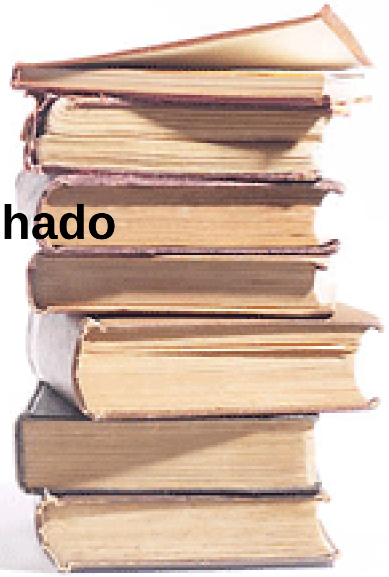
VNiVERSiDAD
D SALAMANCA

Data Mining for Grammatical Inference with Bioinformatics Criteria

Dra. Vivian López Batista
Luis Alonso, María N. Moreno and Juan M. Corchado
Dept. Informática y Automática.
University of Salamanca,
Plaza de la Merced S/N, 37008. Salamanca.

vivian@usal.es



VNiVERSiDAD
D SALAMANCA

VNiVERSiTAS
STVDII
SALAMANTINI

Topics

- Introduction
- Objectives
- The data mining procedure for the grammatical Inference (GI)
- Experiments
- Conclusions

VNiVERSiDAD Ð SALAMANCA

VNiVERSiTAS STVDII SALAMANTiNi

Introduction

In this work we present

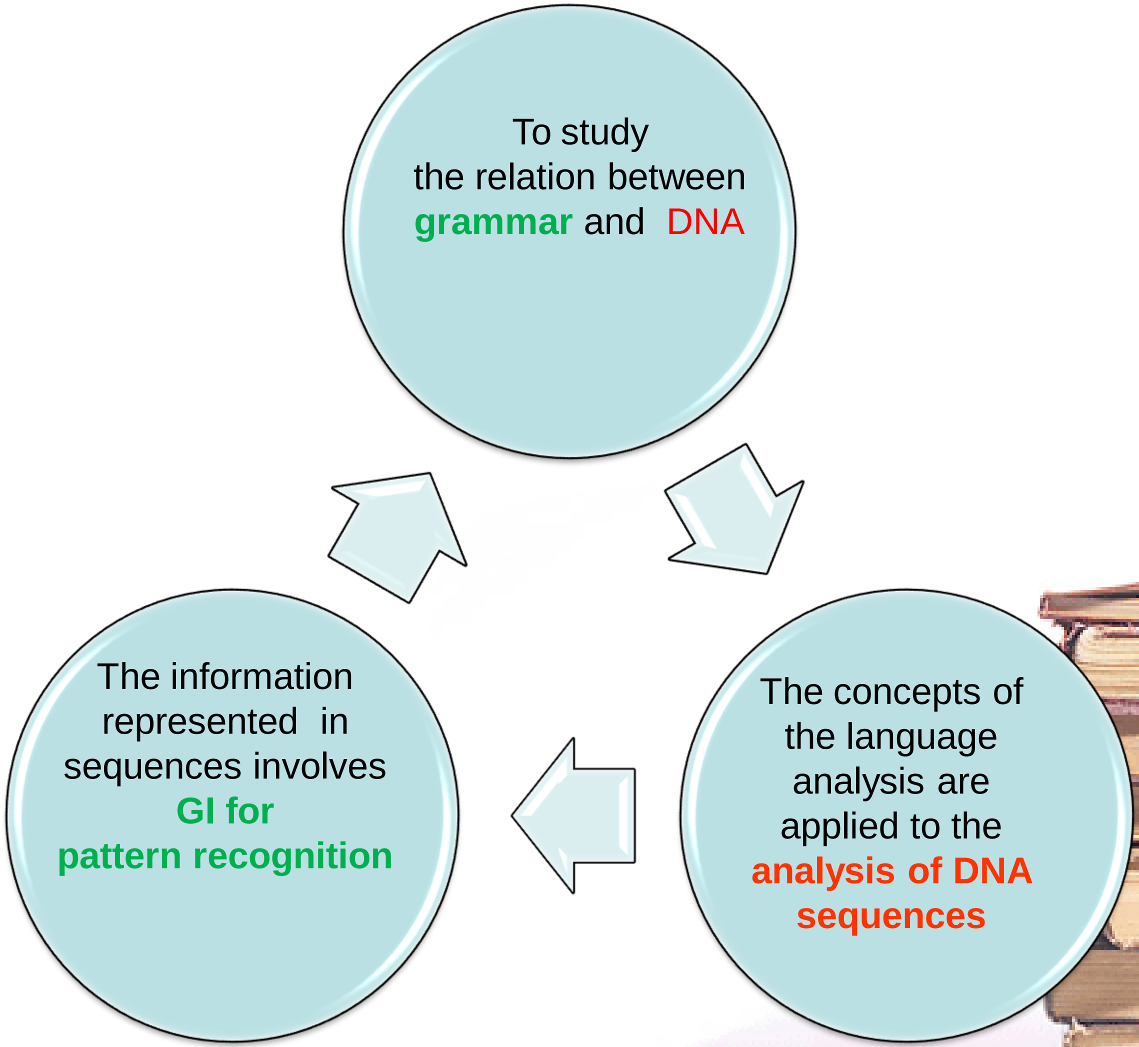
• ***a novel data mining process***
that combines hybrid techniques of genomics:
•of association analysis
• and classical sequentiation algorithms
to generate grammatical structures of a specific language.

. **these structures** are converted
to Context-Free Grammars (CFG).
Initially the method applies to context-free languages
with the possibility of being applied to other languages.

• ***we used a tool*** that allows measuring
the complexity of the obtained grammar automatically
from textual data.

http://dptoia.usal.es

Objectives



To study
the relation between
grammar and DNA

The concepts of
the language
analysis are
applied to the
**analysis of DNA
sequences**

The information
represented in
sequences involves
**GI for
pattern recognition**

http://dptoia.usal.es

Data Mining Procedure for the Grammatical Inference

The idea considers existing theories in order to process data that are not structured in relations or tables with differentiated attributes as **a finite succession of sentences.**

VNiVERSiDAD Ð SALAMANCA

VNiVERSiTAS STVDIi SALAMANTiNi

Data Mining Procedure for the Grammatical Inference

The procedure for the GI can be summarized in the following steps :

1. Language generation by means of a CFG .
2. Codification of the strings of the language into syntactic categories**.**
3. Discovery of Sequential Patterns in the codified language.
4. Replace the discovered sequences by their identifiers.
5. Repeat the two previous steps until all the sentences of the language are replaced by identifiers.

http://dptoia.usal.es

VNiVERSiDAD Ð SALAMANCA

VNiVERSiTAS STVDii SALAMANTiNi

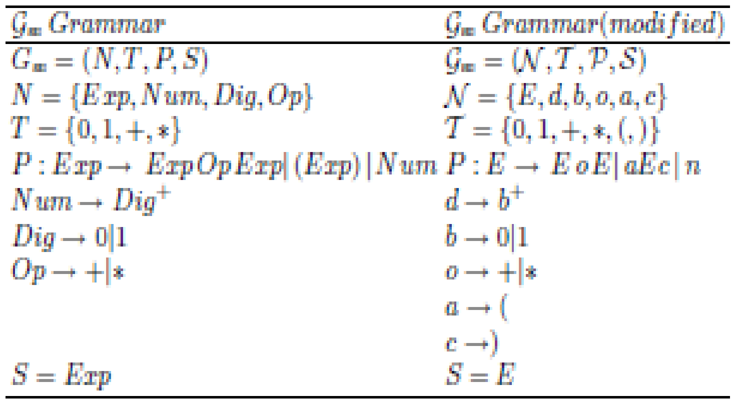
Language generation

http://dptoia.usal.es

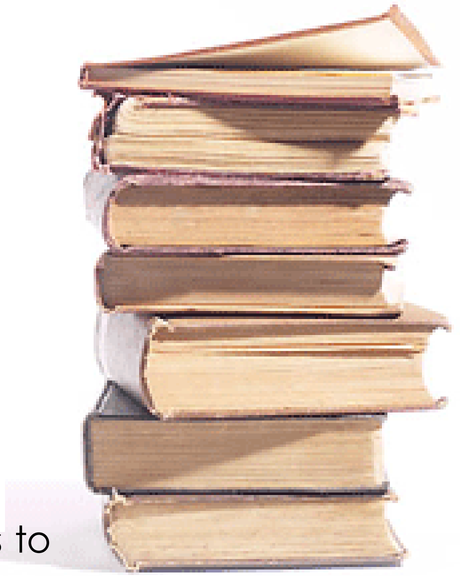
We consider **the CFG, $G_{\text{æ}}$** proposed by Louden on the generation of arithmetic expressions, the majority of the programming languages are generated by grammars of this type.

A DNA molecule can then be represented as a finite string of symbols from this alphabet; a language, formally, is any set of such string .

Table 1. Modification of the $\mathcal{G}_{\text{æ}}$ *Grammar*



$\mathcal{G}_{\text{æ}}$ *Grammar*	$\mathcal{G}_{\text{æ}}$ *Grammar(modified)*
$G_{\text{æ}} = (N, T, P, S)$	$\mathcal{G}_{\text{æ}} = (\mathcal{N}, \mathcal{T}, \mathcal{P}, \mathcal{S})$
$N = \{Exp, Num, Dig, Op\}$	$\mathcal{N} = \{E, d, b, o, a, c\}$
$T = \{0, 1, +, *\}$	$\mathcal{T} = \{0, 1, +, *, (,)\}$
$P : Exp \rightarrow\ Exp\, Op\, Exp \mid (Exp) \mid Num$	$P : E \rightarrow\ E\, o\, E \mid aEc \mid n$
$Num \rightarrow Dig^{+}$	$d \rightarrow b^{+}$
$Dig \rightarrow 0 \mid 1$	$b \rightarrow 0 \mid 1$
$Op \rightarrow + \mid *$	$o \rightarrow + \mid *$
	$a \rightarrow ($
	$c \rightarrow)$
$S = Exp$	$S = E$

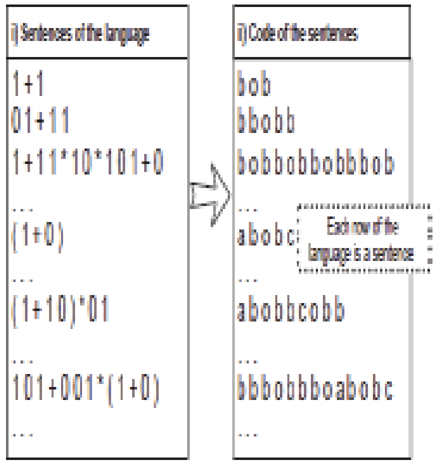


We can modify the CFG (table1) to add new syntactic constructs to ***specify and search for DNA patterns in data***.

Language generation

With the previous criteria, a sample of the language generated by $G_{æ}$ can be seen in the figure 1, point (i).

It is noted that each line corresponds to a sentence accepted by the grammar.



i) Sentences of the language	i) Code of the sentences
1+1	bob
01+11	bbobb
1+11*10*101+0	bobbobbobbbob
...	...
(1+0)	abobc
...	...
(1+10)*01	abobbcobb
...	...
101+001*(1+0)	bbbobbboabobc
...	...

Each row of the language is a sentence

Fig. 1. Language of arithmetic expressions on which its grammar is inferred

Language Codification

Considering the language that is generated with $G_{æ}$,
all the symbols of T can be codified with the symbols of N.

For this particular case the symbols to be used are

$\{b, o, a, c\}$

as syntactic categories.

See figure 1, point (ii).

Incremental Discovery of Sequential Patterns and Associations

The idea consists of finding subsequences,
identifying them with a symbol and
using this symbol to replace the appearances of the
subsequences in the sentences of the population, and
repeating the procedure until each sentence is identified
by a single symbol (see figure 2).

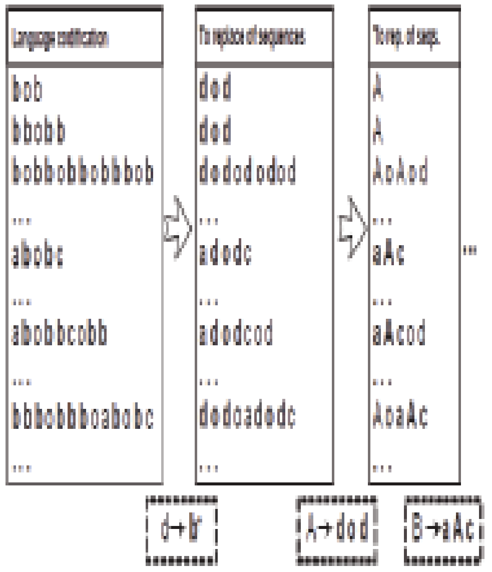
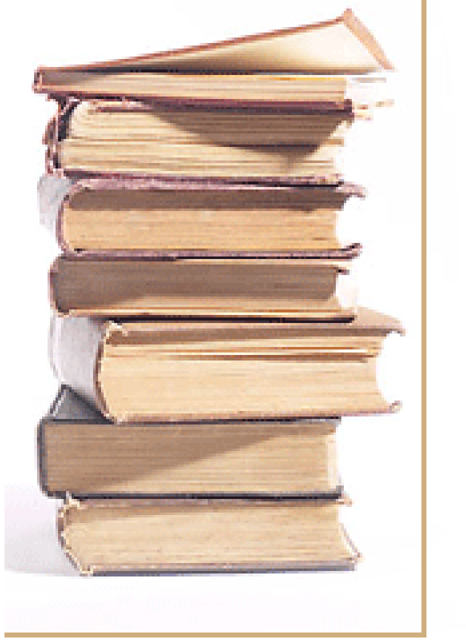


Fig. 2. Hybrid discovery of sequential patterns for the context-free languages



VNiVERSiDAD Ð SALAMANCA

VNiVERSiTAS STVDii SALAMANTiNi

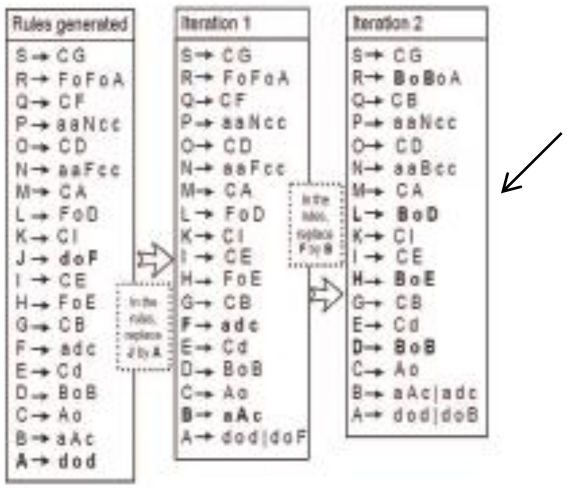
http://dptoia.usal.es

Experiments

Rules Similarity

Using the language ***Læ* of arithmetic expressions** and applying the hybrid algorithm of DSP the production rules of figure 3 were obtained.

The right hand rules form the sequential patterns of the language



Rules generated	Iteration 1	Iteration 2
S → CG	S → CG	S → CG
R → FoFoA	R → FoFoA	R → **BoBoA**
Q → CF	Q → CF	Q → CB
P → aaNcc	P → aaNcc	P → aaNcc
O → CD	O → CD	O → CD
N → aaFcc	N → aaFcc	N → aaBcc
M → CA	M → CA	M → CA
L → FoD	L → FoD	**L → BoD**
K → CI	K → CI	K → CI
J → doF		
I → CE	I → CE	I → CE
H → FoE	H → FoE	**H → BoE**
G → CB	G → CB	G → CB
F → adc	**F → adc**	
E → Cd	E → Cd	E → Cd
D → BoB	D → BoB	**D → BoB**
C → Ao	C → Ao	C → Ao
B → aAc	**B → aAc**	B → aAc|adc
A → dod	A → dod|doF	A → dod|doB

In the rules, replace J by A
In the rules, replace F by B

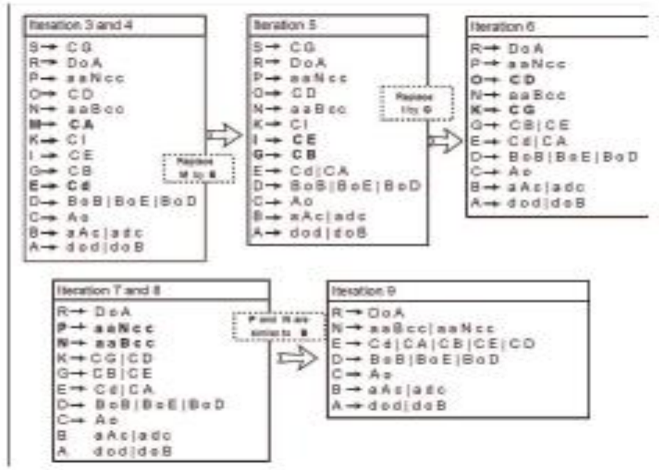
Fig. 3. Production rules generated and some iterations in its simplification

Experiments

Rules Simplification and Compaction

With the right hand part of the productions rules we search for similar sequences to compact them. (**a *substitution matrix is computed).***

This way, the generated rules are simplified and compacted iteratively (figures 3 and 4) until a grammar $G'_{æ}$ is built.



```
Iteration 3 and 4
S→ CG
R→ DoA
P→ aaNcc
O→ CD
N→ aaBcc
M→ CA
K→ CI
I → CE
G→ CB
E→ Cd
D→ BoB|BoE|BoD
C→ Ao
B→ aAc|adc
A→ dod|doB
```

```
Iteration 5
S→ CG
R→ DoA
P→ aaNcc
O→ CD
N→ aaBcc
K→ CI
I → CE
G→ CB
E→ Cd|CA
D→ BoB|BoE|BoD
C→ Ao
B→ aAc|adc
A→ dod|doB
```

```
Iteration 6
R→ DoA
P→ aaNcc
O→ CD
N→ aaBcc
K→ CG
G→ CB|CE
E→ Cd|CA
D→ BoB|BoE|BoD
C→ Ao
B→ aAc|adc
A→ dod|doB
```

```
Iteration 7 and 8
R→ DoA
P→ aaNcc
N→ aaBcc
K→CG|CD
G→CB|CE
E→ Cd|CA
D→ BoB|BoE|BoD
C→ Ao
B   aAc|adc
A   dod|doB
```

```
Iteration 9
R→ DoA
N→ aaBcc|aaNcc
E→ Cd|CA|CB|CE|CD
D→ BoB|BoE|BoD
C→ Ao
B→ aAc|adc
A→ dod|doB
```

Fig. 4. Simplification and compaction of the production rules generated

Experiments

Rules Simplification and Compaction

The grammar $G'_{æ}$ is described in table 2.

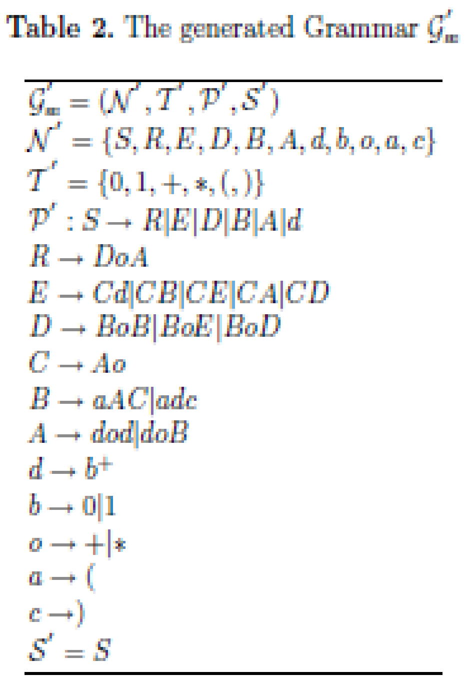


Table 2. The generated Grammar $\mathcal{G}'_{æ}$

$\mathcal{G}'_{æ} = (\mathcal{N}', \mathcal{T}', \mathcal{P}', \mathcal{S}')$

$\mathcal{N}' = \{S, R, E, D, B, A, d, b, o, a, c\}$

$\mathcal{T}' = \{0, 1, +, *, (,)\}$

$\mathcal{P}' : S \rightarrow R|E|D|B|A|d$

$R \rightarrow DoA$

$E \rightarrow Cd|CB|CE|CA|CD$

$D \rightarrow BoB|BoE|BoD$

$C \rightarrow Ao$

$B \rightarrow aAC|adc$

$A \rightarrow dod|doB$

$d \rightarrow b^{+}$

$b \rightarrow 0|1$

$o \rightarrow +|*$

$a \rightarrow ($

$c \rightarrow)$

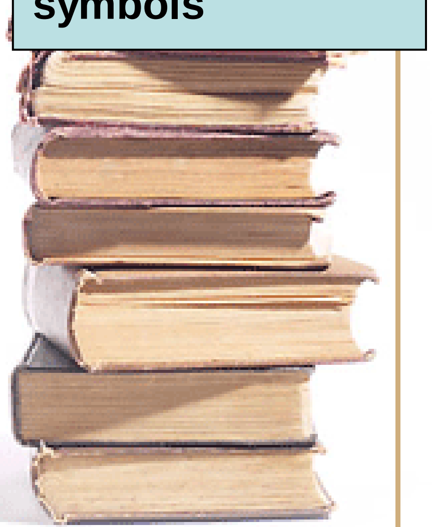
$\mathcal{S}' = S$

For ***Læ*** *it generated* **19 symbols**

A, B,..., S *with* ***d, o, c***

form

***23* non terminal symbols**



http://dptoia.usal.es

VNiVERSiDAD Ð SALAMANCA

VNiVERSiTAS STVDii SALAMANTiNi

Experiments

Practical Results using GAS 1.0

The GAS 1.0 tool provides the basis to create new components in data mining
for the discovery of biological data.

Following this aim, the languages like $\mathcal{L}_{æ}$ created with the grammar $G'_{æ}$ were considered
for automatic design methods to generate analyzers and/or language translators.

In this respect, we used the compiler generator GAS 1.0, to automatically generate **a scanner and a parser** for the language specification.

Experiments

Practical Results using GAS 1.0

Taking as input the grammar specification ***G'***$_{æ}$, the syntactic analysis tables are created, giving as result:

- a Decorated Abstract Syntax Tree (DAST),

  the kind of output that is desired in describing certain biological sequence data

- or the syntactic error.

VNiVERSiDAD D SALAMANCA

VNiVERSiTAS STVDII SALAMANTiNi

Experiments

Practical Results using GAS 1.0

We used a measurement of complexity that can be used in the objective evaluation of the quality of the grammars:

- Number of non terminals: It allows us to measurement of the size of the CFG.

- Ciclomatic complexity: the complexity of McCabeis defined ***V*** of a flow graph ***G***.

  This complexity is defined as

  $$V(G) = A - N + 2$$

  where

  A is the number of edges of the flow graph

  N is the number of nodes.

VNiVERSiDAD
Ð SALAMANCA

VNiVERSiTAS STVDii SALAMANTiNi

Experiments

Practical Results using GAS 1.0

Our approach confirms the idea that the grammar complexity has been applied successfully.

For example for the grammars $G'_{æ}$ ***the complexity is of the order of 9,***

Its high values confirm that the obtained grammars are good, and they can offer

the best results **to the analysis of biosequences, providing sufficient discrimination**

http://dptoia.usal.es

Conclusions

1. In the experiments, a language $L_{æ}$ generated by predetermined CFG $G'_{æ}$ is used. But later none of the properties of that grammar were utilized to generate the set of production rules that then formed the grammar $G'_{æ}$.

2. We have proposed a new method of automatic generation of syntactic categories in a codified language.

3. The approach extends to the processing of data that are believed to have a grammatical structure that could be automatically generated.

Conclusions

5. The tool allows measuring the complexity of the obtained grammar automatically from textual data

6. The tool can be applied to the analysis of DNA sequences

http://dptoia.usal.es

VNiVERSiDAD D SALAMANCA

VNiVERSiTAS STVDIi SALAMANTiNi

VNiVERSiDAD D SALAMANCA

http://dptoia.usal.es

I’m afraid I don’t speak English well enough to answer your questions now, but I’ ll be happy to answer any questions by email (vivian@usal.es).

Thank you for your attention