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. 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 : A B A B B

What are LCS ?

Notation

"LCS" = Longest Common Subsequence(s)

Example 1

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

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

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

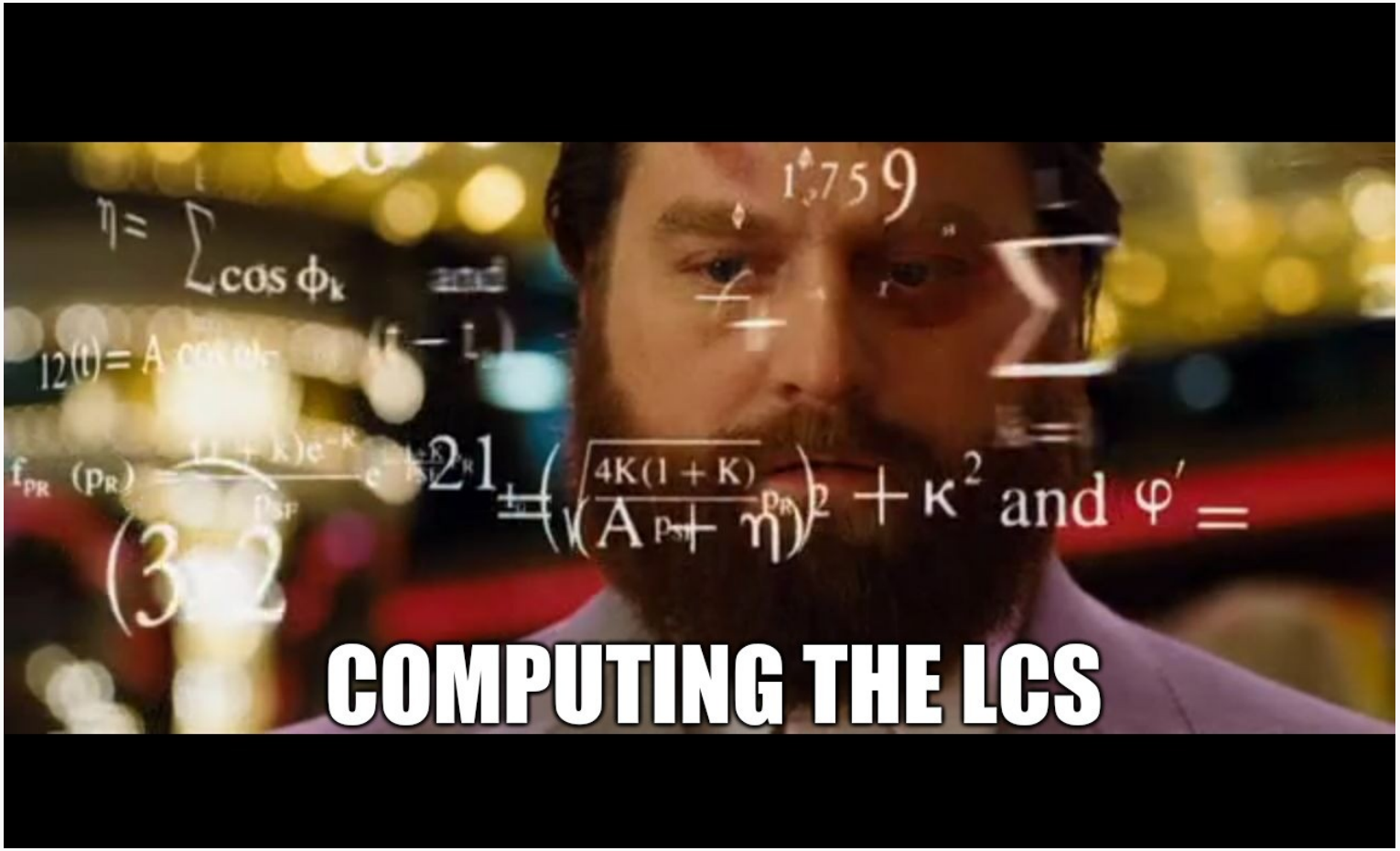
What is the LCS of the following sequences ?

S_3 : AABBAABAAABABAAABAAABBABAAABAAABAAA
AAAABBABBBBAAABABBAABBAABBBBBAAAABA
BBABAAAABABAABBBABABBBBBAAABBBAABBB
AABAABBABABAABABBBBBBAABBBBBBBAAAAAB
AABAAAAABAABAABAAABBABBBBABBAAAABBB

S_4 : BABBBABAABAABBBABBABBBBBBBABABAAABB
BBABBABBABBBABBBBABBABBABABABBAABABA
BAABABAAAABABBABABBAAAABABBAABABABB
BABBBBBBAABAAABBABBBBAAAAABBBBBAAAB
ABBAAAABBBABABAABBABBBAABABBBABAABA

Example 2

What is the LCS of the following sequences ?



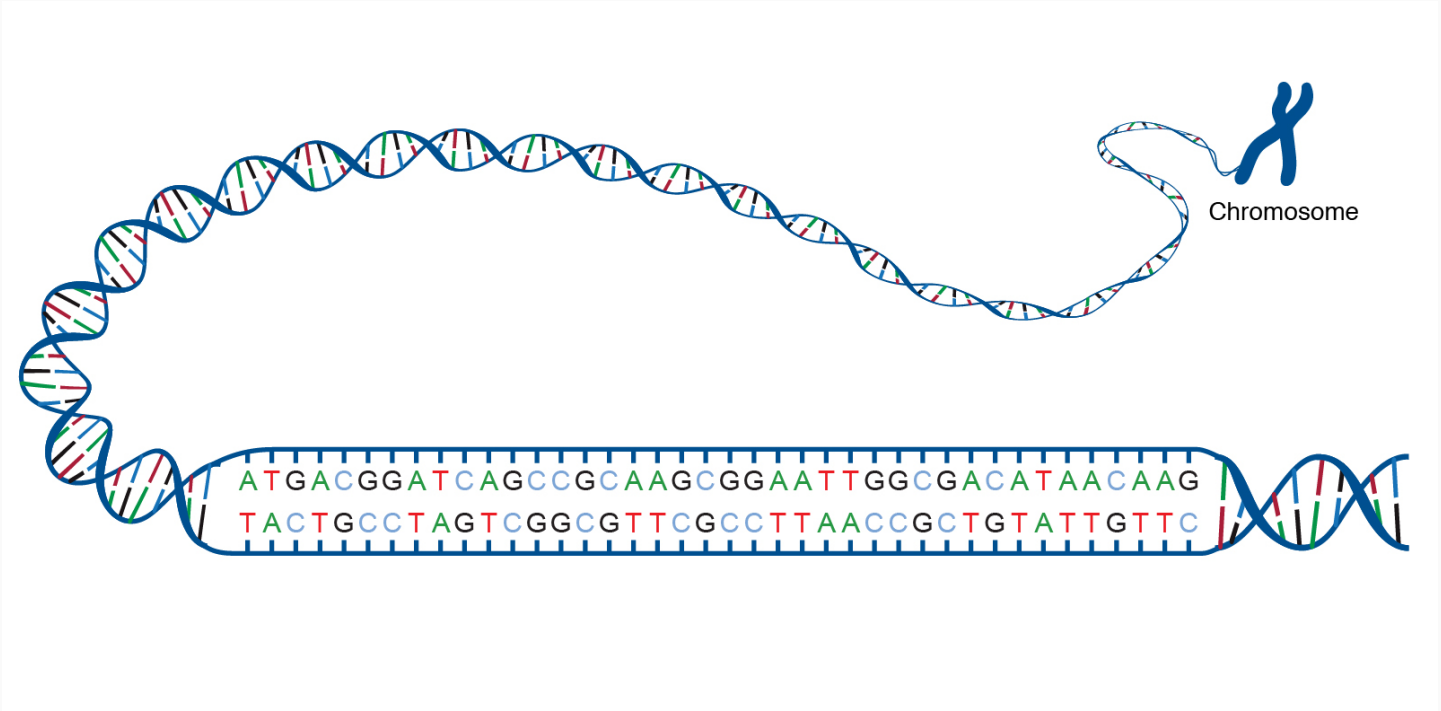
1. Introduction

1.2 Why are we interested in LCS ?

Applications

Applications

- Bioinformatics: Compare sequences of nucleotides (DNA)

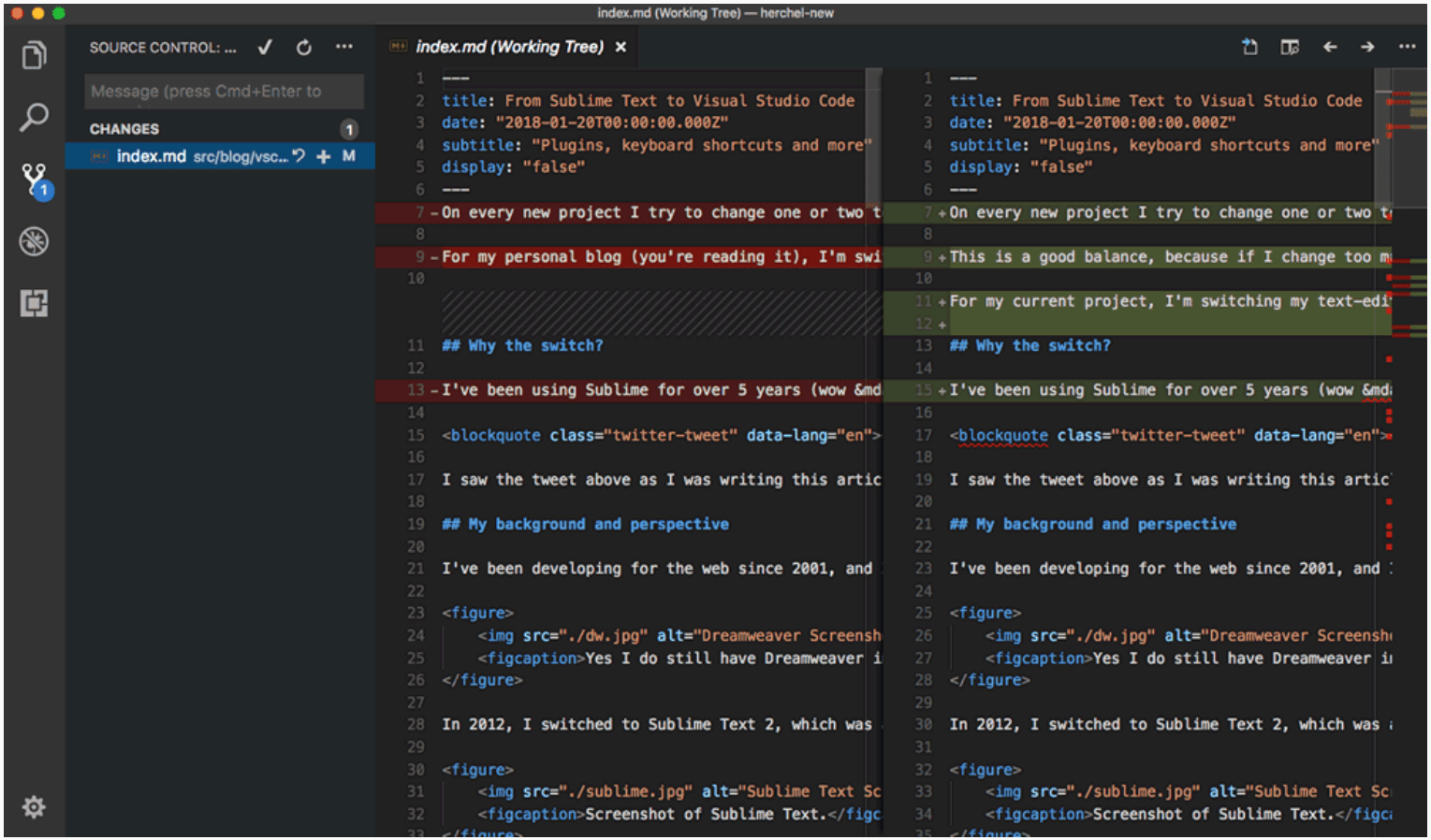


Applications

- 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



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.

	$\varnothing$	A	B	A	B	B
$\varnothing$	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 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				
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			
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		
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 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	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.

	∅	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

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 $\in$ {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 : _ _ **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

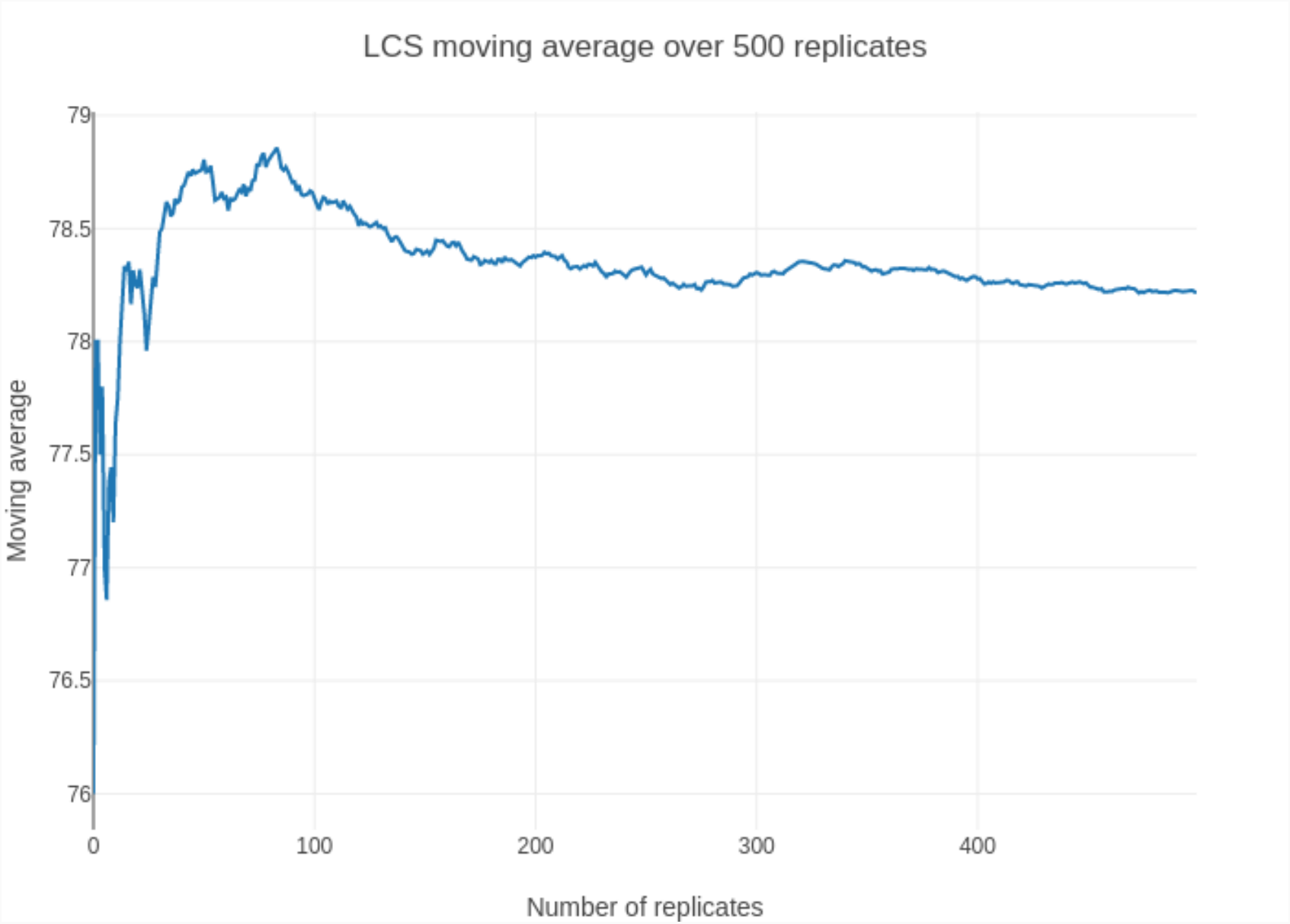
Question

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**

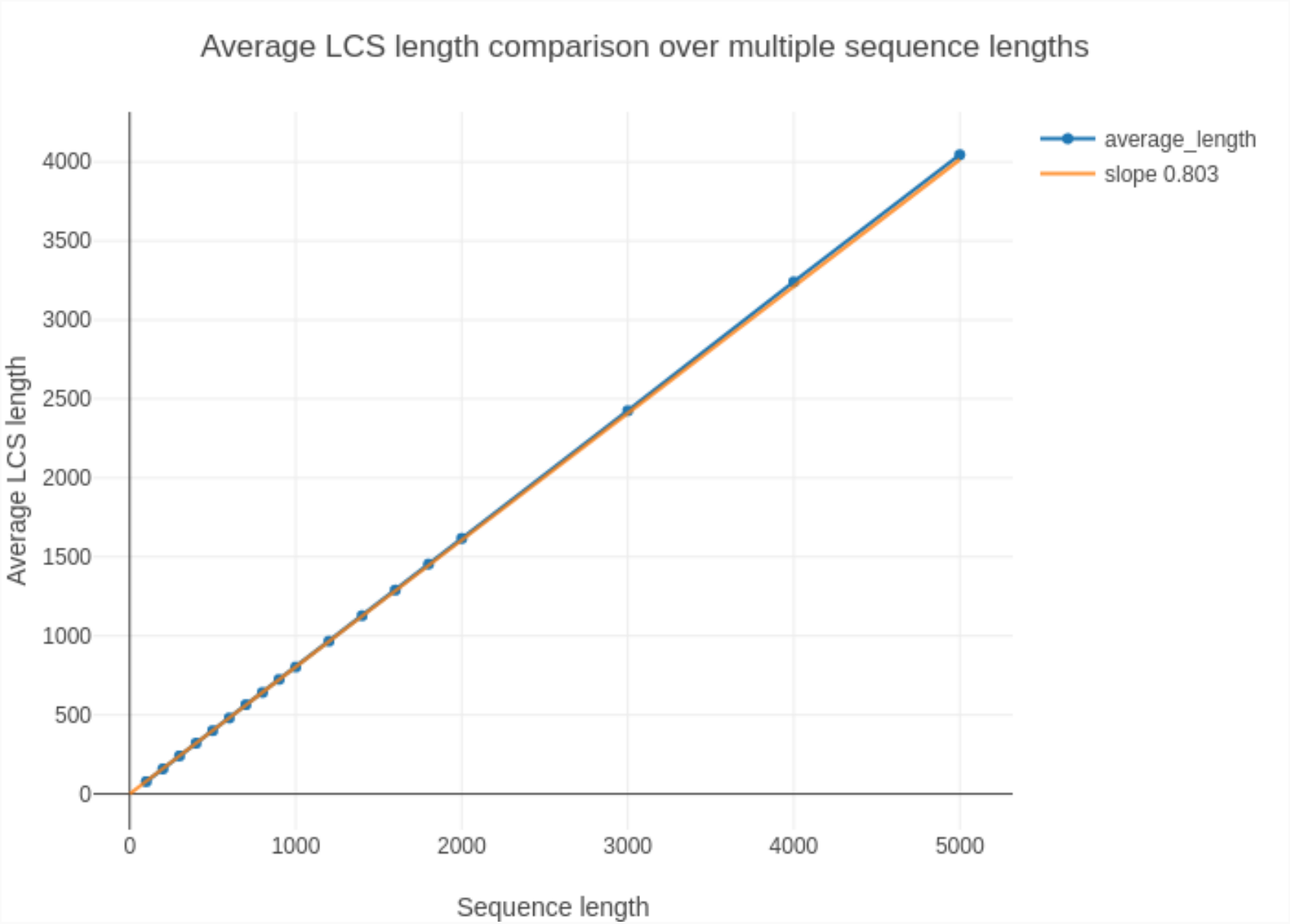
LCS:	**ABAB**	**A**
Length:	**4**	**1**

Intuition:

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

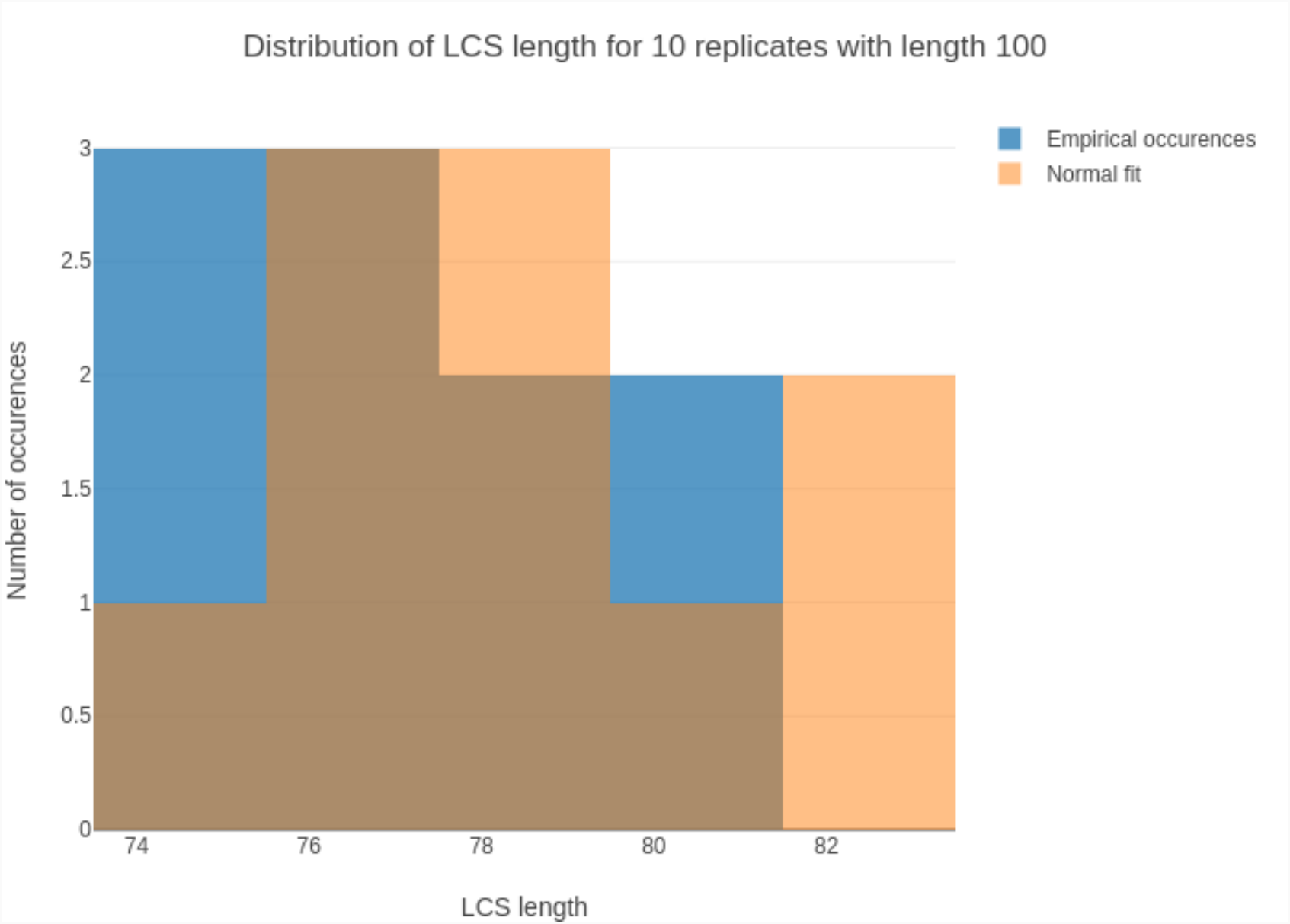
LCS: **ABABBA**

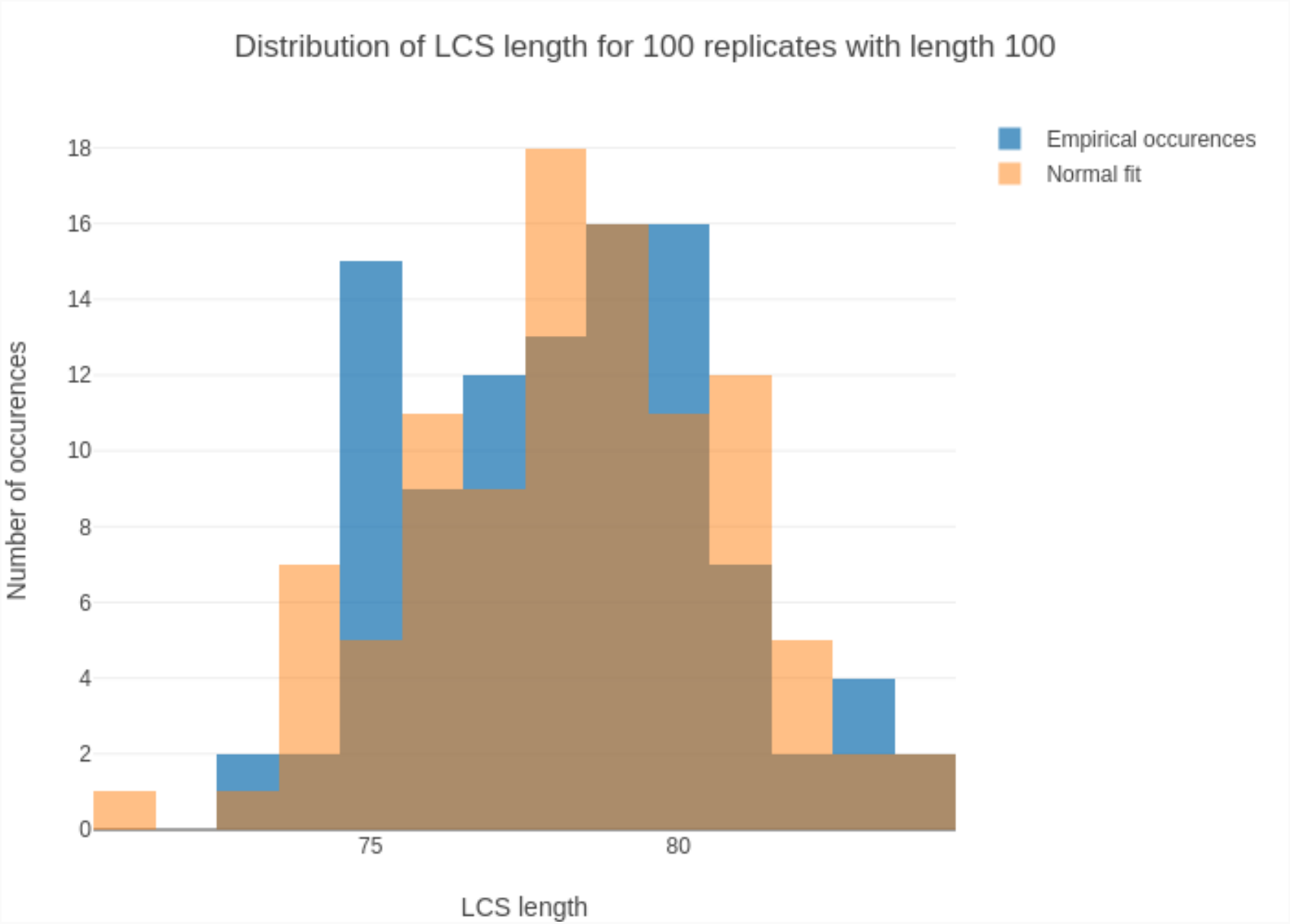
Length: **6**

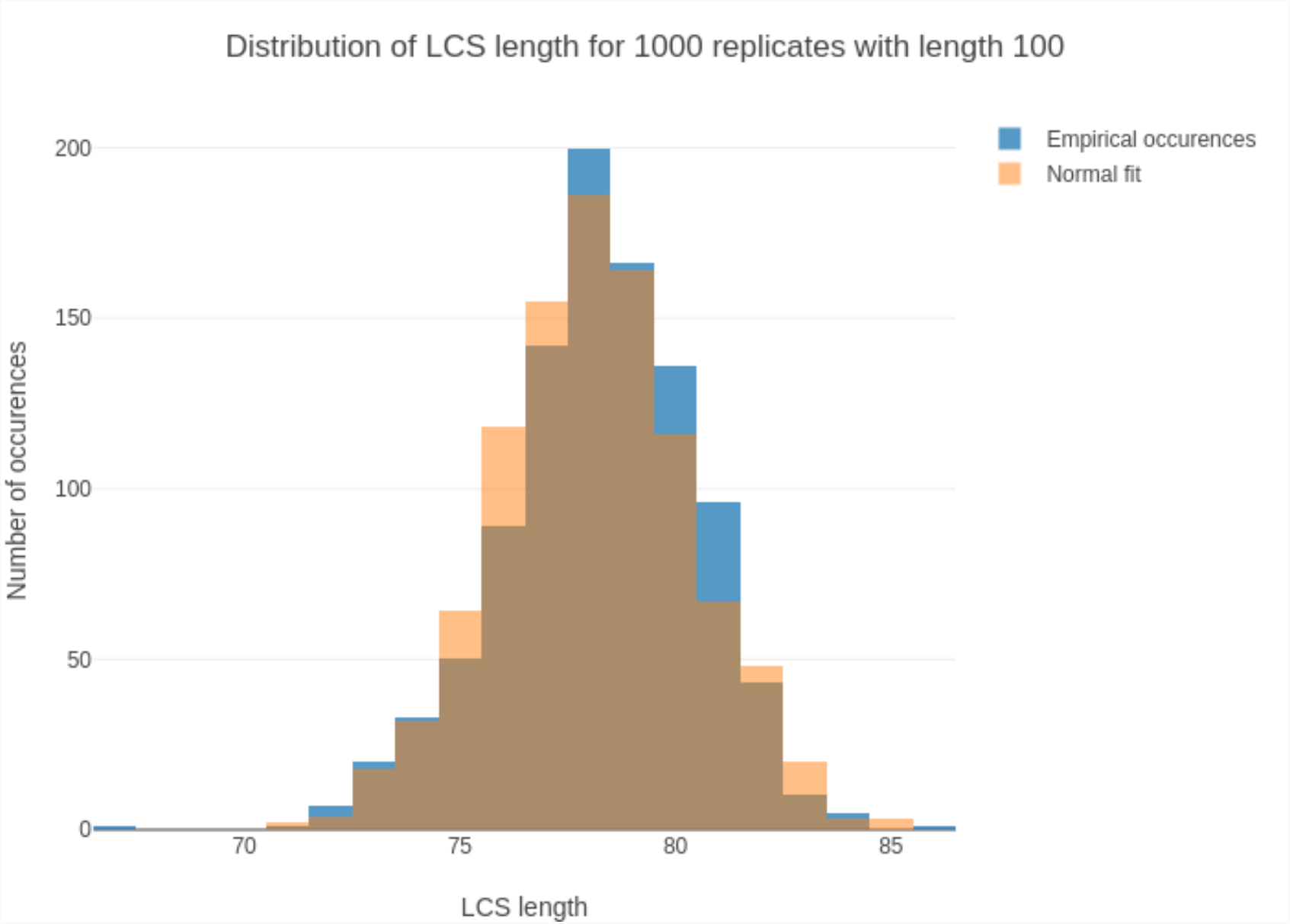

Average LCS length comparison over multiple sequence lengths
average_length
slope 0.803
Average LCS length
Sequence length
0
500
1000
1500
2000
2500
3000
3500
4000
0
1000
2000
3000
4000
5000

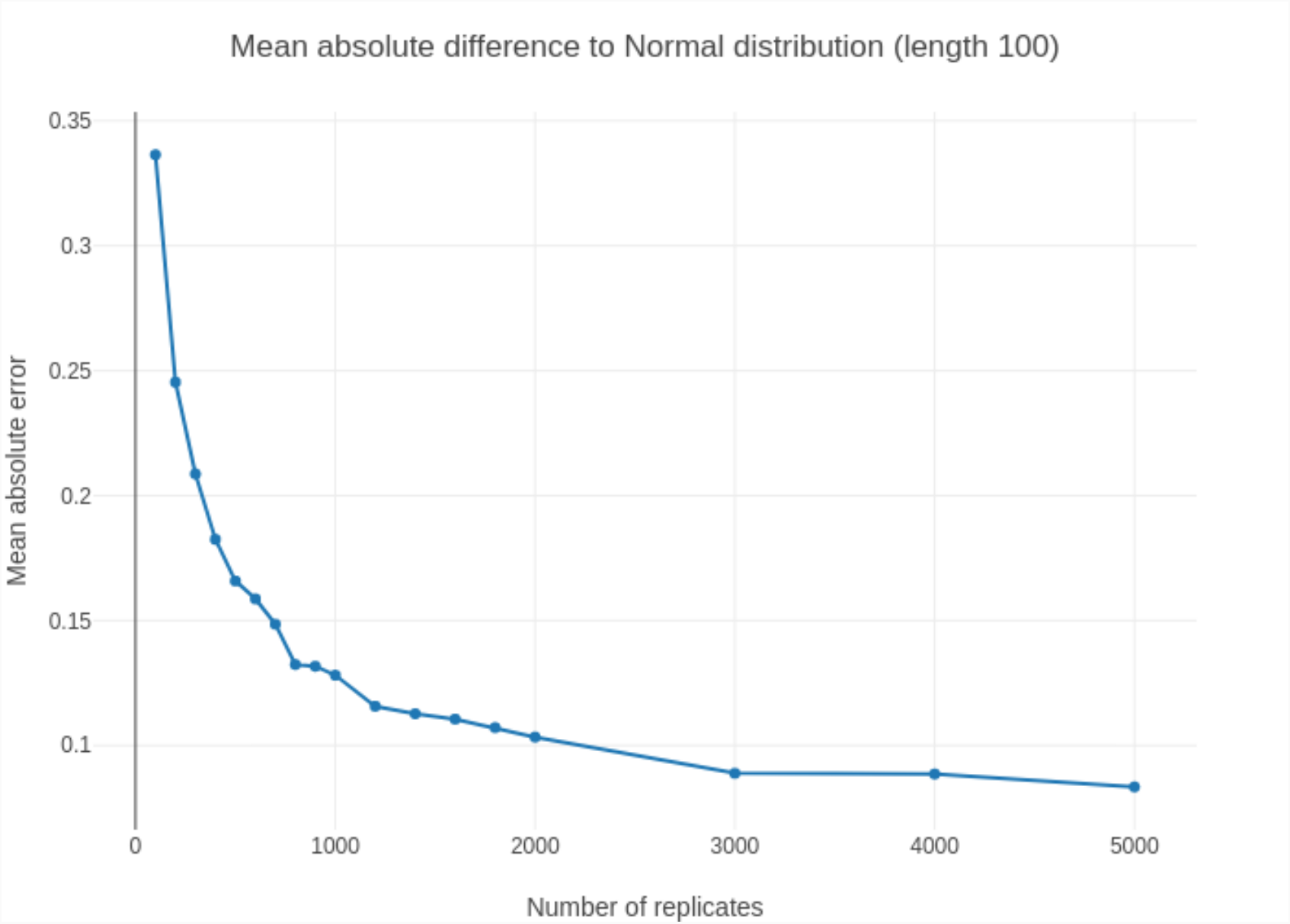
3. Data analysis of LCS results

3.2 Normal fit


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


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


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