uni.lu
FACULTY OF SCIENCE,
TECHNOLOGY
AND MEDICINE

Longest Common Subsequences

Seminar 2

Joris LIMONIER

Supervised by George KERCHEV

May 31, 2021

Table of Contents

1. Introduction

1.1 What are LCS ?

1.2 Why are we interested in LCS ?

2. How to find LCS ?

2.1 Step A: Building the table

2.2 Step B: Crawling back up the table

3. Data analysis of LCS results

3.1 Average LCS length

3.2 Normal fit

1. Introduction

1. Introduction

1.1 What are LCS ?

What are LCS ?

Notation

“LCS” = Longest Common Subsequence(s)

What are LCS ?

Notation

"LCS" = Longest Common Subsequence(s)

Example 1

$$S_1 : \quad A \quad B \quad A \quad B \quad B$$

What are LCS ?

Notation

“LCS” = Longest Common Subsequence(s)

Example 1

$$\begin{array}{lccccc} S_1 : & \mathrm{A} & \mathrm{B} & \mathrm{A} & \mathrm{B} & \mathrm{B} \\ S_2 : & \mathrm{A} & \mathrm{A} & \mathrm{B} & \mathrm{A} & \mathrm{B} \end{array}$$

What are LCS ?

Notation

“LCS” = Longest Common Subsequence(s)

Example 1

$$\begin{array}{lccccc} S_1 : & \mathbf{A} & \mathrm{B} & \mathrm{A} & \mathrm{B} & \mathrm{B} \\ S_2 : & \mathbf{A} & \mathrm{A} & \mathrm{B} & \mathrm{A} & \mathrm{B} \end{array}$$

What are LCS ?

Notation

"LCS" = Longest Common Subsequence(s)

Example 1

$$\begin{array}{lccccc}
S_1: & \mathbf{A} & \mathbf{B} & \mathrm{A} & \mathrm{B} & \mathrm{B} \\
S_2: & \mathbf{A} & \mathrm{A} & \mathbf{B} & \mathrm{A} & \mathrm{B}
\end{array}$$

What are LCS ?

Notation

"LCS" = Longest Common Subsequence(s)

Example 1

S_1 : **A** **B** **A** B B

S_2 : **A** A **B** **A** B

What are LCS ?

Notation

“LCS” = Longest Common Subsequence(s)

Example 1

S_1 :	**A**	**B**	**A**	**B**	B
S_2 :	**A**	A	**B**	**A**	**B**

What are LCS ?

Notation

"LCS" = Longest Common Subsequence(s)

Example 1

S_1 : **A** **B** **A** **B** B

S_2 : **A** A **B** **A** **B**

$\Longrightarrow$ The LCS between S_1 and S_2 is **A B A B**

What are LCS ?

Notation

“LCS” = Longest Common Subsequence(s)

Example 1

S_1 : **A** **B** **A** **B** B

S_2 : **A** A **B** **A** **B**

$\Longrightarrow$ The LCS between S_1 and S_2 is **A B A B**

NB: LCS may not be unique, A A B B also works.

Example 2

What is the LCS of the following sequences ?

Example 2

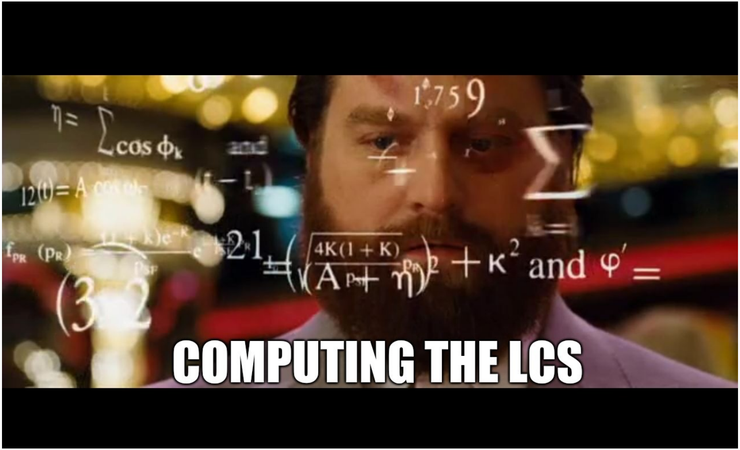
What is the LCS of the following sequences ?

S_3 :
AABBAABAAABABAAABAAABBABAAABAAABAAA
AAAABBABBBBAAABABBAABBAABBBBBAAAABA
BBABAAAABABAABBBABABBBBBAAABBBAABBB
AABAABBABABAABABBBBBBAABBBBBBAAAAAB
AABAAAAAABAABAABAAABBABBBBABBAAAABBB

S_4 :
BABBBABAABAABBBABBABBBBBBBABABAAABB
BBABBABBABBBABBBABBABBABABABBAABABA
BAABABAAAABABBABABBAAAABABBAABABABB
BABBBBBBAABAAABBABBBBAAAAABBBBBAAAB
ABBAAAABBBABABAABBABBBAABABBBABAABA

Example 2

What is the LCS of the following sequences ?



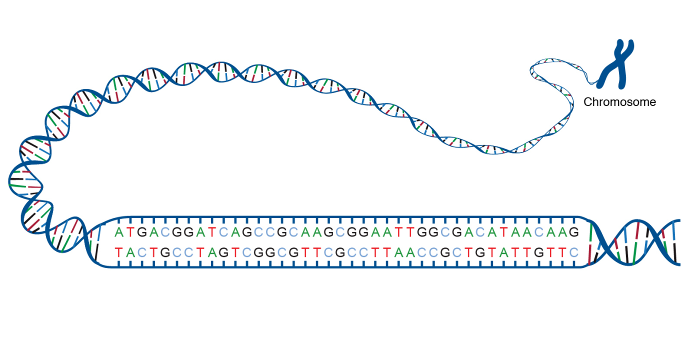
1. Introduction

1.2 Why are we interested in LCS ?

Applications

Applications

- Bioinformatics: Compare sequences of nucleotides (DNA)

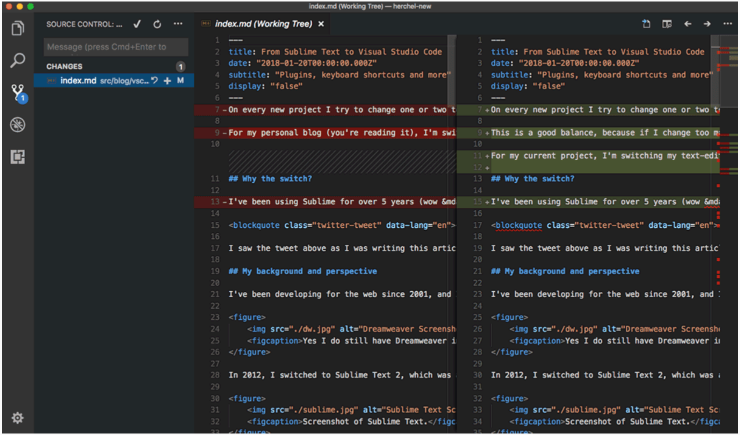

Chromosome
ATGACGGATCAGCCGCAAGCGGAATTGGCGACATAACAAG
TACTGCCTAGTCGGCGTTCGCCTTAACCGCTGTATTGTTC

Applications

- Bioinformatics: Compare sequences of nucleotides (DNA)
- Natural Language Processing: Compare texts

Applications

- Bioinformatics: Compare sequences of nucleotides (DNA)
- Natural Language Processing: Compare texts
- Computer Science: Detect differences in texts



index.md (Working Tree) — herchel-new
SOURCE CONTROL: ...
Message (press Cmd+Enter to
CHANGES 1
index.md src/blog/vsc... M
index.md (Working Tree) ×

```
---
title: From Sublime Text to Visual Studio Code
date: "2018-01-20T00:00:00.000Z"
subtitle: "Plugins, keyboard shortcuts and more"
display: "false"
---
-On every new project I try to change one or two t

-For my personal blog (you're reading it), I'm swi


## Why the switch?

-I've been using Sublime for over 5 years (wow &md

<blockquote class="twitter-tweet" data-lang="en">

I saw the tweet above as I was writing this artic

## My background and perspective

I've been developing for the web since 2001, and

<figure>
    <img src="./dw.jpg" alt="Dreamweaver Screensh
    <figcaption>Yes I do still have Dreamweaver i
</figure>

In 2012, I switched to Sublime Text 2, which was

<figure>
    <img src="./sublime.jpg" alt="Sublime Text Sc
    <figcaption>Screenshot of Sublime Text.</figc
</figure>
```

```
---
title: From Sublime Text to Visual Studio Code
date: "2018-01-20T00:00:00.000Z"
subtitle: "Plugins, keyboard shortcuts and more"
display: "false"
---
+On every new project I try to change one or two t

+This is a good balance, because if I change too m

+For my current project, I'm switching my text-edi
+
## Why the switch?

+I've been using Sublime for over 5 years (wow &md

<blockquote class="twitter-tweet" data-lang="en">

I saw the tweet above as I was writing this artic

## My background and perspective

I've been developing for the web since 2001, and

<figure>
    <img src="./dw.jpg" alt="Dreamweaver Screensh
    <figcaption>Yes I do still have Dreamweaver i
</figure>

In 2012, I switched to Sublime Text 2, which was

<figure>
    <img src="./sublime.jpg" alt="Sublime Text Sc
    <figcaption>Screenshot of Sublime Text.</figc
</figure>
```

2. How to find LCS ?

2. How to find LCS ?

2.1 Step A: Building the table

Set-up

Let $S_1 = ABABB$ and $S_2 = AABAB$.

Set-up

Let $S_1 = ABABB$ and $S_2 = AABAB$.

- Make a table where S_1 and S_2 are the column and row names respectively.

	A	B	A	B	B
A					
A					
B					
A					
B					

Set-up

Let $S_1 = ABABB$ and $S_2 = AABAB$.

- Make a table where S_1 and S_2 are the column and row names respectively.
- Add a row (resp. column) at the top (resp. left) of the table. Fill them with 0's.

	∅	A	B	A	B	B
∅	0	0	0	0	0	0
A	0					
A	0					
B	0					
A	0					
B	0					

Procedure

Start from top-left corner. Move left to right, line by line.

- If row and column names match, increment adjascent top-left-diagonal cell by 1.
- Else take the maximum of top and left cells.

	∅	A	B	A	B	B
∅	0	0	0	0	0	0
A	0					
A	0					
B	0					
A	0					
B	0					

Procedure

Start from top-left corner. Move left to right, line by line.

- If row and column names match, increment adjacent top-left-diagonal cell by 1.
- Else take the maximum of top and left cells.

	∅	A	B	A	B	B
∅	0	0	0	0	0	0
A	0	1				
A	0					
B	0					
A	0					
B	0					

Procedure

Start from top-left corner. Move left to right, line by line.

- If row and column names match, increment adjacent top-left-diagonal cell by 1.
- Else take the maximum of top and left cells.

	∅	A	B	A	B	B
∅	0	0	0	0	0	0
A	0	1	1			
A	0					
B	0					
A	0					
B	0					

Procedure

Start from top-left corner. Move left to right, line by line.

- If row and column names match, increment adjacent top-left-diagonal cell by 1.
- Else take the maximum of top and left cells.

	∅	A	B	A	B	B
∅	0	0	0	0	0	0
A	0	1	1	1		
A	0					
B	0					
A	0					
B	0					

Procedure

Start from top-left corner. Move left to right, line by line.

- If row and column names match, increment adjascent top-left-diagonal cell by 1.
- Else take the maximum of top and left cells.

	∅	A	B	A	B	B
∅	0	0	0	0	0	0
A	0	1	1	1	1	
A	0					
B	0					
A	0					
B	0					

Procedure

Start from top-left corner. Move left to right, line by line.

- If row and column names match, increment adjascent top-left-diagonal cell by 1.
- Else take the maximum of top and left cells.

	∅	A	B	A	B	B
∅	0	0	0	0	0	0
A	0	1	1	1	1	1
A	0					
B	0					
A	0					
B	0					

Procedure

Start from top-left corner. Move left to right, line by line.

- If row and column names match, increment adjascent top-left-diagonal cell by 1.
- Else take the maximum of top and left cells.

	∅	A	B	A	B	B
∅	0	0	0	0	0	0
A	0	1	1	1	1	1
A	0	1				
B	0					
A	0					
B	0					

Procedure

Start from top-left corner. Move left to right, line by line.

- If row and column names match, increment adjascent top-left-diagonal cell by 1.
- Else take the maximum of top and left cells.

	∅	A	B	A	B	B
∅	0	0	0	0	0	0
A	0	1	1	1	1	1
A	0	1	1			
B	0					
A	0					
B	0					

Procedure

Start from top-left corner. Move left to right, line by line.

- If row and column names match, increment adjascent top-left-diagonal cell by 1.
- Else take the maximum of top and left cells.

	∅	A	B	A	B	B
∅	0	0	0	0	0	0
A	0	1	1	1	1	1
A	0	1	1	2		
B	0					
A	0					
B	0					

Procedure

Start from top-left corner. Move left to right, line by line.

- If row and column names match, increment adjascent top-left-diagonal cell by 1.
- Else take the maximum of top and left cells.

	∅	A	B	A	B	B
∅	0	0	0	0	0	0
A	0	1	1	1	1	1
A	0	1	1	2	2	2
B	0					
A	0					
B	0					

Procedure

Start from top-left corner. Move left to right, line by line.

- If row and column names match, increment adjascent top-left-diagonal cell by 1.
- Else take the maximum of top and left cells.

	∅	A	B	A	B	B
∅	0	0	0	0	0	0
A	0	1	1	1	1	1
A	0	1	1	2	2	2
B	0	1	2	2	3	3
A	0					
B	0					

Procedure

Start from top-left corner. Move left to right, line by line.

- If row and column names match, increment adjascent top-left-diagonal cell by 1.
- Else take the maximum of top and left cells.

	∅	A	B	A	B	B
∅	0	0	0	0	0	0
A	0	1	1	1	1	1
A	0	1	1	2	2	2
B	0	1	2	2	3	3
A	0	1	2	3	3	3
B	0					

Procedure

Start from top-left corner. Move left to right, line by line.

- If row and column names match, increment adjacent top-left-diagonal cell by 1.
- Else take the maximum of top and left cells.

	∅	A	B	A	B	B
∅	0	0	0	0	0	0
A	0	1	1	1	1	1
A	0	1	1	2	2	2
B	0	1	2	2	3	3
A	0	1	2	3	3	3
B	0	1	2	3	4	4

$\Longrightarrow$ **The length of the LCS is 4.**

2. How to find LCS ?

2.2 Step B: Crawling back up the table

Procedure

From the table, deduce LCS by starting from the bottom-right cell. Compare cell value with values of top and left cells.

- If cell value $\in$ {top cell value, left cell value}, move to the one with maximum value.
- Else, add character to LCS and move 1 cell diagonally top-left.

	$\varnothing$	A	B	A	B	B
$\varnothing$	0	0	0	0	0	0
A	0	1	1	1	1	1
A	0	1	1	2	2	2
B	0	1	2	2	3	3
A	0	1	2	3	3	3
B	0	1	2	3	4	4

Procedure

From the table, deduce LCS by starting from the bottom-right cell. Compare cell value with values of top and left cells.

- If cell value $\in$ {top cell value, left cell value}, move to the one with maximum value.
- Else, add character to LCS and move 1 cell diagonally top-left.

	∅	A	B	A	B	B
∅	0	0	0	0	0	0
A	0	1	1	1	1	1
A	0	1	1	2	2	2
B	0	1	2	2	3	3
A	0	1	2	3	3	3
B	0	1	2	3	4	4

LCS : __ __ __ __

Procedure

From the table, deduce LCS by starting from the bottom-right cell. Compare cell value with values of top and left cells.

- If cell value $\in$ {top cell value, left cell value}, move to the one with maximum value.
- Else, add character to LCS and move 1 cell diagonally top-left.

	∅	A	B	A	B	B
∅	0	0	0	0	0	0
A	0	1	1	1	1	1
A	0	1	1	2	2	2
B	0	1	2	2	3	3
A	0	1	2	3	3	3
B	0	1	2	3	4	4

LCS : __ __ __ __

Procedure

From the table, deduce LCS by starting from the bottom-right cell. Compare cell value with values of top and left cells.

- If cell value $\in$ {top cell value, left cell value}, move to the one with maximum value.
- Else, add character to LCS and move 1 cell diagonally top-left.

	∅	A	B	A	B	B
∅	0	0	0	0	0	0
A	0	1	1	1	1	1
A	0	1	1	2	2	2
B	0	1	2	2	3	3
A	0	1	2	3	3	3
B	0	1	2	3	**4**	**4**

LCS : __ __ __ __

Procedure

From the table, deduce LCS by starting from the bottom-right cell. Compare cell value with values of top and left cells.

- If cell value $\in$ {top cell value, left cell value}, move to the one with maximum value.
- Else, add character to LCS and move 1 cell diagonally top-left.

	∅	A	B	A	B	B
∅	0	0	0	0	0	0
A	0	1	1	1	1	1
A	0	1	1	2	2	2
B	0	1	2	2	3	3
A	0	1	2	**3**	3	3
B	0	1	2	3	**4**	**4**

LCS : __ __ __ **B**

Procedure

From the table, deduce LCS by starting from the bottom-right cell. Compare cell value with values of top and left cells.

- If cell value $\in$ {top cell value, left cell value}, move to the one with maximum value.
- Else, add character to LCS and move 1 cell diagonally top-left.

	∅	A	B	A	B	B
∅	0	0	0	0	0	0
A	0	1	1	1	1	1
A	0	1	1	2	2	2
B	0	1	**2**	2	3	3
A	0	1	2	**3**	3	3
B	0	1	2	3	**4**	**4**

LCS : __ __ **A** **B**

Procedure

From the table, deduce LCS by starting from the bottom-right cell. Compare cell value with values of top and left cells.

- If cell value ∈ {top cell value, left cell value}, move to the one with maximum value.
- Else, add character to LCS and move 1 cell diagonally top-left.

	∅	A	B	A	B	B
∅	0	0	0	0	0	0
A	0	1	1	1	1	1
A	0	1	1	2	2	2
B	0	1	**2**	2	3	3
A	0	1	2	**3**	3	3
B	0	1	2	3	**4**	4

LCS : __ **B** **A** **B**

Procedure

From the table, deduce LCS by starting from the bottom-right cell. Compare cell value with values of top and left cells.

- If cell value $\in$ {top cell value, left cell value}, move to the one with maximum value.
- Else, add character to LCS and move 1 cell diagonally top-left.

	∅	A	B	A	B	B
∅	0	0	0	0	0	0
A	0	**1**	1	1	1	1
A	0	**1**	1	2	2	2
B	0	1	**2**	2	3	3
A	0	1	2	**3**	3	3
B	0	1	2	3	**4**	**4**

LCS : __ **B** **A** **B**

Procedure

From the table, deduce LCS by starting from the bottom-right cell. Compare cell value with values of top and left cells.

- If cell value $\in$ {top cell value, left cell value}, move to the one with maximum value.
- Else, add character to LCS and move 1 cell diagonally top-left.

	∅	A	B	A	B	B
∅	**0**	0	0	0	0	0
A	0	**1**	1	1	1	1
A	0	1	1	2	2	2
B	0	1	**2**	2	3	3
A	0	1	2	**3**	3	3
B	0	1	2	3	**4**	**4**

LCS : **A B A B**

3. Data analysis of LCS results

3. Data analysis of LCS results

3.1 Average LCS length

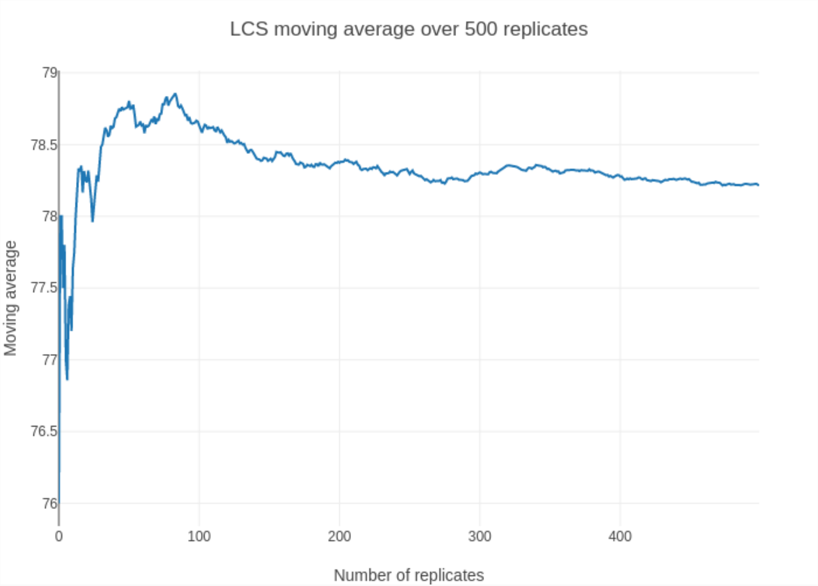
Question

Given two sequences of the same length, what percentage do they have in common ?

Question

Given two sequences of the same length, what percentage do they have in common ?

Answer: $\approx 80\%$


LCS moving average over 500 replicates
Moving average
79
78.5
78
77.5
77
76.5
76
0
100
200
300
400
Number of replicates

Superadditivity

Let L_n be the length of the LCS between two sequences of length n.

Proposition

$(\mathbb{E}[L_n])_{n\geq 1}$ is a superadditive sequence, that is

$$\mathbb{E}[L_{m+n}] \geq \mathbb{E}[L_m] + \mathbb{E}[L_n]$$

Intuition:

S_1:	**A**	**B**	**A**	**B**	B
S_2:	**A**	A	**B**	**A**	**B**

LCS: **ABAB**

Length: **4**

Intuition:

S_1:	**A**	**B**	**A**	**B**	B	S_3:	**A**	B
S_2:	**A**	A	**B**	**A**	**B**	S_4:	B	**A**

LCS:	**ABAB**	**A**
Length:	**4**	**1**

Intuition:

A	**B**	**A**	**B**	B	**A**	B
A	A	**B**	**A**	**B**	B	**A**

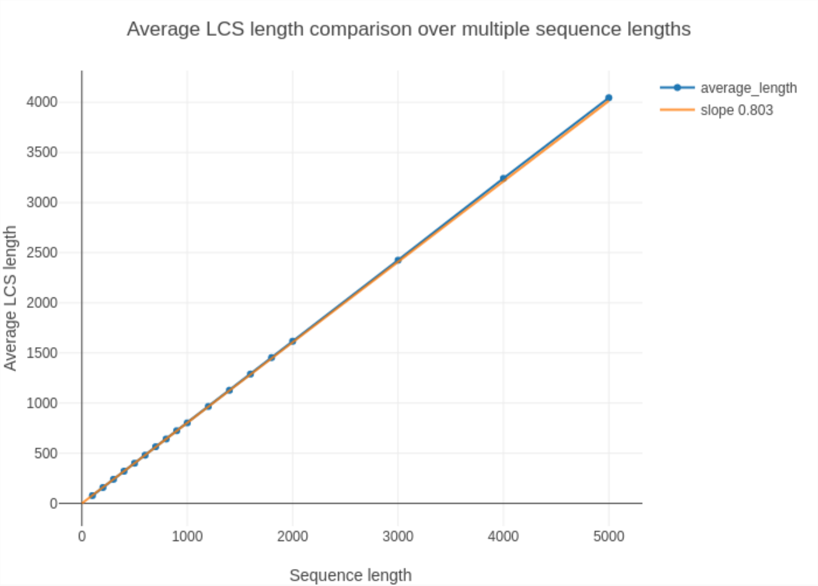
LCS:	**ABAB**	**A**
Length:	**4**	**1**

Intuition:

A	**B**	**A**	**B**	**B**	**A**	B
A	A	**B**	**A**	**B**	**B**	**A**

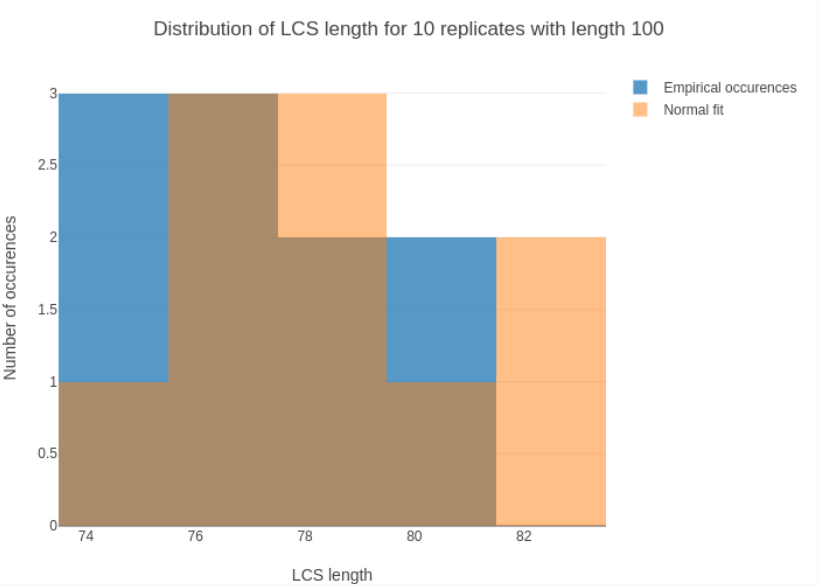
LCS: **ABABBA**

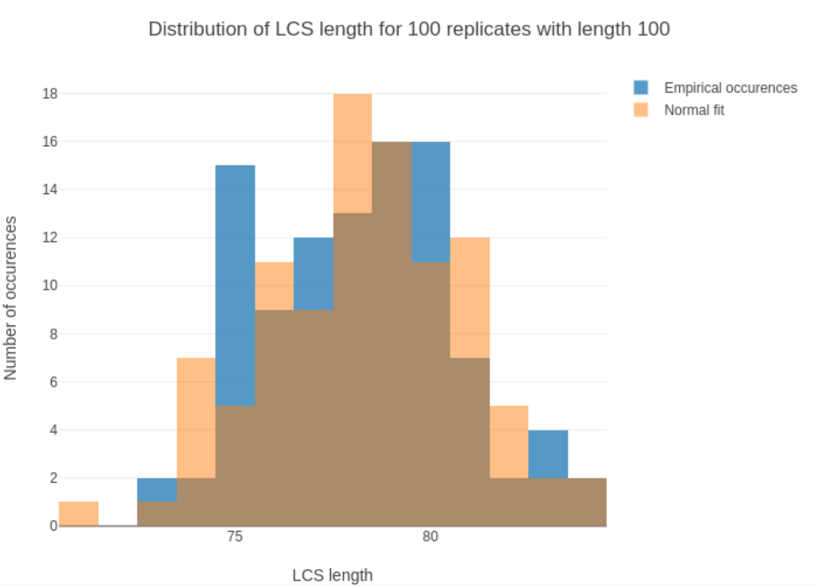
Length: **6**

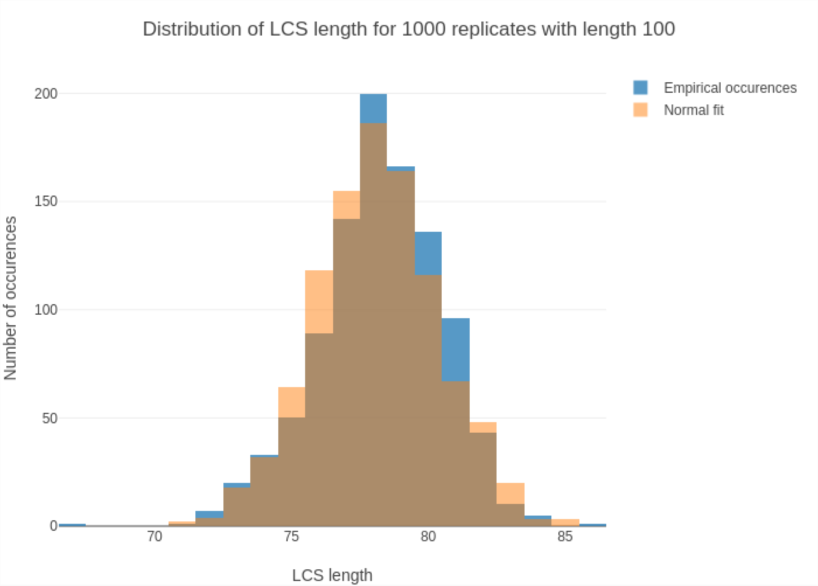

Average LCS length comparison over multiple sequence lengths
Average LCS length
Sequence length
0
500
1000
1500
2000
2500
3000
3500
4000
0
1000
2000
3000
4000
5000
average_length
slope 0.803

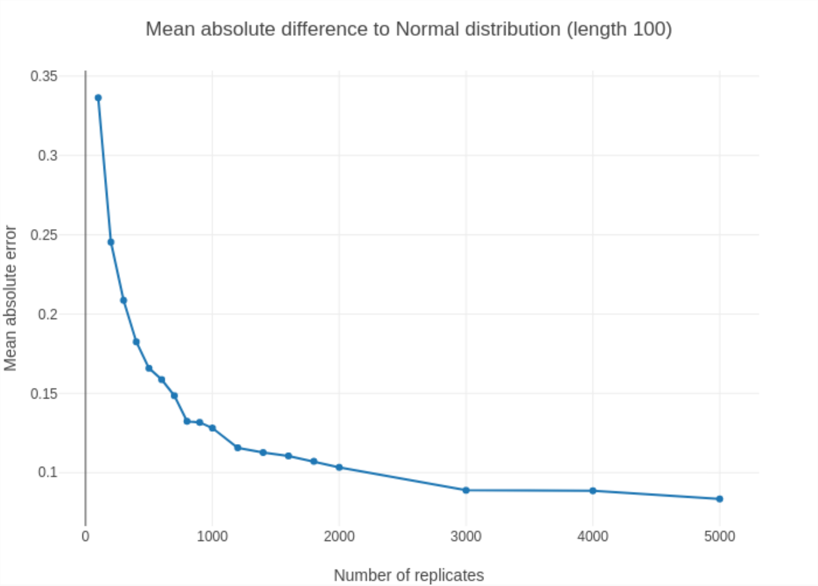
3. Data analysis of LCS results

3.2 Normal fit


Distribution of LCS length for 10 replicates with length 100
Empirical occurences
Normal fit
Number of occurences
0
0.5
1
1.5
2
2.5
3
74
76
78
80
82
LCS length


Distribution of LCS length for 100 replicates with length 100
Empirical occurences
Normal fit
Number of occurences
0
2
4
6
8
10
12
14
16
18
75
80
LCS length


Distribution of LCS length for 1000 replicates with length 100
Empirical occurences
Normal fit
Number of occurences
0
50
100
150
200
70
75
80
85
LCS length


Mean absolute difference to Normal distribution (length 100)
Mean absolute error
0.35
0.3
0.25
0.2
0.15
0.1
0
1000
2000
3000
4000
5000
Number of replicates

Thank you

Questions?

https://github.com/jorislimonier/LCS