Thesis for the Degree of Licentiate of Philosophy

Modeling of bacterial DNA patterns important in horizontal gene transfer using stochastic grammars

Mariana Buongermino Pereira

Division of Mathematical Statistics
Department of Mathematical Sciences
Chalmers University of Technology and University of Gothenburg
Göteborg, Sweden 2015

Modeling of bacterial DNA patterns important in horizontal gene transfer using stochastic grammars
Mariana Buongermino Pereira

Department of Mathematical Sciences
Chalmers University of Technology and University of Gothenburg
SE-412 96 GÖTEBORG, Sweden
Phone: +46 (0)31 772 3558

Author e-mail: mariana.pereira@chalmers.se

Typeset with LaTeX.
Department of Mathematical Sciences
Printed in Göteborg, Sweden 2015

To my parents

Modeling of bacterial DNA patterns important in horizontal gene transfer using stochastic grammars

Mariana Buongermino Pereira

Department of Mathematical Sciences
Chalmers University of Technology and University of Gothenburg

Abstract

DNA contains genes which carry the blueprints for all processes necessary to maintain life. In addition to genes, DNA also contains a wide range of functional patterns, which governs many of these processes. These functional patterns have typically a high variability, both within and between species, which makes them hard to detect. Stochastic models, such as hidden Markov models and conditional random fields, offer flexible frameworks that can be used to describe these patterns, their variability and dependencies. In this thesis, we describe two such models for identification of *attC* sites, patterns necessary for the sharing of genes between bacteria, in a process known as horizontal gene transfer. Acquired genes causing bacteria to become resistant to antibiotics are often associated with *attC* sites, which make their identification highly relevant.

In the first paper we develop a stochastic regular grammar defined by an eight-state generalized hidden Markov model that describes the sequence conservation and length distribution of the different parts of an *attC* site. The different model assumptions were evaluated and improved using cross-validation experiments, which resulted in a high sensitivity in detecting *attC* sites. The model was applied to a real dataset in the form of a well-studied plasmid and was able to find the majority of the present *attC* sites. In addition, six metagenomic samples from polluted and pristine environments were analyzed. The model predicted a 15-fold higher abundance of *attC* sites in the polluted environments compared to the pristine ones. The model implementation, HattCI, was done in R and is freely available at `http://bioinformatics.math.chalmers.se/HattCI`.

AttC sites fold into a three-dimensional structure that is crucial for the horizontal transfer of genes. In the second paper, we extend our previous model to include specific information about this folding. We develop a stochastic context-free grammar, which is suited to describe the nested dependencies induced by the structure. The grammar includes features that describe thermodynamic properties of the folding. The model is formulated in the framework of conditional random fields, with parameter estimation done numerically using structured support vector machines. A first implementation of the model has been completed; further experiments, such as evaluation of the performance using cross-validation is planned.

This thesis demonstrates the flexibility of stochastic grammars for modelling the variability and dependencies in DNA patterns. It also emphasizes the value of the use of stochastic methods in the field of microbiology and infectious diseases.

Keywords: Stochastic context-free grammars, hidden Markov model, conditional random fields, integrons, *attC* sites, secondary structure.

List of appended papers

Paper I **Pereira, M.B.**, Kristiansson, E., Axelson-Fisk, M. HattCI: *attC* site identification using hidden Markov models. *Manuscript*, 2015.

Paper II **Pereira, M.B.**, Sato, K., Kristiansson, E., Axelson-Fisk, M. Improved identification of *attC* sites by secondary structure modeling. *Manuscript*, 2015.

Publications not included in this thesis:

Boulund, F., Johnning, A., **Pereira, M.B.**, Larsson, D.G., Kristiansson E. A novel method to discover fluoroquinolone antibiotic resistance (qnr) genes in fragmented nucleotide sequences. *BMC Genomics*, 13(695), 2012.

Pereira, M.B., Verma, C.S., Fuentes, G. Differences in the binding affinities of ErbB family: heterogeneity in the prediction of resistance mutants. *PLoS One*, 8(10), 2013.

Acknowledgments

First, I would like to thank you my supervisor, Marina Axelson-Fisk, for the supervision, kindness and interesting discussions about our models and life in general. I thank Erik Kristiansson, my co-supervisor, for believing in me when I could not, which alone would be sufficient to be thankful for, but I am also thankful for your brilliant and exciting ideas, very important discussions and contagious excitement about science! A special thank you to Olle Nerman for being my examiner and supporting me in the dark moments with your wise advices.

Also, I would like to thank you, Olle Häggström, for giving us the course I requested in Markov Random Fields where I learnt the theoretical basis for this thesis. Many thanks to you, Serik Sagitov, for lecturing a nice course in stochastic process and for your infinite patience with me, a non-mathematician. Thank you, Sergey Zuyev, for great lectures in the foundations of probability theory and many conversations about music, photography and other forms of art. Also, thank you, Marija Cvijovic, for encouraging me to go abroad and explore new horizons. Thank you, Lotta Fernström and Cecilia Gelin, for all the support with grants, travels and everything else.

Thank you, Sato-sensei, for your supervision in Japan teaching me the methods used in the second project presented here. Also, thank you my Japanese colleagues for being so welcoming and sharing your wonderful culture with me. Especially, Sae, Mariko and Mayumi. どうもありがとうございます.

Many thanks to my Ph.D. student colleagues and friends; you make the department a colorful place, where I feel at home. Especially, thank you, Viktor, for all the many hours spent talking about the meaning of life. Fredrik and Anna J, for being awesome. Emma, for your wonderful sense of humor. Tobias Ö and Anna L, for caring. Also, thank you, Malin, for our girl's talk. Vera, for the piano classes, the relaxing ping-pong and the many hours we studied together. Alexey, for all the shared teaching, it has been fun (!) and conversations that always make me a bit more knowledgeable. Thank you, Medhi, for worrying about my stress. Thank you Matteo, Magnus Ö, Henrike, Roza, Dmitry, Claes, Ivar, Fanny, Anders S, Sandra and Jonatan for many lunch and *fika* breaks. Thank you also Anton, Magnus R and Jose, who have finished their Ph.D.s, I miss you guys a lot.

Thank you my friends outside academia, for making me remember that there is a life outside the department of Mathematical Sciences. Especially, Juna and Mikael for always being good friends. Thank you my parents, Regina and Carlos, and sister, Priscilla, for even far away being always so supportive and present. *Saudades*!

Mari Pereira
Gothenburg, June 17, 2015

Important abbreviations

CRF	Conditional random field
gHMM	generalized hidden Markov model
HGT	Horizontal gene transfer
HMM	Hidden Markov model
SCFG	Stochastic context-free grammar
sSVM	Structured support vector machine

Contents

1 Introduction

Biological sequences

Life depends on biological macromolecules to exist. These macromolecules carry on small tasks that together compose the beautiful mosaic that enables life to happen. Examples of these tasks are production of energy, respiration, and damage repair on the cellular level. The most important of those macromolecules are DNA, RNA and proteins because the proteins are involved in all chemical reactions necessary for these tasks to happen, and DNA and RNA carry the information necessary for its own replication and for protein production. DNA, RNA and protein are long chains of building blocks and therefore called biological sequences.

DNA, RNA and proteins

DNA is the essential molecule of life. It is a long double-stranded polymer found inside the cellular nucleus. It encodes all the information necessary for life to develop and function. This information is organized in genes which are stretches of the DNA molecule. Each gene is translated into proteins that are responsible for the activity in the cells of living organisms. Examples of what proteins represent to life are vast; they are enzymes that catalyze different reactions in organisms such as digestion, synthesis of new proteins, copy of DNA; they also have structural functions, forming cells and extracellular material necessary for support and elasticity, and they are signals that control physiological reactions (Alberts et al., 2009).

The theory that describes the information flow that governs the process of synthetizing proteins is known as the central dogma (Figure 1). According to it, the information encoded in a gene in the DNA sequence is *transcribed* into RNA that in turn is *translated* into a protein. DNA is said to be transcribed into RNA because they are made of the same type of monomers; i.e. nucleotides. There are, however, two differences between DNA and RNA. The first difference is chemical. DNA is made of four nucleotides: A (adenine), C (cytosine), G (guanine) and T (thymine). In RNA molecules the three first nucleotides can be found, but instead of T there is U (uracil). The second difference is structural, DNA is made of two strands that form a double helix maintained by hydrogen bonds between the nucleotides, where A pairs with T and C with G. RNA is, on the other hand, single-stranded, but because of the chemical properties of the nucleotides, hydrogen bonds between them can still exist, which often leads the RNA single strand to fold forming a secondary structure. RNA is said to be translated into proteins because proteins are made of a different type of monomers; i.e. amino-acids. Three nucleotides are needed to code for one amino acid. There are 20 amino acids but 64 combinations of

nucleotide triplets, thus clearly more than one combination of nucleotides can result in the same amino acid.

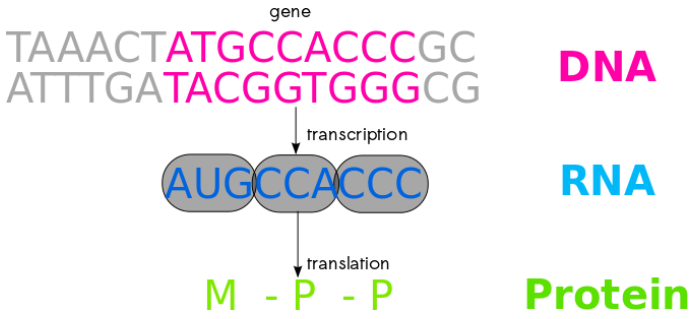

Figure 1: The central dogma. Information carried in the DNA is passed into RNA through transcription and then translated into protein.

In addition to encoding for proteins, DNA sequences contain other types of information, specific motifs where proteins can bind in order to carry on their tasks. For instance, promoters are motifs where transcription factors bind to initiate the gene expression. In the present work, we are interested in another such motif, the *attC* site, which are carried by bacteria and identified by a protein to insert and remove genes into the DNA. The use of *attC* sites is one possible mechanism used by bacteria to exchange genetic material, a process known as horizontal gene transfer.

Integrons and *attC* sites

Bacteria are capable of changing their genetic material in other ways than reproduction through a process known as *horizontal gene transfer* (HGT). One common mechanism of HGT is the acquisition of exogenous mobile genes into the genome via genetic elements known as *integrons* (Figure 2). Integrons (Mazel, 2006) are found in different bacterial species, but share common structures. They all have a gene that codes for an *integrase*, which is an enzyme that mediates the transfer process, a recombination site *attI* used by the integrase during the transfer process, and one common promoter that regulates the expression all incorporated genes. The mobile genes are organized in gene cassettes which are sequentially incorporated downstream of the promoter. Each cassette contains one gene and another recombination site required for the transfer, the *attC* site. Because one *attC* site usually accompanies each mobile gene, the study of *attC* sites can be used to identify mobile genes in the genome. Since many integron-mediated mobile genes confer antibiotic resistance, their identification can, ultimately, improve our understanding of the development and spread of antibiotic resistance.

AttC sites are imperfect reverse palindromes that fold its bottom DNA strand into a hairpin-like secondary structure, similar to RNA secondary structure. This structure is

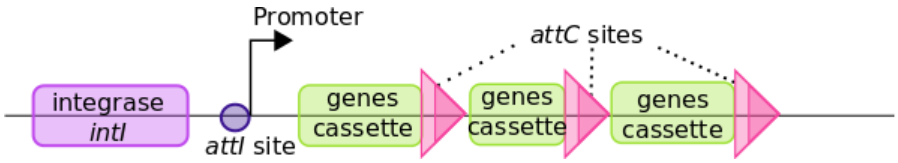

Figure 2: Integrons are genetic elements present in the bacterial genome that facilitates the horizontal transfer of genes. All integrons share a common structure: it contains one integrase, which is the enzyme responsible for mediating the gene incorporation, a promoter that regulates the expression of the incorporated genes, a recombination site *attI* used by the transfer process, and a number of gene cassettes, each of them with one inserted gene and one *attC* site recombination site required by the transfer process.

supported by hydrogen bonds formed between the pairs $\{(A-T),(G-C),(G-T)\}$. *AttC* sites also have two pairs of complementary motifs called R"/R'and L"/L'(Stokes et al., 1997) that are recognized by the integrase (Hall et al., 1991) (Figure 3). See introduction of Paper I for a detailed description of the different parts of the *attC* site (Pereira et al., 2015).

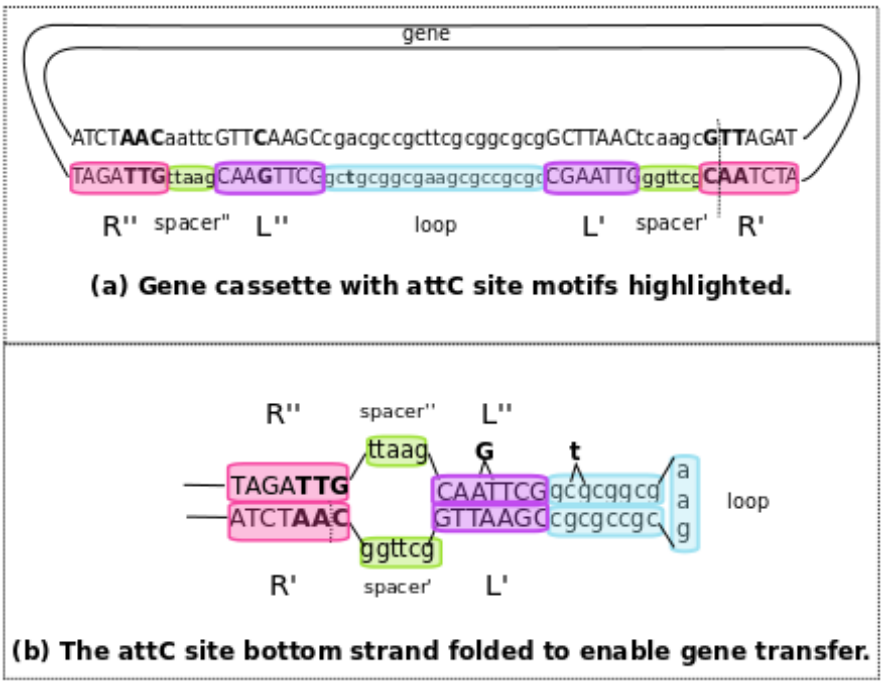

Figure 3: Example of an *attC* site in a (a) gene cassette and (b) folded. Recombination position is indicated by the horizontal line in R".

Stochastic grammars for identification of motifs on DNA sequences

Different motifs found in DNA sequences often show low conservation across different species. This low conservation implies a large variability in sequence composition that can be well addressed by stochastic models. If we think of biological sequences as sentences of a language with its own grammatical rules, stochastic grammars provide a suitable mean to model the motifs we find in this language.

During the 1950's, Chomsky was interested in the question whether a certain sentence grammatically belonged to a language or not. By grammatically he meant that the sentence had a correct grammar structure, even if the sentence itself was meaningless. The famous example is "colorless green ideas sleep furiously", which is grammatically correct but has no meaning. In order to answer the question if a sentence belonged to a certain grammar, Chomsky developed formal "machines", called transformational grammars, that could generate infinitely many grammatically correct sentences of a language (Chomsky, 1956, 1959).

A *transformational grammar* is a system of rules needed to generate sentences in a given language. More formally, we assume that an *alphabet* is a finite set of *terminal symbols*, where the symbols are some kind of observed entity (such as letters, nucleotides, words etc.), a sentence is a string of symbols, and a *language* is a set of strings over that alphabet. In order to be able to decide whether a sentence belongs to a given language or not, we need a set of *rewriting rules*. These rules make use of abstract *non-terminal symbols*, that are transformed into other terminal or non-terminal symbols while the sentence is being generated. The allowed symbols and the corresponding rules form the *grammar*.

More formally, let a sentence in any language take values in a set of terminal symbols $\{a_i\}_{i=1}^{M}$. Let the non-terminal symbols $\{W_i\}_{i=1}^{N}$ be those used in the process of generating a sentence. Let α and β be strings of both terminal and non-terminal symbols. We let a set of rewriting rules $\alpha \to \beta$ coordinate how the non-terminal symbols are transformed into terminal symbols and thus how sentences are generated. We define a *transformational grammar* as the set of terminal and non-terminal symbols together with the rewriting rules in a language.

The rewriting rules impose restrictions on the flexibility of the grammar. This means that the more complex the rules in a grammar the more complex are the dependencies it can describe in the language. Thus, according to the restrictions imposed by the rules, grammars are classified into hierarchical classes with increasing complexity (Figure 4). This theory is known as the *Chomsky hierarchy of transformational grammars* (Durbin et al., 1998).

Following the definition of a transformational grammar, let a be any terminal symbol, W any non-terminal symbol, β any string except the null string and α and γ any string including the null one. The first and simplest class of grammar is the *regular grammar*, whose rewriting rules transform the non-terminal symbols into one terminal and one non-

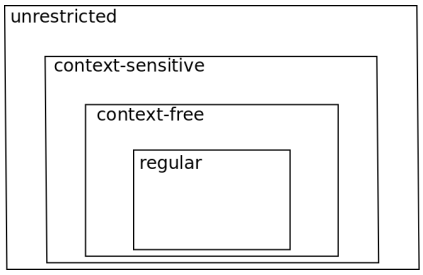

Figure 4: The Chomsky hierarchy of transformational grammars. The restriction imposed by the grammar rules define the level of dependencies they can generate: less restrictions translate into more dependencies while they require more computer power to be parsed (Figure from Durbin et al. (1998)).

terminal symbol independently of any other symbol present in the sentence. In other words, regular grammars only allow rules of the type $W \to aW$ or $W \to a$. For instance, a regular grammar for binary strings would consist of a non-terminal symbol W, terminal symbols $\{0, 1\}$, and the rewriting rules $\{W \to 0W, W \to 1W, W \to 0, W \to 1\}$. As an example, the string '01101'would then be generated by the derivation

$$W \to 0W \to 01W \to 011W \to 0110W \to 01101W \to 01101.$$

On the next level there is the *context-free grammar* where transformations are still done independently of the symbols already present in the sequence. However, here strings are generated by rules that transform the non-terminal symbols into a non-empty string, which can contain any combination of terminal and non-terminal symbols. That is, the rewriting rules take the form $W \to \beta$ where β is a string of terminal and/or non-terminal symbols, allowing for symbols to be generated simultaneously and therefore dependently. Following the binary strings example, a context-free grammar for this type of string could contain rules of the form: $\{W \to 0W0, W \to 1W1, W \to 00, W \to 11\}$. The string '110011'would be generated by the derivation

$$W \to 1W1 \to 11W11 \to 110W011 \to 110011.$$

This example illustrates how efficiently context-free grammars can describe palindromes.

Third, we have *context-sensitive grammars*, where the generated symbols now may depend on the surrounding strings. Thus, with a rewriting rule of the form $\{\alpha_1 W \alpha_2 \to \alpha_1 \beta \alpha_2\}$ the generated string β, consisting of both terminal and non-terminal symbols, depends on the surrounding context strings α_1 and α_2. Finally, we have the *unrestricted grammars* that include all the previous grammars, and have no restrictions on either side of the generated string. In essence it is like the context-sensitive grammars, but the string generated can

be null as well. Thus, rewriting rules are of the form $\{\alpha_1 W \alpha_2 \rightarrow \gamma\}$, where γ can be any combination of terminal and non-terminal symbols or the null string.

Given a specific grammar, the problem of determining whether a certain string belongs to the corresponding language is called *parsing*. Parsing is done by dividing the string into subunits (e.g. words in a natural language or motifs in biological sequences) and in this way checking if the string conforms to the grammar rules. In order to parse a sentence, the specific grammar is modelled by an *automaton*, which can be seen as a self-operating sequence-generating machine, jumping between the grammar non-terminal symbols, and producing an output along the way. The non-terminal symbols would in this case be the grammatical subunits, and the outputs the observed string of terminal symbols. The different levels of grammars described above are modeled and parsed by automata of various levels of sophistication. In an increasing complexity order, first we have *finite state automata* corresponding to regular grammars. This kind of automaton reads each symbol of a sequence at a time and accepts or rejects it according to the grammar rules. If one symbol is rejected the sequence does not belong to that grammar and the parsing stops, if all symbols are accepted the sequence belongs to the grammar. Next, *push-down automata* correspond to context-free grammars. Similarly to finite state automata, push-down automata read sequences from left to right, but instead of keeping only the current non-terminal symbol in memory, push-down automata require a stack of symbols to be in the memory, where these symbols are both non-terminal and terminal ones. The right-handed symbols a transformational rule is added to the stack. In a rule of the form $W \rightarrow \alpha_1 W \alpha_2$, we have $\alpha_1 W \alpha_2$ in the stack. The input sequence is read, if it conforms α_1, α_1 is removed from the stack, W generates a new part of string, which is added to the stack, and α_2 remains in the stack until no non-terminal symbols are left. Then the remaining of the input sequence is checked to conform the stack. *Linear bounded automata* parse context-sensitive grammars and *Turing machines* are necessary for parsing unrestricted grammars. The description of these last two automata is beyond the scope of this work, but it is worth saying that Turing machines are not guaranteed to determine if a sequence belongs a grammar or not in a finite time. The grammars, their rules and corresponding parsing automata are summarized in Table 1. Note that the more complex and flexible the grammar is, the more computer power it requires to be parsed.

The grammars described so far are deterministic. By adding probability measures to their rewriting rules, the grammars become stochastic, which is desirable when we deal with strings with great variability such as in biological sequences. For reasons discussed below, this work focuses on the use of stochastic regular grammars and stochastic context-free grammars for the study and parsing of biological sequences.

Table 1: Transformational grammars, their rewriting rules and the automaton required to parse each of them.

Grammar	Rule	Parsing automaton
Regular	$W \to aW \vert a$	Finite state automaton
Context-free	$W \to \beta$	Push-down automaton
Context-sensitive	$\alpha_1 W \alpha_2 \to \alpha_1 \beta \alpha_2$	Linear bounded automaton
Unrestricted	$\alpha_1 W \alpha_2 \to \gamma$	Turing machine

Here W is a non-terminal symbol, a a terminal symbol, α and γ strings of both terminal and non-terminal symbols including the null string, and β is any string except the null one.

Regular grammars and hidden Markov models

Regular grammars generate strings by using rules of the form $W \to aW$, where W is a non-terminal and a is a terminal symbol. When we assign probabilities to the rules in a grammar of this form, we obtain a *hidden Markov model* (HMM). An HMM is a stochastic process made of two interrelated processes: the first process, the *hidden process* Y, is a Markov chain that jumps between a set of states, or in the grammar nomenclature a set of non-terminal symbols, and obeys the *Markov property*, according to which a stochastic process $\{Y_1, \ldots, Y_T\}$ is said to be Markov if given the present, past and future are conditionally independent, i.e. $P(Y_t|Y_{t-1}, Y_{t-2}, \ldots, Y_1) = P(Y_t|Y_{t-1})$.

The second process, the *observed process* X, depends on the hidden process to generate, or *emit*, an observation given the current state. The second process is not necessarily Markov, and is seen by the observer. In grammar terminology the rewriting rules correspond to the state transitions, and the generation of terminal symbols to the emissions of the observed process. Moreover, in terms of parsing an observed sequence with the use of an automaton, the observed process is referred to as the *input sequence*, and the hidden process as the *parse sequence*, aka state sequence, that results from the parsing. In the study of biological sequences, DNA, RNA or proteins are the observed sequences and the states in the parse sequence correspond to different components of the biological sequence, such as genes, *attC* sites or other relevant motifs.

HMMs model the joint probability $P(X, Y)$ of a parse sequence Y and an observed sequence X. It describes how the hidden process Y jumps between states in a state space S, and emits symbols to the observed sequence X along the way. It is parameterized by the transition probabilities, the initial probabilities and the emission probabilities. The transition probabilities $a_{ij} = P(Y_t = j|Y_{t-1} = i)$, $i, j \in S$ define the probabilities of the hidden process jumping from state i to j, and the initial probabilities $\pi_i = P(Y_1 = i)$ for starting at state j. The emission probabilities $b_j(X_t|X_1^{t-1}) = P(X_t|X_1^{t-1}, Y_t = j)$ establish the probabilities of a state j emitting an observation X_t given, if no restrictions

are imposed, all the past observed sequence. Using these parameters the joint probability $P(X, Y)$ of the hidden and the observed process is given by

$$P(X, Y) = \prod_{t=1}^{T} a_{Y_{t-1}Y_t} b_{Y_t}(X_t | X_1^t), \tag{1.1}$$

where the initial probability π_j is rewritten as a_{Y_0,Y_1} for convenience.

HMMs are *generative models*, since they describe the joint probability of observed and hidden process. Generative models are capable of classifying data and also generating new data once the model is trained. In contrast, *discriminative models* describe the conditional probability of the hidden process given the observed data. Discriminative models classify the hidden process based on the observed one but are not capable of generating new data.

As regular grammars, HMMs have limited power to deal with long-range dependencies between states or non-terminal symbols. In particular, there are two cases that HMMs cannot efficiently describe:

(a) *Palindromic language*, a language that can be read forward or backwards, so that emissions happen in pairs. In the case of DNA, this means that emissions happen as pairs of complementary nucleotides (Figure 5(a)).

(b) *Copy language*, a language that contains copies of itself, such as repeats in DNA (Figure 5(b)).

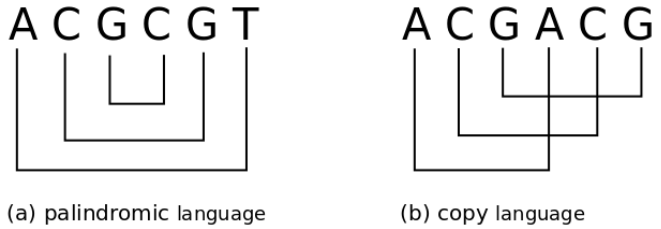

Figure 5: Examples of the two cases that cannot be efficiently modeled by regular grammars: (a) palindromic and (b) copy language. Lines indicate dependencies.

In fact, regular grammars can generate copy or palindromic languages. It cannot, however, generate *only* those and therefore cannot efficiently distinguish between them and a regular language.

Stochastic context-free grammars and conditional random fields

Stochastic context-free grammars (SCFG) extend context-free grammars in the same way HMMs extend regular grammars, by assigning probabilities to the rewriting rules. Recall that the rewriting rules of context-free grammars are on the form $W \rightarrow \beta$, where W is a non-terminal symbol and β any string of both terminal and non-terminal symbols. For our application to the study of secondary structures of DNA sequences, where three possible pairs $\{(A,T),(C,G),(G,T)\}$ need to be described, the SCFG can have rewriting rules of the form

$$W \rightarrow aWt|cWg|gWc|tWa|aW|cW|gW|tW|at|cg|gc|ta|a|c|g|t.$$

In this way, SCFGs describe a complementary palindromic language by allowing for the creation of nested, long-distant, pairwise correlations between terminal symbols, as the ones existing in secondary structures of DNA and RNA sequences. Thus, SCGFs can be used to model the joint probability of observed DNA/RNA sequences and their secondary structures.

While HMMs model the joint probability of the hidden and the observed sequence, it is interesting to model the conditional probability of the hidden secondary structure given the observed biological sequence. Conditional models avoid the difficulty to model the distribution of all possible observed sequences, which is required in the model of the joint probability, but which is not needed for the classification tasks carried on in the parsing anyway. As a result, a conditional model has the flexibility to include information of different natures in it. Conditional random fields (CRF) model a conditional probability, while they also allow for all kinds of dependencies between the non-terminal symbols, or states, and not only in the chain-like manner that the HMMs allow for. The CRF freedom to include dependencies provides an adequate framework for dealing with nested, long-distant, pairwise dependencies in the observed sequence as the ones described in a SCFG.

Let G be an undirected *graph* where a set of random variables $\{Y_1,\ldots,Y_T\}$ are the *vertices* of the graph. We say that two vertices are *neighbors* when there is an edge connecting then. For an undirected graph, the Markov property is generalized to a *spatial Markov property*, which says that given its neighbors, a vertex Y_j is independent of the other vertices in G, i.e. $p(Y_j|Y_i, i \neq j) = p(Y_j|Y_i, j \sim i)$, where $j \sim i$ indicates that j and i are neighbors. If exists another set of random variables X, such that conditioned in X, the random variables Y follow the spatial Markov property, we call (X,Y) a *conditional random field*.

More formally, let G be an undirected graph over Y, such that Y is indexed by the vertices of G. Then (X,Y) is a *conditional random field* if, when conditioned on X, the random variables Y follow the spatial Markov property with respect to G, i.e. $p(Y_j|X, Y_i, j \neq i) =$

$p(Y_j|X, Y_i, j \sim i)$, where $j \sim i$ indicates that j and i are neighbors in G.

Let each rewriting rule in the SCFG correspond to one CRF feature function, which typically is an indicator function of the presence of the rule in the observed and parse sequences. Let $F_k(X, Y)$ indicates how many times the k-th rule appears in the observed and the parse sequences (X,Y). Each feature has one parameter, a feature weight θ_k, associated with it. A CRF assumes that the logarithm of the conditional probability, $logp(Y|X)$ is a linear function of X, such that

$$P(Y|X) = \frac{\exp(\sum_{k=1}^{K} \theta_k \cdot F_k(X, Y))}{\sum_Y \exp(\sum_{k=1}^{K} \theta_k \cdot F_k(X, Y))} \tag{1.2}$$

where the denominator is known as normalization function and its sum is over all possible parse sequences Y. In our applications, X is the observed biological sequence and Y the parse sequence.

To summarize, SCFGs are appropriate models for dealing with palindromic languages such as the one in the secondary structure of biological sequences, and CRFs provide a framework for modeling the conditional probability of the secondary structure given the biological sequence. The advantage of this approach is that, compared to HMMs, the parameters are flexible to incorporate of different types of data into the model. On the downside, the parameter estimation can be quite computationally expensive.

2 Summary of appended papers

In this section, the two appended papers are introduced in the context of this thesis.

Paper I – HattCI: *attC* site identification using hidden Markov models

In paper I we present HattCI, a hidden Markov model (HMM) for *attC* site identification. The model is an eight-state generalized hidden Markov model (gHMM), with one state for non-attC site regions of DNA and seven states to describe the different parts of an *attC* site. Out of those seven, four states have a fixed length and correspond to conserved regions of the *attC* site, while the other three states are modeled to have variable lengths that follow an empirical distribution. These variable length states correspond to two short spacers and one longer central loop that separate the conserved motifs.

The model has been applied to three different datasets: one controlled dataset, where the true *attC* sites are known, and two datasets representing examples of applications, one in a bacterial plasmid and the other in a set of six metagenomic samples. For the controlled dataset, a manually curated database was prepared and the model was tested using a two-fold cross-validation scheme. The sensitivity, measuring the number of *attC* sites correctly predicted, was 94.8%. The false positive rate, measuring the number of false hits found per megabase, was estimated using reshuffled bacterial DNA (where no *attC* sites should be found) and resulted in less than 1.46 hits per megabase of the analyzed genome sequence. In the first application, the model was able to detect the majority of the *attC* sites in a well-annotated bacterial plasmid. In the final application, we compared six metagenomic samples from both pristine and potentially polluted environments. The model was able to detect a 15-fold more *attC* sites in the polluted samples compared to the pristine ones. This is in line with the theory that polluted environments can contain higher levels of integrons. Our results demonstrates that the developed model can efficiently identify *attC* sites and provide a useful tool for the identification of mobile genes, which include many of the genes responsible for the development of antibiotic resistant bacteria.

Paper II – Improved identification of *attC* sites by secondary structure modeling

Paper II, extends the model presented in paper I by enabling the prediction of the *attC* site secondary structure. This secondary structure is required for the gene transfer to happen and supported by the complementary palindromic nature of its sequence. Palindromic features nested dependencies that can stretch over long distances, something that hidden Markov models (HMMs) cannot efficiently capture. Such dependencies are adequately described by stochastic context free grammars (SCGFs) which are included as features of a conditional random field (CRF) and extended to include thermodynamic model characteristics. Using this framework, we extend the model presented in paper I by including one meta-state that includes the fold information and retain the linearity of the of overall model. In CRFs, the parameters do not have analytical maximum likelihood estimates, and need to be estimated from numerical methods. In paper II, the parameter estimation was done using structured support vector machines in the form of stochastic subgradient descent, which updates the parameters online for each training instance presented in a random order. In this case, parameter training is done by predicting the parse sequence from the current parameters and updating those according to a 0-1 error between predictions and training data.

3 Future work

Paper I – HattCI: *attC* site identification using hidden Markov models

The project presented in Paper I, is planned to be extended in four major ways. First, a comparison to existing integron/gene cassettes prediction methods will be added. The methods to be included are ACID (Joss et al., 2009), a heuristic method to score integron features and Attacca (Tsafnat et al., 2009), a deterministic model using context-sensitive grammars. Second, HattCI, which is currently implemented in R, will be implemented in C to speed up its analysis. Third, we intend to analyze a larger set of plasmids, in order to annotate the *attC* sites present there and hopefully identify novel gene cassettes. Fourth, HattCI will be used to analyze more metagenomic samples in order to quantify relative antibiotic resistance in different environments. In addition, the upstream region of the predicted *attC* sites can be BLASTed against databases to retrieve the function of the gene that accompany the *attC* sites. Moreover, HattCI can be used to map the *attC* site presence across different species, which can lead to a better understanding of the antibiotic resistance spread.

Paper II – Improved identification of *attC* sites by secondary structure modeling

Paper II is an ongoing work. The model has not been fully implemented yet. In order to be completed, multi-branch loops will be included in the model. Moreover, the implementation that is currently done in R will be translated into C for better performance, which allows for a faster and more robust cross-validation to be performed. The cross-validation will be done using the same dataset used in paper I, but with annotation that includes the secondary structure of *attC* sites, which is given by the ViennaRNA package (Lorenz et al., 2011). Next, the validated and completed model will be applied to the analysis of different datasets such as plasmids and metagenomic samples, as in paper I. The main difference between the models in paper I and II is that the second method can predict both location and the secondary structure of *attC* sites, while the first predicts only location. This difference indicates that each model may have specific applications depending on what is intended. Thus, experiments will be conducted to compare the performance of the CRF and the HMM methods for the identification of *attC* sites. When this done, applications to the analysis of different datasets will be directed to the most suitable of our methods.

References

Alberts, B., Bray, D., Hopkin, K., Johnson, A. D., Lewis, J., Raff, M., Roberts, K., and Walter, P. *Essential Cell Biology*. Garland Science, 3rd edition, 2009. ISBN 9780815341291.

Chomsky, N. Three models for the description of language. *IRE Transactions on Information Theory*, 2:113–124, 1956. doi: 10.1109/TIT.1956.1056813.

Chomsky, N. On certain formal properties of grammars. *Information and Control*, 2(2): 137–67, 1959.

Durbin, R., Eddy, S. R., Krogh, A., and Mitchison, G. *Biological sequence analysis - Probabilistic models of proteins and nucleic acids*. Cambridge University Press, 1998. ISBN 9780521629713.

Hall, R., Brookes, D., and Stokes, H. Site-specific insertion of genes into integrons: role of the 59-base element and determination of the recombination cross-over point. *Mol. Microbiol.*, 5(8):1941–1959, 1991.

Joss, M. J., Koenig, J. E., Labbate, M., Polz, M. F., Gillings, M. R., Stokes, H. W., Doolittle, W. F., and Boucher, Y. ACID: annotation of cassette and integron data. *BMC Bioinformatics*, 10(118), 2009. doi: 10.1186/1471-2105-10-118.

Lorenz, R., Bernhart, H., Siederdissen, C., Tafer, H., Flamm, C., Stadler, P., and Hofacker, I. ViennaRNA package 2.0. *Algorithms for Molecular Biology*, 6(26), 2011. doi: 10.1186/1748-7188-6-26.

Mazel, D. Integrons: agents of bacterial evolution. *Nat. Rev. Microbiol.*, 4(8):608–620, 2006. doi: 10.1038/nrmicro1462. Review.

Pereira, M., Kristiansson, E., and Axelson-Fisk, M. HattCI: attC sites identification using hidden Markov models. *In preparation*, 2015.

Stokes, H., O'Gorman, D., Recchia, G., Parsekhian, M., and Hall, R. Structure and function of 59-base element recombination sites associated with mobile gene cassettes. *Mol. Microbiol.*, 26(4):731–745, 1997.

Tsafnat, G., Coiera, E., Partridge, S. R., Schaeffer, J., and Iredell, J. R. Context-driven discovery of gene cassettes in mobile integrons using a computational grammar. *BMC Bioinformatics*, 10(281), 2009. doi: 10.1186/1471-2105-10-281.