



# Longest Common Subsequences

## Seminar 2

---

Joris LIMONIER

May 31, 2021

*Supervised by George KERCHEV*

# 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

$S_1$  :     A   B   A   B   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

|         |          |          |          |   |   |
|---------|----------|----------|----------|---|---|
| $S_1$ : | <b>A</b> | <b>B</b> | A        | B | B |
| $S_2$ : | <b>A</b> | A        | <b>B</b> | A | B |

# What are LCS ?

## Notation

“LCS” = Longest Common Subsequence(s)

## Example 1

|         |          |          |          |          |   |
|---------|----------|----------|----------|----------|---|
| $S_1$ : | <b>A</b> | <b>B</b> | <b>A</b> | B        | B |
| $S_2$ : | <b>A</b> | A        | <b>B</b> | <b>A</b> | B |

# What are LCS ?

## Notation

“LCS” = Longest Common Subsequence(s)

## Example 1

|         |          |          |          |          |          |
|---------|----------|----------|----------|----------|----------|
| $S_1$ : | <b>A</b> | <b>B</b> | <b>A</b> | <b>B</b> | B        |
| $S_2$ : | <b>A</b> | A        | <b>B</b> | <b>A</b> | <b>B</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**

$\Rightarrow$  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**

⇒ 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

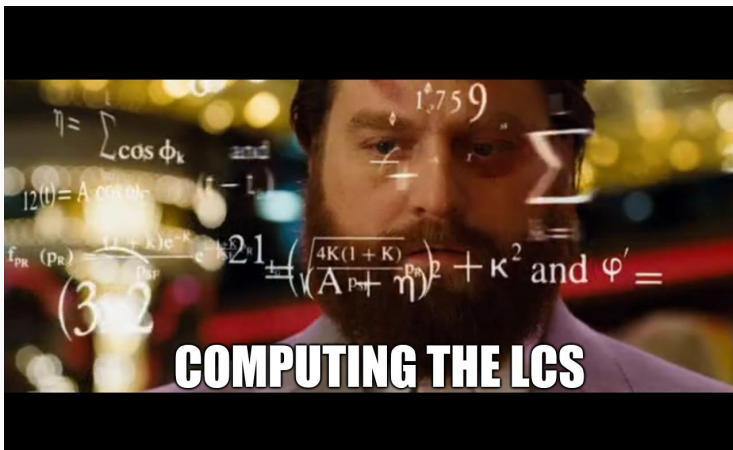
What is the LCS of the following sequences ?

$S_3$  : AABBAABAAABABAAABAAABBABAAABAAABAAA  
AAAABBABBBBAAABABBAABBAABBBBBBAAAABA  
BBABAAAABABAABBBBABBBBBBAAABBBBAABBB  
AABAABBABABAABABBBBBBBAABBBBBBAAAAAB  
AABAAAAABAABAABAAABBABBBBABBAAAABBB

$S_4$  : BABBBABAABAABBBABBABBBBBBBBABABAAABB  
BBABBABBABBBABBBABBABBABABABBAABABA  
BAABABAAAABABBABABBAABABBAABABABB  
BABBBBBBBAABAABBBABBBBBAAAAABBBBBBAAAB  
ABBAAAABBBBABABAABBABBBBAABABBBABAABA

## Example 2

What is the LCS of the following sequences ?





# 1. Introduction

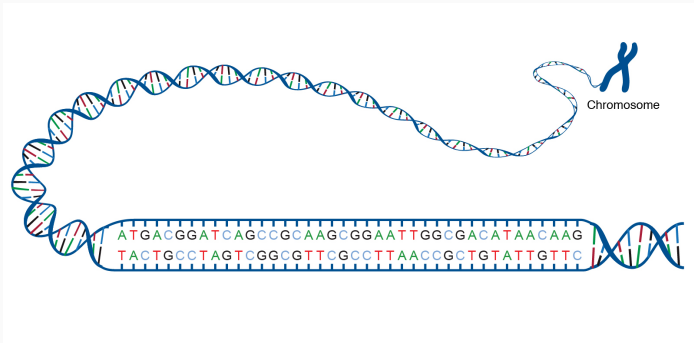
---

## 1.2 Why are we interested in LCS ?

# Applications

# Applications

- Bioinformatics: Compare sequences of nucleotides (DNA)



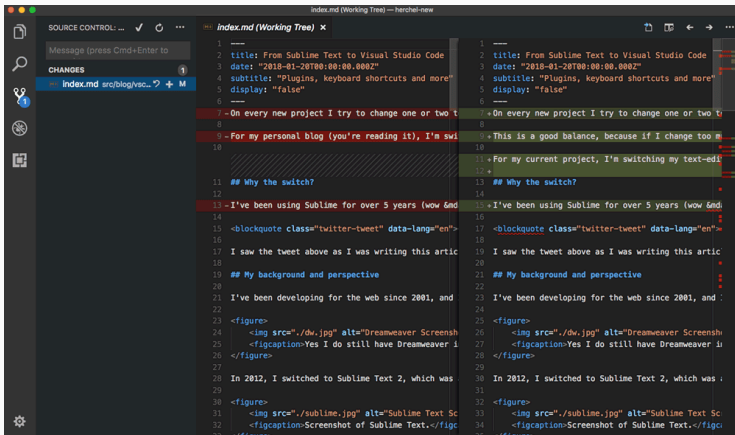
# Applications

- 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.

|             | $\emptyset$ | A | B | A | B | B |
|-------------|-------------|---|---|---|---|---|
| $\emptyset$ | 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.

|             | $\emptyset$ | A | B | A | B | B |
|-------------|-------------|---|---|---|---|---|
| $\emptyset$ | 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.

|             | $\emptyset$ | A | B | A | B | B |
|-------------|-------------|---|---|---|---|---|
| $\emptyset$ | 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.

|             | $\emptyset$ | A | B | A | B | B |
|-------------|-------------|---|---|---|---|---|
| $\emptyset$ | 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.

|             | $\emptyset$ | A | B | A | B | B |
|-------------|-------------|---|---|---|---|---|
| $\emptyset$ | 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 adjacent top-left-diagonal cell by 1.
- Else take the maximum of top and left cells.

|             | $\emptyset$ | A | B | A | B | B |
|-------------|-------------|---|---|---|---|---|
| $\emptyset$ | 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 adjacent top-left-diagonal cell by 1.
- Else take the maximum of top and left cells.

|             | $\emptyset$ | A | B | A | B | B |
|-------------|-------------|---|---|---|---|---|
| $\emptyset$ | 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 adjacent top-left-diagonal cell by 1.
- Else take the maximum of top and left cells.

|             | $\emptyset$ | A | B | A | B | B |
|-------------|-------------|---|---|---|---|---|
| $\emptyset$ | 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 adjacent top-left-diagonal cell by 1.
- Else take the maximum of top and left cells.

|             | $\emptyset$ | A | B | A | B | B |
|-------------|-------------|---|---|---|---|---|
| $\emptyset$ | 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 adjacent top-left-diagonal cell by 1.
- Else take the maximum of top and left cells.

|             | $\emptyset$ | A | B | A | B | B |
|-------------|-------------|---|---|---|---|---|
| $\emptyset$ | 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 adjacent top-left-diagonal cell by 1.
- Else take the maximum of top and left cells.

|             | $\emptyset$ | A | B | A | B | B |
|-------------|-------------|---|---|---|---|---|
| $\emptyset$ | 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 adjacent top-left-diagonal cell by 1.
- Else take the maximum of top and left cells.

|             | $\emptyset$ | A | B | A | B | B |
|-------------|-------------|---|---|---|---|---|
| $\emptyset$ | 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 adjacent top-left-diagonal cell by 1.
- Else take the maximum of top and left cells.

|             | $\emptyset$ | A | B | A | B | B |
|-------------|-------------|---|---|---|---|---|
| $\emptyset$ | 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.

|             | $\emptyset$ | A | B | A | B | B |
|-------------|-------------|---|---|---|---|---|
| $\emptyset$ | 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 |

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

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

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

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

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

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

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

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

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

|             | $\emptyset$ | A | B | A | B | B |
|-------------|-------------|---|---|---|---|---|
| $\emptyset$ | 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

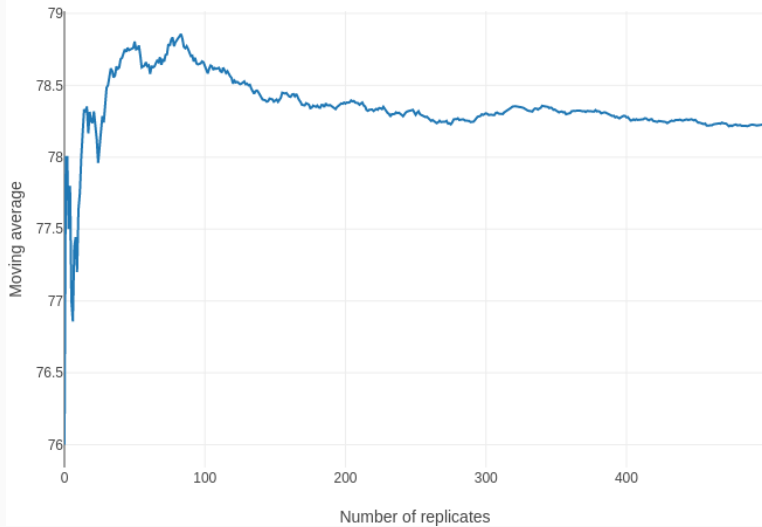
**Given two sequences of the same length,  
what percentage do they have in common ?**

### Question

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

**Answer:  $\approx 80\%$**

LCS moving average over 500 replicates



# 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

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

$S_3$ :   **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 |

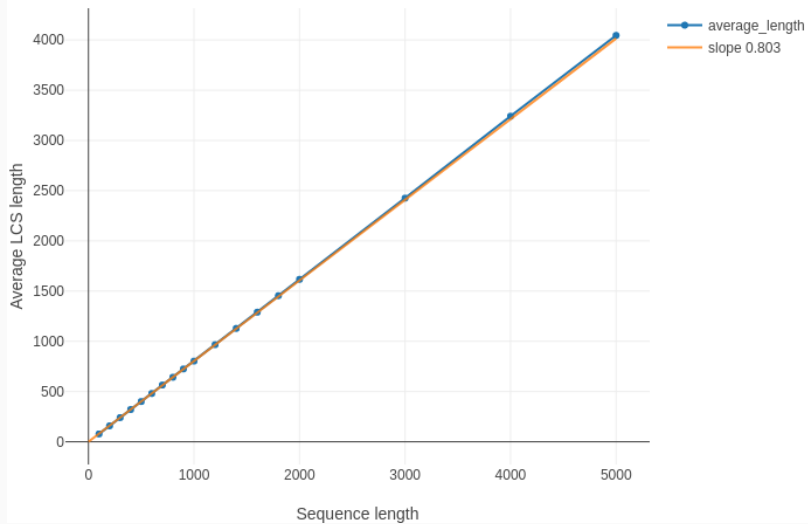
LCS:

**ABABBA**

Length:

**6**

Average LCS length comparison over multiple sequence lengths

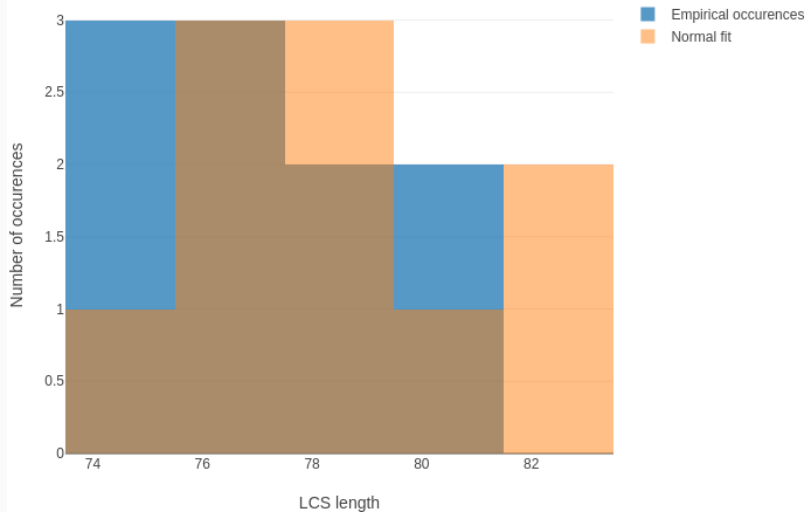


## **3. Data analysis of LCS results**

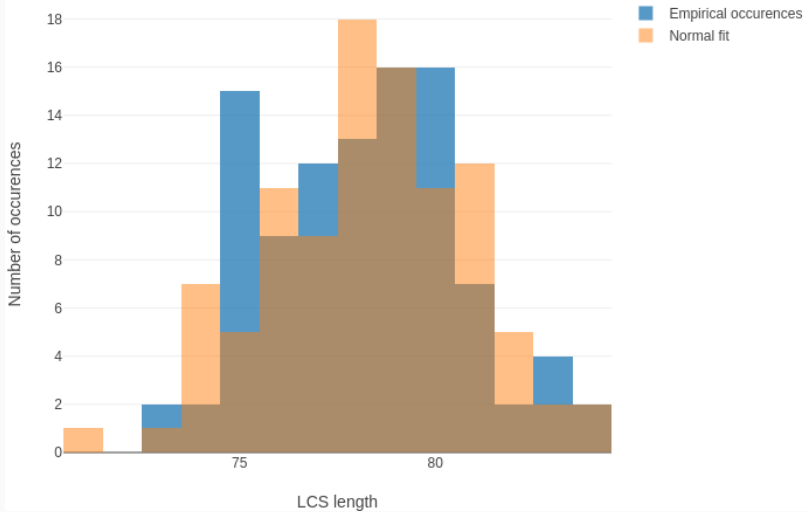
---

### **3.2 Normal fit**

Distribution of LCS length for 10 replicates with length 100

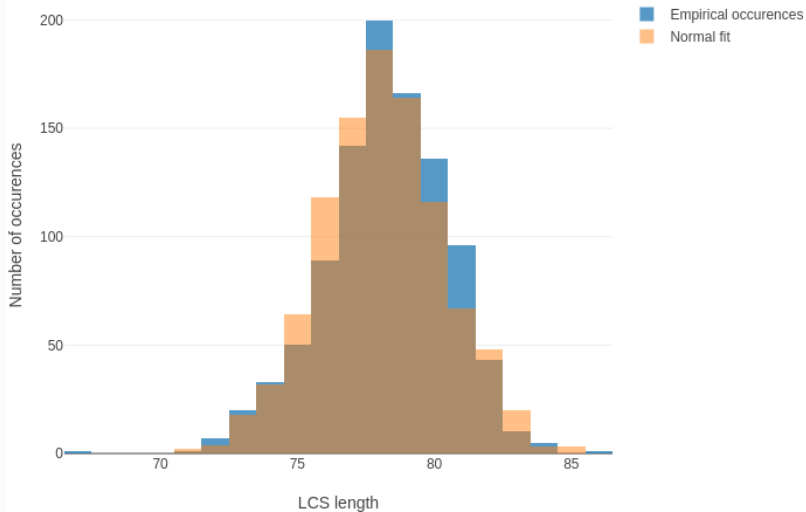


Distribution of LCS length for 100 replicates with length 100

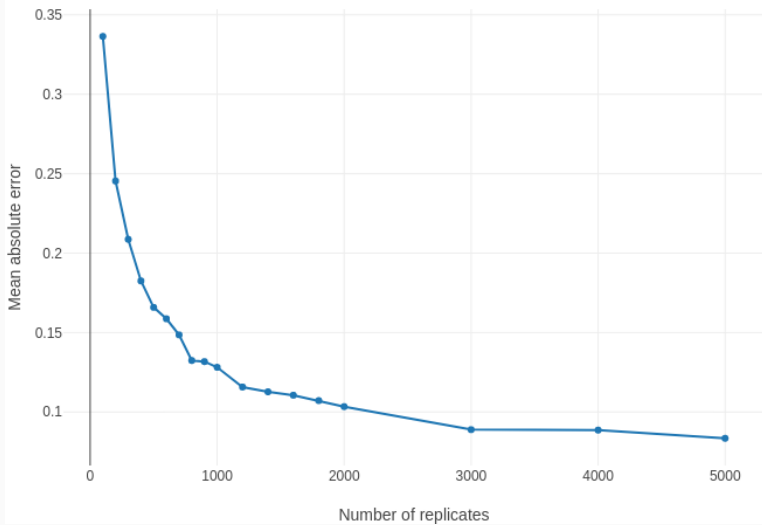




Distribution of LCS length for 1000 replicates with length 100



Mean absolute difference to Normal distribution (length 100)



**Thank you**

**Questions?**

**<https://github.com/jorislimonier/LCS>**