Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Jasmin Avdić

Razvoj spletne aplikacije za zaznavanje variabilnosti v zaporedju DNA

diplomsko delo

visokošolski strokovni študijski program
prve stopnje
računalništvo in informatika

Mentor: viš. pred. dr. Igor Rožanc
Somentor: prof. dr. Kristina Gruden

Ljubljana, 2018

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V diplomskem delu predstavite razvoj spletne aplikacije za zaznavanje variabilnosti v zaporedju DNA. V ta namen najprej kratko predstavite ozadje na področju raziskav DNA in specifične zahteve za zaznavo variacij DNA z različnimi odprtokodnimi orodji. Čelni del aplikacije naj omogoča izbiro več izbranih orodij in nastavitev ustreznih parametrov, medtem ko zaledni del nadzorovano izvede predvideni postopek in posreduje rezultate. Aplikacijo praktično preverite v realnem okolju.

Zahvaljujem se vsem, ki so kakorkoli pomagali pri izdelavi diplomske naloge, še posebej viš. pred. dr. Igorju Rožancu, prof. dr. Kristini Gruden in mladi raziskovalki Maji Zagorščak za strokovno pomoč.

Zahvala gre tudi moji družini, prijateljem in kolegom, ki so mi stali ob strani ter me podpirali na študijski poti.

Kazalo

Seznam uporabljenih kratic

kratica	angleško	slovensko
BAI	Binary Alignment Index	Indeks binarne poravnave
BAM	Binary Alignment Map	Binarna mapa poravnave
BWA	Burrows-Wheeler aligner	Burrows–Wheelerjev poravnalnik
BWT	Burrows–Wheeler Transform	Burrows–Wheelerjeva preslikava
CPU	Central processing unit	Osrednja procesna enota
CSS	Cascading Style Sheets	Kaskadne stilske predloge
DNA	Deoxyribonucleic acid	Deoksiribonukleinska kislina
GATK	Genome Analysis Toolkit	Orodja za analizo genoma
GFF	General Feature Format	Format splošnih značilk
HTML	Hyper Text Markup Language	Označevalni jezik za spletne strani
IGV	Integrative Genomics Viewer	Integrativno orodje za genomsko vizualizacijo
Maser	Management and Analysis System for Enormous Reads	Sistem za upravljanje in analizo ogromne količine odčitkov zaporedij
MNP	Multiple Nucleotide Polymorphisms	Večnukleotidni polimorfizem
NFS	Network File System	Mrežni datotečni sistem

NGS	Next Generation Sequencing	Sekvenciranje naslednje generacije
NIB	National institute of biology	Nacionalni inštitut za biologijo
NK	Nucleic acids (e.g. DNA, RNA)	Nukleinska kislina (npr. DNA, RNA)
PCR	Polymerase chain reaction	Verižna reakcija s polimerazo
PHP	Hypertext Preprocessor	Programski jezik *PHP*
PNG	Portable Network Graphics	Rastrski slikovni format
RNA	Ribonucleic acid	Ribonukleinska kislina
SAM	Sequence Alignment Map	Mapa poravnave zaporedij
SNP	Single-nucleotide polymorphism	Eno-nukleotidni polimorfizem
STAR	Spliced Transcripts Alignment to a Reference	Razdeljena poravnava transkriptov na referenčno zaporedje
VCF	Variant Call Format	Format zaznanih variacij

Razlaga bioinformatskih izrazov

izraz	pojasnilo
BAI	Indeksiran `BAM`.
BAM	Binarna različica `SAM`.
bazni par (bp)	Par nukleotidnih baz na sosednjih, komplementarnih verigah DNA ali RNA.
BCFtools/HTSlib/ SAMtools	V nadaljevanju uporabljen sklic *SAMtools*, skupina orodij za zaznavanje variacij v DNA in upravljanje s formati `VCF` in `BCF` [26].
Bowtie 2	Orodje za gensko poravnavo nizov na referenčno zaporedje. Najbolj se izkaže pri poravnavi 50 – 1000 znakov dolgih nizov na referenčno zaporedje. Tega indeksira z FM indeksom (temelji na BWT) in tako zmanjša porabo pomnilnika [22].

BWA	Programski paket za poravnavo nizov z majhno raznolikostjo na velika referenčna zaporedja, kot je npr. človeški genom. Vsebuje tri algoritme, *BWA-backtrack* (uporaben za nize do 100 bp), *BWA-SW* (uporaben za nize 70 – 1M bp) in najnovejši *BWA-MEM* (uporaben za nize 70 – 1M bp, ki zagotavlja boljšo kakovost in je bolj natančen; uporabljen v naši skripti). Tako kot *Bowtie 2*, referenčno zaporedje indeksira z FM indeksom [23].
Delecija	Skrajšanje zaporedja za enega ali nekaj nukleotidov glede na referenčno zaporedje.
DNA	Zaporedje sestavljeno iz znakov A, C, G in T, nosilka genetkse informacije.
Dry lab	Laboratorij za analizo v *wet lab-u* pridobljenih bioloških podatkov z bioinformatskimi orodji.
FASTA	Tekstovni format za predstavitev nukleotidnih (nt) ali aminokislinskih (aa) zaporedij (1 nt ali 1 aa = 1 črka). Iz njega razvit tudi *FASTQ (fq)* format, ki poleg zaporedja vsebuje še njegovo kakovost [21].
FreeBayes	Programsko orodje za zaznavo variacij v genomu, narejeno tako, da najde majhne polimorfizme zlasti *SNP-jev*, *indelov*, *MNP-jev* in kompleksnih dogodkov (kompozicijskih insercij ter substitucijskih dogodkov) [11].
GATK	Zbirka orodij za analizo visokorazsežnih sekvenčnih podatkov z glavnim poudarkom na odkrivanju variacij [17].
Genom	Zaporedje DNA zapisano v kromosomih organizma.

Genska poravnava (ang. gene mapping)	Proces, s katerim identificiramo in določamo absolutni oziroma relativni položaj in vrstni red genskega mesta na referenčnem zaporedju (npr. kromosomu), ki ga zavzame zaporedje. Poravnava niza na referenčno zaporedje [18].
GFF	Format, uporabljen za opis genov in ostalih atributov DNA, RNA ter proteinskih zaporedij glede na genom [30].
IGV	Orodje za vizualizacijo in interaktivno raziskovanje velikih genomskih podatkovnih nizov [13].
Indel	Kratke insercije in delecije.
Insercija	Podaljšanje zaporedja za enega ali nekaj nukleotidov glede na referenčno zaporedje [**?**].
NGS	Visokorazsežna metoda za dekodiranje zaporedja nukleotidov z veliko hitrostjo po relativno nizki ceni [27].
Nukleotid (nt)	Osnovni gradnik DNA in RNA, ki se povezuje v nukleotidno verigo.
Oligonukleotidni začetnik	Krajša veriga DNA ali RNA ki je potrebna za začetek podvojevanja DNA [8].
PCR	Metoda molekularne biologije, s katero pomnožimo točno določen del DNA. Je enostavna in hitra metoda za številno podvojitev delov DNA. Metodo je leta 1983 razvil ameriški biokemik Kary Mullis [4].
Picard	Skupina orodij v *Javi* za manipulacijo z visokorazsežnimi sekvenčnimi podatki in formati kot so `SAM, BAM, VCF` [15].
Polimorfizem	Različne variacije specifičnih značilnosti v isti populaciji.

SAM	Tekstovni format za zapis nizov poravnanih z referenčnim zaporedjem; format sta razvila Heng Li in Bob Handsaker [24].
Sekvenciranje	Določanje zaporedja nukleotidov (A, C, T, G) na verigah NK.
SNP	Je eno-nukleotidni polimorfizem, ki označuje spremembo v enem samem nukleotidu, ki se pojavi na določenem položaju v referenčnem zaporedju, pri čemer je vsaka sprememba prisotna do neke občutne stopnje znotraj populacije [6].
STAR	*C++* orodje za poravnavo, ki temelji na dveh korakih: iskanje semena (ang. *seed searching step*) in kreiranje gruč (ang. *clustering/stitching/scoring step*) [9].
Substitucija	Zamenjava enega ali nekaj nukleotidov glede na referenčno zaporedje.
Tablet	Visokozmogljivo orodje za vizualizacijo *NGS* podatkov [19].
Transkriptom	Množica izraženih genov ob določenem pogoju.
VCF	Tekstovni format, ki se uporablja za zapis variacij v zaporedjih.
Wet lab	Laboratorij, ki omogoča izvajanje praktičnih znanstvenih raziskav.

Povzetek

Naslov: Razvoj spletne aplikacije za zaznavanje variabilnosti v zaporedju DNA

Avtor: Jasmin Avdić

V diplomski nalogi je predstavljen postopek izdelave spletne aplikacije za zaznavanje variabilnosti v zaporedju DNA z imenom *Amplicode 2018*. DNA je dvoverižno zaporedje, ki je zgrajeno iz štirih nukleotidov: adenina (A), citozina (C), gvanina (G) in timina (T). Tako kot so organizmi različni med seboj navzven se ti razlikujejo tudi v genetskem zapisu. Različne variacije zaporedij v genu določajo različne lastnosti; npr. pri človeku višino, barvo las, oči, naklonjenost za različne bolezni, ...

Spletna aplikacija *Amplicode 2018* omogoča zaznavo takšnih variacij v izbranem genu z različnimi odprtokodnimi orodji. Celoten postopek izbire orodij in potrebnih nastavitev se odvija na uporabniku prijazni spletni strani. To lahko uporabljajo tudi manj izkušeni poznavalci operacijskega sistema *Linux* ter vseh ukazov in stikal, ki jih odprtokodna orodja ponujajo. Končni rezultat je vedno iste oblike, ne glede na različne vmesne korake, kar daje uporabniku možnost analize kakovsti zaznave variacij glede na različna orodja in pomaga pri odločitvah za nadaljnje *wet lab* korake.

Ključne besede: Bash, DNA, DNA variacije, PHP, Python, poravnava, nukleotid, spletna aplikacija, zaporedje.

Abstract

Title: The development of web application for variability detection in the DNA sequence

Author: Jasmin Avdić

This diploma thesis presents a development of web application for variability detection in the DNA sequence called *Amplicode 2018*. DNA consists of four nucleotides adenine (A), cytosine (C), guanine (G) and thymine (T) which are interconnected into sequence. The phenotype level differences are related to genetic code at the genotype level. Sequences with different variations can determine particular attributes, for example height, hair colour, pigmentation of the iris, predispositions to specific diseases, etc.

Web application *Amplicode 2018* enables detection of sequence variations within a gene of choice using different open source tools. The entire tool and parameter selection is carried out using the user-friendly website, that can be used by users with limited knowledge of *Linux* operating system, commands and switches of open source tools. The end result is consistently provided in the same format, regardless the various intermediate steps. It offers the ability to analyse the quality of the detection of variations with respect to different tools and helps user in further *wet lab* decisions.

Keywords: alignment, Bash, DNA, DNA variations, PHP, Python, nucleotide, sequence, web application.

Poglavje 1

Uvod

Področje genetike je v zadnjih 150 letih naredilo velike premike. Od enostavnega križanja rastlin, ki ga je v 60. letih 19. stoletja izvajal Gregor Mendel in kloniranja ovce Dolly leta 1996, smo sedaj prišli do stopnje, ko nam visokorazsežne tehnologije z uporabo računalniške infrastrukture omogočajo hitro in relativno cenovno ugodno določanje ter analizo genetskih zaporedij. Te lahko uporabimo za doslej nepredstavljive postopke npr. personalizirano zdravljenje bolnikov z rakom. Vse hitrejši razvoj sodobnih molekularnih metod je pripeljal do generiranja ogromnih množic podatkov, ki jih je potrebno analizirati. To je privedlo do potrebe po povezovanju računalniškega znanja o algoritmih in podatkovnih strukturah na eni strani ter biološkega znanja o prvinah genomike na drugi strani. Rezultat tega je pojav različnih orodij za analizo bioloških zaporedij. Večina takšnih orodij (npr. *CLC Genomics* [31]), je enostavnih za uporabo, saj deluje po principu črne škatle. Uporabniku notranja zgradba in delovanje tako ni dostopno. Ta orodja so velikokrat plačljiva, konkurirajo pa jim odprtokodna orodja, katerih uporaba je pogosto zapletena. Potrebujemo vsaj osnovno razumevanje programskih jezikov, operacijskega sistema *Linux* in poznavanje ukazov ter stikal, ki jih ta orodja uporabljajo.

1.1 Motivacija

Na NIB-u se je pojavila potreba po aplikaciji za zaznavo variabilnosti v zaporedju DNA, ki bi bila enostavna za uporabo in ponujala uporabniku možnost izbire najprimernejšega orodja izmed palete odprtokodnih orodij za analizo *NGS* podatkov. Eden izmed glavnih navdihov za izdelavo takšne aplikacije je bil članek *Single Nucleotide Polymorphism Identification in Polyploids: A Review, Example, and Recommendations* [7], v katerem so med seboj primerjali orodja in njihovo učinkovitost pri zaznavanju variacij.

Takšna aplikacija je potrebna tudi zaradi praktičnih težav, s katerimi se srečujejo raziskovalci. Veliko nepotrebnega časa je bilo tako recimo izgubljeno za iskanje stabilnih in kompatibilnih različic odprtokodnih orodij.

1.2 Cilji

Glavni cilj diplomske naloge je izdelava spletne aplikacije za zaznavanje variabilnosti v zaporedju DNA z imenom *Amplicode 2018*. Aplikacija naj bo enostavna za uporabo, ponuja naj možnost izbire različnih orodij za poravnavo nizov na izbrano referenčno zaporedje (recimo *Bowtie 2, BWA, STAR*) in za zaznavo variacij na zaporedju (recimo *SAMtools, FreeBayes, GATK*). Prav tako naj bo primerno hitra in naj po možnosti večina orodij uporablja paralelizacijo. Strojna oprema (2 procesorja `Intel(R) Xeon(R) CPU E5645 @ 2.40GHz` in 142 GB delovnega pomnilnika) na NIB-ovem strežniku `Ibis` to omogoča. Rešitev mora biti tudi učinkovita, format končnega izpisa pa mora biti enak ne glede na izbrana orodja. Potrebno je zagotoviti tudi čim širšo dosegljivost raziskovalcem na NIB-u, zato je postavljena v obliki spletne aplikacije, ki je dosegljiva na internem omrežju NIB.

1.3 Sorodna dela

S podobno tematiko so se ukvarjali že mnogi. V nadaljevanju je našteto nekaj najbolj podobnih del:

- *A study on fast calling variants from next-generation sequencing data using decision tree* [25] prikazuje razvoj orodja, ki naj bi bilo hitro in natančno pri zaznavanju variacij z uporabo algoritma, ki temelji na odločitvenem drevesu.

- *Maser: one-stop platform for NGS big data from analysis to visualization* [20] opisuje razvoj platfrorme za upravljanje, analizo in vizualizacijo ogromnih zaporedij v oblaku.

- *Single Nucleotide Polymorphism Identification in Polyploids: A Review, Example, and Recommendations* [7] je prikaz primerjave različnih orodij za zaznavanje variacij.

1.4 Struktura diplomskega dela

V naslednjih poglavjih se bomo seznanili s:

- **problemsko domeno v poglavju 2.** V tem poglavju so podrobnejše opisane zahteve, ki naj jih aplikacija zadovolji ter analizirana različna orodja za poravnavo.

- **opisom rešitve v poglavju 3.** Tu predstavimo tehnični opis rešitve, ki je razdeljena na čelni (ang. *front-end*) in zaledni (ang. *back-end*) del.

- **sklepnimi ugotovitvami v poglavju 4.** To govori o prispevkih diplomske naloge, težavah pri izdelavi, uporabniški izkušnji in možnih izboljšavah.

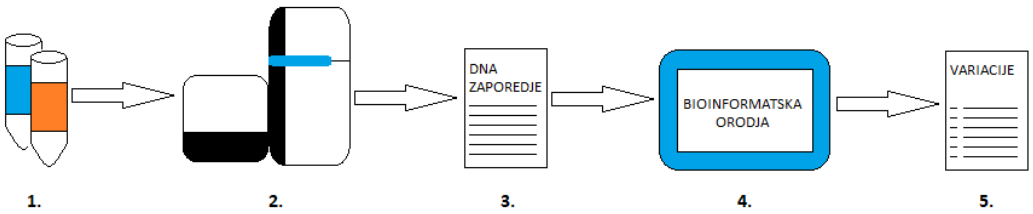
Poglavje 2

Problemska domena

Pred izdelavo aplikacije za zaznavanje variabilnosti v zaporedju DNA *Amplicode 2018* je bilo potrebno določiti vhodne podatke, načrtovati izgled aplikacije, opredeliti njene funkcionalnosti ter se seznaniti z odprtokodnimi *NGS* orodji, ki jih bo aplikacija uporabljala.

2.1 Zajem NGS podatkov

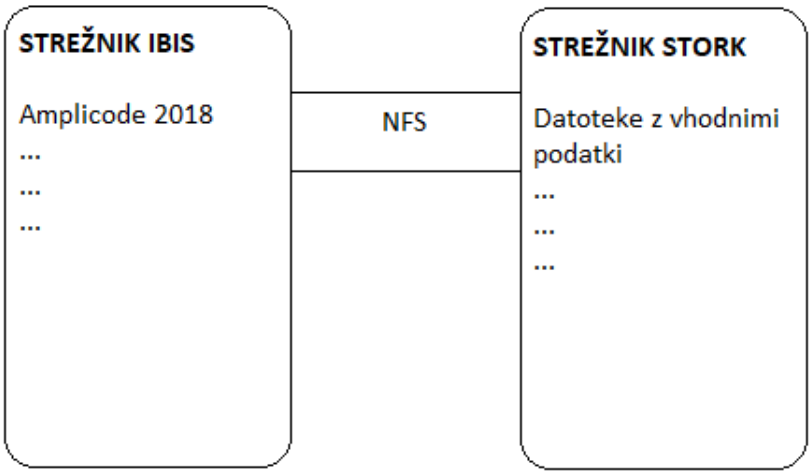
NGS podatke zajamemo s pomočjo analizatorja genoma (npr. *Ilumina HiSeq*), ki nam določi zaporedje nukleotidov v podanem vzorcu (1. in 2. korak na sliki 2.1). Tako dobimo vhodne podatke naše aplikacije, s pomočjo katerih nato z bioinformatskimi orodji (4. korak na sliki 2.1) ugotavljamo, na katerih pozicijah (5. korak na sliki 2.1) naj bi se pojavile variacije.



Slika 2.1: Postopek pridobivanja *NGS* podatkov [16].

Aplikacija *Amplicode 2018* se ukvarja s 3., 4. in 5. korakom na sliki 2.1, prva dva koraka pa sta prepuščena *wet lab* raziskovalcem.

Priporočljivo je, da so zaporedja, ki se uprabljajo kot vhodni podatki shranjena na NIB-ovem strežniku `Stork`. Ta si s pomočjo protokola *NFS* deli datoteke z NIB-ovim strežnikom `Ibis`, na katerem teče naša aplikacija kot je vidno na sliki 2.2.

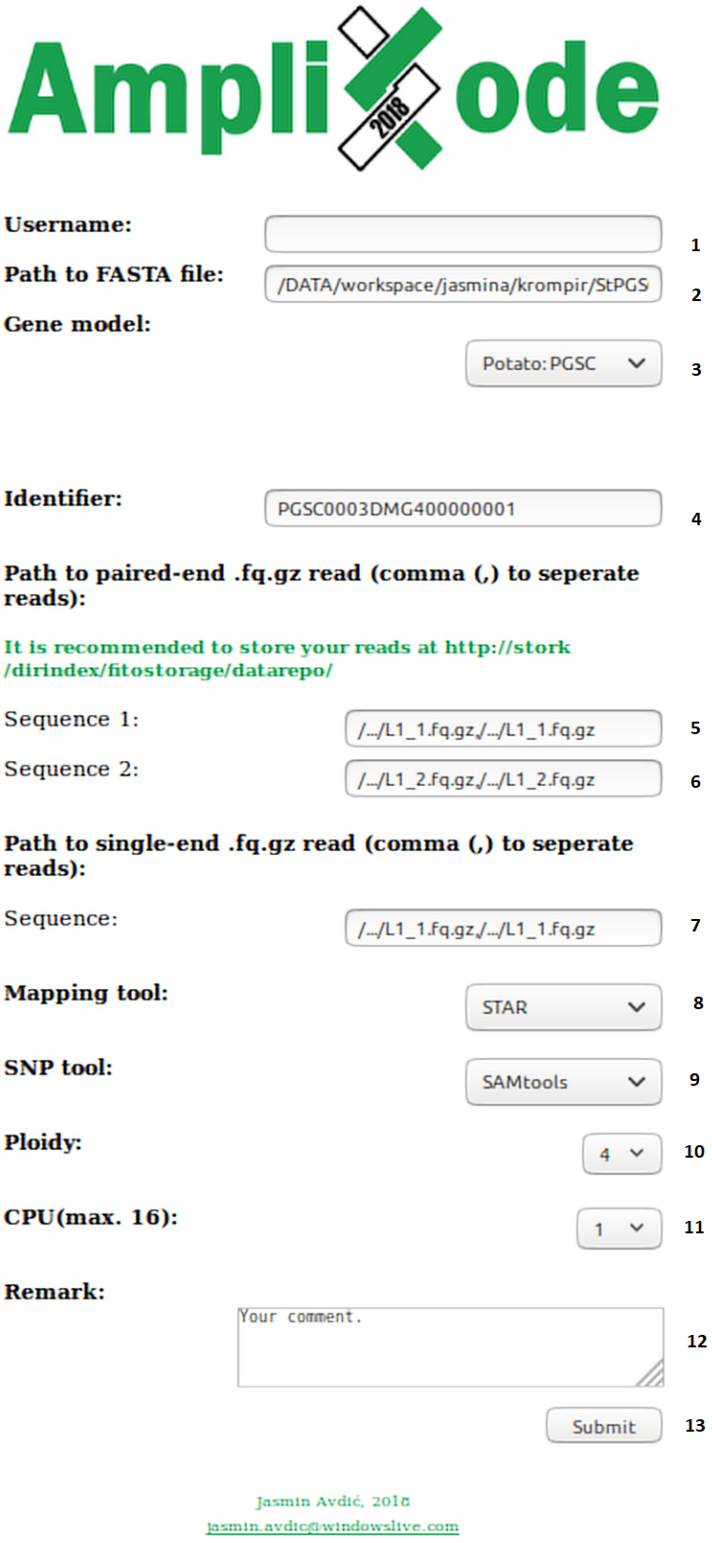


Slika 2.2: Strežnika `Ibis` in `Stork`.

Zaradi velikosti zaporedij (tudi več kot 50 GB) bi bilo vsakokratno nalaganje le-teh na strežnik časovno potratno. Prav tako na ta način zagotovimo tudi nadzor kakovosti podatkov in omogočimo njihovo ponovno uporabo ter tako sledimo dobri praksi upravljanja s podatki [32].

2.2 Zahteve

Aplikacija *Amplicode 2018* je bila načrtovana v skladu z zahtevami na NIB. Najprej sta bili definirani zaslonski maski (vnosne strani in strani med izvajanjem, rezultat viden na slikah 2.3 in 2.4), ki sta določali vizualni izgled aplikacije. Ta se je sicer zaradi dodatnih parametrov še nekajkrat spremenil, a je bila določitev zaslonskih mask dobra osnova za oblikovanje aplikacije.



Slika 2.3: Vnosna stran izdelana po zaslonski maski.

Slika 2.4: Stran med izvajanjem izdelana po zaslonski maski.

Aplikacija deluje tako da:

- Uporabniku najprej ponudi vnos uporabniškega imena, na podlagi katerega mu na koncu pošlje pot do datoteke z rezultati na elektronsko pošto (oznaka 1 na sliki 2.3).
- Nato uporabnik poda pot do referenčnega `FASTA` zaporedja. Referenčno zaporedje je lahko *genom* ali *transkriptom* (oznaka 2 na sliki 2.3).
- Sledi izbira genskega modela (`GFF` datoteke). Za izbiro mu aplikacija ponudi prednaložene modele za krompir in paradižnik (oznaka 3 na sliki 2.3). Obstaja tudi možnost, da uporabnik poda pot do svojega genskega modela. Tudi v tem primeru je priporočeno, da je model shranjen na NIB-ovem `Stork` strežniku.
- Uporabniku nato aplikacija ponudi še vpis genskega identifikatorja, na podlagi katerega dobimo pozicije za izločitev dela referenčnega `FASTA`

zaporedja, ki nas zanima (oznaka 4 na sliki 2.3).

- Sledi določitev poravnave. Aplikacija mora biti zasnovana tako, da omogoča različne oblike poravnave:
  - *paired-end* - poravnava dveh povezanih nizov na izločeno zaporedje, vmes je regija natančno definirane dolžine. Uporabnik poda pot do zapakirane datoteke z nizi (oznaki 5 in 6 na sliki 2.3),
  - *single-end* - poravnava enega niza na izločeno zaporedje, uporabnik poda pot do zapakirane datoteke z nizom (oznaka 7 na sliki 2.3),
  - *paired-end in single-end* - poravnava pri čemer uporabniku ne bo treba posebej zaganjati aplikacije (za *paired-end* in *single-end* poravnavo). Preprosto se vnese podatke za obe poravnavi hkrati in aplikacija jih ustrezno obdela, uporabnik poda pot do zapakirane datoteke z nizi (oznake 5, 6 in 7 na sliki 2.3).
- Zatem določitev orodij, ki jih bo aplikacija uporabila za poravnavo (ang. *mapping tool*). Imamo tri možnosti (oznaka 8 na sliki 2.3), ki so podrobneje opisane v razlagi bioinformatskih izrazov:
  - *Bowtie 2* je zasnovan na *BWT*. Priporočen je za poravnavo nizov dolgih od 50 do 1000 znakov na referenčno zapredje.
  - *BWA* je tudi zasnovan na *BWT*, a uporablja najnovejši *BWA-MEM* algoritem. Zato je priporočen za daljša zaporedja od 70 do 1M bp, zagotavlja tudi boljšo kakovost in je bolj natančen.
  - *STAR*, temelji na iskanju semen (ang. *seed search*) in gručenja (ang. *clustering*).

  Pri poravnavi ujemajočih *paired-end* nizov z *Bowtie 2* naj se uporabniku ponudi še izbira orientacije na referenčno zaporedje, saj je tako priporočeno v priročniku za *Bowtie 2* [22].

- Za zaznavo variacij na zaporedju (*SNP tool*) smo izbrali naslednja orodja (oznaka 9 na sliki 2.3):
  - *SAMtools* [24],
  - *GATK* [17],
  - *FreeBayes* [11].

  Zaradi priporočil [14] je orodje za zaznavo variacij *GATK* povezano z orodjem za poravnavo *BWA* in ga ni mogoče uporabiti z drugimi orodji za poravnavo.

  Navedena orodja lahko zaradi različne statistične analize v ozadju, dajejo različne rezultate ob istih vhodnih podatkih. Na sliki 2.5 vidimo razlike pri zaznavi *SNP-jev*. Orodje *GATK* je zaznalo le 25547 *SNP-jev*, medtem ko je *SAMtools* zaznal 328597 *SNP-jev*. Le 5143 *SNP-jev*, so enako zaznala vsa orodja, kar je samo od 1.4 % vseh zaznanih *SNP-jev* [7].

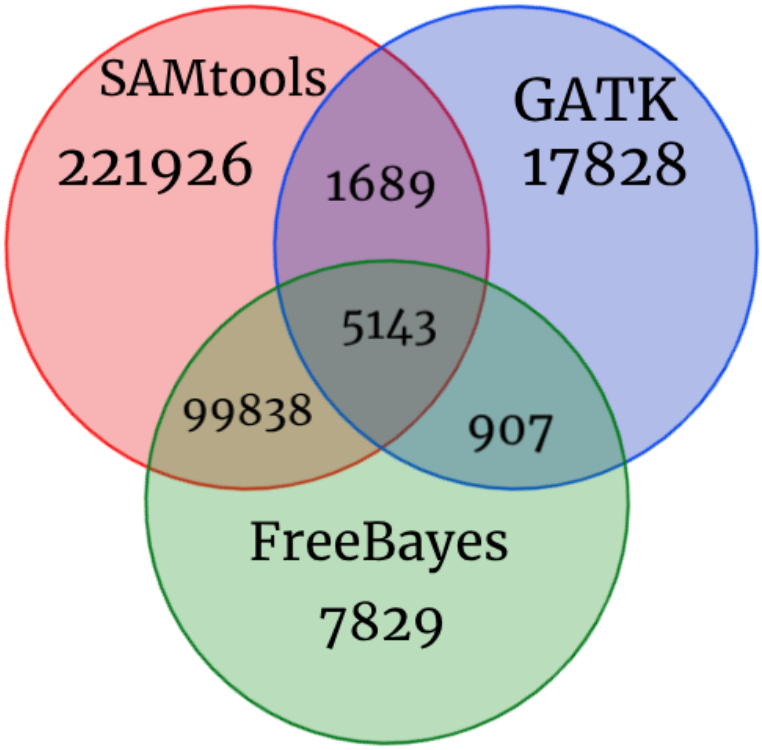
  Poleg tega naj aplikacija ne glede na prej izbrana orodja uporablja še orodje *Picard* za pretvorbo med formati, urejanje, dodajanje skupin (popravki glav, `ID-jev` nizov) itn.

- Naslednja zahteva je izbor `ploidnosti` glede na organizem, katerga DNA se uporablja (oznaka 10 na sliki 2.3). Na primer ljudje smo diploidni (torej je `ploidnost` 2), krompir ima `ploidnost` 4, itn.

- Ker je izvajanje določenih delov aplikacije lahko vzporedno, uporabnik opredeli še število niti, ki jih želi uporabiti za delovanje aplikacije (oznaka 11 na sliki 2.3).

- Zadnji vhodni podatek je opomba oziroma komentar ob izvajanju (oznaka 12 na sliki 2.3).

- Ko uporabnik s klikom na gumb posreduje vnesene podatke (oznaka 13 na sliki 2.3), se mu pokaže stran za spremljanje izvajanja izbranega postopka korak za korakom.



Slika 2.5: Primerjava rezultatov orodij *SAMtools, FreeBayes* in *GATK* [7].

- Na tej strani ima možnost popolne ustavitve aplikacije v primeru kakršnihkoli težav (oznaka 14 na sliki 2.4).

Ko aplikacija konča z izvajanjem, se uporabniku na strani izpiše pot do imenika, v katerem so shranjeni rezultati. Istočasno se uporabniku pošlje obvestilo s to potjo po elektronski pošti.

Ker želimo s prostorom na strežniku `Ibis` ravnati varčno, je imenik z rezultati dostopen le 7 dni, nato pa se samodejno izbriše.

Zaradi omejenih resursov in ostalih uporabnikov je na strežniku `Ibis` dovoljeno izvajanje samo ene aplikacije *Amplicode 2018* naenkrat.

Poglavje 3

Opis rešitve

Z znanimi zahtevami in po kratkem pregledu biološkega ozadja je čas, da izberemo ustrezna računalniška orodja in zasnujemo spletno aplikacijo *Amplicode 2018*.

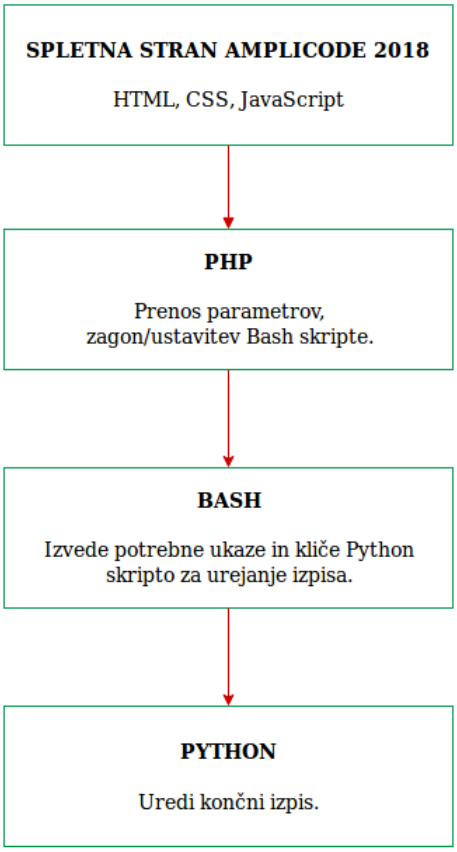
3.1 Splošni načrt

Pred izdelavo smo določili kratek načrt izdelave celotne aplikacije, ki je predstavljen na sliki 3.1. Okvirno smo določili tudi orodja, ki jih bomo uporabljali. Za izdelavo čelnega dela bomo uporabili *HTML, CSS* in *JavaScript*, pri zalednem delu pa smo si pomagali s *PHP*, *Bash* in *Python* skriptami.

3.2 Čelni del rešitve

Podobo spletne strani smo glede na prej narisane zaslonske maske oblikovali s pomočjo *HTML-ja*, *CSS-ja* in *JavaScript-a*. Rešitev je vidna na slikah 2.3 in 2.4. *Amplicode 2018* oznaka je narejena kot *PNG* slika, ki smo jo oblikovali v *Photoshop-u CC 2015* [2]. Klik na oznanko nas vrne na začetno stran.

Sledijo vnosna polja za uporabniško ime (`Username`), pot do `FASTA` datotek (`Path to FASTA file`), genski identifikator (`Identifier - ID`), pot do *paired-end* zaporedij (`Sequence1, Sequence2`), pot do *single-end* zaporedij



Slika 3.1: Načrt izdelave.

(`Sequence`) ter komentar (`Remark`).

Vmes so izbirna polja za izbiro genskega modela (`Gene model`), orodja za poravnavo (`Mapping tool`), orodja za iskanje variacij (`SNP tool`), `ploidnost` (`Ploidy`) ter število niti (`CPU`). Število niti je kljub največjemu številu 24, ki ga ponujata 2 procesorja (`Intel(R) Xeon(R) CPU E5645 @ 2.40GHz`), omejeno na 16, saj nismo edini uporabniki strežnika.

V primeru, ko uporabnik pri izbirnem polju izbere `My own GFF model`, je s pomočjo *JavaScript* funkcije prikazano dodatano vnosno polje, kjer lahko uporabnik poda pot do svojega modela (`/.../PathTo/myModel.gff3`). Kot vidimo v priloženi kodi se glede na to izbiro spreminja tudi nastavljena vrednost genskega identifikatorja (`Identifier - ID`), ki jo seveda uporabnik

lahko spremeni in je tu le za zgled. Prav tako se po potrebi pokaže opozorilo (ang. *warning*), da naj bo podani model shranjen na NIB-ovem `Stork` strežniku.

```
function showInput(izbira){
    if (izbira.value == 1) {
        // izbran je prvi izmed modelov za krompir
        document.getElementById('myinput').style.display = "none";
        document.getElementById('id1').value = "PGSC0003DMG400000001";
        document.getElementById('warning').style.display = "none";
    }
    else if (izbira.value == 2) {
        // izbran je drugi izmed modelov za krompir
        document.getElementById('myinput').style.display = "none";
        document.getElementById('id1').value = "Sotub01g005000.1.1";
        document.getElementById('warning').style.display = "none";
    }
    else if (izbira.value == 3) {
        // pokazemo uporabniku dodatno vnosno polje za njegov model
        document.getElementById('myinput').style.display = "block";
        document.getElementById('id1').value = "";
        document.getElementById('warning').style.display = "block";
    }
    else {
        // izbran je model za paradižnik
        document.getElementById('myinput').style.display = "none";
        document.getElementById('id1').value = "Solyc00g005000.3";
        document.getElementById('warning').style.display = "none";
    }
}
```

Podobno je z *JavaScriptom* izvedeno, da se v primeru podanih *paired-end* zaporedij in izbranega orodja za poravnavo *Bowtie 2* pokaže izbirno

polje (`Orientation`), kjer lahko uporabnik izbere orientacijo nizov glede na referenčno zaporedje.

JavaScript je pomagal tudi pri priporočeni omejitvi orodja za zaznavo variacij *GATK* na orodje za poravnavo *BWA* [14]. Seveda se uporabniku ob tem na strani pokaže pojasnilo, zakaj je do tega prišlo.

Vsa vnosna polja razen tistega za uporabniško ime so predizpolnjena z zgledi. Tako se uporabnik lahko ravna po njih in ne pride do nepotrebne zmede. Predvsem je to pomembno pri podajanju poti do *paired-end* zaporedij (primer predizpolnjene poti do zaporedij `/.../L1_1.fq.gz,/.../L1_1.fq.gz`), saj se mora vrstni red zaporedij v poljih `Sequence1` in `Sequence2` paroma ujemati. Tako mora biti par tretje naštetega zaporedja v polju `Sequence1` prav tako na tretjem mestu v polju `Sequence2`. Zaporedja pa pri naštevanju ločimo z vejico.

Z gumbom (`Submit`) uporabnik potrdi svojo izbiro in sproži izvajnaje postopka. Ob tem preide na novo stran, kjer lahko spremlja izvajanje postopka.

Med izvajanjem se na vrhu strani predvaja slika z oznako aplikacije. Ko se izvajanje aplikacije ustavi, se slika spremeni v statično oznako aplikacije.

Na strani je še gumb (`STOP`). S klikom nanj uporabnik pošlje ukaz za ustavitev delovanja in se vrne na začetno stran.

Med postopkom se uporabniku pod oznako aplikacije izpisujejo koraki, ki jih aplikacija trenutno izvaja.

3.3 Zaledni del rešitve

Zaradi lažjega razumevanja je predstavitev zalednega dela rešitve razčlenjena na več delov.

3.3.1 Zagon Bash skripte

PHP skripta poskrbi za prenos vhodnih podatkov in zagon *Bash* skripte. Skrbi pa tudi za to, da se izpisi *Bash* skripte pokažejo na spletni strani in tako lahko uporabnik natančno ve, kaj se trenutno dogaja z aplikacijo. V

nadaljevanju podan primer *PHP* kode, ki prikazuje prenos vhodnih podatkov in zagon *Bash* skritpe:

```
// primer prenosa vhodnih podatkov
$iden = $_POST['idf'];
$seq1 = $_POST['s1'];
$seq2 = $_POST['s2'];
$sonly = $_POST['so'];
$mm = $_POST['map'];

/* primer zagona Bash skripte, ki svoje korake izpisuje direktno
na spletni strani */
while (@ ob_end_flush());

$proc = popen("./script.sh '$name' '$fasta' '$gen' '$iden' '$seq1'
'$seq2' '$sonly' '$mm' '$snp' '$cpu' '$comm' '$forw' '$ploid'", 'r');
echo '<pre>';
while (!feof($proc))
{
    echo fread($proc, 4096);
    @ flush();
}
echo '</pre>';
```

3.3.2 Preverjanje vhodnih podatkov

Zagnana *Bash* skripta najprej preveri, ali se na NIB-ovem strežniku `Ibis` že izvaja kakšna aplikacija *Amplicode 2018*, saj se zaradi omejenih resursov lahko izvaja le ena *Amplicode 2018* aplikacija naenkrat. To storimo tako, da vsaka skripta ob zagonu zapiše svoj `pid` v datoteko `pid.txt`.

```
#najprej preveri, če datoteka pid.txt sploh obstaja,
```

```
#če obstaja, preveri ali se proces s pidom,
#ki je v njej zapisan, izvaja

FILE="/DATA/workspace/amplicode/pid.txt"
if [ -f "$FILE" ]; then
    pi=$(cat "$FILE" | cut -d " " -f 1)
    if ps -p $pi > /dev/null
    then
    echo "Another version of Amplicode 2018 is already running."
    exit 1
fi
fi

#zapiše pid trenutno izvajajoče skripte
pi=$$
echo $pi > "$dat/pid.txt"
```

Sledi preverjanje, če je vpisani uporabnik na seznamu uporabnikov s pravico zagona *Amplicode 2018* aplikacije. Tega je posredoval sistemski administrator. Tudi spreminjanje seznama uporabnikov je v domeni sistemskega administartorja.

Aplikacija dodeli vsakem uporabniku svoj imenik. V njem so glede na čas izvajanja poimenovani podimeniki, v katerih se nahajajo rezultati (na primer `DATA/workspace/amplicode/UPORABNIK/Fri_Apr_13_12:29:24_CEST_2018`). Zaradi varčnega ravnanja s prostorom na strežniku se brišejo tiste datoteke, ki jih skripta tvori za potrebe izvajanja ukazov orodij za analizo *NGS* in za končnega uporabnika niso zanimive. Prav tako se podimeniki s končnimi rezultati avtomatsko brišejo po enem tednu. Brisanje je realizirano z uporabo *Cron-a* [12]. Ta vsako jutro ob enih kliče skripto `del.sh`:

```
0 1 * * * /DATA/workspace/amplicode/del7days.sh
```

Skripta `del.sh` preveri, če obstaja kakšen podimenik, ki je starejši od enega tedna. Če ga najde, ga izbriše:

```
#!/bin/bash
find /DATA/workspace/amplicode/ -mindepth 2 mtime +6 -delete
```

Glavna skripta ustvari tudi `log.txt` datoteko, v kateri so sprotni izpisi uporabljenih programskih orodij in morebitnih napak, do katerih je prišlo pri izvajanju. Primer vsebine datoteke:

```
AMPLICODE LOG FILE CREATED
Fri_Aug_31_14:26:51_CEST_2018
[M::bwa_idx_load_from_disk] read 0 ALT contigs
[M::process] read 1666668 sequences (60000048 bp)...
[M::process] read 1666668 sequences (60000048 bp)...
[M::mem_process_seqs] Processed 1666668 reads
in 35.020 CPU sec, 5.843 real sec
[M::process] read 466844 sequences (16806384 bp)...
[M::mem_process_seqs] Processed 1666668 reads
in 35.523 CPU sec, 5.847 real sec
[M::mem_process_seqs] Processed 466844 reads
in 11.073 CPU sec, 1.689 real sec
[main] Version: 0.7.17-r1188
.....
```

Nato se preveri, če na strežniku teče *Virtual box* [28]. Ta zahteva veliko delovnega pomnilnika in procesorske moči, zato se v primeru njegove prisotnosti ustrezno omeji največje število niti, ki jih lahko aplikacija uporabi. O tem se obvesti tudi uporabnika.

Glavna skripta vsebuje kar nekaj časovno potratnih ukazov, zato pred začetkom izvajanja preveri veljavnost vhodnih podatkov. Najprej preveri obstoj in veljavnost `FASTA` datoteke (prvi znak `FASTA` datoteke mora biti `>`).

Če uporabnik poda pot do svojega `GFF` modela, se preveri obstoj tudi te datoteke ter vsebovanost podanega genskega `ID-ja` v njej. Primer vsebine `GFF` datoteke [10]:

```
##gff-version 3
ctg123 . mRNA        1300  9000  .  +  .  ID=mrna0001;Name=soni..
ctg123 . exon        1300  1500  .  +  .  ID=exon00001;Parent=m..
ctg123 . exon        1050  1500  .  +  .  ID=exon00002;Parent=m..
ctg123 . exon        3000  3902  .  +  .  ID=exon00003;Parent=m..
ctg123 . exon        5000  5500  .  +  .  ID=exon00004;Parent=m..
ctg123 . exon        7000  9000  .  +  .  ID=exon00005;Parent=m..
```

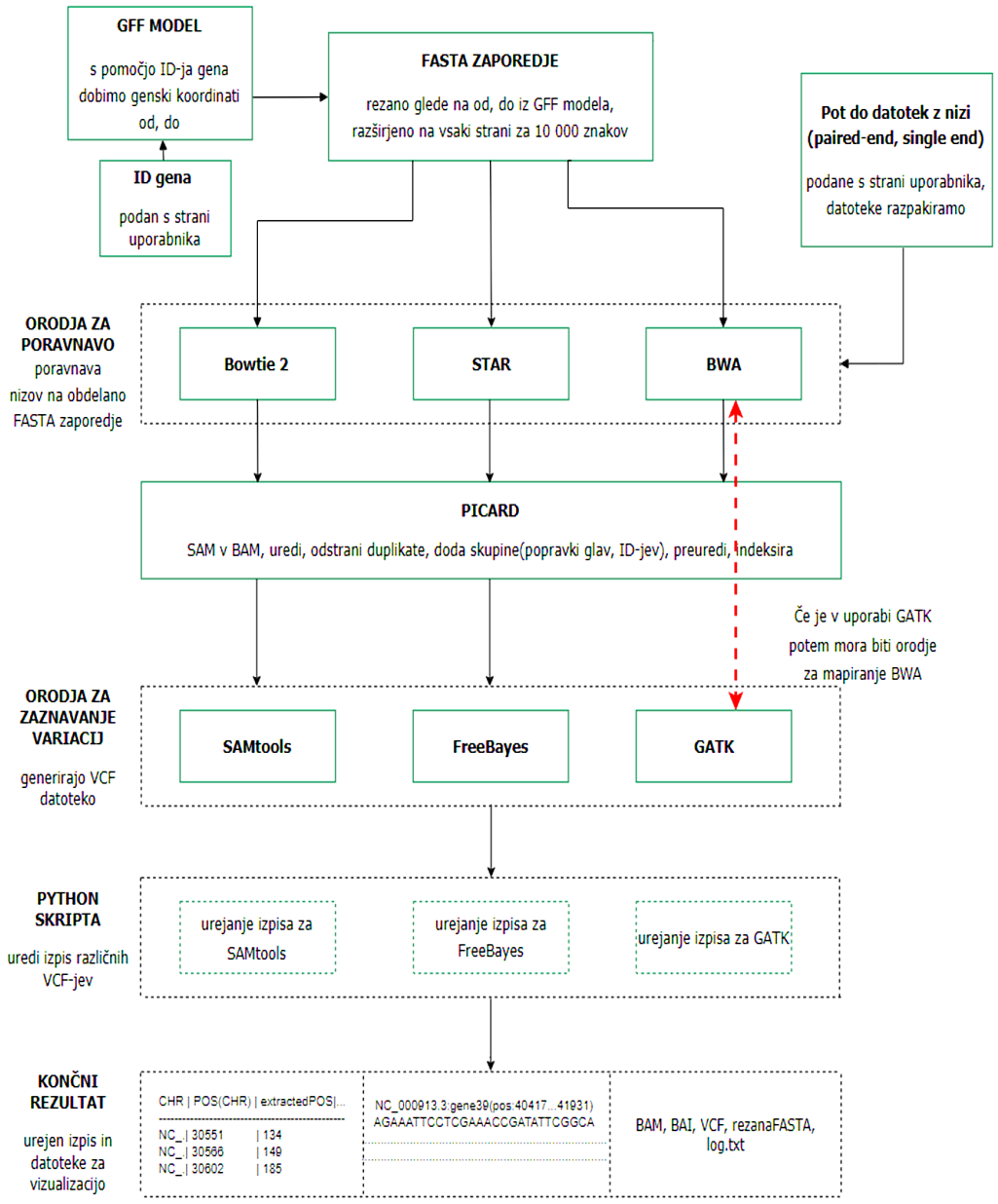
Gen razširimo za 10000 znakov na vsaki strani. S postopkom razširitve upoštevamo tudi sosednje gene, kar je koristno v primeru napačne ali nenatančne anotacije. Tako zagotovimo poravnavo nizov na sosednje gene in medgenske regije. Če postopka razširitve ni mogoče izvesti (gremo po odštevanju 10000 v negativno vrednost ali po prištevanju 10000 čez dolžino `FASTA` zaporedja), določimo vrednost `od` na 1 in vrednost `do` na dolžino referenčnega `FASTA` zaporedja.

Iz danega primera, je razvidno, da bi skripta ob podanem `ID-ju` `mrna0001` nastavila `od` na 1300 in `do` na 9000. Nato bi prvega zmanjšala in drugega povečala za 10000. Ker pa bi vrednost prvega s tem postala negativna, je končni rezultat `od` enak 1, `do` pa 19000. Iz celotnega `FASTA` zaporedja bi nato izločili znake glede na poziciji `od` in `do`.

Poleg podane `GFF` datoteke skripta preveri tudi obstoj *single* in *paired-end* zaporedij, če so le ta podana.

3.3.3 Osrednji del Bash skripte

Zaradi boljšega razumevanja osrednjega delovanja skripte smo njeno delovanje prikazali na sliki 3.2. Ta nam v gorbem predstavi potek skripte, ki je bolj natančno opisan na naslednjih straneh.



Slika 3.2: Potek glavne Bash skripte.

Skripta poskrbi, da je `FASTA` datoteka zapisana v dvovrstični obliki, ki nam bo olajšala štetje znakov pri izločitvi dela `FASTA` zaporedja glede na poziciji, ki ju dobimo iz `GFF` datoteke s pomočjo `ID-ja`. Prva vrstica vsebuje

enoličen opis zaporedja, v drugi pa je samo zaporedje. Primer vsebine `FASTA` datoteke:

```
>StPGSC4.04n_Chr01
AGATACAAATTACAATACACGAGCATCGTTTAATACTAATGTATCATGATTTAC...
```

V praksi izločitev naredimo tako, da režemo drugo vrstico `FASTA` datoteke. Števila znakov v prvi vrstici ne spremljamo, saj gre samo za enoličen opis celotnega zaporedja, ki pri izločitvi dela zaporedja ne igra pomebne vloge.

Nato skripta razpakira podane *single* in *paired-end* datoteke, ki vsebujejo nize za poravnavo. To stori z uporabo *pigz* [1] programa, ki lahko teče vzporedno ter tako prihrani nekaj časa.

Sledi poravnava razpakiranih nizov na prej obdelano `FASTA` zaporedje z uporabo orodij za analizo *NGS* podatkov. Najprej uporabimo orodja za poravnavo, ki so odvisna od izbire uprabnika: *Bowtie 2*, *BWA* ali *STAR*. Večina takšnih orodij (pri nas *Bowtie 2* in *BWA*) temelji na FM indeksu [1], ki sloni na Burrows-Wheelerjevi preslikavi. Psevdokoda za Burrows-Wheelerjevo preslikavo v primeru podanega niza a je naslednja:

1. ustvarimo tabelo v kateri so vrstice vse možne rotacije niza a,
2. vrstice uredimo po abecedi,
3. zadnji stolpec sestavlja niz Burrows-Wheelerjeve preslikave.

Na sliki 3.3 podan primer za niz BANANA [5].

Tako oblikovan niz je lažje zapakirati, kar nam pride prav pri ogromnih DNA zaporedjih, ki jih je potrebno med obdelavo spraviti v delovni pomnilnik. Takšen niz je uporaben pri FM indeksu, s pomočjo katerega je mogoče, kljub kompresiji relativno hitro poizvedovati po podnizih (z manj kot linearno časovno zahtevnostjo) [29].

Po uporabi orodij za poravnavo dobimo `SAM` datoteko, ki jo s pomočjo orodja *Picard* spremenimo v `BAM`, uredimo, odstranimo duplikate, dodamo

[1] Gre za zahteven postopek, ki je v bioinformatiki uporaben za poravnavo genskih zaporedij. Podrobna predstavitev FM indeksa presega namen diplomskega dela.

Vhodni niz a	Vse rotacije niza a	Uredimo po abecedi	Določimo zadnji stolpec	Izpišemo zadnji stolpec
^BANANA|	^BANANA| |^BANANA A|^BANAN NA|^BANA ANA|^BAN NANA|^BA ANANA|^B BANANA|^	ANANA|^B ANA|^BAN A|^BANAN BANANA|^ NANA|^BA NA|^BANA ^BANANA| |^BANANA	ANANA|^B ANA|^BAN A|^BANAN BANANA|^ NANA|^BA NA|^BANA ^BANANA| |^BANANA	BNN^AA|A

Slika 3.3: Primer Burrows-Wheelerjeve preslikave [5].

potrebne oznake (skupine), preuredimo in indeksiramo. Glede na izbiro (*SAMtools, FreeBayes, GATK*) uporabnika uporabimo ustrezno orodje za zaznavo variacij na zaporedju (*SNP tool*), ki nam generira rezultat v obliki `VCF`.

3.3.4 Python skripta in rezultati

Nato kličemo *Python* skripto, ki ne glede na uporabljeno orodje za zaznavo variacij na zaporedju (vsako namreč generira rahlo drugačen `VCF`), vrne izpis v poeenotenem tabelaričnem formatu. Primer:

```
CHR     | POS(CHR) | extractedPOS | REF | ALT | qualCOV..| altF
-------------------------------------------------------------
NC_00.. | 37484    | 7067         | C   | T   | 22       | 1.0
NC_00.. | 37534    | 7117         | A   | G   | .        | .
NC_00.. | 37535    | 7118         | C   | T   | .        | .
NC_00.. | 37812    | 7395         | C   | A   | 15       | 1.0
NC_00.. | 38557    | 8140         | A   | G   | 20       | 1.0
NC_00.. | 38676    | 8259         | A   | G   | 10       | 1.0
```

```
NC_00.. | 38937    | 8520         | T   | C   | .        | .
NC_00.. | 39006    | 8589         | G   | A   | 10       | 1.0
NC_00.. | 39121    | 8704         | T   | A   | .        | .
NC_00.. | 39172    | 8755         | T   | C   | 27       | 0.93
NC_00.. | 39216    | 8799         | A   | G   | 16       | 1.0
NC_00.. | 39305    | 8888         | C   | T   | .        | .
...
```

V prvem stolpcu imamo podano ime kromosoma. Drugi stolpec vsebuje pozicijo, na kateri se variacija nahaja glede na kromosom (celotno `FASTA` zaporedje, pred izločitvijo). V tretjem stolpcu se nahaja pozicija glede na izločeno `FASTA` zaporedje. V četrtem stolpcu je referenčni nukleotid, v petem pa predlagana alternativa. V šestem stolpcu imamo podano pokritost[2] in v zadnjem frekvenco alternative, če je le ta dovolj visoka (sicer je tudi tam pika), kar je odvisno od `ploidnosti` organizma. Primer:

```
#slovar, katerega ključi predstavljajo ploidnost
#vrednosti pa minimalno potrebno frekvenco za izpis v tabeli
ploidy = {2: 0.30,
          4: 0.15,
          6: 0.1125,
          8: 0.075
          }
```

Celoten izpis nato preuredimo, tako da imamo v prvi vrstici ime kromosoma (`NC_000913.3`), `ID` gena (`gene39`), pozicijo na kateri se gen nahaja (`40417...41931` - to sta poziciji `od` in `do` iz datoteke `GFF`, še preden vzamemo dodatnih 10000 znakov na vsaki strani), podano obliko zapisa (`Ref:Ref:Alt`) ter poziciji, glede na kateri je bila izločena `FASTA` (`30417...51931`, to sta poziciji po razširitvi za 10000 znakov). V drugi vrstici se nahaja obdelana

[2]Upoštevane so samo pokritosti višje kakovosti tiste, ki so večje ali enake 20. Če pogoji niso izpolnjeni, je na tem mestu pika. Posledično ne moremo izračunati *altF*, zato je v tem primeru tudi tam pika.

FASTA datoteka (razširjena z 10000 znaki na vsaki strani), v tretji referenca, ki jo določi orodje za zaznavanje variacij na zaporedju, in v četrti predlagane alternative. Primer izpisa je naslednji (prva vrstica je tukaj zaradi lepšega zapisa razdeljena v dve vrstici):

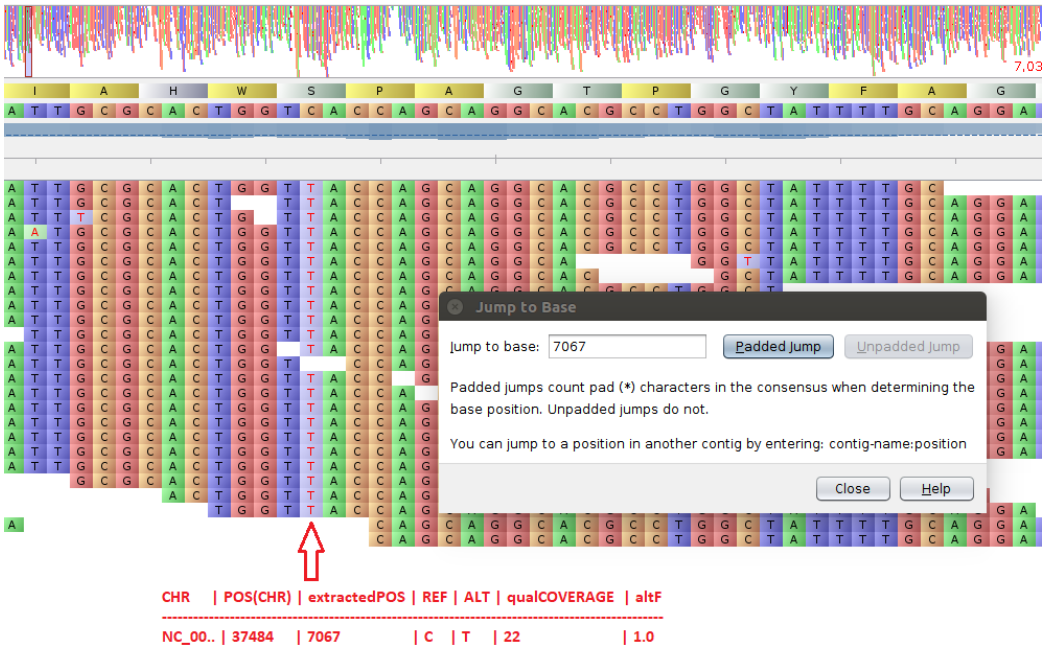
```
>NC_000913.3:gene39(pos:40417...41931)
_Ref:Ref:Alt_extracted_pos:30417...51931
AGAAATTCCTCGAAACCGATATTCCGGTATTCGGCATCTGTCTCGGTCATCAGCTGCTGGCGCT
.............................................................
.............................................................
```

Spodaj viden primer različnih alternativ: *SNP-ja* in *insercije*, lahko pa se pojavi tudi *delecija*. Pike so tu samo zaradi bolj preglednega prikaza presledkov.

```
.............................................................
CAAGCTTCATGTTCCATTTACAAATATTTCAG.GTATTGCCTGTCATGGTAATGGGGGCTTTC
.................TTA.....ATTTCAG...A.........................
.................GTT.....CTGCCAAG..T.........................
```

Poleg danih rezultatov sta uporabniku v njegovem podimeniku na voljo še datoteki log.txt in VCF, ki ga je generiralo orodje za zaznavo variacij. Tu so tudi datoteke BAM, BAI, FASTA, ki jih lahko uporabimo za grafičen prikaz rezultatov v orodjih za vizualizacijo (kot sta npr. *IGV* [13] in *Tablet* [19]). Tam lahko glede na pozicije iz tabelaričnega izpisa (tretji stolpec) skočimo (z npr. pri *Tablet-u* gumbom Jump to Base) na izbrano mesto in še sami preverimo pokritost. Pokritost v vizualizatorjih zna biti enaka ali večja kot v tabelaričnem izpisu, saj smo tam upoštevali le kakovostne pokritosti. Primer viden na slikah 3.4 in 3.5.

Slika 3.4: Primer grafičnega prikaza podatkov z orodjem *Tablet*.



Slika 3.5: Povezava med tabelaričnim izpisom in orodjem *Tablet*.

Pot do imenika z rezultati se izpiše na trenutni strani, pošlje pa se tudi uporabniku po elektronski pošti. Skripta za to je predstavljena v nadaljevanju.

```
mail -s "Amplicode 2018" "$naslov" <<EOF
Thank you for using Amplicode 2018, you can see your results
here: $pot


Sincerely,
AMPLICODE 2018 TEAM
EOF
```

Samih rezultatov zaradi njihove velikosti ni mogoče pošiljati po elektronski pošti, saj so datoteke običajno zelo velike, največja velikost priloge pa je omejena (pri npr. *Microsoft Outlook-u* je 20Mb [3]).

Uporabnik ima na spletni strani tudi možost zaustavitve aplikacije. Če klikne na gumb `STOP`, se prek *PHP-ja* zažene naslednja skripta `stop.sh`:

```
#!/bin/bash

pi=$(cat "/DATA/workspace/amplicode/pid.txt" | cut -d " " -f 1)
kill -1 $pi
```

Ta pošlje signal 1 (ang. *hangup*), ki se ulovi v past glavne skripte

```
trap 'kill $(jobs -p)' EXIT
```

in tako ustavi izvajanje glavne skripte ter vsa njena opravila, uporabnik pa se vrne na začetno stran.

Poglavje 4

Sklepne ugotovitve

Spletna aplikacija *Amplicode 2018* je končana in pripravljena za uporabo. Trenutno se testno izvaja na NIB-u. V tem delu predstavljamo še prispevke diplomske naloge, nekaj večjih težav pri izdelavi in odziv oziroma izkušnjo uporabnikov. Nalogo sklenemo z možnostmi izboljšav in idejami za nadgradnjo z vidika računalništva.

4.1 Prispevki diplomske naloge

Glavni prispevek diplomske naloge je razvita spletna aplikacija. Ta bo koristila predvsem *wet lab* raziskovalcem, saj bodo na podlagi rezultatov lahko primerjali kakovost različnih orodij za analizo *NGS*. To bo omogočilo tudi izbiro nadaljnjih (*wet lab*) korakov kot je npr. pri *PCR* analizi oblikovanje *oligonukleotidnih začetnikov* za podvojevanje ali *amplifikacijo*.

Diplomsko delo pa bo koristilo tudi vsem *dry lab* razvijalcem, ki se bodo znašli pred podobno nalogo. Opis razvoja aplikacije jim bo dal dobro izhodiščno točko za rešitev nekaterih problemov, do katerih lahko pride pri razvoju podobnih aplikacij.

4.2 Težave pri izdelavi

Največjo težavo pri izdelavi so predstavljali odprtokodni programi za analizo *NGS* podatkov. Pri teh bi lahko izpostavili nekaj ključnih pomanjkljivosti:

- **Pomanjkljiva in včasih neobstoječa dokumentacija.** Dokumentacija običajno napisana površno, vključuje samo opise okrnjenih delov uporabe programa in ne celovite razlage. Raziskovalec je pogosto prepuščen sam sebi in skupnosti uporabnikov ter odzivnosti razvijalca, čeprav obstajajo tudi svetle izjeme.

- **Pomanjkanje širše skupnosti za razpravo o orodjih.** Obstaja sicer nekaj bioloških forumov (kot je npr. *Biostars*), kjer se včasi pojavi tudi kakšno vprašanje iz uporabe odprtokodnih orodij za analizo *NGS* podatkov. Žal pa tovrstna vprašanja prevečkrat ostanejo brez odgovora ali samo delno odgovorjena.

- **Nestabilne različice orodij.** Nekatere različice orodij niso popolnoma izpopolnjene in marsikdaj se izkaže, da je v njih vse preveč *hroščev*, ki povzročajo nepravilno delovanje programa ali celo odpoved.

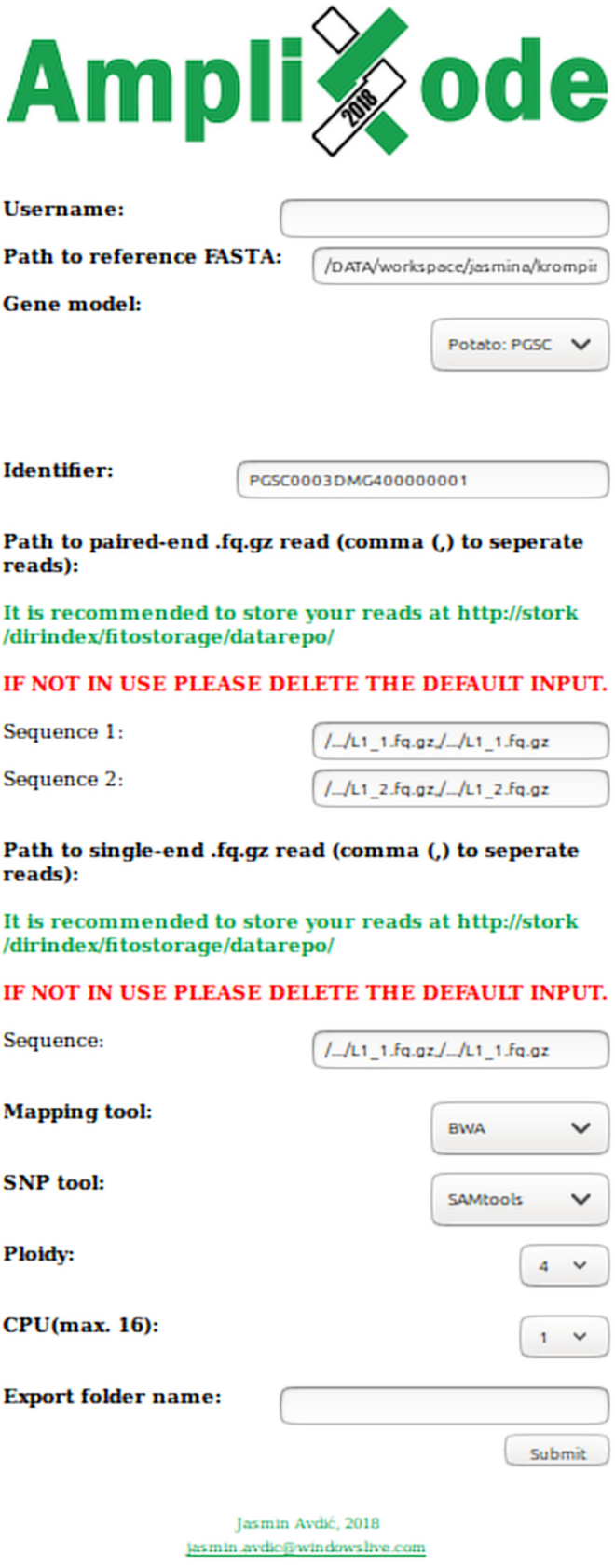
- **Nove različice orodij so nekompatibilne s starimi in z ostalimi orodji.** Nova različica orodja včasih popolnoma zamenja/ukine nastavitve, ki so v prejšnji različici delovale. Takšen preskok po nepotrebnem zmede uporabnika in mu daje občutek uporabe popolnoma drugega orodja ne pa nadgrajene različice. Prav tako se lahko zgodi, da postane cevovod (ang. *pipeline*), kjer je izpis iz prve aplikacije vhodni podatek za drugo, ob posodobitvi popolnoma neuporaben. Če prva aplikacija spremeni svoj izpis, pride do nekompatibilnosti.

4.3 Uporabniška izkušnja in možnosti izboljšav

Ko je bila aplikacija izdelana, je bil čas za praktični preizkus. Izvedla ga je bodoča uporabnica aplikacije, ki smo jo sprva vodili skozi postopek uporabe

in na koncu povprašali po njenem mnenju. Predlagane možnosti izboljšav so bile:

- **Priročnik.** Za aplikacijo je potrebno napisati kratek priročnik, ki bo olajšal uporabo, prav tako pa dodati kratka navodila v program, ki bi se pokazala, ko gre uporabnik z miško čez vnosno polje.
- **Dodati `ploidnost` 1.** Trenutno so na izbiro `ploidnosti` 2, 4, 6 in 8. Pri `ploidnosti` 1 ne bi bilo nastavljene minimalne potrebne frekvence za tabelarični izpis, pač pa bi izpisali vse, brez pikic.
- **Dodati možnost *No Identifier* pri izbirnem seznamu *Gene model*.** V tem primeru uporabniku ni treba dodajati ali izbrati `ID-ja` in `GFF` datoteke, pač pa naj se za poravnavo uporabi kar celotno podano `FASTA` zaporedje, brez izločitve glede na `ID`. Pri tem se spremeni tudi končni izpis, saj ni potrebno več izpisovati dveh pozicij glede na izločeno in celo `FASTA` zaporedje.
- **Dodati v elektronsko pošto obvestilo, da so podimeniki z rezultati na voljo le teden dni od nastanka.**
- **Nakazati v elektronski pošti, naj se dobljena pot do podimenika prepiše v *WinSCP* [33].**
- **Spremeniti opis pred vnosnimi polji.** Npr. iz `Path to FASTA file` v `Path to reference FASTA` in poudariti, da je potrebno iz tistih vnosnih polj, ki jih ne bomo uporabljali, izbrisati predizpolnjene vrednosti.
- **Odstarniti polje `Remark` in namesto njega dodati polje `Export folder name`.** Poimenovanju podimenika z rezultati bo, poleg datuma tudi dodan vhodni podatek iz tega polja. Tako bodo podimeniki z rezultati poimenovani npr. `Fri_Apr_13_12:29:24_CEST_2018_vneseno_ime` namesto samo `Fri_Apr_13_12:29:24_CEST_2018`, kar bo uporabniku olajšalo razlikovanje med njimi.



Slika 4.1: Vnosna stran po izboljšavah.

- **Spremeniti imena končnih `BAM`, `BAI`, `VCF` datotek v uporabniku bolj prijazna.**
- **Analizirati rezultate in tako dobiti najzanesljivejše variacije.** Aplikacijo bi lahko nadgradili tako, da bi ponujala možnost analize rezultatov. Izpisala bi tiste variacije, ki se ujemajo pri vseh orodjih in so posledično najbolj zanesljive.

Večino izmed naštetih izboljšav smo že implementirali, kar je spremenilo začetno stran aplikacije. To je prikazano na sliki 4.1.

Glede na to, da je aplikacija *Amplicode 2018* zaživela v praksi je pričakovano, da bo v začetnem obdobju še kar nekaj predlogov izboljšav. Zaradi tega bo potrebno nuditi tehnično pomoč in jih realizirati. Verjetno pa bo z daljšim obdobjem uporabe (in bolj širokem spektrom uporabnikov) le treba strmeti k splošnosti uporabe, zato se bo število izboljšav aplikacije temu primerno zmanjšalo.

4.4 Ideje za nadgradnjo z vidika računalništva

Obstaja tudi nekaj idej za nadgradnjo z vidika računalništva:

- **Možnost dodajanja več orodij.** Glavna skripta je že zdaj napisana tako pregledno, da omogoča hitro dodajanje novih kompatibilnih orodij.
- **Personalizacija.** Vsakemu uporabniku bi se lahko glede na njegovo uporabniško ime ponudile prednastavitve. Npr. poti do `FASTA` datoteke ne bi bilo treba vsakič vpisovati, pač pa bi se uporabniku pokazal seznam njegovih `FASTA` datotek.
- **Optimizacija.** Na strežniku `Ibis` bi dovolili izvajanje več aplikacij *Amplicode 2018* naenkrat. To je primerno takrat, ko izvajanje posamezne aplikacije ne porabi veliko resursov. V tem primeru je treba spremljati predvsem zasedenost procesorja in delovnega pomnilnika.

Literatura

[1] Mark Adler. A parallel implementation of gzip for modern multiprocessor, multi-core machines. Dosegljivo: `https://zlib.net/pigz/`, [Dostopano: 16. 7. 2018].

[2] Adobe. Adobe photoshop cc. Dosegljivo: `https://www.adobe.com/si/products/photoshop.html`, [Dostopano: 16. 7. 2018].

[3] Outlook Apps. Maximum email size limit for gmail, outlook.com, etc. Dosegljivo: `https://www.outlook-apps.com/maximum-email-size/`, [Dostopano: 16. 7. 2018].

[4] John MS Bartlett and David Stirling. A short history of the polymerase chain reaction. In *PCR protocols*, pages 3–6. Springer, 2003.

[5] Michael Burrows and David J Wheeler. A block-sorting lossless data compression algorithm. 1994.

[6] Scitable by nature education. Snp. Dosegljivo: `https://www.nature.com/scitable/definition/single-nucleotide-polymorphism-snp-295`, [Dostopano: 16. 7. 2018].

[7] Josh Clevenger, Carolina Chavarro, Stephanie A Pearl, Peggy Ozias-Akins, and Scott A Jackson. Single nucleotide polymorphism identification in polyploids: a review, example, and recommendations. *Molecular plant*, 8(6):831–846, 2015.

[8] CW Dieffenbach, TM Lowe, and GS Dveksler. General concepts for pcr primer design. *PCR Methods Appl*, 3(3):S30–S37, 1993.

[9] Alexander Dobin, Carrie A Davis, Felix Schlesinger, Jorg Drenkow, Chris Zaleski, Sonali Jha, Philippe Batut, Mark Chaisson, and Thomas R Gingeras. Star: ultrafast universal rna-seq aligner. *Bioinformatics*, 29(1):15–21, 2013.

[10] Ensembl. Gff3 file format - definition and supported options. Dosegljivo: `https://www.ensembl.org/info/website/upload/gff3.html`, [Dostopano: 16. 7. 2018].

[11] Erik Garrison and Gabor Marth. Haplotype-based variant detection from short-read sequencing. *arXiv preprint arXiv:1207.3907*, 2012.

[12] The Open Group. Cron. Dosegljivo: `http://pubs.opengroup.org/onlinepubs/9699919799/utilities/crontab.html`, [Dostopano: 16. 7. 2018].

[13] Broad Institute. Integrative genomics viewer. Dosegljivo: `http://software.broadinstitute.org/software/igv/`, [Dostopano: 16. 7. 2018].

[14] Broad Institute. Mapping, processing, and duplicate marking with picard tools. Dosegljivo: `https://docs.google.com/file/d/0B2dK2q40HDWeaGVrbE1GVV9SQkE/preview`, [Dostopano: 16. 7. 2018].

[15] Broad Institute. Picard readme. Dosegljivo: `https://github.com/broadinstitute/picard/blob/master/README.md`, [Dostopano: 16. 7. 2018].

[16] Broad Institute. pipeline_overview.png. Dosegljivo: `https://software.broadinstitute.org/gatk/img/pipeline_overview.png`, [Dostopano: 16. 7. 2018].

[17] Broad Institute. Quick start guide. Dosegljivo: https://software.broadinstitute.org/gatk/documentation/quickstart, [Dostopano: 16. 7. 2018].

[18] National Human Genome Research Institute. Genetic mapping. Dosegljivo: https://www.genome.gov/10000715/, [Dostopano: 16. 7. 2018].

[19] The James Hutton Institute. Tablet. Dosegljivo: https://ics.hutton.ac.uk/tablet/, [Dostopano: 16. 7. 2018].

[20] Sonoko Kinjo, Norikazu Monma, Sadahiko Misu, Norikazu Kitamura, Junichi Imoto, Kazutoshi Yoshitake, Takashi Gojobori, and Kazuho Ikeo. Maser: one-stop platform for ngs big data from analysis to visualization. *Database*, 2018, 2018.

[21] Zhong lab. What is fasta format? Dosegljivo: https://zhanglab.ccmb.med.umich.edu/FASTA/, [Dostopano: 16. 7. 2018].

[22] Ben Langmead. Bowtie2 manual. Dosegljivo: https://github.com/BenLangmead/bowtie2/blob/master/MANUAL, [Dostopano: 16. 7. 2018].

[23] Heng Li and Richard Durbin. Fast and accurate long-read alignment with burrows–wheeler transform. *Bioinformatics*, 26(5):589–595, 2010.

[24] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, and Richard Durbin. The sequence alignment/map format and samtools. *Bioinformatics*, 25(16):2078–2079, 2009.

[25] Zhentang Li, Yi Wang, and Fei Wang. A study on fast calling variants from next-generation sequencing data using decision tree. *BMC bioinformatics*, 19(1):145, 2018.

[26] Genome Research Ltd. Bcftools manual. Dosegljivo: http://www.htslib.org/doc/bcftools.html, [Dostopano: 16. 7. 2018].

[27] International Society of Geneticy Genealogy Wiki. Next generation sequencing. Dosegljivo: https://isogg.org/wiki/Next_generation_sequencing, [Dostopano: 16. 7. 2018].

[28] Oracle. Virtualbox. Dosegljivo: https://www.virtualbox.org/, [Dostopano: 16. 7. 2018].

[29] Rossano Venturini Paolo Ferragina. Fm index. Dosegljivo: http://pages.di.unipi.it/ferragina/Libraries/fmindexV2/index.html, [Dostopano: 16. 7. 2018].

[30] Ensembl Project. Gff/gtf file format - definition and supported options. Dosegljivo: https://www.ensembl.org/info/website/upload/gff.html, [Dostopano: 16. 7. 2018].

[31] QIAGEN. Clc genomics workbench. Dosegljivo: https://www.qiagenbioinformatics.com/about/, [Dostopano: 16. 7. 2018].

[32] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3, 2016.

[33] WinSCP.net. Free sftp, scp and ftp client for windows. Dosegljivo: https://winscp.net/eng/download.php, [Dostopano: 16. 7. 2018].