JMLR: Workshop and Conference Proceedings 21:97–111, 2012 The 11th ICGI

Local Substitutability for Sequence Generalization

François Coste FRANCOIS.COSTE@INRIA.FR
Gaëlle Garet GAELLE.GARET@INRIA.FR
Jacques Nicolas JACQUES.NICOLAS@INRIA.FR
INRIA, Centre Inria Rennes - Bretagne Atlantique, Campus universitaire de Beaulieu, Rennes, France

Editors: Jeffrey Heinz, Colin de la Higuera and Tim Oates

Abstract

Genomic banks are fed continuously by large sets of DNA or RNA sequences coming from high throughput machines. Protein annotation is a task of first importance with respect to these banks. It consists of retrieving the genes that code for proteins within the sequences and then predict the function of these new proteins in the cell by comparison with known families. Many methods have been designed to characterize protein families and discover new members, mainly based on subsets of regular expressions or simple Hidden Markov Models. We are interested in more expressive models that are able to capture the long-range characteristic interactions occurring in the spatial structure of the analyzed protein family. Starting from the work of Clark and Eyraud (2007) and Yoshinaka (2008) on inference of substitutable and k,l-substitutable languages respectively, we introduce new classes of substitutable languages using local rather than global substitutability, a reasonable assumption with respect to protein structures to enhance inductive leaps performed by least generalized generalization approaches. The concepts are illustrated on a first experiment using a real proteic sequence set.

Keywords: Local substitutable languages, context-free grammars, inference, proteins

1. Biological Motivation

Since the first entirely sequenced genome in 1995 -*Haemophilus influenzae*-, scientists strive towards a systematic investigation of chromosomal DNA sequences for other living species. Technological improvements now allow many biological laboratories to obtain new genomic sequences of good quality for an affordable price and the cumulated volume of these data has greatly increased. To date, almost 2000 species have their genome completely sequenced. Protein annotation is a task of first importance with respect to these genomes. It consists of retrieving the genes that code for proteins within the sequences and then predict the function in the cell of these new proteins by comparison of their sequences with known families. Indeed, proteins are in general highly conserved through species and this allows to delineate families and super-families of related elements and predict some functional properties with good accuracy on the basis of common shared motifs. As stated in Galperin and Koonin (2010), the annotation of protein families is far more realistic than the annotation of individual proteins since it allows to clearly focus on a particular functional aspect (individual proteins have generally multiple functional effects leading to different phenotypic traits, a phenomenon referred as 'pleiotropy'). The experimental validation of predictions remain a very demanding task and the level of precision of putative functional

assignments is crucial to limit their number. Moreover, the new generation of sequencing technologies has introduced the study of sequence variations within populations and it becomes even more important to be able to identify which of the variants are causal with respect to a particular disease state or phenotypic trait (Zhang et al. (2011)).

There exists several databases (Hunter et al. (2012)) and a broad range of methods and softwares for the discovery of characteristic protein motifs from sets of sequences. The difficulties come from the tradeoff to be established between the degree of expressivity of motifs and their learnability. Overall, two types of methods are available : probabilistic and combinatorial. The first one includes in particular various types of Position-Weight matrices and profile Hidden Markov Models (HMM) based on the estimation of a score for each amino-acid and each position in a fixed-size motif (Bailey et al. (2009); Durbin (1998)). The second one, which we have followed, considers that sequences belong to some formal language and the task is to learn an expression, an automaton, or a grammar that represents this language (Jonassen et al. (1995); Yokomori et al. (1994) and Peris et al. (2006); Coste and Kerbellec (2005)). In most cases, model's elaboration is preceded by an alignment phase of the family sequences, which is in charge of finding the best correspondence between positions in each sequence from the observation of common subsequences. These alignments can be done either by pairwise (Altschul et al. (1990)), or by multiple sequences alignments (Thompson et al. (1994)). They can act globally on the whole length of the sequences or look for local similarities. Models are deduced from these alignments, generally by computing a score for the presence of a letter at a given position in the model. The final sensitivity/specificity of a method depends thus on the flexibility of alignment procedures and on the type of inductive leap allowed by the model construction procedure. Kerbellec (2008) uses a fine-grained approach producing partial local multiple alignments (PLMA). Partial means that a subset of the whole set of sequences is considered for each alignment and local refers to a subset of positions for each subsequence (see Figure 1).

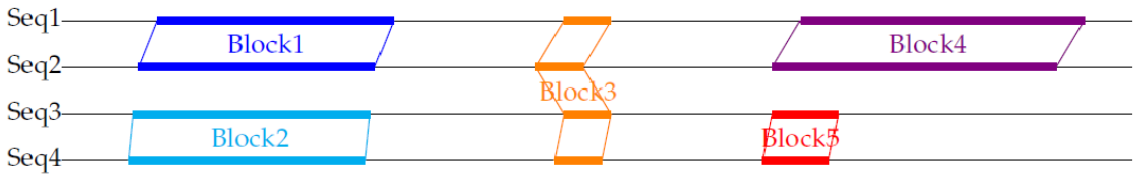


Figure 1: Sequences with PLMA blocks

We are interested in expressive models that are able to capture the long-range characteristic interactions occurring in the spatial structure of the analyzed protein family. HMM or regular language representations are insufficient to represent such dependencies unless they are used as approximations of more complex structures, unfolding them in a finite number of bounded length cases. This has important consequences on the learning phase. The induction step is generally based on the optimization of some property that is directly related to the chosen representation. Looking for a finite state automaton of minimal size with respect to a set of sequences will give poor results if the automaton is the translation of a more complex structure and has thus no reason to be of minimal size.

Using context-free grammars as a family's signature is more adequate for protein characterization but learning them with an alphabet of size 20 (the number of different amino-acids in proteins) and sequences of length 1000 (a common value for proteins) remains unfeasible. We used thus the conserved blocks identified by partial local multiple sequence alignment as a preprocessing step to build an initial canonical maximal grammar, representing the PLMA blocks and the succession of their occurrences in the training examples (Figure 2). The generalization starts from this initial grammar.

$$\begin{aligned} \mathbf{S} &\longrightarrow \text{Block1 Block3 Block4} \mid \text{Block2 Block3 Block5} \\ \text{Block1} &\longrightarrow \ldots \\ \text{Block2} &\longrightarrow \ldots \end{aligned}$$

Figure 2: Grammar based on PLMA of Figure 1

To generalize, an appealing approach was to rely on the substitutability principle and to learn substitutable and k, l-substitutable languages as initiated by Clark and Eyraud (2007) and Yoshinaka (2008).

Before defining substitutable and k, l-substitutable properties, we briefly introduce general definitions and notations. An *alphabet* Σ is a finite nonempty set of symbols called *letters*. A *string* w over Σ is a finite sequence $w = a_1 a_2 ...$ of letters. The term $|w|$ denotes the length of w and the empty string of length 0 will be indicated by λ. Let Σ^* be the set of all strings. A *grammar* is a quadruple $G =< V, \Sigma, P, S >$ where Σ is a finite alphabet of terminal symbols, V is a finite alphabet of variables or non-terminals, P is a finite set of production rules, and $S \in V$ is the start symbol. We denote by $L(G) = \{w \in \Sigma^* : S \Rightarrow_G^* w\}$ the language defined by the grammar.

Substitutable and k, l-substitutable languages are defined by:

Definition 1 (Substitutability (Clark and Eyraud (2007))) *A language L is substitutable iff for any $x_1, y_1, z_1, x_2, y_2, z_2 \in \Sigma^*$,*

$$x_1 y_1 z_1 \in L \wedge x_1 y_2 z_1 \in L \Rightarrow (x_2 y_1 z_2 \in L \Leftrightarrow x_2 y_2 z_2 \in L).$$

Definition 2 (k, l-substitutability (Yoshinaka (2008))) *A language L is k, l-substitutable iff for any $x_1, y_1, z_1, x_2, y_2, z_2 \in \Sigma^*, u \in \Sigma^k, v \in \Sigma^l$, such that $uy_1v, uy_2v \neq \lambda$,*

$$x_1 u y_1 v z_1 \in L \wedge x_1 u y_2 v z_1 \in L \Rightarrow (x_2 u y_1 v z_2 \in L \Leftrightarrow x_2 u y_2 v z_2 \in L).$$

The class of substitutable context free languages is the class of substitutable languages that are context free. k, l-substitutable context free languages are defined similarly.

As pointed out by Yoshinaka (2008), the class of substitutable context-free languages introduced in Clark and Eyraud (2007) are the analogue of zero-reversible regular languages. Like zero-reversible languages have been extended to the hierarchy of k-reversible regular languages, Yoshinaka (2008) defines the hierarchy of k, l-substitutable context-free languages, where substitutable context-free languages are the $0, 0$-substitutable context-free languages.

In preliminary experiments on learning these classes of grammars to model families of protein sequences, we have remarked that almost no generalization was brought by these approaches because the condition to enable one word y_1 to be replaced by another word y_2 was almost never satisfied. As a matter of fact and with the notations of definition 1, y_1 and y_2 need to be surrounded by the same context x_1, z_1 in two sequences and y_1 has to exist as a substring of another sequence to imply the existence of a fourth sequence. If the sequences are long, observing a double occurrence of the common context x_1, z_1 and a double occurrence of y_1, given that at least one of these substrings has to be long, has a low likelihood in practice. Moreover, in the reversible setting, heads (and tails) have to be completely conserved from the beginning (to the end) of the sequences, *i.e.* the context has to be the same around y_1 and y_2 on the full length of the two sequences (let us note here that this requirement also holds for k, l-substitutability with x_1u, vz_1 as common required context). In our test sets, this did not occur, except for long y_1 and y_2 that were almost never repeated in other sequences. It seems clear that more local characterizations are needed in practice.

2. Locally Substitutable Languages

We propose here to introduce a new class of languages by considering only local contexts around words rather than the global ones required in substitutable languages. A consequence of this relaxation is to introduce the need for an additional parameter, the size of the context used. To allow an asymmetric left or right bias, we introduce two parameters k and l and define the class of k, l-local substitutable languages by:

Definition 3 (k, l-local substitutable language) *A language L is k, l-local substitutable if for any $x_1, y_1, z_1, x_2, y_2, z_2, x_3, z_3 \in \Sigma^*, u \in \Sigma^k, v \in \Sigma^l$, such that $uy_1v, uy_2v \neq \lambda$,*

$$x_1uy_1vz_1 \in L \wedge x_3uy_2vz_3 \in L \Rightarrow (x_2y_1z_2 \in L \Leftrightarrow x_2y_2z_2 \in L).$$

Then in a k, l-local substitutable language, all y_1 can be substituted by y_2 as soon as there exists sequences in this language in which they share a common (local) context u, v of size $|u| = k$ and $|v| = l$. If a language is k, l-local substitutable then it is m, n-local substitutable for any $m \geq k$ and $n \geq l$ and the hierarchy is strict.

To simplify the proofs and the definitions, let us assume that the alphabet Σ can be extended to $\Sigma' = \Sigma \cup \{\$\}$ where \$ is a new symbol not in Σ and that the sequences are padded at their extremities with this new symbol, so as contexts are always defined (each sequence is added k symbols \$ before its beginning and l symbols \$ after its end). Using this convention, the class of substitutable languages (Clark and Eyraud (2007)) can be stated as being the class of ∞, ∞-local substitutable.

By the same way that we have relaxed the condition on substitutable languages, we can get the local counterpart of k, l-substitutable languages:

Definition 4 (k, l-local context substitutable language) *A language L is k, l-local context substitutable if for any $x_1, y_1, z_1, x_2, y_2, z_2, x_3, z_3 \in \Sigma^*, u \in \Sigma^k, v \in \Sigma^l$, such that $uy_1v, uy_2v \neq \lambda$,*

$$x_1uy_1vz_1 \in L \wedge x_3uy_2vz_3 \in L \Rightarrow (x_2uy_1vz_2 \in L \Leftrightarrow x_2uy_2vz_2 \in L).$$

Again, if a language is k,l-local context substitutable then it is m,n-local context substitutable for any $m \geq k$ and $n \geq l$ and the hierarchy is strict.

Compared to k,l-local substitutable languages, the difference is that y_1 can be substituted by y_2 only in the contexts that they share. This raises a distinction between the considered contexts: the contexts used *to define* the equivalence classes (the set of substitutable y_i) and the contexts *for the application* of these equivalence classes. In the definition above, both contexts are of the same length. To mark the difference between the two kinds of contexts and generalize the class of languages, we can define the class of i,j-local k,l-context substitutable language, where (i,j) constrains the context size used in the local way to define the equivalence classes and (k,l) sets the minimal length condition on contexts where these substitutability classes apply.

We define i,j-local k,l-context substitutable languages with respect to the relative lengths of the two kinds of contexts [1] by:

1. $k \leqslant i \wedge l \leqslant j$ (brave substitutability)

   for any $x_1,y_1,z_1,x_2,y_2,z_2,x_3,z_3 \in \Sigma^*, u \in \Sigma^k, v \in \Sigma^l, a \in \Sigma^{i-k}, b \in \Sigma^{j-l}$ such that $uy_1v, uy_2v \neq \lambda$,

   $$x_1auy_1vbz_1 \in L \wedge x_3auy_2vbz_3 \in L \Rightarrow (x_2uy_1vz_2 \in L \Leftrightarrow x_2uy_2vz_2 \in L)$$

2. $k \geqslant i \wedge l \geqslant j$ (cautious substitutability)

   for any $x_1,y_1,z_1,x_2,y_2,z_2,x_3,z_3 \in \Sigma^*, c \in \Sigma^{k-i}, d \in \Sigma^{l-i}, r \in \Sigma^i, s \in \Sigma^j$ such that $ry_1s, ry_2s \neq \lambda$,

   $$x_1cry_1sdz_1 \in L \wedge x_3ry_2sz_3 \in L \Rightarrow (x_2cry_1sdz_2 \in L \Leftrightarrow x_2cry_2sdz_2 \in L)$$

Let us remark that if a language is i,j-local k,l-context sustitutable, then it is i,j-local a,b-context sustitutable with $a \geqslant k$ and $b \geqslant l$ and that this hierarchy with respect to context size is strict. Similarly, it is m,n-local k,l-context sustitutable with $m \geqslant i$ and $n \geqslant j$ and this hierarchy with respect to locality size is strict. Obviously, it is also m,n-local a,b-context sustitutable for $a \geqslant k$ and $b \geqslant l$ and $m \geqslant i$ and $n \geqslant j$.

The following table sums up the different classes of substitutable languages seen so far in the general framework of the i,j-local k,l-context substitutable language:

Language	local definition	contextual application
substitutable Clark and Eyraud (2007)	(∞,∞)	$(0,0)$
k,l-substitutable[2] Yoshinaka (2008)	(∞,∞)	(k,l)
k,l-local substitutable	(k,l)	$(0,0)$
k,l-local context substitutable	(k,l)	(k,l)
i,j-local k,l-context substitutable	(i,j)	(k,l)

1. Only the two main cases are detailed here, the two other cases could be defined in a similar way
2. Could be named k,l-context substitutable

Link with k-testable languages

k, l-local context substitutable languages form an appealing class mixing local and contextual substitution in a symmetrical way with few parameters. Moreover, we can establish a link between this class of languages and the family of locally testable languages, an interesting subclass of regular languages learnable from positive examples only, according to theoretical and practical points of view (Garcia et al. (1990); Garcia and Vidal (1990); Yokomori et al. (1994)).

Definition 5 (strictly k-testable language) *Let $L_k(w)$ and $R_k(w)$ be the prefix and the suffix of w of length k, respectively. Further, let $I_k(w)$ be the set of interior solid substrings of w of length k. A language L over S is strictly k-testable if and only if there exist finite sets A, B, C such that $A, B, C \subseteq S^k$, and for all w with $|w| \geq k$, $w \in L$ if and only if $L_k(w) \in A, R_k(w) \in B, I_k(w) \subseteq C$.*

In other words, according to Caron (2000), "Let L be a k-testable language. Let u and v be two words of Σ^* such that u and v have the same prefixes of length k, the same suffixes of length k and the same set of interior factors of length k. Then we have: $u \in L \Leftrightarrow v \in L$".

Let $x_1, y_1, x_2, y_2, x_3 \in \Sigma^*, u \in \Sigma^k$. We can thus characterize a k-testable language by:

$$\begin{array}{c} x_1uy_1 \in L \wedge x_2uy_2 \in L \Rightarrow x_1uy_2 \in L \wedge x_2uy_1 \in L \\ \vdash \\ \left\{ \begin{array}{l} x_1uy_1 \in L \wedge x_2uy_2 \in L \wedge x_3uy_2 \in L \Rightarrow x_2uy_1 \in L \\ x_1uy_1 \in L \wedge x_2uy_1 \in L \wedge x_3uy_2 \in L \Rightarrow x_2uy_2 \in L \end{array} \right. \\ \vdash \\ x_1uy_1 \in L \wedge x_3uy_2 \in L \Rightarrow (x_2uy_1 \in L \Leftrightarrow x_2uy_2 \in L) \end{array}$$

In terms of definition 4, left k-testable languages are thus $k, 0$-local context substitutable. We can proceed symmetrically by reading from the right to the left and define right l-testable languages that are then identical to $0, l$-local context substitutable. The k, l-local context substitutable languages are thus left k-testable and right l-testable languages.

Like k, l-substitutable languages are the counter part of reversible language, we have defined here the class of k, l-local context substitutable languages that can be seen as a bidirectional extension of local languages. Before studying the inference of such languages, we present some of their properties in the next section.

3. Closure properties

Yoshinaka (2008) has demonstrated some properties on k, l-substitutable languages. We present here similar results on the locally substitutable languages.

Proposition 1 *Locally substitutable languages are not closed under intersection with regular sets*

Let $L_0 = ae^*cf^*a \cup ae^*df^*a \cup be^*cf^*b$ and $L_1 = L_0 \cup be^*df^*b$. L_0 is regular and L_1 is $1, 1$-local substitutable. But $L_1 \cap L_0 = L_0$ is not substitutable for any k, l. Example : $ae^kcf^la \in L_0 \wedge ae^kdf^la \in L_0 \not\Rightarrow (be^kcf^lb \in L_0 \Leftrightarrow be^kdf^lb \in L_0)$

Proposition 2 *Locally substitutable languages are not closed under union*

Let $L_2 = ae^*cf^*a \cup ae^*df^*a$ and $L_3 = be^*cf^*b$. L_2 is 1, 1-local substitutable. But $L_2 \cup L_3 = L_0$ is not k, l-substitutable for any k, l.

Proposition 3 *Locally substitutable languages are not closed under concatenation*

Let $L_4 = ae^*c$ and $L_5 = e^*a$. L_4 and L_5 are k, l-local substitutable. But $L_4L_5 = ae^*ce^*a$ is not k, l-local context substitutable. Example : $ae^kee^lcea \in L_4L_5 \wedge ae^kce^la \in L_4 \cap L_5 \nRightarrow ae^xe^kee^lce^xa \in L_4L_5 \Leftrightarrow ae^xe^kce^lce^xa \in L_4L_5$

Proposition 4 *Locally substitutable languages are not closed under complement*

$L_6 = a^*b$ is k, l local substitutable, but the complement L_6^c is not k, l-substitutable for any k, l. Example : $ba^kaa^l \in L_6^c \wedge ba^kba^l \in L_6^c \nRightarrow a^kba^lb \in L_6^c \Leftrightarrow a^kaa^lb \in L_6^c$

Proposition 5 *Locally substitutable languages are not closed under Kleene closure*

$L_7 = (a^*ba^*)^+d$ is k, l local substitutable, but the kleene closure is not k, l-local context substitutable for any k,l. Example : $a^kba^ld \in L_7^* \wedge aba^kda^lbad \in L_7^* \nRightarrow a^kba^ld \in L_7^* \Leftrightarrow a^kda^ld \in L_7^*$

Proposition 6 *Locally substitutable languages are not closed under λ-free homomorphism*

$L_8 = ae^*cf^*a \cup be^*cf^*b \cup gy^*dx^*g$ is k, l-local substitutable. Let h be the homomorphism : $h(a) = a, h(b) = b, h(c) = c, h(d) = d, h(e) = e, h(f) = f, h(g) = a, h(x) = f, h(y) = e$. $h(L_8) = L_0$ is not k, l-substitutable.

Proposition 7 *Locally substitutable languages are not closed under inverse homomorphism*

$L_6 = a^*b$ is k, l-local substitutable. Let h be such that $h(a) = a, h(b) = b, h(e) = \Lambda$. $h^{-1}(L_6)$ is not k, l-substitutable. Example : $e^kae^lb \in h^{-1}(L_6) \wedge e^kae^lb \in h^{-1}(L_6) \nRightarrow be^kee^l \in h^{-1}(L_6) \Leftrightarrow be^kae^l \in h^{-1}(L_6)$

Proposition 8 *Locally substitutable languages are not closed under reversal (for k, l-local substitutable languages with $k \neq l$)*

If L is k, l-local substitutable, its reversal is l, k-local substitutable. So if $k \neq l$, it is not closed under reversal.

Proposition 9 *Locally substitutable languages are closed under intersection*

Let L and L' be k, l-local substitutable. If $x_1vy_1uz_1$,
$x_3vy_2uz_3, x_2y_1z_2 \in L \cap L'$ for some $v \in \Sigma^k, u \in \Sigma^l$, then those are in both L and L'. Since L and L' are k, l-local substitutable, $x_2y_2z_2$ is in both L and L' and thus in $L \cap L'$.

Proposition 10 *Locally substitutable languages are closed under λ-free inverse homomorphism*

Let L be a k,l-local substitutable language and h a λ-free homomorphism. Let $h(w) = \overline{w}$ for readability. If $x_1uy_1vz_1, x_3uy_2vz_3$, $x_2uy_1vz_2 \in h^{-1}(L)$ for some $u \in \Sigma^k$,$v \in \Sigma^l$ and $uy_1v, uy_2v \in \Sigma^+$, then $\overline{x_1uy_1vz_1}, \overline{x_1uy_2vz_1}, \overline{x_2uy_1vz_2} \in L$. Since L is k,l-local substitutable and $|\overline{u}| \geqslant |u| = k$, $|\overline{v}| \geqslant |v| = l$, $|\overline{uy_1v}|, |\overline{uy_2v}| \geqslant 1$, we have $\overline{x_2uy_2vz_2} \in L$ and so $x_2uy_2vz_2 \in h^{-1}(L)$.

While the main results are negative, except for the closure under intersection, this last proposition suggests that, as for local languages, morphic generator grammatical inference methodologies embedding expert knowledge in the sequences by renaming symbols Garcia et al. (1987) could be developed for learning locally substitutable languages.

4. Algorithms

We straightforwardly adapt here to locally substitutable languages the simple learning algorithm presented in Yoshinaka (2008). From a given sample set K and the pair of parameters k and l, a grammar is built according to k,l-local substitutability constraints in algorithm 1 and k,l-local context substitutability constraints in algorithm 2. It can be noticed here that like in Yoshinaka (2008), the considered grammars have at most two non-terminals in the right-hand-side and that these algorithms do not necessarily return a grammar of the right class of languages, even if they are expected to converge towards the target language when enough examples are available.

Algorithm 1 $\hat{G}_{LS}$ (k,l-local substitutability)

Input: Set of sequences K, parameters k and l
Output: Grammar $\hat{G} = \langle \Sigma_K, V_K, P_K, S \rangle$
$V_K = \{[y] \mid xyz \in K, y \neq \lambda\} \cup \{S\}$
$P_K = \{S \to [w] \mid w \in K\}$
$\cup \{[a] \to a \mid a \in \Sigma\}$
$\cup \{[xy] \to [x][y] \mid [xy], [x], [y] \in V_K\}$
$\cup \{[y_1] \to [y_2] \mid x_1uy_1vz_1 \in K, x_2uy_2vz_2 \in K, \mid u \mid = k, \mid v \mid = l\}$

Algorithm 2 $\hat{G}_{LCS}$ (k,l-local context substitutability)

Input: Set of sequences K, parameters k and l
Output: Grammar $\hat{G} = \langle \Sigma_K, V_K, P_K, S \rangle$
$V_K = \{[y] \mid xyz \in K, y \neq \lambda\} \cup \{S\}$
$P_K = \{S \to [w] \mid w \in K\}$
$\cup \{[a] \to a \mid a \in \Sigma\}$
$\cup \{[xy] \to [x][y] \mid [xy], [x], [y] \in V_K\}$
$\cup \{[uy_1v] \to [uy_2v] \mid x_1uy_1vz_1 \in K, x_2uy_2vz_2 \in K, \mid u \mid = k, \mid v \mid = l\}$

Note that the last rule of algorithm 2 contains the two kinds of contexts defined in the previous sections: $\{ \underbrace{[uy_1v] \rightarrow [uy_2v]}_{<u,v> \text{ is application context}} | \underbrace{x_1uy_1vz_1 \in K, x_2uy_2vz_2 \in K,}_{<u,v> \text{ is definition context}} | u |= k, | v |= l\}$

while algorithm 1 only uses the local context for defining the content of the substitutability classes.

Given that the k,l-local substitutable languages and the k,l-local context substitutable languages are included in the class of the k,l-substitutable languages, they are learnable with the algorithm from Yoshinaka (2008). It has still to be investigated, but the similarity of the representations and algorithms should allow to obtain learnability results similar to those obtained by Yoshinaka (2008) or Luque and López (2010) in our setting. We limit ourselves here to a more pragmatic perspective and focus on the difference that these new classes of languages introduce in learning by minimal generalization approaches. First let us illustrate by some examples the different languages that can be learned by minimal generalization from the set of positive data $K = \{abcde, abfde, yzcji, vzmjk\}$ with respect to the chosen class of languages for small values of k and l.

- $0,0$-substitutability (Clark and Eyraud (2007))

  $N \rightarrow abXde|yzXji|vzmjk$
  $X \rightarrow c|f$

  $L = \{abcde, abfde, yzcji, vzmjk, yzfji\}$

- $1,1$-substitutability (Yoshinaka (2008))

  $N \rightarrow aXe|yzcji|vzmjk$
  $X \rightarrow bcd|bfd$

  $L = \{abcde, abfde, yzcji, vzmjk\}$

- $1,1$-local substitutability

  $N \rightarrow abXde|yzXji|vzXjk$
  $X \rightarrow c|f|m$

  $L = \{abcde, abfde, yzcji, vzmjk, yzmji, vzcjk, yzfji, vzfjk, abmde\}$

- $1,1$-local context substitutability

  $N \rightarrow aXe|yX_2i|vX_2k$
  $X \rightarrow bcd|bfd$
  $X_2 \rightarrow zmj|zcj$

  $L = \{abcde, abfde, yzcji, vzmjk, yzmji, vzcjk\}$

More generally, given a learning sample set K, we can establish an inclusion hierarchy between the least general generalizations produced for the different kinds of language constraints. If we denote by $L_X(K)$ the least general language of class X including K, we have the following inclusions.

Proposition 11 $L_{k,l\text{-substitutable}}(K) \subseteq L_{k,l\text{-local context substitutable}}(K)$

Proof $x_1uy_1vz_1 \in L \wedge x_3uy_2vz_3 \in L \Rightarrow (x_2uy_1vz_2 \in L \Leftrightarrow x_2uy_2vz_2 \in L)$
If $x_1 = x_3 \wedge z_1 = z_3$:
$x_1uy_1vz_1 \in L \wedge x_1uy_2vz_1 \in L \Rightarrow (x_2uy_1vz_2 \in L \Leftrightarrow x_2uy_2vz_2 \in L)$
So, all the words that are added to satisfy k, l-substitutability are also added for k, l-local context substitutability. ■

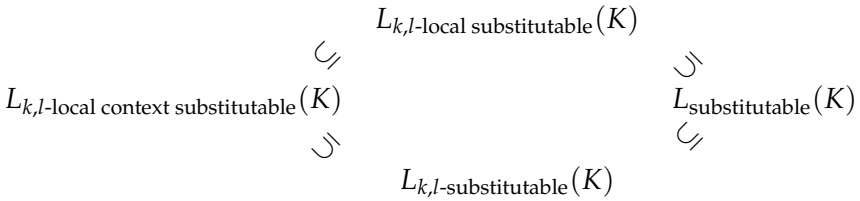
Proposition 12 $L_{substitutable}(K) \subseteq L_{k,l\text{-local substitutable}}(K)$

Proof $x_1uy_1vz_1 \in L \wedge x_3uy_2vz_3 \in L \Rightarrow (x_2y_1z_2 \in L \Leftrightarrow x_2y_2z_2 \in L)$
If $x_1u = x_3u(= x_4) \wedge vz_1 = vz_3(= z_4)$:
$x_4y_1z_4 \in L \wedge x_4y_2z_4 \in L \Rightarrow (x_2y_1z_2 \in L \Leftrightarrow x_2y_2z_2 \in L)$ ■

Proposition 13 $L_{k,l\text{-local context substitutable}}(K) \subseteq L_{k,l\text{-local substitutable}}(K)$

Proof $x_1uy_1vz_1 \in L \wedge x_3uy_2vz_3 \in L \Rightarrow (x_2y_1z_2 \in L \Leftrightarrow x_2y_2z_2 \in L)$
If $x_2 = x_4u \wedge z_2 = vz_4$:
$x_1uy_1vz_1 \in L \wedge x_3uy_2vz_3 \in L \Rightarrow (x_4uy_1vz_4 \in L \Leftrightarrow x_4uy_2vz_4 \in L)$ ■

To sum up these propositions, we have the following inclusions between the different substitutable language closures of a given set of sequences K:



5. Experiments

We evaluated our method using a set of sequences which are members of the legume lectins protein family corresponding to PROSITE entry PS00307[3](Hulo et al. (2006)). This protein family is used as a benchmark in one of the rare study in the literature applying grammar learning on protein sequences with higher order dependencies (Dyrka and Nebel (2009)). Prosite is a database collecting protein domains and families with an associated signature that is either a regular expression or a HMM profile matching a characteristic region of the protein sequence. Prosite provides for each family the set of known true positive, false positive and false negative hits with respect to the proposed signature. Lectins are proteins that are generally found in plant seeds and play a role in binding calcium and manganese ions.

3. http://prosite.expasy.org/PS00307

This section presents the results obtain for the various generalization criteria defined so far: substitutable, local substitutable and local context substitutable.

The experimental setting distinguishes three sets of sequences from the Prosite data:

- the training set is the training set used in Dyrka and Nebel (2009), except that the entire protein sequences are processed rather than the subsequences of length 50 around the active site. This makes the issue a bit harder but is more realistic on protein families. The training set contains 22 sequences.
- the negative test set is made of the ten first sequences in the list of false positive hits provided by Prosite.
- the positive test set is also made of ten sequences from Prosite data, six in the true positive list and four in the false negative hits list.

As explained in section 1, the raw amino-acids sequences are preprocessed, leading to a set of PLMAs and a set of sequences of PLMA occurrences. We have computed the necessary multiple alignments using paloma v.1.9 with default parameters, the minimal quorum being set to 2. Each PLMA is itself coded as a sequence of ambiguous characters, that is a sequence of subsets of amino-acids. This is achieved by transforming each amino-acid in the PLMA by a subset of amino-acids known to be interchangeable without functional loss. Subsets are extracted from a standard amino acid substitution matrix (Matrix Blosum62) that scores the degree to which a given amino-acid may be substituted by another one. The recognition of each PLMA is ensured via the integration of proper rules in the learned grammar.

Our algorithm is applied on the sequences of PLMA occurrences, leading to a training set with average sequence length 20. We used values $k = l = 4$ for the locality parameters. A final filtering post-processing step simplifies the grammar in order to keep a good level of grammar comprehensibility and parsing efficiency.

Indeed, the grammar generated by our algorithm would contain a lot of redundancy and ambiguity by keeping all the possibly relevant contexts found in the examples. For instance, the presence of long repeated words in the training set generates systematically a number of rules corresponding to the combinatorics of repeat inclusion.

In practice, we have defined the following transformations on the learned grammar:

- *Factorization*

  If all the productions of a nonterminal have the same prefix (or suffix), they are deleted from the grammar.
  For instance, if the productions of nonterminal X are $X \longrightarrow aY|aZ$, this rule can be deleted. The two equivalent rules $A \longrightarrow Y|Z$ and $B \longrightarrow aA$ exist anyway (unless Y or Z equals ϵ) and thus the transformation will be safe most of the time with respect to the language that is recognized.

- *Disambiguation*

  The following set of rules in Chomsky normal form:

  $$S \longrightarrow XU$$
  $$S \longrightarrow VZ$$
  $$U \longrightarrow YZ$$
  $$V \longrightarrow XY$$

  will be replaced by the single rule :

  $$A \longrightarrow XYZ$$

- *Cleaning*

  All the production rules with a head that is non reachable from the axiom are deleted.

All sequences of the test sets have been parsed using the NLTK chart parser(Bird et al. (2009)) . Table 1 provides a summary of the results in terms of Precision, Recall and F-measure (Precision is the ratio of true positive over all predicted positive, Recall is the ratio of true positive over all positive in the test set and F-measure is the harmonic mean of precision and recall). The last lines of the table give an overview of the behaviour of the grammar learned in Dyrka and Nebel (2009) on our test sets. As stated before, these results are not fully comparable to ours since the grammar is learned on carefully chosen substrings of the sequences. In this approach, each parsed sequence obtains a score. Thus, precision and recall depend on a threshold value for this score. We have provided results for three characteristic values of the threshold: a maximal precision, a maximal recall, and a maximal F-measure. Of course it is hard in practice to fix the threshold and the true result would be an intermediary point on the ROC curve.

	Precision	Recall	F-measure
Substituable	1	0.2	0.33
Local context substitutable	1	0.6	0.75
Local substitutable	1	0.7	0.82
Stochastic CFG	1	0.1	0.18
	0.3	1	0.46
	0.8	0.9	0.85

Table 1: Sequence annotation by grammars obtained for the PS00307 family

Although such a test can only be considered as an illustrative experiment, locality clearly allows to greatly improve the generality of learned grammars and the sensitivity of the recognition. Moreover, this has not impaired the specificity of the characterization since no member of the negative training set could be parsed. This is a consequence of considering high level dependencies observable in the data in addition to local dependencies. Applying local context substitutability rather than local substitutability gives a

slightly less sensitive prediction, an expected outcome since the corresponding generalization is weaker. A last remark is that our approach compares favourably to the stochastic CFG learning approach since it enables to obtain a good sensitivity at the intended maximal level of specificity whereas stochastic CFG sensitivity reaches an unacceptable level of generalization. Furthermore, our approach almost reaches the maximal F-measure and does not need the knowledge of active sites in training sequences, nor to fix a threshold parameter for the recognition. Overall, such results seem very encouraging for turning into practice the theoretically important concept of substitutability in grammatical inference.

6. Conclusion

We have introduced the classes of locally substitutable languages extending the notion of k-testability beyond regular languages like substitutable and (k, l)-substitutable languages extends k-reversibility. This parallel, their natural definition and the qualitative change of the inductive leap brought by these characterization let us expect that this class of languages can be fruitful for grammatical inference from positive examples. This work raise interesting open questions. From the point of view of formal languages, this class of languages has very strong links with the k, l-Greibach Normal Form introduced by Yoshinaka (2008) and we would like to investigate more deeply on this point. From the point of view of learning, the theoretical learnability of the classes has to be established. The previous point seems a good first step in this direction. First results on protein sequences are promising. We are currently working on the design of an efficient learning algorithm able to cope with larger sets of protein sequences. Indeed, the practical application of inference requires to check a number of parameters values and to cross-validate results on many sequences. A fast implementation of the core learning algorithm is thus a critical point with respect to its transfer to non toy problems.

Acknowledgments

The authors wish to acknowledge the reviewers for their insightful comments to the manuscript. This work has been partially funded by French ANR under project Idealg.

References

S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *Journal of molecular biology*, pages 403–410, October 1990.

T. L. Bailey, M. Boden, F. A. Buske, M. Frith, C. E. Grant, L. Clementi, J. Ren, W. W. Li, and W. S. Noble. Meme suite: tools for motif discovery and searching. *Nucleic Acids Research*, 37(suppl 2):W202–W208, 2009. doi: 10.1093/nar/gkp335.

Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly, 2009.

P. Caron. Families of locally testable languages. *Theoretical Computer Science*, 242(1–2):361 – 376, 2000.

A. Clark and R. Eyraud. Polynomial identification in the limit of substitutable context-free languages. *Journal of Machine Learning Research*, 8:1725–1745, August 2007.

F. Coste and G. Kerbellec. A similar fragments merging approach to learn automata on proteins. In João Gama, Rui Camacho, Pavel Brazdil, Alípio Jorge, and Luís Torgo, editors, *ECML*, volume 3720 of *Lecture Notes in Computer Science*, pages 522–529. Springer, 2005. ISBN 3-540-29243-8.

R. Durbin. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998. ISBN 9780521629713.

Witold Dyrka and Jean C. Nebel. A stochastic context free grammar based framework for analysis of protein sequences. *BMC Bioinformatics*, 10(1):323+, October 2009.

M. Y. Galperin and E. V. Koonin. From complete genome sequence to 'complete' understanding? *Trends in Biotechnology*, 28(8):398 – 406, 2010. ISSN 0167-7799. doi: 10.1016/j.tibtech.2010.05.006.

P. Garcia and E. Vidal. Inference of k-testable languages in the strict sense and application to syntactic pattern recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(9):920–925, 1990.

P. Garcia, E. Vidal, and F. Casacuberta. Local languages, the succesor method, and a step towards a general methodology for the inference of regular grammars. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-9(6):841 –845, nov. 1987. ISSN 0162-8828. doi: 10.1109/TPAMI.1987.4767991.

P. Garcia, E. Vidal, and J. Oncina. Learning locally testable languages in the strict sense. In *First int. workshop on Algorithmic Learning theory, ALT'90*, pages 325–338, 1990.

Nicolas Hulo, Amos Bairoch, Virginie Bulliard, Lorenzo Cerutti, Edouard De Castro, Petra S. Langendijk-genevaux, Marco Pagni, and Christian J. A. Sigrist. The prosite database. *Nucleic Acids Res*, 34:227–230, 2006.

S. Hunter, P. Jones, A. Mitchell, R. Apweiler, T. K. Attwood, A. Bateman, T. Bernard, D. Binns, P. Bork, S. Burge, E. de Castro, P. Coggill, M. Corbett, U. Das, L. Daugherty, L. Duquenne, R. D. Finn, M. Fraser, J. Gough, D. Haft, N. Hulo, D. Kahn, E. Kelly, I. Letunic, D. Lonsdale, R. Lopez, M. Madera, J. Maslen, C. McAnulla, J. McDowall, C. McMenamin, H. Mi, P. Mutowo-Muellenet, N. Mulder, D. Natale, C. Orengo, S. Pesseat, M. Punta, A. F. Quinn, C. Rivoire, A. Sangrador-Vegas, J. D. Selengut, C. J. A. Sigrist, M. Scheremetjew, J. Tate, M. Thimmajanarthanan, P. D. Thomas, C. H. Wu, C. Yeats, and S. Yong. Interpro in 2011: new developments in the family and domain prediction database. *Nucleic Acids Research*, 40(D1):D306–D312, 2012. doi: 10.1093/nar/gkr948.

I. Jonassen, J.F. Collins, and D. Higgins. Finding flexible patterns in unaligned protein sequences. *Protein Science*, 4(8):1587–1595, 1995.

G. Kerbellec. *Apprentissage d'automates modélisant des familles de séquences protéiques*. PhD thesis, Université Rennes 1, 2008.

F. M. Luque and G. G. Infante López. Pac-learning unambiguous k, l-nts $^{<=}$ languages. In J. M. Sempere and P. García, editors, *ICGI*, volume 6339 of *Lecture Notes in Computer Science*, pages 122–134. Springer, 2010. ISBN 978-3-642-15487-4.

P. Peris, D. López, M. Campos, and J. M. Sempere. Protein motif prediction by grammatical inference. In Yasubumi Sakakibara, Satoshi Kobayashi, Kengo Sato, Tetsuro Nishino, and Etsuji Tomita, editors, *ICGI*, volume 4201 of *Lecture Notes in Computer Science*, pages 175–187. Springer, 2006. ISBN 3-540-45264-8.

J. D. Thompson, D. G. Higgins, and T. J. Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22:4673–4680, November 1994.

T. Yokomori, N. Ishida, and S. Kobayashi. Learning local languages and its application to protein α-chain identification. In *HICSS (5)*, pages 113–122, 1994.

R. Yoshinaka. Identification in the limit of (k,l)-substitutable context-free languages. In *Proceedings of the 9th international colloquium conference on Grammatical inference: theoretical results and applications*, ICGI'09, pages 266–279, 2008.

J. Zhang, R. Chiodini, A. Badr, and G. Zhang. The impact of next-generation sequencing on genomics. *Journal of Genetics and Genomics*, 38(3):95 – 109, 2011. ISSN 1673-8527. doi: 10.1016/j.jgg.2011.02.003.