

Algorithms in Comparative Genomics

Universität Bielefeld, WS 2020/2021

Dr. Marília D. V. Braga · Leonard Bohnenkämper

<https://gi.cebitec.uni-bielefeld.de/teaching/2020winter/cg>

Exercise sheet 7, 10.12.2020

Exercise 1 (Optimal cover of component tree)

(7 pts)

Given canonical circular chromosomes

$$\mathbb{A} = (0 \ 2 \ \bar{7} \ 6 \ \bar{5} \ 3 \ 4 \ 8 \ 10 \ \bar{12} \ 9 \ \bar{11} \ 13 \ \bar{1} \ 14 \ 16 \ 21 \ 17 \ 19 \ 18 \ 20 \ 22 \ 27 \ 23 \ 25 \ 24 \ 26 \ 28 \ 15)$$

$$\mathbb{B} = (0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 16 \ 17 \ 18 \ 19 \ 20 \ 21 \ 22 \ 23 \ 24 \ 25 \ 26 \ 27 \ 28)$$

1. Draw the relational (or the breakpoint) diagram of \mathbb{A} and \mathbb{B} .

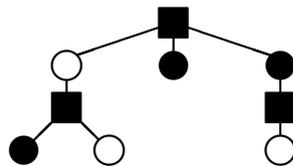
(You can use the Java program InversionVisualization provided on the course website: enter the values for chromosome \mathbb{A} , without the first value (0) and assume that the first vertex of the outputted diagram is 0^h and the last vertex is 0^t .)

2. Based on the diagram, construct both the chained component tree $\Upsilon_{\blacksquare}(\mathbb{A}, \mathbb{B})$ and the component tree $\Upsilon_{\circ}(\mathbb{A}, \mathbb{B})$.
3. Find an optimal cover (i.e. a cover with minimum cost) for $\Upsilon_{\circ}(\mathbb{A}, \mathbb{B})$.
4. Compute the inversion distance $d_{\text{INV}}(\mathbb{A}, \mathbb{B})$.

Exercise 2 (From a chained component tree to canonical chromosomes)

(3 pts)

Given the following chained component tree Υ'_{\blacksquare} :



Find canonical circular chromosomes \mathbb{A} and \mathbb{B} with the minimum number of genes such that

$$\Upsilon'_{\blacksquare} = \Upsilon_{\blacksquare}(\mathbb{A}, \mathbb{B}).$$

Exercise 3 (Canonical inversion sorting)

(5 pts)

Sort circular chromosome $\mathbb{A} = (0 \ 3 \ \bar{1} \ \bar{4} \ 2)$ into $\mathbb{B} = (0 \ 1 \ 2 \ 3 \ 4)$. Indicate all intermediate steps by drawing the respective overlap graphs, annotating each vertex with its corresponding score. Indicate your choice of a safe inversion by marking the corresponding vertex in the overlap graph.

(You may use InversionVisualization again! - again omitting the first value (0) and assuming that the first vertex of the outputted diagram is 0^h and the last vertex is 0^t .)