

Zahvaljujem svojem mentoru doc. dr. sc. Krešimiru Križanoviću na strpljenju i pomoći.

Zahvaljujem svojoj obitelji, prijateljima i kolegama na pruženoj podršci tijekom studiranja i izrade ovog rada.

Sadržaj

Uvod.....	1
1. Korištene tehnologije i alati.....	3
1.1. Java	3
1.2. Linux.....	3
1.3. Git.....	3
1.4. Apache Maven.....	3
1.5. Minimap2.....	3
1.6. Ram.....	4
1.7. Raven	4
1.8. Quast	4
2. Formati podataka.....	5
2.1. FASTA.....	5
2.2. FASTQ.....	6
2.3. SAM.....	7
2.4. PAF.....	8
3. Pokretanje procesa	10
3.1. Pomoćni program <i>ProcessRunner</i>	10
3.2. Stvaranje datoteka	12
4. Pokretanje aplikacije	15
4.1. Izbornik <i>Tools</i>	15
4.2. Izbornik <i>Processes</i>	16
4.3. Izbornik <i>Help</i>	17
4.4. Sučelja za pokretanje procesa	19
4.4.1. Sučelje <i>Minimap2 alignment</i>	19

4.4.2.	Sučelje <i>Minimap2 mapping</i>	21
4.4.3.	Sučelje <i>Minimap2 indexing</i>	21
4.4.4.	Sučelje <i>Ram</i>	22
4.4.5.	Sučelje <i>Raven</i>	23
5.	Analiza rezultata	24
5.1.	Analiza SAM datoteka.....	24
5.2.	Analiza PAF datoteka.....	24
5.3.	Analiza FASTA datoteka nastalih <i>de novo</i> sastavljanjem	25
	Zaključak.....	27
	Literatura	28
	Sažetak	29
	Summary	30

Uvod

Gen je osnovna molekularna jedinica nasljeđivanja svih živih organizama. Skup svih gena nekog organizma naziva se genom. Genom sadrži sveukupnu nasljednu informaciju nekog organizma, stoga je određivanje slijeda nukleinskih baza unutar njega od velike koristi. Određivanje tog slijeda pomaže nam u boljem razumijevanju svih bioloških interakcija organizma, nasljednih bolesti, tumora i drugih zloćudnih pojava i stanja u organizmu te njihovom liječenju i dijagnosticiranju [1]. Proces određivanja slijeda pojedinih nukleinskih baza unutar DNA ili RNA lanca nazivamo sekvenciranje. Broj nukleotida koji se može sekvencirati varira ovisno o metodi sekvenciranja i njejoj izvedbi. Bez obzira na tehnologiju sekvenciranja, trenutno nismo u stanju odjednom pročitati cijele genome, pa je stoga prije sekvenciranja potrebno podijeliti dulje nukleotidne lance u kraće. Najdominantnija strategija sekvenciranja je *shotgun* sekvenciranje kod kojeg se višestruka očitavanja DNA na slučajan način lome na kraće dijelove koji se kasnije pomoću računalnih programa sastavljaju u kontinuiranu sekvencu [1].

Za poravnanje i sastavljanje genoma koriste se razne metode od kojih su najznačajnije: algoritmi poravnanja, mapiranje na referentni genom i *de novo* sastavljanje. Algoritmi poravnanja koriste dinamičko programiranje kako bi opisali razliku između dva niza, odnosno način na koji se jedan niz može dobiti iz drugog [2]. Najjednostavniji način implementiranja algoritma poravnanja je korištenje matrice dimenzija $(n + 1) * (m + 1)$, pri čemu su n i m duljine nizova koje poravnavamo. Vrijednost svake ćelije dobivamo iz rezultata dobivenih zbrajanjem gornje ćelije s cijenom zamjene baza, zbrajanjem lijeve ćelije s cijenom umetanja baze te zbrajanjem gornje lijeve ćelije s cijenom usklađenosti, odnosno neusklađenosti baza nakon čega uzimamo najveću vrijednost od svih dobivenih rezultata kao vrijednost ćelije [1]. Vrijednosti ćelija prvog stupca i prvog retka ovise o vrsti poravnanja koje može biti: globalno, poluglobalno ili lokalno. Praćenjem puta kojim smo dobili ciljnu ćeliju (koja također ovisi o vrsti poravnanja) dobivamo optimalno poravnanje dvaju niza. Mapiranjem na referentni genom dobivamo područja na referentnom genomu koje najbolje odgovaraju pojedinom fragmentu. Najpopularniji način određivanja regije s kojim se fragment najbolje podudara je usporedbom *minimizera*, leksikografski najmanjih podnizova (tzv. *k-merova*) od svih w podnizova duljine k unutar podniza sekvence [3]. Proces

određivanja *minimizera* sekvence naziva se indeksiranje sekvence. Mapiranje na referentni genom često prethodi algoritmu poravnanja u smislu određivanja regija nad kojima će se vršiti poravnanje. U slučaju kad ne postoji referentni genom onda govorimo o *de novo* sastavljanju koje traži najbolja preklapanja između fragmenata, te ih tako slaže u kontinuiranu sekvencu.

Kako duljina genoma varira između nekoliko tisuća i više od stotinu milijardi parova baza, tako i trajanje procesa pomoću navedenih metoda može varirati između nekoliko minuta i nekoliko tjedana. Postoje brojni alati koji koriste navedene metode za sastavljanje genoma. Neki od tih alata su: *Minimap2*, *Ram* i *Raven*. Problem kod navedenih alata je što se moraju pokretati iz naredbenog retka što većini korisnika ne odgovara jer se nisu navikli na takav način pokretanja programa. Također, jednom kad se pokrene proces, naredbeni redak mora biti uključen dok se proces ne završi. Ako proces traje duži vremenski period, korisnik može vrlo lako zaboraviti na pokrenuti proces, te ukoliko je proces završen ili prekinut korisnik neće biti obavješten. Cilj ovog rada je napraviti aplikaciju s grafičkim sučeljem koja pokreće navedene procese pomoću navedenih alata. Aplikacija mora pokretati procese u pozadini računala kako ne bi ometali korisnika, te mora obavijestiti korisnika kada je proces uspješno završen ili prekinut. Aplikacija također treba pružiti jednostavnu analizu rezultata procesa ukoliko su uspješno završeni.

1. Korištene tehnologije i alati

1.1. Java

Aplikacija je u potpunosti napisana u programskom jeziku *Java*. Za pisanje aplikacije korišten je *Open Java Development Kit 14.0.2*. Za izradu grafičkog sučelja korištena je Javina biblioteka *Swing*.

1.2. Linux

Aplikacija je napravljena za *Linux* operacijske sustave. Izrađena je u *Linux* distribuciji *Ubuntu 20.04* unutar virtualnog stroja *VMware Workstation Pro 16.1.2*.

1.3. Git

Za upravljanjem verzijama koda korišten distribucijski sustav *Git*. Čitav je projekt cijelo vrijeme bio ažuriran i pohranjen na *GitHub* repozitoriju.

1.4. Apache Maven

Za izgradnju cijelog projekta korišten je *Apache Maven 3.6.3*.

1.5. Minimap2

Minimap2 je svestran program za indeksiranje genoma, te poravnanje ili mapiranje DNA ili mRNA sekvenci na referentni genom. Neke od njegovih najčešća uporaba su: mapiranje *PacBio* ili *Oxford Nanopore* genomskih očitavanja na ljudski genom, pronalazak preklapanja između dugačkih očitavanja sa stopom pogreške do ~15%, *splice-aware* poravnanje *Pac-Bio Iso-Seq* ili *Nanopore cDNA* ili direktna RNA očitavanja na referentni genom itd. Za ~10 kb *noisy reads* sekvenci, *Minimap2* se pokazao nekoliko desetaka puta brži od ostalih uobičajenih alata za mapiranje dugačkih očitavanja kao što su *BLASR*, *BWA-MEM*, *NGMLR* i *GMAP*. Također se pokazao trostruko bržim od alata *BWA-MEM* i *Bowtie2* na >100bp *Illumina* kratkim očitanjima. *Minimap2* napisan je u programskom jeziku C sa API-jima u

Pythonu i C-u. Alat nudi brojne opcije određivanja vlastitih parametara za indeksiranje *minimizera*, poravnanja, te mapiranja sekvenci. Također nudi brojne popularne *presetove* za mapiranje i poravnanje kao što su: PacBio ili Nanopore mapiranje na referentni genom, PacBio ili Nanopore preklapanje očitavanja, mapiranje genomskih kratkih očitavanja itd. [4]

1.6. Ram

Ram je C++ inačica alata *Minimap* uz nekoliko dodatnih modifikacija. Služi za mapiranje sekvenci na referentni genom [5]. Korisniku je omogućena izmjena određenih parametara po vlastitoj želji. Alat *Ram* izradio je dr. sc. Robert Vaser sa Zavoda za elektroničke sustave i obradbu informacija u suradnji s mag. ing. Josipom Marićem.

1.7. Raven

Raven je alat za *de novo* sastavljanje dugačkih neispravnih očitavanja uz mogućnost mijenjanja određenih parametara po vlastitoj želji. Alat *Raven* također je izradio dr. sc. Robert Vaser u suradnji s dr. Thanh Le Vietom [6].

1.8. Quast

Quast je alat koji služi za analizu kvalitete očitavanja. U ovom radu koristit će se za analizu rezultata dobivenih *de novo* sastavljanjem. Rezultati analize su brojne datoteke koje sadrže vrijednosti nekih parametara očitavanja kao što su ukupna duljina, N50 duljina, ukupan broj baza gvanina i citozina u sekvenci itd.

2. Formati podataka

2.1. FASTA

FASTA format je format tekstualne datoteke koja u sebi sadrži nizove nukleinskih baza ili aminokiselina s njihovim opisima, odnosno imenima. Svaki niz najčešće započinje s linijom opisa koja započinje sa znakom „>“ koju nazivamo zaglavljem sekvence. U zaglavlju nalazi se ime ili identifikator sekvence uz mogući dodatak njenog opisa. Iza ili umjesto zaglavlja sekvence mogu se nalaziti linije komentara koje sadrže dodatne opise sekvenci. Linije komentara započinju sa znakom „;“. Iza zaglavlja slijedi niz slova znakova kod kojeg svaki znak predstavlja određenu nukleinsku bazu ili aminokiselinu [7] (Tablica 2.1, Slika 2.1). Ne postoji dogovorena ekstenzija datoteke u FASTA formatu, stoga neki alati ne mogu prepoznati neke datoteke kao datoteke u FASTA formatu. Najčešće ekstenzije su: .fasta, .fna, .ffn, .faa, .frn, .fa [8].

Tablica 2.1: Opis i značenje kodnih izraza za nukleinske baze

Kodni izraz nukleinske baze	Značenje
A	adenin
C	citozin
G	guanin
T	timin
U	uracil
(i)	inozin
R	A ili G (purin)
Y	C, T ili U (pirimidin)
K	G, T ili U (baze koje su ketoni)
M	A ili C (baze s amino skupinama)

S	C ili G (jaka veza)
W	A, T ili U (slaba veza)
B	C, G, T ili U
D	A, G, T ili U
H	A, C, T ili U
V	A, C ili G
N	A, C, G, T ili U
-	praznina određene duljine

```
;LCBO - Prolactin precursor - Bovine
; a sample sequence in FASTA format
MDSKGSSQKGSRLLLLLLVSNLLLCQGVVSTPVCNPGNQCQVSLRDLFDRAVMVSHYIHDLSSEMFNEFDKRYAQKGKGFITMALNSCHTSSLPTPEDKEQAQQTTHHEVLMSLILGLLRSWNDPLYHLVTEVRGMKGAPDAILSRATIEEEENKRLLLEGMEMIFGQVIPGAKETEPYPVWSGLPSLQTKDEDARYSAFYNNLLHCLRRDSSKIDTYLKLLNCRITNNNC*

>MCHU - Calmodulin - Human, rabbit, bovine, rat, and chicken
MADQLTEEQIAEFKEAFSLFDKDGDTITTKELGTVMRSLGQNPTEAELQDMINEVDADGNGTIDFPEFLTMMARKMKDSTDSEEEIREAFRVFDKDGNGYISAAELRHVMTNLGEKLTDEEVDEMIREADIDGDGQVNYEEFVQMMTAK*

>gi|5524211|gb|AAD44166.1| cytochrome b [Elephas maximus maximus]
LCLYTHIGRNIYYGSYLYSETWNTGIMLLITMATAFMGYVLPWGQMSFWGATVITNLFSAIPYIGTNLV
EWIWGGFSVDKATLNRFAPFHFILPFTMVALAGVHLTFLHETGSNNPLGLTSDSDKIPFHPYYTIKDFLG
LLILILLILLILLALLSPDMLGDPDNHMPADPLNTPHLIKPEWYFLFAYAILRSVPNKLGGVIALFLSIVIL
GLMPFLHTSKHRSMMLRPLSQALFWTLTMDLLTLTWIGSQPVEYPYTIIGQMASILYFSIILAFPLIAGX
IENY
```

Slika 2.1: Primjer sadržaja FASTA datoteke

2.2. FASTQ

FASTQ format je format tekstualne datoteke koja, uz sve podatke koje sadrži i datoteka u FASTA formatu, sadrži informacije o kvaliteti očitavanja. Svaka sekvenca u datoteci FASTQ formata sadrži 4 linije [11] (Slika 2.2). Prva linija započinje znakom „@“ i sadrži ime i/ili

identifikator sekvence, po mogućnosti uz još neke dodatne informacije. Druga linija predstavlja niz nukleinskih baza sekvence. Treća linija sastoji se od znaka „+“ koji predstavlja separator uz koji se ponekad nalazi ponovljeno ime i opis iz prve linije. Četvrta linija sadrži niz znakova koji predstavljaju kvalitetu očitavanja sekvence iz druge linije. Četvrta linija mora imati jednak broj znakova kao druga linija [9]. Kao i kod FASTA formatiranih datoteka, ne postoji dogovorena ekstenzija za datoteke u FASTQ formatu zbog čega ih neki alati ne mogu uvijek prepoznati. Najčešće korištene ekstenzije su: .fq i .fastq [10].

```
@SIM:1:FCX:1:15:6329:1045 1:N:0:2
TCGCACTCAACGCCCTGCATATGACAAGACAGAATC
+
<>;##=><9=AAAAAAAAA9#:<#<;<<?????#=#
```

Slika 2.2: Primjer sadržaja FASTQ datoteke

2.3. SAM

SAM je tekstualni format koji opisuje poravnanja između fragmenata i referentnog genoma. Sastoji se od dva dijela: zaglavlja i dijela koji opisuje poravnanja. Sve vrijednosti unutar SAM datoteke odvojene su znakom TAB. Svaka linija zaglavlja počinje znakom „@“ nakon čega slijedi oznaka linije zaglavlja koja se sastoji od dva slova. Nakon toga u svakoj liniji zaglavlja slijedi niz parova OZNAKA:VRIJEDNOST odvojenih znakom TAB. U drugom dijelu datoteke opisuje se poravnanje svakog fragmenta na referentni genom. Svaka linija sastoji se od najmanje 11 vrijednosti međusobno odvojenih znakom TAB koje nam daju opis i informacije o poravnanju (Tablica 2.2) [12].

Tablica 2.2: Opis svakog polja u liniji koja predstavlja poravnanje u SAM datoteci

Indeks polja	Naziv polja	Vrsta podatka	Opis
1	QNAME	string	naziv fragmenta
2	FLAG	integer	bitovi koji predstavljaju zastavice
3	RNAME	string	naziv reference
4	POS	integer	početak mapiranja (počinje od 1)
5	MAPQ	integer	kvaliteta mapiranja
6	CIGAR	string	CIGAR string
7	RNEXT	string	naziv referentne sekvence idućeg očitavanja fragmenta
8	PNEXT	integer	početak referentne sekvence idućeg očitavanja fragmenta (počinje od 1)
9	TLEN	integer	duljina fragmenta
10	SEQ	string	niz baza u sekvenci (A, C, G, T, U ili N)
11	QUAL	string	niz koji opisuje kvalitetu očitavanja sekvence

2.4. PAF

PAF je tekstualni format datoteka koji služi za opis regija mapiranja između sekvenci. Svaka linija unutar PAF datoteke sadrži vrijednosti odvojene znakom TAB koje opisuju pojedinu regiju mapiranja sekvence na referentni genom (Tablica 2.3) [13].

Tablica 2.3: Opis vrijednosti svakog polja unutar svake linije PAF datoteke

Indeks polja	Vrsta podataka	Opis
1	string	ime fragmenta
2	integer	duljina fragmenta
3	integer	početak fragmenta (počinje od nule, zatvoren kraj intervala)
4	integer	završetak fragmenta (počinje od nule, otvoren kraj intervala)
5	char	odnos između sekvenci (originalan položaj ili reverzno komplementaran; + ili -)
6	string	ime sekvence referentnog genoma
7	integer	duljina sekvence referentnog genoma
8	integer	početak sekvence referentnog genoma na originalnom lancu (počinje od nule)
9	integer	završetak sekvence referentnog gena na originalnom lancu (počinje od nule)
10	integer	broj usklađenih parova baza
11	integer	duljina bloka poravnanja
12	integer	kvaliteta mapiranja (0-255; 0 je apsolutno podudaranje)

3. Pokretanje procesa

3.1. Pomoćni program *ProcessRunner*

Broj nukleinskih baza u genomu može biti veći i od nekoliko milijardi baza po lancu. Upravo zbog toga neki od procesa mogu trajati satima, a nekad i danima. Zato je bilo potrebno omogućiti pokretanje procesa neovisno o aplikaciji. Radi toga napravljen je pomoćni program *ProcessRunner* čija je glavna zadaća pokretanje procesa na računalu te praćenje njegovog stanja. *ProcessRunner* je također pisan u programskom jeziku Java.

Nakon što korisnik upiše sve podatke u aplikaciju i stisne gumb za pokretanje procesa, aplikacija uzima podatke koje je korisnik upisao i slaže ih u naredbu koja bi se inače pokretala preko naredbene linije. Ovisno o alatu i vrsti procesa, svaka naredba se slaže drugačije, stoga svaka vrsta procesa ima svoju klasu za slaganje naredbi koja nasljeđuje Javinu klasu *SwingWorker*. Objekti te klase pomoću Javine biblioteke *ProcessBuilder* pokreću program *ProcessRunner*. Na taj način je osigurano da se program *ProcessRunner* pokreće neovisno o samoj aplikaciji. *ProcessRunner* će cijelo vrijeme raditi u pozadini računalnog sustava prateći proces koji je pokrenuo. Kao argumente mu se predaju putanja do alata, polje stringova od kojih se sastoji naredba i još jedan string koji označava vrstu procesa koji *ProcessRunner* pokreće.

Prije pokretanja, program odredi procesu njegov identifikacijski broj te stvori dvije datoteke. U jednu datoteku se spremaju rezultati procesa, dok se u drugu preusmjerava tok ispisa alata za stanje pojedinog procesa. Program zatim pokrene proces preko naredbenog retka operacijskog sustava pomoću Javine biblioteke *ProcessBuilder*. Nakon pokretanja procesa, program zapiše informacije o procesu u *all_process.log* datoteku (Kôd 3.1) te korisniku iskoči dijaloški prozor koji ga obavještava da je proces pokrenut i koji je njegov identifikacijski broj (Slika 3.1). U datoteku *all_process.log* se pohranjuju sljedeće informacije o svakom procesu:

- identifikacijski broj procesa
- putanja do datoteke referentnog genoma (ako postoji)
- putanja ili putanje do datoteke ili datoteka fragmenata
- putanja do datoteke sa rezultatima procesa

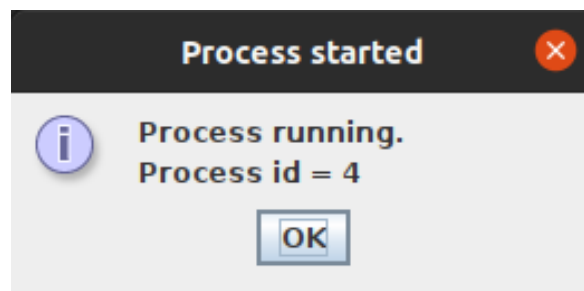
- datum i vrijeme početka procesa
- datum i vrijeme završetka procesa
- vrsta procesa
- status procesa

Navedeni podatci o procesu se zapisuju u jednoj liniji, te su međusobno odvojeni nizom znakova „ : “. Aplikacija svakom procesu dodjeli jednu od sljedećih oznaka kao vrstu procesa:

- MINIMAP2_ALIGN
- MINIMAP2_MAPPING
- MINIMAP2_INDEXING
- RAM_MAPPING
- RAVEN

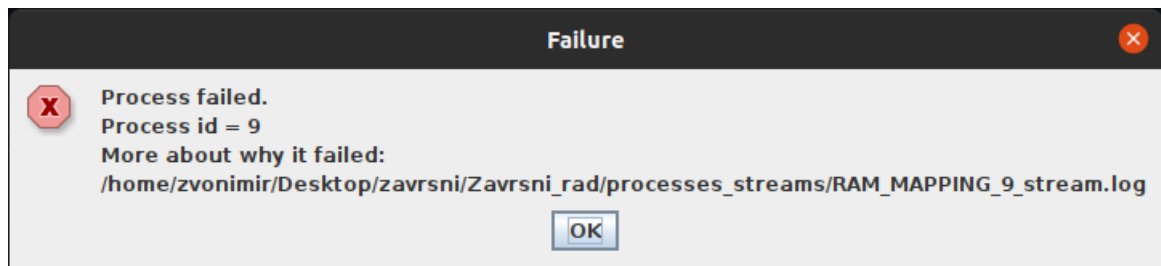
Procesi u aplikaciji mogu imati jedan od sljedećih statusa ovisno o stadiju u kojem se procesi nalaze i njihovoj uspješnosti:

- FINISHED
- FAILED
- RUNNING



Slika 3.1: Primjer dijaloškog okvira prilikom uspješnog pokretanja procesa

Prilikom pokretanja procesu se dodjeljuje oznaka status *RUNNING*. Ako se proces uspješno izvrši, korisnik dobije obavijest putem dijaloškog prozora identifikacijski broj procesa koji je završio, te mu program *ProcessRunner* ažurira status u *FINISHED* u *all_process.log* datoteci. Ukoliko se proces ne uspije uspješno izvršiti, korisnik je obavješten putem dijaloškog prozora u kojem stoji putanja do datoteke koja sadrži ispis stanja od strane alata za taj proces, te mu se ažurira stanje u *FAILED* (Slika 3.2). U oba slučaja se također ažurira vrijeme završetka.



Slika 3.2: Primjer dijaloškog okvira procesa koji se nije uspješno izvršio

```

ProcessBuilder pb = new ProcessBuilder(commands);
pb.redirectError(errorFile);
pb.redirectOutput(outputFile);
Process process = pb.start();
String timeStampStart =
    new SimpleDateFormat("dd.MM.yyyy. HH:mm:ss").format(new
        Date());
StringBuilder sb = new StringBuilder();
sb.append(Integer.toString(id) + " : ");
if (!type.equals(App.PanelType.RAVEN.toString()) &&
    !type.equals(App.PanelType.MINIMAP2_INDEXING.toString()))
    sb.append(refFile + " : ");
else
    sb.append("- : ");
sb.append(querysFiles + " : ");
sb.append(outputFile.getAbsolutePath() + " : ");
sb.append(timeStampStart + " : ");
sb.append("- : ");
sb.append(type + " : ");
sb.append(ProcessStates.RUNNING.toString());
fileContent.add(sb.toString());
Files.write(allProcessLog.toPath(), fileContent);

```

Kôd 3.1: Zapisivanje informacija o procesu u *all_process.log* datoteku prilikom njegovog pokretanja

3.2. Stvaranje datoteka

Kao što je već spomenuto, program *ProcessRunner* prilikom pokretanja svakog procesa stvara dvije nove datoteke (Kôd 3.2). U prvu datoteku spremaju se rezultati procesa. Naziv datoteke koja sprema rezultate procesa je sljedećeg formata:

<vrsta procesa>_<id procesa>.<ekstenzija datoteke>

Pohranjuje se u direktorij *output_files*. Ekstenzija datoteke ovisi o vrsti procesa. Ukoliko se radi o procesu poravnanja pomoću alata *Minimap2*, ekstenzija će glasiti „sam“, pošto je rezultat poravnanja niz vrijednosti u SAM formatu (2.3). Slično tome, ako se radi o procesu mapiranja na referentni genom ekstenzija će glasiti „paf“, pošto je rezultat niz vrijednosti u PAF formatu (2.4). Ako se pak radi o *de novo* sastavljanju, ekstenzija će glasiti „fasta“, pošto je rezultat kontinuirana sekvenca u FASTA formatu (2.1). Putanja do datoteke rezultata procesa je spremljena u datoteku *all_process.log* zajedno s ostalim informacijama o procesu.

U drugu datoteku će se preusmjeriti tok ispisa stanja procesa od strane alata kako bi korisnik mogao pratiti stanje procesa te provjeriti grešku koja je nastupila ukoliko je proces prekinut zbog pogreške u izvođenju. Datoteke se pohranjuju unutar direktorija *processes_streams*. Naziv datoteke sljedećeg je formata:

<vrsta procesa>_<id procesa>_stream.log

```
List<String> fileContent =
    new ArrayList<>(Files.readAllLines(allProcessLog.toPath()));
int id = 1;
if (!fileContent.isEmpty()) {
    String lastProcess =
        fileContent.get(fileContent.size() - 1);
    int lastId =
        Integer.parseInt(lastProcess.split(" : ")[0]);
    id = lastId + 1;
}
String fileName = type + "_" + Integer.toString(id) + ext;
File outputFile = new File("output_files/" + fileName);
outputFile.createNewFile();
```



```
File errorFile = new File("processes_streams/" +  
fileName.substring(0, fileName.length() - ext.length()) +  
"_stream.log");  
errorFile.createNewFile();
```

Kôd 3.2: Stvaranje datoteka za pohranu rezultata procesa i za pohranu ispisa stanja procesa od strane alata

4. Pokretanje aplikacije

Za pokretanje bilo kojeg procesa pomoću određenog alata potrebno je na računalu imati instaliran taj alat. Aplikacija sadrži grafička sučelja za pokretanje svakog procesa unutar pojedinog alata. Prema zadanim postavkama, prilikom pokretanja aplikacije otvara se grafičko sučelje za pokretanje procesa poravnanja pomoću alata *Minimap2*.

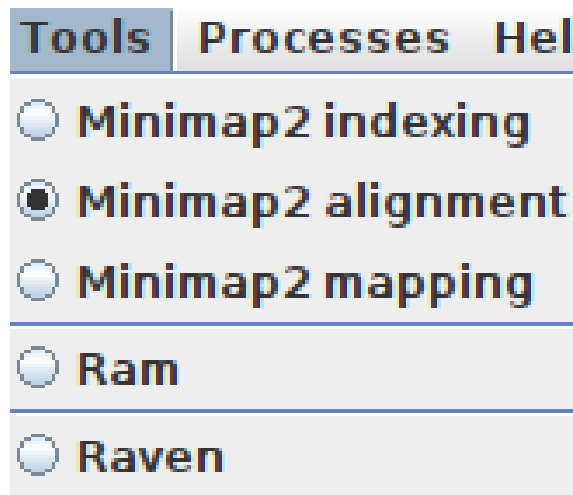
Na vrhu aplikacije nalazi se traka izbornika koja se sastoji od tri izbornika:

- Tools
- Processes
- Help

4.1. Izbornik *Tools*

Klikom na *Tools* izbornik, korisnik vidi popis svih alata koje je moguće pokrenuti i vrste procesa koje je moguće pokrenuti pomoću njih ako ih ima više po pojedinom alatu (Slika 4.1). Klikom na pojedini alat, odnosno na pojedinu vrstu procesa, korisniku se otvara grafičko sučelje s poljima koje je potrebno ispuniti za pokretanje željenog procesa. Procesi koji se nude prilikom klika na izbornik *Tools* su:

- Minimap2 indexing
- Minimap2 alignment
- Minimap2 mapping
- Ram
- Raven



Slika 4.1: Prikaz izbornika *Tools*

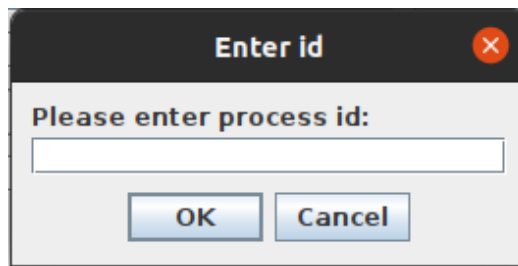
4.2. Izbornik *Processes*

Izbornik *Processes* sadrži opcije pomoću kojih je moguće vidjeti tablični popis procesa sa informacijama o pojedinom procesu (opcija *Checkout processes*) ili pokrenuti analizu pojedinog procesa ukoliko je proces uspješno završio (opcija *Analyze process*). Klikom na opciju *Checkout processes*, korisniku se otvara prozor naziva *Your processes* koji sadrži popis svih pokrenutih te uspješno ili neuspješno završenih procesa s njihovim osnovnim informacijama (Slika 4.2). Procesi u tablici mogu se filtrirati prema vrsti i/ili statusu procesa.

ID	Reference file	Query file(s)	Output file	Start timestamp	End timestamp	Process type	Status type
1	/home/zvonimir/Desktop/brown...	/home/zvonimir/Desktop/brown...	/home/zvonimir/Desktop/zavrsn...	07.06.2021. 17:17:08	07.06.2021. 17:21:57	MINIMAP2_ALIGN	FAILED
2	/home/zvonimir/Desktop/brown...	/home/zvonimir/Desktop/brown...	/home/zvonimir/Desktop/zavrsn...	07.06.2021. 17:19:18	07.06.2021. 17:24:25	MINIMAP2_ALIGN	FAILED
3	/home/zvonimir/Desktop/brown...	/home/zvonimir/Desktop/brown...	/home/zvonimir/Desktop/zavrsn...	07.06.2021. 17:19:44	07.06.2021. 17:25:08	MINIMAP2_ALIGN	FINISHED
4	/home/zvonimir/Desktop/brown...	/home/zvonimir/Desktop/brown...	/home/zvonimir/Desktop/zavrsn...	07.06.2021. 17:20:15	07.06.2021. 17:20:18	MINIMAP2_ALIGN	FAILED
5	/home/zvonimir/Desktop/brown...	/home/zvonimir/Desktop/brown...	/home/zvonimir/Desktop/zavrsn...	07.06.2021. 17:27:15	07.06.2021. 17:28:13	MINIMAP2_ALIGN	FINISHED
6	/home/zvonimir/Desktop/brown...	/home/zvonimir/Desktop/brown...	/home/zvonimir/Desktop/zavrsn...	07.06.2021. 17:28:34	07.06.2021. 17:30:20	MINIMAP2_ALIGN	FINISHED
7	/home/zvonimir/Desktop/brown...	/home/zvonimir/Desktop/brown...	/home/zvonimir/Desktop/zavrsn...	07.06.2021. 17:28:51	07.06.2021. 17:28:52	MINIMAP2_ALIGN	FAILED
9	/home/zvonimir/Desktop/brown...	/home/zvonimir/Desktop/brown...	/home/zvonimir/Desktop/zavrsn...	07.06.2021. 17:36:17	07.06.2021. 17:36:30	MINIMAP2_MAPPING	FAILED
10	/home/zvonimir/Desktop/brown...	/home/zvonimir/Desktop/brown...	/home/zvonimir/Desktop/zavrsn...	07.06.2021. 17:37:17	07.06.2021. 17:37:27	MINIMAP2_MAPPING	FINISHED
11	/home/zvonimir/Desktop/brown...	/home/zvonimir/Desktop/brown...	/home/zvonimir/Desktop/zavrsn...	07.06.2021. 17:37:44	07.06.2021. 17:39:31	MINIMAP2_MAPPING	FINISHED
12	/home/zvonimir/Desktop/brown...	/home/zvonimir/Desktop/brown...	/home/zvonimir/Desktop/zavrsn...	07.06.2021. 17:38:44	07.06.2021. 17:39:39	MINIMAP2_MAPPING	FINISHED
13	/home/zvonimir/Desktop/brown...	/home/zvonimir/Desktop/brown...	/home/zvonimir/Desktop/zavrsn...	07.06.2021. 18:13:26	07.06.2021. 18:22:14	RAM_MAPPING	FINISHED
14	/home/zvonimir/Desktop/brown...	/home/zvonimir/Desktop/brown...	/home/zvonimir/Desktop/zavrsn...	07.06.2021. 18:14:02	07.06.2021. 18:30:33	RAM_MAPPING	FINISHED

Slika 4.2: Prozor *Your processes* s tabličnim prikazom svih procesa

Odabirom opcije *Analyze process* otvara se mali prozor koji sadrži polje u koje korisnik može upisati identifikacijski broj procesa čije rezultate želi analizirati (Slika 4.3). Rezultati procesa mogu se analizirati ako je proces uspješno završen (ima status *FINISHED*) i ako se ne radi o procesu indeksiranja pomoću alata *Minimap2*.



Slika 4.3: Prozor za upis identifikacijskog broja za analizu rezultata procesa

4.3. Izbornik *Help*

Pod izbornikom *Help* korisnik može pronaći informacije o pojedinom alatu, samoj aplikaciji i o dodatnim parametrima koje korisnik može mijenjati (Slika 4.4). Opcije koje nude više informacija o pojedinom alatu su ustvari hiperveze koje otvaraju GitHub stranicu alata na kojem pišu sve ažurirane informacije o alatu. Također, izbornik *Help* nudi opcije koje korisniku ispisuju verziju alata kojeg koriste na svojem računalu u dijaloškom prozoru (Slika 4.5).

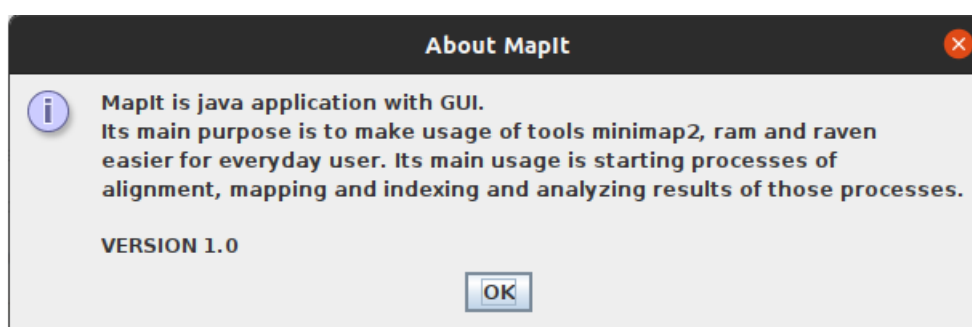


Slika 4.4: Prikaz izbornika *Help*

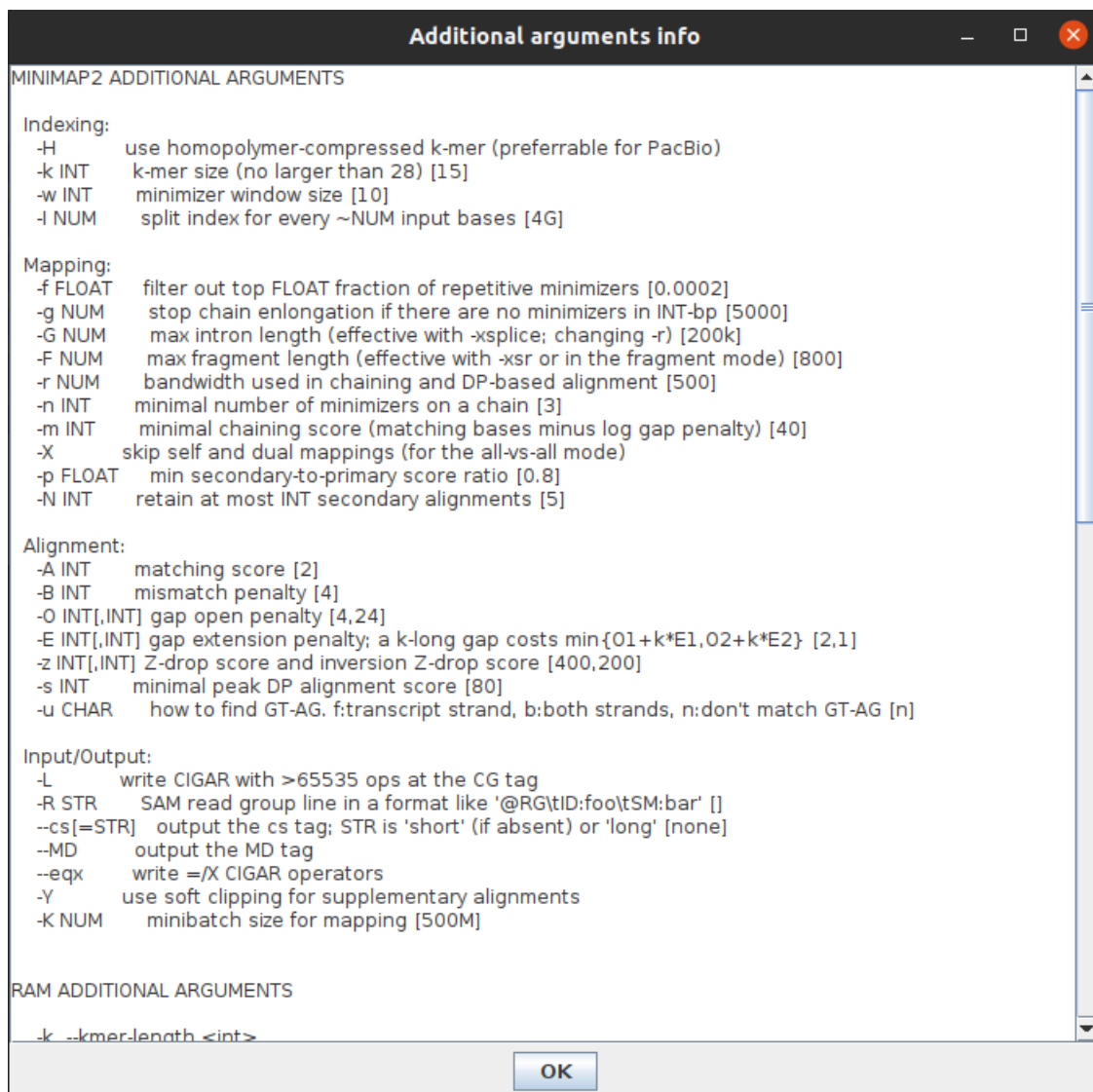


Slika 4.5: Primjer ispisa verzije alata ram

Opcija koja nudi više informacija o samoj aplikaciji također otvara dijaloški prozor koji nudi kratki opis aplikacije i njenu verziju (Slika 4.6). Zadnja opcija u izborniku nudi više informacija koje dodatne parametre korisnik može mijenjati prilikom pokretanja svake vrste procesa. Zbog velikog broja parametara koje korisnik može mijenjati, većina parametara nije uključeno u grafička sučelja za pokretanje procesa. Umjesto toga, svako grafičko sučelje sadrži polje za unos dodatnih parametara ukoliko ih korisnik želi mijenjati. Dodatni parametri upisuju se u polje na isti način kojim se upisuju u naredbenu liniju. Klikom na tu opciju otvara se prozor koji sadrži popis svih dodatnih parametara koje nisu uključene kao polje za unos unutar grafičkih sučelja te njihove opise te zadane (default) vrijednosti (Slika 4.7).



Slika 4.6: Prozor s ispisom opisa aplikacije



Slika 4.7: Popis svih dodatnih parametara s njihovim opisom

4.4. Sučelja za pokretanje procesa

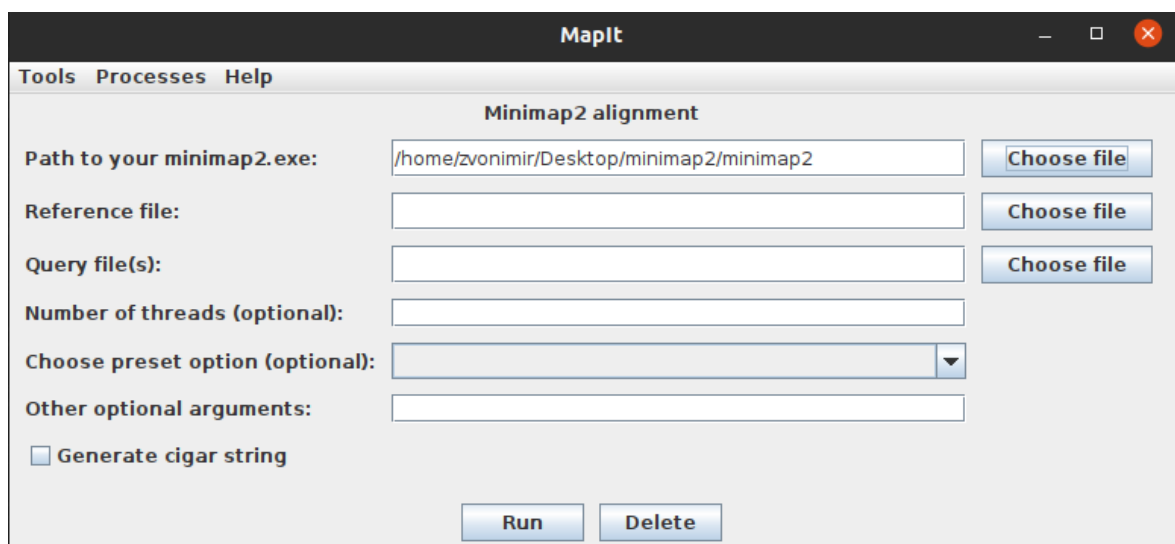
4.4.1. Sučelje *Minimap2 alignment*

Sučelje služi za unos podataka i pokretanje procesa poravnanja između dvije sekvence pomoću alata *Minimap2*. U prvom retku sučelja *Minimap2 alignment* nalazi se putanja do korisnikovog *Minimap2* programa. Aplikacija prilikom prvog pokretanja pomoću naredbe *find* u Linux operacijskom sustavu pokušava pronaći putanju do alata *Minimap2*. Kako prvi rezultat te naredbe nije uvijek točna putanja do programa, korisnik može samostalno

promijeniti putanju do programa nakon čega će ta putanja biti spremljena u .log datoteku. Ostatak sučelja sadrži sljedeća polja (Slika 4.8):

- putanja do FASTA ili FASTQ datoteke referentnog genoma
- putanja ili putanje do FASTA ili FASTQ datoteka sa nizom sekvenci koje želimo poravnati s referentnim genomom
- broj dretvi (opcionalno)
- odabir *preseta* (opcionalno)
- polje za unos dodatnih parametara po korisnikovoj želji
- *checkbox* koji označava da li je potrebno stvoriti CIGAR string (opcionalno)

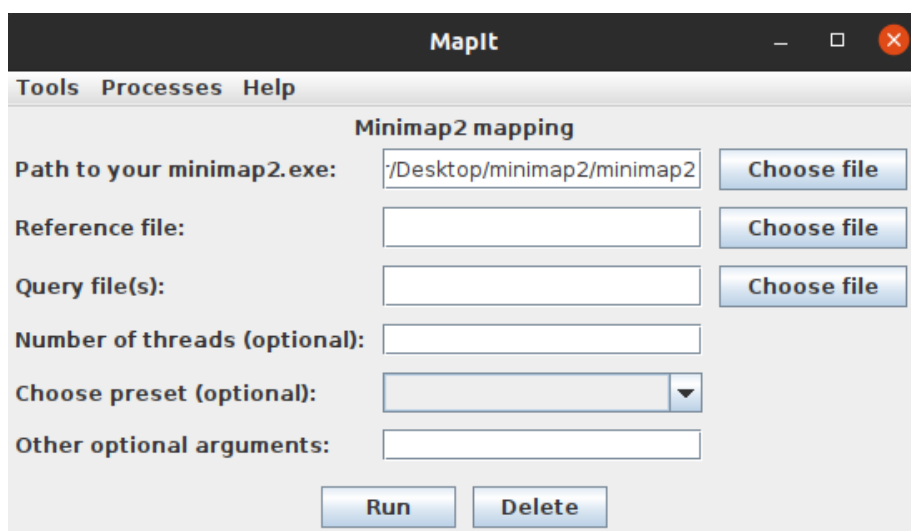
Pored polja za unos putanja do datoteka nalaze se gumbi koje korisniku otvaraju *FileChooser* sučelje kako bi mogao lakše locirati datoteke. Polje za unos dretvi nudi korisniku mogućnost da procesu dodjeli više dretvi kako bi se proces brže izveo. U polje za unos dodatnih parametara korisnik upisuje po želji dodatne parametre s kojima želi konfigurirati proces. Pritiskom gumba *Run* pokreće se proces s unesenim podacima, dok se pritiskom gumba *Delete* brišu svi uneseni podaci osim putanje do alata.



Slika 4.8: Prikaz sučelja *Minimap2 alignment*

4.4.2. Sučelje *Minimap2 mapping*

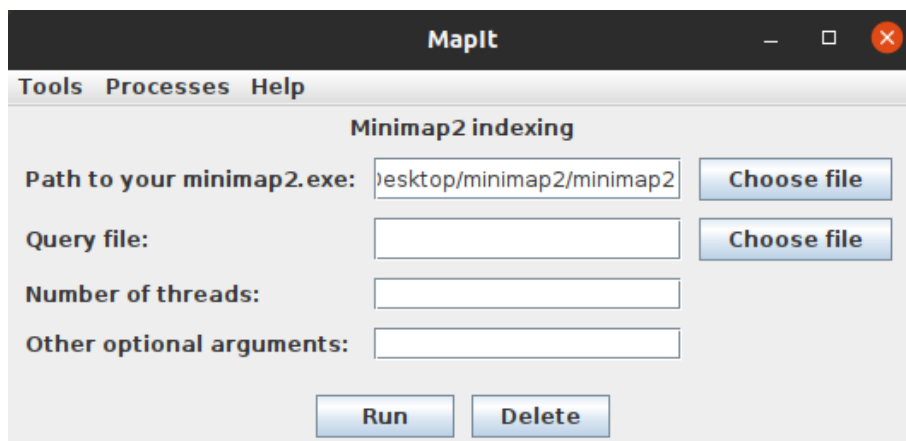
Sučelje *Minimap2 mapping* služi za unos podataka i pokretanje procesa mapiranja sekvenci na referentni genom pomoću alata *Minimap2*. Sadrži sva polja kao i sučelje *Minimap2 alignment* (Slika 4.9) te također prima FASTA i FASTQ datoteke. Razlika između ta dva sučelja je u tome što ovo sučelje ne nudi opciju stvaranja CIGAR stringa te nudi drugačije *presetove*. Ispisuje se ista putanja do alata kao i u sučelju *Minimap2 alignment* te su uloge gumbova *Run* i *Delete* također jednake.



Slika 4.9: Prikaz sučelja *Minimap2 mapping*

4.4.3. Sučelje *Minimap2 indexing*

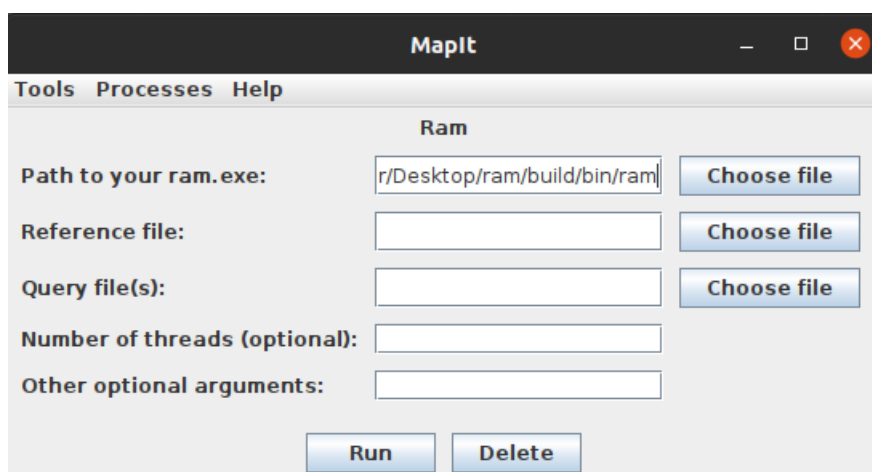
Sučelje *Minimap2 indexing* služi za unos podataka i pokretanje procesa indeksiranja sekvence, odnosno stvaranja njezinih *minimizera*. Rezultati tog procesa mogu se kasnije koristiti prilikom pokretanja procesa poravnanja pomoću alata *Minimap2* kako bi se brže izveo. Sučelje također sadrži polje koje sadrži putanju do alata *Minimap2*. Iza njega slijedi polje za unos putanje do FASTA datoteke, polje za unos broja dretvi i polje za unos dodatnih parametara (Slika 4.10).



Slika 4.10: Prikaz sučelja *Minimap2 indexing*

4.4.4. Sučelje *Ram*

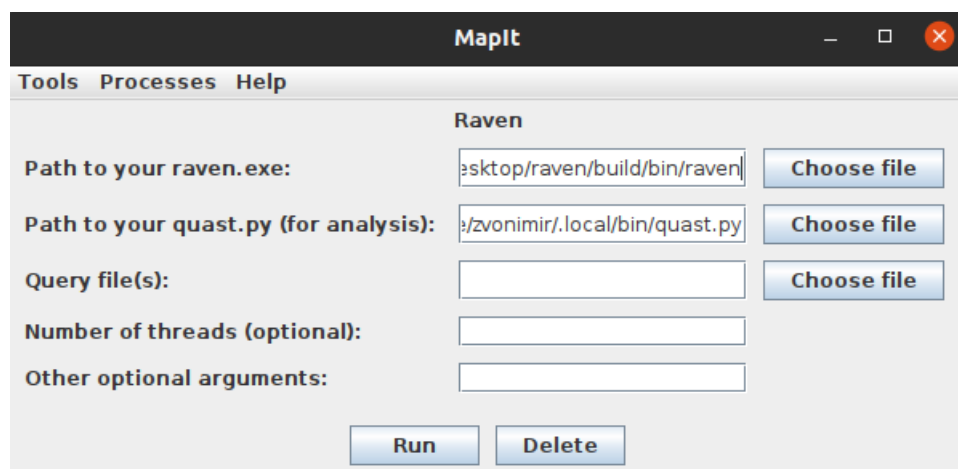
Sučelje *Ram* služi za unos podataka i pokretanje procesa mapiranja sekvenci na referentni genom. U prvom retku sučelja nalazi se polje za putanju do alata *Ram* koje funkcionira po istom principu kao polja za putanju do alata *Minimap2* u sučeljima za minimap2 procese. Iza njega slijede polja za unos putanje do FASTA ili FASTQ datoteke referentnog genoma i za unos putanje ili putanja do FASTA ili FASTQ datoteka koje sadrže sekvence koje želimo mapirati na referentni genom. Zatim slijedi polje za unos broja dretvi, te polje za unos dodatnih argumenata (Slika 4.11). Gumbi *Run* i *Delete* imaju istu funkcionalnost kao i u sučeljima za *Minimap2* procese.



Slika 4.11: Prikaz sučelja *Ram*

4.4.5. Sučelje *Raven*

Sučelje *Raven* također započinje s poljem koje sadrži putanju do alata *Raven*. Iza njega slijedi polje koje sadrži putanju do programa *Quast* koji je potreban za analizu rezultata procesa pokrenutih s *Ravenom*. Ispis putanja u oba polja funkcionira po istom principu kao i u ostalim sučeljima. Iza njih slijedi polje za unos putanje ili putanja do datoteka u FASTA ili FASTQ formatu koje sadrže sekvence za *de novo* sastavljanje genoma, te polje za unos broja dretvi i polje za unos dodatnih parametara (Slika 4.12). Gumb *Run* ima istu funkcionalnost kao i u ostalim sučeljima, dok gumb *Delete* briše vrijednosti svih polja osim onih koje sadrže putanje do alata *Raven* i *Quast*.



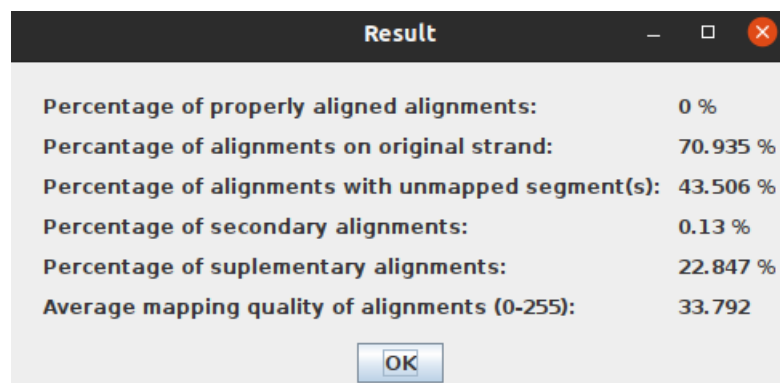
Slika 4.12: Prikaz sučelja *Raven*

5. Analiza rezultata

5.1. Analiza SAM datoteka

Datoteke u SAM formatu nastaju kao rezultat algoritma poravnanja između sekvenci u alatu *Minimap2*. Ukoliko se proces poravnanja pomoću alata *Minimap2* uspješno završi, korisniku se nudi mogućnost analize rezultata zapisanih u datoteci. Nakon što korisnik upiše identifikacijski broj uspješno završenog procesa poravnanja u prozor za upis (Slika 4.3), otvara mu se prozor koji sadrži analizu svih poravnanja ostvarenih između dvije sekvence (Slika 5.1). Prozor sadrži sljedeće podatke sa vrijednostima zaokruženim do treće decimalne:

- postotak poravnanja u kojem se sve baze usklađene
- postotak poravnanja na originalnom nizu
- postotak poravnanja sa nemapiranim segmentima
- postotak sekundarnih poravnanja
- postotak suplementarnih poravnanja
- prosječna kvaliteta mapiranja u svakom poravnanju



Result	
Percentage of properly aligned alignments:	0 %
Percentage of alignments on original strand:	70.935 %
Percentage of alignments with unmapped segment(s):	43.506 %
Percentage of secondary alignments:	0.13 %
Percentage of supplementary alignments:	22.847 %
Average mapping quality of alignments (0-255):	33.792
OK	

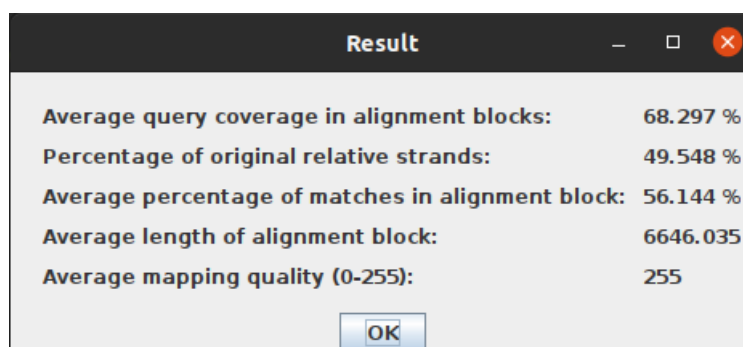
Slika 5.1: Primjer analize SAM datoteke dobivene pomoću alata *Minimap2*

5.2. Analiza PAF datoteka

Datoteke u PAF formatu nastaju kao rezultat mapiranja sekvenci na referentni genom pomoću alata *Minimap2* i *Ram*. Ukoliko se procesi mapiranja korisniku se nudi mogućnost analize rezultata datoteke koja sadrži rezultate. Analiza se pokreće na isti način kao i analiza

SAM datoteka. Prozor sa rezultatima analize sadrži sljedeće podatke sa vrijednostima zaokruženim do treće decimalne:

- prosječna pokrivenost sekvenci fragmenata
- postotak mapiranja na originalnom nizu
- prosječan postotak broja usklađenih baza u blokovima poravnanja
- prosječna duljina bloka poravnanja
- prosječna kvaliteta mapiranja



The screenshot shows a window titled 'Result' with a dark header bar. Inside, there is a table of statistics. At the bottom of the table is an 'OK' button.

Average query coverage in alignment blocks:	68.297 %
Percentage of original relative strands:	49.548 %
Average percentage of matches in alignment block:	56.144 %
Average length of alignment block:	6646.035
Average mapping quality (0-255):	255

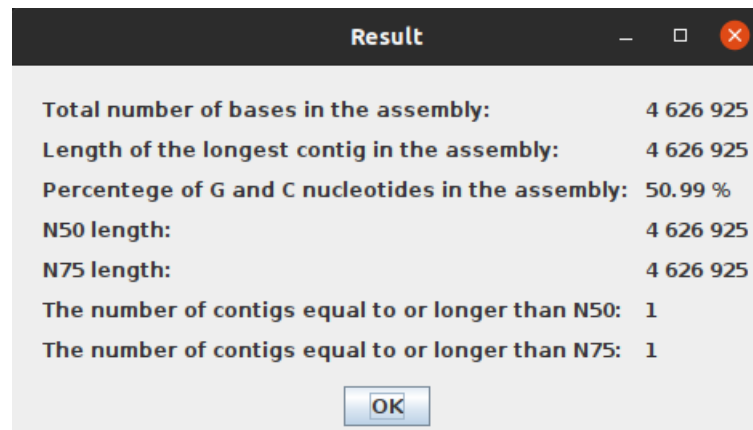
Slika 5.2: Primjer analize PAF datoteke dobivene pomoću alata ram

5.3. Analiza FASTA datoteka nastalih *de novo* sastavljanjem

Datoteke nastale uporabom alata *Raven* su FASTA datoteke koje sadrže kontinuirane sekvence dobivene *de novo* sastavljanjem. Zbog složenosti izračuna vrijednosti određenih parametara koje bi mogle zanimati korisnika (npr. N50 ili N75 duljina), analiza ovih datoteka izvodi se pomoću alata *Quast*. Ukoliko korisnik nema instaliran alat *Quast*, analiza ovih datoteka neće biti moguća. Nakon što korisnik upiše identifikacijski broj završenog procesa *de novo* sastavljanja čije rezultate želi analizirati, otvara mu se prozor sa sljedećim podacima:

- ukupan broj baza
- duljina najdulje sekvence u datoteci
- postotak nukleinskih baza gvanin i citozin u datoteci

- N50 duljina
- N75 duljina
- broj sekvenci u datoteci koje su dulje od N50 duljine
- broj sekvenci u datoteci koje su dulje od N75 duljine



Slika 5.3: Primjer analize FASTA datoteke dobivene *de novo* sastavljanjem

Zaključak

Svakodnevni korisnik računala nema naviku koristiti naredbenu liniju operacijskog sustava za pokretanje programa na svom računalu. Grafičko sučelje aplikacije olakšava korištenje alata i pokretanje njihovih procesa. Također je jako bitno da se ti procesi pokreću u pozadini operacijskog sustava kako ne bi ometali korisnika.

Procesi poravnanja, mapiranja i *de novo* sastavljanja su procesi izrazito velike vremenske i memorijske složenosti, stoga je prilikom izrade ove aplikacije bilo jako bitno da se procesi izvode neovisno o aplikaciji u pozadini sustava kako ne bi ometali korisnika. Također je bitno da je korisnik obavješten kad je proces uspješno završen ili prekinut. To je postignuto izradom pomoćnog programa koji ima zadaće pokretanja tih procesa u pozadini operacijskog sustava, te prikazivanja obavijesti korisniku kad je proces završen ili prekinut. Aplikacija također nudi korisniku jednostavnu analizu rezultata njegovih procesa kako bi korisnik mogao znati učinkovitost svojih procesa.

Literatura

- [1] Šikić, M., Domazet-Lošo, M. *Bioinformatika – skripta*, (2013., prosinac). Poveznica: [https://www.fer.unizg.hr/download/repository/bioinformatika_skripta_v1.2\[1\].pdf](https://www.fer.unizg.hr/download/repository/bioinformatika_skripta_v1.2[1].pdf); pristupljeno: 5. lipnja 2021.
- [2] Clausen, P *Finding the appropriate method, with a special focus on: Mapping and alignment*. Poveznica: https://www.eurl-ar.eu/CustomerData/Files/Folders/31-training-course-kgs-lyngby-2018/435_alignment-and-mapping-philip-clausen.pdf; pristupljeno: 5. lipnja 2021.
- [3] Roberts, M., Hayes, W., R. Hunt, B., M. Mount, S., A. Yorke, J. *Reducing storage requirements for biological sequence comparison*, *Bioinformatics*, 20, 18, (2004., prosinac), str. 3363-3369. Poveznica: <https://academic.oup.com/bioinformatics/article/20/18/3363/202143>; pristupljeno: 5. lipnja 2021.
- [4] Heng, Li *Minimap2: pair alignment for nucleotide sequences*, *Bioinformatics*, 34, 18, (2018., rujan), str. 3094–3100. Poveznica: <https://academic.oup.com/bioinformatics/article/34/18/3094/4994778>; pristupljeno: 6. lipnja 2021.
- [5] Vaser, R. *Ram*, GitHub (2021., ožujak). Poveznica: <https://github.com/lbcb-sci/ram>; pristupljeno: 6. lipnja 2021.
- [6] Vaser, R. *Raven*, GitHub (2021., lipanj). Poveznica: <https://github.com/lbcb-sci/raven>; pristupljeno: 6. lipnja 2021.
- [7] *Blast topics*, U.S. National Library of Medicine. Poveznica: https://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE_TYPE=BlastDocs&DOC_TYPE=BlastHelp; pristupljeno: 6. lipnja 2021.
- [8] *FASTA format*, Wikipedia (2021. svibanj). Poveznica: https://en.wikipedia.org/wiki/FASTA_format; pristupljeno: 6. lipnja 2021.
- [9] *FASTQ files explained*, Illumina (2020., listopad). Poveznica: <https://support.illumina.com/bulletins/2016/04/fastq-files-explained.html>; pristupljeno: 6. lipnja 2021.
- [10] *FASTQ format*, Wikipedia (2021., svibanj). Poveznica: https://en.wikipedia.org/wiki/FASTQ_format; pristupljeno: 6. lipnja 2021.
- [11] *FASTQ Files*, BaseSpace SEQUENCE HUB. Poveznica: <https://help.basespace.illumina.com/articles/descriptive/fastq-files/>; pristupljeno: 6. lipnja 2021.
- [12] *Sequence Alignment/Map Format Specification*, GitHub (2021., lipanj). Poveznica: <https://samtools.github.io/hts-specs/SAMv1.pdf>; pristupljeno: 6. lipnja 2021.
- [13] Heng, L. *PAF: a Pairwise Mapping Format*, GitHub (2020., srpanj). Poveznica: <https://github.com/lh3/miniasm/blob/master/PAF.md>; pristupljeno: 6. lipnja 2021.

Sažetak

Cilj ovog rada je bio izrada desktop aplikacije s grafičkim sučeljem za pokretanje bioinformatičkih procesa pomoću vanjskih alata te analiza rezultata tih procesa. Vanjski alati koji su korišteni u ovom radu su: *Minimap2*, *Ram* i *Raven*. Problem s navedenim alatima je što se pokreću preko naredbene linije, a procesi koje izvode mogu trajati danima. Aplikacija nudi korisniku mogućnost pokretanja navedenih alata preko grafičkih sučelja dizajniranih za svaku vrstu procesa koje korisnik želi pokrenuti. Proces se pokreću neovisno o aplikaciji u pozadini računalnog sustava tako da aplikacija ne mora biti upaljena za vrijeme trajanja procesa. Korisnik može u bilo kojem trenutku provjeriti stanje svojih procesa, te je obavješten ukoliko je proces uspješno završen ili prekinut u izvođenju. Ukoliko je proces uspješno završen, korisnik može analizirati rezultate procesa.

Ključne riječi: bioinformatika, mapiranje, de novo, poravnanje, Minimap2, Raven, Ram, analiza, Java, Swing

Summary

The goal of this thesis was to develop a desktop application with graphical interface for running bioinformatic processes with the help of the outside tools and analyzing the results of those processes. Outside tools which are being used in this thesis are: *Minimap2*, *Ram* and *Raven*. Problem with these tools is that they can only be run through systems command line and the processes which they perform can last couple of days. Application offers the user an ability to run those processes by using its graphical interfaces specially designed for each type of process. Processes are being run independently from application in the background of computers operating system which means that it is not necessary that the application is turned on during the whole process. User can check statuses of his/her processes at any moment. The user will also be informed when the process is finished or when it comes to an unexpected halt. If the process is successfully finished, the user can run the analysis of its result.

Key words: bioinformatics, mapping, de novo, alignment, Minimap2, Raven, Ram, analysis, Java, Swing