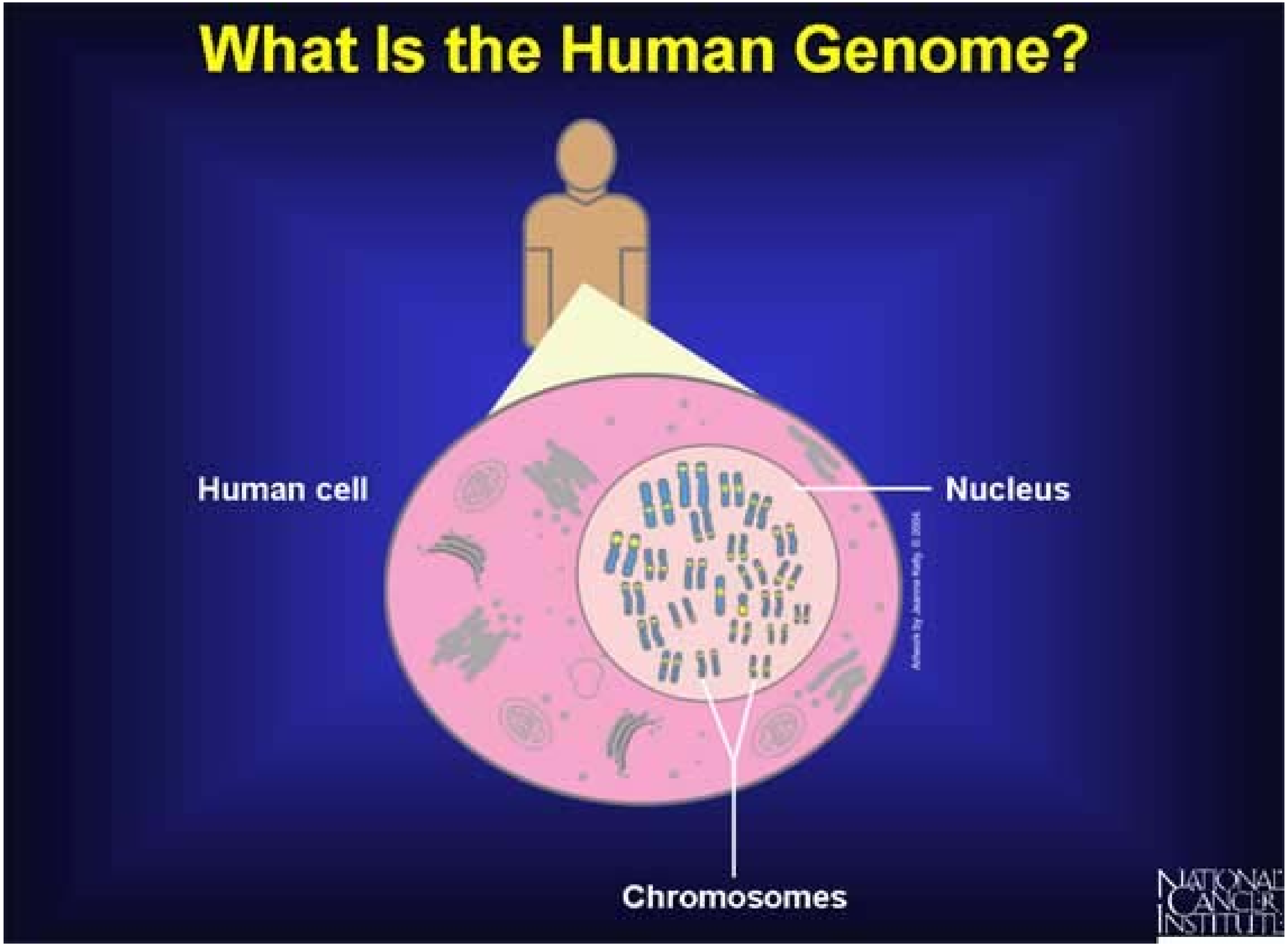
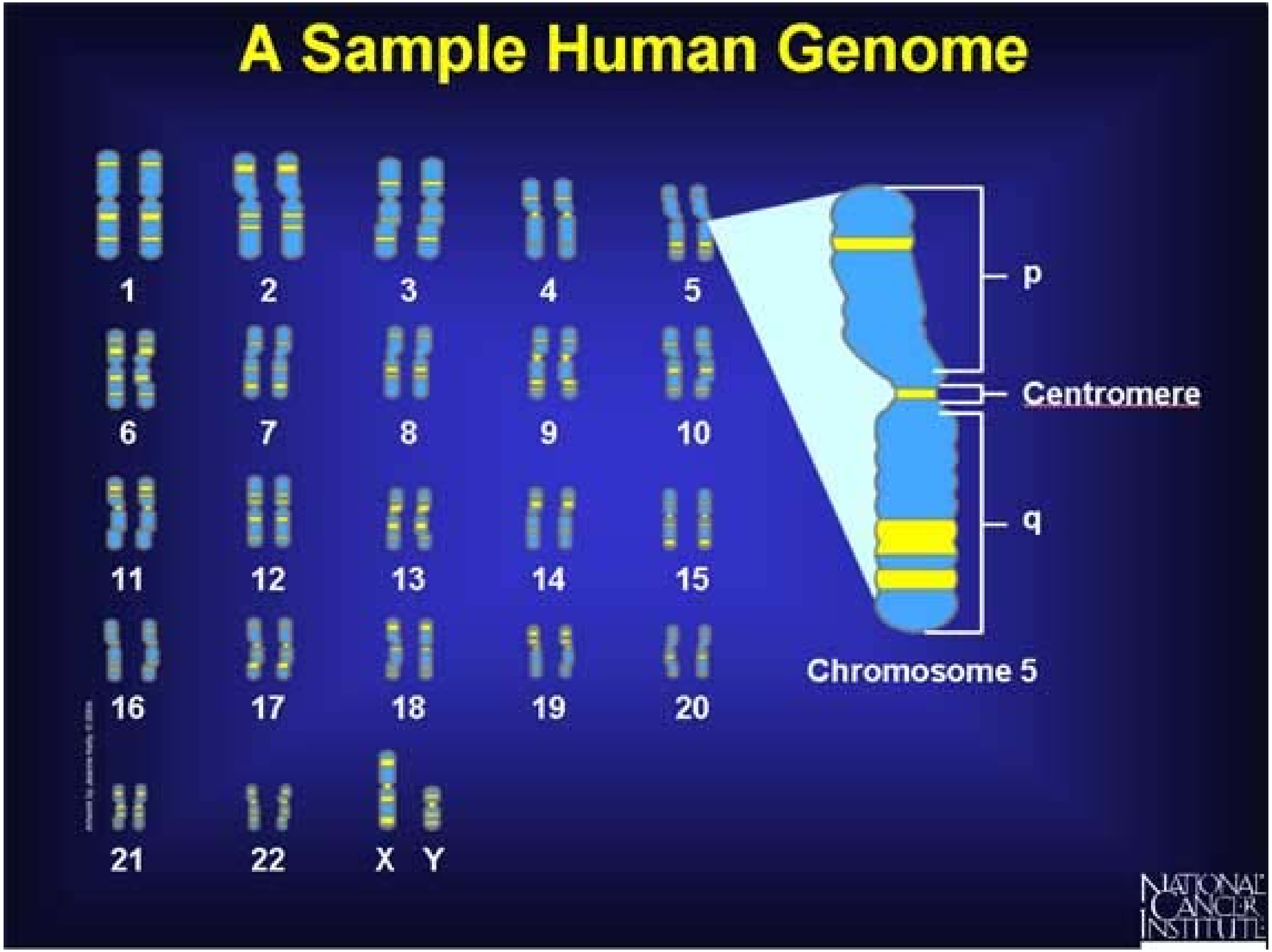
COMPUTING TRANSLOCATION DISTANCES

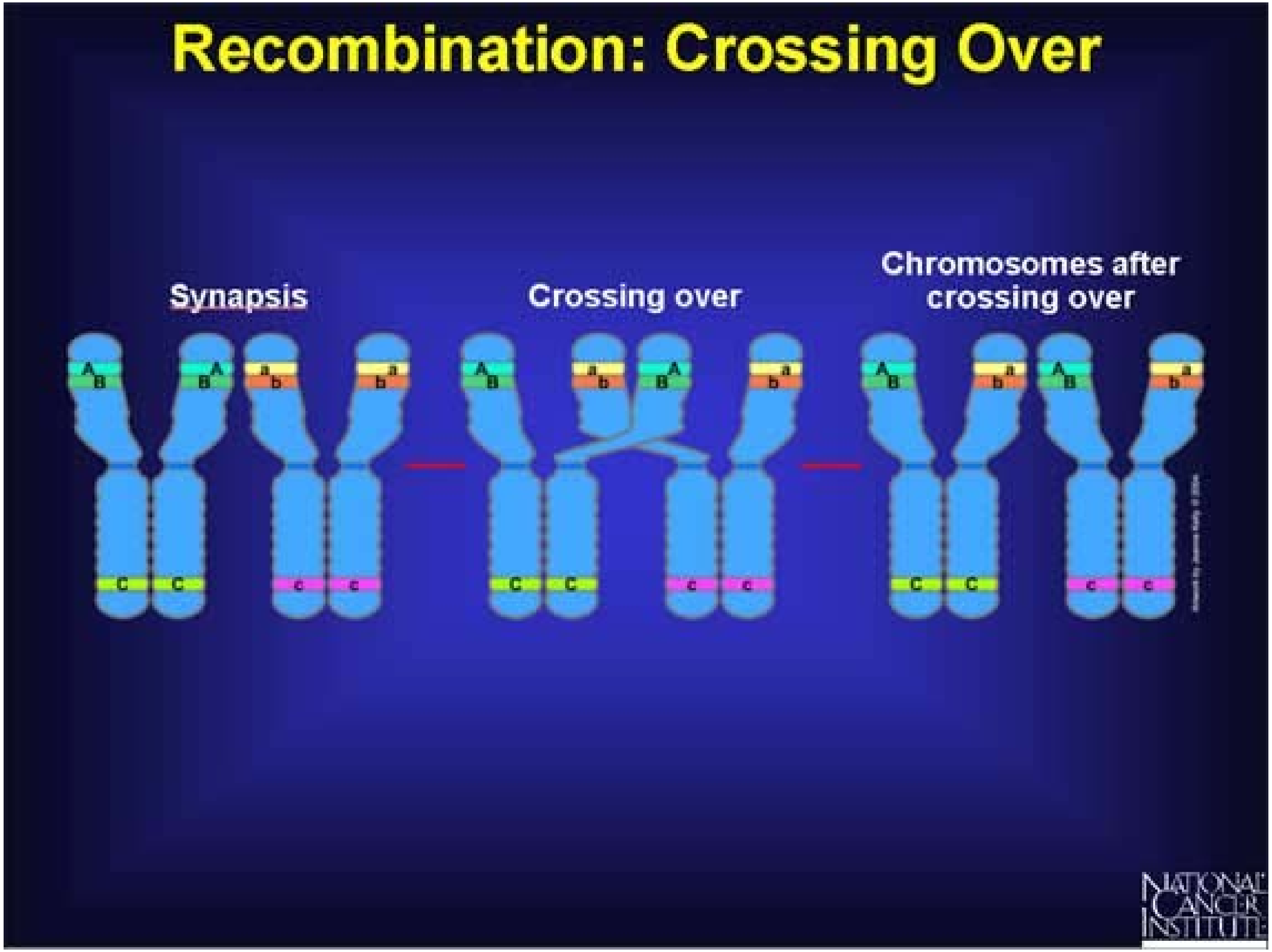
VICTOR MITRANA

victor.mitrana@upm.es

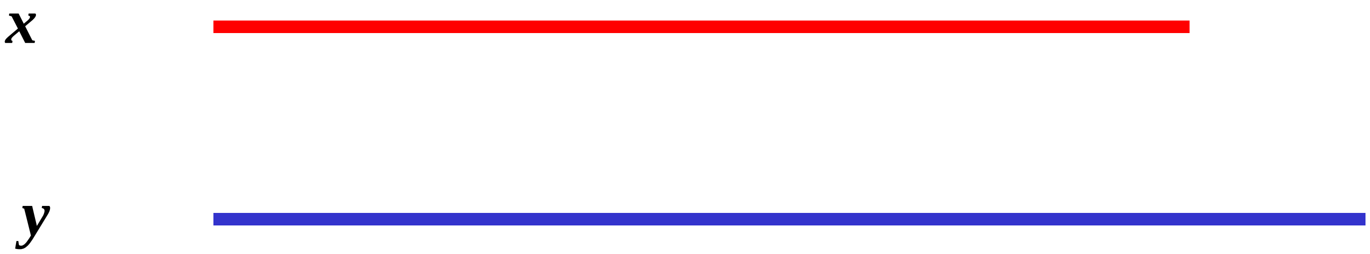
CONTENTS


What Is the Human Genome?
Human cell
Nucleus
Chromosomes
NATIONAL CANCER INSTITUTE

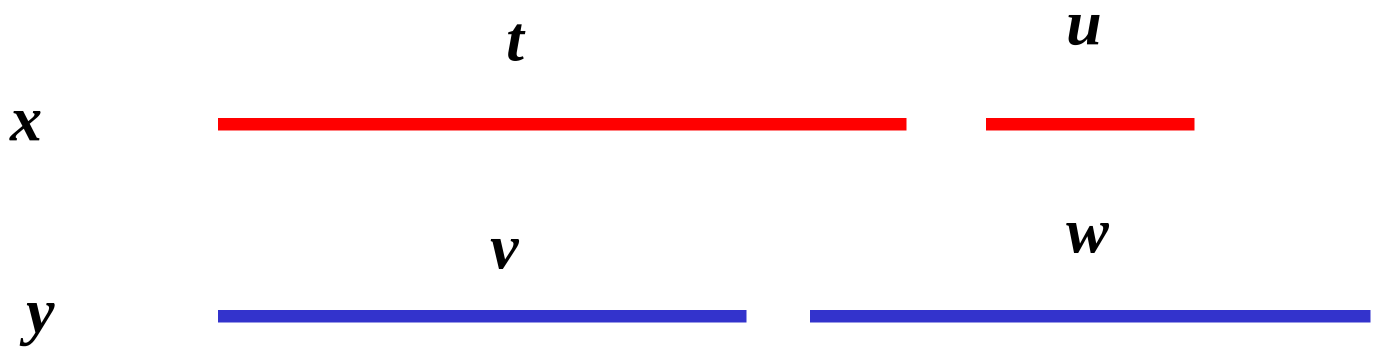

A Sample Human Genome
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
X
Y
p
Centromere
q
Chromosome 5
NATIONAL CANCER INSTITUTE


Recombination: Crossing Over
Synapsis
Crossing over
Chromosomes after crossing over
A
B
a
b
C
c
NATIONAL CANCER INSTITUTE

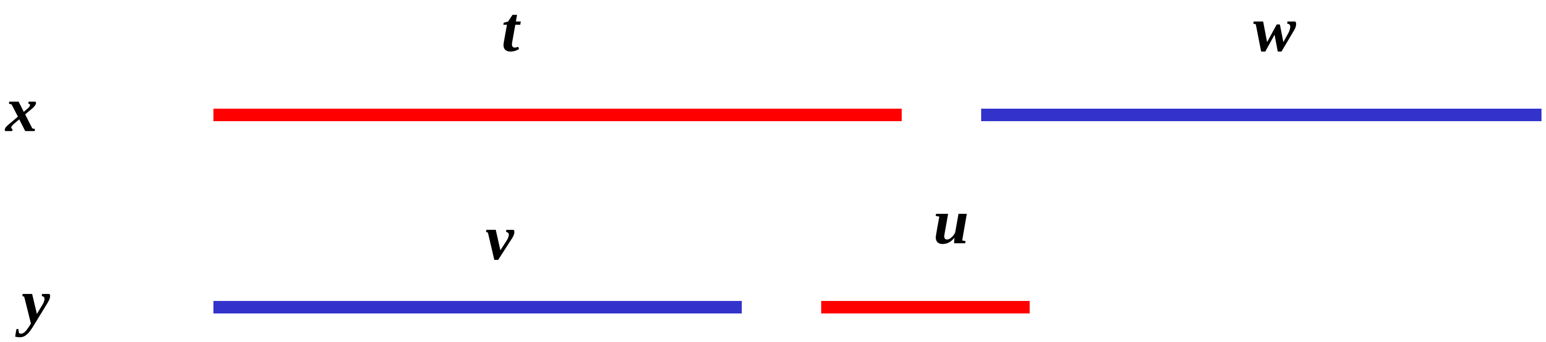
Translocation/Crossover-Formal



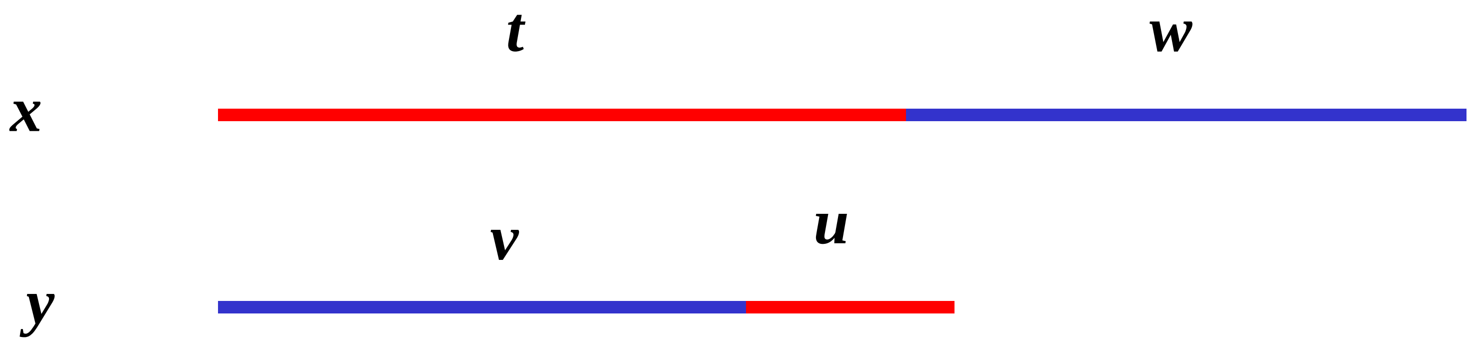
Translocation/Crossover-Formal



Translocation/Crossover-Formal



Translocation/Crossover-Formal



$(x, y) \vdash_{(i,j)} (z_1, z_2)$ iff $x = tu, y = vw, z_1 = tw, z_2 = vu,$ and $|t| = i, |v| = j.$

$\vdash_{(i,j)}$ is said to be *uniform* iff $i=j$, *so that we shall simply write* $\vdash_i$

$$[U]CO(A) = \bigcup_{\{x,y\in A\}} \{z \mid (x,y) \vdash_{(i,j)} (z,w) \text{ or } (x,y) \vdash_{(i,j)} (w,z)\}$$

The Problem: Translocation distance

Given two genomes *G* and *G'* what is the minimal number of translocation mutations that transforms *G* into *G'*?

1. How the translocation is defined: uniform or arbitrar.

2. How the chromosomes in the two genomes are:
they are formed by different segments (markers) or not.

3. How large is the target genome: singleton or arbitrary

Uniform translocation distance

Uniform translocation and unique markers
(J. Kececioglu, R. Ravi)

Assumptions:
1. All chromosomes (words) in both genomes are of the same length k.
2. Each marker (symbol) appears at most once in a chromosome and in only one.
3. If G has n chromosomes, then G' must have n chromosomes as well.

Important note: If a symbol appears on the position i in a word in G, then it will appears on the same position in a word of G'.

Theorem 1. The uniform translocation distance between G and G' can be computed in time and memory $O(kn)$.

Ingredients: Greedy strategy
Cayley (1849): The minimal number of transpositions for sorting π is n-$\Psi(\pi)$.

Uniform translocation distance

1. We label the words in G' in some way from *1* to *n*.
2. Associate with each set G, G' a matrix as follows:
   - each column in the matrix represents a word
   - each symbol from a word is represented by the unique word of G' in which it occurs.

Example: $G = \{a_2a_7a_9a_4, a_5a_1a_{12}a_8, a_{10}a_3a_6a_{11}\}$
$G' = \{a_{10}a_1a_9a_8, a_5a_7a_6a_4, a_2a_3a_{12}a_{11}\}$

$$M_G = \begin{pmatrix} 3 & 2 & 1 \\ 2 & 1 & 3 \\ 1 & 3 & 2 \\ 2 & 1 & 3 \end{pmatrix} \quad M_{G'} = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$$

Problem: Select two columns and a natural $l \leq n-1$ and interchange the elements of the first l rows.

Uniform translocation distance

Let (i, j, l): the columns i and j interchange each other the entries of the first l rows. A solution is a sequence

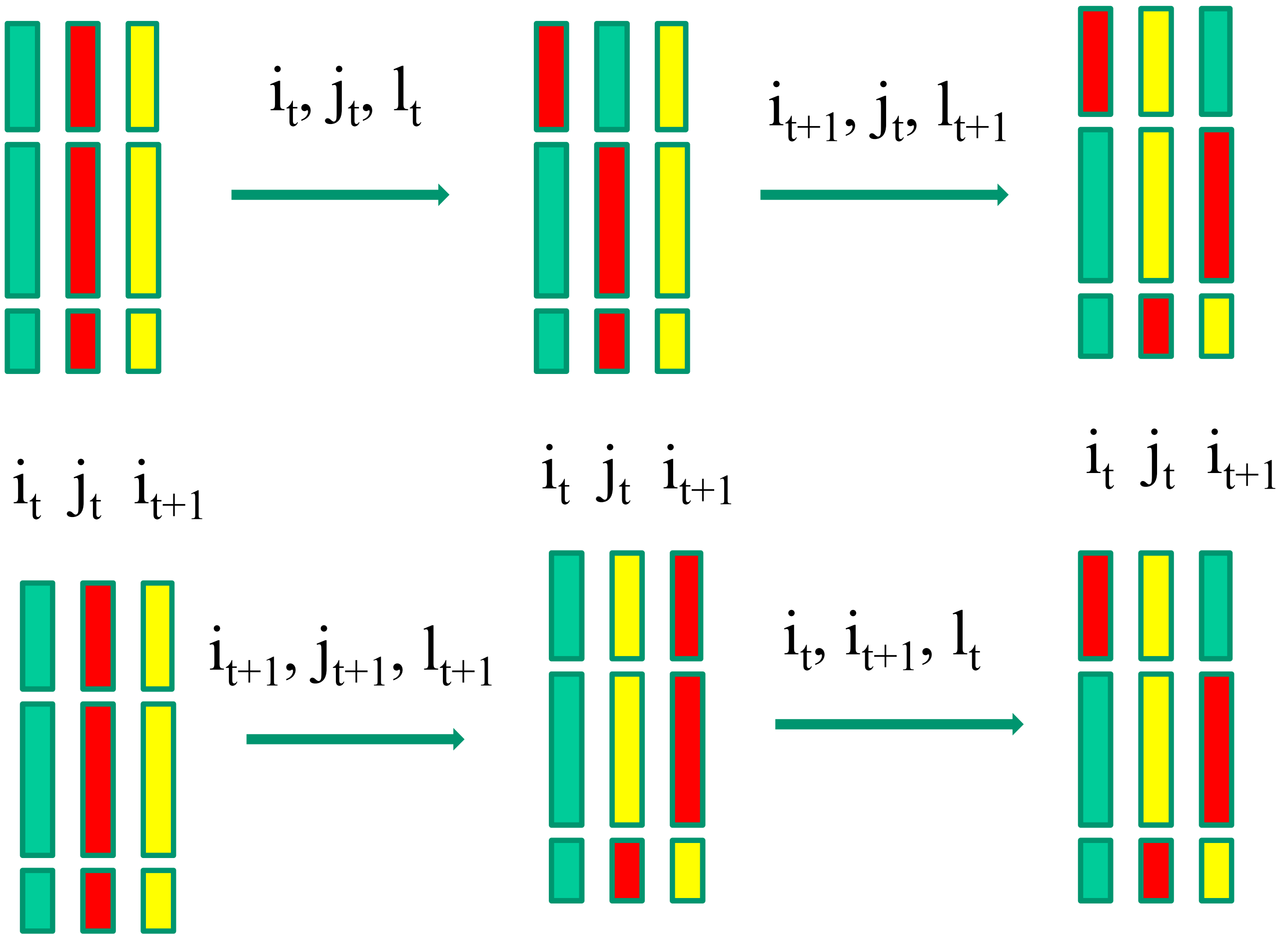
$$(i_1, j_1, l_1), (i_2, j_2, l_2), \ldots (i_p, j_p, l_p)$$

Find the minimal p.

A solution $(i_1, j_1, l_1), (i_2, j_2, l_2), \ldots (i_p, j_p, l_p)$ is “bottom-up if there are no $1 \leq s < q \leq n-1$ such that $l_q > l_s$.

Uniform translocation distance

Lemma: Any instance of the problem has a solution which is bottom-up.



Uniform translocation distance

A bottom-up sequence is *locally optimal* if the number of transformations applied to the current row in order to transform it into the identical permutation is minimal.

Lemma 2 *A bottom-up locally optimal is totally optimal.*

Proof. Let us consider a part of a bottom-up sequence when one starts to "sort the row $i+1$. Let π be the current state of the row $i+1$ and λ_i the state of the row. After sorting the row $i+1$ the state of the row i is

$$\lambda_i \circ \pi^{-1}.$$

Uniform translocation distance

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 5 & 4 & 3 & 1 \end{pmatrix};$$

$$PQ = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 4 & 1 & 3 & 5 \end{pmatrix}\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 3 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 3 & 1 & 4 & 2 \end{pmatrix} \neq QP.$$

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 3 & 2 \end{pmatrix}$$

Uniform translocation distance

Given a permutation π, what is the minimal number m of transpositions $\tau_1, \tau_2, \ldots, \tau_m$ such that

$$\pi \circ \tau_1 \circ \tau_2 \circ \ldots \circ \tau_m = \varepsilon_n$$

Lemma 3 (Cayley) *The minimal number of transpositions for sorting π is $n - \Psi(\pi)$.*

procedure Sort_Crossover_uniform(A,k,n);

Let $\lambda_1, \lambda_2, \ldots, \lambda_k$ *the rows of A*
$d := 0$; $\pi := \varepsilon_n$;
for $i := k$ **downto** 1 **do**
    $\pi := \lambda_i \circ \pi^{-1}$;
    $d := d + n - \Psi(\pi)$;
endfor;
end.

Translocation distance: Our solution

Assumptions:

1. All chromosomes (words) in both genomes are of the same length k.
2. Each marker (symbol) appears may appear more than once in any chromosome and in different chromosomes.
3. If G has n chromosomes, then G' may have as many chromosomes as we want.

A few more definitions:

A translocation sequence: $\boldsymbol{S=s_1,s_2, \ldots ,s_n,\ \ s_i=(x_i,y_i)\vdash_{(k(i),p(i))} (u_i,v_i)}$

$$\boldsymbol{P_i(S,x) = \mathrm{card}\{j\leq i | x = x_j \text{ or } x = y_j\}+\mathrm{card}\{j\leq i | x_j = y_j = x\},}$$

$$\boldsymbol{F_i(S, x) = \begin{cases}\mathrm{card}\{j\leq i | u_j = x_j \text{ or } v_j = y_j\}+\mathrm{card}\{j\leq i | u_j = v_j = x\}, & \text{if } x \notin A,\\ \infty, & \text{otherwise}\end{cases}}$$

A translocation sequence S is contiguous iff:

(i) $\boldsymbol{x_1, y_1 \in A}$,

(ii) $\boldsymbol{F_{i-1}(S, x_i) > P_{i-1}(S, x_i)}$, **and** $\boldsymbol{F_{i-1}(S, y_i) > P_{i-1}(S, y_i)}$,

Translocation distance: Our solution

A CTS S is B-producing if $\boldsymbol{F_n(S, z) > P_n(S, z)}$ **for all** $\boldsymbol{z \in B}$.

$$TD(A,B) = \min\{\lg(S) \mid S \text{ is a } B-\text{producing CTS}\}.$$

Compute $\boldsymbol{TD(A,B)}$ → ***B*** **is a singleton**

→ ***B*** **is an arbitrary set**

Translocation distance: Our solution

Example: $A = \{x_1, x_2, x_3, x_4\}$ with
$x_1 = abcbad$, $x_2 = bbabd$, $x_3 = accbabd$, $x_4 = aaab$,
and
z_1 = bbcbad, z_2 = ababd, z_3 = ababad, z_4 = bbcbd, z_5 = abbababd
z_6 = aabad, z_7 = abababd, z_8 = bbd, z_9 = bbbd, z_{10} = bbabad,
z_{11} = bbbabad, z_{12} = bbababd, z_{13} = bababd, z_{14} = accbd, z_{15} =bbccbabd
z_{16} = aababd, z_{17} = abcccbabd z_{18} = abad

A B-producing CTS, $B = \{z_4, z_6, z_8, z_{11}, z_{15}, z_{16}, z_{18}\}$.

$(x_1, x_2) \ast_{(2,2)} (z_2, z_1)$, $(z_1, z_2) \ast_{(4,4)} (z_4, z_3)$,
$(z_2, x_2) \ast_{(4,2)} (z_7, z_8)$, $(z_3, z_7) \ast_{(2,1)} (z_5, z_6)$, $(x_2, x_3) \ast_{(3,3)} (z_{12}, z_{14})$,
$(z_8, z_{12}) \ast_{(2,5)} (z_9, z_{10})$, $(x_2, x_3) \ast_{(3,3)} (z_{12}, z_{14})$, $(x_2, x_3) \ast_{(3,3)} (z_{12}, z_{14})$,
$(z_{12}, z_{10}) \ast_{(2,1)} (z_{11}, z_{13})$, $(z_{12}, x_3) \ast_{(2,1)} (z_{15}, z_{16})$, $(x_1, x_3) \ast_{(3,1)} (z_{17}, z_{18})$.

Translocation distance: Our solution

Example: $A = \{x_1, x_2, x_3, x_4\}$ with
$x_1 = abcbad$, $x_2 = bbabd$, $x_3 = accbabd$, $x_4 = aaab$,
and
z_1 = bbcbad, z_2 = ababd, z_3 = ababad, z_4 = bbcbd, z_5 = abbababd
z_6 = aabad, z_7 = abababd, z_8 = bbd, z_9 = bbbd, z_{10} = bbabad,
z_{11} = bbbabad, z_{12} = bbababd, z_{13} = bababd, z_{14} = accbd, z_{15} =bbccbabd
z_{16} = aababd, z_{17} = abcccbabd z_{18} = abad

A B-producing CTS, $B = \{z_4, z_6, z_8, z_{11}, z_{15}, z_{16}, z_{18}\}$.

$(x_1, x_2) \ast_{(2,2)} (z_2, z_1)$, $(z_1, z_2) \ast_{(4,4)} (z_4, z_3)$, $\boxed{(x_1, x_2) \ast_{(2,2)} (z_2, z_1),}$
$(z_2, x_2) \ast_{(4,2)} (z_7, z_8)$, $(z_3, z_7) \ast_{(2,1)} (z_5, z_6)$, $(x_2, x_3) \ast_{(3,3)} (z_{12}, z_{14})$,
$(z_8, z_{12}) \ast_{(2,5)} (z_9, z_{10})$, $(x_2, x_3) \ast_{(3,3)} (z_{12}, z_{14})$, $(x_2, x_3) \ast_{(3,3)} (z_{12}, z_{14})$,
$(z_{12}, z_{10}) \ast_{(2,1)} (z_{11}, z_{13})$, $(z_{12}, x_3) \ast_{(2,1)} (z_{15}, z_{16})$, $(x_1, x_3) \ast_{(3,1)} (z_{17}, z_{18})$.

$TD(A,B) \leq 12$

Translocation distance: Our solution

Compute *TD*(*A*,*B*)

***B* is a singleton:**
Let z be a string of length k and A be a set of cardinality n. There is an exact algorithm that computes $TD(A,z)$ in $O(kn)$ time and $O(kn)$ space.

***B* is an arbitrary set:** There is a 2-approximation algorithm for computing the translocation distance from two sets of strings.

Translocation distance: Our solution

Let $A = \{x_1, x_2, \ldots, x_n\}$ and z be an arbitrary string of length k

$$\begin{aligned} MaxSubLen(A, z, p) \;=\; & \max\{q \mid \exists\, 1 \leq i \leq n \text{ such that} \\ & x_i[p, p+q-1] = z[p, p+q-1]\}. \end{aligned}$$

Let $z \in TO_*(A)$; define iteratively the set $H(A, z)$ of intervals of natural numbers as follows:

1. $H(A, z) = \{[1, MaxSubLen(A, z, 1)]\}$;
2. Take the interval $[i, j]$ having the largest j; if $j = k$, then stop, otherwise put into $H(A, z)$ the new interval $[j+1, j+MaxSubLen(A, z, j+1)]$.

Note that we allow intervals of the form $[i, i]$ for some i to be in $H(A, z)$; moreover, for each $1 \leq i \leq k$ there are $1 \leq p \leq q \leq k$ (possibly the same) such that $i \in [p, q] \in H(A, z)$.

Lemma 4 *Let S be a z-producing CTS in $CO_*(A)$. Then,*

$$lg(S) \geq card(H(A, z)) - 1.$$

Translocation distance: Our solution

$$s_i = (x_i, y_i) \vdash_{p_i} (u_i, v_i)$$

$$\begin{aligned} A' &= \{x[MaxSubLen(A,z,1)+1,k] | x \in A\}, \\ z' &= z[MaxSubLen(A,z,1)+1,k]. \end{aligned}$$

For simplicity denote $r = MaxSubLen(A,z,1)$. Clearly, $H(A',z') = \{[i-r,j-r] | [i,j] \in H(A,z) \setminus \{[1,r]\}\}$, hence $card(H(A',z')) = card(H(A,z)) - 1$. Starting from S we construct a CTS in $CO_*(A')$, producing z' $S' = s'_1, s'_2, \ldots s'_m$ in the way indicated by the following procedure:

Translocation distance: Our solution

```
Procedure Construct_CTS(S,r);
begin
```
$m := 0;$

for $i := 1$ to q begin

  if $(p_i > r)$ then

   $m := m+1;\ s'_m = (x_i[r+1,k], y_i[r+1,k]) \vdash_{p_i - r} (u_i[r+1,k], v_i[r+1,k]);$

  endif;

endfor;

end.

Claim 1: *S′ is a CTS.*

Claim 2: *S′ is z′-producing.*

Translocation distance: Our solution

$p_{i_1}, p_{i_2}, \ldots, p_{i_m}$ are all integers from $\{p_1, p_2, \ldots, p_q\}$ bigger than r

$$F_{j-1}(S', x_{i_j}[r+1,k]) = \sum_{x[r+1,k]=x_{i_j}[r+1,k]} F_{i_j-1}(S,x) - card(X) - card(Y),$$

$$P_{j-1}(S', x_{i_j}[r+1,k]) = \sum_{x[r+1,k]=x_{i_j}[r+1,k]} P_{i_j-1}(S,x) - card(X) - card(Y),$$

where

$$X = \{t \leq i_j - 1 | p_t \leq r,\ u_t[r+1,k] = v_t[r+1,k] = x_{i_j}[r+1,k]\},$$
$$Y = \{t \leq i_j - 1 | p_t \leq r,\ u_t[r+1,k] = x_{i_j}[r+1,k] \text{ or } v_t[r+1,k] = x_{i_j}[r+1,k]\}.$$

Translocation distance: Our solution

Theorem 2 *Let* z *be a string of length* k *and* A *be a set of cardinality* n. *There is an exact algorithm that computes* $CD(A,z)$ *in* $O(kn)$ *time and* $O(kn)$ *space.*

Translocation distance: Our solution

Arbitrary Target Sets

Let A be a finite set of strings and $z \in CO_*(A)$; denote by

$$MaxPrefLen(A,z) = \begin{cases} |z|, \text{ iff } z \in A, \\ \max(\{q | q < |z|, \text{ there exists } x \in A, |x| > q, \\ \quad \text{so that } x[1,q] = z[1,q]\} \cup \{0\}), \end{cases}$$

$$MaxSufLen(A,z) = \max(\{q| \text{ there exists } x \in A, |x| \geq |z|,$$
$$\text{so that } x[|x|-q+1,|x|] = z[|z|-q+1,|z|]\}$$
$$\cup\{0\}),$$

$$ArbMaxSubLen(A,z,p) = \max(\{q| \text{ there exists } x \in A \text{ and } |x| \geq p+q$$
$$\text{such that } x[p,p+q-1] = z[p,p+q-1]\}$$
$$\cup\{0\}).$$

Translocation distance: Our solution

We define iteratively the set $ArbH(A, z)$ of intervals of natural numbers as follows, provided that all parameters defined above are nonzero:

1. $ArbH(A, z) = \{[1, MaxPrefLen(A, z)]\}$;

2. Take the interval $[i, j]$ having the largest j; if $j = |z|$, then stop. If $j < |z| - MaxSufLen(A, z)$, then put the new interval $[j + 1, j + ArbMaxSubLen(A, z, j + 1)]$ into $ArbH(A, z)$; otherwise put $[j + 1, |z|]$ into $ArbH(A, z)$.

Translocation distance: Our solution

Theorem 3 *1. Let A be a finite set of strings and B be a finite subset of* $TO_*(A)$. *Then* $\frac{\sum_{z\in B}(card(ArbH(A,z))-1)}{2} \leq TD(A,B) \leq \sum_{z\in B}(card(ArbH(A,z))-1)$.

2. There exist A and $B \subseteq TO_*(A)$ *such that* $TD(A,B) = \frac{\sum_{z\in B}(card(ArbH(A,z))-1)}{2}$.

3. There exist A and $B \subseteq TO_*(A)$ *such that* $TD(A,B) = \sum_{z\in B}(card(ArbH(A,z))-1)$.

Translocation distance: Our solution

Proof. 1. We shall prove the first assertion by induction on the length of the longest string in B, say k. The non-trivial relation is

$$\frac{\sum_{z \in B}(card(ArbH(A,z))-1)}{2} \leq TD(A,B). \qquad (*)$$

If $k = 1$, the relation $(*)$ is satisfied. Assume that the relation $(*)$ holds for any two finite sets X and Y, $Y \subseteq TO_*(X)$, all strings in Y being shorter than k. Assume that $B \setminus A = \{z_1, z_2, \ldots, z_m\}$ and let $S = s_1, s_2, \ldots, s_q$, $s_i = (x_i, y_i) \vdash_{p_i} (u_i, v_i)$, $1 \leq i \leq q$, be a $B \setminus A$-producing CTS in $TO_*(A)$. Note that at least one string in $B \setminus A$ should exist, otherwise the relation $(*)$ being trivially fulfilled.

Translocation distance: Our solution

Consider m new symbols $a_1, a_2, \ldots, a_m$ and construct the sets: $A' = \{x[1,r]a_i x[r+2,|x|] | x \in A, 1 \leq i \leq m\}$, $B' = \{z_i[1,r]a_i z_i[r+2,|z_i|] | 1 \leq i \leq m\}$,, where $r = \min\{p_i | 1 \leq i \leq q\}$. One can construct a B'-producing CTS in $TO_*(A')$ of the same length of S, say S' by applying a procedure $Convert$ illustrated by the next example

Translocation distance: Our solution

$B = \{abacdb,\ aabccb,\ bbaadc\},\ A = \{abbccb, aaaadb, bbbcdc\}.$

The $CTS\ S$ is

$$(abbccb, aaaadb) \vdash_2 (abaadb, aabccb), (abbccb, abaadb) \vdash_3 (abbadb, abaccb),$$
$$(bbbcdc, abaccb) \vdash_2 (bbaccb, abbcdc), (bbaccb, aaaadb) \vdash_3 (bbaadb, aaaccb),$$
$$(bbaadb, bbbcdc) \vdash_5 (bbaadc, bbbcdb), (abaadb, aaaccb) \vdash_2 (abaccb, aaaadb),$$
$$(abaccb, aaaadb) \vdash_4 (abacdb, aaaacb).$$

The procedure $Convert$ runs for $r = 2$ transforming this sequence into the sequence S':

$$(aba_2ccb, aaa_3adb) \vdash_2 (aba_3adb, aaa_2ccb), (aba_1ccb, aba_3adb) \vdash_3$$
$$(aba_1adb, aba_3ccb), (bba_1cdc, aba_3ccb) \vdash_2 (bba_3ccb, aba_1cdc),$$
$$(bba_3ccb, aaa_1adb) \vdash_3 (bba_3adb, aaa_1ccb), (bba_3adb, bba_1cdc) \vdash_5$$
$$(bba_3adc, bba_1cdb), (aba_1adb, aaa_1ccb) \vdash_2 (aba_1ccb, aaa_1adb),$$
$$(aba_1ccb, aaa_1adb) \vdash_4 (aba_1cdb, aaa_1acb).$$

Translocation distance: Our solution

Now S' is transformed into S'' for r previously defined. S'' is a B''-producing CTS in $CO_*(A'')$, where

$A'' = \{a_i x[r+2, |x|] | x \in A, 1 \leq i \leq m\}$, $\quad B'' = \{a_i z_i[r+2, |z_i|] | 1 \leq i \leq m\}$

For each $1 \leq i \leq m$ $card(ArbH(A'', a_i z_i[r+2, |z_i|]))$ is either $card(ArbH(A, z_i))$ or $card(ArbH(A, z_i)) - 1$.

Translocation distance: Our solution

$$card(ArbH(A'', a_i z_i[r+2, |z_i|])) = card(ArbH(A, z_i)) - 1$$

there exist at least one step in S' where the strings exchange prefixes of length at most r. It follows that $lg(S'') \leq lg(S') - \lceil t/2 \rceil$, where $t = card(\{i | card(ArbH(A'', a_i z_i[r+2, |z_i|])) = card(ArbH(A, z_i)) - 1\})$. Consequently,

$$\begin{aligned} lg(S) &= lg(S') \geq lg(S'') + \lceil t/2 \rceil \geq \\ & \frac{\sum_1^m (card(ArbH(A'', a_i z_i[r+2, |z_i|])) - 1)}{2} + \\ & \lceil t/2 \rceil \geq \frac{\sum_1^m (Arbcard(H(A, z_i)) - 1)}{2}. \end{aligned}$$

Translocation distance: Our solution

Theorem 4 *There is a 2-approximation algorithm for computing the translocation distance from two sets of strings.*

Translocation distance: Open problems

1. Is it possible to do it better?

2. Non-uniform translocation?

(i) Non-uniform translocation and unique markers:

2-approximation algorithm

(ii) This definition of translocation distance:

?

Thank You

READY FOR DISCUSSIONS