that $S_+ \subseteq L(e)$, $S_- \cap L(e) = \emptyset$, and $L(e) - S_+$ contain only strings of "similar characteristics" to those of S_+. To this end, a new GI method has been proposed which is especially suited to the fixed-length datasets. The conducted experiments showed that our algorithm outperforms compared methods in terms of a correlation between the observed and predicted binary classification (MCC) and with real datasets taken from a biomedical domain.

The proposed idea is not free from objections. Among the most serious complications is the exponential computational complexity of generating maximal cliques, which is the second phase of the algorithm. However, it can be overcome by using a proposed randomized procedure instead. Our first

Table 4: Performance of compared methods on Waltz, WALTZ-DB, and exPafig databases in terms of Sensitivity (Sen), Specificity (Spe), and Matthews Correlation Coefficient (MCC). The results are ordered by decreasing Average MCC (Ave MCC).

Method	Waltz			WALTZ-DB			exPafig			Ave MCC
	Sen	Spe	MCC	Sen	Spe	MCC	Sen	Spe	MCC	
SFRE	0.30	0.97	0.37	0.33	0.95	0.38	0.25	1.00	0.44	0.40
SVM	0.35	0.90	0.30	0.15	0.98	0.24	0.22	1.00	0.40	0.31
ADIOS	0.36	0.82	0.22	0.64	0.59	0.20	0.51	0.90	0.34	0.25
Traxbar	0.56	0.61	0.17	0.46	0.76	0.20	0.42	0.96	0.37	0.25
Blue-fringe	0.58	0.53	0.11	0.36	0.85	0.23	0.33	0.96	0.32	0.22
DAWG	0.90	0.13	0.04	0.81	0.47	0.24	0.73	0.80	0.30	0.19
Rlb	0.36	0.70	0.07	0.26	0.90	0.22	0.25	0.97	0.29	0.19

experiments on larger datasets uncovered that this is a good direction for the future research.

The high Sensitivity of DAWG approach and high Specificity of SFRE method over tested databases suggest the second direction of future research. These two classifiers could be combined into a metapredictor having, hopefully, both good Sensitivity and Specificity. Such meta-approaches are reported to gain often better results in terms of aggregate indicators (as MCC) than individual predictors [45].

Competing Interests

The authors declare no conflict of interests.

Authors' Contributions

Wojciech Wieczorek proposed and implemented SFRE algorithm; Olgierd Unold designed the methodology and experiments. Wojciech Wieczorek conceived and performed the experiments; Olgierd Unold designed and performed the statistical data analysis. Both authors wrote and approved the final paper.

Acknowledgments

This research was supported by National Science Center (Grant DEC-2011/03/B/ST6/01588) and by a statutory grant of the Wroclaw University of Technology.

References

[1] C. de la Higuera, *Grammatical Inference: Learning Automata and Grammars*, Cambridge University Press, 2010.

[2] J. Tian, N. Wu, J. Guo, and Y. Fan, "Prediction of amyloid fibril-forming segments based on a support vector machine," *BMC Bioinformatics*, vol. 10, supplement 1, article S45, 2009.

[3] S. Maurer-Stroh, M. Debulpaep, N. Kuemmerer et al., "Exploring the sequence determinants of amyloid structure using position-specific scoring matrices," *Nature Methods*, vol. 7, no. 3, pp. 237–242, 2010.

[4] D. Angluin, *An application of the theory of computational complexity to the study of inductive inference [Ph.D. thesis]*, University of California, Oakland, Calif, USA, 1976.

[5] W. Wieczorek and O. Unold, "Induction of directed acyclic word graph in a bioinformatics task," in *Proceedings of the 12th International Conference of Grammatical Inference*, vol. 34 of *JMLR Workshop and Conference Proceedings*, pp. 207–217, Kyoto, Japan, September 2014.

[6] H. Rulot and E. Vidal, "Modelling (sub)string length based constraints through a grammatical inference method," in *Pattern Recognition Theory and Applications*, P. A. Devijver and J. Kittler, Eds., vol. 30 of *NATO ASI Series*, pp. 451–459, Springer, 1987.

[7] D. Angluin, "Inference of reversible languages," *Journal of the ACM*, vol. 29, no. 3, pp. 741–765, 1982.

[8] P. Garcia, E. Vidal, and J. Oncina, *Learning Locally Testable Languages in the Strict Sense*, ALT, 1990.

[9] F. Coste and G. Kerbellec, "Learning automata on protein sequences," in *7th Journées Ouvertes Biologie Informatique Mathématiques (JOBIM '06)*, pp. 199–210, Bordeaux, France, July 2006.

[10] Y. Sakakibara, "Grammatical inference in bioinformatics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 7, pp. 1051–1062, 2005.

[11] D. B. Searls, "The language of genes," *Nature*, vol. 420, no. 6912, pp. 211–217, 2002.

[12] E. Alpaydin, *Introduction to Machine Learning*, MIT Press, Cambridge, Mass, USA, 2nd edition, 2010.

[13] L. Polkowski and A. Skowron, *Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems*, Physica, 1998.

[14] C. P. Jaroniec, C. E. MacPhee, V. S. Bajaj, M. T. McMahon, C. M. Dobson, and R. G. Griffin, "High-resolution molecular structure of a peptide in an amyloid fibril determined by magic angle spinning NMR spectroscopy," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 3, pp. 711–716, 2004.

[15] V. N. Uversky and A. L. Fink, "Conformational constraints for amyloid fibrillation: the importance of being unfolded," *Biochimica et Biophysica Acta (BBA)—Proteins and Proteomics*, vol. 1698, no. 2, pp. 131–153, 2004.

[16] M. J. Thompson, S. A. Sievers, J. Karanicolas, M. I. Ivanova, D. Baker, and D. Eisenberg, "The 3D profile method for identifying fibril-forming segments of proteins," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 103, no. 11, pp. 4074–4078, 2006.

[17] S. J. Hamodrakas, "Protein aggregation and amyloid fibril formation prediction software from primary sequence: towards controlling the formation of bacterial inclusion bodies," *The FEBS Journal*, vol. 278, no. 14, pp. 2428–2435, 2011.

[18] J. Stanislawski, M. Kotulska, and O. Unold, "Machine learning methods can replace 3D profile method in classification of

amyloidogenic hexapeptides," *BMC Bioinformatics*, vol. 14, no. 1, article 21, 2013.

[19] O. Unold, "Fuzzy grammar-based prediction of amyloidogenic regions," *JMLR: Workshop and Conference Proceedings*, vol. 21, pp. 210–219, 2012.

[20] O. Unold, "How to support prediction of amyloidogenic regions—the use of a GA-based wrapper feature selections," in *Proceedings of the 2nd International Conference on Advances in Information Mining and Management (IMMM '12)*, Venice, Italy, October 2012.

[21] B. Liu, W. Zhang, L. Jia, J. Wang, X. Zhao, and M. Yin, "Prediction of 'aggregation-prone' peptides with hybrid classification approach," *Mathematical Problems in Engineering*, vol. 2015, Article ID 857325, 9 pages, 2015.

[22] K. J. Lang, "Random DFA's can be approximately learned from sparse uniform examples," in *Proceedings of the 5th Annual Workshop on Computational Learning Theory (COLT '92)*, pp. 45–52, ACM, Pittsburgh, Pa, USA, July 1992.

[23] K. J. Lang, B. A. Pearlmutter, and R. A. Price, "Results of the abbadingo one DFA learning competition and a new evidence-driven state merging algorithm," in *Proceedings of the 4th International Colloquium on Grammatical Inference, (ICGI '98) Ames, Iowa, USA, July 1998*, pp. 1–12, Springer, 1998.

[24] K. J. Lang, "Merge Order count," Tech. Rep., NECI, Montpelier, Vt, USA, 1997.

[25] Z. Solan, D. Horn, E. Ruppin, and S. Edelman, "Unsupervised learning of natural languages," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 33, pp. 11629–11634, 2005.

[26] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[27] J. Beerten, J. Van Durme, R. Gallardo et al., "WALTZDB: a benchmark database of amyloidogenic hexapeptides," *Bioinformatics*, vol. 31, no. 10, pp. 1698–1700, 2015.

[28] E. Tomita, A. Tanaka, and H. Takahashi, "The worst-case time complexity for generating all maximal cliques and computational experiments," *Theoretical Computer Science*, vol. 363, no. 1, pp. 28–42, 2006.

[29] B. Trakhtenbrot and Y. Barzdin, *Finite Automata: Behavior and Synthesis*, North-Holland Publishing, 1973.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort et al., "Scikit-learn: machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[31] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Computation*, vol. 10, no. 7, pp. 1895–1923, 1998.

[32] S. Kotsiantis and P. Pintelas, "Combining bagging and boosting," *International Journal of Computational Intelligence*, vol. 1, no. 4, pp. 324–333, 2004.

[33] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI '95)*, vol. 2, pp. 1137–1143, Montreal, Canada, August 1995.

[34] U. M. Braga-Neto and E. R. Dougherty, "Is cross-validation valid for small-sample microarray classification?" *Bioinformatics*, vol. 20, no. 3, pp. 374–380, 2004.

[35] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, and R. Tibshirani, *The Elements of Statistical Learning*, vol. 2, no. 1, Springer, Berlin, Germany, 2009.

[36] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection," *Statistics Surveys*, vol. 4, pp. 40–79, 2010.

[37] D. Krstajic, L. J. Buturovic, D. E. Leahy, and S. Thomas, "Cross-validation pitfalls when selecting and assessing regression and classification models," *Journal of Cheminformatics*, vol. 6, no. 1, article 10, 2014.

[38] C. Nadeau and Y. Bengio, "Inference for the generalization error," *Machine Learning*, vol. 52, no. 3, pp. 239–281, 2003.

[39] R. R. Bouckaert and E. Frank, "Evaluating the replicability of significance tests for comparing learning algorithms," in *Advances in Knowledge Discovery and Data Mining*, H. Dai, R. Srikant, and C. Zhang, Eds., vol. 3056 of *Lecture Notes in Computer Science*, pp. 3–12, Springer, 2004.

[40] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *The Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[41] N. Japkowicz and M. Shah, *Evaluating Learning Algorithms: A Classification Perspective*, Cambridge University Press, Cambridge, UK, 2011.

[42] J. P. Romano, A. M. Shaikh, and M. Wolf, "Control of the false discovery rate under dependence using the bootstrap and subsampling," *TEST*, vol. 17, no. 3, pp. 417–442, 2008.

[43] S. Holm, "A simple sequentially rejective multiple test procedure," *Scandinavian Journal of Statistics*, vol. 6, no. 2, pp. 65–70, 1979.

[44] B. W. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *Biochimica et Biophysica Acta (BBA)—Protein Structure*, vol. 405, no. 2, pp. 442–451, 1975.

[45] M. Emily, A. Talvas, and C. Delamarche, "MetAmyl: a METa-predictor for AMYLoid proteins," *PLoS ONE*, vol. 8, no. 11, Article ID e79722, 2013.