BIOINFORMATICS ORIGINAL PAPER

Vol. 30 no. 14 2014, pages 1974–1982
doi:10.1093/bioinformatics/btu165

Sequence analysis

Advance Access publication March 28, 2014

Motifs tree: a new method for predicting post-translational modifications

Christophe Charpilloz[1,2], Anne-Lise Veuthey[2], Bastien Chopard[1,2] and Jean-Luc Falcone[1,2,*]

[1]Department of Computer Science, University of Geneva, 1227 Carouge and [2]Swiss Institute of Bioinformatics, Centre Médical Universitaire, Geneva 4, Switzerland

Associate Editor: John Hancock

ABSTRACT

Motivation: Post-translational modifications (PTMs) are important steps in the maturation of proteins. Several models exist to predict specific PTMs, from manually detected patterns to machine learning methods. On one hand, the manual detection of patterns does not provide the most efficient classifiers and requires an important workload, and on the other hand, models built by machine learning methods are hard to interpret and do not increase biological knowledge. Therefore, we developed a novel method based on patterns discovery and decision trees to predict PTMs. The proposed algorithm builds a decision tree, by coupling the *C4.5* algorithm with genetic algorithms, producing high-performance *white box* classifiers. Our method was tested on the initiator methionine cleavage (IMC) and N^{α}-terminal acetylation (N-Ac), two of the most common PTMs.

Results: The resulting classifiers perform well when compared with existing models. On a set of eukaryotic proteins, they display a cross-validated Matthews correlation coefficient of 0.83 (IMC) and 0.65 (N-Ac). When used to predict potential substrates of N-terminal acetyltransferaseB and N-terminal acetyltransferaseC, our classifiers display better performance than the state of the art. Moreover, we present an analysis of the model predicting IMC for *Homo sapiens* proteins and demonstrate that we are able to extract experimentally known facts without prior knowledge. Those results validate the fact that our method produces *white box* models.

Availability and implementation: Predictors for IMC and N-Ac and all datasets are freely available at http://terminus.unige.ch/.

Contact: jean-luc.falcone@unige.ch

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on June 13, 2013; revised on March 12, 2014; accepted on March 24, 2014

1 INTRODUCTION

Post-translational modifications (PTMs) are modifications occurring during protein maturation or biosynthesis. These modifications can consist of attachments of functional groups (e.g. methylation), changes of the chemical nature (e.g. deamidation), cleavage of one or more residues (e.g. initiator methionine cleavage) or structural changes (e.g. disulfide bonds). The PTMs broaden the diversity of functional groups of the 20 standard amino acids, thus producing diverse forms of proteins that cannot be derived only from its genes (Schwartz *et al.*, 2009; Walsh, 2006). Because the mature form of a protein cannot be inferred only by genes, the knowledge of a protein's PTMs helps to understand the roles, the possible interactions or the activity of a protein.

Numerous predictors for PTMs have been developed, based on different machine learning models. For example, artificial neural networks have been widely used to predict various PTMs, like phosphorylation (Blom *et al.*, 1999), N-terminal myristoylation (Bologna *et al.*, 2004) and C-mannosylation (Julenius, 2007). More recently, Random Forest method has been successfully used to predict PTM sites, for ubiquitination (Radivojac *et al.*, 2010), γ-carboxylation (Zhang *et al.*, 2012) and glycosylation sites (Chuang *et al.*, 2012). Although some of these predictors provide good prediction capabilities for the problem they tackle, they often are *black boxes*. Their mathematical complexity makes them hard to interpret in terms of biological meaning (Berthold *et al.*, 2010). Unfortunately, this restricts the application of these models for biological problems, which, in our opinion, require a model providing explanations for the prediction.

The purpose of this article is to introduce a new method to automatically build a PTM predictor, using only the information contained in the proteins primary structure and which can be interpreted by biologists. We focused on two PTMs: first the N^{α}-terminal acetylation (N^{α}-Ac), a PTM involving the transfer of an acetyl group to the N-terminal residue α-amino group. It is one of the most common covalent irreversible modifications and occurs in ~50% of yeast proteins and ~80% of human proteins (Polevoda and Sherman, 2002, 2003). In eukaryotes, N^{α}-Ac is catalyzed by N-terminal acetyltransferases (Nats) (Gautschi *et al.*, 2003; Pestana and Pitot, 1975; Polevoda *et al.*, 2008). Six Nats have been identified (NatA–NatF), each acetylating specific N-terminal substrates (Polevoda and Sherman, 2003; Polevoda *et al.*, 2009).

Because N^{α}-Ac can occur on proteins having their initiator methionine cleaved or not (Polevoda *et al.*, 2009), it is also required to know if the initiator methionine cleavage (IMC) occurs to produce an accurate predictor for N^{α}-Ac. Hence, the second PTM studied is the IMC, which is catalyzed by methionine aminopeptidases (MetAPs; Kendall and Bradshaw, 1992).

As pointed by Eisenhaber and Eisenhaber (2010), it is unlikely to discover a unique pattern describing the requirement of all enzymes because there is no biological sense to build an acetylation predictor based on an 'average' motif, as no enzyme recognizes this 'average' motif. Our main idea is to combine several discriminant motifs optimized with genetic algorithms (GA).

*To whom correspondence should be addressed.

These motifs are combined using a binary decision tree (DT). Our choice is mainly motivated by the need for *white box* models, that is, to say classifiers that are interpretable by biologists to help identifying the required biological features.

The method described in this article is tested by evaluating its capacity to predict the IMC and the N^{α}-Ac of eukaryotic proteins. The choice of predicting those PTMs has been made because several methods to predict N^{α}-Ac have been published, which allow us to test the efficiency of our method by comparison. The published methods range from *black box* machine learning methods (ML), e.g. support vector machines (Liu and Lin, 2004) and artificial neural networks (Lars *et al*., 2005), to manual pattern detection ('by eye'). For example, Martinez *et al*. (2008), Cai and Lu (2008) and more recently Bienvenut *et al*. (2012) predicted PTMs using manually extracted rules based only on the information provided by the first two or three amino acids in the sequence, which may be insufficient to predict correctly the PTMs.

2 METHODS

Our method is based on combinations of biomolecular motif descriptors. Each descriptor can then be used to compute a similarity score by aligning the descriptor with an amino acid sequence (Gonnet and Lisacek, 2002). These scores are then compared with cutoff values to discriminate sequences into two groups (Bucher *et al*., 1996). These descriptors are then combined in a DT, where they correspond to the test nodes. We call such model a *motifs tree*.

Dataset. Because our method relies on supervised machine learning, we need good-quality datasets to train the classifiers. All data used in this study were extracted from the release 2012_07 of UniProtKB (11 July 2012). (i) *N^{α}-Ac dataset*. To extract entries from UniProtKB, we build two queries, one for the N^{α}-acetylated proteins and one for proteins that do not undergo N^{α}-Ac. Our datasets were based only on experimental evidence. We select entries in the database according to the following criteria: each entry must have been reviewed by a UniProtKB curator; its existence must be experimentally proven and a chromosomic gene must be linked to the entry. An entry is labeled as N^{α}-acetylated (N-Ac) if the residue exposed to the Nat is annotated as N-acetyl. The acetylation must also be experimentally proven. An entry is labeled as non-N^{α}-acetylated if the exposed N-termial residue is not annotated as N-acetyl (regardless the confidence) and one reference must state that the entry has been sequenced at protein level with a method able to detect eventual acetylation. Proteins with N-terminal residues blocked by an unidentified modification are discarded. Those criteria are detailed in Supplementary Information A (see '2.1 Criteria used to build the datasets'). Although it is known that N^{α}-Ac is not always a total modification, this fact is currently not taken into account in the available protein databases. Hence, we qualify a protein as acetylated if the PTM was experimentally observed, regardless of the modification ratio. The extraction process was repeated for several taxonomic groups. Table 1 shows the sizes and the PTM ratio of the datasets extracted from UniProtKB depending on the chosen taxon: *Eukaryota*, *Metazoa* and *Homo sapiens*. We also stress that the taxon datasets are not mutually exclusive: 79% of the *Eukaryota* dataset is composed by *Metazoa* sequences and 65% of the *Metazoa* dataset is composed by *H.sapiens* sequences. (ii) *IMC dataset*. There is no specific query to build an IMC dataset. Our IMC datasets were extracted from the N^{α}-Ac datasets by checking the presence of the *feature* of type *initiator methionine* with the value *removed*. The criteria used for the N^{α}-Ac datasets imply experimental evidences for the IMC too. In the case of the IMC datasets, we have kept 33 proteins that were filtered out of the N^{α}-Ac dataset because the method used to sequence the N-terminus was not able to determine acetylation, while being able to determine the IMC status. The datasets' composition is detailed in Table 1.

Table 1. Number of sequences and content of the different datasets extracted from UniProtKB for the two considered PTMs: the IMC and the N^{α}-Ac

Taxon	Initiator Met cleavage		N^{α}-Terminal acetylation	
	Number of sequences	Ratio	Number of sequences	Ratio
Eukaryota	2519	0.72	2486	0.64
Metazoa	2004	0.72	1971	0.71
H.sapiens	1322	0.69	1289	0.87

Note: The 'Ratio' column indicates the ratio of proteins undergoing the corresponding PTMs.

Model. There are several approaches to define the motif descriptors: regular expression, consensus sequence with degenerated positions, consensus sequence with mismatches, weight matrix, flexible pattern, profile and so on (Bork and Gibson, 1996; Bucher *et al*., 1996). We will, in the context of this study, define a motif as a sequence of elements called here token. The five categories of tokens we used are presented along with their similarity measure with an amino acid. The similarity of an amino acid a with a token t is denoted by $\sigma(t,a)$, and ranges between 0 and 1.

- **Any amino acid:** this token matches with any amino acid and its similarity measure is always 1. This token is represented with the symbol '•'.
- **Fixed amino acid:** which are tokens imposing a match with a single amino acid. The similarity measure is 1 if and only if the token is aligned on the amino acid described by the token, otherwise it is 0.
- **Inclusion:** these tokens describe sets of amino acids. The similarity measure is 1 if and only if the token is aligned on an amino acid included in the set, otherwise it is 0. For instance `[ACM]` is a token having a similarity measure of 1 with Ala, Cys and Met and 0 with the other amino acids.
- **Exclusion:** these tokens are the complement of the previous one. The token similarity measure is 1 if and only if the token is aligned on an amino acid not included in the amino acids set described by the token, otherwise it is 0. For instance `¬[EPT]` has a similarity of 0 with Glu, Pro and Thr and 1 with the other amino acids.
- **Physicochemical similarity:** which are tokens describing how similar is an amino acid to a reference amino acid according to a physicochemical property (the *AAindex1* database, Kawashima and Kanehisa, 2000). Those tokens are represented by the reference amino acid r, followed by the *AAindex1* p (i.e. $t=\{r,p\}$). For example, `{S,KYTJ820101}` is a token where the amino acids with a similar hydropathy index (Kyte and Doolittle, 1982) than Ser have a high similarity score. The similarity is computed as follows:

$$\sigma(\{r,p\},a) = 1 - |\bar{p}(r) - \bar{p}(a)|$$

where $\bar{p}(x)$ is the value of the property for x, normalized between 0 and 1.

Although we restricted our choice only to these five types of tokens, to keep the model as simple as possible, these tokens generate $\sim 2 \times 10^6$ possibilities (as there exist 2^{20} possible sets of amino acids). The previous definition of tokens allows building a similarity matrix between the 20 amino acids and the tokens. Then to compute a similarity score between a

motif and a sequence, we use the Needleman–Wunsch algorithm with the similarity matrix to obtain the score of the best possible global alignment between a motif and an amino acids sequence.

These motifs are then combined in a DT manner. The need of combining motifs arises because a single motif, as described above, was not able to produce an accurate prediction for the N^{α}-Ac prediction. A DT uses motifs as nodes and class labels as leaves. A sequence 'moves' down in the tree the following way: (i) When a sequence reaches a node, it is aligned with the node's motif to get a similarity score. This score is then compared with the node threshold (cutoff values) to select the next branch to take. (ii) When a sequence reaches a leaf, it is classified as undergoing a specific PTM or not, depending on the leaf label. This representation is highly readable: each path in the tree from the root to a given leaf can be represented as a logical clause in conjunctive normal form.

Building the motifs tree. The algorithm used to build a motifs tree is similar to *C4.5* (Quinlan, 1992). This algorithm recursively adds test nodes that split the training set. In our case, the tests conducted by the nodes are based on a motif and its alignment with a sequence. To choose a motif at each node, the *C4.5* algorithm selects the *best* motif among all possible motifs, that is to say the one yielding two subsets with the best class separation. The problem is that with the symbols used to describe our motifs, there exist $\sim 10^{6 \cdot n}$ possible motifs of length n. For example, searching the best motif of length 5 means searching the best motif among 10^{30} motifs.

Because an exhaustive search for the best possible motif is not feasible, we cannot use the *C4.5* algorithm. Therefore, we relied on GA (Goldberg, 1989) to explore the motif space (i.e. the set of all possible token sequences). The idea is that we may not need the best possible motif to build the motifs tree, but a good approximation of the best motif is probably enough.

Approximating the best motif. Genetic algorithms generate a solution to an optimization problem by mimicking Darwinian evolution (reproduction, inheritance, mutation and selection) to explore the space of admissible solutions. The idea is to reproduce a *survival-of-the-fittest* model, where several solutions of the problem are generated, then modified by bio-inspired methods and the best ones are selected for the next round of evolution. In the GA terminology, an admissible solution of the problem is called an individual, its representation in the GA is called a genome and its elements, genes. The individuals of the evolution process form the population. To understand how we used the GA to approximate the best motif, we need to define what is an individual, what is the initial population, how the fitness function is computed and which genetic operators are used.

In our setup, an individual is a token sequence of variable length. The initial population is randomly created by generating n individuals with m tokens, randomly drawn from the category of token described in the model section, with m equal to the length of the sequences in the dataset (six amino acids in this study). The fitness function used is based on the normalized information gain ratio (Russell and Norvig, 2010) and the Matthews correlation coefficient (MCC) (Matthews, 1975). The best threshold is selected among all different scores evaluated in the set. To do so, we consider each score as a potential candidate for the threshold, so each of them is used sequentially as a cutoff value. As the cutoff value allows splitting the training data, the *information gain ratio* can be computed and the score maximizing this gain is chosen to be the threshold. Then we compute the MCC based on the split induced by the threshold. We used the following GA operators: (i) The k-tournament selection operator. (ii) The one point crossover, where the same break point is used in both parents to produce two new offspring of the same length (Banzhaf *et al.*, 1998). The break point is randomly chosen at each application of the operator. (iii) The one point mutation, which changes the value of one gene in the individual. Our mutation operator can add a random new token, delete a random token or substitute a token in the motif at a random position by a random new token. (iv) We define a plague operator, which is used to simplify an individual (i.e. a motif). This operator removes the tokens that do not improve the quality of the solution, and simplifies inclusion and exclusion tokens. Therefore, this operator improves the readability of a motif without altering its discriminant power. This step is important because we try to build an interpretable model. All operators used along with model parameters are detailed in Supplementary Information A ('1 Genetic algorithm').

The retained parameters are *Tournament size*: 5; *Population size*: 250; *Max generations*: 150; *Mutation probability*: 0.75; *Number of plague "remove"*: 20; *Number of plague "clean"*: 100; *Number of amino acids*: 6; *Gap penalty*: −0.0625; *Pruning factor* (α): 0.5 and *Bucket size*: 6. They were chosen by scanning a wide range of values (data not shown). We observed that the method performance is independent of the chosen values for most of the parameters. For example, reducing the maximum generations does not change the quality of prediction but produces deeper trees. The only parameter that affects the quality of prediction is the size of the fragment used for the alignment (the 'Number of amino acids' parameter in Supplementary Information A and Table A1). This parameter specifies the number of amino acids taken into account for the alignment. During tests, we noticed that fragments that are too long, e.g. 15 amino acids, produce classifiers with poor generalization capacities (see Section 3.1). Therefore we used the minimum size that disambiguates the proteins undergoing and not undergoing N^{α}-Ac.

Software to build a motif tree. All software used to build our model, namely a motif tree, has been developed by the authors.

3 RESULTS

In this section, we first assess the learning capability of our method by evaluating the quality of the predictions obtained with the datasets extracted from UniProtKB. We then compare our predictors with the state of the art used to predict IMC and N^{α}-Ac.

3.1 Generalization and stability

Two potential problems arise from the algorithm we used to build our classifier. The first (common to all machine learning algorithms) is a lack of generalization, which is the ability of the algorithm to correctly classify proteins that are not present in the training set. The second problem is the stability of our model, that is to say the consistency of the results despite the stochastic nature the GA. We have no guarantee that every GA evolution will converge to a good solution.

Cross-validation (CV) is a widely used process to evaluate generalization of a classifier, allowing us to estimate the average generalization error of a ML method (Hastie *et al.*, 2001). To evaluate the stability, we simply applied 10 independent stratified 10-fold CVs on our datasets, combining the CV results to obtain the average and the standard deviation. So, if the CV results have a high average classification score with low standard deviations, the method is stable and produces classifiers with good generalization capability. Because we are only taking into account the first six amino acids, redundancies and ambiguities appear in the dataset. Therefore, we have removed those duplicates from the training set of each fold. This ensures that the test set contains only sequences never seen during the learning phase. More details are provided in Supplementary Information A ('2.3 Redundancy and ambiguities resolution'). The CV results are presented in Table 2.

Table 2. Results assessing the quality of the IMC prediction and N^{α}-Ac prediction

PTM	Taxon	Accuracy	Sensitivity	Specificity	MCC
IMC	*Eukaryota*	0.93	0.95	0.89	0.83 (0.0001)
	Metazoa	0.94	0.96	0.91	0.86 (0.0002)
	H.sapiens	0.95	0.96	0.93	0.89 (0.0001)
N-Ac.	*Eukaryota*	0.84	0.89	0.76	0.65 (0.0001)
	Metazoa	0.85	0.90	0.73	0.64 (0.0002)
	H.sapiens	0.93	0.96	0.59	0.56 (0.0006)

Note: Score values are the mean on 10 independent stratified CVs, each made with 10-folds. The MCC standard deviation is given in parentheses.

We can see that the learning and generalization capabilities of our method are good for both IMC and N^{α}-Ac, as it is shown by the classification scores. However, we must pay attention to the accuracy values. Because our training set classes are imbalanced, a trivial classifier could easily reach a high accuracy. For instance, 87% of human proteins are acetylated in our dataset, and a bad classifier that predicts all proteins as acetylated will obtain an accuracy score of 0.87. To evaluate our results, we then compare the obtained accuracy score against a so-called baseline (the 'Ratio' columns in Table 1), which is the proportion of the majority class in the training set. All classifiers obtained here display a significant improvement over the baselines of our training sets. For example, with the *Eukaryota* dataset, the accuracy rises from 0.72 (baseline) to 0.93 in the case of IMC. In the case of N^{α}-Ac, it rises from 0.64 (baseline) to 0.84. The results also show that the method is stable because the standard deviation is <1% of the classification scores, meaning that every run will produce a good classifier.

3.2 Comparison with *TermiNator3*

Now that we know that our method is reliable, we compare our method with the state of the art: *TermiNator3* (Martinez *et al.*, 2008). We choose not to compare our model against *NetAcet*, another well-known N^{α}-Ac predictor, because it has only been trained on NatA substrates from *Saccharomyces cerevisiae*. For this comparison, we trained our classifiers with the full datasets (instead of running a CV experiment) described above, one for each taxon: *Eukaryota*, *Metazoa* and *H.sapiens* and for each PTM: N^{α}-Ac and IMC. Six predictors were produced, whose performances were compared with *TermiNator3*. Those trainings are justified by the fact that, now that we are convinced that our model generalizes well and is stable, we wanted to use all the available information to build the most accurate predictors. Moreover, the patterns used by *TermiNator3* seem to have been built based on their full dataset. The comparison is presented in Table 3 and we obtained cross-validated results close to *TermiNator3* with our method (Table 2). When trained on the full dataset, results are on par with *TermiNator3* for the prediction of IMC. However, our classifiers perform better than *TermiNator3* for N^{α}-Ac prediction.

3.2.1 Potential NatB and NatC As it was introduced above, several enzymes catalyze the N^{α}-Ac; however, the information regarding the Nats catalyzing the PTM is rarely available. But by looking at the known specific substrates, authors have proposed substrates to identify the Nat catalyzing the acetylation depending on the first two amino acids. For the NatB, the following substrates are proposed: MD-, ME-, MN-; for the NatC, the following substrates are proposed: MF-, MI-, ML-, MW- (Polevoda *et al.*, 2009).

Unfortunately, the number of experimentally identified substrates of those specific Nats is scarce. To estimate the capability of our classifiers regarding NatB and NatC substrates, we built two new datasets: one for *potential* NatB and one for *potential* NatC. From the *Eukaryota* dataset, all proteins matching the theoretical requirements for NatB or C are considered as potential substrates and extracted into those new datasets. The proteins are extracted with their original class (i.e. N^{α}-acetylated or not N^{α}-acetylated) because not all proteins matching the substrates are acetylated.

We applied a 10-fold CV on the whole *Eukaryota* dataset. Then, we measured the performance of the model only on the potential NatB and NatC. The results in Table 4 display that the patterns used in *TermiNator3* are too stringent. The results show that if a sequence starts like the NatB-proposed substrate, it is always classified as N^{α}-acetylated (the sensitivity is 1.0 and the specificity is 0.0). For the sequences starting like the NatC-proposed substrates, it is the opposite (the sensitivity of 0.0 and the specificity of 1.0), indicating that all these sequences are classified as not N^{α}-acetylated. Therefore, in both cases the MCC obtained is 0.0, meaning that in this case *TermiNator3* performs no better than random prediction. The pattern used by *TermiNator3* (Martinez *et al.*, 2008) takes only into account, at most, the first three amino acids, but the information provided by these three amino acids is probably insufficient to decide whether a protein undergoes N^{α}-Ac. Our model takes into account the first six amino acids and produces a cross-validated MCC >0.0; therefore, it performs better than random. So, our model has been able to find specificities between proteins undergoing N^{α}-Ac, as it is showed by the increase of specificity in the case of the NatB substrates (+0.39) and the increase of sensitivity in the case of the NatC substrates (+0.55).

Finally, we tested our predictor on the five experimentally identified substrates of NatB and C in *H.sapiens* (Starheim *et al.*, 2008, 2009). As shown in Table 5, all substrates were correctly predicted by the motifs tree. This shows that our model is able to discover subtle features specific to those proteins, even when they are accounting only for <20% of the whole dataset.

3.3 Analysis of the initiator Met cleavage motifs tree

The main goal of this article is to present a new automatic approach to predict PTMs based only on the protein primary structure, called *motifs tree*, and we have presented the performances of our classifiers to predict N^{α}-Ac and IMC. In this section, we will show how we can use our model to infer knowledge about the underlying biological process (e.g. enzyme–substrate specificity). Owing to the lack of space, we will illustrate this feature by analyzing the smallest motifs tree, predicting IMC in *H.sapiens*. However, the same approach can be applied to all motifs trees

Table 3. Prediction scores for Terminus (our online predictor) and TermiNator3

Taxon	Service	Initiator Met cleavage				N^{α}-Terminal acetylation			
		Accuracy	Sensitivity	Specificity	MCC	Accuracy	Sensitivity	Specificity	MCC
Eukaryota	*Terminus*	0.99	0.99	0.99	0.98	0.99	0.98	0.99	0.97
	TermiNator3	0.96	0.99	0.89	0.91	0.87	0.92	0.77	0.71
Metazoa	*Terminus*	0.99	0.99	0.99	0.98	0.99	1.0	0.96	0.96
	TermiNator3	0.97	1.00	0.90	0.92	0.88	0.92	0.80	0.72
H.sapiens	*Terminus*	0.99	0.99	0.99	0.97	0.99	0.99	0.96	0.96
	TermiNator3	0.97	0.99	0.92	0.93	0.90	0.91	0.82	0.63

Table 4. Cross-validated scores obtained by *Eukaryota* classifiers versus TermiNator3

Service	Potential substrate	Number of sequences	N-Acet.	Accuracy	Sensitivity	Specificity	MCC
Motifs tree	NatB	384	0.91	0.89	0.93	0.39	0.31
TermiNator3				0.90	1.00	0.00	0.00
Motifs tree	NatC	100	0.38	0.66	0.55	0.73	0.28
TermiNator3				0.64	0.00	1.00	0.00

Note: The Potential substrate means that only sequences matching to the potential NatB or potential NatC substrates are considered. The 'N-Acet.' column indicates the ratio of sequences undergoing N^{α}-Ac in each dataset.

Table 5. Predictions of acetylated proteins with known Nats using the Terminus *H.sapiens* classifier

UniProt ID	Taxon	Sequence	Nat	Terminus	TermiNator3
Q04206	*H.sapiens*	MDELFPL	B	Ac-M(1)	Ac-M(1)
Q9NVJ2[a]	*H.sapiens*	MLALISR	C	Ac-M(1)	M(1)
P42345	*H.sapiens*	MLGTGPA	C	Ac-M(1)	M(1)
P31943[a]	*H.sapiens*	MLGTEGG	C	Ac-M(1)	M(1)
P52597[a]	*H.sapiens*	MLGPEGG	C	Ac-M(1)	M(1)

Note: The 'Sequence' column displays only the first seven amino acids of the protein exposed to the Nat. The 'Nat' column indicates which Nat catalyzes the N^{α}-Ac. The 'Terminus' and the 'TermiNator3' columns indicate, respectively, the prediction of the services. [a]Sequences used to build the classifiers.

(article in preparation). For details about the other motifs trees produced for this article, see Supplementary Information B ('1 Motifs trees').

The analyzed tree is the product of a training on the full *H.sapiens* dataset. We point out that the motifs found during different runs of training are close and combined in similar trees. As it seems that all learning phases converge to a particular point in the solution space, we can focus our analysis on one motifs tree.

As this model is based on combination of motifs, it can be interesting to analyze the discovered motifs to propose assumptions about the substrates of the enzymes catalyzing the chemical process of a given PTM. Hence, we studied how sequences are split at each node and we tried to extract the features that separate the two sets of sequences induced by the split. Let us note that there are two genes encoding for MetAPs in human, MetAP1 and MetAP2 (Bradshaw *et al*., 1998), but the information about which enzyme catalyzes the cleavage is not known and is not taken into consideration in the model. Also, even if it does not add information, the initiator Met is kept in the sequences.

First of all, we see that the motifs tree (Fig. 1) is composed of three tests (motifs), all of them leading to at least one leaf (i.e. a predicted class):

- The sequences that do not contain the signal described by the first motif are classified as not undergoing the IMC;

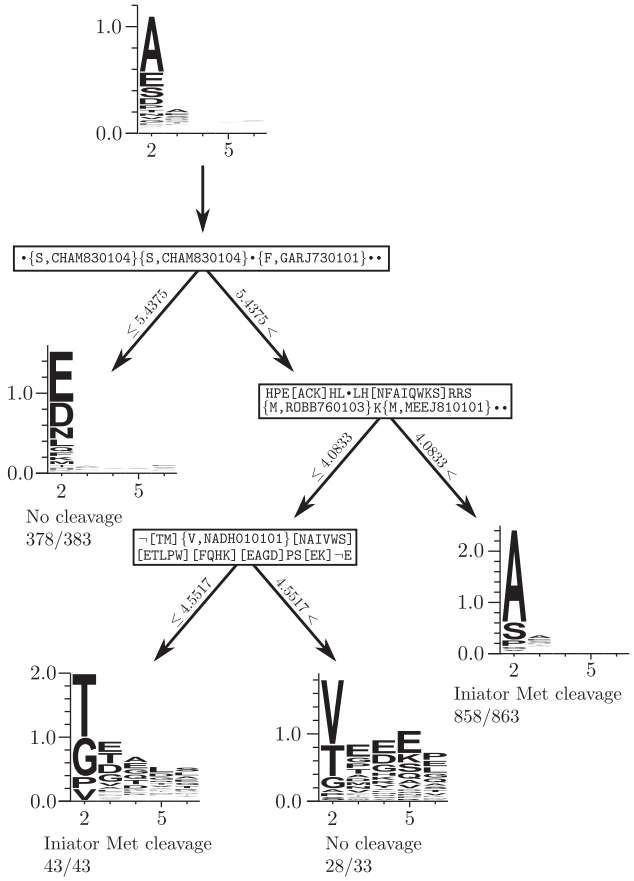

Fig. 1. The motifs tree for the prediction of IMC for *H. sapien* proteins extracted from UniProtKB. Each node of the tree is represented by the motif used for its test. Leaves are represented using a sequence logo made with all the sequences ending in that leaf, and the label under a leaf specifies the class corresponding to the prediction made at the leaf and its accuracy. The initiator Met is always present in all sequences, but is not displayed in the logo because it does not provide any information. Moreover, each sequence logo is rescaled according to its highest value (the maximum being 4.32 bits). The branches are labeled with the alignment score condition required on the test to follow the path indicated by the branch. The sequence logo on top illustrates the composition of the *H. sapien* proteins extracted from UniProtKB

- The sequences containing both the first two motif signals are classified as undergoing the IMC;
- The sequences reaching the last node are classified as not undergoing the IMC if the signal of the third motif is detected in the sequence.

To understand what features are exploited by the motifs tree to discriminate the sequences, we will focus on the first node. The first motif is described by the following token sequence:

```
.{S,CHAM830104}{S,CHAM830104}.
{F,GARJ730101}..
```

Our analysis is split into two steps: (i) scores analysis and identification of discriminant tokens in the motif; and (ii) positions of interest in the amino acid sequences.

We begin by identifying the discriminant tokens (step 1). To do so, we compute the average *motif score profile*. The profile is computed for a set of sequences aligned on a motif. For a given alignment, each token contributes to the alignment score either by its similarity with the aligned amino acid or by being gapped. If all contributions of each token on each sequence are summed and normalized, we obtain an average motif score profile. Formally, let $m = (t_1, t_2, \ldots, t_k)$, a k token motif, and $S = \{s_j\}$, a set of amino acids sequences, $s_j = (a_{j1}, a_{j2}, \ldots, a_{jn})$. The profile of m on all sequences in S is a vector $c = (c_1, c_2, \ldots, c_k)$, whose c_i are given by

$$c_i = \frac{1}{|S|} \sum_{j=1}^{|S|} \sigma(t_i, x_{ji}) \tag{1}$$

where x_j is the aligned sequence, i.e. s_j with the alignment gaps. So x_{ji} is the i-th symbols in the sequence j, which is aligned with m. It can be either an amino acid or a gap (σ with a gap always equals the gap penalty, i.e. -0.0625). So, to identify discriminant tokens in the motif, we compute the profiles for the sequences following the left (c_l) and the right (c_r) branch and plot the following difference: $c_r - c_l$. A positive difference points to a token increasing the score of the sequences following the right branch; a negative difference points to a token increasing the score of the sequences following the left branch. So, as we want to identify the features contributing to the signal strength, we are interested in the positive differences. In the case of the first motif, the profile difference emphasizes the discriminant power of the tokens at position 2 and 3 in the motif (Fig. 2a). The two tokens are the same, namely the token `{S,CHAM830104}`. This property is interesting because it gives the maximum similarity (i.e. 1.0) with the Ser and the following amino acids: A, C, G, P, T and obviously S. The property gives a similarity of 0.5 with the Ser for the amino acids D, E, F, I, K, M, N, Q, R, W, Y and V and has no similarity with L (i.e. 0.0). So, it clearly promotes the presence of A, C, T G, P, S and T. Regarding the amino acids producing a similarity of 0.5, it is interesting to note that the threshold is 5.4375, which is the maximum alignment score possible with the motif minus 0.5. So, the use of this property in the first motif seems to play the role of a selector for the amino acids having a similarity score of 1.

Now that we have identified two tokens having an impact on the alignment score, we must identify where, in the protein sequence, the specificity induced by the token is discriminant (step 2). To do so, we rely on a plot showing how many times a token i is aligned with the residue at position j of the sequences following the right branch. This histogram shows that the two tokens of interest are mainly aligned on the second amino acid (the one immediately after the initiator Met) and, in less extent, on the third amino acid (Fig. 2b).

This rough analysis allows us to conclude that this node splits the protein set based on the presence of an Ala, Cys, Gly, Pro and Ser immediately after the initiator Met. Moreover, as this node leads to a leaf for the sequences in which the signal is not detected, we can observe that the proteins not having those amino acids at the second position do not undergo the IMC. Therefore, the following rule can be proposed: if a sequence starts with `M¬[ACGPSTV]`, the Met is not cleaved. This has

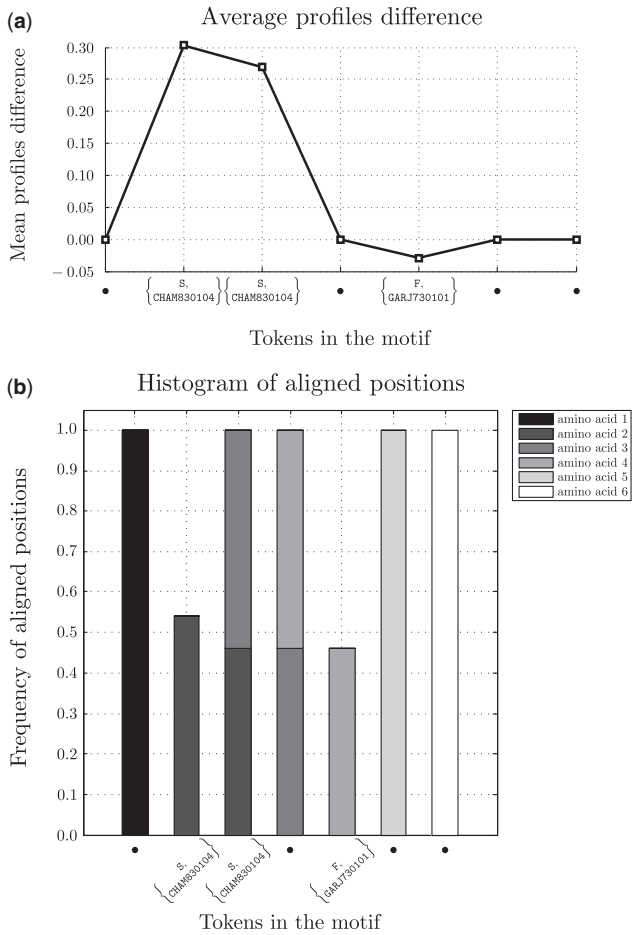

Fig. 2. (**a**) The motif score profile [Equation (1)] difference between the sequences achieving an alignment score less than or equal to the threshold and the sequences achieving an alignment score greater than the threshold. On this plot, we can see that the tokens at position 2 and 3 in the motif have an important contribution in the alignment scores of sequences achieving a score higher than the threshold. (**b**) The normalized histogram of aligned position illustrates on which positions in the amino acids sequence a token is aligned. The sequences considered to build this histogram are the one following the right branch after the first motif. The colors of the stack indicate the position in the amino acids sequences. A stack lower than 1.0 reflects that the token is aligned with sequence gaps. For example, a stack with a height of 0.4 means the token is aligned with an amino acid for 40% of the alignments and is gapped for the remaining 60%

been experimentally observed (Burstein and Schechter, 1978; Meinnel *et al*., 2005) and is corroborated by our model. Moreover, this rule is compatible with the pattern in Martinez *et al*. (2008). If we take into account only the information regarding the IMC in the cited publication, we can build the following rule: a match with `M¬[ACGPST]` for the first two amino acids imply no IMC.

The same approach can be used to extract information from the other motifs. We will summarize the main lines here. The motifs, the motif score profiles and histograms of the aligned positions, allowing us to extract these results are provided in Supplementary Information B ('1.1 Motifs tree for init. Met. cleavage prediction in *H.sapiens*'). First, it is important to remember that we are going through a decision tree, and the alignments are applied on sequences that have been *selected* by the preceding motifs. The profile difference of the second motif (Supplementary Information B, 1.1.2) indicates that the token at position 10 has a major contribution in producing discriminant alignment score between proteins. This token is `[AFIKNQSW]` and the histogram of aligned positions shows that it is almost always aligned with the second amino acid in the sequence. But we already know that the sequences reaching this node should carry `[ACGPSTV]` as the second residue. So, we can *denoise* this token by only considering the intersection between `[AFIKNQSW]` and `[ACGPST]`, leading to a simplified form of the token: `[AS]`. The motif seems to detect the presence of an Ala and a Ser in the second position. Another token contributes well to the profile difference, the token 13, which is a fixed amino acid token for the Ser. This token is mainly aligned on the second and third amino acid in the sequences. As a relevant match implies that the sequence undergoes the IMC, this leads us to propose that sequences starting with `M[AS]` are cleaved. But the `MA` sequences are highly represented in the set of sequences having a relevant match with the motif (68% of the set) and may hide the contribution of other tokens. So, we removed those sequences from the protein set and produced a new profile difference. These new profiles emphasize the contribution of the second token in the motif, which is a fixed amino acid token for the Pro and is always aligned on the second amino acid in the sequence. So, considering the preceding motif and the information provided by the tokens at position 10 and 13, we can conclude that proteins starting with `M[APS]` undergo IMC.

Therefore, proteins reaching the last motif should be mainly composed of sequences starting with `M[CGTV]`. Again the profile difference (Supplementary Information B, 1.1.3) indicates that the token 9, `[EK]`, has the greatest difference in the profiles. The histogram shows that it is always aligned on the fifth residue. This is an interesting feature because it shows that the MetAPs activity is not only influenced by the amino acid on the second and third position in the sequence but also by amino acids farther in the sequence. In this case, it is also interesting to note that these two amino acids are charged. We conclude that a protein starting by `M[CGVT]` does not undergo IMC if a Glu or Lys is present in the sequence at position 5.

We have extracted simplified rules for each motif. These rules can be combined to produce the sequence requirement for the IMC to occur or not:

- A match with `M[ACGPS]` implies that IMC occurs;
- A match with `M[TV]..¬[EK]` implies that IMC occurs;
- Otherwise the protein does not undergo IMC.

To conclude this short analysis, we wonder whether the simplified rules perform as well as the motifs tree. The rules produce a MCC of 0.93, whereas the full motifs tree produces a MCC of 0.97. This is a good result; we have been able to use the model to infer good rules that allowed us to find the sequence requirement for IMC in *H.sapiens*. We also showed that the motifs tree is sensitive to subtle features that are hardly detectable by human, as the motifs tree produces better classification scores than the inferred rules. Our model, and the tools proposed to analyze it,

lead us to draw conclusion on human MetAPs substrates that are similar to the experimental results published in literature (Burstein and Schechter, 1978; Frottin *et al*., 2006; Martinez *et al*., 2008; Meinnel *et al*., 2005; Xiao *et al*., 2010). Therefore, we can claim that our model is a *white box*.

3.4 N-terminus prediction service

We developed a free and open online service to allow researchers to use our motifs trees for predicting IMC and N^{α}-Ac on their sequences of interest. The service is accessible both through a web interface and through a simple REST API (supported in almost every programming language) and can be used to access the predictors programmatically. The service is available at the following address: http://terminus.unige.ch/.

4 DISCUSSION

We presented a new method to predict PTMs called motifs tree. The method was tested for the IMC and N^{α}-Ac by building a classifier for proteins in different taxa. The resulting models are accurate on our datasets and perform as well as the previously published state-of-the-art results, namely *TermiNator3*. Moreover, our results are cross-validated, showing that our model can build classifiers with good generalization capabilities. We did not compare our model with *NetAcet* because it has been trained only on a small dataset restricted to NatA substrates from *S.cerevisiae*.

Also, we have shown that our N^{α}-Ac classifier can take into account subtle information allowing it to improve the classification of potential NatB and NatC substrates, which is a feature that is lacking in *TermiNator3* and *NetAcet*.

As with all machine learning approaches, the quality of the predictor depends on the quality of the dataset. In biology, negative sets are difficult to build because they rely on the non-observation of a phenomenon, which is not directly annotated in databases. To confirm that our predictor was not biased because of noise in the dataset, we have used a *hold-out* test set. This set is only composed of experimentally confirmed non-acetylated eukaryotic proteins (Bienvenut *et al*., 2012). All proteins in the *hold-out* test set were not seen during training. The *Eukaryota* motifs tree produce a specificity of 0.85 on this *hold-out* test set, which is above the cross-validated specificity (+0.09). This good result illustrates that our algorithm induces correct rules to predict non-acetlyation, probably because our methodology can cope with noise in the dataset, or because our dataset is clean enough to produce accurate predictors. We add that the ability to learn with noise is a desirable feature for a ML method. For more details, see Supplementary Information A ('3.3 Validation for N-terminal acetylation classifiers').

Also, our method produces a *white box* model that shows how features are used to classify sequences. In a preliminary analysis, we have illustrated that our model can provide helpful information about the composition of sequences that promote or inhibit a PTM. This is also a valuable advantage versus the predictors presented in Lars *et al*. (2005) and Liu and Lin (2004), which is hard or impossible to interpret. We are convinced that a model used for classification in biology should be readable by experts. Models used in machine learning are able to capture characteristics that are hard to see in the data. Those characteristics or features may be exploited to understand the studied biological process.

The purpose of this first analysis was only to validate the *white box* quality of our model, by retrieving experimentally known biological facts from our motifs trees. However, in the future we may be able to use the motifs tree as a tool to propose new biological hypothesis that could be tested experimentally.

ACKNOWLEDGEMENT

The authors would like to thank Alexandros Kalousis from the Computer Science department (University of Geneva) for the discussion that helped us to improve our methodology.

Funding: The SIB activities are supported by the State Secretariat for Education, Research and Innovation (SERI).

Conflict of interest: none declared.

REFERENCES

Banzhaf,W. *et al.* (1998) *Genetic Programming: An Introduction: on the Automatic Evolution of Computer Programs and its Applications.* Morgan Kaufmann Publishers Inc., San Francisco, CA.

Berthold,M.R. *et al.* (2010) *Guide to Intelligent Data Analysis: How to Intelligently Make Sense of Real Data,* 1st edn. Springer Publishing Company, Incorporated, London.

Bienvenut,W.V. *et al.* (2012) Comparative large scale characterization of plant versus mammal proteins reveals similar and idiosyncratic N-α-acetylation features. *Mol. Cell. Proteomics*, **11,** M111.015131.

Blom,N. *et al.* (1999) Sequence and structure-based prediction of eukaryotic protein phosphorylation sites. *J. Mol. Biol.*, **294**, 1351–1362.

Bologna,G. *et al.* (2004) N-terminal myristoylation predictions by ensembles of neural networks. *Proteomics*, **4**, 1626–1632.

Bork,P. and Gibson,T.J. (1996) Applying motif and profile searches. *Methods Enzymol.*, **266**, 162–184.

Bradshaw,R.A. *et al.* (1998) N-terminal processing: the methionine aminopeptidase and N-α-acetyl transferase families. *Trends Biochem. Sci.*, **23**, 263–267.

Bucher,P. *et al.* (1996) A flexible motif search technique based on generalized profiles. *Comput. Chem.*, **20**, 3–23.

Burstein,Y. and Schechter,I. (1978) Primary structures of N-terminal extra peptide segments linked to the variable and constant regions of immunoglobulin light chain precursors: implications on the organization and controlled expression of immunoglobulin genes. *Biochemistry*, **17**, 2392–2400.

Cai,Y.-D. and Lu,L. (2008) Predicting N-terminal acetylation based on feature selection method. *Biochem. Biophys. Res. Commun.*, **372**, 862–865.

Chuang,G.-Y. *et al.* (2012) Computational prediction of N-linked glycosylation incorporating structural properties and patterns. *Bioinformatics*, **28**, 2249–2255.

Eisenhaber,B. and Eisenhaber,F. (2010) Prediction of posttranslational modification of proteins from their amino acid sequence. In: Carugo,O. and Eisenhaber,F. (eds.) *Data Mining Techniques for the Life Sciences. Methods in Molecular Biology*. Vol. 609, Humana Press, New York, NY, pp. 365–384.

Frottin,F. *et al.* (2006) The proteomics of N-terminal methionine cleavage. *Mol. Cell. Proteomics*, **5**, 2336–2349.

Gautschi,M. *et al.* (2003) The yeast n-α-acetyltransferase nata is quantitatively anchored to the ribosome and interacts with nascent polypeptides. *Mol. Cell. Biol.*, **23**, 7403–7414.

Goldberg,D.E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*. 1st edn. Addison-Wesley Longman Publishing Co., Inc., Boston, MA.

Gonnet,P. and Lisacek,F. (2002) Probabilistic alignment of motifs with sequences. *Bioinformatics*, **18**, 1091–1101.

Hastie,T. *et al.* (2001) *The Elements of Statistical Learning. Springer Series in Statistics*, 2nd edn. Springer New York Inc., New York, NY.

Julenius,K. (2007) Netcglyc 1.0: prediction of mammalian c-mannosylation sites. *Glycobiology*, **17**, 868–876.

Kawashima,S. and Kanehisa,M. (2000) AAindex: amino acid index database. *Nucleic Acids Res.*, **28**, 374.

Kendall,R.L. and Bradshaw,R.A. (1992) Isolation and characterization of the methionine aminopeptidase from porcine liver responsible for the co-translational processing of proteins. *J. Biol. Chem.*, **267**, 20667–20673.

Kyte,J. and Doolittle,R.F. (1982) A simple method for displaying the hydropathic character of a protein. *J. Mol. Biol.*, **157**, 105–132.

Lars,K. *et al.* (2005) NetAcet: prediction of N-terminal acetylation sites. *Bioinformatics*, **21**, 1269–1270.

Liu,Y. and Lin,Y. (2004) A novel method for N-terminal acetylation prediction. *Genomics Proteomics Bioinform.*, **2**, 253–255.

Martinez,A. *et al.* (2008) Extent of N-terminal modifications in cytosolic proteins from eukaryotes. *Proteomics*, **8**, 2809–2831.

Matthews,B.W. (1975) Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim. Biophys. Acta*, **405**, 442–451.

Meinnel,T. *et al.* (2005) Processed N-termini of mature proteins in higher eukaryotes and their major contribution to dynamic proteomics. *Biochimie*, **87**, 701–712.

Pestana,A. and Pitot,H.C. (1975) Acetylation of nascent polypeptide chains on rat liver polyribosomes *in vivo* and *in vitro*. *Biochemistry*, **14**, 1404–1412.

Polevoda,B. and Sherman,F. (2002) The diversity of acetylated proteins. *Genome Biol.*, **3**, reviews0006.

Polevoda,B. and Sherman,F. (2003) N-terminal acetyltransferases and sequence requirements for N-terminal acetylation of eukaryotic proteins. *J. Mol. Biol.*, **325**, 595–622.

Polevoda,B. *et al.* (2008) Yeast n-α-terminal acetyltransferases are associated with ribosomes. *J. Cell. Biochem.*, **103**, 492–508.

Polevoda,B. *et al.* (2009) A synopsis of eukaryotic n-α-terminal acetyltransferases: nomenclature, subunits and substrates. *BMC Proc.*, **3** (**Suppl. 6**), S2.

Quinlan,J.R. (1992) *C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)*. 1st edn. Morgan Kaufmann, San Mateo, CA.

Radivojac,P. *et al.* (2010) Identification, analysis, and prediction of protein ubiquitination sites. *Proteins*, **78**, 365–380.

Russell,S.J. and Norvig,P. (2010) *Artificial Intelligence—A Modern Approach*. 3rd edn. Pearson Education, Upper Saddle River, NJ.

Schwartz,D. *et al.* (2009) Predicting protein post-translational modifications using meta-analysis of proteome scale data sets. *Mol. Cell. Proteomics*, **8**, 365–379.

Starheim,K.K. *et al.* (2008) Identification of the human N-α-acetyltransferase complex b (hNatB): a complex important for cell-cycle progression. *Biochem. J.*, **415**, 325–331.

Starheim,K.K. *et al.* (2009) Knockdown of human N-α-terminal acetyltransferase complex C leads to p53-dependent apoptosis and aberrant human Arl8b localization. *Mol. Cell. Biol.*, **29**, 3569–3581.

Walsh,C. (2006) *Posttranslational Modification of Proteins: Expanding Nature's Inventory*. Roberts and Company Publishers, Englewood, CO.

Xiao,Q. *et al.* (2010) Protein N-terminal processing: substrate specificity of *Escherichia coli* and human methionine aminopeptidases. *Biochemistry*, **49**, 5588–5599.

Zhang,N. *et al.* (2012) Computational prediction and analysis of protein γ-carboxylation sites based on a random forest method. *Mol. Biosyst.*, **8**, 2946–2955.