# Extended SCFGs for LTR Identification

M.L. Souza
University of California Berkeley
Biophysics

August 22, 2011

We consider slightly extended stochastic context-free grammars for use in parsing languages of limited repeats.

## 1 Overview

Background

## 2 Extending SCFGs

Consider grammar $G = (N,\ S,\ T,\ P,\ \psi)$
where $N$ are non-terminal symbols, $S \in N$ the start symbol, $T$ terminals, $P$ a set of production rules, and $\psi$ the probability distribution over the production rules $P$.

### 2.1 Production Rules

In the following, let $n, m \in T$ be a terminal symbols, and $x,\ x' \in T^*$ be strings.
We form a superset of "RNA normal form" for SCFGs (Reference? Durbin?):

1. Bifurcation: $L \to R\ M$
2. Pass-through: $L \to R$
3. Left emission: $L \to nR$
4. Right emission: $L \to Rn$
5. Paired emission: $L \to mRn$
6. Terminal emission: $L \to n$
7. Null emission: $L \to \epsilon$

Adding an additional rule:

8. Repeat emission: $L \to R_{\text{rep}}(M)$

Where $R_{\text{rep}}(M)$ is a distinguished non-terminal in which: $L \to R_{\text{rep}}(M) \Leftrightarrow L \to xMx'$
With $x,\ x' \in T^*$, and $x'$ an approximate-repeat of $x$, to be made more precise below.

### 2.1.1 Repeat Emissions

Non-terminals such as $R_{\text{rep}}(M)$ are an embedded constrained linear indexed grammar defined by the following rules:

1. $R_{\text{rep}} \to X[]$
2. $X[\sigma] \to nX[\sigma\ n]$
3. $X[\sigma] \to X'[\sigma]$
4. $X'[\sigma n] \to X'[\sigma]m$
5. $X'[] \to M$

The above rules define a grammar capable of generating the language:

$$L_{\text{rep}} = \left\{ xmx' \mid x, x', m \in T^* \right\}$$

With $|w| = |w'|$ and each terminal $w_i$ dictating the probability of emitting symbol $w'_i$ for $0 \leq i \leq |w|$, and m denoting the substring generated by the non-terminal $M$.
I.e. it generates repetitions of precisely the same length with pointwise mutations.

We will show that parsing of an extended SCFG having repeat emissions as above can be performed in $\mathrm{O}(N^4)$ time, where $N$ is the length of the input string.

We can extend the repeat grammar to allow insertions and deletions by introducing the following production rules:

6. $X'[\sigma n] \to X'[\sigma]$ (Popping a symbol off the stack; corresponds to a deletion)
7. $X'[\sigma] \to X'[\sigma]m$ (A right-emission without stack modification; corresponds to an insertion.)

An important property of this grammar is that the growing stack for non-terminal $X$ is exactly the substring which the grammar emitted.

## 2.2 Parsing: Recursive Definition

We now consider an extension of the CYK algorithm to determine the maximum-likelihood parse for a given input string $s$.
We first give a recursive definition for each element of matrix $C \in \mathbb{R}^{|s|} \times \mathbb{R}^{|s|} \times N$.

Let $C(i, j-i+1, S) = \max_{\substack{\text{parse trees } \pi \\ \text{deriving } x_{i\cdots j}}} P(\pi)$ be defined as follows:

$$C(i, j-i+1, S) = \max \begin{cases} \max_{R,M} \max_{0 \le k \le j-i} C(i,k,R)C(i+k, j-k, M)P(L \to RM) \\ \max_R C(i, j-i, R)P(L \to R) \\ \max_R P(L \to nR)C(i+1, j-(i+1), R) \\ \max_R C(i, j-1, R)P(L \to Rn) \\ \max_R P(L \to nRm)C(i+1, j-(i+2), R) \\ P(L \to x_i) \\ P(L \to \epsilon) \\ f(i,j) \text{ defined below} \end{cases}$$

We consider $f(i,j)$ for two cases, with and without rules 6 & 7 of the repeat SLIG defined above.

### 2.2.1 Mutations, Insertions, and Deletions

Let $x_{i\cdots j}$ be a substring of $x$ partitioned into three parts of arbitrary length: (Diagram would be nice here.)

$$x_{i\cdots j} = x_{\text{lhs}} x_{\text{mid}} x_{\text{rhs}}$$

With $x_{\text{lhs}} = x_{i\cdots l-1}$, $x_{\text{mid}} = x_{l\cdots k-1}$, and $x_{\text{rhs}} = x_{k\cdots j}$.

We seek to calculate the probability of generating $x_{\text{lhs}}, x_{\text{rhs}}$ with the embedded SLIG, while generating $x_{\text{mid}}$ with its containing SCFG.

The probability of the SLIG non-terminal $R_{\text{rep}}$ generating $x_{\text{lhs}}$ and the containing SCFG non-terminal $M$ generating $x_{\text{mid}}$ are:

$$P(R_{\text{rep}} \Rightarrow x_{\text{lhs}}) = \prod_{m=i}^{l-1} P(X[\sigma] \to x_m X[\sigma x_m])$$

$$P(M \Rightarrow x_{\text{mid}}) = P(X'[] \to M)C(l, k-l, M)$$

To calculate $P(X'[x_{\text{lhs}}] \Rightarrow x_{\text{rhs}})$, the probability of generating $x_{\text{rhs}}$ allowing SLIG rules 4, 6, and 7 can be thought of as an alignment problem. I.e. we can interpret $P(X'[\sigma x_m] \to X'[\sigma]x_n)$ as a match/mismatch score to emit $x_n$ given $x_m$, $P(X'[\sigma] \to X'[\sigma]x_n)$ as insertion score to insert $x_n$, and $P(X'[\sigma x_m] \to X'[\sigma])$ as deletion score to omit $x_m$ from the right.

Suppose for each $(i,j)$ we construct a matrix $A$ such that $A(k,l) = \max P(x_{\text{lhs}} \text{ aligns to } x_{\text{rhs}})$. Then take:

$$A(k,l) = \max \begin{cases} A(k-1, l-1)P(X'[\sigma x_k] \to X'[\sigma]x_l) \quad \text{(match/mismatch)} \\ A(k-1, l)P(X'[\sigma] \to X'[\sigma]x_l) \quad \text{(insertion)} \\ A(k, l-1)P(X'[\sigma x_k] \to X'[\sigma]) \quad \text{(deletion)} \end{cases}$$

So that:

$$f(i,j) = \max_{M} \max_{k,l} P(R_{\text{rep}} \Rightarrow x_{\text{lhs}}) P(M \Rightarrow x_{\text{mid}}) A(k,l).$$

#### 2.2.2 Mutations Only

The grammar of only pointwise mutations is that the repetitive strings are necessarily the same length, removing a single degree of freedom and reducing time complexity to $O(N^4)$.
Let $c = \lfloor \frac{i+j}{2} \rfloor$, and:

$$LHS = \prod_{l=i}^{c-k-1} P(X[\sigma] \to x_l X[\sigma x_l])$$

$$RHS = \prod_{l=i}^{c+k+1} P(X'[\sigma x_l] \to X'[\sigma] x_l)$$

then:

$$f(i,j) = \max_{M} \max_{0 \le k \le \lfloor \frac{j-i}{2} \rfloor} (LHS) C(c-k, k, M)(RHS) P(L \to R_{\text{rep}}(M)) P(X'[] \to M) P(X[\sigma] \to X'[\sigma])$$

### 2.3 Improving Efficiency with Constraints (Hints)

### 2.4 Training

#### 2.4.1 Pseudocode Implementation

## References