Longest Common Subsequence (and an ILP)

A subsequence

For a string S a **subsequence** is a subset of characters from the string in the same relative order.

Formally: a subsequence is specified by a set of indices $i_1 < i_2 < ... < i_k$, such that $k \leq m.$ The subsequence is then $S[i_1] \cdot S[i_2] \cdot S[i_3] \cdot ... \cdot S[i_k].$

The LCS Problem

Given

- two strings *S* and *T*

Find

- the subsequences $\hat{S} = \hat{T}$
- of maximum length

The LCS Problem

Given

- two strings *S* and *T*

Find

- the subsequences $\hat{S} = \hat{T}$
- of maximum length

S = *ATTCGATACAGTG*

T = *TATCTGAGGTGA*

The LCS Problem

Given

- two strings S and T

Find

- the subsequences $\hat{S} = \hat{T}$
- of maximum length

$S = \mathit{ATTCGATACAGTG}$

$T = \mathit{TATCTGAGGTGA}$

ATTATG

The LCS Problem

Given

- two strings S and T

Find

- the subsequences $\hat{S} = \hat{T}$
- of maximum length

$S = ATTCGATACAGTG$

$T = TATCTGAGGTGA$

$ATTATG$

The LCS Problem

Given

- two strings S and T

Find

- the subsequences $\hat{S} = \hat{T}$
- of maximum length

$S = ATTCGATACAGTG$

$T = TATCTGAGGTGA$

ATTATG

ATTGATG

A Dynamic Program

L[0][j] = L[i][0] = 0

$$L[i][j] = \max \begin{cases} L[i-1][j-1]+1, & \text{if } S[i] == T[j] \\ L[i-1][j] \\ L[i][j-1] \end{cases}$$

A Dynamic Program

L[0][j] = L[i][0] = 0

$$L[i][j] = \max \begin{cases} L[i-1][j-1]+1, & \text{if } S[i] == T[j] \\ L[i-1][j] \\ L[i][j-1] \end{cases}$$

What about as an ILP?

An ILP

maximize $\sum_{1 \le i \le n} \sum_{1 \le j \le m} P(i, j)$

subject to

$$\sum_{1 \le j \le m} P(i, j) \le 1 \qquad \forall i$$

$$\sum_{1 \le i \le n} P(i, j) \le 1 \qquad \forall j$$

$$P(i, j) + P(i', j') \le 1 \qquad i < i' \,\&\, j' < j$$

$$P(i, j) \in 1{,}0 \qquad \forall i, j$$

$$P(i, j) \le 0 \qquad \forall i, j : S[i] \ne T[j]$$