

УНИВЕРЗИТЕТ У БЕОГРАДУ
МАТЕМАТИЧКИ ФАКУЛТЕТ



Филип Јовашевић

АЛГОРИТМИ ЗА ПРЕУРЕЂИВАЊЕ
ГЕНОМА - ЕЛЕКТРОНСКА ЛЕКЦИЈА

мастер рад

Београд, 2024.

Ментор:

др Јована КОВАЧЕВИЋ, ванредни професор
Универзитет у Београду, Математички факултет

Чланови комисије:

др Мирјана МАЉКОВИЋ РУЖИЧИЋ, доцент
Универзитет у Београду, Математички факултет

Невена ТИРИЋ, асистент
Универзитет у Београду, Математички факултет

Датум одбране: . септембар 2024.

Најмилијима

Наслов мастер рада: Алгоритми за преуређивање генома - електронска лекција

Резиме: Рад је осмишљен као интерактивно штиво којим се објашњавају могући сценарији процеса и обрасци преуређивања генома који су се догодили природним путем пре више десетина милиона година. Алгоритмима за преуређивање генома претходи обрада теоријских основа задате теме. Корисник може тестирати дефинисане алгоритме уносом различитих података и уз помоћ конкретних примера боље разумети материју. Специфични циљ рада је да се креирањем електронске лекције да прилог електронском уџбенику који би се користио као додатни ресурс приликом наставе и припремања испита из области биоинформатике, и за ширење знања свих студената заинтересованих за ову област. Сама лекција је комбинација теоријских објашњења и имплементације алгоритама из поменуте области. Алати за израду лекције су *Typescript*, *React frontend framework* и *NestJs backend framework*.

Кључне речи: еволуција, ДНК, ген, геном, хромозом, преуређивање, преокрети, места преуређивања, сортирање, хеуристика, прекидне тачке, транслокација, фузија, физија, секвенцирање, блокови синтеније, заједнички предак, места преуређивања, прекидни граф, 2-прекид

Садржај

1	Увод	1
2	Дефиниције	2
2.1	Основни појмови из генетике	2
2.2	Преуређивање генома	5
3	Трансформација генома миша у геном човека	7
3.1	Преуређивање кроз итерације	8
3.2	Места преуређивања	10
4	Преуређивање генома	13
4.1	Сортирање помоћу преокрета	13
4.2	Похлепна хеуристика	14
4.3	Прекидне тачке	19
4.4	Транслокација, фузија и физија	22
4.5	Графовска репрезентација генома	24
4.6	Места преуређивања у људском геному	43
4.7	Формирање блокова синтеније	44
5	Закључак	47
	Библиографија	49

Глава 1

Увод

Сви геноми у живим организмима могу се променити под утицајем унутрашњих или спољашњих фактора. Због тога се састав генома обично дефинише као динамички ентитет и претпоставља се да је он више пута мењан и преуређен од почетка живота на планети [23]. Преуређивање генома представља основу еволуционих процеса и заснива се на померању његових фрагмената који могу да садрже један или више гена, такозваних блокова синтеније. У класичној генетици, гени синтеније су првобитно дефинисани као гени који леже на истом хромозому. Данас, међутим, биолози обично говоре о синтенији као о очувању блокова са истим редоследом гена унутар два сегмента хромозома различитих организама који се међусобно упоређују [18]. Овај принцип редоследа гена у еволутивно сродним врстама подразумева да се гени позиционирани један близу другог на геному једне врсте, вероватно, налазе близу један другом и код друге врсте [22]. Претпоставља се да су се променама унутар блокова синтеније по одређеном обрасцу геноми током еволуције преуређивањем мењали и на тај начин еволуирали од изумрлог заједничког претка до различитих врста организама које данас постоје.

Коришћење синтенијског мапирања за праћење еволуционе историје генома је приступ којим може да се сагледа генетски састав давно изумрлих врста организама који нису оставили своју стварну, нетакнуту ДНК, већ само донекле испреpletене ланце хромозома њихових потомака [18], док се алгоритмима за преуређивање генома врши симулација померања блокова по итерацијама и приказује се највероватнији еволутивни процес који је у природним условима трајао око више десетина милиона година.

Глава 2

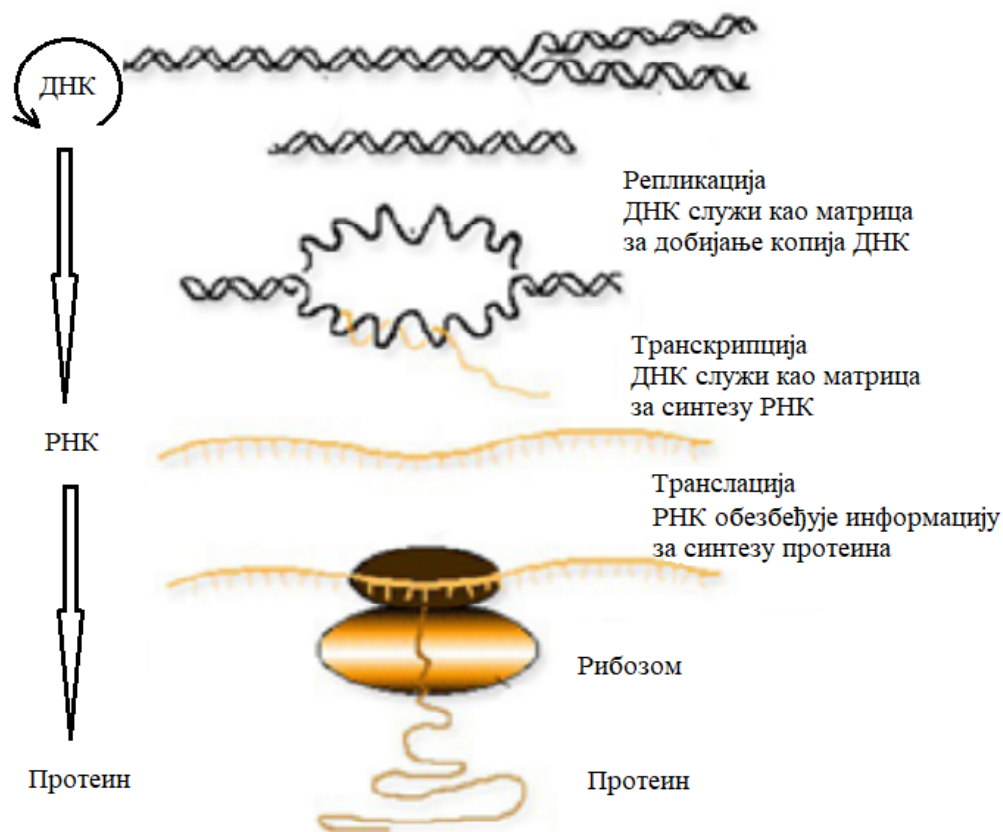
Дефиниције

2.1 Основни појмови из генетике

У наставку су објашњени основни појмови из генетике, чије је познавање потребно како би се схватили процеси описани у раду:

- Нуклеотид је основни градивни блок нуклеинских киселина (ДНК и РНК). Нуклеотид се састоји од молекула шећера (рибозе у РНК или дезоксирибозе у ДНК) везаног за фосфатну групу и азотне базе [11]. Азотне базе које улазе у састав нуклеотида и на основу којих се нуклеотиди разликују су аденин, цитозин, гуанин, тимин и урацил, обично скраћено означене као *A, C, G, T* и *U* [14].
- ДНК - Дезоксирибонуклеинска киселина се састоји од мањих јединица, нуклеотида. Четири азотне базе нуклеотида везују се у парове код наспрамних ланаца ДНК молекула, налик на пречке мердевина. Сваки молекул ДНК садржи два испреплетана ланца нуклеотида. Парови нуклеотида између сваког ланца увек се комбинују на специфичан начин, тако да се аденин на једном ланцу ДНК увек упарује са тимином на супротном ланцу ДНК. Исто тако гуанин на једном ланцу ДНК се увек упарује са цитозином на супротном ланцу [1].
- РНК - Рибонуклеинска киселина има значајну улогу у кодирању, декодирању, регулацији и експресији гена. Као и ДНК, молекули РНК се састоје од дугих ланаца нуклеотида, с тим што су РНК молекули једноланчани и тимин је замењен урацилом [10].

- Ген представља сегмент ДНК састављен од низа нуклеотида који представља основну јединицу генетског материјала. Људи имају две копије сваког гена, по једну наслеђену од сваког родитеља, што значи да је ген основна физичка и функционална јединица наслеђа.
- Протеини су велики, сложени молекули који обављају различите функције у ћелијама и потребни су за структуру, функцију и регулацију ћелијских процеса. Они се састоје од стотина или хиљада мањих јединица, аминокиселина, које су повезане једна за другу у дугим ланцима. Редослед аминокиселина одређује јединствену тродимензионалну структуру сваког протеина и његову специфичну функцију [16].

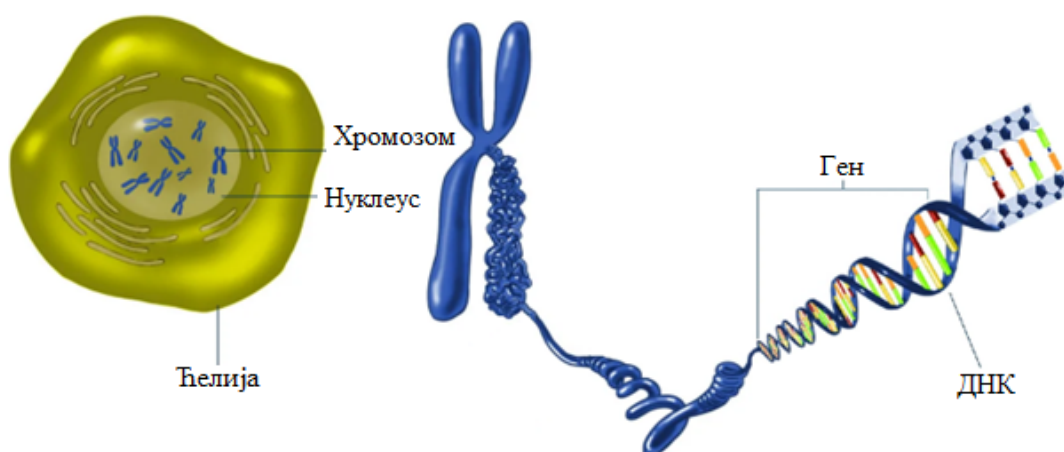


Слика 2.1: Централна догма [12]

- Синтеза протеина одвија се у рибозомима, у цитоплазми ћелије, док се њихови гени налазе у једру ћелије. Услед тога, као посредници између гена са једне и протеина са друге стране учествују молекули РНК. У процесу транскрипције се ген са ДНК преписује на РНК, а затим проце-

сом транслације преводи у ланце аминокиселина које чине протеин [6]. На слици 2.1 је приказан процес синтезе протеина познат под називом централна догма. Природа протеина је строго утврђена и одређена генима. Око 19900 гена код човека дају упутства за стварање протеина који су потребни за функционисање организма, при чему многи гени не кодирају протеине већ помажу у контроли других гена [17].

- Генетски код је мапирање које сваком триплету нуклеотида додељује аминокиселину или стоп кодон. Према овом коду се врши синтеза протеина на основу гена, односно секвенце нуклеотида од којих је ген састављен. Од 64 могућа кодона, 61 кодира аминокиселине, а три представљају тзв. стоп кодоне који указују на крај протеина [5].
- Ортологи гени или ортолози су гени код различитих врста који су еволуирали од истог гена код заједничког претка тих врста. Они су сачували своје главне функције током еволуције [3].
- Хромозом - Сваки ланац ДНК је чврсто умотан око протеина који се зове хистон. Ова намотана структура се назива хромозом. Свака људска ћелија обично садржи 23 пара хромозома, при чему по један од сваког пара даје сваки родитељ [1]. За разлику од људи, шимпанзе и мишеви на пример, имају респективно 20 и 24 пара хромозома.



Слика 2.2: Структура генома [19]

- Геном или генотип је скуп свих гена организма и он садржи комплетан наследни материјал једног организма. Упрошћена структура генома је приказана на слици 2.2. Геном је јединствен за сваког појединца. С обзиром на то шта чини геном, изводи се закључак да се промене на геному дешавају променама распореда гена на хромозомима тј. преуређивањем хромозома.
- Секвенцирање је метода помоћу које се одређује секвенца ДНК, односно редослед нуклеотида који чине ДНК молекула и користи се за одређивање делова или целокупног генетског састава одређеног организма или типа ћелије.

2.2 Преуређивање генома

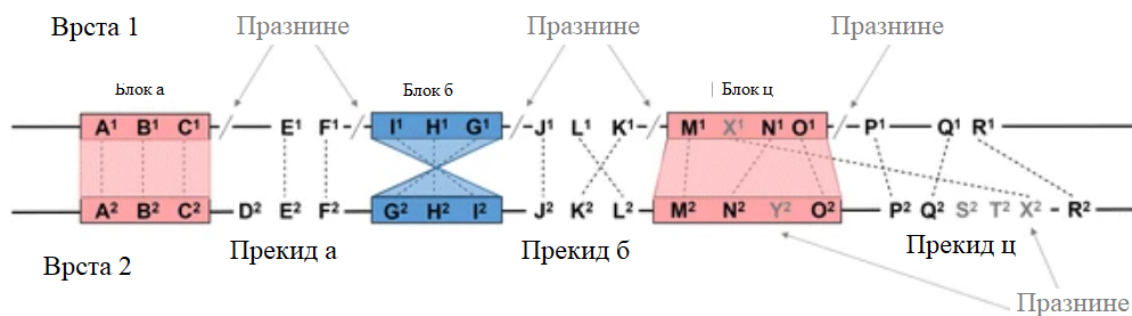
Људи више од половине генетског материјала деле са пилићима, воћним мушицама и бананама [7]. То је резултат секвенционирања људског генома, које је открило да човек дели заједничку ДНК са многим другим живим организмима. Ипак, тај проценат заједничког генетског материјала није довољан за закључак да сви наведени организми имају истог претка. Процентат подударности са шимпанзама од 96% - 99% у зависности од извора, или око 90% са мишевима, псима, слоновима и говедима [13], већ указује на заједничко порекло. Процент од 90% делује веома високо за толико велике разлике између наведених сисара, али, све постаје јасније уз чињеницу да сви људи имају 99,9% идентичан генетски материјал и да све разлике међу њима стају у 0,01% секвенци ДНК [4].

2.2.1 Блокови синтеније и прекидне тачке

Региони хромозома који деле исте секвенце гена потеклих од заједничког претка, поређаних на хромозому по одређеном редоследу идентичном у геномима других врста, називају се блоковима синтеније. Преуређивање блокова синтеније представља срж еволутивних промена и циљ њиховог проучавања је да се дође до одговора на питање кроз какве све промене пролази геном током еволуције.

У току еволуције, нису сви хромозомски региони сачували првобитан распоред гена. Региони хромозома чији се распоред и садржај гена не подудара

са регионима код других врста, налазе се између блокова синтеније и називају се прекидне тачке.



Слика 2.3: Блокови синтеније и прекиди [15]

На слици 2.3 су приказани блокови синтеније и прекидне тачке. Гени који се налазе на хромозомима две различите врсте су означени словима и бројевима који представљају врсту којој припадају (врста 1 или 2). Ортологи гени су повезани испрекиданим линијама, а гени који искачу из шаблона се третирају као прекиди.

На примеру три блока синтеније, уочава се да блок синтеније може бити запис истог редоследа гена (блок а), запис са обрнутим редоследом гена (блок б) или запис који дозвољава неке празнине (блок ц). Насупрот томе, случајеви изазивања прекида синтеније могу бити недостатак ортологих гена (прекид а), поремећен редослед гена (прекид б) или празнине изазване, на пример, великим бројем прескочених гена (прекид ц).

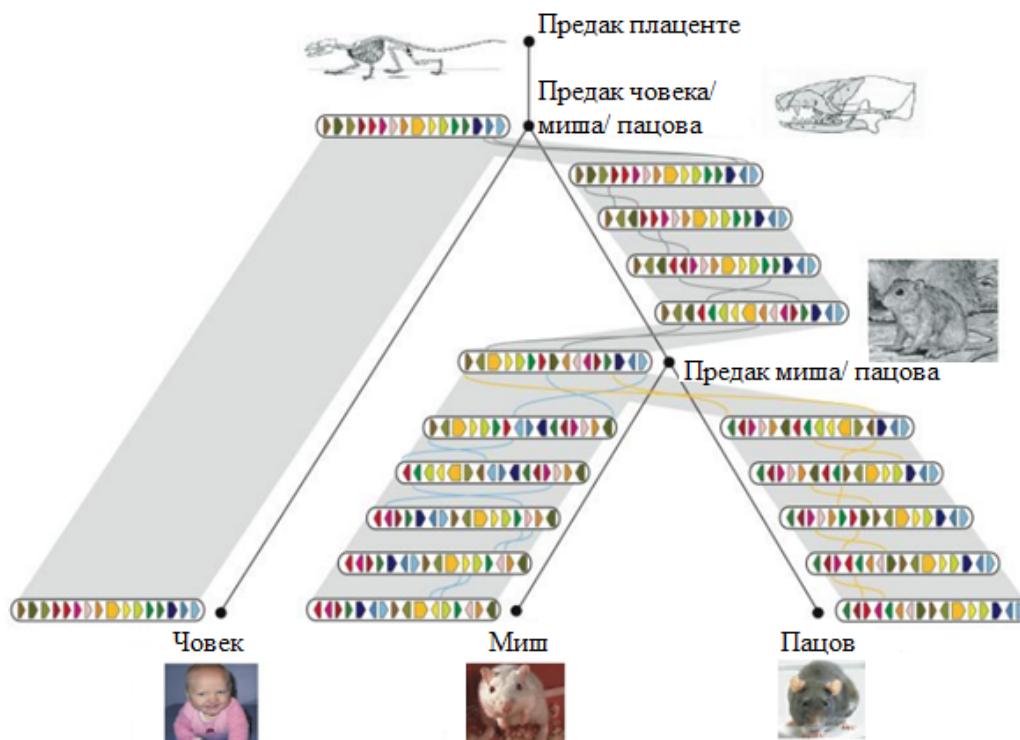
Глава 3

Трансформација генома миша у геном човека

Када би било могуће направити породично стабло миша и човека уназад скоро 80 милиона година, на врху стабла би био исти предак. До тог закључка је довело поређење генома миша и човека и чињеница да више од 90% мишјих гена постоји у сличном облику у људском геному. Људи и мишеви, према наводима *Lawrence Livermore National Laboratory*, имају приближно исти број гена и то око 140000, од којих је проучавано око 4000. Скоро сваки ген који је до сада пронађен у једној врсти пронађен је у блиском облику у другој. Од приближно 4000 гена који су проучавани, мање од 10 се налази у једној врсти, али не и у другој [8]. То значи да је ток еволуције и мешање ДНК унутар хромозома, као и између њих, определио да неки организам од пре више десетина милиона година, постане човек или, наизглед драстично различит сисар, миш. Најсликовитије приказано, то би изгледало као слагање 280 коцкица, тј. блокова синтеније, у случају човека у 23 пара хромозома, а у случају слагања генома миша, у 20 парова хромозома. То значи да се еволутивни процеси заснивају на постепеном преуређивању генома, које, у довољно дугом временском периоду, доводи до функционално и визуелно драстичних разлика код потомака, у односу на заједничког претка. Доступност генома човека и миша утрла је пут за развој алгоритама за мапирање синтеније великих размера и боље разумевање еволутивних процеса.

3.1 Преуређивање кроз итерације

Поређење мишјег и људског хромозома је рађено на једном од два хромозома који одређују пол, хромозому X , јер је кроз еволуцију сачувао скоро све гене. На слици 3.1, приказана је историја хромозома X .



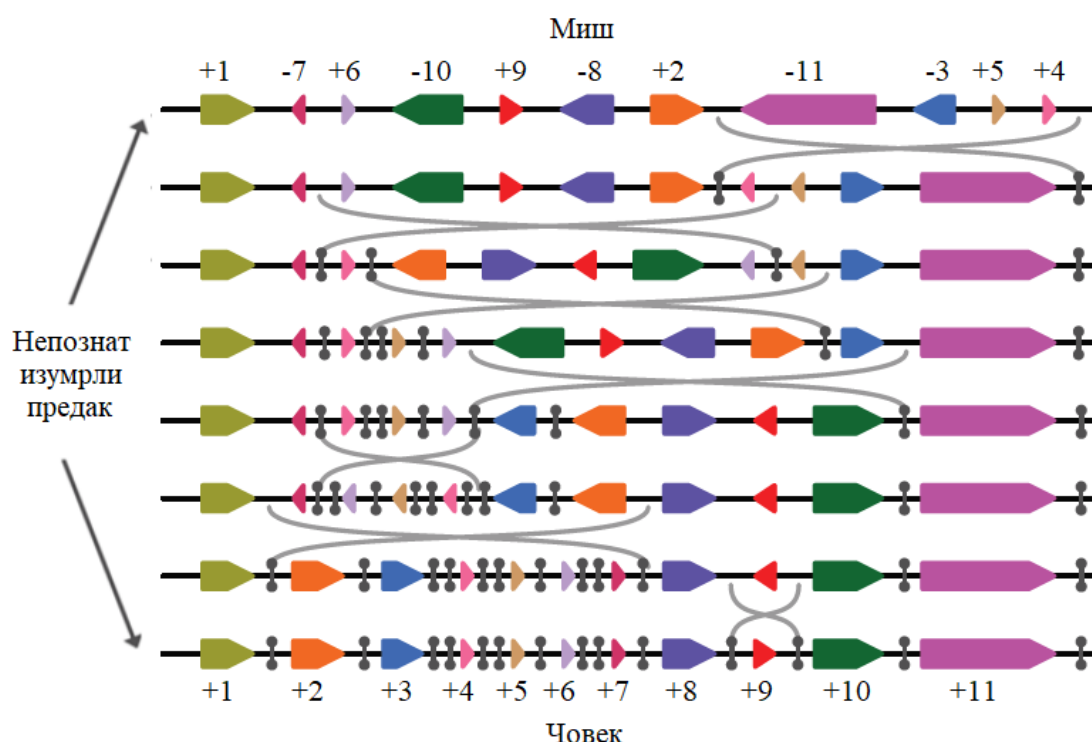
Слика 3.1: Еволуција X хромозома [21]

Једна од могућих трансформација мишјег X хромозома у људски X хромозом у седам итерација, приказана је на слици 3.2. Сваки блок синтеније је обојен и означен целим бројем од 1 до 11 и смером који је предзнак сваке целобројне ознаке (+ на десно, - на лево). Промена знака указује на то да су се нуклеотиди у том сегменту ДНК обрнули, па, на пример, $AAGTAG$ након обртања постаје $GATGAA$. Два кратка вертикална сегмента означавају крајње тачке преуређеног интервала у свакој итерацији и овај облик преуређивања се назива преокрет. Редослед блокова синтеније између тако означених тачака преуређивања се окреће у датој итерацији, док се обртање врши појединачно унутар сваког блока синтеније.

Када би претпостављено преуређивање у седам итерација било исправно, генетски материјал заједничког претка би се налазио у неком од средишњих

ГЛАВА 3. ТРАНСФОРМАЦИЈА ГЕНОМА МИША У ГЕНОМ ЧОВЕКА

итерација преуређивања, до којег би се миш враћао уназад кроз еволуциони процес, све до претка који је заједнички њему и човеку и од којег следеће трансформације представљају еволутивни процес у другом смеру, ка хромозому човека. Наравно, приказани пример је само један од 1070 могућих сценарија од седам итерација кроз који се X хромозом миша може трансформисати у људски X хромозом.



Слика 3.2: Итерације у једном од могућих сценарија трансформације мишјег у људски X хромозом [20]

Распоред блокова синтеније назива се пермутацијама, упркос томе што то математички нису пермутације. Пермутација у којој су блокови синтеније поређани по растућем редоследу и сви имају позитиван предзнак назива се идентичком пермутацијом. Она је у овом раду додељена као модел људском X хромозому, али то није правило. Могло је бити и обрнуто, и само је ствар избора онога ко моделује који геном ће изабрати за референтни и коме ће доделити идентичку пермутацију.

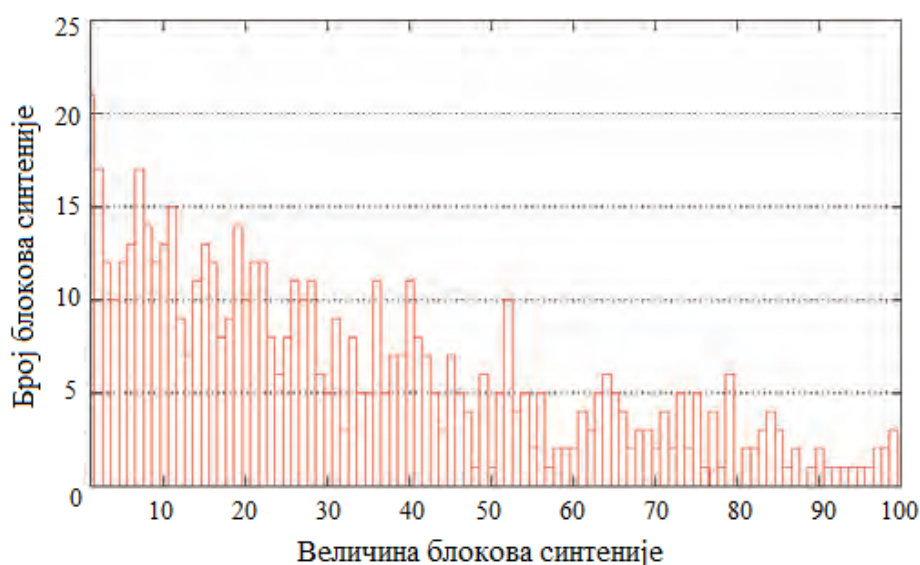
3.2 Места преуређивања

Познато је да се земљотреси чешће дешавају дуж и у близини линија раседа земљине коре, па се аналогно томе, може поставити питање да ли се можда и преуређења унутар генома дешавају чешће на одређеним местима у хромозому, односно, постоје ли у том смислу *вирусна* подручја унутар хромозома.

На слици 3.2, крајње тачке сваког преокрета су обележене вертикалним сегментима. Некада се крајње тачке преокрета гомилају уз исте блокове синтеније, као што се може уочити на региону иза зашиљене стране блока 3 на људском хромозому. Унутар региона су се преокрети кроз еволуцију догађали више пута што може указивати на то да је вероватноћа да овај регион поново буде крајња тачка преокрета већа него унутар неког другог региона. Стога овакве регионе унутар хромозома називамо *местима преуређивања*. С обзиром на то да ће прави сценарио преуређивања остати непознат, питање је да ли се уопште и може утврдити да ли постоје места преуређивања и да ли прекиди настају на случајним местима унутар хромозома или се по неком обрасцу дешавају унутар места преуређивања. Хипотеза да су прекиди распоређени на случајан начин унутар хромозома назива се *модел случајних региона преуређивања* и предложен је 1973. године. Ипак, ову хипотезу је било немогуће доказати управо услед недостатка тачних сценарија о преуређењу генома, насупрот претпостављеним који су били на располагању.

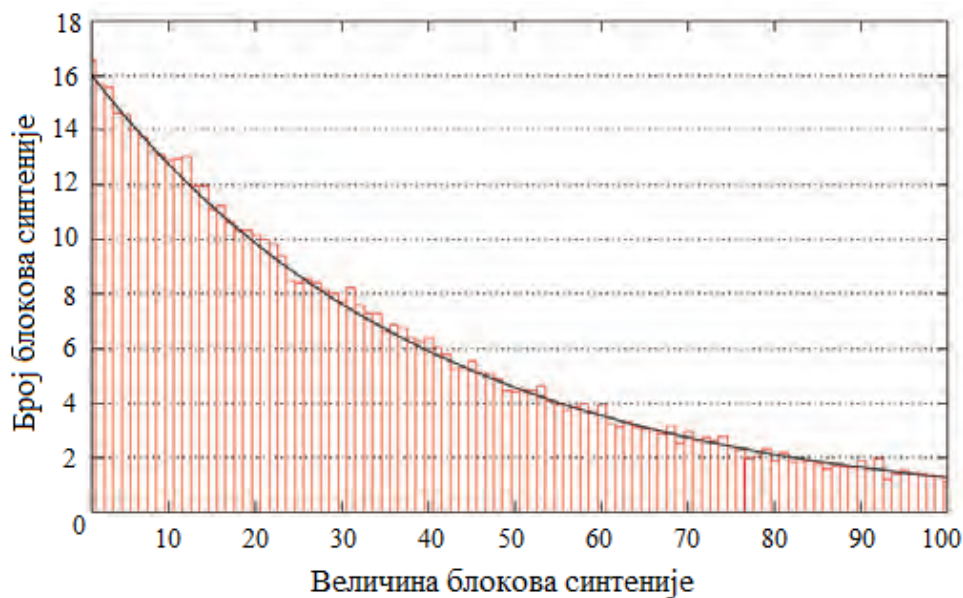
Поставља се питање да ли различита дужина блокова синтеније указује на тачност овог модела? На пример, дужине блокова синтеније човека и миша варирају у великој мери, при чему највећи блок (блок 11 на слици 3.2) заузима скоро 25% дужине X хромозома. Ако се искључи мало вероватан догађај да два случајна преокрета пресеку хромозом у потпуно истој позицији, тада n случајних преокрета пресеца хромозом на $2n$ локација и произведе $2n + 1$ блокова синтеније.

Слика 3.3 приказује резултат рачунарског експеримента у коме се 320 случајних преокрета примењује на симулирани хромозом који се састоји од 25000 гена, који производе $2 \cdot 320 + 1 = 641$ блок синтеније. Просечна величина блока синтеније је $25000/641$, што је приближно 34 гена, али то не значи да сви блокови синтеније треба да имају око 34 гена. Избором случајних локација за прекидне тачке, неки блокови могу имати само неколико гена, док

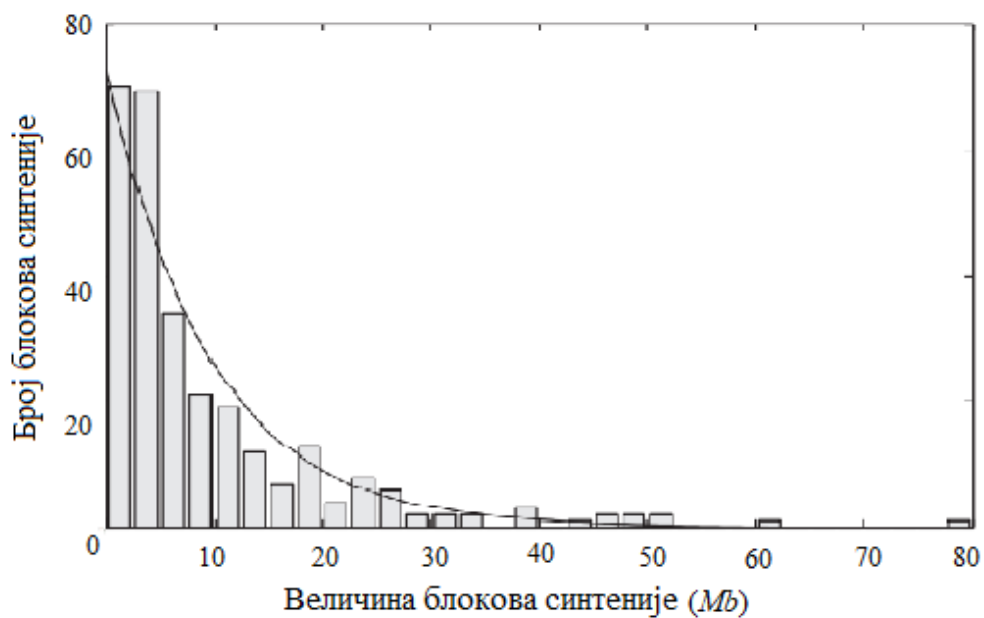


Слика 3.3: Симулација тачака ломова [20]

други блокови могу садржати преко сто. Слика 3.4 даје просечне резултате 100 симулација случајних прекида и илуструје да дистрибуција дужина блокова синтеније може да апроксимира криву која одговара експоненцијалној расподелу. Експоненцијална дистрибуција предвиђа да ће постојати око седам блокова који имају по 34 гена и један или два много већа блока са по 100 гена. С обзиром на то да хистограми дужине блокова синтеније израђени на основу реалних генетских података који се односе на миша и човека такође прате експоненцијалну расподелу и приказани су наслици 3.5, потврђено је да еволутивно преуређивање генома прати модел случајних региона преуређивања.



Слика 3.4: Дистрибуција дужине блокова синтеније [20]



Слика 3.5: Дистрибуција дужине блокова синтеније човека и миша [20]

Глава 4

Преуређивање генома

Преуређивање генома се може обављати на више начина:

1. преокретима
2. премештањем (транслокацијом)
3. физијом (цепањем)
4. фузијом (спајањем)

4.1 Сортирање помоћу преокрета

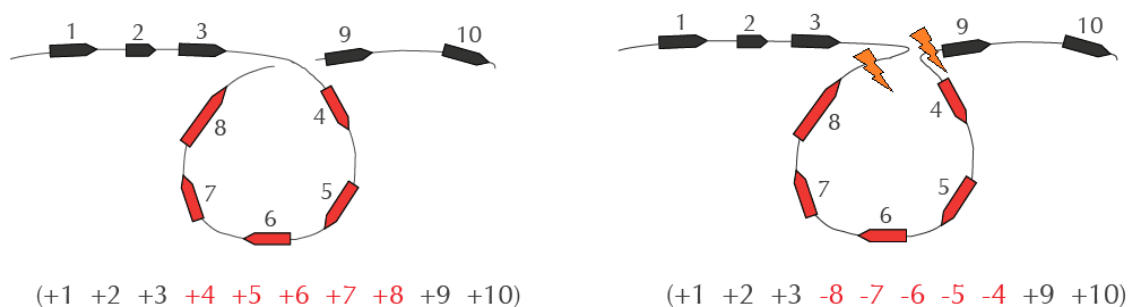
Под претпоставком да се еволуционе промене дешавају према најкраћем сценарију, тј. са најмањим могућим бројем промена између предака и потомака, треба пронаћи минимални број преокрета у складу са принципом тзв. Окамове општрице. Еволуција се можда неће увек кретати најкраћим путевима, али чак и када то не чини, број корака у правом еволуционом сценарију се често приближава броју корака у најштедљивијем сценарију. Студије преуређивања генома обично занемарују дужину блокова синтеније и представљају блокове синтеније означеним пермутацијама. Сваки блок је означен бројем, коме се додељује знак $+$ или $-$ у зависности од оријентације блока. Број елемената у пермутацији одређује њену дужину. Као што је приказано на слици 3.2, људски и мишји X хромозоми могу бити представљени следећим означеним пермутацијама дужине 11:

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

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

Низ корака којим се сегмент генома једног организма (у приказаном примеру миша) своди на сегмент генома другог организма (у приказаном примеру човека којем је додељена идентичка пермутација) назива се сортирање пермутација.

Битно је нагласити да су блокови синтеније јединствени и да понављање бројева, са или без знака, није дозвољено. Моделовање преокрета се заснива на обртању елемената, тј. блокова синтеније унутар интервала пермутације, након чега се мења предзнак елементима унутар обрнутог интервала. На примеру разбијања пермутације између блокова +3 и +4 као и између блокова +8 и +9, то подразумева одсецање дела између вертикалних сегмената (+1 +2 +3 | +4 +5 +6 +7 +8 | +9 +10), затим се интервал окреће и мења се знак елементима унутар њега (+1 +2 +3 | -8 -7 -6 -5 -4 | +9 +10) и, на крају, лепи уз сегменте који су остали непромењени (+1 +2 +3 -8 -7 -6 -5 -4 +9 +10), као што је приказано на слици 4.1.



Слика 4.1: Преокрети са прекидиним тачкама [20]

4.2 Похлепна хеуристика

Похлепна хеуристика за сортирање помоћу преокрета представља један од начина за проналажење најкраћег низа преокрета који претвара једну пермутацију у другу. Она доноси похлепне, локално оптималне изборе у сваком кораку, надајући се да ће на тај начин постићи и глобално оптимално решење.

Пример похлепног сортирања је приказан на слици 4.2. Преуређивање се врши постављањем сваког елемента редом на њихове исправне позиције. Интервал који ће бити преокренут је приказан црвеном бојом, а елементи који су постављени у исправан положај су приказани плавом бојом. Види се да

је блок синтеније 1 од старта на правом месту па је означен плавом бојом. Блок синтеније 2 долази на своје место у трећој итерацији, блок синтеније 3 у четвртој и тако редом.

1	(+1	-7	+6	-10	+9	-8	+2	-11	-3	+5	+4)
2	(+1	-2	+8	-9	+10	-6	+7	-11	-3	+5	+4)
3	(+1	+2	+8	-9	+10	-6	+7	-11	-3	+5	+4)
4	(+1	+2	+3	+11	-7	+6	-10	+9	-8	+5	+4)
5	(+1	+2	+3	-4	-5	+8	-9	+10	-6	+7	-11)
6	(+1	+2	+3	+4	-5	+8	-9	+10	-6	+7	-11)
7	(+1	+2	+3	+4	+5	+8	-9	+10	-6	+7	-11)
8	(+1	+2	+3	+4	+5	+6	-10	+9	-8	+7	-11)
9	(+1	+2	+3	+4	+5	+6	-7	+8	-9	+10	-11)
10	(+1	+2	+3	+4	+5	+6	+7	+8	-9	+10	-11)
11	(+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	-11)
12	(+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11)

Слика 4.2: Имплементација похлепне хеуристике у електронској лекцији [20]

Математички изражено, елемент k у пермутацији $P = (p_1 \dots p_n)$ је сортиран ако је $p_k = +k$, у противном је несортиран. Пермутација P је k -сортирана ако је њених првих $k - 1$ елемената сортирано и ако је елемент k несортиран. За сваку $(k - 1)$ -сортирану пермутацију P , постоји један преокрет, назван k -сортирајући преокрет, који фиксира првих $k - 1$ елемената пермутације и помера елемент k на исправну, k -ту позицију. У случају када је елемент k већ на k -тој позицији пермутације P , али није добро оријентисан, k -сортирањем се само окреће k -ти елемент (пример блока 11 у последњем кораку).

У случају преуређивања мишјег X хромозома, на овај начин се до идентичке пермутације долази у 11 корака, што не представља оптималну дужину, јер је сортирањем помоћу преокрета преуређивање краће за чак 4 корака, што се види са слике 3.2. Тако је на овом примеру доказано да похлепно сортирање не резултира најмањим бројем преокрета.

Алгоритам имплементиран на програмском језику *Typescript* којим се израчунава најкраћи низ преокрета између две пермутације исте дужине, приказан је у листингу 4.1.

```

1 greedySortingByReversals(input: { P: number[]; verbose: boolean }) {
2   let { P, verbose } = input;
3   let permutationDistance: number = 0;
4   const permutations: number[][] = [...P];
5   const lefts: number[][] = [];
6   const mids: number[][] = [];
7   let p, left, mid;
8
9   const n: number = P.length;
10
11  for (let k = 0; k < n; k++) {
12    if (P[k] !== k + 1) {
13      // Доводимо елемент k на k-ту позицију
14      [p, left, mid] = applySortingReversals(P, k);
15      P = p;
16      permutationDistance++;
17      if (verbose) {
18        permutations.push([...P]);
19        lefts.push(left);
20        mids.push(mid);
21      }
22    }
23    // Додатно обрћање се врши уколико је елемент погрешно усмерен
24    if (P[k] === -(k + 1)) {
25      P[k] = -P[k];
26      permutationDistance++;
27      if (verbose) {
28        permutations.push([...P]);
29        lefts.push(left);
30        mids.push([- (k + 1)]);
31      }
32    }
33  }
34  lefts.push(Array.from({ length: P.length }, (_, i) => i + 1));
35  return { permutationDistance, permutations, lefts, mids };
36 }

```

Листинг 4.1: Похлепни алгоритам

Помоћна функција којом се одређује следећи елемент за премештање, приказана је у листингу 4.2.

```

1 export const applySortingReversals = (
2   P: number[],
3   k: number,
4 ): [number[], number[], number[]] => {
5   const n: number = P.length;
6
7   for (let i = k; i < n; i++) {
8     // Поштрага за елементом који треба да дође на k-ту позицију
9     if (Math.abs(P[i]) === k + 1) {
10      // Издвајамо већ сортиран леви део
11      // Од сортираног дела узимамо све елементе до оног којег треба
12      // преместити на праву позицију (средњи део)
13      // остатак (десни део)
14      const Left: number[] = P.slice(0, k);
15      const Mid: number[] = P.slice(k, i + 1);
16      const Right: number[] = P.slice(i + 1);
17
18      // На леви део се леви средњи (извршено обрћање) и
19      // на њих остатак шј. десни део
20      return [Left.concat(Mid.map((x) => -x).reverse(), Right), Left, Mid];
21    }
22  }
23  return [P, [], []];
24 };

```

Листинг 4.2: Помоћна функција за постављање елемента на исправну позицију

У електронској лекцији у делу предвиђеном за имплементацију алгорита, као први улазни параметар се уноси дужина пермутације, након чега се појављује првим параметром дефинисан број поља за унос саме пермутације. Кликком на дугме *Израчунај растојање*, покреће се алгоритам за израчунавање најкраћег растојања за трансформацију унете пермутације у идентичку. Као излаз, осим броја корака у процесу сортирања, излистане су и пермутације настале након сваког од корака. Пример примене алгорита за похлепно сортирање у електронској лекцији је приказан на слици 4.3.

Pohlepna heuristika

Unesite dužinu permutacije

Unesite permutaciju

Izračunaj rastojanje

Rastojanje je: 6

Preokreti:

1.	2	-5	1	-3	-4
2.	-1	5	-2	-3	-4
3.	1	5	-2	-3	-4
4.	1	2	-5	-3	-4
5.	1	2	3	5	-4
6.	1	2	3	4	-5
7.	1	2	3	4	5

Слика 4.3: Пример примене алгоритма за похлепно сортирање у електронској лекцији

4.3 Прекидне тачке

Помоћу слике 4.4 се може учити како се сваком следећом пермутацијом почетна приближава идентичкој пермутацији. У првом преокрету секције $(-12 -8 -7 -6 +1 +2 +10 +9 -11)$ на десној крајњој тачки уместо првобитна два узастопна елемента $(-11 +13)$, добијају се пожељни елементи $(+12 +13)$. У четвртом преокрету $(+11 -9)$ добијају се узастопни елементи $(-11 -10)$ који простим обраћањем формирају интервал са пожељним узастопним елементима $(+10 +11)$.

	Прекиди (P)
$ +3 +4 +5 -12 -8 -7 -6 +1 +2 +10 +9 -11 +13$	8
$ +3 +4 +5 +11 -9 -10 -2 -1 +6 +7 +8 +12 +13$	7
$+1 +2 +10 +9 -11 -5 -4 -3 +6 +7 +8 +12 +13$	6
$+1 +2 +3 +4 +5 +11 -9 -10 +6 +7 +8 +12 +13$	5
$+1 +2 +3 +4 +5 +9 -11 -10 +6 +7 +8 +12 +13$	4
$+1 +2 +3 +4 +5 +9 -8 -7 -6 +10 +11 +12 +13$	3
$+1 +2 +3 +4 +5 +6 +7 +8 -9 +10 +11 +12 +13$	2
$+1 +2 +3 +4 +5 +6 +7 +8 +9 +10 +11 +12 +13$	0

Слика 4.4: Сортирање помоћу преокрета [20]

Наслућује се да је формирање секција са узастопним елементима као што су $(+12 +13)$ или $(-11 -10)$ пожељно. За наведене парове је заједничко то што је други елемент за 1 већи од првог, па се за узастопне елементе $(p_i p_{i+1})$ у пермутацији $P = (p_1 \dots p_n)$ каже да формирају суседност ако важи да је $p_{i+1} - p_i = 1$. По дефиницији, за било који позитиван цео број $k < n$, суседи су $(k k + 1)$, исто као и $(-(k + 1) - k)$. Ако $p_{i+1} - p_i$ није једнако 1, онда је $(p_i p_{i+1})$ прекидна тачка.

Једноставније речено, прекидна тачка настаје између парова узастопних елемената који су „ван реда“ у поређењу са идентичком пермутацијом $(+1 + 2 \dots + n)$. На пример, $(+2 +10)$, $(+9 -11)$ и $(-11 +13)$ су очигледно прекидне тачке јер нису суседи. Али, и пар $(+10 +9)$ је такође прекидна тачка, упркос томе што садржи узастопне бројеве. Разлог је то што су његови предзнаци ван реда у поређењу са идентичком пермутацијом и јер је $p_{i+1} - p_i = -1$. Додавањем елемената на почетак (нула) и крај пермутације $(n + 1)$, добија се пермутација $(0 p_1 \dots p_n (n + 1))$, која има $n + 1$ парова узастопних елемената:

$$(0 p_1), (p_1 p_2), (p_2 p_3) \dots (p_{n-1} p_n), (p_n (n + 1)).$$

4.3.1 Одређивање броја прекидних тачака

Пошто је објашњено да сваки пар узастопних елемената формира прекидну тачку или представља суседне блокове, за сваку пермутацију P , дужине n , важи:

$$\text{Суседи}(P) + \text{Прекиди}(P) = n + 1$$

Једина пермутација код које су сви узастопни елементи уједно и суседи, је идентичка пермутација. Најмањи број промена потребних за трансформацију у идентичку пермутацију представља растојање пермутације.

Сортирањем помоћу преокрета се, у ствари, елиминишу прекидне тачке. На слици 4.4 се број прекидних тачака смањује са 8 на 0, јер пермутација после сваког преокрета постаје уређенија и ближа идентичкој пермутацији, па је самим тим, преостали број прекидних тачака све мањи. Преокретањем интервала, прекидне тачке унутар интервала увек остају прекидне тачке, док се прекидне тачке на границама интервала могу елиминисати, смањујући укупни број истих. На пример, постоји пет прекидних тачака унутар распона следећег преокрета пермутације:

$$(0 + 3 + 4 + 5 - 12 - 8 - 7 - 6 + 1 + 2 + 10 + 9 - 11 + 13 + 14 \ 15)$$

и то су:

$$(-12 - 8), (-6 + 1), (+2 + 10), (+10 + 9), (+9 - 11)$$

Након преокрета овог интервала, наведене прекидне тачке постају нове прекидне тачке:

$$(+11 - 9), (-9 - 10), (-10 - 2), (-1 + 6), (+8 + 12)$$

Дакле, једине прекидне тачке које се могу елиминисати преокретом су две прекидне тачке које се налазе на границама обрнутог интервала. То значи да једним обртањем можемо елиминисати максимално две прекидне тачке. Овај закључак нас доводи до Теореме о прекидној тачки.

Теорема 1 (Теорема о прекидној тачки). *Растојање преокрета, \bar{w}_j , број обртања није мањи од половине броја прекидних тачака:*

$$\text{BrojObrtanja}(P) \geq \text{BrojPrekidnihTacka}(P)/2$$

□

Са друге стране, максималан број обртања које доводи до идентичке пермутације је $n + 1$ и то у случају обрнуте пермутације:

$$(+n + (n - 1) \dots + 1)$$

Листинг 4.3 приказује алгоритам за одређивање броја прекидних тачака, док је пример примене алгоритма у електронској лекцији приказан на слици 4.5. Улазни параметри су дужина пермутације и пермутација $(+2 +3 +1 +5 -4)$, а резултат је број прекидних тачака који у задатом примеру износи 5.

Одређивање броја прекидних тачака

Unesite dužinu permutacije

5

Unesite permutaciju

2

3

1

5

-4

**Izračunaj broj prekidnih
tačaka**

Број прекидних тачака је: 5

| 2 3 | 1 | 5 | -4 |

Слика 4.5: Пример примене алгоритма за одређивање броја прекидних тачака у електронској лекцији

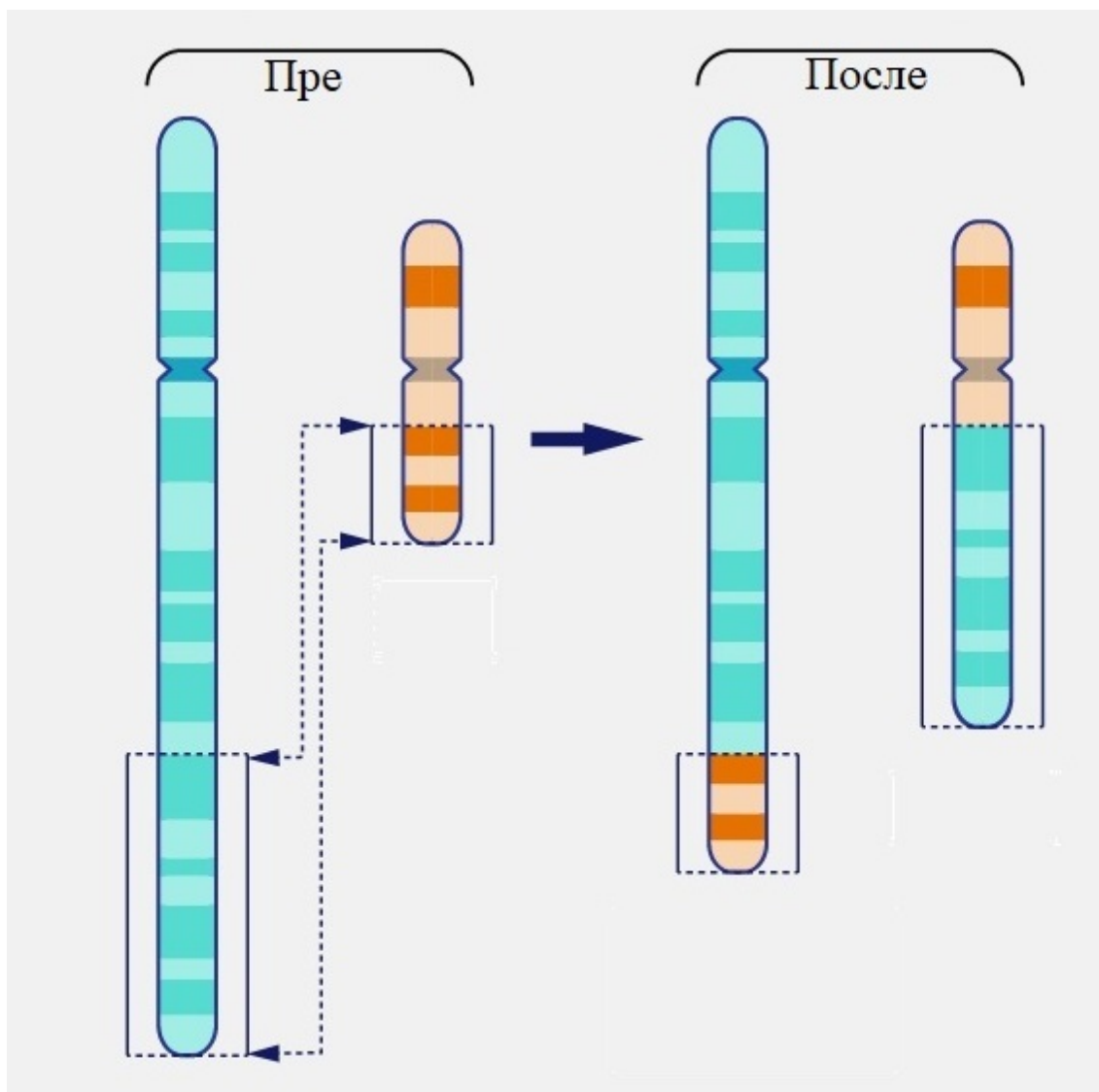
```
1 numberOfBreakpoints(input: { P: number[] }): [number, number[]] {
2   const { P } = input;
3   const length = P.length;
4   let adjacencies = 0;
5   const indexes = new Array(length + 1).fill(1);
6
7   // У прекиде не рачунамо почешњи елементи ако се он налази на
8   // добром месту
9   if (P[0] === 1) {
10    indexes[0] = 0;
11  }
12
13  for (let i = 0; i < length - 1; i++) {
14    // суседи су уколико је разлика следеће и шренућној једнака 1
15    if (P[i + 1] - P[i] === 1) {
16      adjacencies++;
17      indexes[i + 1] = 0;
18    }
19  }
20
21  // У прекиде не рачунамо последњи елементи уколико се он налази на
22  // добром месту
23  if (P[length - 1] === length) {
24    indexes[length] = 0;
25  }
26
27  // у пермутацији идентитета број суседа је једнак n + 1.
28  // Дакле број шачака прекида ће бити n + 1 - бројСуседа
29  const breakpoints = length + 1 - adjacencies;
30  return [breakpoints, indexes];
31 }
```

Листинг 4.3: Број прекидних тачака

4.4 Транслокација, фузија и физија

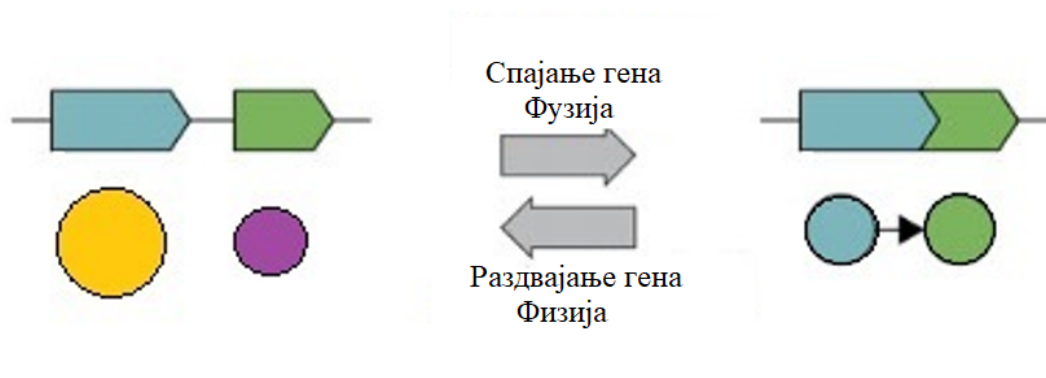
Преуређивање генома унутар једног хромозома преокретима и премештањима скупова гена, нису једини начини. Код мултихромозомалних генома преуређивање се може десити и на следеће начине: транслокацијом, фузијом и физијом.

Транслокација представља размењивање сегмената различитих хромозома. На пример, геном (+1 +2 +3 +4 +5 +6)(+7 +8 +9 +10 +11) се састоји од



Слика 4.6: Транслокација генома [9]

два хромозома (+1 +2 +3 +4 +5 +6) и (+7 +8 +9 +10 +11). Транслокацијом сегмената ова два хромозома, могу се добити два нова хромозома (+1 +2 +3 +4 +9 +10 +11) и (+7 +8 +5 +6), на пример. Значи да је транслокација генетска промена у којој се део једног хромозома одваја и везује за други хромозом.



Слика 4.7: Фузија и физија генома

Осим транслокацијом, преуређивање генома може се извршити фузијом и физијом. Већ споменути два хромозома (+1 +2 +3 +4 +5 +6) и (+7 +8 +9 +10 +11), могу се даљим процесима спојити (фузија) у један хромозом (+1 +2 +3 +4 +5 +6 +7 +8 +9 +10 +11), који би се онда раздвојио (физија) у два потпуно нова хромозома у односу на почетне: (+1 +2 +3 +4) и (+5 +6 +7 +8 +9).

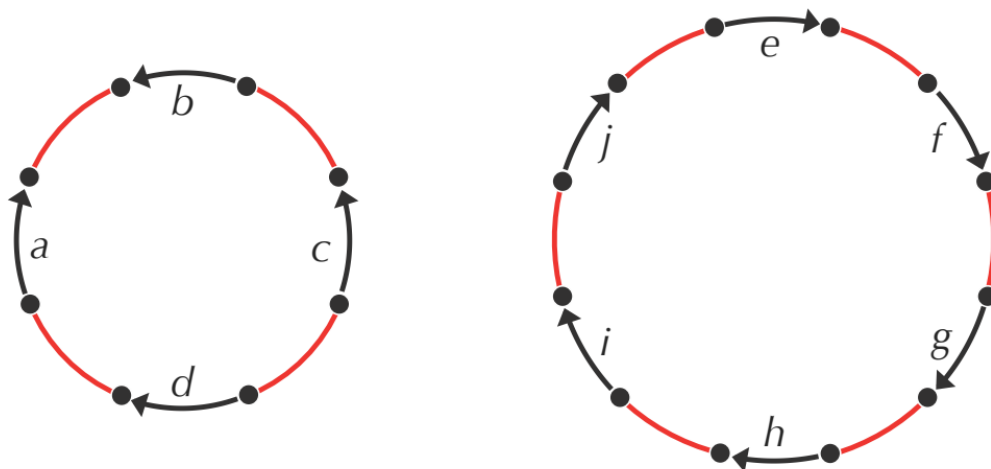
4.5 Графовска репрезентација генома

Биолошки гледано, геноми су линеарне структуре, али су зарад лакше анализе, у наставку представљени као циркуларне структуре. Сваки блок синтеније је представљен црном усмереном граном. Сваки блок синтеније тј. свака усмерена црна грана је повезана са другом усмереном црном граном преко неусмерене црвене гране, као што је приказано на слици 4.8.

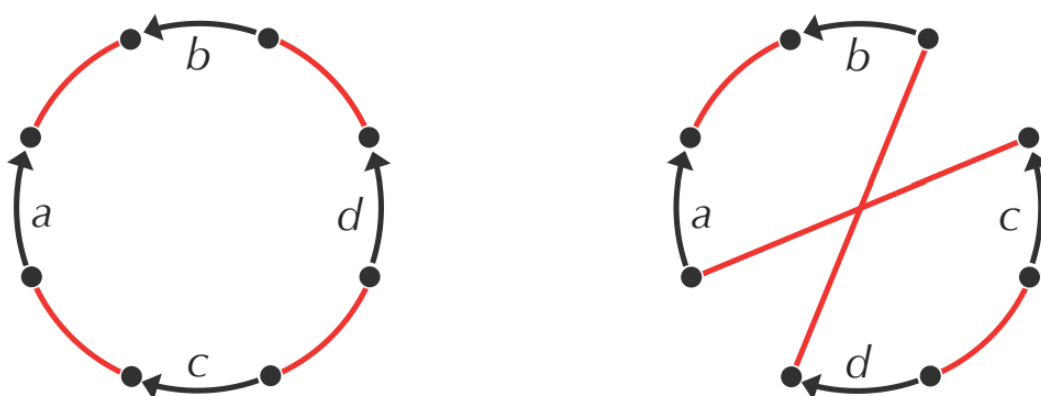
Хромозоми од n блокова синтеније могу бити представљени циркуларно на $2n$ различитих начина. На пример, леви хромозом мултихромозоналног генома са слике, може се написати на следеће начине: $(+a -b -c +d)$, $(-b -c +d +a)$, $(-c +d +a -b)$, $(+d +a -b -c)$, $(-a -d +c +b)$, $(-d +c +b -a)$, $(+c +b -a -d)$, и $(+b -a -d +c)$, све у зависности од тога одакле се започне циклус и у ком смеру се иде.

На слици 4.9 приказана су два еквивалентна приказа истог хромозома, тј хромозома $Q = (+a -b -d +c)$, насталог трансформацијом кружног хромозома $P = (+a -b -c +d)$.

Хромозом Q се може визуализовати на различите начине, у зависности



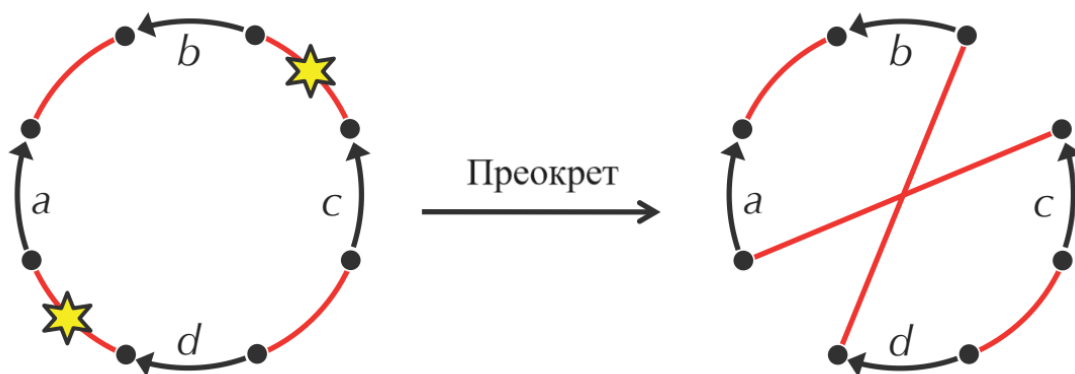
Слика 4.8: Кружни хромозоми [20]



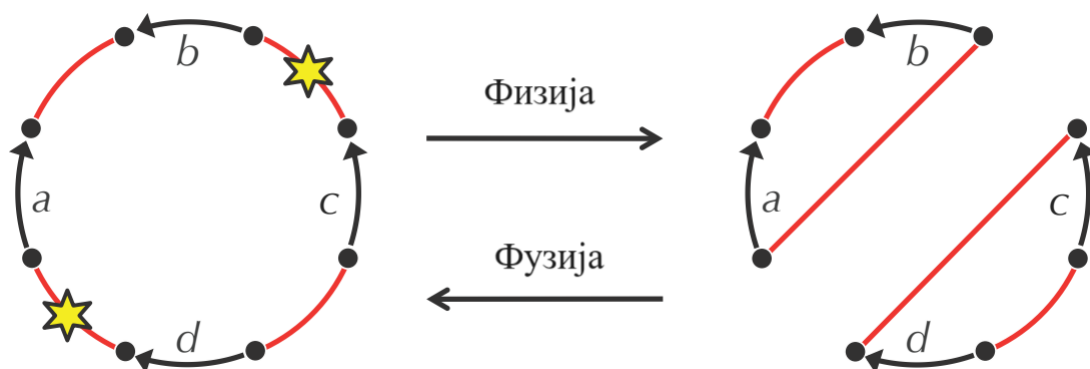
Слика 4.9: Трансформација кружних хромозома [20]

од распореда црних грана. Иако је први приказ природнији, анализираће се други приказ, зато што су црне гране распоређене истим редоследом као у приказу хромозома $P = (+a - b - c + d)$. Као што је приказано на слици 4.10, овакав приступ омогућава лакше визуализацију трансформације. Она подразумева брисање (или прекидање) две црвене гране у хромозому P које повезују b са d и c са a , што представља тзв. 2-прекид.

На слици 4.11 је илустрована физија хромозома $P = (+a - b - c + d)$ у два мања хромозома $Q = (+a - b)(-c + d)$. Обрнути процес ове операције, то јест фузија два хромозома Q како би се поновно формирао хромозом P , такође се може представити преко концепта 2-прекида. Операције фузије и физије, слично као и у претходној трансформацији, укључују брисање две гране у једном геному и њихову замену са две нове у другом.



Слика 4.10: Преокрет [20]

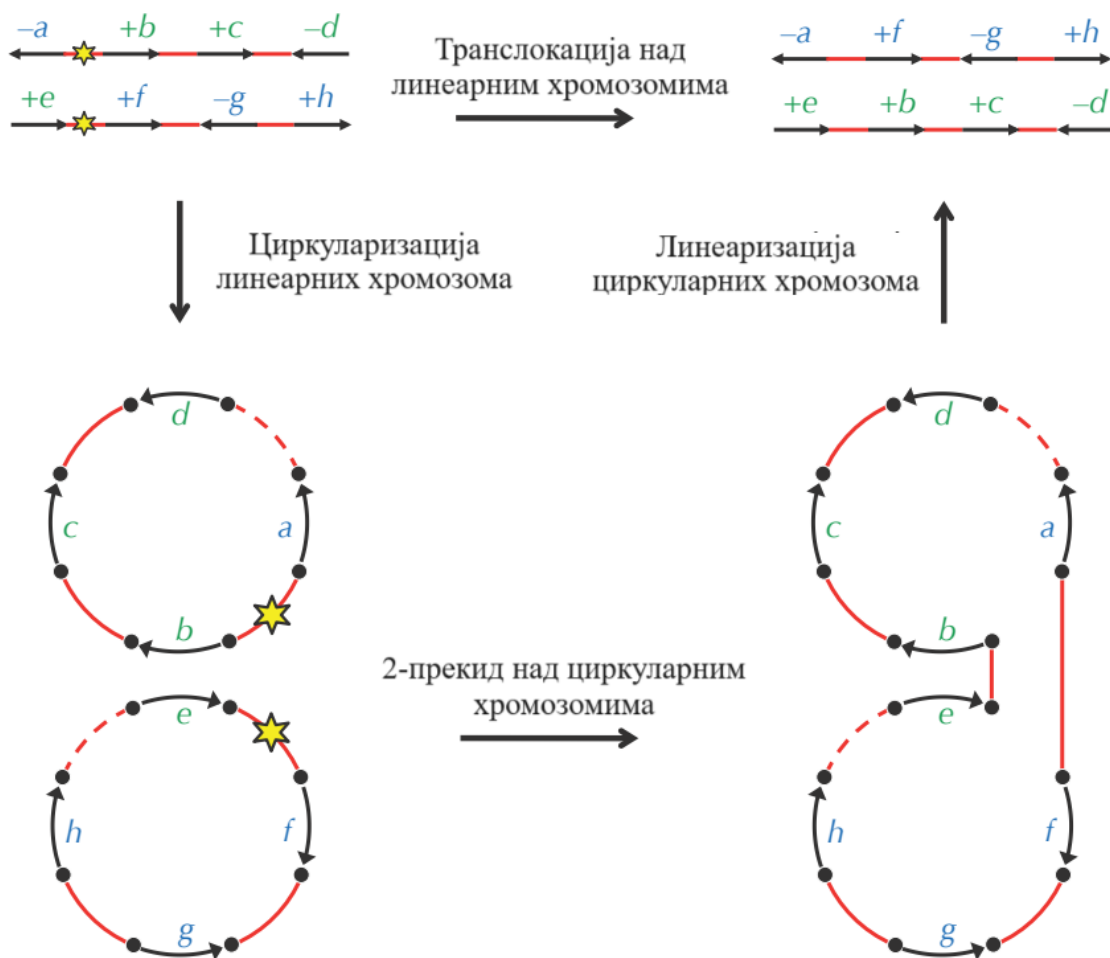


Слика 4.11: Физија и фузија два хромозома [20]

Операција транслокације се такође може представити и на циркуларној репрезентацији хромозома заменом две црвене ивице са две различите црвене ивице, као што је приказано на слици 4.12.

На основу приказаног примера, може се закључити да се све поменуте операције над линеарним хромозомима могу представити у циркуларној форми и то као прекидање две црвене гране и њихова замена са две нове црвене гране над истим чворовима. Ова операција на геномском графу се назива 2-прекид. Број 2-прекида којим се извршава трансформација генома P у Q , назива се растојање 2-прекида између P и Q и означава се са $d(P, Q)$. У складу са тенденцијом да се увек оптимизује процес, потребно је наћи најмање растојање 2-прекида, тј. најмањи број 2-прекида за трансформисање једног генома у други.

У листингу 4.4 је приказан алгоритам за креирање циркуларне репрезента-



Слика 4.12: Циркуларна транслокација линеарног хромозома [20]

ције хромозома, док су алгоритми за претварање циркуларне репрезентације у линеарну репрезентацију приказани у листингу 4.5 и листингу 4.6.


```

1 export const chromosomeToCycle = (chromosome: number[]): number[] => {
2   const n: number = chromosome.length;
3   const cycle: number[] = new Array(2 * n).fill(0);
4
5   // Креирамо цикличну репрезентацију
6   // [1, -2, -3, 4] постаје [1 2 4 3 6 5 7 8]
7   for (let i = 0; i < n; i++) {
8     const j: number = chromosome[i];
9
10    if (j > 0) {
11      cycle[2 * i] = 2 * j - 1;
12      cycle[2 * i + 1] = 2 * j;
13    } else {
14      cycle[2 * i] = -2 * j;
15      cycle[2 * i + 1] = -2 * j - 1;
16    }
17  }
18
19  return cycle;
20 };

```

Листинг 4.4: Креирање циркуларне репрезентације хромозома

Хромозом се преводи у циркуларну репрезентацију тако што се сваки блок синтеније представља као уређени пар по формули за n : $(2n - 1, 2n)$, а за $-n$: $(2n, 2n - 1)$. Зато се у примеру приказаном у листингу 4.4, хромозом $(1, -2, -3, 4)$, трансформише у $(1, 2, 4, 3, 6, 5, 7, 8)$, јер 1 постаје $(1, 2)$, -2 постаје $(4, 3)$ и тако редом.

```

1 const cycleToChromosome = (
2   breakPointGraph: Record<number, number[]>,
3 ): number[][] => {
4   const P: number[][] = [];
5
6   //Формирамо циклусе из непрекидног графа
7   const cycles = getCycles(breakPointGraph);
8
9   //Од сваког циклуса формирамо линеарну репрезентацију хромозома
10  for (const cycle of cycles) {
11    const chromosome = convertToChromosome(cycle);
12    P.push(chromosome);
13  }
14
15  return P;
16 };

```

Листинг 4.5: Трансформација циркуларне репрезентације хромозома(прекидног графа) у линеарну репрезентацију

```

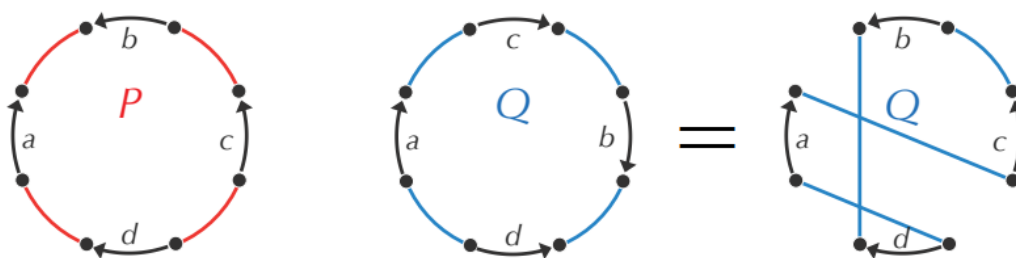
1 export const convertToChromosome = (cycleNodes: number[]): number[] => {
2   const m: number = cycleNodes.length;
3   const chromosome: number[] = new Array(m / 2).fill(0);
4
5   //[[1 2 4 3 6 5 7 8] њосњаје [1, -2, -3, 4]
6   for (let j = 0; j < m; j += 2) {
7     if (cycleNodes[j] < cycleNodes[j + 1]) {
8       chromosome[j / 2] = cycleNodes[j + 1] / 2;
9     } else {
10      chromosome[j / 2] = -cycleNodes[j] / 2;
11    }
12  }
13
14  return chromosome;
15 };

```

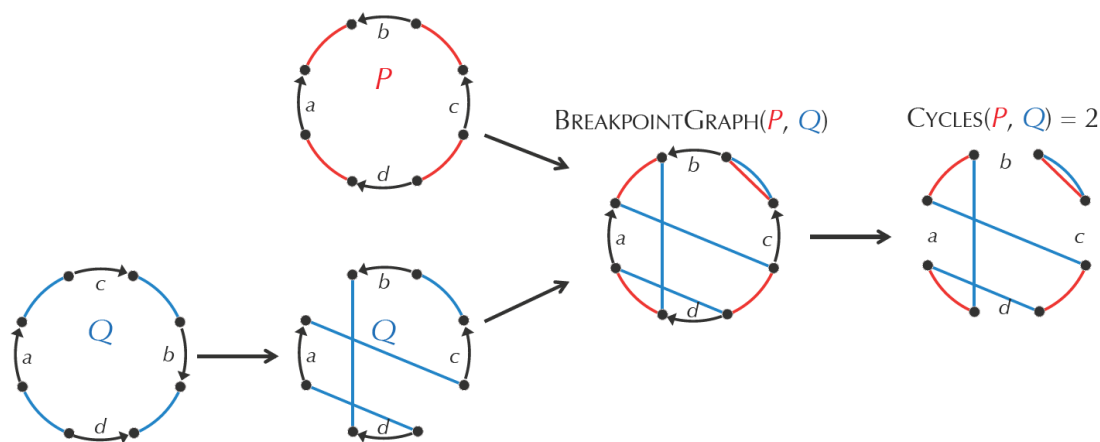
Листинг 4.6: Помоћна функција за трансформацију циркуларне репрезентације хромозома(прекидног графа) у линеарну репрезентацију

4.5.1 Прекидни граф

Да би се израчунало растојање 2-прекида, потребно је упоредити два генома. Унихромозомални геноми $P = (+a - b - c + d)$ и $Q = (+a + b - d + c)$ су приказани на слици 4.13, на којој се уочава да су неусмерене гране обојене различито у различитим геномима: у геному P су црвене, а у геному Q су плаве. Геном Q је приказан у изворном облику и у облику у којем су црне ивице које одговарају истим генима преуређене тако да буду распоређене тачно као у P . Уколико би се приказани графови преклопили у један, добио би се тзв. прекидни граф или $BreakPointGraph(P, Q)$, као на слици 4.14. У листингу 4.7 је приказан алгоритам за одређивање обојених грана, док је алгоритам за креирање прекидног графа приказан у листингу 4.8



Слика 4.13: Приказ генома P и Q [20]



Слика 4.14: Прекидни граф код унихромозомалних генома [20]

```

1 export const coloredEdges = (P: number[][]): [number, number][] => {
2   const edges: [number, number][] = [];
3   for (const chromosome of P) {
4     // Преварамо хромозом у цикличну репрезентацију
5     const cycleNodes: number[] = chromosomeToCycle(chromosome);
6     const m: number = cycleNodes.length;
7
8     // Неусмерена трана је она која спаја крај прве усмерене
9     // и почешак следеће усмерене тране, па зашто иешља почине од 1
10    for (let j = 1; j < m - 1; j += 2) {
11      edges.push([cycleNodes[j], cycleNodes[j + 1]]);
12    }
13    // Треба додати и преосшалу неусмерену трану која завара циклус
14    edges.push([cycleNodes[m - 1], cycleNodes[0]]);
15  }
16  return edges;
17 };

```

Листинг 4.7: Одређивање неусмерених (обојених) грана

```

1 export const initializeGraph = (
2   edges: [number, number][],
3 ): Record<number, number[]> => {
4   //Граф дефинишемо преко листе суседа
5   const adjacencyList: Record<number, number[]> = {};
6   //За сваку трану (u, v) дефинишемо да је u сусед од v и обрнуто
7   edges.forEach(([u, v]) => {
8     if (!adjacencyList[u]) {
9       adjacencyList[u] = [];
10    }
11    if (!adjacencyList[v]) {
12      adjacencyList[v] = [];
13    }
14    adjacencyList[u].push(v);
15    adjacencyList[v].push(u);
16  });
17  return adjacencyList;
18 };

```

Листинг 4.8: Формирање прекидног графа

У новоформираном графу, црвене и плаве гране формирају алтернирајуће црвено-плаве циклусе. Број ових циклуса у прекидном графу се означава са $cycles(P, Q)$. На слици 4.14 се уочава да је број циклуса у овом случају једнак 2. У листингу 4.9 је приказан алгоритам за одређивање циклуса у прекидном графу.

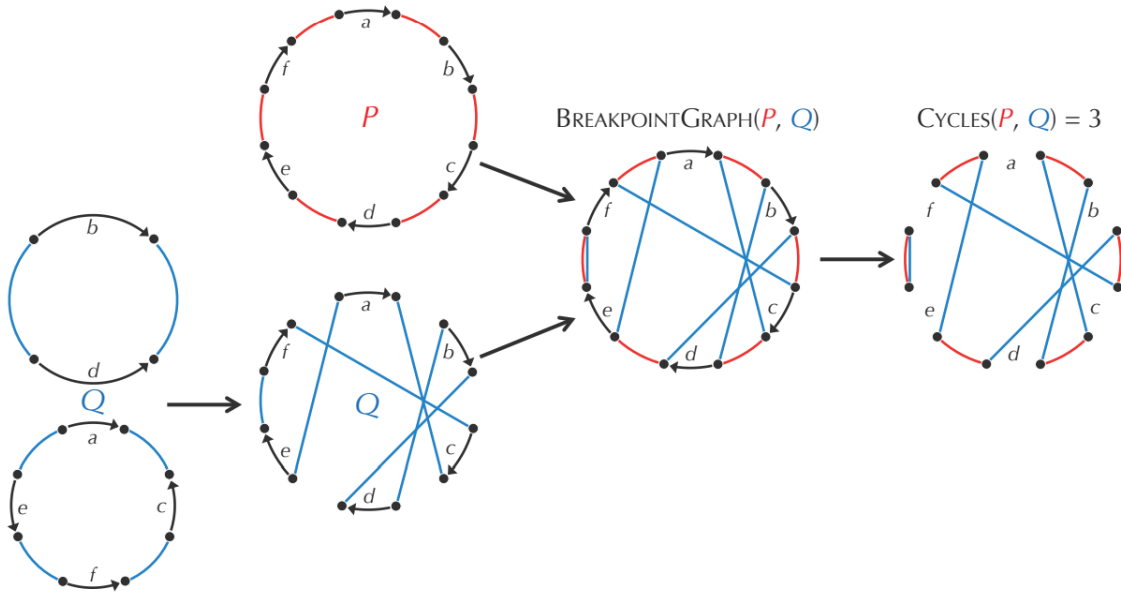
```

1 export const getCycles = (
2   breakPointGraph: Record<number, number []>,
3 ): number [] [] => {
4   const unvisited: number [] = Object.keys(breakPointGraph).map((key) =>
5     Number(key),
6   );
7   const cycles: number [] [] = [];
8   //Полазећи од првог чвора, пролазимо кроз суседе и тако формирамо циклус
9   while (unvisited.length > 0) {
10    let v = unvisited[0];
11    const currentCycle: number [] = [v];
12    unvisited.splice(unvisited.indexOf(v), 1);
13    while (true) {
14      let nextV: number | null = null;
15      for (const w of breakPointGraph[v]) {
16        // Уколико чвор није обиђен, настављамо пролазак кроз граф
17        if (unvisited.includes(w)) {
18          nextV = w;
19          break;
20        }
21      }
22      // Уколико нисмо нашли чвор који већ није посећен, затворили смо циклус
23      if (!nextV) {
24        break;
25      }
26      currentCycle.push(nextV);
27      unvisited.splice(unvisited.indexOf(nextV), 1);
28      v = nextV;
29    }
30    cycles.push(currentCycle);
31  }
32  return cycles;
33 };

```

Листинг 4.9: Одређивање свих циклуса у прекидном графу

Осим код једнохромозомалних генома, прекидни граф се може конструисати и у случају мултихромозомалних генома и то на потпуно исти начин. На слици 4.15 је приказан прекидни граф за унихромозомални геном $P = (+a+b+c+d+e+f)$ и мултихромозомални геном $Q = (+a-c-f-e)(+b-d)$.

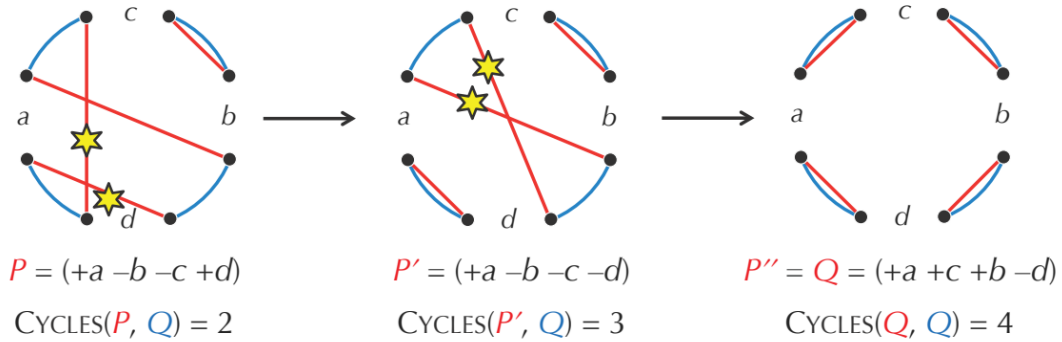


Слика 4.15: Прекидни граф код мултихромозомалних генома [20]

Уколико P и Q геноми садрже исти број блокова синтеније, онда њихов број означавамо са $blocks(P, Q)$. Сваки циклус дужине 2 називамо тривијалним циклусом. Такви циклуси могу настати спајањем различитих гена, али уколико вршимо спајање 2 иста гена, као резултат добијамо n тривијалних циклуса, где важи да је $n = blocks(P, Q)$. Такав граф називамо тривијалним прекидним графом.

Трансформисање генома P у геном P' уз помоћ 2-прекида можемо представити и преко прекидног графа, као трансформацију из $BreakPointGraph(P, Q)$ у $BreakPointGraph(P', Q)$.

Самим тим трансформацију генома P у геном Q можемо представити и као трансформацију $BreakPointGraph(P, Q)$ у $BreakPointGraph(Q, Q)$. Као резултат добијамо тривијални граф.



Слика 4.16: Трансформација у тривијални прекидни граф [20]

Са слике 4.16 се јасно види да се приликом трансформације генома коришћењем 2-прекида повећава број циклуса, па се може дефинисати и формула повећања броја циклуса у новонасталом графу као: $\text{cycles}(Q, Q) - \text{cycles}(P, Q)$.

4.5.2 Израчунавање растојања 2-прекида

Дефиниција: Растојање 2-прекида $d(P, Q)$ је минималан број двоструких прекида потребних да се геном P трансформише у геном Q .

Теорема 2 (Циклусна теорема). *За сваке геноме P и Q , примена 2-прекид трансформације над геномом P повећава број циклуса за највише 1.*

Доказ. Пошто се 2-прекид састоји од додавања две нове црвене гране, самим тим и додаје највише 2 нова циклуса, и брисања две црвене гране, што значи и брисање најмање једног циклуса, може се закључити да је највећи број циклуса који можемо добити приликом примењивања 2-прекида јесте једнак 1. \square

Теорема јасно указује на то да сваки 2-прекид може повећати број циклуса за највише 1. Идеалан сценарио би био када би се са сваким 2-прекидом повећавао број циклуса за један. Следећа теорема говори да увек постоји бар један 2-прекид који ће довести до повећања броја циклуса.

Теорема 3 (Теорема о растојању 2-прекида). *Растојање 2-прекида између генома P и Q је једнако $\text{blocks}(P, Q) - \text{cycles}(P, Q)$.*

Доказ. Претходно је наведено да се број циклуса приликом коришћења 2-прекида увећао и да на крају трансформације он износи $\text{cycles}(Q, Q) - \text{cycles}(P, Q)$.

Пошто је $cycles(Q, Q)$ једнак броју блокова синтеније у оба генома, то јест $blocks(P, Q)$, због тога што тривијални граф садржу само тривијалне циклусе, формула за повећање броја циклуса се може написати и на следећи начин: $blocks(P, Q) - cycles(P, Q)$. Што значи да је за трансформацију генома P у геном Q , неопходно је повећати број циклуса од $cycles(P, Q)$ до $blocks(P, Q)$. Сваки 2-прекид може повећати број циклуса за највише 1. Дакле, најмањи број ових прекида потребан за трансформацију је: $d(P, Q) \geq blocks(P, Q) - cycles(P, Q)$. Пошто су P и Q два различита генома, значи да у графу постоји барем један нетривијални циклус (садржи више од 2 гране). Разбијање таквог циклуса коришћењем 2-прекида, добијају се 2 циклуса, што значи да број циклуса бива повећан за 1, што даље имплицира:

$$d(P, Q) = blocks(P, Q) - cycles(P, Q)$$

□

Начин на који се израчунава минималан број 2-прекида потребних да се геном P трансформише у геном Q приказан је у листингу 4.10:

```

1  getTwoBreakDistance(input: { P: number[] []; Q: number[] [] }): number {
2      const syntenyBlocks = input.P[0].length;
3      // Креирају се црвене и плаве гране
4      const redEdges = coloredEdges(input.P);
5      const blueEdges = coloredEdges(input.Q);
6      // Од црвених и плавих грана се формира прекидни граф
7      const breakPointGraph = initializeGraph(redEdges.concat(blueEdges));
8      // Израчунава се број циклуса
9      const numberOfCycles = getCycles(breakPointGraph);
10
11     // Постојање 2-прекида се рачуна као разлика
12     // блокова синтеније и броја циклуса
13     return syntenyBlocks - numberOfCycles.length;
14 }

```

Листинг 4.10: Растојање 2-прекида

На слици 4.17 је приказан резултат имплементације алгоритма на примеру две пермутације дужине 11. Улазни параметри су дужина пермутација и пермутације између којих треба израчунати растојање 2-прекида, које представља резултат извршавања алгоритма.

2-prekid растојanje

Unesite dužinu permutacije

Unesite permutacije

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

Izračunaj растојanje

Rastoјanje je: 7

Слика 4.17: Пример примене алгоритма за одређивање 2-прекид растојања

Листинг 4.11 приказује најкраћи сценарио, тј. минималан број корака у процесу трансформације генома P у геном Q за који је претпостављено да је идентичка пермутација. У сваком кораку се бирају гране над којима ће бити извршен 2-прекид којим се добија један тривијалан циклус. Овај процес се понавља докле год постоје нетривијални циклуси. Уз коначан резултат растојања 2-прекида, приказују се и сви кораци трансформације, тј. све пермутације у процесу.

```

1  shortestRearrangementScenario(input: { P: number [][]; Q: number [][] }){
2    let { P, Q } = input;
3    let permutationDistance = 0;
4    const permutations = [];
5
6    // Одређивање црвених и плавих ирана
7    const redEdges = coloredEdges(P);
8    const blueEdges = coloredEdges(Q);
9
10   // Формирамо ирекидни ираф од црвених и плавих ирана
11   let breakpointGraph = initializeGraph([...redEdges, ...blueEdges]);
12

```

```

13   while (true) {
14
15     // Формирамо све циклусе у непрекидном графу
16     const cycles = getCycles(breakpointGraph);
17     let selectedCycle: number[] | null = null;
18
19     // Тражимо нештривијалне циклусе
20     for (const cycle of cycles) {
21       if (cycle.length > 2) {
22         selectedCycle = cycle;
23         break;
24       }
25     }
26
27     // Уколико не више не постоје штривијални циклуси,
28     // штрансформација је завршена
29     if (selectedCycle === null) {
30       break;
31     }
32
33     // Креирање грана у одабраном циклусу
34     const selectedCycleEdges = selectedCycle.map((_, i) =>
35       i < selectedCycle.length - 1
36         ? [selectedCycle[i], selectedCycle[i + 1]]
37         : [selectedCycle[selectedCycle.length - 1], selectedCycle[0]],
38     );
39
40     const n = selectedCycleEdges.length;
41     let k = Math.floor(Math.random() * n);
42
43     // Бирамо произвољну грану из циклуса
44     let selectedEdge = selectedCycleEdges[k];
45     let [u, v] = selectedEdge;
46
47     // Уколико одабрана грана није плава, к повећавамо за 1
48     // знајући да црвене и плаве гране иду наизменично у непрекидном
49     // графу
50     if (
51       !blueEdges.some(
52         (edge) =>
53           (edge[0] === u && edge[1] === v) ||
54           (edge[0] === v && edge[1] === u),
55     )

```

```

56     ) {
57         k = (k + 1) % n;
58     }
59
60     // Одабрали смо њлаву трану да бисмо преко ње моћли да изаберемо
61     // њој две суседне црвене тране над којима желимо да вршимо
62     // прекиде, јер оне предшавају редослед блокова у геному P
63     selectedEdge = selectedCycleEdges[k];
64     const previousRedEdge = selectedCycleEdges[(k - 1 + n) % n];
65     const nextRedEdge = selectedCycleEdges[(k + 1) % n];
66
67     const [i_p, _] = previousRedEdge;
68     const [i, __] = selectedEdge;
69     const [j, j_p] = nextRedEdge;
70
71     // У оквиру одабраног циклуса, вршимо 2-прекид
72     breakpointGraph =
73         twoBreakOnGenomeGraph(breakpointGraph, i, i_p, j, j_p);
74     permutationDistance += 1;
75
76     // Вршимо 2-прекид и над геномом да бисмо имали
77     // приказ промена по корацима
78     P = twoBreakOnGenome(P, i, i_p, j, j_p);
79     permutations.push(P);
80 }
81
82 return { permutations, permutationDistance };
83 }

```

Листинг 4.11: Најкраћи сценарио P у Q

Као резултат имплементације претходног алгоритма, добијају се израчунато 2-прекид растојање и листа свих пермутација које су претходиле идентичкој. Улазни параметри су дужина пермутације и пермутација коју треба трансформисати до идентичке. Пример примене алгоритма за одређивање најкраћег сценарија трансформације генома P у геном Q у електронској лекцији приказан је на слици 4.18.

Najkraći scenario transformacije P u identičku permutaciju

Unesite dužinu permutacije

Unesite permutaciju

Prikaži permutacije i rastojanje

2-prekid rastojanje je: 8

Preokreti:

1. [1 -7 6 -10 9 -8 2 3 11 5 -4]
2. [1 -7 6 -3 -2 8 -9 10 11 5 -4]
3. [1 -7 6 -3 -2 8 -9 10 11] [4 -5]
4. [1 2 3 -6 7 8 -9 10 11] [4 -5]
5. [1 2 3 4 -5 -6 7 8 -9 10 11]
6. [1 2 3 4 7 8 -9 10 11] [5 6]
7. [1 2 3 4 5 6 7 8 -9 10 11]
8. [1 2 3 4 5 6 7 8 9 10 11]

Слика 4.18: Пример примене алгоритма за одређивање најкраћег сценарија трансформације хромозома P у идентичку пермутацију у електронској лекцији

4.5.3 Помоћне функције

Због своје комплексности, функција која врши трансформацију хромозома P у хромозомом Q је подељена на неколико мањих функција од којих су неке приказане у претходним секцијама. Све преостале функције које чине овај алгоритам су приказане у оквиру ове секције.

```

1 export const twoBreakOnGenomeGraph = (
2   breakPointGraph: Record<number, number[]>,
3   i: number, i_p: number, j: number, j_p: number,
4 ): Record<number, number[]> => {
5   breakPointGraph = removeUndirectedEdge(breakPointGraph, [i, i_p]);
6   breakPointGraph = removeUndirectedEdge(breakPointGraph, [j, j_p]);
7   breakPointGraph = addUndirectedEdge(breakPointGraph, [i, j]);
8   breakPointGraph = addUndirectedEdge(breakPointGraph, [i_p, j_p]);
9
10  return breakPointGraph;
11 };

```

Листинг 4.12: Извршавање 2-прекида над прекидним графом

```

1 const removeUndirectedEdge = (
2   breakPointGraph: Record<number, number[]>,
3   [edgeNode1, edgeNode2]: [number, number],
4 ): Record<number, number[]> => {
5
6   // За неусмерену грану (u, v) бришемо v из листе суседа од u и
7   // обрнуто
8   const index1 = breakPointGraph[edgeNode1].indexOf(edgeNode2);
9   if (index1 > -1) {
10    breakPointGraph[edgeNode1].splice(index1, 1);
11  }
12
13  const index2 = breakPointGraph[edgeNode2].indexOf(edgeNode1);
14  if (index2 > -1) {
15    breakPointGraph[edgeNode2].splice(index2, 1);
16  }
17
18  return breakPointGraph;
19 };

```

Листинг 4.13: Брисање постојећих неусмерених грана

```

1 const addUndirectedEdge = (
2   breakPointGraph: Record<number, number[]>,
3   [edgeNode1, edgeNode2]: [number, number],
4 ): Record<number, number[]> => {
5   if (!breakPointGraph[edgeNode1]) {
6     breakPointGraph[edgeNode1] = [];
7   }
8   if (!breakPointGraph[edgeNode2]) {
9     breakPointGraph[edgeNode2] = [];
10  }
11
12  //За формирање нове неусмерене њране (u, v)
13  //догајемо v у листу суседа од u и обрнуто
14  breakPointGraph[edgeNode1].push(edgeNode2);
15  breakPointGraph[edgeNode2].push(edgeNode1);
16
17  return breakPointGraph;
18 };

```

Листинг 4.14: Додавање нових неусмерених грана

```

1 export const twoBreakOnGenome = (
2   P: number[][],
3   i: number,
4   i_p: number,
5   j: number,
6   j_p: number,
7 ): number[][] => {
8   //Прећварамо линеарну рејрезентацију хромозома у циркуларну
9   let genomeGraph = initializeGraph([...blackEdges(P), ...coloredEdges(P)]);
10  //Вршимо 2-прекид над њграфом
11  genomeGraph = twoBreakOnGenomeGraph(genomeGraph, i, i_p, j, j_p);
12
13  //Враћамо хромозом у линеарну рејрезентацију
14  P = graphToGenome(genomeGraph);
15
16  return P;
17 };

```

Листинг 4.15: Приказ 2-прекида над геномом

```
1 export const blackEdges = (P: number[][]): [number, number][] => {
2   const edges: [number, number][] = [];
3
4   for (const chromosome of P) {
5     // [1, -2, -3, 4] њосѡаје [1 2 4 3 6 5 7 8]
6     const cycleNodes: number[] = chromosomeToCycle(chromosome);
7     const m: number = cycleNodes.length;
8
9
10
11    // Усмерене ѓране су 1-2 4-3 6-5 итд.
12    for (let j = 0; j < m; j += 2) {
13      edges.push([cycleNodes[j], cycleNodes[j + 1]]);
14    }
15  }
16
17  return edges;
18 };
```

Листинг 4.16: Креирање усмерених (црних) грана

4.6 Места преуређивања у људском геному

4.6.1 Модел случајних региона преуређивања и теорема о растојању 2-прекида

У поглављу 3.2 је било речи о моделу случајних региона преуређивања и хипотези да у људском геному не постоји образац за места прекидних тачака, названој моделом случајних прекида. Ако се претпостави да је ова хипотеза тачна, n преокрета на линеарном хромозому ће произвести приближно $2 \cdot n + 1$ блокова синтеније, пошто је вероватноћа да ће се две суседне локације у геному користити као тачке 2-прекида веома мала. Слично томе, n случајних 2-прекида примењених на кружни хромозом, произвешће $2n$ блокова синтеније. С обзиром на то да је експериментално утврђено да и код човека и код миша постоји 280 блокова синтеније [20], изводи се закључак да на еволуционом путу који дели миша и човека, 2-прекида има приближно $280/2 = 140$. Са друге стране, графикон тачке прекида конструисан на 280 блокова синтеније човека и миша садржи 35 наизменичних циклуса, тако да је, на основу теореме о растојање 2-прекида, то растојање између ових генома минимум $280 - 35 = 245$. Добијен број прекида је значајно већи од броја до којег доводи модел случајног прекида. То, даље, иако нема чврстог математичког доказа, модел случајног прекида проглашава нетачним.

4.6.2 Модел трусних региона преуређивања

Претходно наведена контрадикција модела случајног прекида са теоријом растојања 2-прекида, довела је до алтернативног дела названог *модел крхког прекида*. По овом моделу, сваки геном сисара је састављен од дугих чврстих региона на које ретко утиче преуређивање и кратких и крхких региона који служе као места преуређивања. Код људи и мишева, крхки региони чине око 3% генома. Али, као и велики број биолошких теорија и модела који су у супротности са математичким постулатима и овај модел није математички доказан.

4.7 Формирање блокова синтеније

У основи свих претходно приказаних алгоритама су били подаци који се односе на блокове синтеније. Од почетка је постављена претпоставка да су блокови синтеније већ формирани, без улажења у то на који начин се они формирају, па је потребно размотрити и то питање.

Један од начина формирања блокова синтеније је помоћу k -грама, односно, лоцирањем ниски нуклеотида дужине k које су идентичне у генима различитих врста. Блок синтеније је формиран од низа сличних гена који се појављују у два генома у истом редоследу, па процес поређења генома миша и човека почиње проналажењем свих позиција идентичних k -грама. Лоцирање таквих ниски у оба генома, поготово када је k велики број (на пример 30), сугерише да потичу од сродних гена.

Сматра се да је k -грам дељен између два генома уколико се тај k -грам или његов комплемент налазе у оба генома. На слици 4.19 се може видети да постоје четири дељена триграма између генома АААСТСАТС и ТТТСАААТС. Други триграм, означен плавом бојом, приказује случај када су подниске међусобно обрнуто комплементарне.



Слика 4.19: 3-мер примери између два генома [20]

Сваки пронађени k -грам у обе секвенце се може представити као блок синтеније који се у сваком од генома налази на различитим позицијама као последица преуређивања насталих током еволуције.

Листинг 4.17 приказује начин одређивања индекса почетка k -грама у оба генома. Листинг 4.18 приказује помоћну функцију која је искоришћена у наведеном алгоритму.

```

1 getKmers(input: {
2   k: number;
3   chromosome1: string;
4   chromosome2: string;
5 }): [number, number][] {
6   const { k, chromosome1, chromosome2 } = input;
7   const n = chromosome1.length;
8   let kMers: [number, number][] = [];
9
10  // Пролазак кроз оба генома
11  for (let i = 0; i < n - k + 1; i++) {
12    const substring1 = chromosome1.slice(i, i + k);
13    for (let j = 0; j < n - k + 1; j++) {
14      const substring2 = chromosome2.slice(j, j + k);
15      // Тражи се њоклапање њодниски дужине k
16      if (isKmer(substring1, substring2)) {
17        // Уколико је услов задовољен, чувамо индексе њочешка
18        // $k$ грана у оба генома
19        kMers.push([i, j]);
20      }
21    }
22  }
23
24  return kMers;
25 }

```

Листинг 4.17: Одређивање низа индекса k -грама

```

1 export const isKmer = (substring1: string, substring2: string) => {
2   if (substring1 === substring2) {
3     return true;
4   }
5   // Ако њодниске нису једнаке, треба њроверити да ли су комплеметарне
6   const substring2Complement = substring2
7     .split('')
8     .map((nucleotide) => nucleotideMapper[nucleotide])
9     .join('');
10
11   return substring1 === substring2Complement;
12 };

```

Листинг 4.18: Провера да ли су две подниске k -грами

Пример имплементације алгорита за одређивање k -грама је приказан на слици 4.20. Улазни параметри су дужина k -грама и 2 секвенце генома. Као излаз, добијају се сви k -грами задате дужине за унете секвенце генома.

Одређивање индекса k -грама

Unesite duzinu k -grama

3

Unesite 2 hromozoma, dozvoljene vrednosti su A, G, C i T

AAACGTATCATC

TTCCATAAATC

Izračunaj

AAACGTATCATC	AAACGTATCATC	AAACGTATCATC	AAACGTATCATC
TTCCATAAATC	TTCCATAAATC	TTCCATAAATC	TTCCATAAATC
(0, 0)	(0, 7)	(4, 4)	(5, 5)
AAACGTATCATC	AAACGTATCATC	AAACGTATCATC	
TTCCATAAATC	TTCCATAAATC	TTCCATAAATC	
(6, 9)	(8, 4)	(9, 9)	

Слика 4.20: Пример примене алгорита за одређивање k -грама у електронској лекцији

Глава 5

Закључак

Веза математике као научне дисциплине и природе је нераздвојива. Већину природних облика и појава је могуће представити математичким обрацима, па не чуди што научници покушавају да процес еволуције представе помоћу математичких модела. Захваљујући информатици која је омогућила мапирања и симулације великих размера, донекле се и дошло до закључка на који начин се одвијала еволуција.

Промене преуређивања у основи имају процес сегментације генома на одређеним местима, такозваним прекидним тачкама, након чега се ти сегменти које називамо блоковима синтеније, поново састављају али у другачијем распореду. Блокови синтеније нису случајно формирани већ представљају исте или сличне секвенце гена два генома која поредимо.

У раду су приложени, а у електронској лекцији имплементирани алгоритми који симулирају претпостављене еволутивне процесе. Сваки за циљ има да на најкраћи начин изврши трансформацију задатог генома у геном са жељеним распоредом гена.

Процес сортирања по преокретима је симулиран кроз похлепни алгоритам који приказује све кораке у преуређивању од задате до идентичке пермутације, а имплементиран је и алгоритам за одређивање броја прекидних тачака.

Представљајући геноме као графовске структуре омогућена је лакша анализа преуређивања генома. Коришћењем тог концепта су имплементирани алгоритми за одређивање растојања 2-прекида и најкраћег растојања у процесу трансформације из једног у други геном. Уз помоћ овакве репрезентације, изведена је теорема о растојању 2-прекида, којом је оповргнут модел случајних прекида који је настао као резултат рачунарских симулација. Из

ове контрадикције је произведен закључак да упркос томе што места преуређивања на хромозомима настају случајно, постоје одређени сегменти који су подложнији преуређивању и који се називају крчки региони.

С обзиром на то да процес стварања блокова синтеније није насумичан, већ су формирани од низа сличних гена који се појављују у истом редоследу, у секцији 4.7 је описано формирање блокова синтеније коришћењем k -грама. Такође је имплементиран и алгоритам за одређивање k -грама.

Због ефикасности појединих алгоритама, било је неопходно коришћење помоћних функција. У њима се може сагледати на које начине су представљени различити процеси: начин на који се геном у линеарној репрезентацији може приказати у цикличној и обрнуто, креирање прекидног графа помоћу одређивања неусмерених грана, приступ приликом одређивања циклуса у прекидном графу и како се приликом примене 2-прекида врше додавање и брисање неусмерених грана.

Комплексност теме која је обрађивана оставља простор за даљи рад и проширивање колекције алгоритама за симулацију еволутивних процеса. Један од изазова може бити симулација растојања k -прекида који су код сисара мало вероватни за $k > 3$, према тврдњи већине биолога [2]. Ипак, за $k = 3$ се несумњиво дешавају у еволуцији, иако је још увек нејасно колико често код сисара [2].

Библиографија

- [1] Kandola A. What to know about genetic disorders. <https://www.medicalnewstoday.com/articles/genetic-disorders>, 2020 (pristupljeno: jun 2024).
- [2] Max A. Alekseyev and Pavel A. Pevzner. Multi-break rearrangements and chromosomal evolution. *Theoretical Computer Science*, 395(2):193–202, 2008. SAIL – String Algorithms, Information and Learning: Dedicated to Professor Alberto Apostolico on the occasion of his 60th birthday.
- [3] Boundless. Homologs, orthologs, and paralogs. [https://bio.libretexts.org/Bookshelves/Microbiology/Microbiology_\(Boundless\)/07%3A_Microbial_Genetics/7.13%3A_Bioinformatics/7.13C%3A_Homologs_Orthologs_and_Paralogs](https://bio.libretexts.org/Bookshelves/Microbiology/Microbiology_(Boundless)/07%3A_Microbial_Genetics/7.13%3A_Bioinformatics/7.13C%3A_Homologs_Orthologs_and_Paralogs), (pristupljeno: jun 2024).
- [4] Goran Brajušković. Genomika. *Vojnosanitetski pregled*, 63(6):604, 2006.
- [5] Encyclopædia Britannica. Genetic code. <https://www.britannica.com/science/genetic-code>, Last Updated: Jul 13, 2024 (pristupljeno: avgust 2024).
- [6] Nature Education. Ribosomes, transcription, and translation. <https://www.nature.com/scitable/topicpage/ribosomes-transcription-and-translation-14120660/>, 2014 (pristupljeno: jul 2024).
- [7] Pfizer Inc. How genetically related are we to bananas? https://www.pfizer.com/news/articles/how_genetically_related_are_we_to_bananas, 2024 (pristupljeno: jun 2024).

- [8] National Human Genome Research Institute. Why mouse matters. <https://www.genome.gov/10001345/importance-of-mouse-genome>, 2010 (pristupljeno: jun 2024).
- [9] National Human Genome Research Institute. Translocation. <https://www.genome.gov/genetics-glossary/Translocation>, 2024 (pristupljeno: jun 2024).
- [10] National Human Genome Research Institute. Ribonucleic acid (rna). <https://www.genome.gov/genetics-glossary/RNA-Ribonucleic-Acid>, Last Updated: August 29, 2024 (pristupljeno: avgust 2024).
- [11] National Human Genome Research Institute. Nucleotide. <https://www.genome.gov/genetics-glossary/Nucleotide#:~:text=A%20nucleotide%20is%20the%20basic, and%20a%20nitrogen%2Dcontaining%20base.>, Last Updated: August 5, 2024 (pristupljeno: avgust 2024).
- [12] I. Jovanović M. Živković A. Stanković J. Kuveljić, T. Đurić. Osnove primene analize DNK u medicini. *Srce i krvni sudovi*, 32(2):104–109, 2013.
- [13] Crow J.F. Unequal by nature: a geneticist's perspective on human differences. <https://www.amacad.org/publication/unequal-nature-geneticists-perspective-human-differences#:~:text=As%20a%20result%2C%20we%20share,2%20percent%20of%20our%20D,2002> (pristupljeno: jun 2024).
- [14] Stubbs L. Of mice and men. <https://www.llnl.gov/sites/www/files/2020-05/compgen-str-dec-99.pdf>, 1999 (pristupljeno: jun 2024).
- [15] Tsai I.J. Liu D., Hunt M. Inferring synteny between genome assemblies: a systematic evaluation. *BMC Bioinformatics* 19, 26 (2018). <https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-018-2026-4>, (pristupljeno: jun 2024).
- [16] MedlinePlus.gov. What are proteins and what do they do? <https://medlineplus.gov/genetics/understanding/howgeneswork/protein/>, 2021 (pristupljeno: jun 2024).
- [17] MedlinePlus.gov. What is a gene? <https://medlineplus.gov/genetics/understanding/basics/gene/>, 2024 (pristupljeno: jun 2024).

- [18] P. Z. Myers. Synteny: Inferring ancestral genomes. <https://www.nature.com/scitable/topicpage/synteny-inferring-ancestral-genomes-44022/>, 2008 (pristupljeno: jul 2024).
- [19] National Institute of General Medical Sciences. Chromosome-hoarding ovarian cancer cells may help treatment. <https://www.cancer.org/research/acs-research-highlights/ovarian-cancer-research-highlights/chromosome-hoarding-ovarian-cancer-cells-may-help-treatment.html>, 2023 (pristupljeno: jun 2024).
- [20] Pavel Pevzner Phillip Compeau. *Bioinformatics Algorithms: An Active Learning Approach*. Active Learning Publishers, 2014.
- [21] Nature Rat Consortium. Why mouse matters. <https://ksvi.mff.cuni.cz/~mraz/bioinf/BioAlg11-6.pdf>, 2004 (pristupljeno: jun 2024).
- [22] Ole Kristian Tørresen, Bastiaan Star, Sissel Jentoft, Kjetill S. Jakobsen, and Alexander J. Nederbragt. The new era of genome sequencing using high-throughput sequencing technology: generation of the first version of the atlantic cod genome. In *Genomics in Aquaculture*, pages 1–20. Academic Press, San Diego, 2016.
- [23] Özlem Barış, Mehmet Karadayı, Derya Yanmış, and Medine Gulluce. Genomic rearrangements and evolution. In *Current Progress in Biological Research*, chapter 2. IntechOpen, Rijeka, 2013.

Биографија аутора

Филип Јовашевић рођен је је у Београду. 7. августа 1996. године. Након завршене Основне школе „Љуба Ненадовић” и XV београдске гимназије у Београду, као носилац Вукове дипломе у дотадашњем школовању, 2015. године уписује Математички факултет, смер Информатика. Мастер студије, програм Информатика, уписује 2019. године. Након положених испита са студијског програма, јануара 2021. године, запошљава се у фирми *Levi9 Technology Services* као *JavaScript developer* у којој се задржао до јула 2024. године, када прелази у *Deversity*. У досадашњем раду, активно је учествовао на пројектима који подразумевају примену различитих програмских језика, оквира и алата као што су: *NodeJs*, *Typescript*, *Storybook*, *React*, *NextJs*, *AWS services*, *Gitlab-ci*, *Jest*.