RNA-RNA Interaction Prediction with Stochastic Grammars

Sebastian Wild Markus Nebel Anika Scheid

{s_wild,nebel}@cs.uni-kl.de

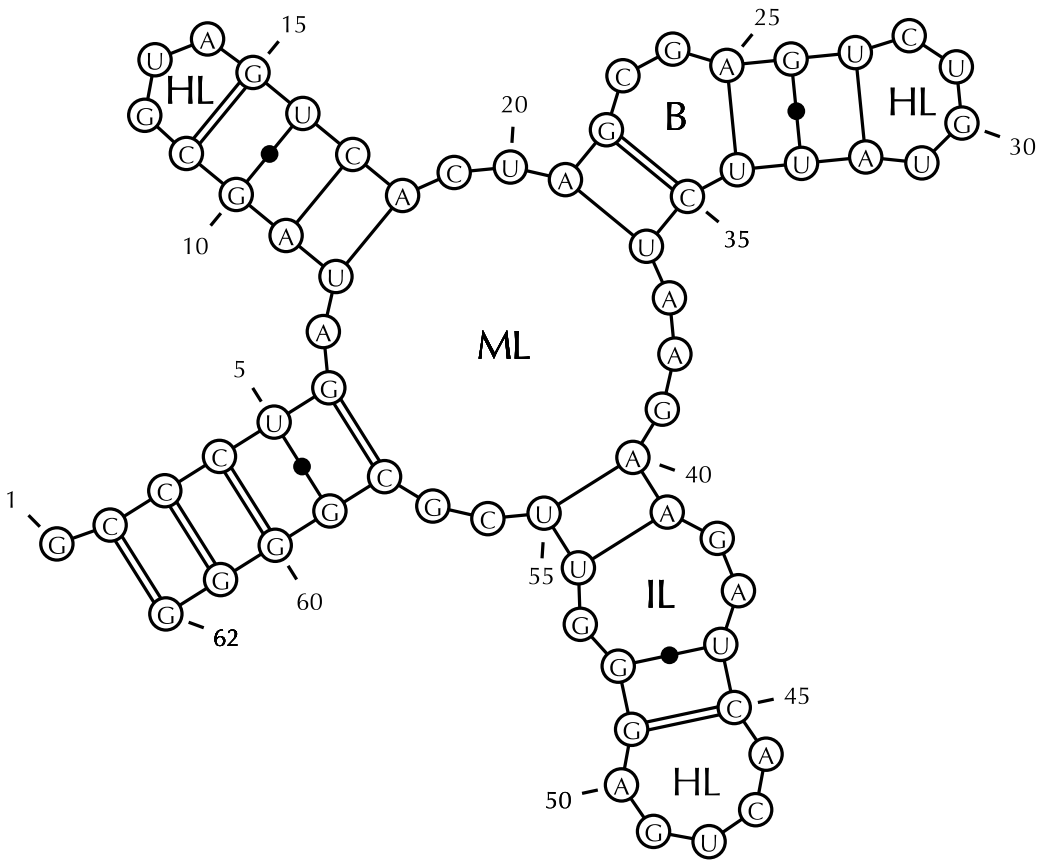
Fachbereich Informatik

Technische Universität Kaiserslautern

5. October 2012

10. Herbstseminar der Bioinformatik

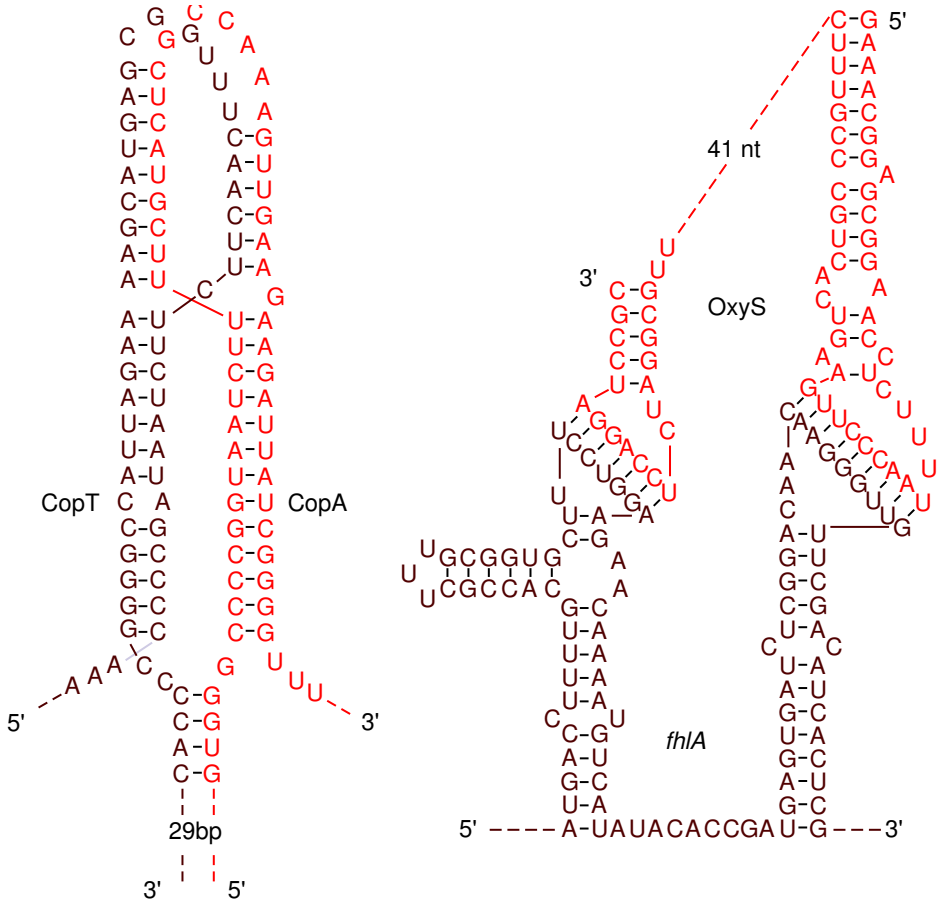
RNA secondary structure: model



- **primary structure:**
  word over $\{a, c, g, u\}$
- **secondary structure:**
  parentheses word $\{(, |,)\}$

```
g c c c u g a u a g c g u a g u c a c u a g c g a g u c u g u a u u c u a a g a a g a u c a c u g a g g g u u c g c g g g g
| ( ( ( ( ( | ( ( ( ( | | | ) ) ) ) ) | | ( ( | | ( ( ( | | | | ) ) ) ) ) | | | ( ( | | ( ( | | | | | ) ) | ) ) | | ) ) ) ) )
```

RNA also interacts!

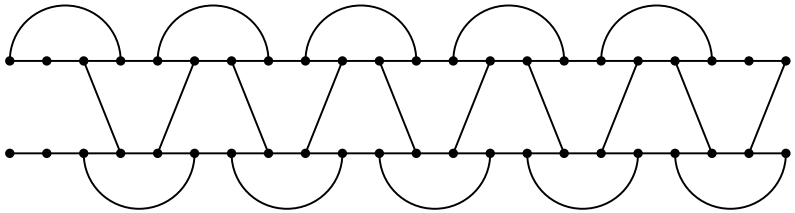


- bacterial antisense RNA
- interact non-trivally, “knotted” structure
- **goal:** predict whole interaction structure, not only interaction sites

RNA-RNA interaction problem (RIP)

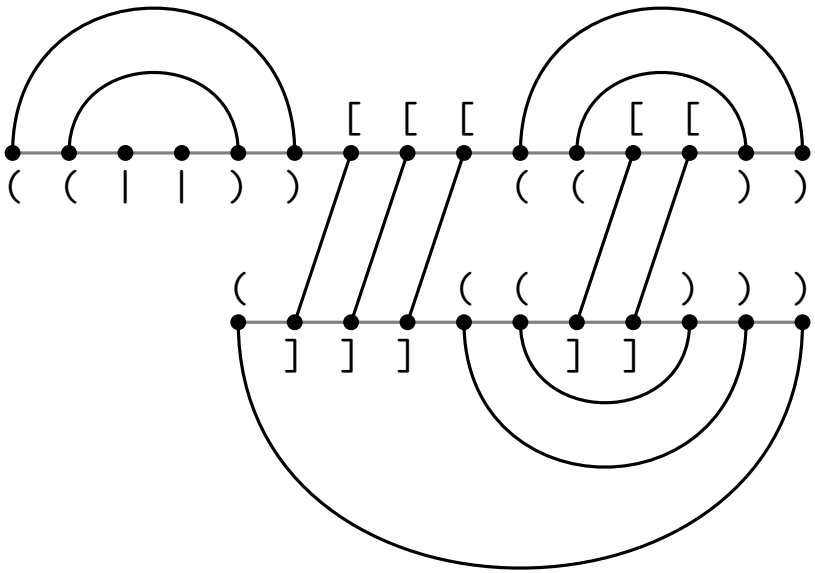
- **two** RNA molecules interact, i. e.
  form a ***joint*** *secondary structure*
- predict the "best" one possible

- RIP in general $\mathcal{NP}$-complete $\rightsquigarrow$ restrict structure
  1. Exclude **pseudo knots** (internal & external)
  2. Exclude **Zig-Zags**:



Example joint secondary structure

- $\rightsquigarrow$ joint structure can be encoded as **"2D-word"**:
  - pair of upper and lower word
  - matching ( ) $\rightsquigarrow$ internal bond
  - matching [ ] $\rightsquigarrow$ external bond (between two molecules)
  - | $\rightsquigarrow$ unpaired base



$\rightsquigarrow$

$$\begin{pmatrix} ((||))[[[((\,[[\,)) \\ \quad\quad (]]]((]])))\end{pmatrix}$$

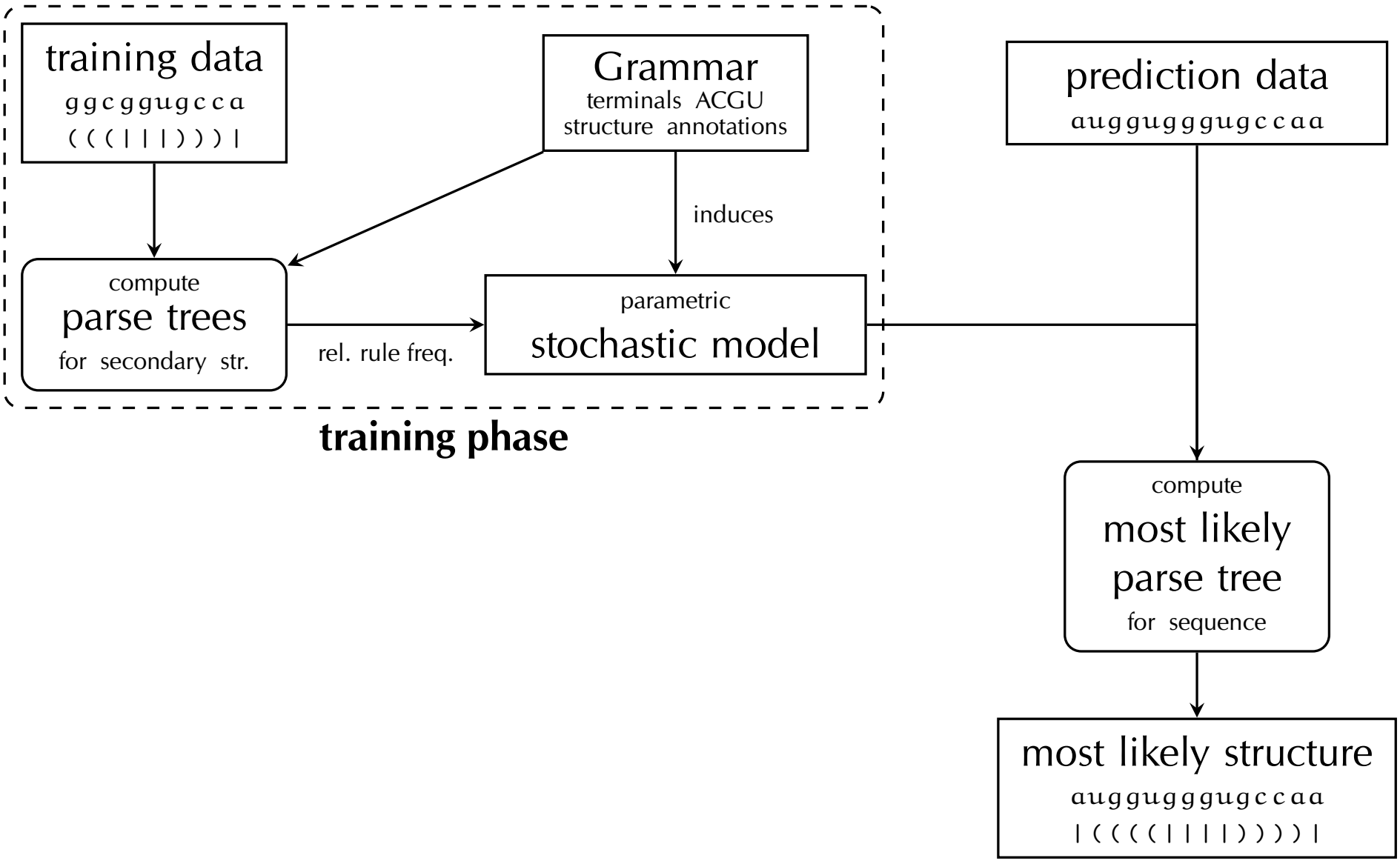
Stochastic CFG

Stochastic Context-free Grammars

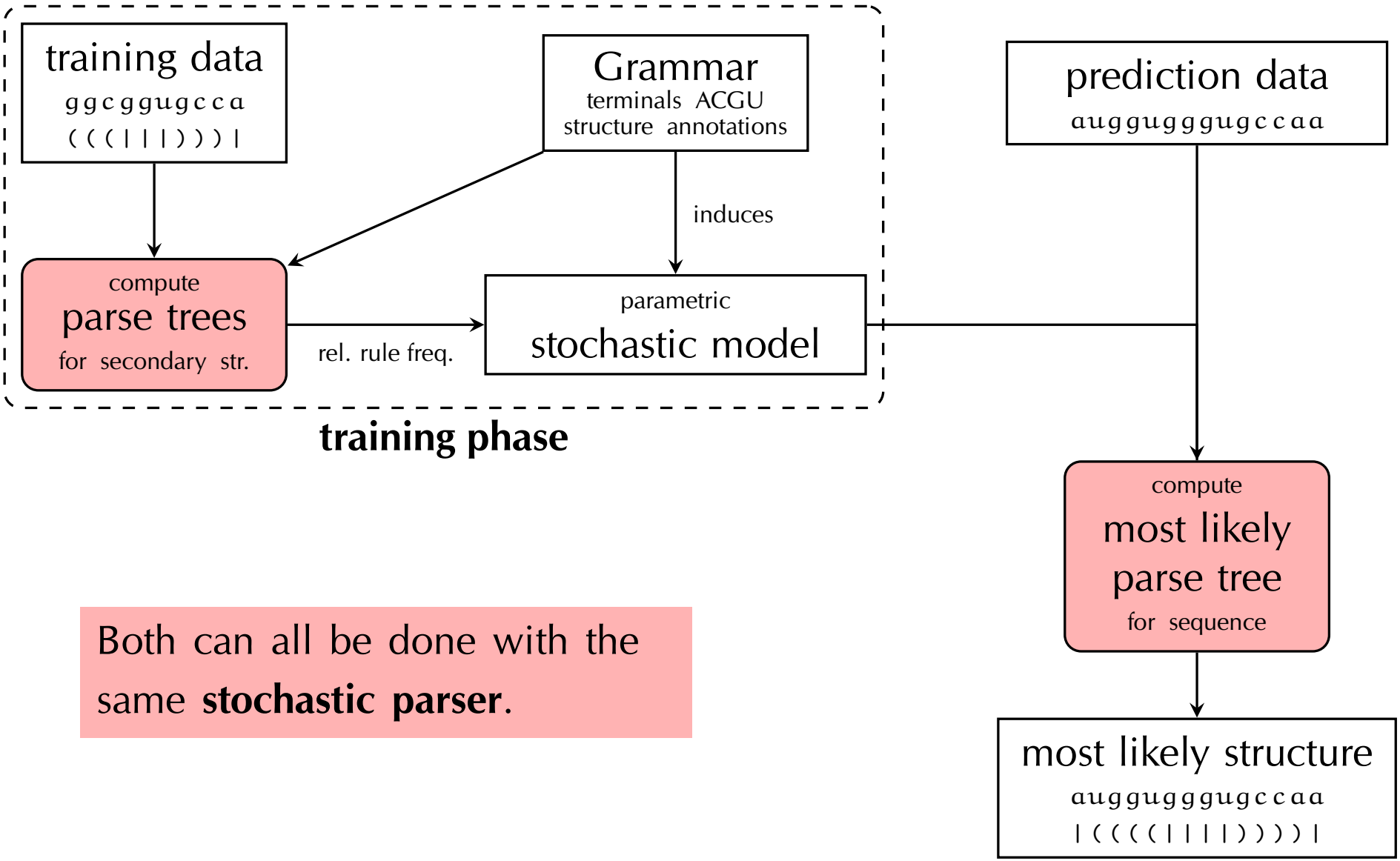
- $G = (N, \Sigma, R, S, P)$
  - nonterminals N
  - terminal alphabet Σ
  - rule set $R \subset N \times (N \cup \Sigma)^*$
  - start nonterminal $S \in N$
  - $P : R \to [0, 1]$: rule probabilities

- Probability of a **derivation tree**:
  Product of used rules' probabilities
- For structure prediction:
  - $\Sigma = \{ |_b, (_b,)_b \quad : \quad b \in \{a, c, g, u\} \}$
  - **unambiguous** w. r. t. structure

Structure Prediction with Formal Grammars



Structure Prediction with Formal Grammars



Both can all be done with the same **stochastic parser**.

Stochastic CFG

Stochastic Context-free Grammars

- $G = (N, \Sigma, R, S, P)$
  - nonterminals N
  - terminal alphabet Σ
  - rule set $R \subset N \times (N \cup \Sigma)^\star$
  - start nonterminal $S \in N$
  - $P : R \to [0, 1]$: rule probabilities

2-dimensional CFG

2-dimensional Context-free Grammars

- $G = (N, \Sigma, R, S, P)$
  - nonterminals N
  - terminal alphabet Σ
  - rule set $R \subset N \times \left(N \cup (\Sigma^*)^2\right)^*$
  - start nonterminal $S \in N$
  - $P : R \to [0, 1]$: rule probabilities

A Simplistic Grammar for RIP

$$S \to (\,|\,), \qquad S \to (\,|\,)S, \qquad S \to (\,(\,)\,), \qquad S \to (\,(\,)\,)S$$

$$S \to (\,|\,), \qquad S \to (\,|\,)S, \qquad S \to (\,(\,)\,), \qquad S \to (\,(\,)\,)S$$

$$S \to (\,[\,]\,), \qquad S \to (\,[\,]\,)S$$

Example:

S

A Simplistic Grammar for RIP

$$S \to (\,|\,), \qquad S \to (\,|\,)S, \qquad S \to (\,(\,)\,), \qquad S \to (\,(\,)\,)S$$

$$S \to (\,|\,), \qquad S \to (\,|\,)S, \qquad S \to (\,(\,)\,), \qquad S \to (\,(\,)\,)S$$

$$S \to (\,[\,]\,), \qquad S \to (\,[\,]\,)S$$

Example:

$(\,|\,)S$

A Simplistic Grammar for RIP

$$S \to (\,|\,), \qquad S \to (\,|\,)S, \qquad S \to (\,(\,)S(\,)\,), \qquad S \to (\,(\,)S(\,)\,)S$$

$$S \to (\,|\,), \qquad S \to (\,|\,)S, \qquad S \to (\,(\,)S(\,)\,), \qquad S \to (\,(\,)S(\,)\,)S$$

$$S \to (\,[\,]\,), \qquad S \to (\,[\,]\,)S$$

Example:

$$(\,|\,)(\,(\,)S(\,)\,)S$$

A Simplistic Grammar for RIP

$$S \to (\,|\,), \qquad S \to (\,|\,)S, \qquad S \to (\,(\,)S(\,)\,), \qquad S \to (\,(\,)S(\,)\,)S$$

$$S \to (\,|\,), \qquad S \to (\,|\,)S, \qquad S \to (\,(\,)S(\,)\,), \qquad S \to (\,(\,)S(\,)\,)S$$

$$S \to (\,[\,]\,), \qquad S \to (\,[\,]\,)S$$

Example:

$$(\,|\,)(\,(\,)(\,(\,)S(\,)\,)(\,)\,)S$$

A Simplistic Grammar for RIP

$$S \to (|), \qquad S \to (|)S, \qquad S \to (())S, \qquad S \to (())S$$

$$S \to (|), \qquad S \to (|)S, \qquad S \to ()S(), \qquad S \to ()S()S$$

$$S \to ([]), \qquad S \to ([])S$$

Example:

$$(|)(())(([])S())S$$

A Simplistic Grammar for RIP

$$S \to (\,|\,), \qquad S \to (\,|\,)S, \qquad S \to (\,(\,)S(\,)\,), \qquad S \to (\,(\,)S(\,)\,)S$$

$$S \to (\,|\,), \qquad S \to (\,|\,)S, \qquad S \to (\,(\,)S(\,)\,), \qquad S \to (\,(\,)S(\,)\,)S$$

$$S \to (\,[\,]\,), \qquad S \to (\,[\,]\,)S$$

Example:

$$(\,|\,)(\,(\,)(\,(\,[\,]\,)(\,[\,]\,)S(\,)\,)(\,)\,)S$$

A Simplistic Grammar for RIP

$$S \to (\,|\,), \qquad S \to (\,|\,)S, \qquad S \to (\,(\,)\,), \qquad S \to (\,(\,)\,)S$$

$$S \to (\,|\,), \qquad S \to (\,|\,)S, \qquad S \to (\,(\,)\,), \qquad S \to (\,(\,)\,)S$$

$$S \to (\,[\,]\,), \qquad S \to (\,[\,]\,)S$$

Example:

$$(\,|\,)(\,(\,)(\,[\,]\,)(\,[\,]\,)(\,[\,]\,)(\,)\,)S$$

A Simplistic Grammar for RIP

$$S \to (|), \qquad S \to (|)S, \qquad S \to (())S, \qquad S \to (())S$$

$$S \to (_|), \qquad S \to (_|)S, \qquad S \to (_())S, \qquad S \to (_())S$$

$$S \to ([]), \qquad S \to ([])S$$

Example:

$$(|)(())([])([])([])())()(|)S$$

A Simplistic Grammar for RIP

$$S \to (\,|\,), \qquad S \to (\,|\,)S, \qquad S \to (\,(\,)S(\,)\,), \qquad S \to (\,(\,)S(\,)\,)S$$

$$S \to (\,|\,), \qquad S \to (\,|\,)S, \qquad S \to (\,(\,)S(\,)\,), \qquad S \to (\,(\,)S(\,)\,)S$$

$$S \to (\,[\,]\,), \qquad S \to (\,[\,]\,)S$$

Example:

$$(\,|\,)(\,(\,)(\,(\,[\,]\,)(\,[\,]\,)(\,[\,]\,)(\,)\,)(\,)\,)(\,|\,)(\,|\,)$$

A Simplistic Grammar for RIP

$$S \to (\,|\,), \qquad S \to (\,|\,)S, \qquad S \to (\,(\,)S(\,)\,), \qquad S \to (\,(\,)S(\,)\,)S$$

$$S \to (\,|\,), \qquad S \to (\,|\,)S, \qquad S \to (\,(\,)S(\,)\,), \qquad S \to (\,(\,)S(\,)\,)S$$

$$S \to (\,[\,]\,), \qquad S \to (\,[\,]\,)S$$

Example:

$$(\,|\,)(\,(\,)(\,(\,[\,]\,)(\,[\,]\,)(\,[\,]\,)(\,)\,)(\,)\,)(\,|\,)(\,|\,)$$

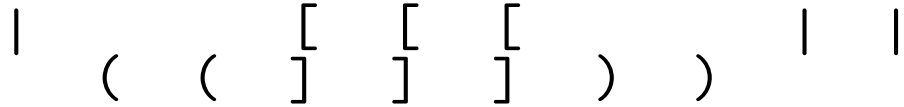
A Simplistic Grammar for RIP

$S \to (|)$, $S \to (|)S$, $S \to (())$, $S \to (())S$

$S \to (\,|\,)$, $S \to (\,|\,)S$, $S \to (\,(\,)\,)$, $S \to (\,(\,)\,)S$

$S \to (\,[\,]\,)$, $S \to (\,[\,]\,)S$

Example:



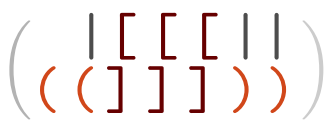
A Simplistic Grammar for RIP

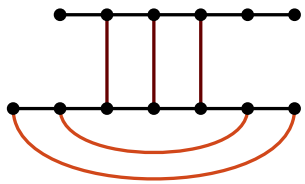
$S \to (|)$, $S \to (|)S$, $S \to (())$, $S \to (())S$

$S \to (\,)$, $S \to (\,)S$, $S \to (\,())$, $S \to (\,())S$

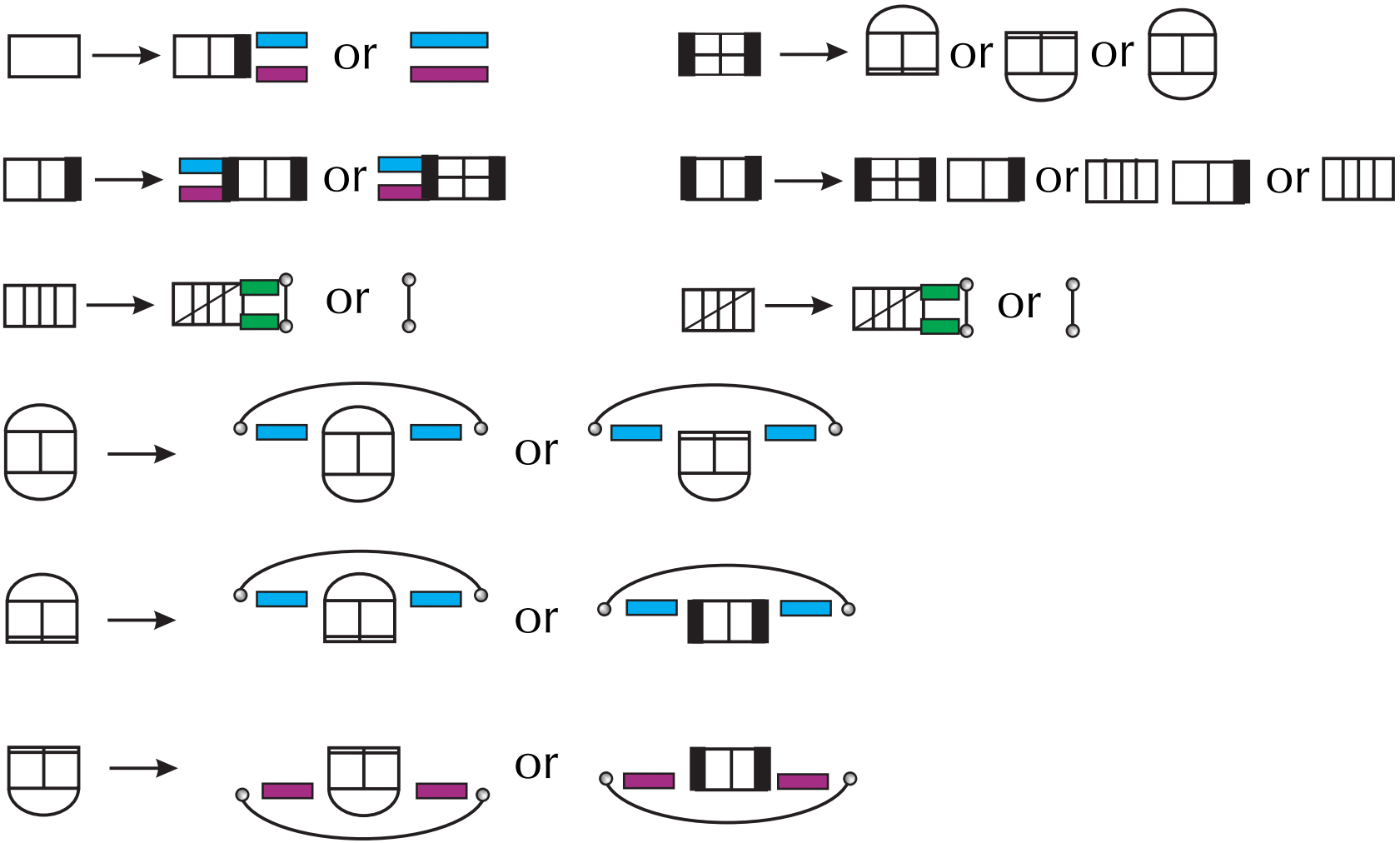
$S \to ([])$, $S \to ([])S$

Example:





Used Grammar



Implementation

- stochastic parser, independent of grammar
- mixture of Earley-Parser and dynamic programming
- fast manual implementation in C++

RNA pair	n	m	runtime	memory
DIS DIS	35	35	2 min	300 MB
CopA CopT	56	57	1 h	2 GB
ompA MicA	137	72	2 d	18 GB
U2 and U6 snRNAs in yeast 21	144	95	1 week	34 GB

Summary

This Talk:

- 2D-CFGs give stochastic model for RNA joint structures (only slight extension of CFGs needed)
- Earley parsing can be used to train model and compute most likely structures
- efficient implementation available

Open Problems:

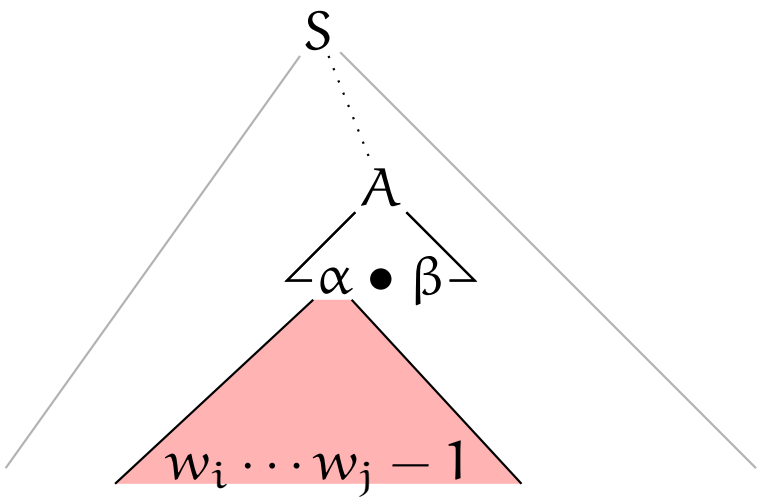
- get good training data
- full empirical evaluation of prediction quality

Earley-Parsing

- does not need normal form
- here: as formal calculus
- defined in terms of **items**

  $(i \;\; j\,,\; A \to \alpha \bullet \beta)$ can be derived iff

$$S \;\Rightarrow^*\; w_{1,i-1}A\gamma \;\Rightarrow\; w_{1,i-1}\alpha\beta\gamma \;\Rightarrow^*\; w_{1,j-1}\beta\gamma$$



Earley-Parser for SCFGs

- Items: $(i \;\; j\,,\; A \to \alpha \bullet \beta)$

  derivable **iff** $S' \Rightarrow^* w_{1,i-1} A\gamma \Rightarrow w_{1,i-1}\alpha\beta\gamma \Rightarrow^* w_{1,j-1}\beta\gamma$

- Start-Item: $(1 \;\; 1\,,\; S' \to \bullet\, S)$, Goal-Item $(1 \;\; n+1\,,\; S' \to S \bullet)$
- Derivation Rules:
  - Scanner

$$\frac{(i \;\; j-1\,,\; A \to \alpha \bullet w_{j-1}\beta)}{(i \quad j \quad,\; A \to \alpha w_{j-1} \bullet \beta)}$$

  - Predictor:

$$\frac{(i \;\; j\,,\; A \to \alpha \bullet B\gamma)}{(j \;\; j\,,\; B \to \bullet\, \beta)}$$

  - Completer:

$$\frac{(i \;\; r\,,\; A \to \alpha \bullet B\gamma) \qquad (r \;\; j\,,\; B \to \beta \bullet)}{(i \;\; j\,,\; A \to \alpha B \bullet \gamma)}$$

Earley-Parser for 2D-CFGs

- Items: $\begin{pmatrix} i & j \\ k & l \end{pmatrix}, A \to \begin{matrix} \alpha_1 \bullet \beta_1 \\ \alpha_2 \bullet \beta_2 \end{matrix}\Big)$

  derivable **iff** $S' \Rightarrow^* \begin{matrix} u_{1,i-1} \\ v_{1,k-1} \end{matrix} A\gamma \Rightarrow \begin{matrix} u_{1,i-1} \\ v_{1,k-1} \end{matrix} \alpha\beta\gamma \Rightarrow^* \begin{matrix} u_{1,j-1} \\ v_{1,l-1} \end{matrix} \beta\gamma$

- Start-Item: $\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, S' \to \begin{matrix} \bullet S_1 \\ \bullet S_2 \end{matrix}\Big)$, Goal-Item $\begin{pmatrix} 1 & n+1 \\ 1 & m+1 \end{pmatrix}, S' \to \begin{matrix} S_1 \bullet \\ S_2 \bullet \end{matrix}\Big)$
- Derivation Rules:
  - Scanner (**upper**) (lower similar):

$$\frac{\begin{pmatrix} i & j-1 \\ k & l \end{pmatrix}, A \to \begin{matrix} \alpha_1 \bullet u_{j-1} \beta_1 \\ \alpha_2 \bullet \quad \beta_2 \end{matrix}\Big)}{\begin{pmatrix} i & j \\ k & l \end{pmatrix}, A \to \begin{matrix} \alpha_1 u_{j-1} \bullet \beta_1 \\ \alpha_2 \quad \bullet \beta_2 \end{matrix}\Big)}$$

  - Predictor:

$$\frac{\begin{pmatrix} i & j \\ k & l \end{pmatrix}, A \to \begin{matrix} \alpha_1 \bullet B_1\gamma_1 \\ \alpha_2 \bullet B_2\gamma_2 \end{matrix}\Big)}{\begin{pmatrix} j & j \\ l & l \end{pmatrix}, B \to \begin{matrix} \bullet \beta_1 \\ \bullet \beta_2 \end{matrix}\Big)}$$

  - Completer:

$$\frac{\begin{pmatrix} i & r \\ k & s \end{pmatrix}, A \to \begin{matrix} \alpha_1 \bullet B_1\gamma_1 \\ \alpha_2 \bullet B_2\gamma_2 \end{matrix}\Big) \quad \begin{pmatrix} r & j \\ s & l \end{pmatrix}, B \to \begin{matrix} \beta_1 \bullet \\ \beta_2 \bullet \end{matrix}\Big)}{\begin{pmatrix} i & j \\ k & l \end{pmatrix}, A \to \begin{matrix} \alpha_1 B_1 \bullet \gamma_1 \\ \alpha_2 B_2 \bullet \gamma_2 \end{matrix}\Big)}$$

Implementation Note: Dense Grammars

- Number of items: $\Theta(\frac{1}{4}n^2m^2)$
  for n and m lengths of two RNA sequences
- structure prediction grammars are **dense:**
  most items are derivable
- $\rightsquigarrow$ compute values for **all** items à la dynamic programming