

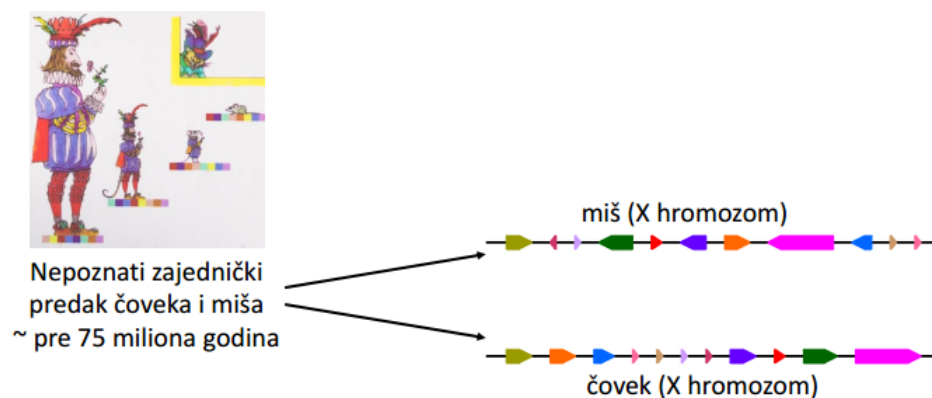
Bioinformatics chapter 6

May 2018

1 Transformacija čoveka u miša

Koji blokovi genoma su slični i kako da ih nađemo?

Kakav bi bio evolucionni scenario za transformisanje jednog genoma u drugi?

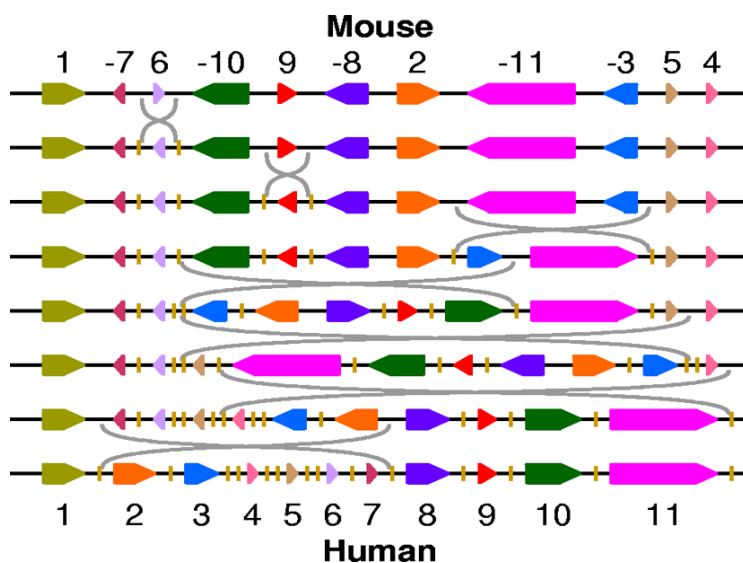


Slika 1: X hromozom miša i čoveka

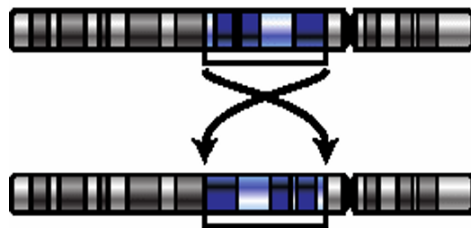
Kada je Šarl Pero opisao transformaciju čoveka u miša u delu "Mačak u čizmama", jedva je mogao očekivati da će 3 veka kasnije istraživanje pokazati da su ljudski i mišiji genom iznenađujuće slični.

Zapravo, ako bismo isekli 23 ljudska hromozoma na 280 delova i pomerali ove fragmente DNK, a zatim zalepili delove zajedno u novom redosledu, formiralo bi se 20 mišijih hromozoma. Međutim, evolucija nije koristila samo operaciju "*cut and-paste*", već manju promenu poznatu kao **preuređenje genoma**, što će biti naš fokus u ovom poglavlju.

Blokovi sintenije



Slika 2: Blokovi sintenije



Slika 3: Blokovi sintenije

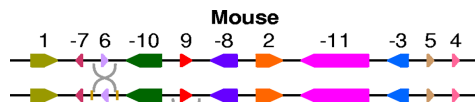
Svaki od jedanaest obojenih segmenata na slici predstavlja blok sličnih gena i naziva se **blok sintenije**. U nastavku će biti objašnjeno kako se izgrađuju blokovi sintenije i šta označavaju levi i desni pravac blokova.

Slika 2 prikazuje niz 7 promena koje transformišu mišiji X hromozom u ljudski X hromozom. Nažalost, ovaj niz od 7 promena predstavlja samo jedan od 1.070 različitih scenarija od 7 promena koji transformišu X hromozom miša u X hromozom čoveka.

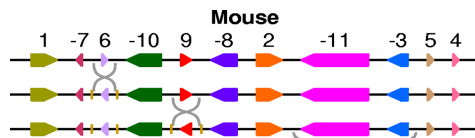
Možemo li pretvoriti X hromozom miša u ljudski X hromozom koristeći samo šest promena?

Bez obzira na to koliko promena razdvaja mišije i ljudske X hromosome, promene moraju biti *retki genomske događaji*. Zapravo, obično preuređeni genomi uzrokuju smrt ili sterilnost mutiranog organizma, čime se sprečava prenošenje preuređenja na narednu generaciju. Međutim, mali deo preuređenja genoma može imati pozitivan efekat na preživljavanje i propagirati se kroz vrstu kao rezultat prirodne selekcije. Kada stanovništvo postane izolovano od ostatka njene vrste dovoljno dugo, preuređenja mogu čak stvoriti i novu vrstu.

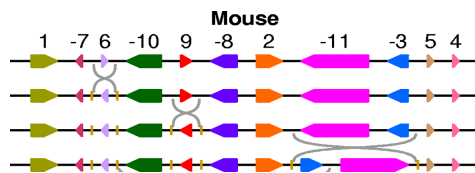
Promenu možemo zamisliti kao prekid genoma sa obe strane hromozomskog intervala, pomeranje intervala, a zatim lepljenje rezultujućih segmenata u novom redosledu.



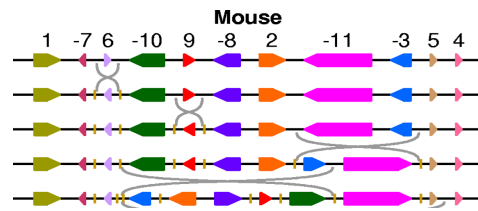
Slika 4: Promena 1



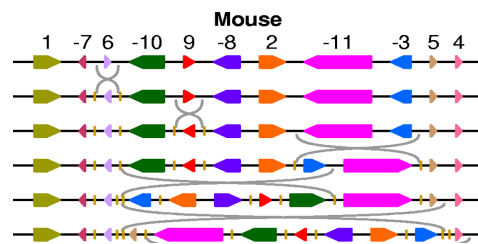
Slika 5: Promena 2



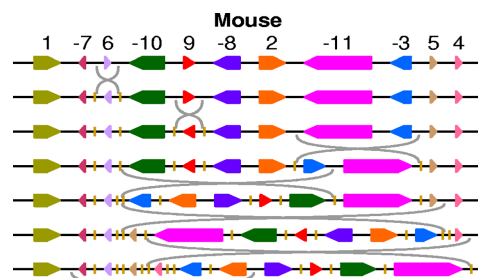
Slika 6: Promena 3



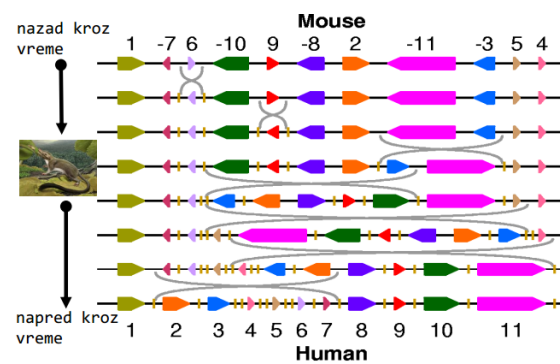
Slika 7: Promena 4



Slika 8: Promena 5



Slika 9: Promena 6



Slika 10: Završena transformacija

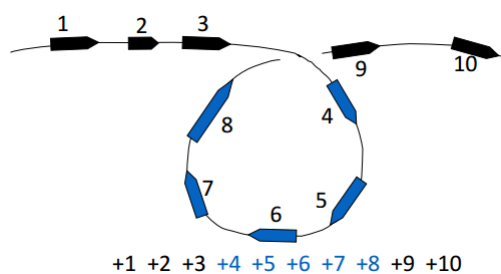
Slika 10 predstavlja transformaciju mišijeg X hromozoma u ljudski X hromozom sa sedam promena. Svaki sinteni blok je jedinstveno obojen i označen celim brojem između 1 i 11. Pozitivni ili negativni znak svakog celog broja ukazuje na smer sintenog bloka (pokazivanje desno ili levo, respektivno). Dva kratka vertikalna segmenta obeležavaju krajnje tačke obrnutog intervala u svakom preokretu.

Pretpostavimo da je evolucijski scenario tačan i recimo peti sinteni blok od vrha predstavlja uređenje hromozoma pretka. Zatim su se desile prve četiri promene na evolucionom putu od miša do zajedničkog pretka, a poslednje tri promene su se desile na evolucionom putu od zajedničkog pretka ka čoveku.

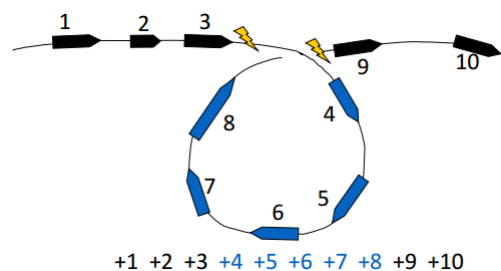
2 Sortiranje po promenama

Glavni problem je, kao što je pomenuto u uvodu, nalaženje minimalnog broja promena koje omogućavaju transformaciju X hromozoma miša u X hromozom čoveka.

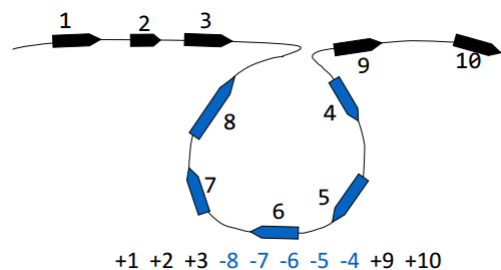
Možemo posmatrati niz blokova sintenije numerisanih kao na slici 11.



Slika 11: Blokovi sintenije

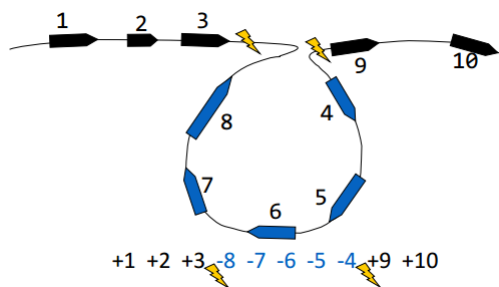


Slika 12



Slika 13

Nakon izvršene promene, dobijamo preuređen niz blokova sintenije u genomu.



Slika 14: 2 tačke prekida

Promene u genomu su dovele do stvaranja dve tačke prekida koje predstavljaju poremećaj u redosledu gena u genomu (Slika 14).

Posmatraćemo 2 scenarija sa različitim brojem promena.

Step 0:	2	-4	-3	5	-8	-7	-6	1
Step 1:	2	3	4	5	-8	-7	-6	1
Step 2:	2	3	4	5	6	7	8	1
Step 3:	2	3	4	5	6	7	8	-1
Step 4:	-8	-7	-6	-5	-4	-3	-2	-1
Step 5:	1	2	3	4	5	6	7	8

Slika 15: Scenario sa 5 promena

Step 0:	2	-4	-3	5	-8	-7	-6	1
Step 1:	2	3	4	5	-8	-7	-6	1
Step 2:	-5	-4	-3	-2	-8	-7	-6	1
Step 3:	-5	-4	-3	-2	-1	6	7	8
Step 4:	1	2	3	4	5	6	7	8

Slika 16: Scenario sa 4 promene

Definicija 1. *Rastojanje permutacija* je najmanji broj promena potrebnih za transformisanje jedne permutacije u drugu.

Naredni problem koji posmatramo je **problem sortiranja po promenama** koji predstavlja izračunavanje rastojanja između date permutacije i identične permutacije $(+1 +2 \dots +n)$

Ulaz: permutacija P

Izlaz: rastojanje između permutacije P i identične permutacije

Pohlepno sortiranje po promenama

Prva aproksimacija rastojanja između 2 permutacije je **pohlepno sortiranje po promenama**.

```

(+1 -7 +6 -10 +9 -8 +2 -11 -3 +5 +4)
(+1 -2 +8 -9 +10 -6 +7 -11 -3 +5 +4)
(+1 +2 +8 -9 +10 -6 +7 -11 -3 +5 +4)
(+1 +2 +3 +11 -7 +6 -10 +9 -8 +5 +4)
(+1 +2 +3 -4 -5 +8 -9 +10 -6 +7 -11)
(+1 +2 +3 +4 -5 +8 -9 +10 -6 +7 -11)
(+1 +2 +3 +4 +5 +8 -9 +10 -6 +7 -11)
(+1 +2 +3 +4 +5 +6 -10 +9 -8 +7 -11)
(+1 +2 +3 +4 +5 +6 -7 +8 -9 +10 -11)
(+1 +2 +3 +4 +5 +6 +7 +8 -9 +10 -11)
(+1 +2 +3 +4 +5 +6 +7 +8 +9 +10 -11)
(+1 +2 +3 +4 +5 +6 +7 +8 +9 +10 +11)

```

Slika 17: Pohlepno sortiranje

Prvi korak je da se izvrši promena koji postavlja +1 na pravo mesto (na prvu poziciju), a zatim slede promene koji postavljaju +2 na drugu poziciju, i tako dalje. Na primer, element 1 je već na pravom mestu i ima ispravan znak (+) u X hromozomu miša, ali element 2 nije na ispravnom položaju. Element 1 možemo zadržati fiksiran i premestiti element 2 na pravi položaj jednom promenom. Još jedna promena je potrebna da bi element 2 imao ispravan znak.

Daljim iteriranjem postupka dovodimo sve veće elemente na njihove ispravne pozicije.

Definicija 2. Element k u permutaciji $P = (p_1, p_2, \dots, p_n)$ je **sortiran**, ako je $p_k = k$, a u suprotnom je **nesortiran**.

Definicija 3. Permutacija P je **k -sortirana**, ako je prvih $k-1$ elemenata sortirano, a k -ti element nesortiran.

Sledeći primer pokazuje da je pohlepno sortiranje loša aproksimacija rastojanja između dve permutacije.

```
(-6 +1 +2 +3 +4 +5)
(-1 +6 +2 +3 +4 +5) step 1
(+1 +6 +2 +3 +4 +5) step 2
(+1 -2 -6 +3 +4 +5) step 3
(+1 +2 -6 +3 +4 +5) step 4
...
(+1 +2 +3 +4 + -6 +5) step 9
(+1 +2 +3 +4 + -5 +6) step 10
(+1 +2 +3 +4 + +5 +6)
```

Slika 18: Pohlepno sortiranje

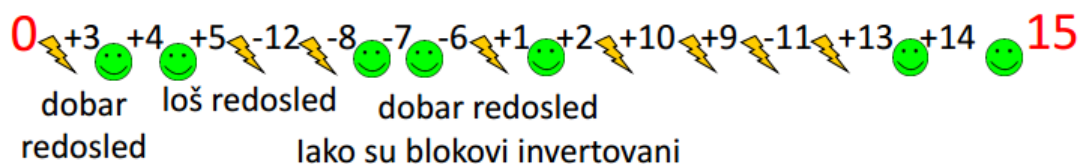
```
(-6 +1 +2 +3 +4 +5)
(-5 -4 -3 -2 -1 +6)
(+1 +2 +3 +4 +5 +6)
```

Slika 19: Kraći način

3 Teorema o prekidnoj tački

Uočimo da su uzastopni elementi (npr. $(+12 +13)$) poželjni, jer se javljaju u istom redosledu kao i u identičnoj permutaciji. Takodje, i $(-11 -10)$ su poželjni, jer se mogu inverzijom postaviti u pravi redosled. Ova dva para elemenata imaju zajedničku osobinu da je drugi element za 1 veci od prvog. Stoga, definišemo pojam suseda.

Definicija 4. (p_i, p_{i+1}) u permutaciji $P = (p_1, p_2, \dots, p_n)$ predstavljaju **susede**, ako je $p_{i+1} - p_i = 1$, a u suprotnom čine **prekid**.

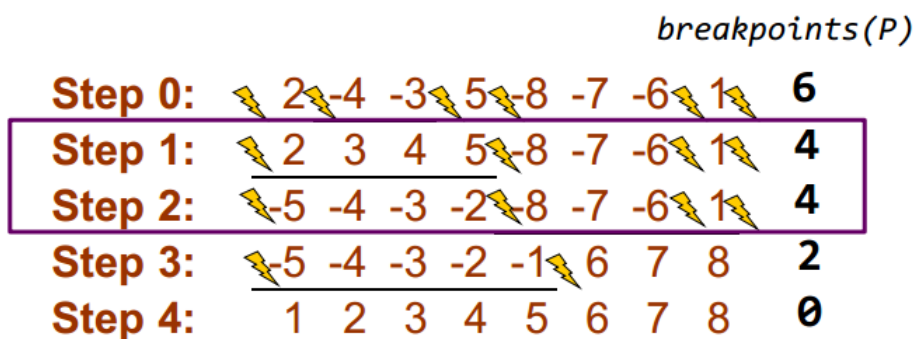


Slika 20: Susedi i prekidi

Važi:

$$adjacencies(P) + breakpoints(P) = |P| + 1$$

Sortiranje po promenama eliminacijom prekidnih tacaka



Slika 21

Koliko prekidnih tačaka može biti eliminisano u jednoj promeni?

Teorema o prekidnoj tački

$$\text{rastojanje između permutacija} \geq breakpoints(P)/2$$

- Nema garancije da će svaka promena eliminisati 2 prekidne tačke (step 2)
- Najveći broj promena bi bio za permutaciju $(+n + (n - 1) \dots + 1)$ i iznosi $n+1$ (gornja granica)
- Donja granica: $(n + 1)/2$

Velika razlika između donje i gornje granice nam sugerise da moramo preći na drugi način za rešavanje ovog problema.

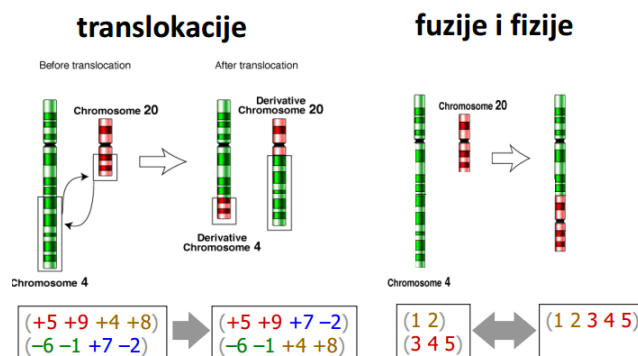
4 Preuređivanje u multihromozomalnim genomima

Umesto što posmatramo preuređivanje gena u okviru jednog hromozoma (hromozom X kod čoveka i miša), generalizujemo problem i posmatramo sve genome hromozoma.

U ovoj generalizaciji će biti više oblika preuređivanja blokova u genomu (do sada su bila samo obrtanja).

Problem je naizgled komplikovaniji, u nastavku će se ispostaviti da nije tako.

4.1 Translokacije, fuzije i fizije



Slika 22: Translokacije, fuzije i fizije

Za modeliranje translokacija posmatramo multihromozomalni genom sa k hromozoma kao permutaciju koja je podeljena na k delova.

Na primer, genom $(+1 +2 +3 +4 +5 +6) (+7 +8 +9 +10 +11)$ je sastavljen od dva hromozoma $(+1 +2 +3 +4 +5 +6)$ i $(+7 +8 +9 +10 +11)$.

Translokacija razmenjuje segmente različitih hromozoma, npr, translokacija dva hromozoma $(+1 +2 +3 +4 +5 +6) (+7 +8 +9 +10 +11)$ može dovesti do sledeća 2 hromozoma $(+1 +2 +3 +4 +9 +10 +11) (+7 +8 +5 +6)$. Možemo zamišljati translokaciju kao prvo cepanje svakog od hromozoma $(+1 +2 +3 +4 +5 +6) (+7 +8 +9 +10 +11)$ na 2 dela, $(+1 +2 +3 +4) (+5 +6) (+7 +8) (+9 +10 +11)$, a zatim lepljenje rezultujućih segmenata u 2 nova hromozoma, $(+1 +2 +3 +4 +9 +10 +11) (+7 +8 +5 +6)$.

Preuređenja u multihromozomalnim genomima nisu ograničena na promene i translokacije. Ona takođe uključuju hromozomske fuzije, koje spajaju 2 hromozoma u 1, kao i fisije, koje dele 1 hromozom na 2 hromozoma.

Na primer, 2 hromozoma $(+1 +2 +3 +4 +5 +6) (+7 +8 +9 +10 +11)$ mogu biti fuzionisani (spojeni) u 1 hromozom $(+1 +2 +3 +4 +5 +6 +7 +8 +9 +10 +11)$. Sledeća fisija ovog hromozoma može dovesti do 2 hromozoma $(+1 +2 +3 +4) (+5 +6 +7 +8 +9 +10 +11)$.

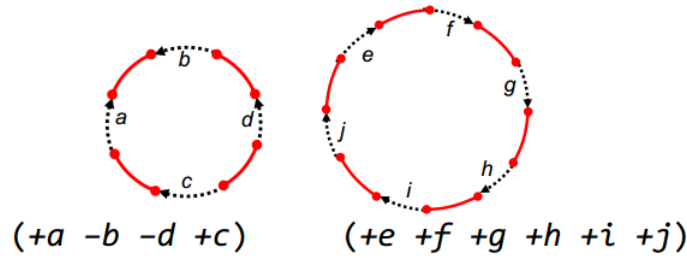
Pre pet miliona godina, ubrzo nakon razdvajanja čoveka i šimpanze, fuzija dva hromozoma (nazvana 2A i 2B) u jednom od naših predaka stvorila je ljudski hromozom 2 i smanjila broj hromozoma sa 24 na 23.

5 Problem rastojanja 2-prekida

5.1 Od linearnih do cirkularnih hromozoma

Sada se fokusiramo na jedan od hromozoma u multihromozomalnom genomu i razmotrimo transformacije promene kružnog hromozoma $P = (+ a -b -c + d)$ u $Q = (+ a -b -d + c)$.

Uvodimo pojam crnih usmerenih i crvenih neusmerenih grana.

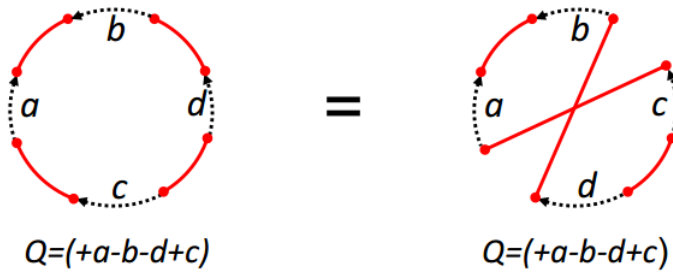


Slika 23: Hromozomi P i Q

Crne usmerene grane predstavljaju blokove sintenije.

Crvene neusmerene grane povezuju susedne blokove sintenije.

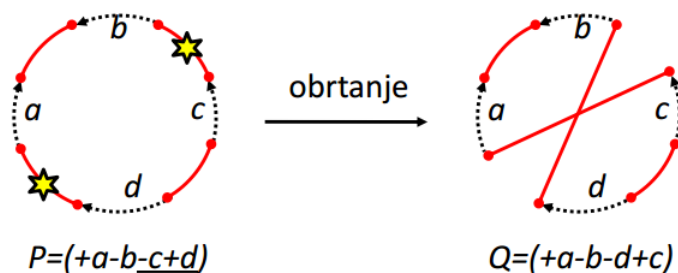
Možemo nacrtati Q na različite načine, zavisno od toga kako se odlučimo da uredimo crne grane. Slika 24 prikazuje dve takve ekvivalentne reprezentacije.



Slika 24: Ekvivalentne reprezentacije hromozoma Q

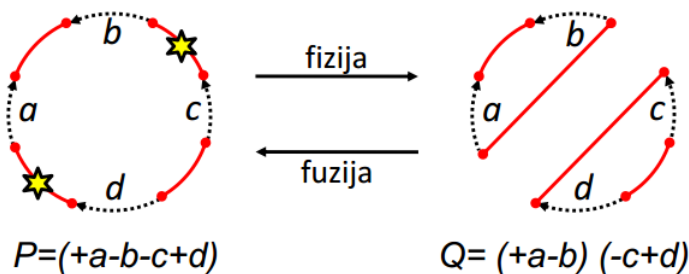
Iako je prvi crtež Q na slici njegova najprirodnija reprezentacija, koristićemo drugu reprezentaciju, jer su joj crne grane raspoređene kružno u potpuno istom redosledu kako se pojavljuju u prirodnoj reprezentaciji $P = (+a -b -c +d)$.

Kao što je prikazano na slici 25, fiksiranje crnih grana omogućava nam da vizualizujemo efekat promena. Kao što možemo videti, promena briše ("deli") dve crvene grane iz P (povezivanje b sa c i d sa a) i zamenjuje ih sa dve nove crvene grane (povezivanje b sa d i c sa a). Ova promena se naziva **obrtnanje**.



Slika 25: Obrtnanje

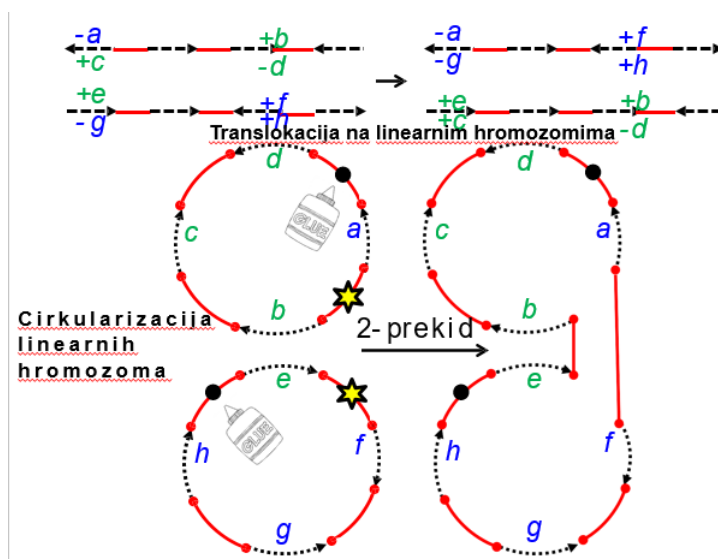
Slika 26 ilustruje fiziju $P = (+ a -b -c + d)$ u $Q = (+ a -b) (- c + d)$. Inverzna operacija fiziji odgovara fuziji dva hromozoma iz Q u hromozom P. Operacije fuzije i fizije, kao i promene, odgovaraju brisanju dve grane u jednom genomu i njihovim zamenjivanjem sa 2 nove grane u drugom genomu.



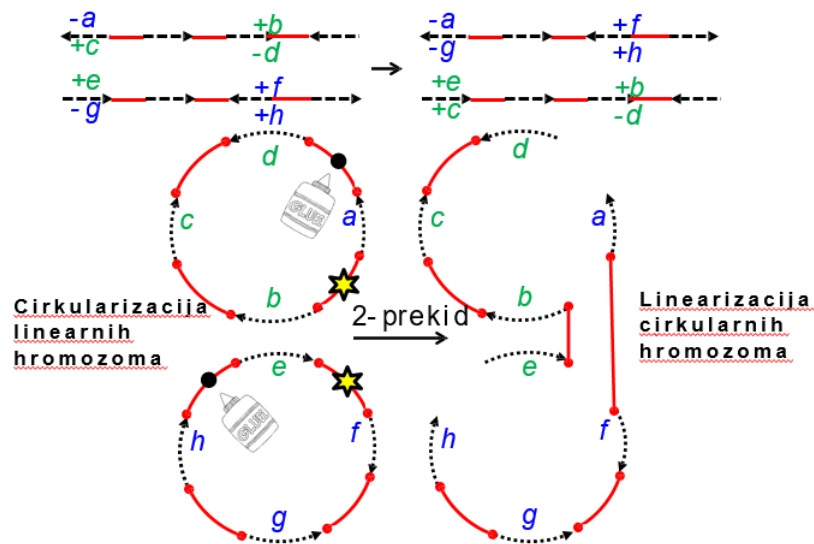
Slika 26: Fizija i fuzija

Translokacija koja uključuje dva linearna hromozoma takođe se može simulirati cirkularizacijom ovih hromozoma, a zatim zamenjivanjem dve crvene grane sa dve različite crvene grane, kao što je prikazano na slici 27. Zbog toga se mogu objediniti ova 4 različita tipa preuređenja. Svi oni se mogu posmatrati kao cepanje 2 crvene grane grafa genoma i zamena sa dve nove crvene grane na ista 4 cvora.

Iz tog razloga definišemo opštu operaciju na grafu genoma koja zamenjuje crvenu granu sa dve nove crvene grane pri čemu čvorovi ostaju isti i nazivamo je **2-prekid**.



Slika 27: 2-prekid



Slika 28: Objedivanje sva 4 preuredjenja

5.2 Rastojanje 2-prekida

Definicija 5. *Rastojanje 2-prekida $d(P, Q)$: Minimalni broj 2-prekida koji transformišu genom P u genom Q .*

Definicija 6. *Problem rastojanja 2-prekida: Naći rastojanje 2-prekida između dva genoma.*

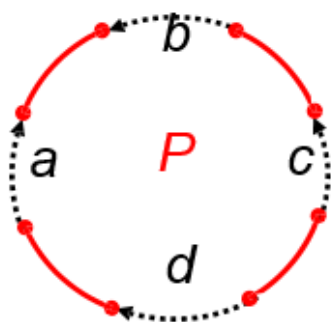
Ulaz. Dva genoma nad istim skupom blokova sintenije.

Izlaz. Rastojanje 2-prekida između ovih genoma.

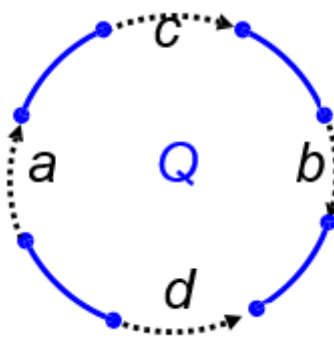
6 Grafovi prekidnih tačaka

Za izračunavanje rastojanja 2-prekida konstruisaćemo graf za upoređivanje dva genoma.

Posmatrajmo genome P i Q .

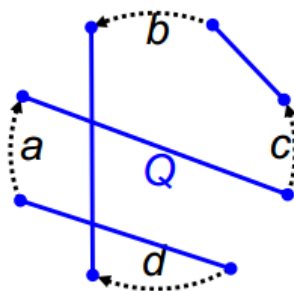


Slika 29: Genom P



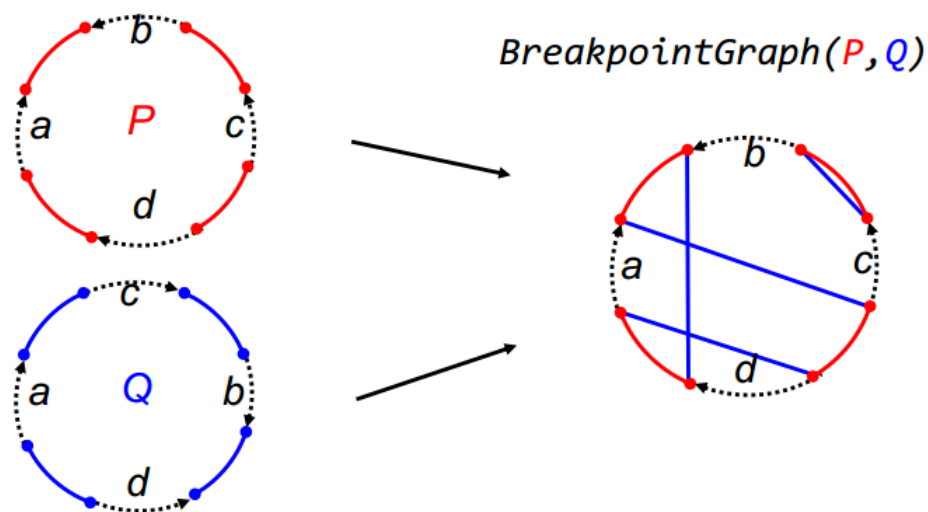
Slika 30: Genom Q

Genom Q mozemo predstaviti i na drugi nacin.



Slika 31: Drugačija reprezentacija genoma Q

Nadgradnjom genoma P i Q dobijamo $BreakpointGraph(P, Q)$.



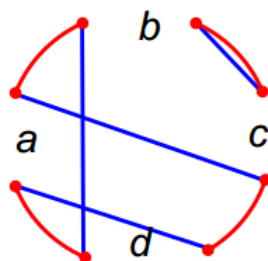
Slika 32: Nadgradnja genoma P i Q

Crvene i crne grane u grafu prekidnih tačaka formiraju genom P .

Plave i crne grane u grafu prekidnih tačaka formiraju genom Q .

Crvene i **plave** grane formiraju **alternirajuće crveno-plave** cikluse.

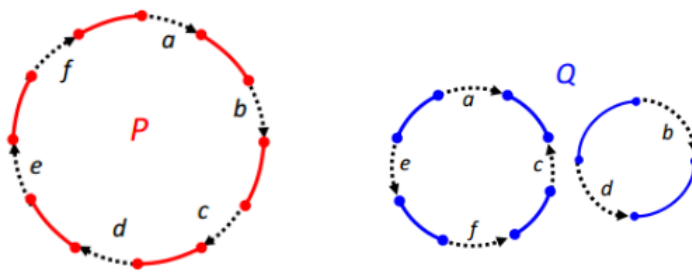
$BreakpointGraph(P, Q)$



Slika 33: Alternirajući crveno-plavi ciklusi

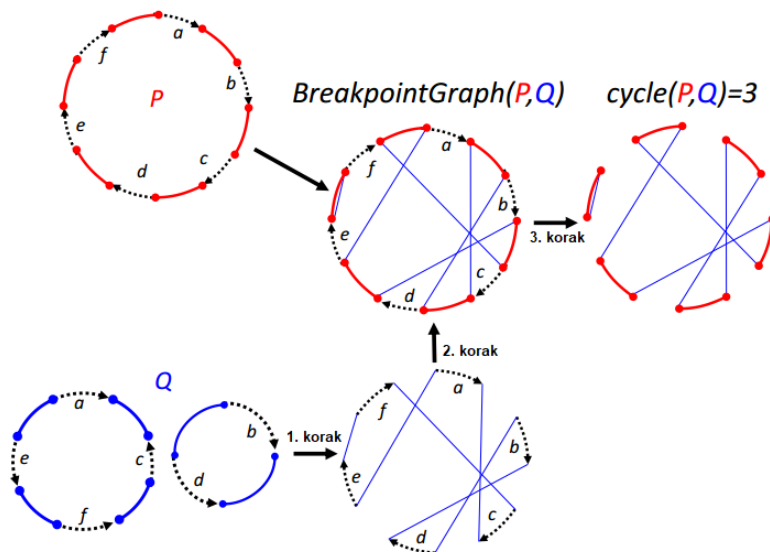
Koristimo oznaku $\text{cycle}(\mathbf{P}, \mathbf{Q})$: broj alternirajućih crveno-plavih ciklusa.

Šta predstavlja $\text{cycle}(\mathbf{P}, \mathbf{Q})$?



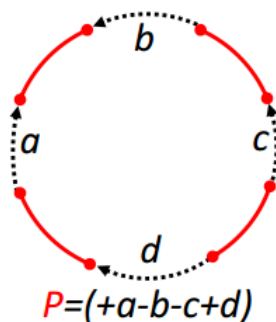
Slika 34: Grafovi genoma P i Q

1. korak: Poređamo crne grane genoma Q u isti redosled kao u genomu P
2. korak: Nadgradnja genoma P i Q u jedan
3. korak: Uklanjanje crnih grana



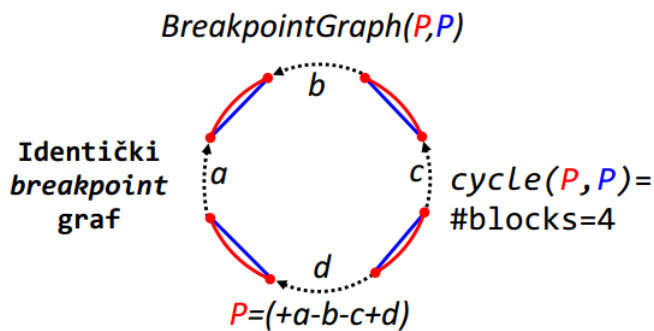
Slika 35: $\text{cycle}(\mathbf{P}, \mathbf{Q})$

Za dato P , koje Q maksimizuje $\text{cycle}(P, Q)$?



Slika 36: Genom P

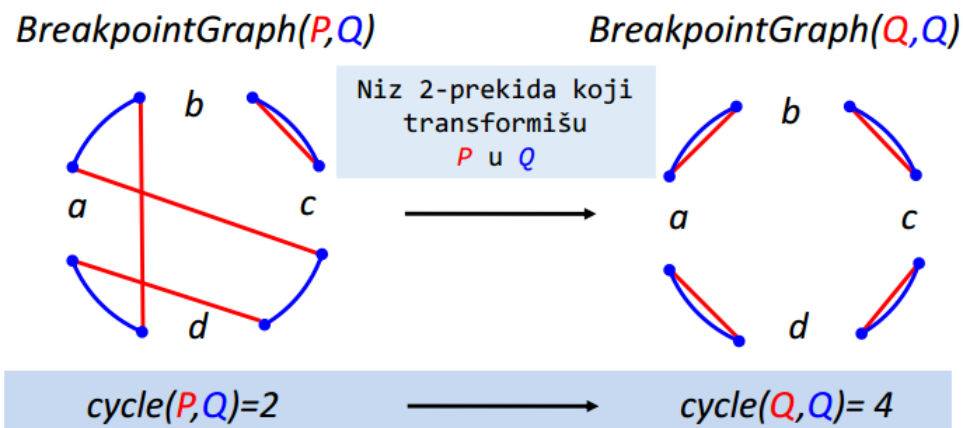
U slučaju da P i Q imaju isti broj blokova sintenije, označimo taj broj sa $\mathbf{Blocks}(P, Q)$. Ako su P i Q identični, njihov graf prekidnih tačaka se sastoji od $\text{Blocks}(P, Q)$ ciklusa dužine 2 od kojih svaki sadrži jednu crvenu i jednu plavu granu. Cikluse dužine 2 nazivamo **identičkim ciklusima**, a graf prekidnih tačaka formiran na osnovu identičkih genoma nazivamo **identičkim grafom prekidnih tačaka**.



Slika 37: Identički graf prekidnih tačaka

Preuređenje genoma utiče na crveno-plave cikluse.

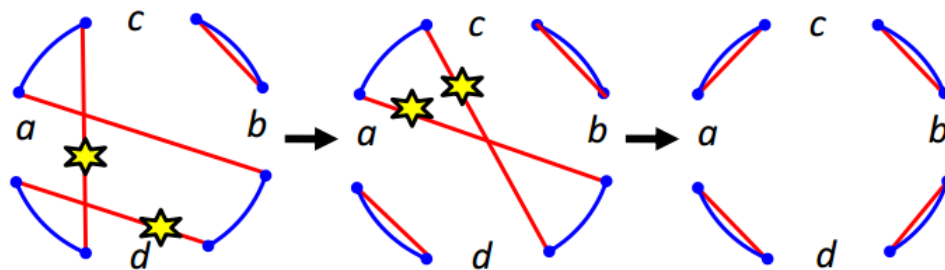
Svaka transformacija $P \rightarrow Q$ odgovara transformaciji:



Slika 38: Transformacija $P \rightarrow Q$

Preuređenja genoma takođe utiču i na $cycle(P, Q)$:

$$\begin{aligned}
 P &= (+a -b -c +d) \rightarrow P' = (+a -b -c -d) \rightarrow P'' = Q = (+a +c +b -d) \\
 BreakpointGraph(P, Q) &\rightarrow BreakpointGraph(P', Q) \rightarrow BreakpointGraph(Q, Q) \\
 cycle(P, Q) &= 2 \rightarrow cycle(P', Q) = 3 \rightarrow cycle(Q, Q) = 4
 \end{aligned}$$



Slika 39: Uticaj preuređenja na $cycle(P, Q)$

7 Teorema o rastojanju 2-prekida

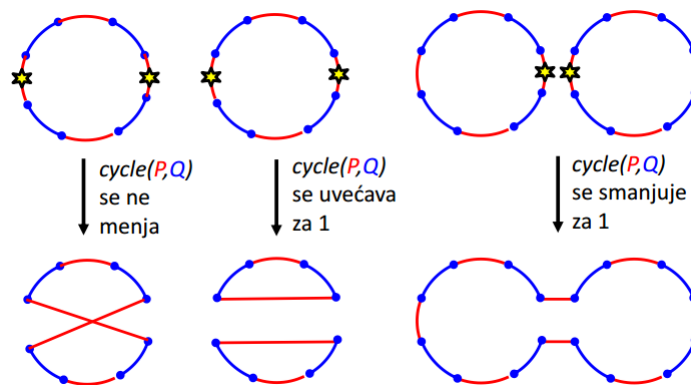
Posmatramo problem sortiranja po 2-prekidima:

$$\begin{array}{c}
 \text{2-prekidi} \\
 P \rightarrow \dots \rightarrow Q \\
 \text{BreakpointGraph}(P, Q) \rightarrow \dots \rightarrow \text{BreakpointGraph}(Q, Q) \\
 \text{cycle}(P, Q) \rightarrow \dots \rightarrow \text{cycle}(Q, Q) = \text{blocks}(Q, Q) \\
 \text{blocks}(P, Q) - \text{broj blokova koji u\u0107estvuje u izgradnji P i Q} \\
 \text{broj crveno-plavih ciklusa se uve\u0107ava za} \\
 \text{blocks}(P, Q) - \text{cycle}(P, Q)
 \end{array}$$

Slika 40: Sortiranje po 2-prekidima

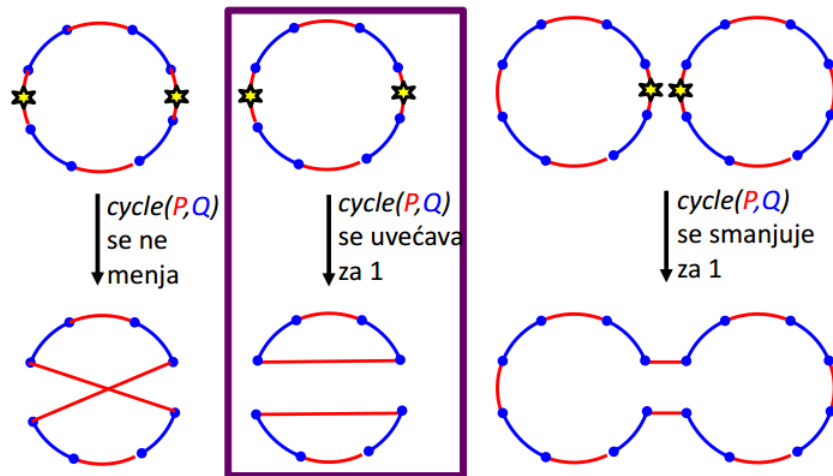
Koliko mo\u017ee svaki 2-prekid da doprinese ovom uve\u0107anju?

2-prekid mo\u017ee izmeniti $\text{cycle}(P, Q)$ za 1.



Slika 41

Postoji 2-prekid povecanje velicine $\text{cycle}(P, Q)$ za 1



Slika 42

Teorema o rastojanju 2-prekida

- Svaki 2-prekid povećava broj ciklusa najviše za 1
- Za svaki 2-prekid postoji povećanje broja ciklusa za tačno 1
- Svako sortiranje po 2-prekidima mora povećati broj ciklusa za $\text{blocks}(P, Q) - \text{cycle}(P, Q)$
- 2-prekid rastojanje između genoma P i Q:

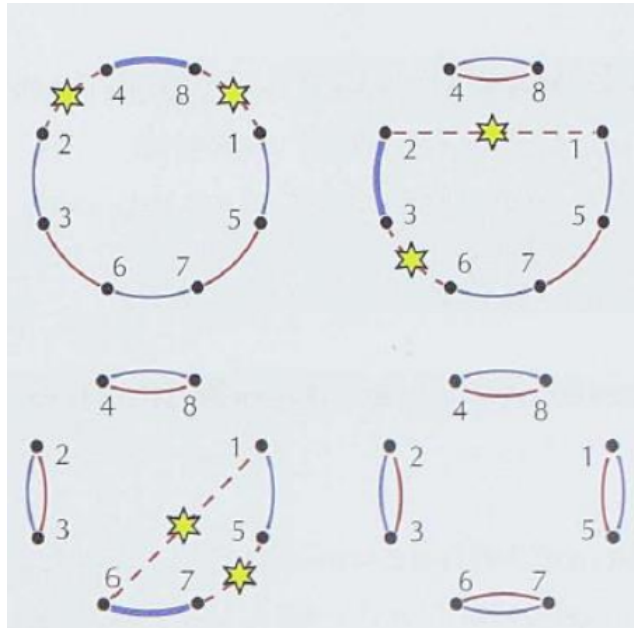
$$d(P, Q) = \text{blocks}(P, Q) - \text{cycle}(P, Q)$$

Rastojanje 2-prekida između genoma čoveka i miša

- Genomi čoveka i miša se mogu rastaviti na 280 blokova sintenije (dužine bar pola miliona nukleotida)
- Graf prekidnih tačaka nad ovim blokovima ima ukupno 35 ciklusa
- Na osnovu teoreme o rastojanju 2-prekida:

$$d(H, M) = \text{blocks}(H, M) - \text{cycle}(H, M) = 280 - 35 = 245$$
- Postoje različite verzije scenarija sa 245 koraka.
- Pravi evolutivni scenario je možda imao i više od 245 koraka.

Shortest rearrangement scenario



Slika 43: Shortest rearrangement scenario

ShortestRearrangementScenario(P, Q)**output P***RedEdges* \leftarrow *ColoredEdges*(*P*)*BlueEdges* \leftarrow *ColoredEdges*(*Q*)*BreakpointGraph* \leftarrow the graph formed by *RedEdges* and *BlueEdges***while** *BreakpointGraph* has a non-trivial cycle *Cycle*(j, i') \leftarrow an arbitrary edge from *BlueEdges* in a nontrivial red-blue cycle(i, j) \leftarrow an edge from *RedEdges* originating at node j(i', j') \leftarrow an edge from *RedEdges* originating at node i'*RedEdges* \leftarrow *RedEdges* with edges(i, j) and (i', j') removed*RedEdges* \leftarrow *RedEdges* with edges(j, i') and (j', i) added*BreakpointGraph* \leftarrow the graph formed by *RedEdges* and *BlueEdges**P* \leftarrow **2-BreakOnGenome**(**P, i, i', j, j'**)**output P**

2 – *BreakOnGenome*(*P, i, i', j, j'*) - uklanja grane (*i, i'*) i dodaje grane (*i, j*) i (*i', j'*) (genom predstavljen grafom prekidnih tacaka)

8 Zadaci sa vezbi

8.1 ChromosomeToCycle

```
def chromosome_to_cycle(chromosome):

    nodes = [0 for i in range(2*len(chromosome))]
    #gde je -1 u knjizi u algoritmu, pisemo -2

    for j in range(len(chromosome)):
        i = chromosome[j]
        if i > 0:
            #dodajemo cvorove
            nodes[2*j - 1] = 2*i - 1
            nodes[2*j] = 2*i
        else:
            nodes[2*j - 1] = -2*i
            nodes[2*j] = -2*i - 1

    return nodes

def main():
    print(chromosome_to_cycle([1, -2, -3, 4]))

if __name__ == "__main__":
    main()
```

8.2 CycleToChromosome

```
def cycle_to_chromosome(nodes):

    chromosomes = [0 for i in range(len(nodes)//2)]

    for j in range(len(nodes)//2):
        if nodes[2*j] < nodes[2*j + 1]:
            chromosomes[j] = nodes[2*j + 1] // 2
        else:
            chromosomes[j] = -nodes[2*j] // 2
```

```

    return chromosomes

def main():
    print(cycle_to_chromosome([1,2, 4, 3, 5, 6, 7 ,8]))

if __name__ == "__main__":
    main()

```

8.3 GreedySorting

```

def find(P, start, n):
    for i in range(start, len(P)):
        if P[i] == n or P[i] == -n:
            return i

def reversal(P, start, stop):
    rev = [-i for i in P[start:stop+1]]
    rev.reverse()
    P[start:stop+1] = rev

    return P

def greedy_sorting(P):
    approx_reversal_distance = 0

    print(P)

    for k in range(len(P)):
        if P[k] != k+1: #jer brojimo od 1
            i = find(P, k, k+1)
            P = reversal(P, k, i) #apply red iz
    ↪ algoritma
        approx_reversal_distance += 1
        print(P)

        if P[k] < 0:
            P[k] = -P[k]
            approx_reversal_distance += 1

```

```

        print(P)

    return approx_reversal_distance

def main():
    # P = [+1, -7, +6, -10, +9, -8, +2, -11, -3, +5, +4]
    # P = [+6, -7, +1, -10, +9, -8, +2, -11, -3, +5, +4]
    P = [-2, -5, +3, -4, +1]

    print(greedy_sorting(P))

if __name__ == "__main__":
    main()

```

8.4 ShortestRearrangementScenario

```

def cycle_to_chromosome(nodes):

    chromosomes = [0 for i in range(len(nodes)//2)]

    for j in range(len(nodes)//2):
        if nodes[2*j] < nodes[2*j + 1]:
            chromosomes[j] = nodes[2*j + 1] // 2
        else:
            chromosomes[j] = -nodes[2*j] // 2

    return chromosomes

def cycle_to_chromosome(nodes):

    chromosomes = [0 for i in range(len(nodes)//2)]

    for j in range(len(nodes)//2):
        if nodes[2*j] < nodes[2*j + 1]:
            chromosomes[j] = nodes[2*j + 1] // 2
        else:
            chromosomes[j] = -nodes[2*j] // 2

    return chromosomes

```

```

def color_edges(P):
    edges = {}

    for chrom in P:
        nodes = chromosome_to_cycle(chrom)

        for j in range(len(chrom)):
            edges[nodes[2*j - 1]] = nodes[2*j]
            edges[nodes[2*j]] = nodes[2*j - 1]

    return edges

def graph_to_genom(genome_graph):
    P = []
    for nodes in genome_graph:
        chromosome = cycle_to_chromosome(nodes)
        P.append(chromosome)
    return P

def main():
    print(color_edges([[1,-2,-3,4]]))

if __name__ == "__main__":
    main()

```