Chapter 8
Learning the Language of Biological Sequences

François Coste

Abstract The application to biological sequences is an appealing challenge for Grammatical Inference. While some first successes have already been recorded, such as the inference of profile Hidden Markov Models or stochastic Context-Free Grammars which are now part of the classical Bioinformatics toolbox, it is still a nice and open source of problems or inspiration for our research, with the possibility to apply our ideas to real fundamental applications. In this chapter, we survey biological sequences' main specificities and how they are handled in Pattern/Motif Discovery in order to introduce the important concepts and techniques used and present the latest successful approaches in that field by Grammatical Inference.

8.1 Linguistic Metaphor

New sequencing technologies are giving access to an ever increasing amount of DNA, RNA or protein sequences for more and more species. One major challenge in the post-genomic era is now to decipher this set of genetic sequences composing what has been popularly named "the language of life" [1].

As witnessed by this expression, the linguistic metaphor has been used for a long time in genetics. Indeed, the discovery of the double helix structure of DNA in 1953 showed that the genetic information contained in this biological macromolecule can be represented by two (long) complementary sequences over a four-letter alphabet $\{A, C, G, T\}$ symbolizing the *nucleotides*, the complementary letters (called Watson–Crick *base pairs*) being A–T and C–G. This genetic information is used to construct and operate a living organism by the *transcription* when needed of pieces of DNA sequences, named *genes*, into RNA single strand macromolecules which can also be represented by a sequence on almost the same four-letter alphabet $\{A, C, G, U\}$, where T has been replaced by its unmethylated form U. Sequences of RNAs coding for proteins are in turn

F. Coste (✉)
Inria Rennes—Bretagne Atlantique, Campus de Beaulieu, 35042
Rennes Cedex, France
e-mail: francois.coste@inria.fr

J. Heinz and J.M. Sempere (eds.), *Topics in Grammatical Inference*,
DOI 10.1007/978-3-662-48395-4_8

translated into protein sequences of *amino acid residues*, over the 20 amino acid's alphabet $\{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y\}$, that determine their three-dimensional conformations and functions in the cells (see for instance [2] for a more detailed introduction to the production of RNAs and proteins encoded in DNA). Sequences are thus at the core of storage of heredity information and its expression into the functional units of the cells: the natural language metaphor arises then quickly. This metaphor may be convenient for vulgarization but can also be a source of inspiration for scientists trying to discover the functional units of the genome and how this "text" is structured.

Applying computational linguistics tools to represent, understand and handle biological sequences is a natural continuation of the linguistic metaphor. Using formal grammars, such as the ones introduced in 1957 by Noam Chomsky [3] to describe natural languages and study syntax acquisition by children, has been advocated in particular by Searls: his articles provide a good introduction to the different levels of expressiveness required to model biological macromolecules by grammatical formalisms [4–8]. Basically, copies and long-distance correlations are common in genomic sequences, calling rapidly for context-sensitive grammars in Chomsky's hierarchy to model them, which makes parsing unworkable. As in linguistics, a solution to getting polynomial-time parsing and still representing many of the non-local constraints from genomic sequences is to use mildly context-sensitive languages [9]. Along this way, Searls introduced String Variable Grammars as an expressive formalism for describing the language of DNA that has led to several generic practical parsers: the precursor Genlang [10] and its successors Stan [11], Patscan [12], Patsearch [13] and Logol [14]. Many specialized parsers have also been devised, as for instance RNAMotif [15], RNAbob [16], Hypasearch [17, 18], Palingol [19] and Structator [20], tailored to handle efficiently RNA stem-loop secondary structures.

But one has still to design the grammar. In contrast with all the expertise available on natural languages, little is known about the syntax of DNA and the functional/semantic role of its parts. For instance, how are the equivalents of "words", "sentences" and even "punctuation marks" defined? In some specific cases, expert knowledge can be used to build a grammar, eventually by successive trial-and-error refinements with respect to the sequences retrieved by the model. In the other cases, expert knowledge is missing or is insufficient.

On the other hand, a huge number of genomic sequences are available, opening the door to grammar inference from these sequences. In this chapter, we will present advances made towards the big challenge of learning automatically the language of genomic sequences. The first step we consider is to discover what the genomic "words" are: this is mainly the domain of Motif Discovery, and related work is presented in Sect. 8.2. The second step is then to learn the "syntax" governing the admissible chaining of "words" in macromolecules: this is the classical goal of Grammatical Inference and we present the first successes obtained at the intersection of this field and Bioinformatics in Sect. 8.3.

8.2 Discovering and Modeling Biological Words

"Words" can be looked at different levels in DNA, requiring different levels of modelization. We investigate in this section how this has been classically handled in Bioinformatics from the simplest historical first steps, introducing and illustrating some specificities of biological sequences, to the more elaborate techniques from today's state of the art.

8.2.1 Short DNA Words

Simple Words A classical example of an identified DNA substring is AAGCTT (on the upper strand and its complement TTCGAA on the lower strand), that is specifically recognized in *H. influenzae* bacteria by one of its enzymatic proteins named HindIII that cleaves the double strand DNA of invading viruses at the sites where this substring occurs, while the bacteria's occurrence sites of the substring in its DNA are protected from cleavage by a prior methylation. The HindIII protein is said to be a restriction enzyme. More than 800 different restriction enzymes and more than 100 corresponding recognition sequences have been identified in bacterial species, with important applications in genetic engineering. These recognition sequences show a great variability among species, many of them being palindromic on complementary strands (meaning the sequence reads the same backwards and forwards in complementary DNA strands, like in AAGCTT and TTCGAA), reflecting that both strands of DNA have to be cut, often by a complex of two identical proteins operating on each strand. The main characteristic of these substrings is their short length (about four to eight base pairs), that makes them likely to appear frequently in any genome, providing them an efficient defense against unknown invading viruses. These sequences are thus rather ubiquitous and do not support information by themselves (they are only substrings recognized by the restriction enzymes), and it could be discussed whether they are "words" in a linguistic sense.

Conserved Words Another example of a well-known short sequence is the Pribnow box, early identified in the DNA of *E. coli* bacteria. It was discovered by Pribnow [21] by looking at the DNA sequences around six, experimentally determined, starting points of the transcription of genes into RNAs by a molecule named RNA-polymerase. Would you find in these sequences, shown hereafter and aligned on the known transcription start site formatted in bold, the protein binding site initiating the transcription by the RNA-polymerase?

```
Site1:      ...AAGTAAACACGGTACGATGTACCAC A TGAAACGACAGTGAGTCA...
Site2:        ...TGCTTCTGACTATAATAGACAGG G TAAAGACCTGATTTTTGA...
Site3:       ...TTTATTGCAGCTTATAATGGTTAC A AATAAAGCAATAGCA...
Site4:        ...CCACTGGCGGTGATACTGAGCAC A TCAGCAGGACGCACTGAC...
Site5:        ...CGTCATTTGATATGATGCGCCCC G CTTCCCGATAAGGGAGCA...
Site6:        ...CTTCCGGCTCGTATGTTGTGTGG A ATTGTGAGCGGATAACAA...
```

By looking carefully at the sequences, one can find a conserved region (underlined below), located about 10 positions before the transcription start site, that may have been conserved despite mutations for its function through natural selection:

```
Site1:  ...AAGTAAACACGG TACGATG TACCAC A TGAAACGACAGTGAGTCA...
Site2:     ...TGCTTCTGAC TATAATA GACAGG G TAAAGACCTGATTTTTGA...
Site3:  ...TTTATTGCAGCT TATAATG GTTAC A AATAAAGCAATAGCA...
Site4:     ...CCACTGGCGGT GATACTG AGCAC A TCAGCAGGACGCACTGAC...
Site5:     ...CGTCATTTGA TATGATG CGCCCC G CTTCCCGATAAGGGAGCA...
Site6:     ...CTTCCGGCTCG TATGTTG TGTGG A ATTGTGAGCGGATAACAA...
```

Consensus Sequences and Motifs Looking at the underlined alignment of this conserved region, only two positions (the second and the sixth) are strictly conserved out of seven and the farthest sequences share only three identical positions for four mismatches. But the *consensus sequence* TATAATG of the alignment, built by keeping only the most abundant letter at each position, appears with no more than two mismatches, and one may consider it as an archetypal (eventually ancestral) sequence for the region and the other sequences as its variants by meaningless mutations. Searching for this consensus sequence TATAATG without mismatch, we would retrieve only one of the six conserved sites and we would expect one match per $4^7 \simeq 16{,}000$ bp in whole DNA. Allowing one mismatch, we would retrieve three of the six sites and we would expect one match per 700 bp. Allowing two mismatches, we would retrieve all the sites but we would expect one match per 70 bp, which is likely to be too much.

We can remark that the nucleotides A and G are evenly distributed at the fourth underlined position and TATGATG would also have been a good candidate consensus sequence. Actually, it would be more informative to know that the fourth position has to be a purine (*A* or *G* bases) and, as done by Pribnow, we can use the *consensus motif* TAT[AG]ATG (where brackets specify a set of alternative bases at the position) to designate the sequences probably engaged by RNA polymerase. This motif retrieves two sites for one expected match per 8,000 bp, and five sites if one error is allowed for one expected match per 400 bp. If we assume that the base at the fifth position is not important, we can also relax the consensus to TAT[AG]xTG where x is a wildcard for any base. This consensus retrieves four of the sites with about one match per 2,000 bp and all the sites if one error is allowed with a match per 100 bp. And choosing the full *consensus* [GT]A[CT][AG][ACT]T[AG] would recognize all the sites and would expect a match per 350 bp. As shown in this example, consensus offers many ways to model a word and its possible variants in DNA, ranging from consensus sequences allowing a limited number of errors to full consensus motifs, with all the intermediate ambiguity/sensitivity trade-offs.

"De novo" discovery of such words can be done by enumerating them and returning those over-represented in a collection of genome sequences, i.e. occurring more frequently than expected by chance. This approach has been successful in Motif Discovery, particularly for the discovery of short words and rather simple motifs (to enable a practical enumeration, even if efficient datastructures can be used and enumerating only the motifs that have sufficient support in the sequences can help); see [22, 23] for details.

Position Specific Matrices Yet, consensus sequences or motifs are not completely satisfactory for representing and discovering biological words. Taking again the example of the full consensus motif [TG]A[TC][GA][ACT]T[GA], do we really want GACACTA to be recognized like TATAATG? Or if a more specific consensus, such as the consensus sequence TATAATG, is chosen, allowing a limited number of errors, how is it possible to express that some positions can mutate more easily than others and that some base mutations occur more likely at some positions? Moreover, while conserved on average, the binding sites involved in the initiation of transcription occur rarely as exact matches of their specific consensus sequence. On average in bacteria, only half of the positions in each site match with the consensus sequence. A first explanation is that bindings have to be reversible. Different affinities with binding proteins enable as also to tune at a fine level the concentration of the RNA (and eventually protein) genes expressed in the cell, which would be interesting to estimate from the motif.

Weighting the consensus motifs addresses these issues. This is usually done on the basis of a summary of the sites by their base count at each position in a *position-specific count matrix* (PSCM). For the Pribnow sites example, the PSCM for the aligned conserved region would be:

	1	2	3	4	5	6	7
A	0	6	0	3	4	0	1
C	0	0	1	0	1	0	0
G	1	0	0	3	0	0	5
T	5	0	5	0	1	6	0

If we denote by $o_i(a)$ the observed count of base a at position i of the sites, estimation of probability of a at i in the site is given by:

$$\hat{p}_i(a) = \frac{o_i(a)}{\sum_{a' \in \{A,C,G,T\}} o_i(a')}.$$

Under the strong assumption that the probability of a base at a position depends only on the position, the probability of a sequence on $a_1 a_2 \ldots a_k$ given a *position-specific probability matrix* (PSPM) $\mathbf{P} = [\mathbf{p_1}, \mathbf{p_2} \ldots, \mathbf{p_k}]$ is $\Pi_{i=1}^{k} p_i(a_i)$. For instance, for the example above, the probability of TATAATG would be $\frac{5}{6} \times \frac{6}{6} \times \frac{5}{6} \times \frac{3}{6} \times \frac{4}{6} \times \frac{6}{6} \times \frac{5}{6} \simeq 1.2 \times 10^{-4}$ while for GACACTA it would only be $\frac{1}{6} \times \frac{6}{6} \times \frac{1}{6} \times \frac{3}{6} \times \frac{1}{6} \times \frac{6}{6} \times \frac{1}{6} \simeq 6.8 \times 10^{-5}$. By way of comparison, both sequences would have a probability of $(\frac{1}{4})^7 \simeq 6.1 \times 10^{-5}$ of being generated randomly by an equiprobable choice of the bases.

In the genome of *S. cerevisiae* which contains 64 % of A and T, the probability of TATAATG and GACACTA would respectively be 2×10^{-4} and 6×10^{-6}, making the second word more exceptional and thus more interesting than the first word with respect to this background model. When positions are assumed to be independent, the odd-score of the probability of a sequence $a_1 a_2 \ldots a_k$ by $[\mathbf{p_1}, \mathbf{p_2} \ldots, \mathbf{p_k}]$ with respect to its probability in a background model where each base a has a probability $p(a)$

can directly be computed by $\Pi_{i=1}^{k} \frac{p_i(a_i)}{p(a_i)}$, and the comparison with respect to expected background probability can directly be embedded in a *Position Weight Matrix* (PWM) [24], also called Position-Specific Weight Matrix (PSWM) or *Position-Specific Scoring Matrix* (PSSM), in logarithm form to facilitate computation (sum instead of product and better precision for rounded computation). In a PWM, the score of base a at position i is usually defined by

$$s_i(a) = \log_2 \frac{p_i(a)}{p(a)}$$

and the score of a sequence $a_1 a_2 \ldots a_k$ is given by

$$S(a_1 a_2 \ldots a_k) = \sum_{i=1}^{k} s_i(a_i).$$

The PWM computed from the PSCM above, assuming that the bases are equiprobable in the background model ($p(A) = p(C) = p(G) = p(T)$), would be:

	1	2	3	4	5	6	7
A	$-\infty$	2	$-\infty$	1	1.42	$-\infty$	−0.58
C	$-\infty$	$-\infty$	−0.58	$-\infty$	−0.58	$-\infty$	$-\infty$
G	−0.58	$-\infty$	$-\infty$	1	$-\infty$	$-\infty$	1.74
T	1.74	$-\infty$	1.74	$-\infty$	−0.58	2	$-\infty$

Bases over-represented with respect to background probability have positive scores, while under-represented bases have negative scores. Using a sliding window of width k, PWM can assign a score at each site of a genome reflecting its likelihood of being part of the motifs. The highest score for a sequence with the PWM above is 11.64, obtained for TATAATG, while the lowest score (except $-\infty$) is 2.68, obtained for GACACTA.

First Pseudocounts Let us remark that a mutation from A to G at the fifth position of the best sequence TATAATG will directly result in $-\infty$ score. Nucleotides that occur rarely in the motif at a specific position may not be seen in a small sample by chance but will force the probability of any sequence containing one of these missing nucleotides to 0. Pseudocounts are thus usually added to compensate for small samples counts. This can be done by adding systematically 1 to the observed counts, and the estimate of probability of a at i will then be:

$$\hat{p}_i(a) = \frac{o_i(a) + 1}{\sum_{a'} (o_i(a') + 1)}.$$

More elaborate pseudocounts can be used; for instance, in

$$\hat{p}_i(a) = \frac{o_i(a) + A\, p(a)}{\sum_{a'} (o_i(a') + A\, p(a'))}$$

the pseudocount added is proportional to background probability $p(a)$ and the weight A given to the prior. Choosing $A = 2$ and keeping the equiprobability of the bases, the PWM on the Pribnow example would be

	1	2	3	4	5	6	7
A	−2.00	1.70	−2.00	0.81	1.17	−2.00	−0.42
C	−2.00	−2.00	−0.42	−2.00	−0.42	−2.00	−2.00
G	−0.42	−2.00	−2.00	0.81	−2.00	−2.00	1.46
T	1.46	−2.00	1.46	−2.00	−0.42	1.70	−2.00

and we would have $S(\text{TATAATG}) = 9.76$, $S(\text{GACACTA}) = 2.53$ and $S(\text{TATAGTG}) = 6.59$, the minimal score being -14. By adding pseudocounts, all the sequences have a score strictly greater than $-\infty$. One can still discriminate a set of sequences by choosing a cut-off value, chosen as a compromise between desired recall and precision, with the advantage over sequence consensus or motifs of being better suited for the representation of similar sequences without a strict conservation per position.

Measuring Conservation The conservation of a site can be evaluated according to a measure named information content [25] that measures the information gain on the site provided by the PSPM with respect to a uniform random choice of the bases. The information content IC_i at position i is given by the formula:

$$IC_i = 2 + \sum_a p_i(a) \log_2 p_i(a).$$

Assuming positional independence, information content of the complete site is simply the sum of the information contents:

$$IC = \sum_{i=1}^{k} IC_i.$$

Information content is the basis of a convenient visualization of PSPM named sequence logos [26] that displays simultaneously conservation of each position, and their proportional base composition (see Fig. 8.1).

Information content can be generalized to account for the background model with biased base probability distribution **p**:

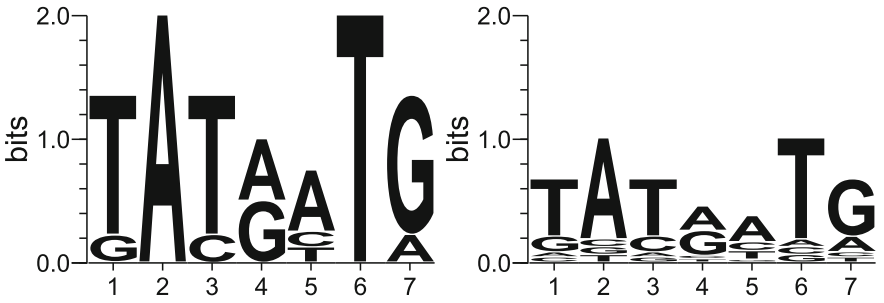

Fig. 8.1 Sequence logos for the Pribnow example (*left* without pseudocounts, *right* with pseudocounts). Height of stacks of symbols shows the information content of the position and the relative heights of the bases indicates their probability at the position (logo generated with WebLogo 3.3 [27])

$$IC_{i||\mathbf{p}} = \sum_a p_i(a) \log_2 \frac{p_i(a)}{p(a)}$$

$IC_{i||\mathbf{p}}$ is known as the relative entropy (a.k.a. Kullback-Liebler divergence [28]) and measures how much the $p_i(x)$ diverge from the background distribution $\mathbf{p}$ at the position. Let us note that when $\forall a \in \{A, C, G, T\}, p(a) = \frac{1}{4}$, the formula can be rewritten into IC_i. The generalized information content of the site is once again the sum over the positions $IC_{||\mathbf{p}} = \sum_{i=1}^{k} IC_{i||\mathbf{p}}(i)$: it measures how much the distribution defined by the PSPM contrasts with the distribution obtained by a Bernoulli-like process.

Information content is thus related to how exceptionally conserved is the set of underlying words with respect to such background models. It is thus a good objective function for PWM motif discovery programs that aim at identifying such sets of words in a set of sequences (for instance, to find binding sites near the transcription sites as in the Pribnow example). In its simplest setting, the problem can be stated as looking for a word of length k per sequence such that the corresponding information content, or a related score, is maximized. Many strategies for the exploration of the search space have been proposed. This includes the greedy algorithm consensus [29–31], expectation maximization algorithms like MEME [32] and several algorithms based on a Gibbs sampling strategy: Gibbs [33–35], AlignACE [36], MotifSampler [37] or BioProspector [38]. The scores used are information content (IC) (consensus, MotifSampler), log-likelihood ratio (LLR) (MotifSampler, Gibbs), E-value of the log-likelihood (MEME) or E-value of the IC (consensus).

Usage PWM/PSSM are widely used in popular databases such as TRANSFAC [39] and JASPAR [40] to model binding sites, identified experimentally by techniques such as SELEX or now ChIP-Seq, with the help of motif discovery programs to refine the site localization, and are then available to scan new genomes for the prediction of putative binding sites. There is still a large number of false positives, and regulation in more complex organisms than bacteria is still incompletely under-

stood. Whether those sites are actually bound by a protein and play a functional role in transcription, and under what conditions, must still be determined experimentally by traditional molecular techniques like promoter bashing, reporter gene assays, ChIP experiments, etc.

8.2.2 Longer Words

Binding sites, involved in the regulation of the transcription of DNA genes into RNA and the production of proteins, are examples of short words in DNA. Gene coding for RNA or proteins, that are the functional products of DNA in the cell, can also easily be considered as (longer) DNA words.

In the context of natural evolution, genes as well as other DNA sequences, are subject to genomic mutations (substitutions, insertions, deletions or recombinations) under natural selection pressure. Most of these mutations are lethal or harmful, but about a third of them are either neutral or weakly beneficial. There is thus a sequence conservation of the genes transmitted among the individuals or the species, but with substitutions of bases and insertions or deletions of (eventually stretches of) bases. Biologists use the term of "homologs" to designate sequences inherited in two species by a common ancestor. Homology is the base of comparative genomics to annotate the sequences that can be considered as variants of the same word. But homology does not imply necessarily that function is preserved. The TIGRFAM protein database introduced the term "equivalogs" to designate homologs that are conserved with respect to function since their last common ancestor. This later concept matches more closely the linguistic closely of a "word" (with literal or practical meaning) but is more difficult to establish, especially *in silico*.

Similarity of Two Proteins Homology of two proteins can be estimated by aligning their sequences so as to optimize the number of exact matches between aligned amino acids and by reporting the percentage of identity between the two aligned sequences. To better evaluate their functional kinship, it is better to take into account the different physico-chemical properties of the amino acids (see Fig. 8.2). For instance, if the electric charge of an amino acid is important for the function of the protein, the function is more likely to be conserved by mutations preserving this charge. In some other cases, the hydrophoby of the amino acid will be its important feature.

Substitution matrices such as Blosum62 [42] score the similarity of amino acids according to their propensity to be exchanged with each other in blocks of conserved regions (Table 8.1). Such matrices reflects the mean physico-chemical similarity between amino acids under natural selection pressure, as well as some similarity or redundancy of the genetic code.

Substitution matrices provide a way to score the similarity (instead of their percentage of identity) of two proteins by aligning their sequence of amino acids so as to maximize the sum of the amino acid substitution scores. This can be computed in quadratic time by a dynamic programming *global alignment* algorithm known as the

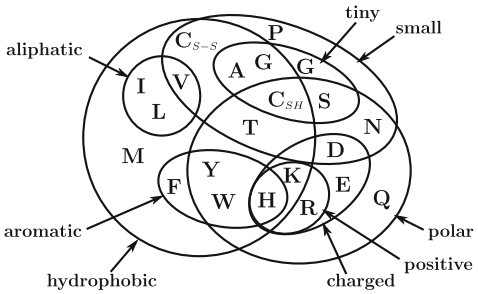

Fig. 8.2 Venn diagram of amino acid properties (adapted from: [41])

Table 8.1 BLOSUM62 substitution matrix

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	
C	**9**																				**C**
S	-1	**4**																			**S**
T	-1	1	**5**																		**T**
P	-3	-1	-1	**7**																	**P**
A	0	1	0	-1	**4**																**A**
G	-3	0	-2	-2	0	**6**															**G**
N	-3	1	0	-2	-2	0	**6**														**N**
D	-3	0	-1	-1	-2	-1	1	**6**													**D**
E	-4	0	-1	-1	-1	-2	0	2	**5**												**E**
Q	-3	0	-1	-1	-1	-2	0	0	2	**5**											**Q**
H	-3	-1	-2	-2	-2	-2	1	-1	0	0	**8**										**H**
R	-3	-1	-1	-2	-1	-2	0	-2	0	1	0	**5**									**R**
K	-3	0	-1	-1	-1	-2	0	-1	1	1	-1	2	**5**								**K**
M	-1	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	**5**							**M**
I	-1	-2	-1	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	**4**						**I**
L	-1	-2	-1	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	**4**					**L**
V	-1	-2	0	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	1	**4**				**V**
F	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	**6**			**F**
Y	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	**7**		**Y**
W	-2	-3	-2	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	**11**	**W**
	C	**S**	**T**	**P**	**A**	**G**	**N**	**D**	**E**	**Q**	**H**	**R**	**K**	**M**	**I**	**L**	**V**	**F**	**Y**	**W**	

Frequently observed substitutions receive positive scores and seldom observed substitutions are given negative scores (log odds ratio)

Needleman–Wunsch algorithm [43], that copes also with *insertions and deletions* of subsequences that are common in DNA sequences by the addition of affine penalty scores for 'gaps'.

Global alignment enables one to compare two protein sequences over their whole length, but many proteins are composed of several domains that are stable units of protein spatial structures able to fold autonomously. Domains may have existed, or may still exist, as independent proteins: they constitute the protein building blocks selected by evolution and recombined in different arrangements to create proteins with different functions. Comparing proteins at this level requires local rather than global alignments. The best *local alignment* of two sequences can be computed by the Smith–Waterman algorithm [44], a variation of the global alignment dynamic programming algorithm not penalizing gaps at both ends of the sequences. To search an entire database for homologous (sub-)sequences of a given protein sequence in

reasonable time, heuristic and approximate local alignment algorithms have been developed, such as FASTA [45] or BLAST [46], one of the most widely used bioinformatics programs.

Modeling Conserved Protein Sequences When getting more than two related protein sequences, switching from pairwise sequence alignment to *multiple sequence alignment* enables one to identify evolutionarily or structurally conserved regions and key positions in all the sequences. Most formulations of multiple sequence alignment lead to NP-complete problems; therefore, classical multiple sequence alignment programs rely on heuristics. Most of them perform global multiple sequence alignment such as ClustalW [47], T-Coffee [48], Probcons [49], MUSCLE [50] or MAFFT [51]. Local multiple sequence alignment can be found by the methods cited above to build PWM, the set of conserved k-words being a specific case of local alignment without gaps. In between global and local alignment, DIALIGN [52, 53] proposes an original approach based on significant local pairwise alignment of segments that enables it to identify a set of multiple sequence local alignments shared by all the sequences without any gap penalty.

Profile HMM Modeling locally conserved regions identified by multiple sequence alignment can be done once again with PWM. To handle larger regions with insertions and deletions, PWMs have been generalized to so-called *profile* models by the addition of insertion/deletion penalties at each position [55] and furthermore to *profile Hidden Markov Models* (pHMM) by adding also probabilities for entering into insertion, deletion or matching mode at next position given the current position and mode [56, 57]. Namely, pHMMs are hidden Markov models with a predefined specific k-position left-to-right architecture, with three (hidden) states per position (see Fig. 8.3): a *match* state generating amino acids according to the conserved position distribution (the equivalent of a PWM column), an *insert* state generating amino acids with respect to their distribution in gaps (by default, their background probability) and a *delete* silent state enabling passing a match state without emitting any amino acid.

Transitions are only allowed between states from one position to the next one and are probabilized, enabling one to tune the likelihood of inserting or deleting amino acids at each position and the likelihood of continuing insertions or deletions after entering one of these modes, as seen in protein sequence families.

If the topology of a pHMM is set, its probabilistic parameters can be estimated from available sequences of the family by a classical Expectation-Maximization scheme such as the Baum–Welch algorithm [58]. Nevertheless, the classical workflow in Bioinformatics is rather to start from a multiple sequence alignment of the sequences, assign for each column of the alignment involving enough sequences (say more than half of the family) a match state (and its insertion and deletion companion states) and convert observed counts of symbol emissions and state transitions into probabilities from the alignment.

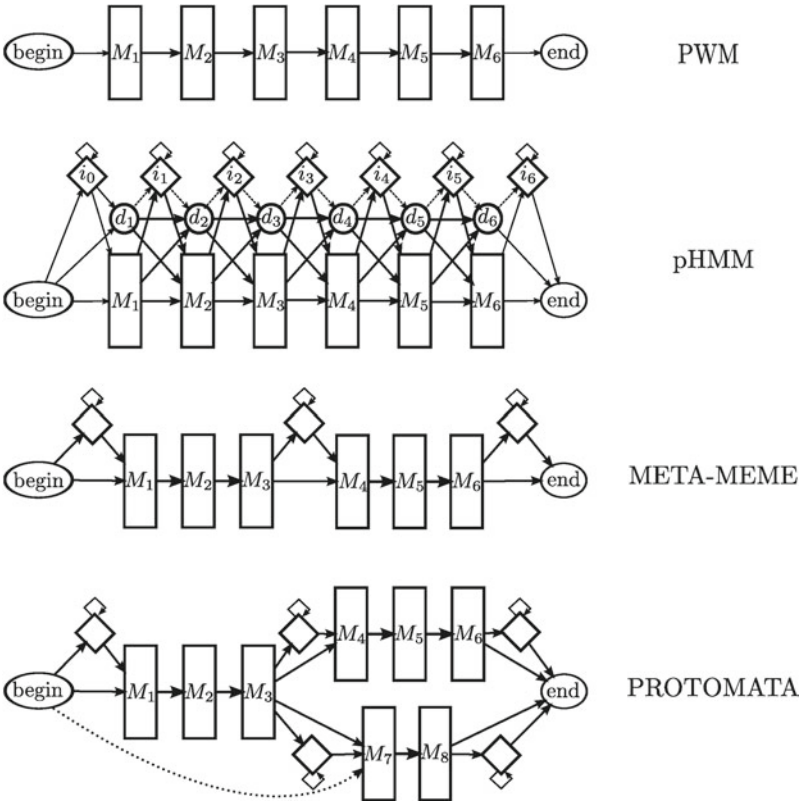

Fig. 8.3 PWM, pHMM, Meta-MEME and Protomata types of architecture (inspired from [54])

Elaborate Pseudocounts Even if the topology of pHMM is simple, the number of free parameters to estimate is still big compared to the number of sequences usually available. Much work has thus been done to avoid overspecialization and compensate for the lack of data or its biases by the development of transition regularizers, sequence weighting schemes and, especially, sophisticated pseudocount schemes based on the usage and elaboration of a priori knowledge on amino acid substitutability. As a matter of fact, the alphabet size of amino acids is greater than that of nucleotides, and targeted characterizations with pHMMs tend to be longer than with PWMs: pseudocounts are thus even more important here.

Classical pseudocounts presented above for nucleotides can be used, but taking into account the substitutability preferences of amino acids arising from their shared physico-chemical properties leads to better performances. A first way is to use available substitution matrices such as BLOSUM62: if we denote by $m(a|b)$ the probability of having a mutation to a from b, derived from the corresponding score in the chosen substitution matrix (see [42]), an intuitive scheme introduced for PWM with many variants [59] is to make each amino acid b contribute to pseudocounts of

amino acid a in proportion to its abundance at the position and its probability $m(a|b)$ of mutating into a. If we denote by $m_i(a) = \sum_b \frac{o_i(b)}{\sum_{b'} o_i(b')} m(a|b)$ the probability of getting a by mutation of residues at position i, an estimate for the probability of a in i can be:

$$\hat{p}_i(a) = \frac{o_i(a) + A\, m_i(a)}{\sum_{a'} (o_i(a') + A\, m_i(a'))}.$$

This pseudocount scheme has the advantage of interpolating between the score of pairwise alignment, such as with BLAST, when a small number of sequences is available (consider for instance the case of only one sequence and $A \gg 1$) and the maximum likelihood approach when more sequences are available (when $\sum_a o_i(a) \gg A$). In practice, A has to be chosen to tune the importance of pseudocounts with respect to observed counts, classical proposed policies being to choose $\min(20, \sum_a o_i(a))$ [60] or $5R$ [59] where R is the number of different amino acids observed in the column, a simple measure of its diversity.

This pseudocount scheme performs well but does not take full advantage of the column composition knowledge. Instead of distributing pseudocounts from each amino acid count independently, one may wish to distribute them according to the whole column distribution. For instance, if the column is biased towards small hydrophobic amino acids, one would like to bias the pseudocounts towards this combination of physico-chemical properties. To this end, [62] proposed using Dirichlet mixture densities as a means of representing prior information about typical amino acid column distributions in multiple sequence alignments and derived the formulas to compute the corresponding posterior distributions given observed counts in the Bayesian framework.

Dirichlet mixtures can be thought of as mixtures of M pseudocount vectors $\boldsymbol{\alpha_1}, \ldots, \boldsymbol{\alpha_M}$ corresponding to M different typical distributions of amino acids having each a prior probability of $q_j, 1 \leq j \leq M$, where each Dirichlet density $\boldsymbol{\alpha_j} = (\alpha_j(A), \alpha_j(C), \ldots, \alpha_j(W))$ contains the appropriate amino acid pseudocounts (the equivalent of $A\ p(a)$ or $A\, m_i(a)$ in the pseudocount formulas above) for the typical distribution j.

An example of a Dirichlet mixture from [61] is given in Table 8.2. This Dirichlet mixture and more recent ones can be found on the site of the Bioinformatics and Computational Biology group at UCSC at http://compbio.soe.ucsc.edu/dirichlets/. These mixtures were estimated by maximum likelihood inference from the columns of available large "gold standard" datasets of protein multiple alignments that are assumed to be accurate and representative.

According to the authors, the mixture shown here was one of their first really good Dirichlet mixtures. It is composed of nine components that favor each a different distribution of amino acids biased towards one or several physico-chemical properties from Fig. 8.2: for instance, Dirichlet density α_2 favors aromatic amino acids (Y, F, W, H) by assigning them higher pseudocounts (relatively to what would be expected from their background frequency; see [61] for details) while α_5 favors aliphatic or large and non-polar amino acids. The last component is specific: it favors columns with few different amino acids, with a preference for P, G, W or C, by

Table 8.2 Parameters of Blocks9, a nine components Dirichlet mixture prior [61]

j	1	2	3	4	5	6	7	8	9
$\alpha_j(A)$	0.271	0.021	0.561	0.070	0.041	0.116	0.093	0.452	0.005
$\alpha_j(C)$	0.040	0.010	0.045	0.011	0.015	0.037	0.005	0.115	0.004
$\alpha_j(D)$	0.018	0.012	0.438	0.019	0.006	0.012	0.387	0.062	0.007
$\alpha_j(E)$	0.016	0.011	0.764	0.095	0.010	0.018	0.348	0.116	0.006
$\alpha_j(F)$	0.014	0.386	0.087	0.013	0.154	0.052	0.011	0.284	0.003
$\alpha_j(G)$	0.132	0.016	0.259	0.048	0.008	0.017	0.106	0.140	0.017
$\alpha_j(H)$	0.012	0.076	0.215	0.077	0.007	0.005	0.050	0.100	0.004
$\alpha_j(I)$	0.023	0.035	0.146	0.033	0.300	0.797	0.015	0.550	0.002
$\alpha_j(K)$	0.020	0.014	0.762	0.577	0.011	0.017	0.094	0.144	0.005
$\alpha_j(L)$	0.031	0.094	0.247	0.072	0.999	0.286	0.028	0.701	0.006
$\alpha_j(M)$	0.015	0.022	0.119	0.028	0.210	0.076	0.010	0.277	0.001
$\alpha_j(N)$	0.048	0.029	0.442	0.080	0.006	0.015	0.188	0.119	0.004
$\alpha_j(P)$	0.054	0.013	0.175	0.038	0.013	0.015	0.050	0.097	0.009
$\alpha_j(Q)$	0.021	0.023	0.531	0.185	0.020	0.011	0.110	0.127	0.004
$\alpha_j(R)$	0.024	0.019	0.466	0.507	0.015	0.013	0.039	0.144	0.007
$\alpha_j(S)$	0.216	0.029	0.583	0.074	0.012	0.028	0.119	0.279	0.003
$\alpha_j(T)$	0.147	0.018	0.446	0.072	0.036	0.088	0.066	0.358	0.004
$\alpha_j(V)$	0.065	0.036	0.227	0.043	0.180	0.944	0.025	0.662	0.003
$\alpha_j(W)$	0.004	0.072	0.030	0.011	0.013	0.004	0.003	0.062	0.003
$\alpha_j(Y)$	0.010	0.420	0.121	0.029	0.026	0.017	0.019	0.199	0.003
$\sum_a \alpha_j(a)$	1.181	1.356	6.664	2.081	2.081	2.568	1.766	4.988	0.100
q_j	0.183	0.058	0.090	0.079	0.083	0.091	0.116	0.066	0.234

assigning tiny pseudocounts to all amino acids so that the observed count will dominate. This component has the highest prior probability ($q_9 = 0.234$) since many positions in alignments exhibit a unique conserved amino acid, followed by the first component ($q_1 = 0.183$) that favors small neutral amino acids that appear to be often mixed together in alignment columns, while the more specific density of the second component has the lowest prior probability of the mixture ($q_2 = 0.058$).

Basically, the Dirichlet density $\boldsymbol{\alpha_j}$ of a Dirichlet mixture component embeds a prior in the form of a pseudocount that enables one to compute the posterior probability $\hat{p}_i(a|\boldsymbol{\alpha_j})$ of each amino acid a from observed counts at position i with respect to this prior by:

$$\hat{p}_i(a|\alpha_j) = \frac{o_i(a) + \alpha_j(a)}{\sum_{a'} \left(o_i(a') + \alpha_j(a')\right)}.$$

This formula can be extended to a mixture of M Dirichlet densities $\Theta = (\boldsymbol{\alpha_1}, \ldots, \boldsymbol{\alpha_M}, q_1, \ldots, q_M)$ by distributing these probabilities proportionally to the likelihood $p_i(j)$ of each component for the observed count distribution:

$$\hat{p}_i(a|\Theta) = \sum_{j=1}^{M} p_i(j) \frac{o_i(a) + \alpha_j(a)}{\sum_{a'} \left(o_i(a') + \alpha_j(a')\right)}.$$

$p_i(j)$ is named the *posterior mixture coefficient* of component j and can be estimated by application of Bayes rule from the *prior Dirichlet mixture coefficient* q_j and the likelihood of the observed counts for component j determined by density $\boldsymbol{\alpha_j}$:

$$\hat{p}_i(j) = \frac{q_j \; p(\mathbf{o_i}|\boldsymbol{\alpha_j})}{\sum_{j'=1}^{M} q_{j'} \; p(\mathbf{o_i}|\boldsymbol{\alpha_{j'}})}$$

where $p(\mathbf{o_i}|\boldsymbol{\alpha_j})$, the likelihood of the observed counts according to Dirichlet density $\boldsymbol{\alpha_j}$, is given by the complicated but simple to calculate formula

$$p(\mathbf{o_i}|\boldsymbol{\alpha_j}) = \frac{\left(\sum_a o_i(a)\right)!}{\prod_a o_i(a)!} \cdot \frac{\prod_a \Gamma(o_i(a) + \alpha_j(a))}{\Gamma(\sum_a o_i(a) + \alpha_j(a))} \cdot \frac{\Gamma(\sum_a \alpha_j(a))}{\prod_a \Gamma(\alpha_j(a))}$$

where $\Gamma(x)$, the gamma function, is the standard continuous generalization of the integer factorial function.

These formulas obtained by Bayesian inference provide a powerful pseudocount scheme to estimate the distribution at a position from a small number of observation counts and priors on different typical column amino acid distributions. From more than hundred sequences required to build a good characterization of a family of homologous sequences, one comes down to fifty sequences, or even as few as ten or twenty examples with the latest pseudocount schemes.

Usage Profile HMMs have thus become a method of choice for the classification and the annotations of homologous protein sequences. Instead of using BLAST to search in a database of annotated sequences for one homolog to the sequence to annotate, the idea is to build first a pHMM for each family of homologous sequences and then to predict to which family the sequence belongs by testing which pHMM recognize it. This way, information from the whole family, rather than from only one sequence, can be used for more sensitive annotation. The most popular pHMM packages are HMMER (pronounced hammer) [54] and SAM [63]. The HMMER package is used in particular in the PFAM [64, 65] and TIGRFAM [66] databases gathering alignments and pHMM signatures for domains and proteins that are widely used by biologists for the annotation of new sequenced genomes. The SAM package is more directed towards the recognition of a remote homolog sharing a common structural fold: it was applied to search for protein structure templates in several structure prediction competitions CASP [67] and it is used by the SUPERFAMILY [68] library of profile hidden Markov models that represent all proteins of known 3D structure.

Thanks to the work done to require fewer and fewer examples by the incorporation of a priori knowledge on the similarity of homologous sequences, the recent trend has been to build a pHMM starting from only one proteic sequence as initiated

by PSI-BLAST with PWM [69] to provide a more sensitive alternative to BLAST. Starting from a unique query sequence, the strategy is to bootstrap the search with close homologs: a pHMM is built from the query sequence and then progressively refined by searching and including iteratively the most significant sequence matches in comprehensive sequence databases such as UniProt [70] or the non-redundant (nr) database from NCBI [71]. The result of this procedure is a sensitive pHMM and the retrieved homologous sequences to the query. This strategy was used by SAM-T98 and its successor SAM-T2K for the CASP competitions [72–74]. pHMM packages implementing this strategy with fast heuristic prefilters, such as in the new HMMER3 [75], are now as fast as BLAST. The idea has been pushed one step further by HHSearch [76] and its filtered speeded-up version HHBlits [77] that preprocess the sequences from the databases to group them in sets of close homologs represented by a pHMM and perform then iterative pHMM-pHMM alignments to obtain more sensitive results for the search of remote homologs sharing the same structural fold, helped by sequence context-specific pseudocounts.

Modeling Conserved RNA Sequences Profile HMMs have been especially successful for modeling protein homologs and they are also starting to be used for modeling DNA homologs [78, 79]. However, they are not adapted so well for modeling RNA not translated into proteins. These so-called non-coding RNA (ncRNA) molecules play vital roles in many cellular processes. One of the best known examples of functional ncRNA is the family of transfer RNAs (tRNA) that is central for the synthesis of proteins. A tRNA molecule is shown in Fig. 8.4: one can see from this example that, like proteins, RNAs are single-strand molecules that fold into a three-dimensional structure ("tertiary structure") that determines the function, and, as in DNA, the complementarity between the bases (A–U and C–G) is a key determinant of RNA structure that is typically composed of short helices packed together and is often simply represented by the base pairing on the sequence ("secondary structure").

The contiguous paired bases that form the helices, named stems, predominantly occur in a nested fashion in the RNA sequences as complementary palindromic subsequences. These kinds of long-distance correlations in the sequence that are crucial for RNA structure are typically context-free and lie beyond the expressiveness of pHMMs that are restricted to position-based characterizations.

RNA and Context-Free Grammars In Fig. 8.5, an example is given of how a context-free Grammar can be designed in a straightforward way to capture the non crossing base pairing. The idea is to have a pair matching rule $S_i \rightarrow aS_{i+1}b$ for each paired base (a, b) and a base matching rule of the form $S_i \rightarrow aS_{i+1}$ or $S_i \rightarrow S_{i+1}a$ for each unpaired base a. By ordering this rule with respect to sequence order and introducing a branching rule $B_i \rightarrow S_i S_j$ to chain successive nested structures, one gets a grammar recognizing the RNA sequence with a derivation tree mirroring its secondary structure.

The secondary structure is often more conserved than the sequence of non-coding RNAs: mutations in one strand of a stem are often compensated for by a mutation in the complementary strand. These compensatory mutations restore base pairing at

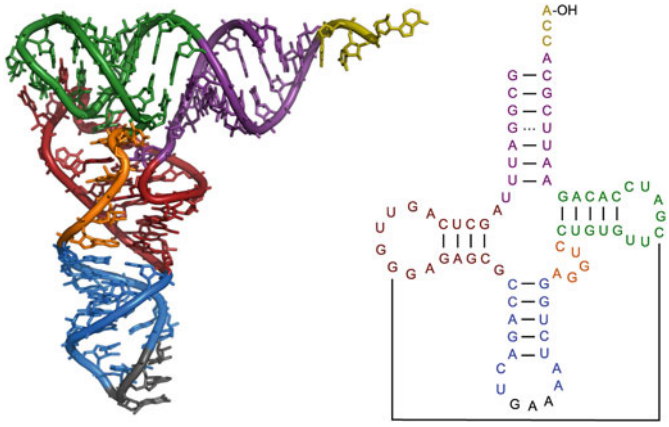

Fig. 8.4 Tertiary (*left*) and secondary (*right*) structure of yeast tRNA-Phe

Grammar $G = \langle \Sigma = \{A, C, G, U\}, N = \{S_1 \ldots S_{17}, B_1, \}, S_1, P \rangle$, s.t. the production rules in P are:

$S_1 \to \mathrm{A}S_2$, $S_2 \to \mathrm{A}B_1$, $B_1 \to S_3 S_{11}$, $S_3 \to S_4\mathrm{U}$, $S_4 \to \mathrm{G}S_5\mathrm{C}$, $S_5 \to \mathrm{A}S_6\mathrm{U}$, $S_6 \to \mathrm{C}S_7\mathrm{G}$, $S_7 \to \mathrm{U}S_8$, $S_8 \to S_9\mathrm{G}$, $S_9 \to \mathrm{U}S_{10}$, $S_{10} \to \mathrm{C}$, $S_{11} \to \mathrm{G}S_{12}\mathrm{C}$, $S_{12} \to \mathrm{G}S_{13}\mathrm{C}$, $S_{13} \to \mathrm{C}S_{14}$, $S_{14} \to \mathrm{G}S_{15}\mathrm{C}$, $S_{15} \to S_{16}\mathrm{A}$, $S_{16} \to \mathrm{A}S_{17}$, $S_{17} \to \mathrm{C}$

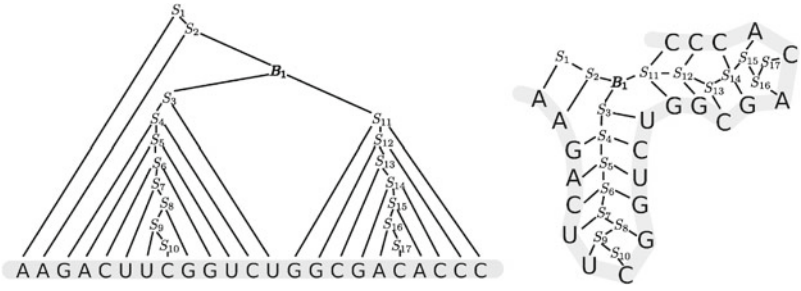

Fig. 8.5 Example of context-free grammar and derivation tree mirroring the secondary structure of an RNA sequence

a position and contribute to the conservation of the RNA secondary structure and therefore its function. Let us remark here that single mutations can also occur on non-paired bases without changing the secondary structures. The grammar above can easily be generalized to cope with these kinds of mutations, preserving the structure that the sequence can undergo. To do so, each pair matching rule can be complemented to match the other complementary pairs of bases and each single base matching rule can also be complemented to match the other bases.

For instance, in the example of Fig. 8.5, $S_4 \to \mathrm{G}S_5\mathrm{C}$ would be complemented to get a rule $S_4 \to aS_5\tilde{a}$ for each pair of bases $(a, \tilde{a})$, where $\tilde{a}$ denotes the complementary base to a, and $S_1 \to \mathrm{A}S_2$ would be complemented to get a rule $S_1 \to aS_2$ for each base a; and so on for the other matching rules.

Profile SCFG By doing so, the resulting grammar would model the secondary structure and would lose the information of the initial RNA sequence even if this can be important for homology search or functional characterization. A trade-off between sequence and secondary structure conservation can be achieved by weighting differently each base or pair of bases matched by each rule according to its probability of occurring at the position. At this point, the obtained grammar could be seen as a stochastic context-free counterpart of the (regular) PWMs seen above, allowing us to match a base a at one position i with weight $w_{i,a}$ as with a PWM by a base matching rule $S_i \rightarrow aS_{i+1}/w_{i,a}$, but allowing us also to match paired bases $(a, \tilde{a})$ at paired positions (i, j) with a weight $w_{i,(a,\tilde{a})}$ by a pair matching rule $S_i \rightarrow aS_{i+1}\tilde{a}/w_{i,(a,\tilde{a})}$. To obtain the context-free counterpart of pHMM, named *profile stochastic context-free grammars* (pSCFG) [81] or *covariance models* (CM) [82], each matching rule S_i is completed with position-based deletion rules (of the form $S_i \rightarrow S_{i+1}/w_i^{del}$) and insertion rules (of the form $I_i \rightarrow aI_i/w_{i,a}^{ins}$ or $I_i \rightarrow I_i a/w_{i,a}^{ins}$). For positions matching one base, this is done as for pHMM. For positions matching paired bases, deletion and insertion rules are added in a similar way but taking care to enable insertion or deletion on each side (left or right) of the nested sequence, which requires the equivalent of six states instead of three by position.

As with pHMMs, pSCFG's parameters can be trained by likelihood maximization approaches from a set of aligned sequences, but this requires additionally an RNA consensus (nested) secondary structure indicating the paired bases and the unpaired bases to set up the topology. This secondary structure can be known for one of the aligned sequences, be predicted by free energy minimization on a sequence or be the inferred common secondary structure from a set of multiple, homologous sequences. In Fig. 8.6, an example of three aligned RNA sequences with such a secondary structure is given with nested '>' and '<' indicating the paired positions, 'x' the unpaired positions and '.' the insertions with respect to the structure. From this information, one can automatically only keep the matching positions sufficiently shared among the sequence to get the paired ('>', '<') and unpaired ('x') matching positions of the pSCFG corresponding to the template secondary structure displayed on the left of Fig. 8.6. Each matching position is systematically completed with its companion insertion/deletion rules to get the complete pSCFG topology and parameters can

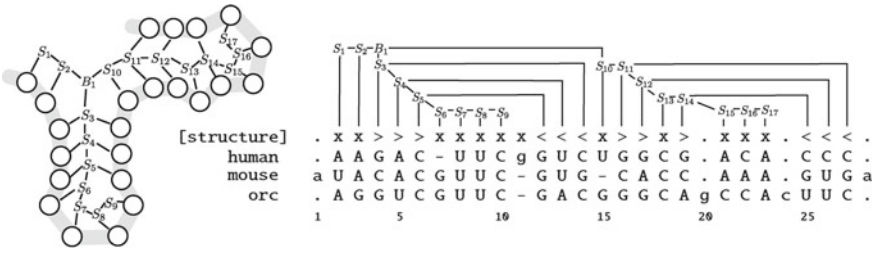

Fig. 8.6 Setting pSCFG's topology from multiple sequence alignment annotated by a secondary structure (example adapted from [80])

then be trained to maximize the likelihood of the alignment, eventually completed by pseudocounts.

Usage By using a context-free representation, PCFGs and CM extend pHMMs nicely to handle not only the base distribution at each position but also the pairs of base distribution at each (nested) paired position, capturing this way an important structural feature of ncRNA sequences that make it suitable to retrieve successfully RNA homologs. The Rfam database [83] that is an authoritative collection of non-coding RNA families represents each family by a multiple sequence alignment, predicted secondary structure and CM, and is powered by Infernal [84], the kinship software package to HMMER dedicated to modeling RNA with CM.

To get finer results on the characterization of ncRNA, one would need to be able to represent also cross-correlations such as pseudoknots (typical RNA structures with two stems in which half of one stem is intercalated between the two halves of another stem), which with all the computational hardness that they involve, is beyond the generative power of context-free grammars. Even if some proposals have been made to represent this kind of struture by grammatical models [85–88], learning such models will be extremely difficult. Finding good representations with practical computational time for learning that kind of correlation on genomic sequences is still an open and challenging research area.

Towards Sentences So far, we have seen approaches modeling homologous proteins or RNA genes in their maximal alignable length. To find more distant homologs or to focus on functionally important parts of the sequences, other approaches prefer to target the identification and the characterization of the most conserved parts shared by a set of sequences.

For instance, Meta-MEME [89] is based on an iterative search by MEME of a set of significant local alignments on a set of DNA or proteic sequences [32] that are used to build a simplified profile HMM where all the delete states are removed and only the insertion states between each block modeling a local alignment found are kept (see Fig. 8.3).

Pratt [90] searches for even more strict conservation: instead of local alignments on all the sequences, it searches by enumeration for interspaced strictly conserved amino acid or nucleotide symbols occurring in a sufficiently large subset of the sequences and then refines heuristically these patterns with new matching components offering a choice between sets of symbols. The patterns potentially returned by Pratt are composed of a suite of symbols or choice of symbols separated by wildcards indicating an insertion of a stretch of symbols bounded by a minimal and a maximal length. To remain feasible, the search has to be constrained by many user-defined parameters limiting the size of the pattern and the number of insertions, the program returning then the best patterns in this search space with respect to an information base or a minimum description length score.

An example of a well-known pattern is the C2H2 signature of 'zinc finger' in proteins: C-x(2,4)-C-x(3)-[LIVMFYWC]-x(8)-H-x(3,5)-H, read as a C followed by two to four amino acids, then a C followed by three amino acids and then one of

the amino acid chosen in [LIVMFYWC] followed by eight amino acids, an H, three to five amino acids and finally an H. These patterns are among the most expressive patterns used in Bioinformatics and can be seen as the deterministic counterpart of the Meta-MEME models, with blocks arising from exact conservation rather than from local similarity. They are known as Prosite's patterns from the name of the database [91] that popularized them as exact signatures of many domains, families and functional sites on proteins. While the patterns in Prosite were initially mostly built semi-automatically from multiple sequence alignments, Pratt is now the default pattern discovery software proposed to users on Prosite's website to find patterns without the need for a sequence alignment.

These later methods enable one to discover shorter functional or structural conserved units than genes or domains—the highly conserved blocks of Meta-MEME in all sequences or the adjacent groups of conserved positions identified by Pratt in a sequence subset—introducing each unit as a new potential genomic word or the succession of these units as a more complex, interspaced in the sequence (but eventually close in space), word.

8.3 Learning Syntax

So far, we have seen the state-of-the-art methods actually used in practice by biologists to discover and model (conserved) words in genomic sequences. The achievements in Bioinformatics for expressive characterizations are strongly linked with multiple sequence alignments, resulting in position-specific signatures that represent a suite of independent, uncorrelated conserved positions (or pairs of positions for RNA), eventually augmented with the ability to insert symbols between these positions or to skip some of them. Learning is then based on (1) the choice by the expert of the most adequate simple topology, (2) the identification and alignment of the conserved positions among the sequences and, for stochastic models, (3) training the parameters to maximize the likelihood of the sample with respect to priors.

In this section, we are interested in overtaking the position-specific characterization of (conserved) words. In particular, we would like to learn models with dependencies between the symbols of the sequences. In other words, this would allow us to make progress towards the goal of learning not only the words but also the syntax (the grammar) of genomic sequences. The difficulty is that, with dependencies being unknown, one cannot then cannot anymore rely on predefined topologies such as the pHMMs or pSCFGs: the structure of the grammar has to be learnt from the sample, which constitutes a complete Grammatical Inference task and a challenging application for that field.

Learning k-Testable Languages A first step towards learning grammars on genomic sequences is the early work of Yokomori et al. [92, 93] on learning automata representing locally k-testable languages applied to the identification of hemoglobin α-chains. The class of locally k-testable languages, very similar to the class of k-testable languages in the strict sense [94, 95], is linked to *n-grams* and, more

Dayhoff's coding

AminoAcids	Properties	Symbol
C	Sulfurpoly merization	a
G, S, T, A, P	Small	b
D, E, N, Q	Acid and amide	c
R, H, K	Basic	d
L, V, M, I	Hydrophobic	e
Y, F, W	Aromatic	f

Binary coding

Amino Acids	Hydrophoby index	Symbol
A, C, F, G, I, L, M, N, S, T, V, W, Y	High	0
D, E, H, K, P, Q, R	Low	1

Fig. 8.7 Dayhoff's and binary amino acid encodings used in [92, 93, 96, 97]

biologically, to (persistent) splicing systems. Languages of this class have the property that it is sufficient to parse the substrings of length k to decide whether a sequence is accepted or not; dependencies are therefore limited to the length k but cover all the length of the sequences in contrast to motifs. Given k, learning such a language can be done by a simple efficient algorithm building an automaton memorizing the subwords of length k appearing in the positive sample and the corresponding one-letter admissible transitions between them. This algorithm ensures identification in the limit of k-testable languages when k is known. In practice, however, the value of k is estimated by cross-validation and is usually small, the inference being then less subject to over-specialization. To apply this simple inference algorithm to proteins, Yokomori et al. reduce the 20 letter alphabet to a six letter alphabet, clustering amino acids according to main substitutability classes following Dayhoff's coding method, or drastically to a binary alphabet according to hydropathy (see Fig. 8.7). Recoding the sequences with these reduced alphabets help greatly the generalization and enables us to bootstrap the inference by some biological knowledge on amino acids similarities.

This first work is the root of recent studies applying similar approaches to learn grammatical models for the prediction of coiled-coil proteins [96] and transmembrane regions in proteins [97], whose performances are close to those of dedicated tools built with human expertise. In these works, the application scope of learning a k-testable language is extended from a sequence classification to a sequence labeling task through preliminary sequence recoding and automata to transducer post transformation. Sequence recoding is done first by reducing the alphabet according to Dayhoff's code as in [92, 93] but the alphabet is hereafter augmented by combining letters of the reduced alphabet with their label in the labeled sequences forming the training sample for the task. For instance, using an example from [97], one protein sequence of the training set,

M R V T A P R T L L L L L W G A V A L T E T W A G S H S M R,

would be encoded first following Dayhoff's coding into

```
e d e b b b d b e e e e e f b b e b e b c b f b b b d b e d
```

and, from its known transmembrane topology, this sequence could be labeled as follows (see [97] for alternative labeling):

```
e d e b b b d b e e e e e f b b e b e b c b f b b b d b e d
O O O C M M M M M M D I I I I I I I A M M M M M M B O O O O
```

where M labels residues in transmembrane regions, I labels residues in the cell while O labels residues out of the cell and A, B, C, D label the shift from outer/inner regions to/from transmembrane regions. Then, in the augmented alphabet, composed of a symbol xL for each letter x labeled by L, the sequence encoding the labeled example would begin by the following symbols (separated by white spaces):

```
eO dO eO bC bM bM dM bM eM eM eD eI eI fI bI bI eI bI eI bA cM...
```

By encoding the sequences from the positive sample this way, one can learn a k-testable language by a classical algorithm, such as k-TSSI, designed to learn k-testable languages in the strict sense [94, 95], with the advantage that, as in the morphic generator methodology [98], identical letters can be distinguished by their label during the inference. By transforming each transition labeled by a symbol xL from the learned automaton into a transition by letter x and output label L, one gets back a labeling transducer that can then be weighted and used for the task, eventually with the help of error correcting parsing techniques to compensate for the lack of data. These studies show that grammatical inference techniques can be applied with encouraging results to genomic sequences, even with such a limited class of languages when helped by pertinent pre- and post-processing techniques. We will now focus on learning more expressive grammatical representations of languages, and thus more complex dependencies, on these kinds of sequences.

Learning Automata At the first level of Chomsky's hierarchy (regular languages), we have investigated in our team the inference of full automata to model functional or structural families of protein directly from their complete sequence. RPNI [99, 100], EDSM and Blue-Fringe [101] (see Chap. 4, *On the Inference of Finite State Automata from Positive and Negative Data*, López and García) having been shown to be successful in practice on artificial data, testing these methods on this task was appealing. Our preliminary attempts showed also that these methods, even improved by taking into account similarities of amino acids, were performing very badly on leave-one-out experiments. Our analysis of these results yields that protein sequences whose length is about 300 symbols on average on a 20-letter alphabet, and whose functional parts are not necessarily at the beginning or the end of the sequences, are not well suited for these algorithms relying mainly on common sequence heads and tails for the inference. To avoid these pitfalls, we have proposed shifting from a deterministic to non-deterministic automata and adapt consequently the idea of evidence introduced by EDSM to merge common (similar) substrings rather than common tails, obtaining a first successful application of the classical state merging grammatical inference framework to learn automata on protein sequences: Protomata-Learner [102–107].

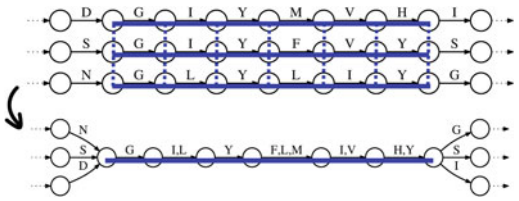

Fig. 8.8 Merging similar substrings

Shifting from a deterministic to a non-deterministic setting in the state merging approach requires simply starting from the Minimal Canonical Automaton (MCA) of the sample set (the non-deterministic automaton that is the union of the canonical automata built on each sample) rather than the Prefix Tree Acceptor (PTA) and proceeding by merging some of its states (without merging for determinisation) [108] or, inspired by EDSM, by merging successively the states on paths labeled by common substrings.

Similar Substring Merging Approach To deal with amino acid similarities, the heuristic has been generalized to look at common *similar* substrings, on the basis of the significantly similar pairs of substrings (named diagonals) precomputed by Dialign to serve as multiple sequence alignment building blocks [53]. A Dialign's diagonal d is a pair of equal-length substrings (d_1, d_2), implicitly aligned from left-to-right and whose similarity $s(d)$ is computed by summing the substitution score (given by a substitution matrix) of the aligned amino acids. Dialign computes also for each diagonal the weight of its similarity $w(d)$ as the negative logarithm of the similarity's p-value, namely of the probability of finding a diagonal of the same length with a greater or equal similarity in random amino acid sequences. The weight measuring how exceptional the similarity of the diagonal is relative to its length enables us to compare diagonal of different lengths and to define similar diagonals: random diagonals ought to have a weight of 0; similar diagonals are thus those whose weight is greater than 0, or greater than a positive weight threshold parameter t if one wants more significant similarity before considering the substrings in the diagonal.

The task is then to distinguish the similar diagonals that are characteristic of the family from those that are similar by chance or for another unrelated reason. This is done in Protomata-Learner by a best-first greedy approach: at each iteration, the best similar substrings are selected by one heuristic (maximizing their support in the training set and also their similarity) and states aligned by these substrings are merged (see Fig. 8.8), discarding from the future choices the remaining similar substrings that are incompatible with the selected ones.

Incompatible daigonals are those with an overlap presenting conflicting alignments inside the diagonals and forcing us to choose at most one of them[1] (see

[1]This corresponds to the preservation constraint from [104] forbidding us to merge together the states resulting from merging a diagonal to prevent identified conserved words from being damaged.

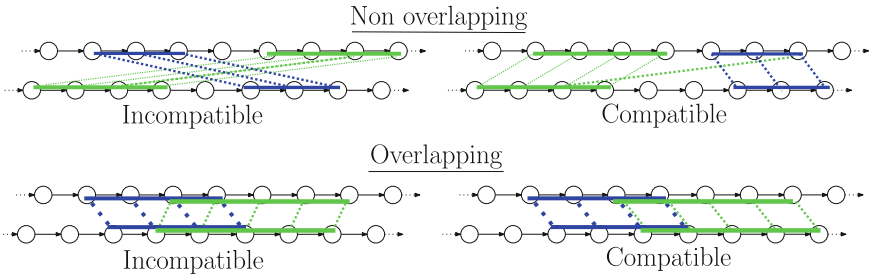

Fig. 8.9 Incompatible and compatible diagonals

Fig. 8.9 top). Another kind of incompatible diagonal can be introduced to help the inference when it is assumed that the protein sequence family does not undergo shuffling mutations (that are unlikely to occur without structure and function change): in that case, the order of the similar substrings in the sequences is preserved and crossing diagonals are incompatible (see Fig. 8.9 bottom). This greedy similar substring merging algorithm halts when no more compatible similar substrings are available for merging, relying so on incompatibilities and on the chosen threshold t to stop the inference. No negative sample is required, the characterization being directed towards maximizing the global unexpected similarity of substrings with respect to random sequences and adopting in this way a Minimum Description Length perspective rather than the discriminative Occam's razor inspiration of RPNI or EDSM.

A New Kind of Alignment The similar substring merging approach of Protomata-Learner under such incompatibility constraints can be linked to the classical Bioinformatics field by considering the sets of similar substrings merged as a new kind of multiple sequence alignment, named *partial local multiple alignment* (PLMA), exhibiting conserved regions that can be local, involving only a contiguous subset of the amino acids in the sequences as defined for classical local alignments, but also partial, involving contiguous amino acids from only a subset of the sequences instead of all the sequences. This later property enables us to represent unrelated conserved regions among subsets of the sequences: instead of being limited to the identification of conserved positions in all the sequences, one can identify alternative conserved words in some sequences, not necessarily aligned, and their chaining, paving thus the way to modeling syntax in addition to conserved words. For the inference of automata, the aligned substrings from conserved regions of the PLMA are merged, weighting eventually amino acid transitions thanks to efficient PWM or pHMM weighting schemes, and insertion states are added to link consecutive conservation regions (see Fig. 8.10), enabling learning topologies that can be seen as a generalization of pHMM or Meta-MEME architectures overtaking these position-specific characterizations by enabling us to model alternative paths (see Fig. 8.3).

Learning Context-Free Grammars Even if automata enable us to take a important step toward more expressive models, they are limited to successive short-term dependencies while it is well known that, from protein folding, residues that are far in the

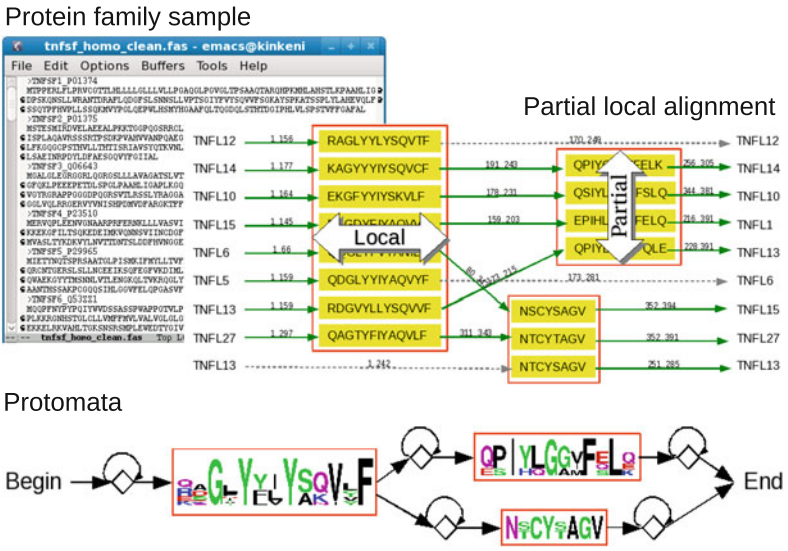

Fig. 8.10 Learning automata by partial local alignment from set of protein sequences

sequence may be close in space and interact together or are simply correlated. To represent this kind of long-distance interaction, one needs to learn more expressive grammatical representations.

From a General Template Grammar A first attempt towards this goal is the framework introduced in [109] based on a genetic algorithm training the weights of a complete stochastic context-free grammars in Chomsky's normal form to maximize the likelihood of the training sample. A complete grammar is such that the rule $A \rightarrow BC$ exists for each non-terminal A, B, C: the number of rules grows thus extremely fast with respect to the chosen number of non-terminals. The framework aims at limiting the number of non-terminals by proposing biasing the topology of the grammar towards nested dependencies and more drastically by an original way of coping with the size of the amino acid alphabet and introducing knowledge on their physico-chemical properties: all the amino acids are generated from only three non-terminals, corresponding to three discretized levels (low, medium or high level) of a chosen property of interest (for instance the van der Waals volume), the probability of generating the amino acid being fixed with respect to these levels (and thus not subject to training). Then a grammar considers amino acids only with respect to one property and if more than one property is of interest, one needs to train several grammars and to combine parsing scores for membership predictions. Experiments restricted to binding site regions of protein sequences and nine non-terminals show a good recognition accuracy on this task and pertinent parse trees illustrating the interest of this kind of context-free model.

By Local Substitutability We have recently proposed a different approach [110] showing the versatility and the efficiency of distributional learning of context-free languages (see Chap. 6, *Distributional Learning of Context-Free and Multiple Context-Free Grammars*, Clark and Yoshinaka) by applying it to protein sequences. PLMAs are used once again, but here as a pre-processing step to deal with amino acid similarity: by using parameters that allows to identify all short highly conserved regions under overlapping and crossing incompatibilities, the sequences are recoded according to these conservation blocks and provided as input for the actual generalization step performed by a grammatical inference algorithm. To be able to parse non-encoded protein sequences, a post-processing of the inferred grammar is performed to replace each terminal corresponding to a conserved region by a new non-terminal generating amino acid from the region (by introducing a succession of new non-terminals for each set of aligned amino acid from the region, in charge of generating indifferently any amino acid from the set) and introduce new non-terminals in charge of generating any amino acid for non-conserved regions. Used this way, PLMAs detect and align similar amino acids but entails almost no generalization when no grammatical inference algorithm is used, as testified by leave-one-out experiments. More surprisingly, when we tried state-of-the-art grammatical inference algorithms learning substitutable [111, 112] or k, l-substitutable [113] context-free languages, based on a formalization of substitutability idea introduced in linguistics by Zellig Harris in the 1950s [114], no additional generalization was performed.

Learning such languages is based on the identification of substrings appearing in a common context, to generalize the language by allowing these substrings to be substituted for each other (a contextual constraint for substitutability being added for k, l-substitutable language): i.e. if xyz and $xy'z$ are both in the training set, then any occurrence of y (or a subset of them for k, l-substitutable language) can be substituted by y', and vice versa, in the language. The problem in the preliminary experiments on protein sequences is that this criterion was never met in the training samples. As a matter of fact, if the sequences are long, observing a double occurrence of the common context (x, z) and a double occurrence of y, given that at least one of these substrings has to be long, has low likelihood in practice. Moreover, these characterizations rely on conserved heads and tails that, as already stated for the inference of automata, are not necessarily informative and conserved in protein sequences.

In [110], we proposed thus a variant of the substitutability generalization criterion that considers *local* rather than global context to define the substitutable substrings: local substitutability criterion states that it is sufficient to have both $xuyvz$ and $x'uy'vz'$ for a common local context (u, v) of sufficient length in the training set to allow us to substitute any occurrence of y (or a subset of them for k, l-substitutable languages) by y'. At the price of adding two additional parameters on the required left and right lengths of the common local context enabling us to define substitutability of the substrings (or only one parameter when right and left contexts are considered symmetrically), one has been able to get a real and pertinent generalization. Thanks to the development and the implementation of a faster algorithm for learning local substitutable context-free grammars, named ReGLiS, combined with the encoded pre- and post-processing scheme, these results have been confirmed on the complete

set of protein families used for the testing in [109]: using the entire protein sequences rather than only the short binding site substrings, our leave-one-out experiments show a good recall and a perfect precision [115]. These preliminary results, obtained without any weights on the rules, are really encouraging and should be easily improved. They already show, with other works presented in this section, that the application of grammatical inference can be successful for non-trivial syntactic characterizations of protein families. More generally, learning syntax on genomic sequences is a very nice open playground for grammatical inference, enabling us to apply ideas or techniques from the field but being also a source of inspiration for novel practical and theoretical challenging developments.

8.4 Conclusion

We have presented here the first successful steps towards learning the language of biological sequences. So far, the state of the art is mainly at the word level: the discovery of exceptional words, the alignment of conserved words and their modeling by the parametrization of simple adequate topologies based on biological priors. Some recent advances have also been made on learning non-trivial grammar topologies for proteins but we are only at the beginning of this exciting challenge addressed by Grammatical Inference.

To draw the lines of future research in that field, one can guess that the focus on learning topologies with (long-distance) correlations will continue. In proteins and RNA, it would allow us to capture correlations between positions that are far in the sequences but close in the 3D space. In DNA, the problem seems more complicated since the challenge is then to deal with palindromes and copies, requiring us to use and learn more expressive grammars. For DNA, recent advances have thus rather been on a simpler task: discovering the hierarchical structure of DNA as an instance of the smallest grammar problem, along the lines initiated by Sequitur [116] and its successors [117–123]. These studies have not been presented in this chapter since it is still difficult to assert and compare their biological pertinence, but these approaches based on repeats may help us to better understand what are the important words and where are their occurrences in DNA and to decipher its word structure as a preliminary step to learning grammars. Moreover, the repeats used in these approaches are not that far from the variables used in the current state-of-the-art DNA parsers of the first section. This is an interesting convergence when the goal is to design automatically, or help the expert to design, the grammars for these parsers.

We have proposed in this chapter an overview from a Grammatical Inference point of view of the achievements and open challenges in this research field as well as some keys to enter it. To further investigate this area, we propose a short list of additional reading recommendations.

Further Readings First, Wikipedia (http://www.wikipedia.org/) covers fairly well the related concepts in biology or bioinformatics and these pages are usually well written. Good entry points to Pattern Discovery are [22, 124] while [23] offers a comprehensive algorithmic and theoretical treatment of the subject. For probabilistic models on sequences, an excellent review with a grammatical inference point of view is [125], while the reference books [126, 127] contain non-grammatical machine learning techniques. Finally, on Grammatical Inference, the other chapters of this book should be helpful, as well as the reference book [128].

References

1. Beadle, G.W., Beadle, M.: The language of life: an introduction to the science of genetics. American Institute of Biological Sciences (1966)
2. Clancy, S., Brown, W.: Translation: DNA to mRNA to protein. Nature Education (2008)
3. Chomsky, N.: Syntactic Structures. Mouton (1957)
4. Searls, D.B.: The computational linguistics of biological sequences. In Hunter, L., ed.: Artificial Intelligence and Molecular Biology. AAAI Press (1993) 47–120
5. Searls, D.B.: Linguistic approaches to biological sequences. Computer Applications in the Biosciences **13** (1997) 333–344
6. Searls, D.B.: The language of genes. Nature **420** (2002) 211–217
7. Chiang, D., Joshi, A.K., Searls, D.B.: Grammatical representations of macromolecular structure. Journal of Computational Biology **13** (2006) 1077–1100
8. Searls, D.B.: A primer in macromolecular linguistics. Biopolymers **99** (2013) 203–17
9. Joshi, A.K., Weir, D.J., Vijay-Shanker, K.: The convergence of mildly context-sensitive grammar formalisms. Technical Report MS-CIS-90-01, University of Pennsylvania (1990)
10. Dong, S., Searls, D.B.: Gene structure prediction by linguistic methods. Genomics **23** (1994) 540–551
11. Nicolas, F., Rivals, E.: Hardness results for the center and median string problems under the weighted and unweighted edit distances. J. Discrete Algorithms **3** (2005) 390–415
12. Dsouza, M., Larsen, N., Overbeek, R.: Searching for patterns in genomic data. Trends in Genetics **13** (1997) 497–498
13. Pesole, G., Liuni, S., D'Souza, M.: Patsearch: a pattern matcher software that finds functional elements in nucleotide and protein sequences and assesses their statistical significance. Bioinformatics **16** (2000) 439–450
14. Belleannée, C., Sallou, O., Nicolas, J.: Logol: Expressive Pattern Matching in Sequences. Application to Ribosomal Frameshift Modeling. In Comin, M., Kall, L., Marchiori, E., Ngom, A., Rajapakse, J., eds.: PRIB2014 - Pattern Recognition in Bioinformatics, 9th IAPR International Conference. Volume 8626 of Lecture Notes in Computer Science, Stockholm, Springer (2014) 34–47
15. Macke, T.J., Ecker, D.J., Gutell, R.R., Gautheret, D., Case, D.A., Sampath, R.: Rnamotif, an RNA secondary structure definition and search algorithm. Nucleic acids research **29** (2001) 4724–4735
16. Eddy, S.: RNABOB: a program to search for RNA secondary structure motifs in sequence databases (1996)
17. Graf, S., Strothmann, D., Kurtz, S., Steger, G.: Hypalib: a database of RNAs and RNA structural elements defined by hybrid patterns. Nucleic Acids Res. **29** (2001) 196–198
18. Strothmann, D., Gräf, S.A., Kurtz, S., Steger, G.: The syntax and semantics of a language for describing complex patterns in biological sequences. Technical report, Universität Bielefeld, Technische Fakultät, Arbeitsgruppe Praktische Informatik (2000)

19. Billoud, B., Kontic, M., Viari, A.: Palingol: a declarative programming language to describe nucleic acids' secondary structures and to scan sequence database. Nucleic Acids Res **24** (1996) 395–403
20. Meyer, F., Kurtz, S., Backofen, R., Will, S., Beckstette, M.: Structator: fast index-based search for RNA sequence-structure patterns. BMC Bioinformatics **12** (2011) 214
21. Pribnow, D.: Nucleotide sequence of an RNA polymerase binding site at an early t7 promoter. Proceedings of the National Academy of Sciences of the United States of America **72** (1975) 784–8
22. van Helden, J.: The Analysis of Regulatory Sequences. In: Multiple Aspects of DNA and RNA: from Biophysics to Bioinformatics: Lecture Notes of the Les Houches Summer School 2004. Gulf Professional Publishing (2005)
23. Parida, L.: Pattern Discovery in Bioinformatics: Theory & Algorithms. Chapman & Hall/CRC (2007)
24. Stormo, G.D., Schneider, T.D., Gold, L., Ehrenfeucht, A.: Use of the "perceptron" algorithm to distinguish translational initiation sites in E. coli. Nucleic Acids Res **10** (1982) 2997–3011
25. Schneider, T.D., Stormo, G.D., Gold, L., Ehrenfeucht, A.: Information content of binding sites on nucleotide sequences. Journal of molecular biology **188** (1986) 415–31
26. Schneider, T.: Information theory primer (1995)
27. Crooks, G.E., Hon, G., Chandonia, J.M., Brenner, S.E.: Weblogo: a sequence logo generator. Genome Res **14** (2004) 1188–1190
28. Kullback, S., Leibler, R.A.: On information and sufficiency. Ann. Math. Statistics **22** (1951) 79–86
29. Hertz, G.Z., Hartzell, 3rd, G., Stormo, G.D.: Identification of consensus patterns in unaligned DNA sequences known to be functionally related. Comput Appl Biosci **6** (1990) 81–92
30. Hertz, G.Z., Stormo, G.D.: Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. Bioinformatics **15** (1999) 563–577
31. Stormo, G.D., Hartzell, 3rd, G.: Identifying protein-binding sites from unaligned DNA fragments. Proc Natl Acad Sci U S A **86** (1989) 1183–1187
32. Bailey, T.L., Elkan, C.: Fitting a mixture model by expectation maximization to discover motifs in biopolymers. Proc Int Conf Intell Syst Mol Biol **2** (1994) 28–36
33. Lawrence, C.E., Altschul, S.F., Boguski, M.S., Liu, J.S., Neuwald, A.F., Wootton, J.C.: Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. Science **262** (1993) 208–214
34. Neuwald, A.F., Liu, J.S., Lawrence, C.E.: Gibbs motif sampling: detection of bacterial outer membrane protein repeats. Protein Sci **4** (1995) 1618–1632
35. Neuwald, A.F., Liu, J.S., Lipman, D.J., Lawrence, C.E.: Extracting protein alignment models from the sequence database. Nucleic Acids Res **25** (1997) 1665–1677
36. Roth, F.P., Hughes, J.D., Estep, P.W., Church, G.M.: Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. Nat Biotechnol **16** (1998) 939–945
37. Thijs, G., Lescot, M., Marchal, K., Rombauts, S., De Moor, B., Rouzé, P., Moreau, Y.: A higher-order background model improves the detection of promoter regulatory elements by Gibbs sampling. Bioinformatics **17** (2001) 1113–1122
38. Liu, X., Brutlag, D.L., Liu, J.S.: Bioprospector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. Pac Symp Biocomput (2001) 127–138
39. Matys, V., Kel-Margoulis, O.V., Fricke, E., Liebich, I., Land, S., Barre-Dirrie, A., Reuter, I., Chekmenev, D., Krull, M., Hornischer, K., Voss, N., Stegmaier, P., Lewicki-Potapov, B., Saxel, H., Kel, A.E., Wingender, E.: TRANSFAC and its module TRANSCompel: transcriptional gene regulation in eukaryotes. Nucleic Acids Research **34** (2006) D108–D110
40. Sandelin, A., Alkema, W., Engström, P., Wasserman, W.W., Lenhard, B.: Jaspar: an open-access database for eukaryotic transcription factor binding profiles. Nucleic Acids Research **32** (2004) D91–D94
41. Taylor, W.R.: The classification of amino acid conservation. J Theor Biol **119** (1986) 205–218

42. Eddy, S.R.: Where did the BLOSUM62 alignment score matrix come from? Nat Biotechnol **22** (2004) 1035–1036
43. Needleman, S.B., Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology **48** (1970) 443–453
44. Smith, T., Waterman, M.: Identification of common molecular subsequences. Journal of Molecular Biology **147** (1981) 195–197
45. Pearson, W.R., Lipman, D.J.: Improved tools for biological sequence comparison. Proc Natl Acad Sci U S A **85** (1988) 2444–2448
46. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: A basic local alignment search tool. J. Mol. Biol. **215** (1990) 403–410
47. Thompson, J.D., Higgins, D.G., Gibson, T.J.: Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acids Res **22** (1994) 4673–4680
48. Notredame, C., Higgins, D.G., Heringa, J.: T-coffee: A novel method for fast and accurate multiple sequence alignment. J Mol Biol **302** (2000) 205–217
49. Do, C.B., Mahabhashyam, M.S.P., Brudno, M., Batzoglou, S.: Probcons: Probabilistic consistency-based multiple sequence alignment. Genome Res **15** (2005) 330–340
50. Edgar, R.C.: Muscle: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res **32** (2004) 1792–1797
51. Katoh, K., Misawa, K., Kuma, K.i., Miyata, T.: MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. Nucleic Acids Res **30** (2002) 3059–3066
52. Morgenstern, B., Frech, K., Dress, A., Werner, T.: Dialign: finding local similarities by multiple sequence alignment. Bioinformatics **14** (1998) 290–294
53. Morgenstern, B.: Dialign 2: improvement of the segment-to-segment approach to multiple sequence alignment. Bioinformatics **15** (1999) 211–218
54. Eddy, S.R.: Profile hidden markov models. Bioinformatics **14** (1998) 755–763
55. Gribskov, M., McLachlan, A.D., Eisenberg, D.: Profile analysis: detection of distantly related proteins. Proceedings of the National Academy of Sciences of the United States of America **84** (1987) 4355–8
56. Krogh, A., Brown, M., Mian, I.S., Sjölander, K., Haussler, D.: Hidden Markov models in computational biology. applications to protein modeling. Journal of molecular biology **235** (1994) 1501–31
57. Baldi, P., Chauvin, Y., Hunkapiller, T., McClure, M.A.: Hidden Markov models of biological primary sequence information. Proceedings of the National Academy of Sciences of the United States of America **91** (1994) 1059–63
58. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. In: Proceedings of the IEEE. (1989) 257–286
59. Henikoff, J.G., Henikoff, S.: Using substitution probabilities to improve position-specific scoring matrices. Computer applications in the biosciences : CABIOS **12** (1996) 135–43
60. Claverie, J.M.: Some useful statistical properties of position-weight matrices. Comput Chem **18** (1994) 287–294
61. Sjölander, K., Karplus, K., Brown, M., Hughey, R., Krogh, A., Mian, I., Haussler, D.: Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology. Computer applications in the biosciences : CABIOS **12** (1996) 327–345
62. Brown, M., Hughey, R., Krogh, A., Mian, I.S., Sjölander, K., Haussler, D.: Using Dirichlet mixture priors to derive hidden Markov models for protein families. In Hunter, L., Searls, D.B., Shavlik, J.W., eds.: Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology, Bethesda, MD, USA, July 1993, AAAI (1993) 47–55
63. Hughey, R., Krogh, A.: Hidden Markov models for sequence analysis: extension and analysis of the basic method. Comput Appl Biosci **12** (1996) 95–107
64. Sonnhammer, E.L., Eddy, S.R., Durbin, R.: Pfam: a comprehensive database of protein domain families based on seed alignments. Proteins **28** (1997) 405–420
65. Finn, R.D., Bateman, A., Clements, J., Coggill, P., Eberhardt, R.Y., Eddy, S.R., Heger, A., Hetherington, K., Holm, L., Mistry, J., Sonnhammer, E.L.L., Tate, J., Punta, M.: Pfam: the protein families database. Nucleic Acids Res (2013)

66. Haft, D.H., Selengut, J.D., Richter, R.A., Harkins, D., Basu, M.K., Beck, E.: TIGRFAMS and genome properties in 2013. Nucleic Acids Res **41** (2013) D387–D395
67. Moult, J.: A decade of CASP: progress, bottlenecks and prognosis in protein structure prediction. Curr Opin Struct Biol **15** (2005) 285–289
68. Gough, J., Karplus, K., Hughey, R., Chothia, C.: Assignment of homology to genome sequences using a library of hidden Markov models that represent all proteins of known structure. J Mol Biol **313** (2001) 903–919
69. Altschul, S.F., Madden, T.L., Schäffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J.: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res **25** (1997) 3389–3402
70. UniProt: Update on activities at the universal protein resource (UniProt) in 2013. Nucleic Acids Res **41** (2013) D43–D47
71. Pruitt, K.D., Tatusova, T., Maglott, D.R.: Ncbi reference sequence (refseq): a curated nonredundant sequence database of genomes, transcripts and proteins. Nucleic Acids Res **33** (2005) D501–D504
72. Karplus, K.: Hidden Markov models for detecting remote protein homologies. Bioinformatics **14** (1998) 846–865
73. Karplus, K., Karchin, R., Barrett, C., Tu, S., Cline, M., Diekhans, M., Grate, L., Casper, J., Hughey, R.: What is the value added by human intervention in protein structure prediction? Proteins **Suppl 5** (2001) 86–91
74. Karplus, K., Karchin, R., Draper, J., Casper, J., Mandel-Gutfreund, Y., Diekhans, M., Hughey, R.: Combining local-structure, fold-recognition, and new fold methods for protein structure prediction. Proteins **53 Suppl 6** (2003) 491–496
75. Eddy, S.R.: Accelerated profile HMM searches. PLoS Comput Biol **7** (2011) e1002195
76. Söding, J.: Protein homology detection by HMM-HMM comparison. Bioinformatics **21** (2005) 951–960
77. Remmert, M., Biegert, A., Hauser, A., Söding, J.: HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. Nat Methods **9** (2012) 173–175
78. Wheeler, T.J., Eddy, S.R.: nhmmer: DNA homology search with profile hmms. Bioinformatics **29** (2013) 2487–2489
79. Wheeler, T.J., Clements, J., Eddy, S.R., Hubley, R., Jones, T.A., Jurka, J., Smit, A.F.A., Finn, R.D.: Dfam: a database of repetitive DNA based on profile hidden markov models. Nucleic Acids Res **41** (2013) D70–D82
80. Eddy, S.R.: A memory-efficient dynamic programming algorithm for optimal alignment of a sequence to an RNA secondary structure. BMC Bioinformatics **3** (2002) 18
81. Sakakibara, Y., Brown, M., Hughey, R., Mian, I.S., Sjölander, K., Underwood, R.C., Haussler, D.: Recent methods for RNA modeling using stochastic context-free grammars. In: Proceedings of the Asilomar Conference on Combinatorial Pattern Matching, New York, NY, Springer-Verlag (1994) 289–306
82. Eddy, S.R., Durbin, R.: RNA sequence analysis using covariance models. Nucleic Acids Res **22** (1994) 2079–2088
83. Burge, S.W., Daub, J., Eberhardt, R., Tate, J., Barquist, L., Nawrocki, E.P., Eddy, S.R., Gardner, P.P., Bateman, A.: Rfam 11.0: 10 years of RNA families. Nucleic Acids Res **41** (2013) D226–D232
84. Nawrocki, E.P., Eddy, S.R.: Infernal 1.1: 100-fold faster RNA homology searches. Bioinformatics **29** (2013) 2933–2935
85. Uemura, Y., Hasegawa, A., Kobayashi, S., Yokomori, T.: Tree adjoining grammars for RNA structure prediction. Theoretical Computer Science **210** (1999) 277–303
86. Rivas, E., Eddy, S.: The language of RNA: a formal grammar that includes pseudoknots. Bioinformatics **16** (2000) 334
87. Cai, L., Malmberg, R.L., Wu, Y.: Stochastic modeling of RNA pseudoknotted structures: a grammatical approach. Bioinformatics **19 Suppl 1** (2003) i66–i73
88. Matsui, H., Sato, K., Sakakibara, Y.: Pair stochastic tree adjoining grammars for aligning and predicting pseudoknot RNA structures. Proc IEEE Comput Syst Bioinform Conf (2004) 290–299

89. Grundy, W.N., Bailey, T.L., Elkan, C.P., Baker, M.E.: Meta-meme: motif-based hidden Markov models of protein families. Comput Appl Biosci **13** (1997) 397–406
90. Jonassen, I. Collins, J., Higgins, D.: Finding flexible patterns in unaligned protein sequences. Protein Science **4** (1995) 1587–1595
91. Hulo, N., Bairoch, A., Bulliard, V., Cerutti, L., Cuche, B.A., de Castro, E., Lachaize, C., Langendijk-Genevaux, P.S., Sigrist, C.J.A.: The 20 years of PROSITE. Nucleic Acids Res **36** (2008) D245–D249
92. Yokomori, T., Ishida, N., Kobayashi, S.: Learning local languages and its application to protein α-chain identification. In: 27th Annual Hawaii International Conference on System Sciences (HICSS-27), January 4-7, 1994, Maui, Hawaii, USA, IEEE Computer Society (1994) 113–122
93. Yokomori, T., Kobayashi, S.: Learning local languages and their application to DNA sequence analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (1998) 1067–1079
94. Garcia, P., Vidal, E., Oncina, J.: Learning locally testable languages in the strict sense. In: Proceedings of the International Conference on Algorithmic Learning Theory. (1990) 325–338
95. Garcia, P., Vidal, E.: Inference of k-testable languages in the strict sense and application to syntactic pattern recognition. IEEE Trans. Pattern Anal. Mach. Intell. **12** (1990) 920–925
96. Peris, P., López, D., Campos, M., Sempere, J.M.: Protein motif prediction by grammatical inference. In Sakakibara, Y., Kobayashi, S., Sato, K., Nishino, T., Tomita, E., eds.: Ig TM. Volume 4201 of Lecture Notes in Computer Science, Springer (2006) 175–187
97. Peris, P., López, D., Campos, M.: IGTM: An algorithm to predict transmembrane domains and topology in proteins. BMC Bioinformatics **9** (2008)
98. Garcia, P., Vidal, E., Casacuberta, F.: Local languages, the succesor method, and a step towards a general methodology for the inference of regular grammars. IEEE Trans. Pattern Anal. Mach. Intell. **9** (1987) 841–845
99. Oncina, J., Garcia, P.: Inferring regular languages in polynomial update time. In: Pattern Recognition and Image Analysis. (1992) 49–61
100. Lang, K.J. In: Random DFA's can be approximately learned from sparse uniform examples. Association for Computing Machinery (1992) 45–52
101. Lang, K.J., Pearlmutter, B.A., Price, R.A.: Results of the Abbadingo One DFA learning competition and a new evidence-driven state merging algorithm. In: Proceedings of the 4th International Colloquium on Grammatical Inference. ICGI '98, London, UK, Springer-Verlag (1998) 1–12
102. Coste, F., Kerbellec, G., Idmont, B., Fredouille, D., Delamarche, C.: Apprentissage d'automates par fusions de paires de fragments significativement similaires et premières expérimentations sur les protéines MIP. In: JOBIM. (2004)
103. Coste, F., Kerbellec, G.: A similar fragments merging approach to learn automata on proteins. In Gama, J., Camacho, R., Brazdil, P., Jorge, A., Torgo, L., eds.: ECML. Volume 3720 of Lecture Notes in Computer Science., Springer (2005) 522–529
104. Coste, F., Kerbellec, G.: Learning Automata on Protein Sequences. In Denise, A., Durrens, P., Robin, S., Rocha, E., de Daruvar, A., Groppi, A., eds.: JOBIM, Bordeaux, France (2006) 199–210
105. Kerbellec, G.: Apprentissage d'automates modélisant des familles de séquences protéiques. PhD thesis, Université de Rennes 1 (2008)
106. Bretaudeau, A., Coste, F., Humily, F., Garczarek, L., Corguillé, G.L., Six, C., Ratin, M., Collin, O., Schluchter, W.M., Partensky, F.: Cyanolyase: a database of phycobilin lyase sequences, motifs and functions. Nucleic Acids Research **41** (2013) 396–401
107. Burgos, A., Coste, F., Kerbellec, G.: Learning automata on protein sequences by partial multiple sequence alignment. (in preparation)
108. Coste, F., Fredouille, D.: What is the Search Space for the Inference of Non Deterministic, Unambiguous and Deterministic Automata? Rapport de recherche RR-4907, INRIA (2003)
109. Dyrka, W., Nebel, J.C.: A stochastic context free grammar based framework for analysis of protein sequences. BMC Bioinformatics **10** (2009) 323

110. Coste, F., Garet, G., Nicolas, J.: Local Substitutability for Sequence Generalization. In Heinz, J., de la Higuera, C., Oates, T., eds.: ICGI 2012. Volume 21 of JMLR Workshop and Conference Proceedings, University of Maryland, MIT Press (2012) 97–111
111. Clark, A., Eyraud, R.: Identification in the limit of substitutable context free languages. In Jain, S., Simon, H.U., Tomita, E., eds.: Proceedings of the 16th International Conference on Algorithmic Learning Theory, Springer-Verlag (2005) 283–296
112. Clark, A., Eyraud, R.: Polynomial identification in the limit of substitutable context-free languages. Journal of Machine Learning Research **8** (2007) 1725–1745
113. Yoshinaka, R.: Identification in the limit of k, l-substitutable context-free languages. In Clark, A., Coste, F., Miclet, L., eds.: ICGI. Volume 5278 of Lecture Notes in Computer Science., Springer (2008) 266–279
114. Harris, Z.: Distributional structure. Word **10** (1954) 146–162
115. Coste, F., Garet, G., Nicolas, J.: A bottom-up efficient algorithm learning substitutable languages from positive examples. In Clark, A., Kanazawa, M., Yoshinaka, R., eds.: ICGI 2014. Volume 34 of JMLR Workshop and Conference Proceedings. (2014) 49–63
116. Nevill-Manning, C.G., Witten, I.H.: Compression and explanation using hierarchical grammars. The Computer Journal **40** (1997) 103–116
117. Cherniavsky, N., Lander, R.: Grammar-based compression of DNA sequences. In: DIMACS Working Group on the Burrows-Wheeler Transform. (2004) 21
118. Lanctot, J.K., Li, M., Yang, E.H.: Estimating DNA sequence entropy. In: ACM-SIAM Symposium on Discrete Algorithms. (2000) 409–418
119. Apostolico, A., Lonardi, S.: Off-line compression by greedy textual substitution. Proceedings of the IEEE **88** (2000) 1733–1744
120. Apostolico, A., Lonardi, S.: Compression of biological sequences by greedy off-line textual substitution. In: Data Compression Conference. (2000) 143–153
121. Nevill-Manning, C., Witten, I.: On-line and off-line heuristics for inferring hierarchies of repetitions in sequences. In: Data Compression Conference, IEEE (2000) 1745–1755
122. Carrascosa, R., Coste, F., Gallé, M., López, G.G.I.: The smallest grammar problem as constituents choice and minimal grammar parsing. Algorithms **4** (2011) 262–284
123. Carrascosa, R., Coste, F., Gallé, M., López, G.G.I.: Searching for smallest grammars on large sequences and application to DNA. J. Discrete Algorithms **11** (2012) 62–72
124. Brejova, B., Vinar, T., Li, M.: Pattern Discovery: Methods and Software. In Krawetz, S.A., Womble, D.D., eds.: Introduction to Bioinformatics. Humana Press (2003) 491–522
125. Sakakibara, Y.: Grammatical inference in bioinformatics. IEEE Trans. Pattern Anal. Mach. Intell. **27** (2005) 1051–1062
126. Durbin, R., Eddy, S., Krogh, A., Mitchison, G.: Biological Sequence Analysis : Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press (1999)
127. Baldi, P., Brunak, S.: Bioinformatics: The Machine Learning Approach. 2nd edn. Cambridge: MIT Press (2001)
128. de la Higuera, C.: Grammatical Inference: Learning Automata and Grammars. Cambridge University Press, (2010)