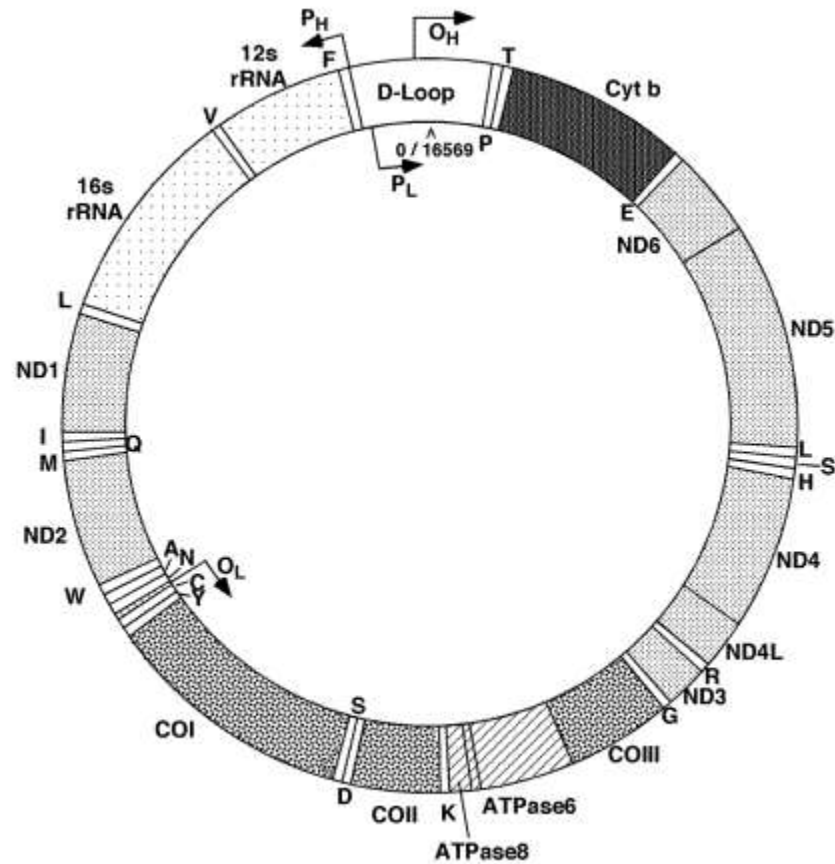


CIRCULAR SEQUENCE COMPARISON: ALGORITHMS AND APPLICATIONS



Circular Sequence Comparison: Algorithms and Applications

Authors:

Nadia Pisanti, Roberto Grossi (University of Pisa)
and 5 others (King's College London).

Soon to be published on BMC Bioinformatics.

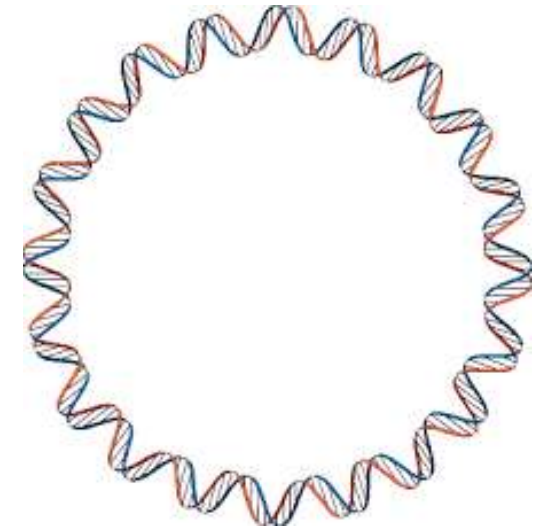
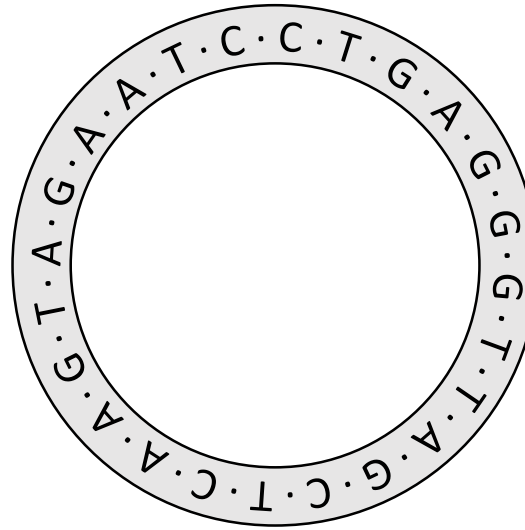
These slides are available at:

<https://github.com/robzan8/csc>

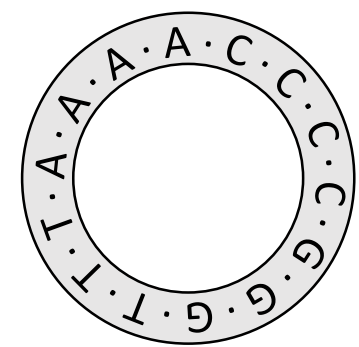
Circular Sequences

Examples:

Bacterial chromosomes and plasmids;
Mitochondrial DNA;
Viral genomes;
Circular proteins;
And more...



Comparisons



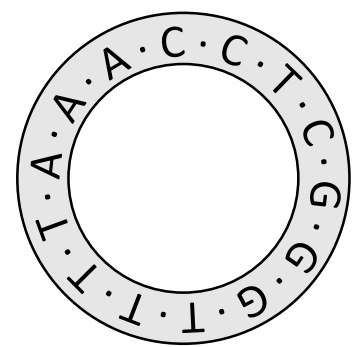
A A A A C C C C G G G T T T

A A A A C C C C G G G T T T

A A A A C C C C G G G T T T _

A A A _ C C C C G G G T T T T

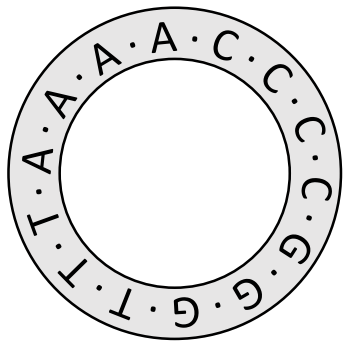
? A A A A C C C C G G G T T T ?
G G T T T T A A A C C T C G



A A A C C T C G G G T T T T

G G T T T T A A A C C T C G

Rotation



x A A A A C | C C C G G G T T T

x^5 C C C G G G T T T | A A A A C

$(xx)[5 \dots n + 5]$ A A A A C | C C C G G G T T T | A A A A C | C C C G G G T T T

q-gram distance

Approximation of edit distance, can be computed in linear time and space.

$$D_q(x, y) = \sum_{v \in \Sigma^q} |G_q(x)[v] - G_q(y)[v]|$$

β -blockwise q-gram distance, more accurate, linear time and $\mathcal{O}(\frac{m+n}{\beta})$ space.

$D_{\beta,q}(x, y)$

x

A A A A | C C C C | G G G T | T T A A

y

A A A C | C T C G | G G T T | T T A A

Σ for each block pair

Circular Sequence Comparison problem (CSC)

Input: strings x, y of lengths m and $n \geq m$, integers $\beta \geq 1$ and $q < m$

Output: i such that $D_{\beta,q}(x^i, y)$ is minimal

Naïve algorithm (nCSC) complexity: $\mathcal{O}(m(m + n))$

Heuristic algorithm (hCSC)

Step 1: divide xx in 2β blocks and y in β blocks.

Step 2: calculate $\delta_j = D_{\beta,q}(x^{j\frac{m}{\beta}}, y)$ shifting the window block by block.

Step 3: starting form position with best δ_j , refine search by moving left and right by m/β characters.

xx

A A A A | C C C C | G G G T | T T A A | A A A A | C C C C | G G G T | T T A A

y

A A A C | C T C G | G G T T | T T A A

$j = 3$

Analysis of hCSC

Step 2 (block-by-block search) $\mathcal{O}(\beta(m + n))$

Step 3 (char-by-char local search) $\mathcal{O}\left(\frac{m}{\beta}(m + n)\right)$

Total $\mathcal{O}\left(\left(\beta + \frac{m}{\beta}\right)(m + n)\right)$

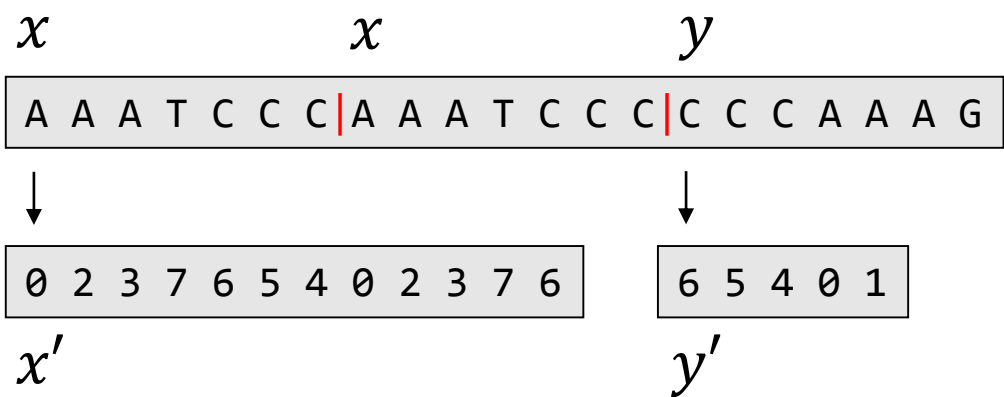
By letting $\beta = \sqrt{m}$ $\mathcal{O}(\sqrt{m}(m + n))$

Linear additional space

Does not necessarily find global optimum!

Exact algorithm based on suffix array (saCSC)

Step 1: Construct the suffix array of xy
and substitute each q-gram with its rank



SA	
A A A	0
A A G	1
A A T	2
A T C	3
C A A	4
C C A	5
C C C	6
T C C	7

Exact algorithm based of suffix array (saCSC)

Step 2: Compute the blockwise q-gram distance for the initial window position

$$\begin{array}{l} x' \quad [9 \ 6 \ 2 \ 2 \mid 8 \ 0 \ 1 \ 5 \mid 7 \ 9 \ 0 \ 1 \mid 3 \ 7 \ 4 \ 5 \mid 9 \ 6 \ 2 \ 2 \ 8 \ 0 \ 1 \ 5 \ 7 \ 9 \ 0 \ 1 \ 3 \ 7 \ 4 \ 5] \\ y' \quad [0 \ 1 \ 3 \ 7 \mid 4 \ 5 \ 9 \ 6 \mid 2 \ 2 \ 8 \ 0 \mid 1 \ 5 \ 7 \ 9] \end{array}$$

Step 3: Slide the window char-by-char, update the distance and keep the best position

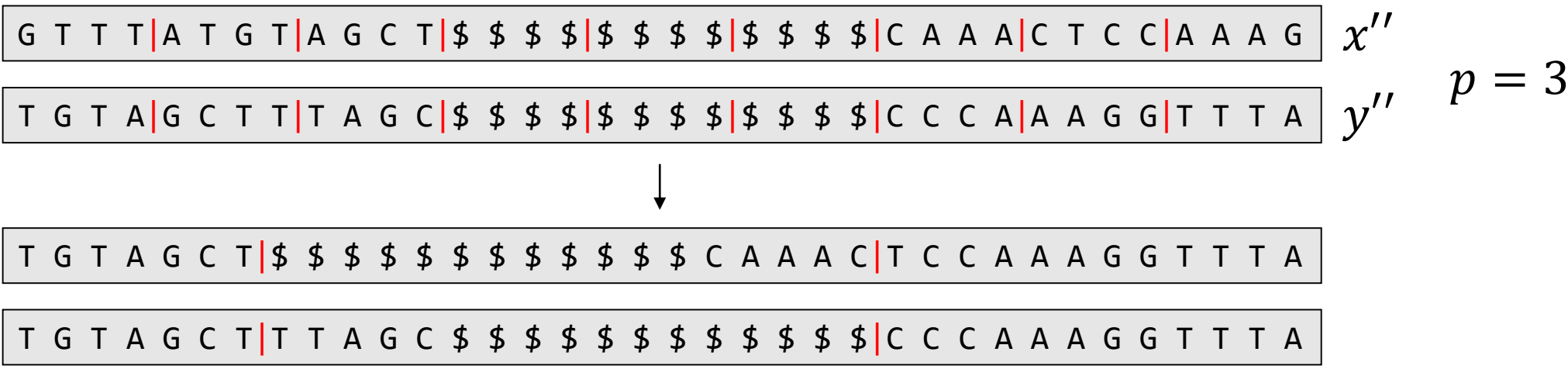
$$\begin{array}{l} x' \quad \boxed{9 \ 6 \ 2 \ 2 \ 8 \ 0 \ 1 \ 5 \ 7 \ 9 \mid 0 \ 1 \ 3 \ 7 \mid 4 \ 5 \ 9 \ 6 \mid 2 \ 2 \ 8 \ 0 \mid 1 \ 5 \ 7 \ 9 \mid 0 \ 1 \ 3 \ 7 \ 4 \ 5} \\ y' \quad \boxed{} \end{array}$$

Analysis of saCSC

Suffix array calculation	$\mathcal{O}(m + n)$
Distance with sliding window	$\mathcal{O}(\beta m + n)$
Total (for both time and space)	$\mathcal{O}(\beta m + n)$

saCSC refinement step (saCSCr)

Alignment of the first and last p blocks of x^i and y with Needleman-Wunsch, considering all possible rotations



Time complexity: $O\left(\left(p \frac{m}{\beta}\right)^3\right)$

Experimental results

Algorithm tested on various real and synthetic data

Applications on real data:

Chimpanzee - human MtDNA (from GenBank) comparison:

85% similarity and ~1200 gaps with EMBOSS Needle vs

91% similarity and 77 gaps with correct rotation;

Distance-based phylogenetic reconstruction:

MtDNA, viroid RNA, circular proteins.

saCSCr gives the same results as cNW (“brute force” Needleman-Wunsch), but with much smaller execution times!

Time performance

Experimental performance is in line with theoretical expectations

saCSC	(q-gram dist, suffix array)	$\mathcal{O}(\beta m + n)$
hCSC	(q-gram dist, heuristic)	$\mathcal{O}(\sqrt{m}(m + n))$
hSW	(Smith-Waterman, heuristic)	$\mathcal{O}(mn)$
nCSC	(q-gram dist, naïve)	$\mathcal{O}(m(m + n))$
cNW	(Needleman-Wunsch, naïve)	$\mathcal{O}(m^2n)$

Conclusions

- β -blockwise q-gram distance can be computed efficiently and used effectively
- saCSC solves the CSC problem exactly and fast
- Refinement step bridges the gap between q-gram approximation and optimal solution
- saCSCr to be implemented in BEAR (state-of-the-art tool for multiple circular sequence alignment)

Q&A!