Local Substitutability for Sequence Generalization

François Coste, Gaëlle Garet, Jacques Nicolas

Dyliss Bioinformatic Team
Inria Rennes-Bretagne Atlantique
France

ICGI, September 6, 2012
Table of Contents

1 Biological Problem to Grammatical Inference
2 Generalization using Substitutability
3 Generalization using Local Substitutability
4 First Experiments
Biological Problem

Prediction of Protein Function

Protein:

- Amino acid sequence: length \( \approx 500 \), alphabet of size 20

KETAAAKFERQHMDSSTSAASSSNYCNQMMKSRNLTKDRCKPVTTFVHESLADVQA VCSQKNVACKNGQNTNCYQSYSTM

- Structure: determined by sequence
- Function: largely dependent on structure

A lot of sequences available (sequencing projects)

\( \Rightarrow \) Find the protein’s function from its sequence
Characterization of a Protein Functional Family

- Usual representations: Sub-regular expressions, profiles, ...

- Proteins:
  - short term interactions
  - long term interactions
Characterization of a Protein Functional Family

- Usual representations:
  Sub-regular expressions, profiles, ...

- Proteins:
  short term interactions
  long term interactions

KETAAAKFERQHMDSSTSAASSSNYCN-
QMMKSRNL...

alpha helix
beta sheet
Characterization of a Protein Functional Family

- Usual representations:
  - Sub-regular expressions, profiles, ...

- Proteins:
  - short term interactions: automata[Ker08]
  - long term interactions

KETAAKFERQHMDSSSTSAASSSNYCN-QMMKSRNL...
Characterization of a Protein Functional Family

- Usual representations:
  - Sub-regular expressions, profiles, ...

- Proteins:
  - short term interactions
  - long term interactions
Characterization of a Protein Functional Family

- Usual representations:
  - Sub-regular expressions, profiles, ...

- Proteins:
  - short term interactions
  - long term interactions

Abstraction
Context free grammars enable modeling important protein contacts.

Issue
How to infer such CFG from a set of protein sequences?
**Protomata-inspired Approach**

- Detection of blocks of conservation by partial local multiple alignment [Ker08]

<table>
<thead>
<tr>
<th>Seq1</th>
<th>SVSLD</th>
<th>IDLQTWLPEWVRVGFSASTG</th>
<th>QNV</th>
<th>ERNSILAWSFSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seq2</td>
<td>TVSYD</td>
<td>VDLKTELPEWVRVGFSGSTG</td>
<td>GYV</td>
<td>QNHNILSWTFNS</td>
</tr>
<tr>
<td>Seq3</td>
<td>HVSAT</td>
<td>VPLEKEVEDWVSVGFSATSG</td>
<td>SKKETT</td>
<td>ETHNVLWSFSS</td>
</tr>
<tr>
<td>Seq4</td>
<td>NVSTT</td>
<td>VELEKEVYDWVSVGFSATSG</td>
<td>AYQWSY</td>
<td>ETHDVLWSFSS</td>
</tr>
<tr>
<td>Seq5</td>
<td>SVSAT</td>
<td>VHLEKEVDEWVSVGFSATSG</td>
<td>LTEDDTT</td>
<td>ETHDVLWSFSS</td>
</tr>
</tbody>
</table>

- Recoding sequences with conservation blocks

<table>
<thead>
<tr>
<th>Seq1</th>
<th>Block1</th>
<th>Block2</th>
<th>Block3</th>
<th>Block4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seq2</td>
<td>Block1</td>
<td>Block2</td>
<td>Block3</td>
<td>Block4</td>
</tr>
<tr>
<td>Seq3</td>
<td>Block5</td>
<td>Block2</td>
<td>Block6</td>
<td>Block4</td>
</tr>
<tr>
<td>Seq4</td>
<td>Block5</td>
<td>Block2</td>
<td>Block6</td>
<td>Block4</td>
</tr>
<tr>
<td>Seq5</td>
<td>Block5</td>
<td>Block2</td>
<td>Block6</td>
<td>Block4</td>
</tr>
</tbody>
</table>

- Grammar induced by recoding

\[
S \rightarrow \text{Block1} \ \text{Block2} \ \text{Block3} \ \text{Block4} \\
| \text{Block5} \ \text{Block2} \ \text{Block6} \ \text{Block4} \\
\text{Block1} \rightarrow \text{P1} \ \text{P2} \ \text{P3} \ \text{P4} \ \text{P5} \\
\text{P1} \rightarrow S | T \\
...
\]

How to generalize more?
Protomata-inspired Approach

- Detection of blocks of conservation by partial local multiple alignment [Ker08]

<table>
<thead>
<tr>
<th>Seq1</th>
<th>Block1</th>
<th>Block2</th>
<th>Block3</th>
<th>Block4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seq2</td>
<td>Block1</td>
<td>Block2</td>
<td>Block3</td>
<td>Block4</td>
</tr>
<tr>
<td>Seq3</td>
<td>Block5</td>
<td>Block2</td>
<td>Block6</td>
<td>Block4</td>
</tr>
<tr>
<td>Seq4</td>
<td>Block5</td>
<td>Block2</td>
<td>Block6</td>
<td>Block4</td>
</tr>
<tr>
<td>Seq5</td>
<td>Block5</td>
<td>Block2</td>
<td>Block6</td>
<td>Block4</td>
</tr>
</tbody>
</table>

- Recoding sequences with conservation blocks

- Grammar induced by recoding

\[ S \rightarrow \text{Block1 Block2 Block3 Block4} \]
\[ \text{Block5 Block2 Block6 Block4} \]
\[ \text{Block1} \rightarrow \text{P1 P2 P3 P4 P5} \]
\[ \text{P1} \rightarrow S \mid T \]

How to generalize more?
### Protopomata-inspired Approach

- **Detection of blocks of conservation by partial local multiple alignment** [Ker08]

<table>
<thead>
<tr>
<th>Seq1</th>
<th>SVSLD</th>
<th>IDLQTVLPEWVRVGFSASTG</th>
<th>QNV</th>
<th>ERNSILAWSFSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seq2</td>
<td>TVSYD</td>
<td>VDLKTELPEWVRVGFSGSTG</td>
<td>GYV</td>
<td>QNHNILSWTFNS</td>
</tr>
<tr>
<td>Seq3</td>
<td>HVSAT</td>
<td>VPLEKEVEDWVSVGFSATSG</td>
<td>SKKETT</td>
<td>ETHNVLSWSFSS</td>
</tr>
<tr>
<td>Seq4</td>
<td>NVSTT</td>
<td>VELEKEVYDWVSVGFSATSG</td>
<td>AYQWSY</td>
<td>ETHDLVLSWSFSS</td>
</tr>
<tr>
<td>Seq5</td>
<td>SVSAT</td>
<td>VHLEKEVDEWVSVGFSATSG</td>
<td>LTEEDTT</td>
<td>ETHDLVLSWSFSS</td>
</tr>
</tbody>
</table>

- **Recoding sequences with conservation blocks**

```
<table>
<thead>
<tr>
<th>Seq1</th>
<th>Block1 Block2 Block3 Block4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seq2</td>
<td>Block1 Block2 Block3 Block4</td>
</tr>
<tr>
<td>Seq3</td>
<td>Block5 Block2 Block6 Block4</td>
</tr>
<tr>
<td>Seq4</td>
<td>Block5 Block2 Block6 Block4</td>
</tr>
<tr>
<td>Seq5</td>
<td>Block5 Block2 Block6 Block4</td>
</tr>
</tbody>
</table>
```

- **Grammar induced by recoding**

```
S  →  Block1 Block2 Block3 Block4 | Block5 Block2 Block6 Block4
Block1  →  P1 P2 P3 P4 P5
P1  →  S | T
...```

### How to generalize more?
Table of Contents

1. Biological Problem to Grammatical Inference
2. Generalization using Substitutability
3. Generalization using Local Substitutability
4. First Experiments
Substitutability\([\text{Har54}]\) Based Inference

- \([\text{CE07}]\): substitutable languages

\[
\forall y_1, y_2 \in \Sigma^+ : \\
\exists \langle x_1, z_1 \rangle : x_1 y_1 z_1 \in L \land x_1 y_2 z_1 \in L \\
\Rightarrow \forall \langle x_2, z_2 \rangle : x_2 y_1 z_2 \in L \iff x_2 y_2 z_2 \in L
\]

Two strings occurring between common left and right contexts are substitutable.

- \([\text{Yos08}]\): \((k,l)\)-substitutable languages

\[
\forall y_1, y_2 \in \Sigma^+, \forall \langle u, v \rangle \in \langle \Sigma^k, \Sigma^l \rangle : \\
\exists \langle x_1, z_1 \rangle : x_1 u y_1 v z_1 \in L \land x_1 u y_2 v z_1 \in L \\
\Rightarrow \forall \langle x_2, z_2 \rangle : x_2 u y_1 v z_2 \in L \iff x_2 u y_2 v z_2 \in L
\]

Two strings occurring between common left and right contexts are substitutable \textit{in} these left and right sub-contexts of length \(k\) and \(l\).
Substitutability [Har54] Based Inference

- [CE07]: substitutable languages

\[ \forall y_1, y_2 \in \Sigma^+ : \]
\[ [\exists \langle x_1, z_1 \rangle : x_1 y_1 z_1 \in L \land x_1 y_2 z_1 \in L] \]
\[ \Rightarrow [\forall \langle x_2, z_2 \rangle : x_2 y_1 z_2 \in L \iff x_2 y_2 z_2 \in L] \]

Two strings occurring between common left and right contexts are substitutable.

- [Yos08]: (k,l)-substitutable languages

\[ \forall y_1, y_2 \in \Sigma^+, \forall \langle u, v \rangle \in \langle \Sigma^k, \Sigma^l \rangle : \]
\[ [\exists \langle x_1, z_1 \rangle : x_1 u y_1 v z_1 \in L \land x_1 u y_2 v z_1 \in L] \]
\[ \Rightarrow [\forall \langle x_2, z_2 \rangle : x_2 u y_1 v z_2 \in L \iff x_2 u y_2 v z_2 \in L] \]

Two strings occurring between common left and right contexts are substitutable in these left and right sub-contexts of length k and l.
Preliminary Experiments on Protein Sequences

- Unsatisfactory results
- No generalization

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before substitutability</td>
<td>1</td>
<td>0.2</td>
<td>0.33</td>
</tr>
<tr>
<td>After substitutability</td>
<td>1</td>
<td>0.2</td>
<td>0.33</td>
</tr>
</tbody>
</table>

- Analysis of failure causes
  - Training sequences are long
  - (Global) Contexts of two strings are never identical

How to generalize more?

Our solution: Introduction of local substitutability

- new classes of languages
- new generalization criterion
Preliminary Experiments on Protein Sequences

- Unsatisfactory results
- No generalization

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before substitutability</td>
<td>1</td>
<td>0.2</td>
<td>0.33</td>
</tr>
<tr>
<td>After substitutability</td>
<td>1</td>
<td>0.2</td>
<td>0.33</td>
</tr>
</tbody>
</table>

- Analysis of failure causes
  - Training sequences are long
  - (Global) Contexts of two strings are never identical

**How to generalize more?**

Our solution: Introduction of local substitutability

- new classes of languages
- new generalization criterion
Preliminary Experiments on Protein Sequences

- Unsatisfactory results
- No generalization

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before substitutability</td>
<td>1</td>
<td>0.2</td>
<td>0.33</td>
</tr>
<tr>
<td>After substitutability</td>
<td>1</td>
<td>0.2</td>
<td>0.33</td>
</tr>
</tbody>
</table>

- Analysis of failure causes
  - Training sequences are long
  - (Global) Contexts of two strings are never identical

How to generalize more?

Our solution: Introduction of local substitutability

- new classes of languages
- new generalization criterion
Table of Contents

1. Biological Problem to Grammatical Inference
2. Generalization using Substitutability
3. Generalization using Local Substitutability
4. First Experiments
(\(k, l\))-Local Substitutability

- (\(k, l\))-local substitutable languages

\[
\forall y_1, y_2 \in \Sigma^+ : \\
[\exists \langle r, s \rangle \in \langle \Sigma^k, \Sigma^l \rangle : x_1 r y_1 s z_1 \in L \land x_2 r y_2 s z_2 \in L] \\
\Rightarrow [\forall \langle x_3, z_3 \rangle : x_3 y_1 z_3 \in L \iff x_3 y_2 z_3 \in L]
\]

Definition

Two strings occurring between common left and right contexts of length \(k\) and \(l\) are substitutable.
(\(k, l\))-Local-Context Substitutability

- \((k, l)\)-local context substitutable languages

\[
\forall y_1, y_2 \in \Sigma^+, \forall \langle u, v \rangle \in \langle \Sigma^k, \Sigma^l \rangle : \\
[ x_1 u y_1 v z_1 \in L \land x_2 u y_2 v z_2 \in L ]
\Rightarrow [ \forall \langle x_3, z_3 \rangle : x_3 u y_1 v z_3 \in L \iff x_3 u y_2 v z_3 \in L ]
\]

**Definition**

Two strings **occurring between** common left and right **contexts** of length \(k\) and \(l\) are **substitutable in** these **contexts** of length \(k\) and \(l\).
Generalization of Sequences: Example

Set of sequences:
- I have arrived after midnight.
- I have driven after midnight.
- She has arrived before me.
- Marie has eaten before him.

To obtain a language, we must add the following sequences:
- She has driven before me.
- She has eaten before me.
- Marie has arrived before him.
- Marie has driven before him.
- I have eaten after midnight.
Generalization of Sequences: Example

Set of sequences:

- I have arrived after midnight.
- I have driven after midnight.
- She has arrived before me.
- Marie has eaten before him.

To obtain a substitutable language, we must add the following sequences:

- She has driven before me.
- She has eaten before me.
- Marie has arrived before him.
- Marie has driven before him.
- I have eaten after midnight.
Generalization of Sequences: Example

Set of sequences:

- I have arrived after midnight.
- I have driven after midnight.
- She has arrived before me.
- Marie has eaten before him.

To obtain a (1,1) substitutable language, we must add the following sequences:

- She has driven before me.
- She has eaten before me.
- Marie has arrived before him.
- Marie has driven before him.
- I have eaten after midnight.
Generalization of Sequences: Example

**Set of sequences:**
- I have arrived after midnight.
- I have driven after midnight.
- She has arrived before me.
- Marie has eaten before him.

To obtain a \((1,1)\) context local substitutable language, we must add the following sequences:
- She has driven before me.
- She has eaten before me.
- Marie has arrived before him.
- Marie has driven before him.
- I have eaten after midnight.
Generalization of Sequences: Example

Set of sequences:

- I have arrived after midnight.
- I have driven after midnight.
- She has arrived before me.
- Marie has eaten before him.

To obtain a (1,1) local substitutable language, we must add the following sequences:

- She has driven before me.
- She has eaten before me.
- Marie has arrived before him.
- Marie has driven before him.
- I have eaten after midnight.
Links between Substitutable Languages

- **Two Complementary Usages of Contexts**

<table>
<thead>
<tr>
<th>Language</th>
<th>local definition</th>
<th>contextual application</th>
</tr>
</thead>
<tbody>
<tr>
<td>substitutable [CE07]</td>
<td>$(\infty, \infty)$</td>
<td>$(0, 0)$</td>
</tr>
<tr>
<td>$k, l$-context substitutable [Yos08]</td>
<td>$(\infty, \infty)$</td>
<td>$(k, l)$</td>
</tr>
<tr>
<td>$k, l$-local substitutable</td>
<td>$(k, l)$</td>
<td>$(0, 0)$</td>
</tr>
<tr>
<td>$k, l$-local context substitutable</td>
<td>$(k, l)$</td>
<td>$(k, l)$</td>
</tr>
<tr>
<td>$i, j$-local $k, l$-context substitutable</td>
<td>$(i, j)$</td>
<td>$(k, l)$</td>
</tr>
</tbody>
</table>

- **Inclusion of substitutable language classes**

```plaintext
diagram
```

- $k, l$-subst.
- subst.
- $k, l$-local subst.
- $k, l$-local context subst.

Hierarchy of \((i,j)\)-Local\((k,l)\)-Context Substitutable Languages
Hierarchy of $(i,j)$-Local$(k,l)$-Context Substitutable Languages

\[(i,j)(k, l)\]

\[(i+1, j)(k, l)\]

\[(i+1, j+1)(k+1, l+1)\]

... Context-free
Hierarchy of (i,j)-Local(k,l)-Context Substitutable Languages

\[
\begin{align*}
(i,j)(k+1,l) \\
(i+1,j)(k,l) \\
(0,0)(0,0) \\
\{\Sigma^*\} \\
(i,j)(k,l+1) \\
(i,j+1)(k,l) \\
(j+1,j+1)(k,l) \\
(i,j)(k+1,l+1) \\
(i+1,j+1)(k+1,l+1) \\
\ldots \\
\text{Context-free}
\end{align*}
\]
Local Substitutability and Testability

<table>
<thead>
<tr>
<th>Languages</th>
<th>Extension of</th>
</tr>
</thead>
<tbody>
<tr>
<td>Substitutable</td>
<td>0-reversible</td>
</tr>
<tr>
<td>k,l-substitutable</td>
<td>k-reversible</td>
</tr>
<tr>
<td>k,l-context local substitutable</td>
<td>k-testable</td>
</tr>
</tbody>
</table>

- **Reversible language**
  \[
  \forall y_1, y_2 \in \Sigma^+ : \\
  [\exists x_1 : x_1y_1 \in L \land x_1y_2 \in L] \Rightarrow [\forall x_2 : x_2y_1 \in L \iff x_2y_2 \in L]
  \]

- **Substitutable language**
  \[
  \forall y_1, y_2 \in \Sigma^+ : \\
  [\exists \langle x_1, z_1 \rangle : x_1y_1z_1 \in L \land x_1y_2z_1 \in L] \Rightarrow [\forall \langle x_2, z_2 \rangle : x_2y_1z_2 \in L \iff x_2y_2z_2 \in L]
  \]
Local Substitutability and Testability

<table>
<thead>
<tr>
<th>Languages</th>
<th>Extension of</th>
</tr>
</thead>
<tbody>
<tr>
<td>Substitutable</td>
<td>0-reversible</td>
</tr>
<tr>
<td>k,l-substitutable</td>
<td>k-reversible</td>
</tr>
<tr>
<td>k,l-context local substitutable</td>
<td>k-testable</td>
</tr>
</tbody>
</table>

- **k-reversible language**
  \[
  \forall y_1, y_2 \in \Sigma^+, \forall u \in \Sigma^k : \\
  [\exists x_1 : x_1uy_1 \in L \land x_1uy_2 \in L] \Rightarrow [\forall x_2 : x_2uy_1 \in L \iff x_2uy_2 \in L]
  \]

- **k, l-substitutable language**
  \[
  \forall y_1, y_2 \in \Sigma^+, \forall u \in \Sigma^k, v \in \Sigma^l : \\
  [\exists \langle x_1, z_1 \rangle : x_1uy_1vz_1 \in L \land x_1uy_2vz_1 \in L] \Rightarrow [\forall \langle x_2, z_2 \rangle : x_2uy_1vz_2 \in L \iff x_2uy_2vz_2 \in L]
  \]
## Local Substitutability and Testability

<table>
<thead>
<tr>
<th>Languages</th>
<th>Extension of</th>
</tr>
</thead>
<tbody>
<tr>
<td>Substitutable</td>
<td>0-reversible</td>
</tr>
<tr>
<td>k,l-substitutable</td>
<td>k-reversible</td>
</tr>
<tr>
<td><strong>k,l-context local substitutable</strong></td>
<td><strong>k-testable</strong></td>
</tr>
</tbody>
</table>

- **k-testable language**
  \[
  \forall y_1, y_2 \in \Sigma^+ \forall u \in \Sigma^k : \\
  [x_1 uy_1 \in L \land x_2 uy_2 \in L] \Rightarrow [\forall x_3 : x_3 uy_2 \in L \iff x_3 uy_1 \in L]
  \]

- **k, l-local context substitutable language**
  \[
  \forall y_1, y_2 \in \Sigma^+, \forall \langle u, v \rangle \in \langle \Sigma^k, \Sigma^l \rangle : \\
  [x_1 uy_1 v z_1 \in L \land x_2 uy_2 v z_2 \in L] \Rightarrow [\forall \langle x_3, z_3 \rangle : x_3 uy_1 v z_3 \in L \iff x_3 uy_2 v z_3 \in L]
  \]
Table of Contents

1 Biological Problem to Grammatical Inference
2 Generalization using Substitutability
3 Generalization using Local Substitutability
4 First Experiments
Learning Algorithm: $k, l$-local substitutability

$\hat{G}_{LS}$

Input: Set of sequences $K$, parameters $k$ and $l$
Output: Grammar $\hat{G} = \langle \Sigma_K, V_K, P_K, S \rangle$

Non-terminals definition
$V_K = \{[y] \mid xyz \in K, y \neq \lambda\} \cup \{S\}$

Induction of rules
Initial rules
$P_K = \{S \rightarrow [w] \mid w \in K\}$
Terminal rules
$\cup \{[a] \rightarrow a \mid a \in \Sigma\}$
Branching rules
$\cup \{[xy] \rightarrow [x][y] \mid [xy], [x], [y] \in V_K\}$
Substitutability rules
$\cup \{[y_1] \rightarrow [y_2] \mid \underbrace{x_1 uy_1 v z_1}_{\text{the local definition context}} \in K, \underbrace{x_2 uy_2 v z_2}_{\text{the local definition context}} \in K, \mid u \mid = k, \mid v \mid = l\}$
Learning Algorithm: \( k, l \)-local context substitutability

\( \hat{G}_{LCS} \)

Input: Set of sequences \( K \), parameters \( k \) and \( l \)
Output: Grammar \( \hat{G} = \langle \Sigma_K, V_K, P_K, S \rangle \)

Non-terminals definition
\( V_K = \{ [y] \mid xyz \in K, y \neq \lambda \} \cup \{ S \} \)

Induction of rules
Initial rules
\( P_K = \{ S \rightarrow [w] \mid w \in K \} \)
Terminal rules
\( \cup \{ [a] \rightarrow a \mid a \in \Sigma \} \)
Branching rules
\( \cup \{ [xy] \rightarrow [x][y] \mid [xy], [x], [y] \in V_K \} \)
Substitutability rules
\( \cup \{ [u y_1 v] \rightarrow [u y_2 v] \mid x_1 u y_1 v z_1 \in K, x_2 u y_2 v z_2 \in K, u \models k, v \models l \} \)

the application context \hspace{1cm} the local definition context
Experiments

**PS00307 Family[DN09]**
Training set of annotated sequences (22)

...TVSLDIDLQTVLPEWVRVGFSASTGQNVERNSILAWSFSS...
...TVSYDVLKTELPEWVRVGFSGSTGGYVQNHNILSWTFLNS...
...HVSATVEVEDWVSAGFSATSGSKKETETTETHNVLSWSFSS...
...NVSTTVKEVYDWWVSAGFSATSGAYQWSYETHDVLSWSFSS...
...SVSATVEVDWVSAGFSATSGLTDDETTETHDVLSWSFSS...
...

Preprocessing

Recoding sequences

...Block1 Block2 Block3 Block4...
...Block1 Block2 Block3 Block4...
...Block5 Block2 Block6 Block4...
...Block5 Block2 Block6 Block4...
...Block5 Block2 Block6 Block4...
...

Algorithm

Application

Grammar

Test set (20)
Positive test set (10)
Negative test set (10)

Recognition rate
### Results

<table>
<thead>
<tr>
<th>Generalization criterion</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Substitutability</td>
<td>1</td>
<td>0.2</td>
<td>0.33</td>
</tr>
<tr>
<td>4,4 - Local context substitutability</td>
<td>1</td>
<td>0.6</td>
<td>0.75</td>
</tr>
<tr>
<td>4-4 - Local substitutability</td>
<td>1</td>
<td>0.7</td>
<td>0.82</td>
</tr>
<tr>
<td>Stochastic CFG[DN09]</td>
<td>1</td>
<td>0.1</td>
<td>0.18</td>
</tr>
<tr>
<td>(with different thresholds)</td>
<td>0.3</td>
<td>1</td>
<td>0.46</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.9</td>
<td>0.85</td>
</tr>
</tbody>
</table>

Good generalization and still specific $\implies$ First encouraging results!
Conclusion

- Introduction of local substitutability
  extension of k-testability for context-free
  - new classes of language
  - new generalization criteria
- Application on proteins
  - first encouraging results
    - more practical (heuristic) algorithms
    - parsing efficiency
- Learnability of language classes
  - implied by learnability results of [Yos08]
    - better learnability results for local substitutable classes?
Questions?
References


