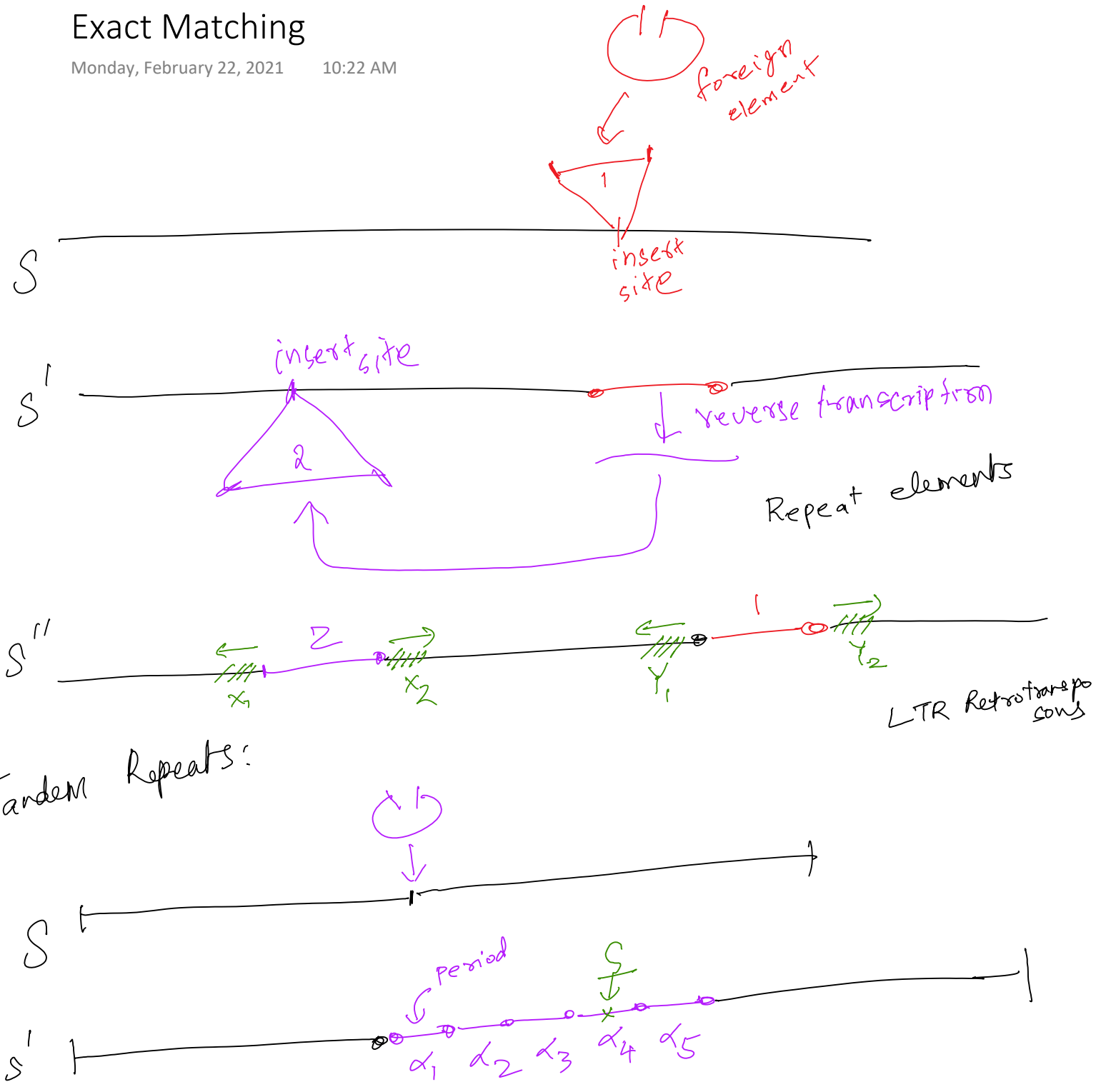
Exact Matching

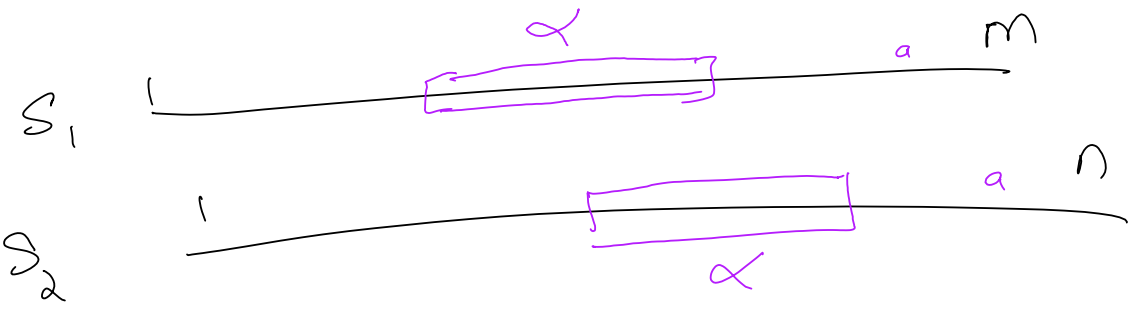
Monday, February 22, 2021 10:22 AM



Exact matches between >=2 sequences

Monday, February 22, 2021 10:23 AM

Usecase #2: Multiple (≥ 2) sequences (input)

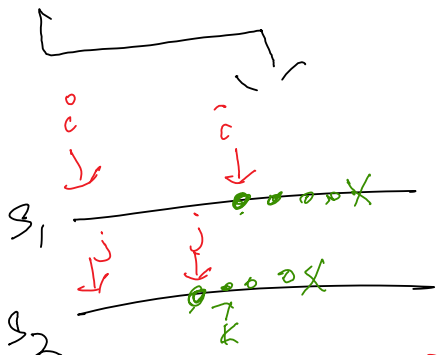


a) Find the longest common substring b/t S_1 & S_2

DP approach:

local

θ: parameters $\{ m_a = +1,\ m_i = -\infty,\ h = -\infty,\ g = -\infty \}$




```
for (i ...   m
  for (j ...   n
    for (k ...)
      ++ matchy
```

use k-mers as building blocks to identify α

$O(mn \times n)$

n^3

lcs (S_1, S_2) // exact matching reduced to

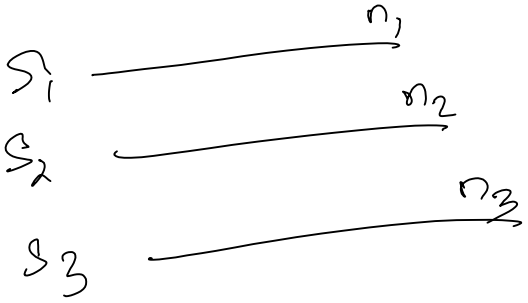
↓

local Align (S_1, S_2) w/: $\theta \{+1, -\sigma, -\beta\}$ // inexact matching.

$O(mn)$ time

n^2

b) Can lcs be solved in $O(m+n)$ time?



$O(n_1 n_2 n_3)$

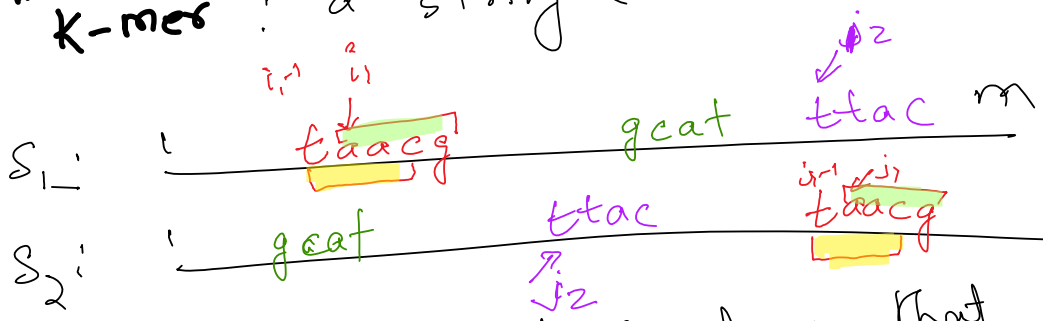
Fixed length matches (k-mers)

Monday, February 22, 2021 10:23 AM

Finding exact matches of a Fixed length.

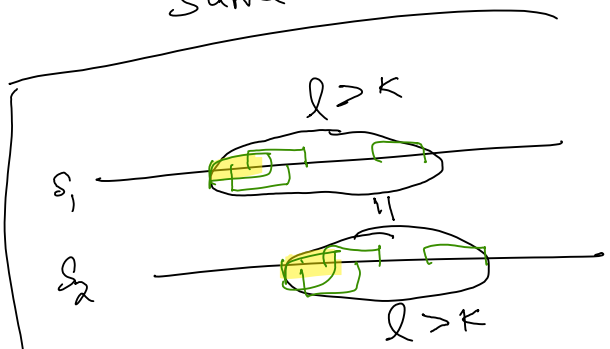
"k-mer": a string (or a substring) of length $k > 0$ ($k=4$)

n-grams
q-grams



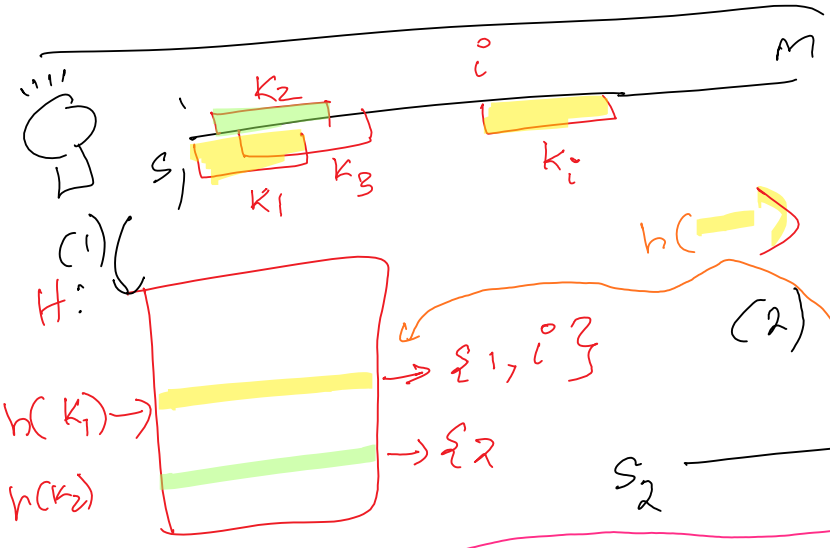
Goal: to identify all locations that start with the same k-mer

$\langle s_1 : i_1, s_2 : j_1 \rangle$
$\langle s_1 : i_2, s_2, j_2 \rangle$
$\langle s_1 : i_{l-1}, s_2, j_1 - 1 \rangle$



If match length is l, how many k-mers are shared in that window?

$l - k + 1$



1.) Build a hash table/index for all k-mers in S_1
2.) For each k-mer in S_2, search in H and retrieve the list of S_1 indices.

Look-up Tables

Wednesday, February 24, 2021 10:23 AM

Lookup tables A collision-free hash table for all k-mers

$K = 2$

$\Sigma = \{a, c, g, t\}$

S: a c c g t a g

"ac" "gt"

$h("ac") = 1$? $h("gt") = 11$

$2 \times |\Sigma|^1 + 3 \times |\Sigma|^0$

$= 11$

H:

0	aa
1	ac
2	ag
3	at
4	ca
5	cc
6	cg
7	ct
8	ga
9	gc
10	gg
11	gt
	ta
	tc
	tg
$\lvert\Sigma\rvert^K - 1$	tt

lookup table size $= |\Sigma|^K$

kmer: msb $K-1$ ... 2 1 0 lsb

msb lsb

"ac" 1 1

v(a) = 0 v(c) = 1

$0 \times |\Sigma|^1 + 1 \times |\Sigma|^0$

$= 0 + 1$

$= 1$

$\Sigma = \{0, 1\}$

K=2: 00 = 0, 01 = 1, 10 = 2, 11 = 3

symbol c	value v(c)
a	0
c	1
g	2
t	3

$\times |\Sigma|^1 + \times |\Sigma|^0$

K

Look-up table construction

Wednesday, February 24, 2021 11:44 AM

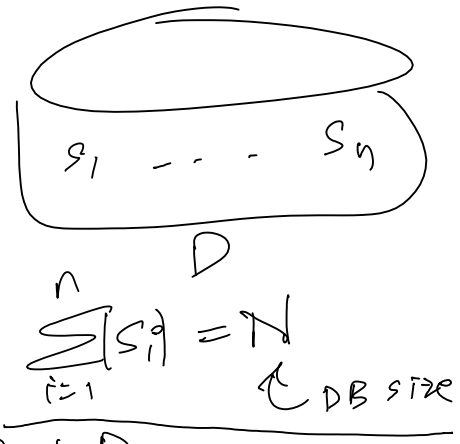
Q) How to build lookup table for the database string D

$D = \{s_1, s_2, \ldots s_n\}$

Input: $D[0 \ldots N-1]$, K, Σ

Output: $LT(D)$

Algo:



1) Init. LT with $|\Sigma|^K$ locations $\rightarrow O(|\Sigma|^K)$

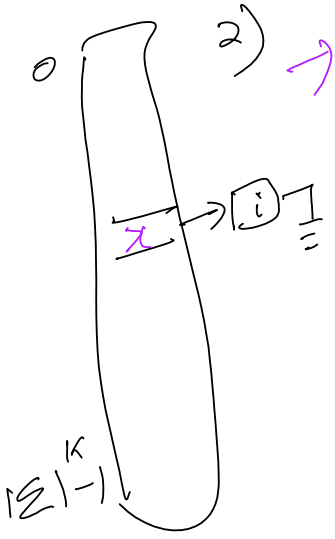
2) for ($i = 0$ to $N-K+1$) $\rightarrow O(N)$

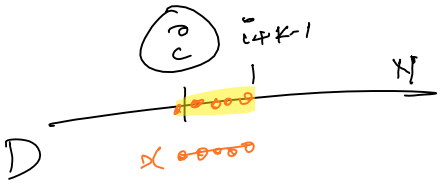
$x \leftarrow$ extract K-mer starting at i in D

insert (x, LT) $\rightarrow O(K)$

$h(x) \rightarrow$ index

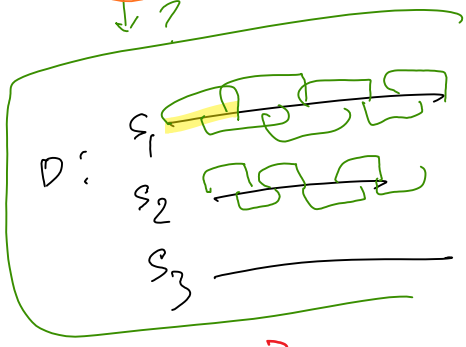
LT[index].append(i)

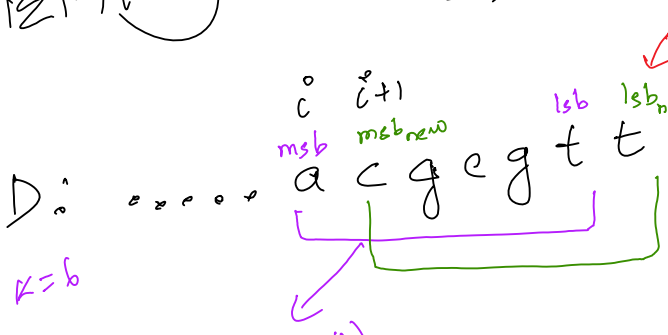




$\Rightarrow$ Runtime $= O(|\Sigma|^K + N(K))$

K = 11
= 12





$x \leftarrow h("acgcgt")$

$h_{new}^{i+1} \leftarrow \left(h_{prev}^{i} - value(D[i]) \times |\Sigma|^{K-1} \right) \times |\Sigma| + value(D[i+K])$

$\Rightarrow$ can calculate new hash value (at i) in $O(1)$ time using the previous value at $i-1$)

$\Rightarrow$ construction time $= O(|\Sigma|^K + N)$ time.

Look-up table: Analysis

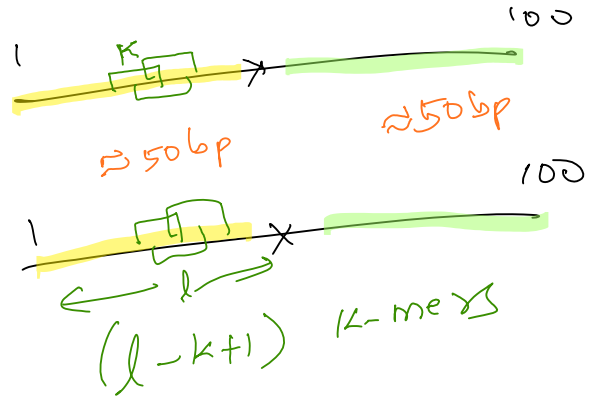
Friday, February 26, 2021 10:49 AM

DNA: $\Sigma = \{a, c, g, t\}$

k:	5	10	11	15	20 ...
size: (#entries)	$4^5 = 2^{10} \approx 1K$	$4^{10} \approx 2^{20} \approx 1M$	$4^{11} = 2^{22} \approx 4M$	$4^{15} = 2^{30} \approx 1B$	high expense $\|\Sigma\|^k$

($k = 11$ used by BLAST)

reads S_1, S_2



Q) variable length in a direct ways.