LATVIJAS UNIVERSITĀTE

DATORIKAS FAKULTĀTE

METODES UN PROGRAMMATŪRA GENOMA DATU ANALĪZEI

MAĢISTRA DARBS

Autors: Iveta Milta

Stud. apl. Nr. im12164

Darba vadītājs: prof. Juris Vīksna

Rīga 2017

Anotācija

Maģistra darba tēma “Metodes un programmatūra genoma datu analīzei”

Maģistra darbā apkopota informācija par nozīmīgākajām genoma datu analīzes metodēm.

Izpētītas praksē biežāk lietotās metodes SNP (single nucleotide polymorphisms) identifikācijai jaunās paaudzes sekvencēšanas (NGS) datos, pieejamā programmatūra, kurā šīs metodes ir implementētas, un veiktas dažādu metožu un programmu darbības salīdzināšana uz simulētām NGS datu kopām.

Salīdzinātas programmas genoma datu analīzei - GemSIM, Bowtie2, Samtools, Bfctools, Gatk. Iegūtie rezultāti pārbaudīti ar un salīdzināti ar simulētām datu kopām.

Darbā aprakstīta četru genotipēšanas algoritmu darbība: GenoSNP, Illuminus, CRLMM, un GenCall. Tie salīdzināti pēc datu precizitātes, kvalitātes rādītājiem paraugu datos.

Atslēgvārdi: SNP, DNS, sekvencēšana, genotipēšanas algoritms

Annotation

A Master thesis topic “Methods and Software genome data analysis”

Master's work contains information about the important genome data analysis methods.

Verified the practice used methods SNP (single nucleotide polymorphisms) identification of a new generation of sequencing (DGS) data, available software, as these methods are implemented, and carried out different methods and program performance comparison of simulated DGS datasets.

Compared the genome data analysis - GemSIM, Bowtie2, Samtools, Bfctools, Gatk. The results obtained and verified by compared with simulated data sets.

The four genotyping algorithm described in: GenoSNP, Illuminus, CRLMM and GenCall. They compared by the degree of accuracy, quality indicators in the sample data.

Keywords: SNP, DNA sequencing and genotyping algorithm

Autoreferāts

Darbā izveidots apraksts par biežāk lietotajām viena nukleotīda SNP meklēšanas un analīzes metodēm un algoritmiem. Kā arī programmām, kas atrod un analizē SNP genoma datos.

Viena nukleotīda polimorfisms (SNP) ir visbiežāk sastopamā DNS sekvences variācija cilvēka genomā, kad viens nukleotīds – A (adenīns), T (timīns), C (citozīns) vai G (guanīns) – genomā atšķiras starp vienas sugas indivīdiem.

SNP ir ideāli marķieri, lai noteiktu gēnu saistības ar kompleksām slimībām, tāpēc ka tie ir blīvi izvietoti cilvēka genomā (1 SNP uz ap-tuveni 500 - 1000 bāzu pāriem) un ir pieejams plašs metožu klāsts to genotipēšanai.

SNP ietekmē arī organisma atbildes reakcijas uz medikamentiem vai vides ķimikālijām.

SNP ir arī liela ietekme visās zāļu izstrādes fāzēs, sākot ar mērķa identifikāciju līdz klīniskajiem pētījumiem. Šo polimorfismu analīze palīdz arī pielāgot medikamentus vai medikamentu grupas konkrētam genotipam, kas ir farmakogenomikas pamatprincips.

Lai noskaidrotu kā SNP marķieri ietekmē references genomu, tika izpētītas vairākas programmas, kas veic genoma analīzi. Tika apskatītas sekojošas programmas - GemSIM, SoapSNP, Atlas2, Samtools, tomēr ne no visām iegūtie rezultāti noderēja tēmas izpētē. Jo dažas no programmām, mainoties versijām, vienkārši nedarbojās. Tāpēc rezultātā tika izmantotas tikai – GemSIM, Samtools, Bowtie2, BCFtools, GATK.

Izmantotā literatūra šim mērķim tika meklēta internetā un dažādos zinātniski pētnieciskajos darbos. Zinātnieskie raksti no zinātniskajām konferencēm, institūtiem un universitātēm bija izmantojami tikai PDF formātā. Pavisam nelielā apjomā tika izmantoti apraksti no interneta vietnēm, tādām kā wikipēdija vai medicīnai paredzētās interneta vietnes. Izmantotā literatūra norādīta izmantotās literatūras sarakstā un atzīmēta darbā, izmantojot atsauces no literatūras saraksta.

Praktiski tika pētītas programmu iespējas genotipa analīzē un rezultāti apkopoti, un tie ir apskatāmi darba praktiskajā daļā. Praktiskajā daļā tika iegūti VCF formāta dati, kuri tika apkopoti un salīdzināti pēc SNP skaita sakritības vai nesakritības.

Saturs

Apzīmējumu saraksts

DNS – dezoksiribonukleīnskābes

RNS – ribonukleīnskābe

SNP – viena nukleotīda polimorfisms

tRNS – transporta ribonukleīnskābe

Ievads

DNS molekulārais marķieris atklāj nukleotīdu sekvences variāciju konkrētā genoma lokācijā. Arvien biežāk šos marķierus izmanto, lai iegūtu informāciju par organismu genoma sastāvu un struktūru.

SNP ir ideāli marķieri, lai noteiktu gēnu saistības ar kompleksām slimībām, tāpēc ka tie ir blīvi izvietoti cilvēka genomā (1 SNP uz aptuveni 500 - 1000 bāzu pāriem) un ir pieejams plašs metožu klāsts to genotipēšanai. SNP arī ietekmē organisma atbildes reakcijas uz medikamentiem vai vides ķimikālijām. Neseni dati liecina to, ka ģenētiskie polimorfismi veido 15-30% no kopējā zāļu reaģēšanas variāciju skaita.

SNP ir arī liela ietekme visās zāļu izstrādes fāzēs, sākot ar mērķa identifikāciju līdz klīniskajiem pētījumiem. Šo polimorfismu analīze palīdz arī pielāgot medikamentus vai medikamentu grupas konkrētam genotipam, kas ir farmakogenomikas pamatprincips.

Darba mērķi:

- Izpētīt esošās SNP (Single Nucleotide Polymorphism) noteikšanas metodes un programmatūru.
- Salīdzināt iegūtos rezultātus uz reālām un simulētām NGS (Next Generation Sequencing) datu kopām.

1. Īss ievads ģenētikā

1.1. Cilvēka ģenētika

Ģenētika (grieķu: genētikos — 'uz dzimšanu vai izcelšanos attiecīgs') ir zinātne par gēniem, iedzimtību un organismu dažādību. Vārds "ģenētika" pirmo reizi tika lietots, lai aprakstītu iedzimtības studēšanu un variāciju zinātni. Terminu piedāvāja izcilais britu zinātnieks Viljams Betsons (William Bateson), pirmoreiz to lietojot personiskā vēstulē ģeologam Ādamam Sedžvikam (Adam Sedgwick) 1905. gada 18. aprīlī. Betsons terminu "ģenētika" publiski pirmoreiz lietoja Trešajā Starptautiskajā ģenētikas konferencē Londonā 1906. gadā. (1)

Zināšanas par ģenētiku cilvēki izmantoja jau senatnē, veicot augu un dzīvnieku selekciju un pavairošanu. Mūsdienu pētījumos ģenētika strauji attīstās un nodarbojas, piemēram, ar specifisku gēnu funkciju izpēti, gēnu mijiedarbības analīzi. Organismos ģenētiskā informācija galvenokārt tiek pārnēsāta hromosomās, kur tā ir iekļauta dezoksiribonukleīnskābes (DNS) molekulu ķīmiskajā struktūrā.

Gēni kodē informāciju, kas nepieciešama proteīnu sintezēšanai no aminoskābēm. Proteīni savukārt nosaka organisma uzbūvi un izskatu.

Gēns satur instrukcijas par to, kā uzbūvēt noteiktu proteīnu, tādējādi gēns ir kodējums proteīnam. Princips "viens gēns, viens proteīns" ir uzskatāms par vienkāršojumu — piemēram, viens gēns var ražot vairākus produktus, atkarībā no tā, kā tā ieraksts tiek regulēts.

Pamatojoties uz DNS gēnu kodu (nukleotīdu sekvenci), ar īpašu fermentu — RNS polimerāžu — palīdzību tiek sintezētas ribonukleīnskābes (RNS). Šo procesu sauc par transkripciju. Lai sintezētu proteīnus, nepieciešamas dažādu veidu RNS. Matricas RNS (mRNS) tieši kodē proteīnu aminoskābju secību. Procesu, kurā uz mRNS matricas ribosomās tiek sintezēti proteīni, sauc par translāciju. Ribosomu RNS jeb rRNS ir ribosomu uzbūves pamats, bet transporta RNS (tRNS) kalpo aminoskābju nogādāšanai uz ribosomām.

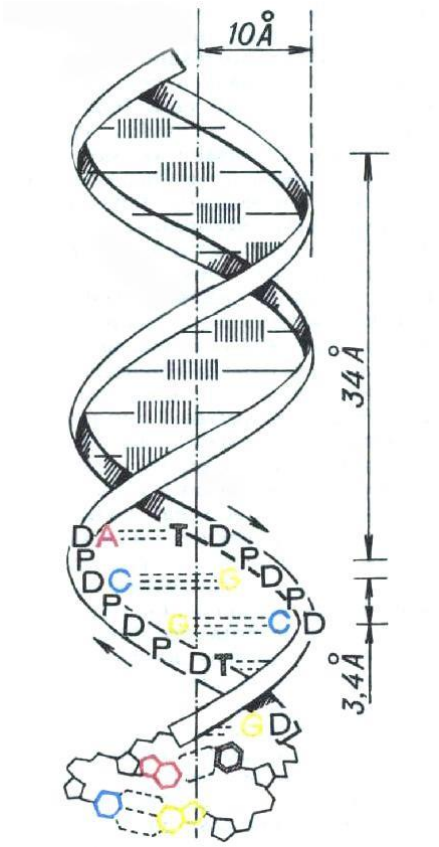
Ģenētika nosaka daudz, bet ne visu, par organisma, ieskaitot cilvēkus, izskatu un, iespējams, arī uzvedību. Apkārtējās vides atšķirības un nejauši faktori arī to iespaido.

Dezoksiribonukleīnskābe (DNS)

Dezoksiribonukleīnskābe jeb (DNS) ir molekulārs organiskss savienojums (nukleīnskābe), kas satur ģenētisko informāciju un veido gēna ķīmisko pamatu. Dezoksiribonukleīnskābes molekula sastāv no 4 nukleotīdu veidiem. Katrs nukleotīds satur purīnbāzi (adenīnu A, guanīnu G) vai piramidīnbāzi (citozīnu C, timīnu T), ogļhidrātu dezoksiribozi un fosforskābi. Nukleotīdi ir savstarpēji savienoti garās virknēs. No nukleotīdu sastāva, daudzuma un izvietojuma secības virknē atkarīgs sugai raksturīgais dezoksiribonukleīnskābes specifiskums. Nukleotīdu secībā ietvertā ģenētiskā informācija

nosaka, kādas olbaltumvielas šūna spēj sintezēt, t.i., kādas aminoskābes kādā secībā iesaistās olbaltumvielas molekulā. (2)

Dezoksiribonukleīnskābes molekula sastāv no 2 virknēm, kas, ar regulāriem vijumiem savītas ap kopēju asi, veido dubultspirāli. Ūdeņraža saites savieno vienas spirāles timīna (T) atlikumu ar otras spirāles adenīna (A) atlikumu un analoģiski – guanīna (G) atlikumu ar citozīna (C) atlikumu. Tādējādi virknes papildina viena otru. Šāda īpatnēja struktūra nosaka ne tikai dezoksiribonukleīnskābes bioloģiskās, bet arī fizikālās un ķīmiskās īpašības - dezoksiribonukleīnskābes molekulas ir stabilas. Šūnu kodolos dezoksiribonukleīnskābe ir saistīta galvenokārt ar olbaltumvielām kā dezoksiribonukleoproteīds, kas ir galvenā hromosomu sastāvdaļa. Šūnas dalīšanās cikla laikā dezoksiribonukleīnskābes daudzums dubultojas. Dezoksiribonukleīnskābes dubultošanās (replikācijas) procesā dubultspirāles virknes atdalās viena no otras, uz katras no tām fermenta ietekmē sintezējas jauna, tai komplementāra virkne. Tādējādi katra no abām jaunajām dezoksiribonukleīnskābes molekulām, kas identiskas vecajai, satur vienu veco un vienu no jauna sintezēto virkni.



Dezoksiribonukleīnskābes *dubultspirāles shēma*: A — *adenīns*; G — *guanīns*; T — *timīns*; C — *citozīns*; D — *dezoksiribozes atlikums*; P — *fosforskābes atlikums*

Ilustrācija 1 DNS dubultspirāles shēma (2)

Hromosoma ir šūnas kodola pašreproducējoša pavedienveida struktūra, kurā ieslēgta šūnas galvenā ģenētiskā informācija. Šūnas dalīšanās laikā hromosomas sadalās pa meitšūnām. Hromosomas sastāv no DNS, kas ieslēgta galvenokārt histonu tipa olbaltumvielu apvalkā.

Hromosoma veidota no spiralizēta hromatīna pavediena, kas sastāv aptuveni no 40% DNS un 60% olbaltumvielu. Šīs olbaltumvielas iedala histonu un nehistonu olbaltumvielās. H2A, H2B, H3 un H4 olbaltumvielas veido nukleosomas, bet H1 noslēdz ienākošo un izejošo DNS pavedienu. Histoni ir pozitīvi lādēti, bet DNS — negatīvi.

1.2. Ribonukleīnskābe (RNS) un protīni

Ribonukleīnskābe (RNS) molekulārs organisks savienojums (nukleīnskābe), kas tieši piedalās olbaltumvielu biosintēzē, kuru veic ribosomas. Ribonukleīnskabe sastāv no 4 veidu nukleotīdiem, kas satur purīnbāzi vai pirimidīnbāzi (adenīnu, guanīnu, citozīnu, uracilu), ogļhidrātu ribozi un fosforskābi. Nukleotīdi, savstarpēji savienoti garā virknē, veido ribonukleīnskābes molekulu. Nukleotīdu secību ribonukleīnskābes molekulā nosaka dezoksiribonukleīnskābe (DNS), kas ir matrice, uz kuras pamata sintezējas ribonukleīnskābe.

Pazīstami 3 ribonukleīnskābes veidi: ribosomu, matrices un transporta ribonukleīnskābe. Ribosomu ribonukleīnskābe ir galvenā ribosomu uzbūves viela. Katrai olbaltumvielai atbilst sava matrices ribonukleīnskābe, kas veidojas šūnā uz dezoksiribonukleīnskābes matrices. Tādējādi ģenētiskā informācija par konkrētās olbaltumvielas sintēzi no dezoksiribonukleīnskābes tiek nodota matrices ribonukleīnskābei (transkripcija). Katrai aminoskābei matrices ribonukleīnskābes molekulā atbilst 3 cits citam sekojoši nukleotīdi (triplets). (3)

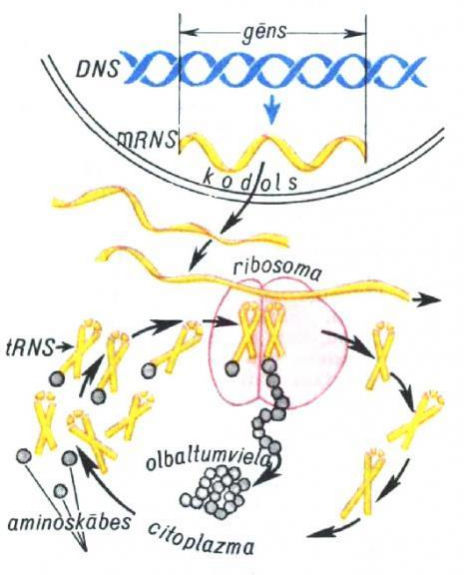
Tripletu un aminoskābju savstarpējās atbilstības sistēmu sauc par ģenētisko kodu. No šūnas kodola matrices ribonukleīnskābe nokļūst citoplazmā, kur sākas olbaltumvielu sintēzes process. Transporta ribonukleīnskābe saista un novieto aminoskābes noteiktā secībā uz matrices ribonukleīnskābes atbilstoši ģenētiskajam kodam. Katrai no 20 dažādām aminoskābēm, kas ietilpst olbaltumvielu molekulā, atbilst sava transporta ribonukleīnskābe. Informācija, kas ierakstīta matricas ribonukleīnskābē, tiek pārkodēta olbaltumvielu molekulas aminoskābju secībā (translācija).

Olbaltumvielas jeb proteīni, ir biopolimēri, molekulāri savienojumi, ko veido līdz pat 20 dažādu α-aminoskābju saturošas lineāras virknes, kurās aminoskābes savstarpēji saistītas ar peptīdsaitēm jeb amīdsaitēm. Polipeptīds ir olbaltumvielas molekula veidošanās procesā, kas savijusies lodveida (globulārā) vai pavedienveida (fibrilārā) formā.

Olbaltumvielas veido ap 45% cilvēka sausnes un ir dzīvo šūnu galvenā sastāvdaļa. Olbaltumvielas arī veic daudzas organismu bioloģiskās funkcijas.

Olbaltumvielas iedala:

- vienkāršajās olbaltumvielās (proteīnos), kas veidotas tikai no vienas vai vairākām aminoskābju virknēm, piemēram, albumīns;
- saliktajās olbaltumvielās (proteīdos), kurās bez aminoskābju virknes/-ēm sastāvā ir arī citas vielas, piemēram, hemoglobīns).



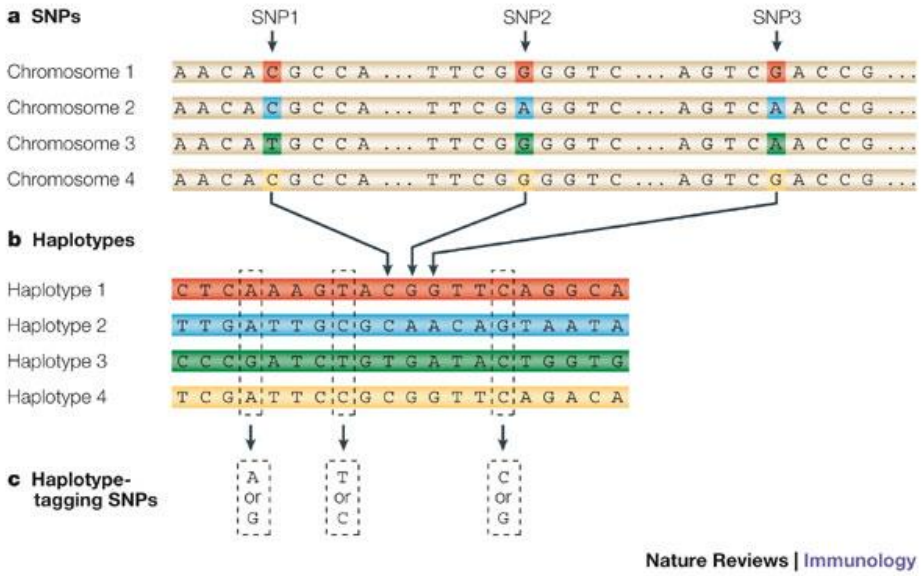
Ilustrācija 2 RNS shēma (3)

1.3. Viena nukleotīda polimorfisms (SNP)

Viena nukleotīda polimorfisms (SNP) ir biežāk sastopamā DNS sekvences variācija cilvēka genomā, kad viens nukleotīds – A, T, C vai G – genomā atšķiras starp vienas sugas indivīdiem. Lai gan katrā sekvences posma pozīcijā jebkura no četrām nukleotīda bāzēm ir iespējama, viena nukleotīda polimorfismi parasti ir abās alēlēs. Ja parādās viena nukleotīda neatbilstība, tā samazina DNS elektroķīmisko potenciālu. Ņemot vērā mutācijas mehānismu, viena nukelotīda polimorfismus var iedalīt divos veidos. Viens no tiem ir tranzīcija, kad nomainās purīns uz purīnu (A↔G) vai pirimidīns uz pirimidīnu (C↔T). Otrs ir transversija, kad izmainās vai nu purīns – pirimidīns, vai nu pirimidīns – purīns (A↔C, A↔T, G↔C, G↔T). Tomēr visbiežāk sastopamā variācija genomā ir C↔T tranzīcija. Lielākā daļa SNP ir sastopami genoma nekodējošajos reģionos un daži SNP lokalizēti šajos rajonos var ietekmēt gēnu ekspresiju. SNP, atrodoties gēna regulējošos reģionos, var ietekmēt transkripciju, tādējādi mainot attiecīgo olbaltumvielu ekspresiju. Kodējošajos reģionos viena nukleotīda polimorfismus iedala:

Nesinonīmie SNP, kuri maina olbaltumvielu produktu aminoskābju sekvenci. Šīs mutācijas var izmainīt gēnu ekspresiju gan RNS, gan proteīnu līmenī.

Sinonīmie SNP, kuri neietekmē produktu primāro sekvenci. Iespējams, vairākas sinonīmās jeb „klusās" mutācijas pārveido gēnu ekspresiju vai olbaltumvielu foldingu.



Ilustrācija 3 Viena nukleotīda polimorfisma (SNP) shēma (5)

Populācijas ģenētikas un saistības pētījumu nolūkos, SNP bieži klasificē sekojošos tipos:

1. tips – iesaistītas nesinonīmas izmaiņas attiecībā uz kodējošo daļu, tā ir nekonservatīva izmaiņa;

2. tips – kodējošo reģionu robežās, nesinonīma, bet konservatīva izmaiņa;

3. tips – variācija kodējošajā sekvencē, bet sinonīma;

4. tips – izmaiņa nekodējošajā 5' sekvencē;

5. tips – variācija nekodējošajā 3' sekvencē;

6. tips – citos nekodējošajos reģionos

Ģenētiskajai analīzei visnoderīgākais ir SNP 1. tips, tāpēc ka tam piemīt fenotipa jeb funkcionālās īpašības.

SNP ir ideāli marķieri, lai noteiktu gēnu saistības ar kompleksām slimībām, tāpēc ka tie ir blīvi izvietoti cilvēka genomā (1 SNP uz aptuveni 500 - 1000 bāzu pāriem) un ir pieejams plašs metožu klāsts to genotipēšanai. SNP arī ietekmē organisma atbildes reakcijas uz medikamentiem vai vides ķimikālijām. Dati liecina par to, ka ģenētiskie polimorfismi veido 15-30% no kopējā zāļu reaģēšanas variāciju skaita. SNP ir arī liela ietekme visās zāļu izstrādes fāzēs, sākot ar mērķa identifikāciju līdz klīniskajiem pētījumiem. Šo polimorfismu analīze palīdz arī pielāgot medikamentus vai medikamentu grupas konkrētam genotipam, kas ir farmakogenomikas pamatprincips

2. Viena nukleotīda polimorfisma (SNP) meklēšana un analīze

2.1. Expectation – Maximisation EM algoritms.

EM algoritms ir efektīva procedūra, lai aprēķinātu maksimālās ticamības aprēķinus, ja tieša maksimizēšana novērotajiem datiem, visticamāk, nav iespējams, un kādai daļai no datiem, iespējamas līdzīgas datu struktūras.

EM algoritms sastāv no 2 soļiem – E un M. E solī aizpilda trūkstošos datus, kuri, ja vienreiz tikuši atjaunoti, to parametri tiek pielāgoti tālāk M solī, kur tos analizē.

2.1.1. EM algoritms

Izvēlas X kas ir iespējamie dati un θ, kas ir interesējošie dati (4).

Izvēlas novērotajiem datiem iespējamo funkciju, ko definē ar $L(\theta) = \ln(f(X|\theta))$.

Kamēr vien *ln(.)* ir augoša funkcija, funkcijas *L(θ)* vērtība θ ir maksimizēta. Ne vienmēr ir vienkārši maksimizēt funkciju *L(θ),* tad tiek izmantots EM algoritms, kas ļauj iteratīvi palielināt *L(θ)* vērtību līdz maksimumam. θ_n ir n-tās iterācijas θ vērtība. Z vērtība būs diskrēta, tie ir iespējamie dati. Tad maksimizētā L(θ) vērtība būs vienāda ar $\ln f(X|\theta) - \ln f(X|\theta_n)$ un

$$f(X|\theta) = \sum_z f(X|z,\theta) f(z|\theta)$$

$$\begin{aligned} L(\theta) - L(\theta_n) &= \ln\Big(\sum_z f(X|z,\theta) f(z|\theta)\Big) - \ln f(X|\theta_n) \\ &= \ln\Big(\sum_z f(z|X,\theta_n) \frac{f(X|z,\theta) f(z|\theta)}{f(z|X,\theta_n)}\Big) - \ln f(X|\theta_n) \quad (2.1.1) \end{aligned}$$

Pēc Jensena nevienādības $\ln \sum_{i=1}^{n} \lambda_i x_i \geq \sum_{i=1}^{n} \lambda_i \ln(x_i)$

Konstantei $\lambda \geq 0$ $\sum_{i=1}^{n} \lambda_i = 1$

Ja $f(z|X,\theta_n \geq 0$ un $\sum_z f(z|X,\theta_n) = 1,$ tad var pārrakstīt iepriekšējo vienādojumu ar konstantēm $f(z|X,\theta_n)$,

$$\begin{aligned} L(\theta) - L(\theta_n) &\geq \sum_z f(z|X,\theta_n) ln\Big(\frac{f(X|z,\theta) f(z|\theta)}{f(z|X,\theta_n)}\Big) - ln f(X|\theta_n) \\ &= \sum_z f(z|X,\theta_n) \ln\Big(\frac{f(X|z,\theta) f(z|\theta)}{f(z|X,\theta_n) f(X|\theta_n)}\Big) \\ &\triangleq \Delta(\theta,\theta_n) \end{aligned}$$

$$L(\theta) \geq L(\theta_n) + \Delta(\theta,\theta_n)$$

$$l(\theta|\theta_n) \triangleq L(\theta_n) + \Delta(\theta,\theta_n)$$

$$L(\theta) \geq l(\theta|\theta_n)$$

Funkcija $l(\theta|\theta_n)$ ir saistīta ar iespējamo funkciju *L(θ)*. EM algoritms izvēlas θ_{n+1} vērtību, kura ir *L(θ)* maksimums.

$$\begin{aligned} l(\theta_n|\theta_n) &= L(\theta_n) + \Delta(\theta_n|\theta_n) \\ &= L(\theta_n) + \sum_z f(z|X,\theta_n) \ln(\frac{f(X|z,\theta_n)f(z|\theta_n)}{f(z|X,\theta_n)f(X|\theta_n)}) \\ &= L(\theta_n) + \sum_z f(z|X,\theta_n) \ln(\frac{f(X,z|\theta_n)}{f(X,z|\theta_n)}) \\ &= L(\theta_n) + \sum_z f(z|X,\theta_n) \ln 1 \\ &= L(\theta_n) \end{aligned}$$

Jebkuram θ_n=θ $l(\theta|\theta_n)$sakrīt ar $L(\theta_n)$

$$L(\theta_{n+1}) \geq l(\theta_{n+1}|\theta_n) \geq l(\theta_n|\theta_n) = L(\theta_n)$$

Tā rezultātā iegūst

$$\begin{aligned} \theta_{n+1} &= \arg\max_\theta\{l(\theta|\theta_n)\} \\ &= \arg\max_\theta\{L(\theta) + \Delta(\theta,\theta_n)\} \\ &= \arg\max_\theta\{\ln(f(X|\theta_n) + \sum_z f(z|X,\theta_n) \ln(\frac{f(X|z,\theta)f(z|\theta)}{f(z|X,\theta_n)f(X|\theta_n)})\} \\ &= \arg\max_\theta\{\sum_z f(z|X,\theta_n) \ln(f(X|z,\theta)f(z|\theta))\} \\ &= \arg\max_\theta\{\sum_z f(z|X,\theta_n) \ln(\frac{f(X,z,\theta)}{f(\theta)})\} \\ &= \arg\max_\theta\{E_{z|X,\theta_n} \ln f(X,z|\theta)\} \end{aligned}$$

2.1.2. EM algoritma pamatdarbības:

Inicializē θ_0 saskaņā ar iepriekš zināmo par optimālo parametru vērtībām, kurām vajadzētu būt nejauši izvēlētām.

Izvēlas iterāciju θ_n kur n = 1, 2, …

E – solis: aprēķina nosacīto iespējamo vērtību $Q(\theta,\theta_n) = E_{z|X,\theta_n} \ln f(X,z|\theta)$

M – solis: maksimizē $Q(\theta,\theta_n)$ iegūstot θ_{n+1} kur $Q(\theta_{n+1},\theta_n) \geq Q(\theta,\theta_n)$

Soļi E un M pārmaiņus atkārtojas līdz iegūst $|L(\theta_{n+1}) - L(\theta_n)|$ mazāku nekā norādītā tolerances kļūda, piemēram 10^{-6}.

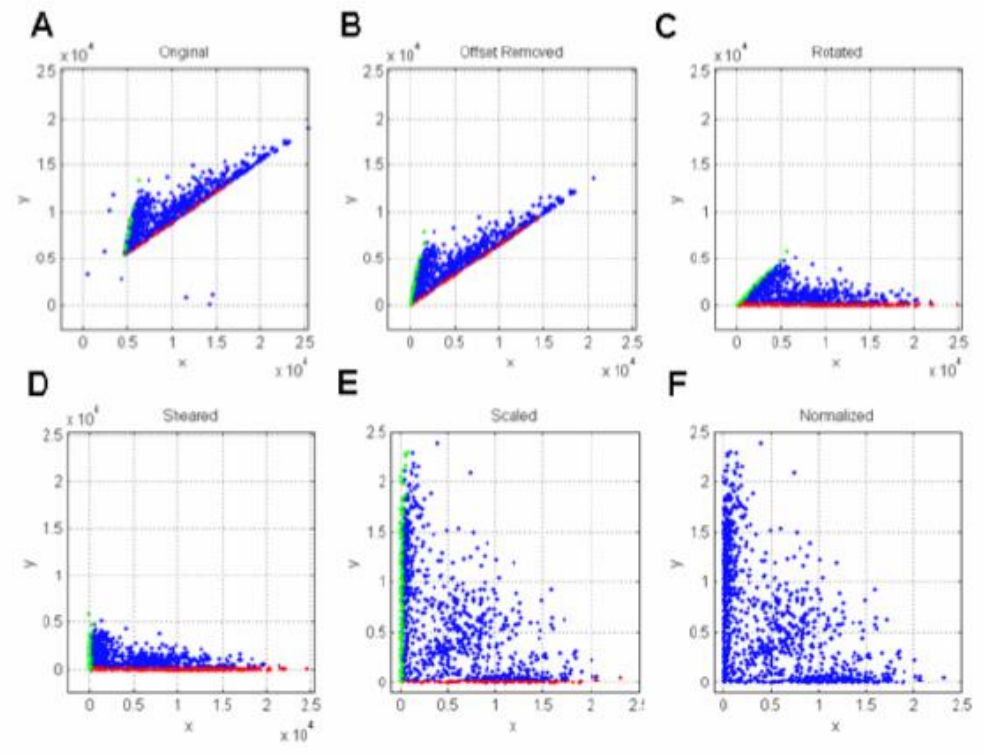
2.2. Normalizācija

Normalizācija ir process, lai samazinātu atšķirības starp masīviem (datiem), kas nav bioloģiskas izcelsmes. Normalizācijas procesam vienmēr ir nepieciešams analizēt mikromasīva datus. To piedāvā lielākais vairums genotipa salīdzināšanas algoritmu. Illuminus, GenCall un CRLMM izmanto X un Y intensitāti ievadei. Illuminus izmanto X un Y no GenCall. Gan Illuminus, gan GenCall izmanto vienu un to pašu normalizācijas metodi.

Katra algoritma pamatiespējas:

Algoritms	Normalizācija	Modelis	Datora Operētājsistēma
GenoSNP	nav	within – sample	Linux/ Windows
Illuminus	Afīnā transformācija	between – sample	Linux
CRLMM	Kvantilā normalizācija	within/between – sample	Linux/Windows/Mac
GenCall	Afīnā transformācija	between – sample	Windows

Sešu grādu brīvās afīnās (Affine) transformācijas tika izveidots 2005 gadā, autors Kermani. Normalizācijas algoritmam ir 5 posmi.



Ilustrācija 4 Pieci posmi normalizācijas procesā (4)

Pirmkārt, pirmais posms, SNP kad alēles intensitātes ir mazākas nekā pirmajos procentos vai lielākas nekā 99 procentos izslēdzot netipisko datu kopas. Šīs netipiskas datu kopas SNPs tiek izņemtas ar katru *Beadpool*. Un izņemti SNPs ar kļūdainām vērtībām. Nākamais, jeb otrais posms ir pamata novērtējums. Lai pieskaņotos kandidātu homozigotām (*homozygotes*), tie tiek

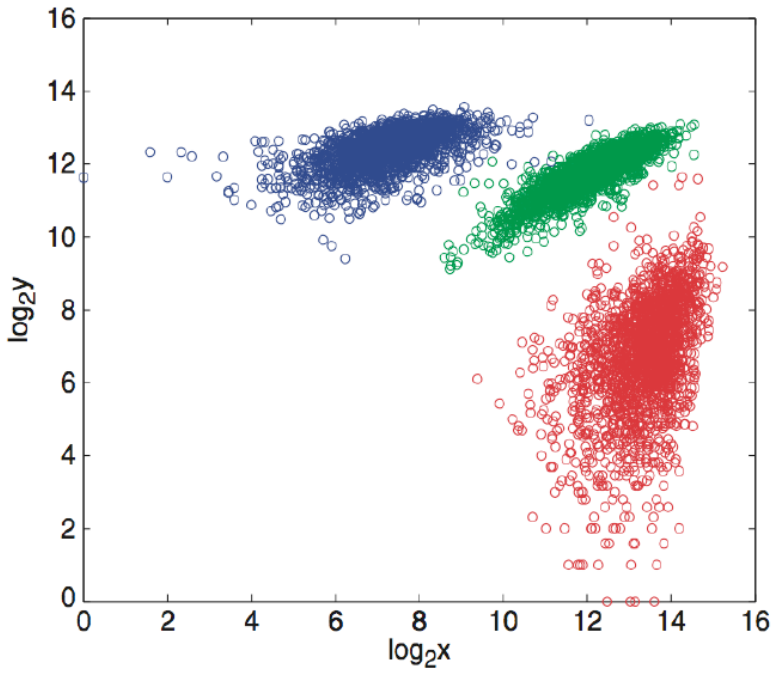
definēti kā izkaisīti punkti, kas bija vistuvāk X ass *sweep* punktiem kā kandidātu homozigotu A kontrolpunkti. Kandidātu homozigotu B kontrolpunkti tiek definēti kā līdzīgi Y ass *sweep* punkti. Divas līnijas tiek piemērotas A un B kontrolpunktiem. Divu līniju krustošanās tika identificēta, izveidojot parametrus tulkošanai. Visas signālu vērtības izkaisītajos punktos tika tulkotas izmantojot izveidotos parametrus. Pēc tulkošanas, signāla vērtība izkaisītajos punktos tika pagriezta un apgriezta.

Pēdējais posms ir palielināt vidējo caur kontroles punktu, lai noteiktu normalizētās intensitātes. Normalizētās intensitātes automātiski tiek noteiktas ar Illumina programmatūru. Abas, gan Illumina programmatūra, gan GenCall lieto vienādus normalizācijas datus datu ievadei.

CRLMM algoritms izmanto citu normalizācijas metodi, kas saucas kvantilā (*quantile*) normalizācija. Neapstrādātas X un Y intensitātes tiek normalizētas starp kanāliem un piemēriem katrā virknē (*stripes*). Normalizācija starp kanāliem var izdzēst jebkuru *due-bias* efektu.

2.3. GenoSNP

GenoSNP algoritms tika izveidots Illumnia Infinitum SNP genotipa pārbaudei. Tas pilnībā atbilst izlases prasībām un neprasa nepieciešamību pēc kontrolparaugiem, nedz parametriem, kas iegūti. Spēja salīdzināt genotipus, izmantojot tikai izlases informāciju, padara metodi lietojamu ne tikai skaitļošanai, bet arī praktiskiem pētījumiem, kas saistīti ar retiem variantiem un maziem izlašu lielumiem. Pēdējās GenoSNP versijas izmanto nenormalizētus datus no lieliem datu masīviem izmantojot kvantiles normalizācijai alēles, ja tas nepieciešams.



Ilustrācija 5 Log intensitātes apgabals visiem SNPs (4)

GenoSNP ir vienīga metode, kura ir piemērota gan paraugu modelim, gan lietojot nenormalizētas intensitātes no GenCall. Ilustrācijā (5) attēlota visu SNP intensitāte. Ar sarkano krāsu apzīmēts genotips AA, zaļā krāsa – genotipam AB, zilā krāsa – genotipam BB.

Tiek izmantotas divas metodes. Pirmā metode balstās uz standarta Expectation-Maximisation algoritmu un otrā metode balstās uz Variational Bayesian EM algoritmu.

2.4. Illuminus

Illuminus ir modeļa bāzēts genotipēšanas algoritms,tas tika izveidots 2007 gadā, autors Teo et al (Teo et al, 2007). Datu apjomus nepieciešamas normalizēt pirms darbināt algoritmu. Normalizācijas algoritms ir 6^0 (grādu) brīvas transformācijas modeļi, kas iegūti 5 posmos. Šī procedūra tiek veikta automātiski Illumina Beadstudio programmatūrā, izvadot normalizētās intensitātes. Šī modeļa metode balstās uz EM algoritmiem. Tā prasa izveidot pamata ieejas datu failu, lietojot normalizētās intensitātes, un izveidot atsevišķu ieejas failu X hromosomai ar dzimuma informāciju. Trīs komponentes Gausa mikstūras modelim tiek noteiktas un EM algoritms tiek izmantots aprēķinot katras SNP interaktīvi.

Dati tiek sadalīti divās daļās – pamata SNP un SNP hromosomai X. SNP hromosomai X piešķir genotipu atsevišķi. Tie veido uz kontrastu un stingrību bāzētu normalizēta signāla intensitāti no GenCall, kurš tiek apzīmēts ar (x_{ij}, y_{ij}). Un kontrasts (c_{ij}) un (s_{ij}) tiek definēts kā

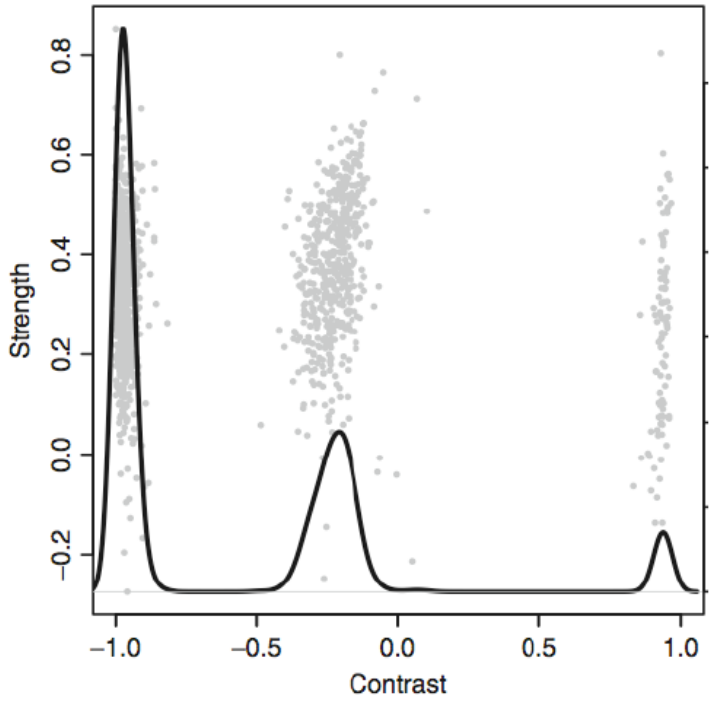
$$c_{ij} = \frac{x_{ij} - y_{ij}}{x_{ij} + y_{ij}}$$
$$s_{ij} = \log(x_{ij} + y_{ij})$$

Šī modeļa izplatībai priekš $X_{ij}=(c_{ij},s_{ij})$ lieto trīs komponenšu maisījumu no multivariācijas t distribūciju saīsinājumiem. Funkcijas X_{ij} blīvums tiek aprakstīts kā

$$F(X_{ij}) = \sum_{k=1}^{3} \lambda_k \phi_k(X_{ij}, \mu_k, \sigma_k, \upsilon_k)$$

Kur λ ir maisījuma proporcijas tiek skaitļotas izmantojot Hardy-Weinberg equilibrum, σ_k ir variācija kovariācijas matricai un v_k ir brīvības pakāpju skaits iepriekš determinēts.

Φ_k=1,2,3 tiek apzīmēts kā varbūtība trīs genotipu funkcijai: AA=1, AB=2 un BB=3.



Source of figure: Teo et al (2007)

Ilustrācija 6 Illuminus metodes masīvs (4)

Ilustrācija (6) parāda Illuminus masīvu, ņemot vērā SNP. Pelēkie punkti reprezentē novērotos datus un melnās līnijas reprezentē kodola blīvumu. Tas parāda profila dispersiju par homogēnu clasteru kopu pār heterozigotu klasteriem. Homozigotu piemēri ar kontrastu intensitāti ir atšķirīgi no heterozigotu piemēriem. Tas parāda atšķirību starp $v_1 = v_3 \leq v_3$. Balstoties uz šo X_{ij} blīvuma forma tiek skatīta kā

$$\phi_1(X_{ij}, \mu_1, \sigma_1, \upsilon_1) = \frac{f(X_{ij}, \mu_1, \sigma_1, \upsilon_1)}{1 - \int_{-\infty}^{1} f(X_{ij}, \mu_1, \sigma_1, \upsilon_1)\, dc_{ij}}$$

$$\phi_2(X_{ij}, \mu_2, \sigma_2, \upsilon_2) = \frac{f(X_{ij}, \mu_2, \sigma_2, \upsilon_2)}{\int_{-1}^{1} f(X_{ij}, \mu_2, \sigma_2, \upsilon_2)\, dc_{ij}}$$

$$\phi_3(X_{ij}, \mu_1, \sigma_1, \upsilon_1) = \frac{f(X_{ij}, \mu_1, \sigma_1, \upsilon_1)}{1 - \int_{-1}^{\infty} f(X_{ij}, \mu_1, \sigma_1, \upsilon_1)\, dc_{ij}}$$

Komponentes, kas atbilst nulles klasei. Lai uzņemtu netipisko datu kopu tiek piedāvāts kā deģenerāts mainīgais ar nulles kovrianci un variācijas kā blīvums iespējamajām vērtībām. Katrai Expectation-Maximisation procedūras inicializācijai, (m+1)th atjauninājums θ var tikt konstruēts ar atjauninājums parametriem.

$$\mu_k^{(m+1)} = (\mu_c^{(m+1)}, \mu_s^{(m+1)}) = \left(\frac{1}{n_k}\sum_i^{n_k} c_{ij}^{(m)}, \frac{1}{n_k}\sum_i^{n_k} s_{ij}^{(m)}\right)$$

$$\sigma_k^{(m+1)} = \frac{1}{n_k - 1}\begin{pmatrix} \sum_i^{n_k}(c_{ij}^{(m)} - \mu_c^{(m+1)})^2 & \sum_j^{n_k}(c_{ij}^{(m)} - \mu_c^{(m+1)})(s_{ij}^{(m)} - \mu_s^{(m+1)}) \\ \sum_j^{n_k}(c_{ij}^{(m)} - \mu_c^{(m+1)})(s_{ij}^{(m)} - \mu_s^{(m+1)}) & \sum_j^{n_k}(s_{ij}^{(m)} - \mu_s^{(m+1)})^2 \end{pmatrix}$$

where $k = 1, 2, 3$.

Priekš SNP ar nulles klasi μ=(0,0) un $\sigma = \begin{pmatrix} 10000 & 0 \\ 0 & 10000 \end{pmatrix}$

2.4.1. Hromosoma X

Hromosomas X SNP genotipēšanas algoritms tiek modificēts ar dzimumu informāciju, atzīmējot, ka genotips vīriešiem nevar būr heterozigots, jo satur tikai hromosomu X. Genotipēšanas procedūra nemainīies pēc dzimumiem $\phi_2(X_i j, \mu_2, \sigma_2, \upsilon_3, \text{gender=male}) = 0$

Skaitļojot maisījuma proporcijas, Hardy-Weinberg equilibrium tiks veikts, pieņemot, ka vīriešiem ir tikai viena alēles kopija.

2.5. CRLMM

CRLMM ir algoritms, kas oriģināli tika izveidots Affimetrix SNP datu masīviem un tika adaptēts Illumina Infinium BeadChips. CRLMM izveidoja 2007 gadā, autors Carvalho, un atjaunota 2008 gadā, autors Lin. Šīs metodes koncepcijas pamatā ir 3 komponenšu mikstūras modelis ar kubiska sadalījuma apvienojumu un 3 genotipu klasifikāciju, lietojot divpakāpju hierarhisko modeli. Kvantiles normalizācijas atbilžu dati jānormalizē pirms genotipēšanas algoritma izmantošanas.

Tiek definēts S kā vidēja intensitāte. Ir grūti atrast gadījumus, kuros intensitātes summa sniedz noderīgu informāciju, lai izveidotu log koeficientu (M), kā daudzuma kontroli. Katram masīvam (X_A, X_B) apzīmē normalizēto intensitāti no alēles A un alēles B, log rādītāji un vidējās intensitātes tiek parādītas kā

$$M = \log_2(x_A) - \log_2(x_B)$$
$$S = \frac{\log_2(x_A) + \log_2(x_B)}{2}$$

Sakarā ar šiem novērojumiem, trīs sastāvdaļu kopējs modelis ir pieejams katram paraugam $[M_i|Z_i = k] = f_k(S_i) + \varepsilon_{i,k}$, kur Z_i ir genotipa klasifikācijas mainīgais k=1,2,3. f_k kubiska funkcija piecu grādu brīvam genotipam k ar vidējo intensitāti S_i un $\varepsilon_{i,k}$ ir kļūdas lielums.

2.6. GenCall

GenCall ir genotipa algoritms ar Illumina iestrādēm. GenCall programmatūra automātiski nosaka drošības nosacījumus ievades datiem. Izejas datus no Gencall izmanto gan GenoSNP, gan Iluminus. Šai genotipu analīzē, dati no datu masīva tiek paši normalizēti kamēr datu masīva speciāla informācija tiek izmantota. Neatbilstošie dati tiek likvidēti normalizācijas procesā. Ja vienreiz dati tiek normalizēti ar sadalījuma algoritmu, tiek iedots individuāls DNS genotips. Sākumā polārās koordinātes (R, θ) tiek rediģētas divos veidos.

Rādiuss R var tikt aprēķināts ar *Manhattan* distanci:

$$R = \sum_{i}^{n} |x_i - y_i|$$

Leņķis θ tiek aprēķināts pēc *Cartesian* standarta konversācijas, polārās koordinātas izsakot:

$$\theta = f(\tan^{-1}(\frac{y_i}{x_i}))$$

Kur (x_i; y_i) ir normalizētas X un Y intensitātes priekš SNP i vērtības. Katram SNP nosaka atrašanās vietu genotipa klasterī polārajās koordinātēs (R, θ), klāsteri var tikt definēti kā labākais klāsteru sadalījums, izmantojot Hapmap datu kopas. GenCall rezultātam tiek iedots ticamības līmenis, kas tiek piešķirts katram sadalījumam. Gencall ietver tikai trīs klāsterus, kuru apzīmējums ir 3 genotipi: AA, AB, BB. Ja sadalījums nav definēts, tad GenCall rezultāts ir mazāks kā 0,15.

3. SNP algoritmu salīdzinājums

3.1. Dažādu SNP algoritmu apstrādes procesi

Līdzinājums (alignment) ir būtisks un izšķirošs solis jebkurai NGS datu analīzei, arī SNP. Ilustrācijā (6) apskatāmi algoritmu darbības daļas.

	SOAPsnp	Atlas-SNP2	SAMtools	GATK
Version	1.03	1.2	1.1.18	1.6
Format of aligned reads	SOAP output	SAM/BAM	BAM	SAM/BAM
Duplicate reads	Penalty	Remove using Atlas-SNP-mapper	Removed	Remove using picard [41]
Reads with multiple-hit	Remove	Keep all hits	Keep all hits	Keep all hits
Quality recalibration	Yes	No	Yes	Yes
Realignment	No	No	Yes	Yes

Ilustrācija 7 Algoritma izpildes daļas (13)

Pirmajā solī visas programmas Atlas-SNP2, Samtools un GATK noņem dubultos ierakstus, atstājot tikai vienu, kas būs labākais kvalitātes salidzināšanai. Bet SOAPsnp nolasa datus, lai samazinātu dubultos datus.

Lai atlasītu ierakstus dažādās vietās genomā, SOAPsnp atlasa tikai unikālos ierakstus, ierakstus ar labāko vietu (izkārtojums ar pēdējo neatbilstību). Pārējām programmām – Samtools, GATK un Atlas-SNP2 nav īpaša veida kā tiek akceptēti rezultāti.

Lai pārliecinātos, ka kvalitātes noteikšanai katru kārtošanas secības kļūdu īpatsvars atspoguļo patieso, SOAPsnp SAMtools un GATK pārkalibrē kvalitātes noteikšanas rezultātus NGS platformai. Galvenie faktori, piemēram, kvalitātes rādītāji, sekvencēšana cikli, un alēles tipi, visi tiek apskatīti.

Lai risinātu jautājumus, kas saistīti ar indels SAMtools un GATK pārvērtēšana ir solis, lai nodrošinātu precīzu variantu noteikšanu. Jo īpaši, GATK konstruē haplotipus, ko vislabāk varētu pārstāvēt kādā reģionā saskaņā ar labāko haplotipu. Savukārt SOAPsnp Atlas-SNP2 un neizmanto īpašu indel pielīdzināšanas algoritmu. SOAPsnp autori ir veikuši simulācijas ar 10,000 indels, un ir pierādīts, ka tikai 0.6%, kas ir nobīdītas indels lasa, un tikai 0.03% no tiem, kas ir nepareizi SNPs ir saglabāti izejas datu SNP genotipa noteikšanai.

3.2. SNP calling

Lai norādītu secību, izmantojot SNPs sekvencēšanas dati un to kvalitātes rādītāji, visās četrās SNP analīzes programmās piemēro Bayesian metodi. SOAPsnp SAMtools, un GATK-UGT, lai aprēķinātu varbūtību katram iespējams genotipu, un pēc tam izvēlass genotipu ar vislielāko varbūtību (pH) kā konsensa genotipu. SNP ir konkrēta pozīcija, ja tās atšķiras no references genotipa. Rezultātā gan SOAPsnp, gan SAMtools phred-like kvalitātes vērtējums, kas parāda SNP precizitāti, aprēķina ka log10 − 10 [HP − 1]. Atšķirīgi no pārējiem trīs algoritmiem, Atlas-SNP2 aprēķina varbūtību katram variantam alēlē atškīrībā no genotipa. Pēc tam pēc skaita attiecība pret atsauces, kas lasa un lasa par visticamāko variantu. Atkarībā no tā,

	SOAPsnp	Atlas-SNP2	SAMtools	GATK
Version	1.03	1.2	1.1.18	1.6
Format of aligned reads	SOAP output	SAM/BAM	BAM	SAM/BAM
Duplicate reads	Penalty	Remove using Atlas-SNP-mapper	Removed	Remove using picard [41]
Reads with multiple-hit	Remove	Keep all hits	Keep all hits	Keep all hits
Quality recalibration	Yes	No	Yes	Yes
Realignment	No	No	Yes	Yes

Ilustrācija 8 Dažādas mērvienības algoritmos (13)

kas katram Bayesian SNP calling, programma izmanto atšķirīgas mērvienības SNP calling procedūrās. Ilustrācijā (8) var apskatīt dažādas mērvienības calling SNP algoritmos.

Vairākiem kopējiem rādītājiem, nereti ir līdzīgi lielākajai daļai programmu (piem., kvalitātes rādītājus, sekvencēšana cikli un alēles). Ir arī daži konkrēti parametri, atšķirīgi katram algoritmam. Īpaši Atlas-SNP2 ir vairāki unikāli rādītāji:

- alēles ir iesaistītas multi-nucleotide polimorfisma (MNP) gadījumā;
- alēles ir "swap-base", kas ir definētas situācijā, ka divas blakus esošās neatbilstības ir to nukleotīdu referencē;
- alēlēs iet blakus kvalitātes standarts (NQS), kas nozīmē, ka kvalitātes ziņā varianta alēlei ir jābūt lielākai par 20, un kvalitātes vērtējums par katru no piecām papildu bāzes abās pusēs jābūt lielākam nekā 15;
- varianta alēles segums ir vismaz 3.

SAMtools ietver divas unikālas metrikas, bāzes līnijas atkarība un neatkarība. Pirmo pārskatu par korelāciju starp bazēm, bet pēdējā pieņem, ka dažādiem virzieniem virknēs ir neatkarīga kļūdu iespējamība.

3.3. Built-in filtri

Pēc iespējamās genotipa vai alēles, vairākus iekšējos filtrus izmanto Atlas-SNP2 SAMtools GATK-UGT un, lai identificētu potenciālos SNPs. Ilustrācijā (9) var apskatīt dažādus kritērijus SNP analīzes algoritmos.

Genotipa noteikšanu rezultāti tiek atspoguļoti VCF formātā un vairāki kritēriji ir jāpiemēro, lai noteiktu galīgo genotipu:

- Abi virzieni atbalsta dažādas alēles;
- Saīsinātu vērtību procentuālās daļas varianti tiek noteikti heterozigotu vai homozigotu genotipiem. Jo īpaši, specifiskā atrašanās vietā, ja tā ir mazāka par 10% no kopējās atlases variantu, kas nosaka genotipa alēles, kas ir homozigoti attiecībā uz šo vietu. Ja procentu vērtība ir starp 10% un 90%, heterozigotu genotips piešķir vietu, ja vērtība ir 90%, tas ir augstāks nekā hromosomā tiek noteikts kā homozigotu variants.
- Binomiālo testu izmanto, lai noteiktu genotipu īpašības un varbūtību, lai norādītu, cik drošs ir algoritmu, apskatot šo kā variantu.

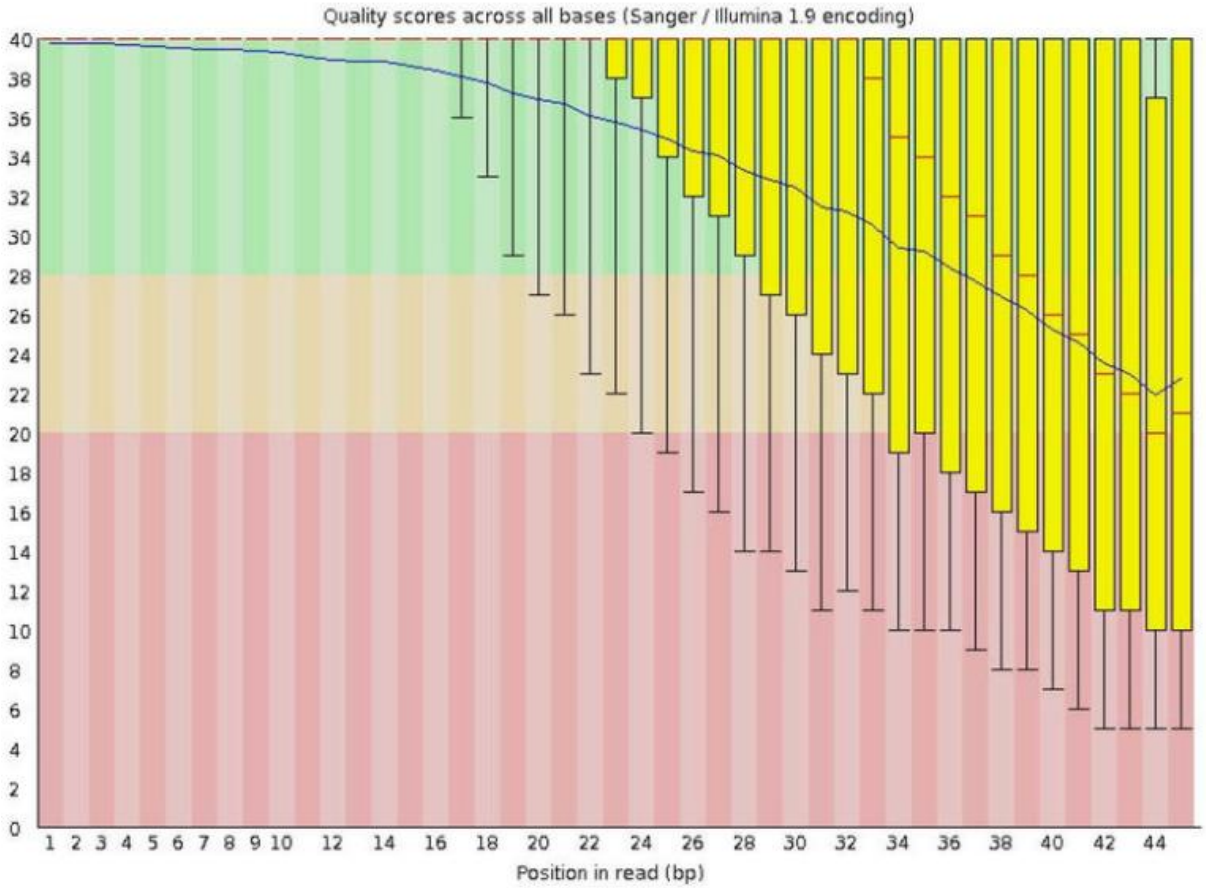
	SOAPsnp	Atlas-SNP2	SAMtools	GATK - UGT
Quality score	No	Yes	Yes	Yes
Strand bias	No	Both strands must be covered by variant allele	Yes	Yes
Coverage limits	No	variant allele coverage ≥ 3 upper limits for coverage	Yes	No
Variant reads percentage	No	Heterozygous: ≥ 10% Homozygous variant: ≥ 90%	No	No
SNP Location	No	No	No	No

Ilustrācija 9 Calling SNP algoritma kritēriji (13)

Līdzīgi, lai Atlas-SNP2 SAMtools un UGT veido SNP calling rezultātus VCF formātā.. Tādēļ iekšējā filtrēšanas kritēriji ir iekļauti GATK-UGT un VCF. VCF satur informāciju par to, kā, piemēram, SNPs neobjektivitāti, kvalitāti, dziļumu (coverage) un kvalitāti, lietotāji var filtrēt, ko sauc par SNPs starpības vērtību. Lai gan SOAPsnp iekšējās filtrēšanas, tas neparedz vairākas metrikas, lai katram SNP būtu vislabākā alēle, otra labākā alēle, un sekvences dziļums. Šos rādītājus var izmantot kā pielāgotas post-output filtrus.

3.4. Datu kopa (Dataset)

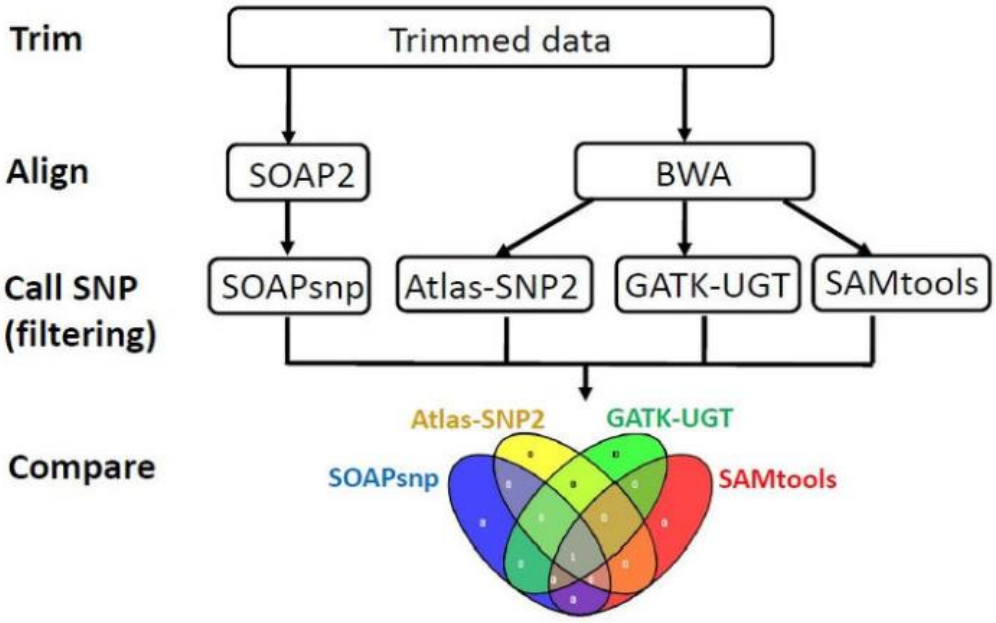
Lai izpētītu šo dažādos SNP calling rīkus, low-coverage datus, izmantojam low-coverage (1-2X) whole-genome sekvencēšanas datu kopas secību no 1 līdz 1000. genoma projekts: ERR000044. Šī datu kopa ir sekvencēšanas paraugs#NA18550 ar 6,333,357 45-bp-long ierakstiem. Ilustrācijā (10) var apskatīt sekvenēšanas kvalitātes rādītājus.



Ilustrācija 10 Sekvencēšanas kvalitātes rādītāji (13)

3.5. SNP noteikšana un salīdzināšana

SNP noteikšanai un salīdzināšanai tiek izmantoti četri posmi. Ilustrācijā (11) tiek demonstrēti šie soļi.



Ilustrācija 11 Četri soļi SNP noteikšanai un analīzei (13)

Pirmkārt, pirms kārtošanas, tiek noņemti zemas kvalitātes virknes, izmantojot izgriešanas (trim) funkciju.

Otrkārt, izlīdzināšana tiek veikta, izmantojot SOAP2 (versija 2.21) vai BWA (versija 0.6.2), izmantojot genomu. Vismaz divas neatbilstības ir atļautas katrā virknē, un tikai unikālas pozīcijas tiek norādītas izvades datos.

Treškārt, SNPs sauc par hromosomām 1. un 2. Visi SOAPsnp callings saskaņošanu veic pēc SOAP2 rezultātiem, jo SOAP2 ir tikai SOAPsnp ievades formāts. Atlas-SNP2, SAMtools, un GATK-UGT visus kārtošanas rezultātus izvada SAM formātā, ko var radīt BWA.

Visbeidzot, salīdzinām SNP calling rezultātus no četriem algoritmiem. Kopš Atlas-SNP2 nepieciešams vismaz 3X pārklājumu (coverage), lai noteiktu atšķirības un nodrošinātu taisnīgu salīdzinājumu, izmantojam tikai SNPs ar vismaz 3X katram algoritmam. Visi atrastie SNVs iedala šādās klasēs:

- Viena nukleotīda variants (SNV) identificēs tikai vienu SNP calling algoritmu.
- SNVs identificēsies ar jebkuriem diviem SNP calling algoritmiem.
- SVNs identificēsies ar jebkuriem trim SNP calling algoritmiem.
- SVNs identificēsies ar jebkuriem četriem SNP calling algoritmiem

Šī procedūra tiek veikta bez jebkādiem post-output filtriem. Tad var lietot filtrus, pamatojoties uz galvenajiem rādītājiem, katram SNP calling algoritmam, Ilustrācija (12), ar dažādu vērtību (coverage) starpību segšanu.

	Metrics
SOAPsnp	Consensus score [0, 99]
Atlas-SNP2	Posterior Probability
SAMtools	Genotype quality [0,99], QUAL
GATK-UGT	Genotype quality [0,99], QUAL, FisherStrand, HaplotypeScore, MappingQualityRankSumTest, ReadPosRankSumTest

Ilustrācija 12 Galvenie rādītāji visiem četriem algoritmiem (13)

4. Programmu salīdzinājums

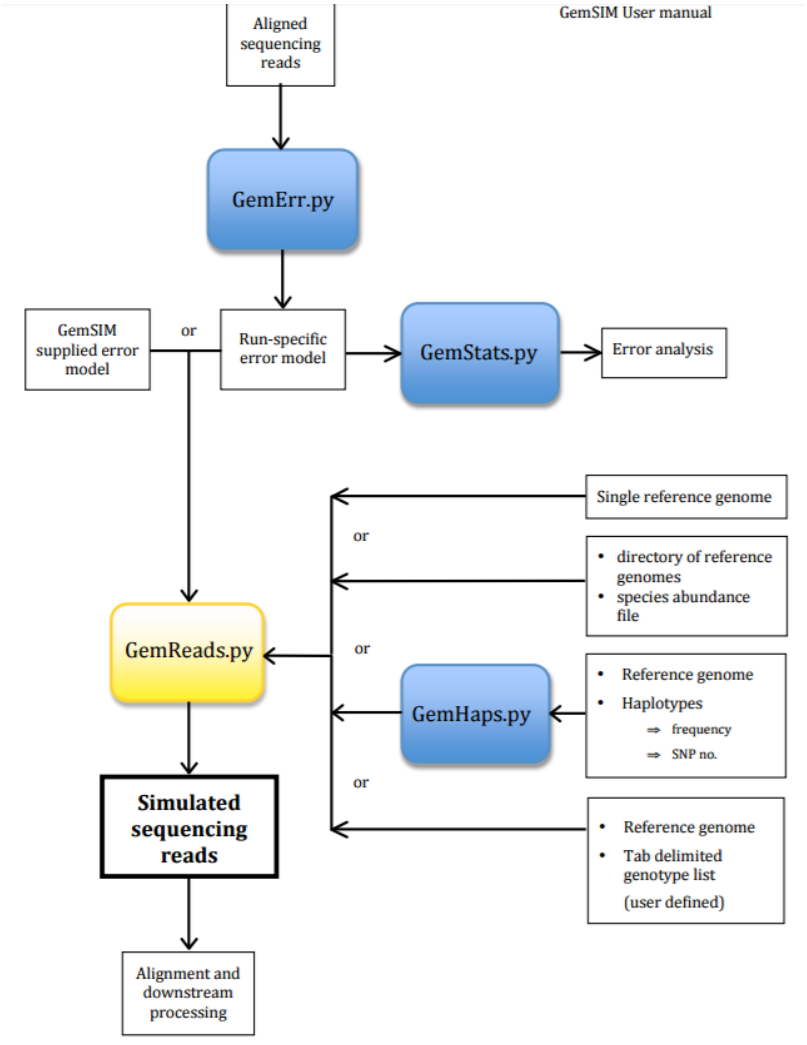
4.1. GemSIM

GemSIM ir programmatūras pakotne ar kuras palīdzību tiek simulēti reāli sekvencēšanas dati. Programma var simulēt datus no jebkurām sekvencēšanas tehnoloģijām kā izvades datus veidojot FASTQ un SAM tipa datus, ieskaitot Illumina un Roche/454 datus. GemSIm izmanto empīriskus kļūdu modeļus un sadali, izmantojot reālus datus. Simulē gan single-, gan paired-end datus no viena genoma datiem, vairākiem genomiem vai lidzvērtīgiem haplotipiem.

GemSIM tiek īstenots, izmantojot Python, kā komandrindas rīks un tas sevī ietver četras programmas:

- GemErr.py;
- GemHaps.py;
- GemReads.py;
- GemStats.py

GemSIM darbības modelis attēlots Ilustrācijā (13).



Ilustrācija 13 GemSIM modelis (5)

4.1.1. GemErr.py

GemErr ģenerē kļūdas modeļus no reāliem datiem. Ievades dati ir SAM tipa. Nolasītie dati tiek secīgi parsēti (parsed), norādot kopējo datu daudzuma un garuma sadalījumu. Visiem nolasītajiem datiem tiek norādīta sekojoša informācija:

- Nukleotīda tips un bāzes pozīcija virknē;
- Atbilstība vai neatbilstība pozīcijai;
- Indels esošā pozīcija;
- Iepriekšējās trīz bāzs pozīcijas virknē;
- Nākamās bāzes pozīcijas virknē;
- Kvalitātes rādītājus atbilstošajās vai neatbilstošajās bāzēs un ievietotajās bāzēs.

Sekvences rādītāji (sequence aligners) ieraksta kļūdas sākuma un beigās homopolimēram. Ņemot vērā šo bāzi, indels tiek ievietots tikai vienreiz garā homopolimērā beigās. Izveiotā informācija tiek saglabāta datnē un izmantota kā ievades dati GemReads.

4.1.2. GemHaps.py

Izmanto references genomu, lai izveidotu saistītus haplotipus (haplotypes), kurus izmanto kā ievada datus GemReads.py. Mutāciju pozīcijas ir nejaušas un haplotipi tiek ierakstīti datnē, ko izmanto kā ievades datus GemReads. Haplotipu frekvenci un snipu (SNP) skaitu var noteikt pats lietotājs.

4.1.3. GemReads.py

GemReads ievades dati ir dati, ko var izveidot, izmantojot GemErr. References genomam ir jābūt FASTA formātā, FASTA tipa datne vai mape. Izvades dati būs formā FASTQ. kurā pieprasītie dati būs single vai paired-end tipa. Tiek norādīts vai atsauces genoms ir cirkulārs vai lineārs un kāds kvalitates rāītājs tiks izmantots (33 vai 64, atkarībā no izvades datnes FASTQ veida) izveidotajā datnē. Pieprasītais datu skaits single vai paired-end virknēs tiek izveidots FASTQ datnēs:

- Izveidotas virknes garums un sākotnējais ievadītais virknes garums (tikai pair-end gadījumā) tiek nejausi izvēlēts no kļūdu modeļa. Nolasīsanas garums var tikt noteikts kā nemainīga vērtība.
- Izveidotas virknes virziens un atrašanās ieta tiek noteikta nejauši un virkne tiek kopēta no ievades genoma.

- Virkne tiek noteikts haplotips, ja pieejams, un tajā tiek atjaunināti SNP pēc vajadzības.
- Kļūdas tiek ieviestas pēc kļūdu modeļa, uzskaitot nolasīšans pozīciju, sequence-context simbolu virkni un pirmo un otro nolasīšanas virkni pārī, paired-ends gadījumos.
- Kvalitātes rādītājus, kas tiek noteikti, izmantojot empīroisko sadales modeli, kas iekļauts kļudas modeļa datnē.

4.1.4. GemStats.py

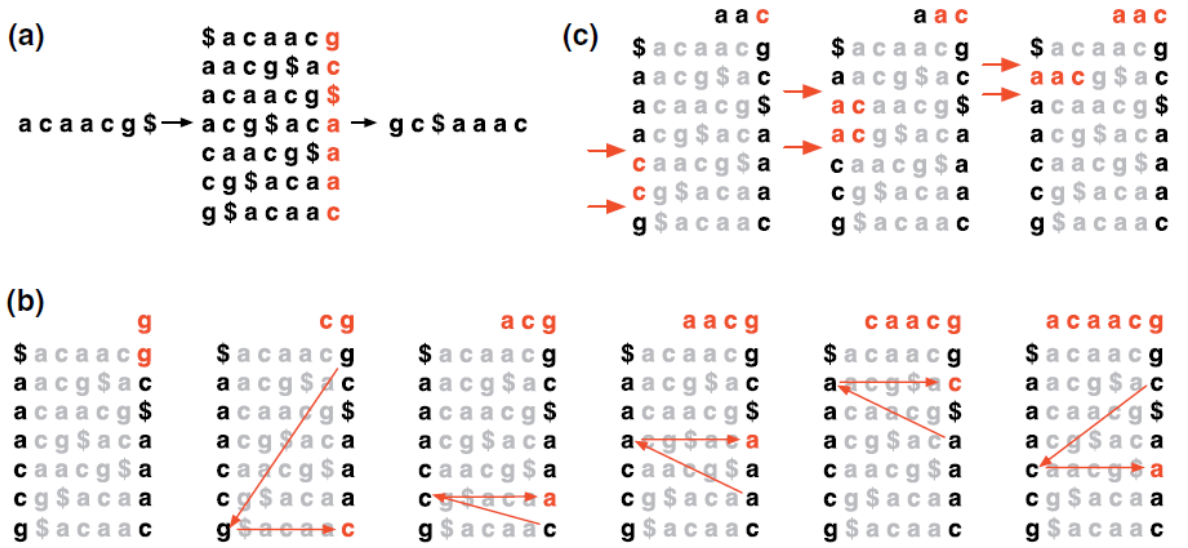
GemStats.py darbības rezultātā tiek izveidots teksta fails (vai divi paired-end datu gadījumā). Kurā ir informācija par kļūdu modeli:

- Vidējais kļūdas biežums;
- Kļūdas biežums katrā nukleotīdā;
- Kļūdas biežums lasīšanas pozīcijai;
- Jebkura nukleotīda sekvenci, kam standarta kļūdas līmenis ir lielāks pr vidējo;
- Iestrādes norma;
- Dzēšanas koeficientu.

Šo informāciju var izmantot kvalitātes kontrolei.

4.2. Bowtie2

Bowtie2 indeksē genoma references datni, izmantojot Burrows-Wheeler transformācijas (BWT) un FM indeksēšanas metodes.



Ilustrācija 14 Burrows-Wheeler transformācijas indeksēšanas metode (8)

Ilustrācijā (14) redzama Burrows-Wheelers transformācijas indeksēšanas metode:

(a) Burrows-Wheeler matrica. Kur parādīta virknes “acaacg” transformācija;
(b) Nosaka rindu sakritības diapazonu, nosakot references suffiksus, prieks virknes “aac”;
(c) Nemainīgi atkārto pēd’’ejo pirmo (LF) kartēšanu, atjaunojot oriģinalo virkni pēc Burrows-Wheelers transformāciju.

4.2.2. Bowtie2 izmantošana

Pirmais solis Bowtie2 izmantošanā ir indexu veidošana genoma references datnei. Šajā procesā ievades dati ir FASTA foremātā un index_prefix ir indeksēta datņu kopa.

bowtie2-build -f input_reference.fasta index_prefix

Otrais solis ir komanda bowtie2, kas, izmantojot indeksētās datnes, sekvencē ievadītos datus un izveido izejas datus SAM formātā.

bowtie2 -x index_prefix [-q|--qseq|-f|-r|-c] [-1 input_reads_pair_1.[fasta|fastq] -2 input_reads_pair_2.[fasta|fastq] | -U input_reads.[fasta|fastq]] -S bowtie2_alignments.sam [options]

4.3. SAMtools

SAMtools ir bibliotēka un programmmatūra pakotne, ko izmanto datu parsēšanai un datu sakārtojuma mainīšanai SAM un BAM formādu datnēs. Datus var konvertēt no citiem sakārtojuma formātiem, sakārtot un apvienot izkartojumu, izdzēst PCR dublikātus, ģenerēt pozīcijas informāciju pileup formātā, atrast SNP un short indel variantus, un apskatīt iegūtos rezultātus teksta formāta lietotnē.

Sagatavotu datni SAM formātā pārveido par BAM gtipa datni.

samtools view -bS paraugs.sam > paraugs.bam

samtools sort paraugs.bam > paraugs.sorted

Atrod SNP un short indels.

samtools mpileup -uf paraugs.sorted.bam > paraugs

bcftools view -vcg paraugs> paraugs.vcf

4.4. Bcftools

Bcftools izmanto SAMtools mpileup izejas datu pārveidošanai VCF formātā.

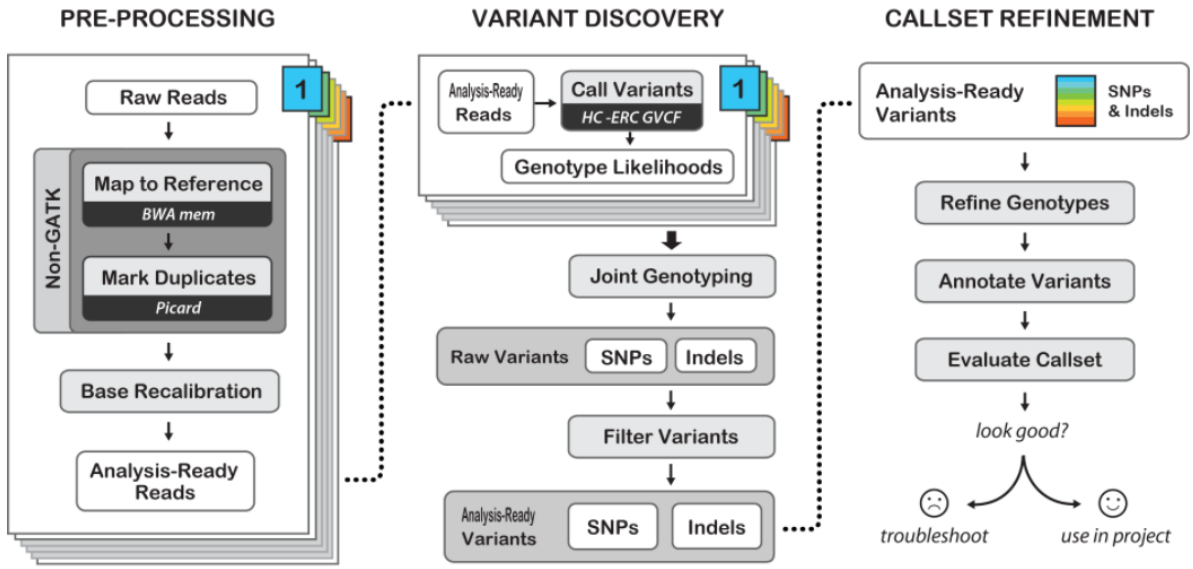
bcftools view -vcg paraugs> paraugs.vcf

4.5. GATK – GenomeAnalysisToolkit

GATK ir komandrindas rīku kopums augstas iedarbības analīzes rīki datu formātiem SAM/BAM/CRAM un VCF, īpašu uzmanību pieēršot variantu noteikšanai. GATK rīki darbojas izmantojot vienkāršus programmas JAR failus.

Pēc noklusējuma HaplotypeCaller no dotajiem genoma references datiem izveido VCF datni ar neapstrādātiem variantiem.

java -Xmx4G -jar GenomeAnalysisTK.jar -R paraugs.fasta -T HaplotypeCaller -I paraugs.bam -o raw_variants.vcf



Ilustrācija 15 GATK izmantošanas shēma (12)

4.6. Izmantotie failu formāti

4.6.1. FASTA

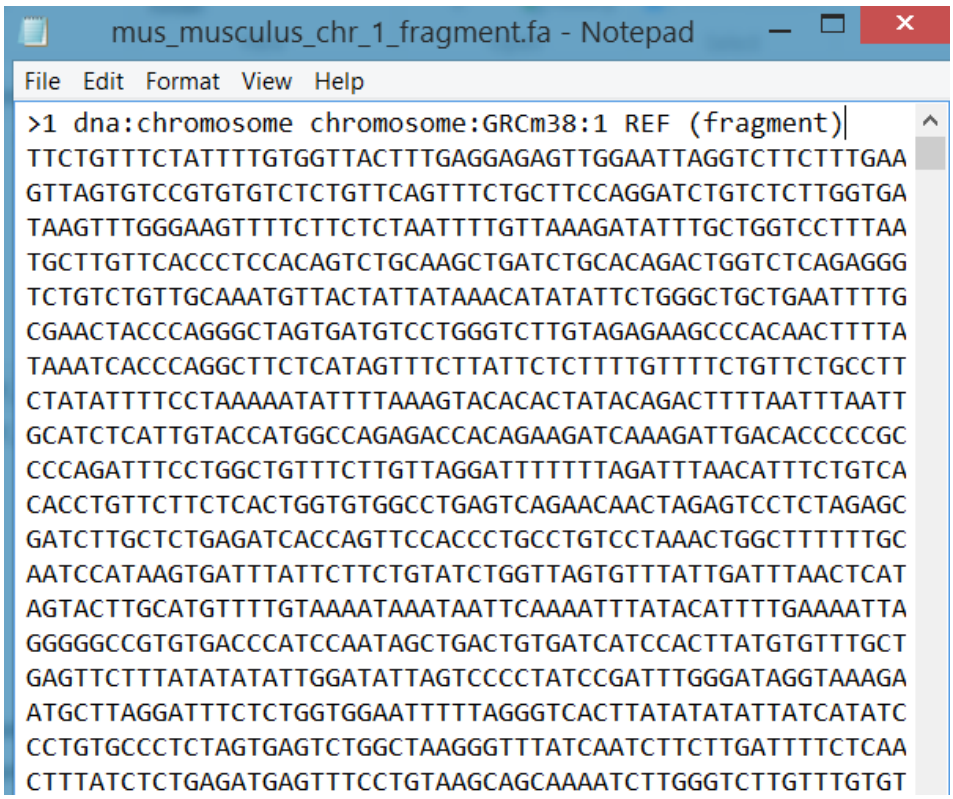
Sekvences FASTA formātā ir sekojošas:

Pirmā rinda sākas ar “>”, kam seko sekvences identifikācijas kods. Pēc tā, brīvi izvēlēts teksta apraksts sekvencei. Tālāk vienā vai vairakas rindās seko sekvences dati.

FASTA formata datnē var būs vairāk neka viena sekvence.

FASTA formāts reizēm var tikt minēts kā “Pearson” formāts (pēc FASTA programmas autora).

FASTA datnes piemērs redzams Ilustrācijā (16)



Ilustrācija 16 FASTA datne

4.6.2. FASTQ

Sekvences datne FASTQ formātā var saturēt vairākas sekvences. FASTQ datne ir teksta bāzēts datnes formāts, kas satur bioloģiskās sekvences (parasti nukleotīdu sekvences) un tās kvalitātes vērtējumus.Šis formāts visbiežāk tiek lietots augstas kvalitātes sekvencēšanas rezultātiem. FASTQ datne satur parasti četras ierakstu rindas katrai sekvencei:

- "@" simbols, kam seko sekvences identifikators un apraksts;
- Neapstrādāta sekvences rinda ar simboliem;
- "+" simbols, kam seko sekvences identifikators un paskaidrojums;
- Kvalitātes vērtības sekvencei otraja rindā.

FASTQ datnes paraugs redzams Ilustrācijā (17).

Ilustrācija 17 FASTQ datnes paraugs

4.6.3. SAM

SAM ir Sequence Alignment/Map formāts. Tā ir teksta formāta datne,kas satur galvenes sekciju un izkārtojuma sekciju. Galvenes sekcija sākas ar "@", kāmēr pārējš izkārtojuma rindām nav īpaša apzīmējuma. Katra sakārtojuma rinda satur 11 obligātus laukus būtiskus informācijas sakārtojumā, tādus kā kārtēšanas (mapping) pozīcijas un mainīga skaita iespējamus laukus, elastīgu vai specifisku informāciju.

Ilustrācijā (18) redzams Sam datnes paraugs.

```
@HD VN:1.5 SO:coordinate
@SQ SN:ref LN:45
r001   99 ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTG *
r002    0 ref  9 30 3S6M1P1I4M *  0   0 AAAAGATAAGGATA    *
r003    0 ref  9 30 5S6M       *  0   0 GCCTAAGCTAA       * SA:Z:ref,29,-,6H5M,17,0;
r004    0 ref 16 30 6M14N5M    *  0   0 ATAGCTTCAGC       *
r003 2064 ref 29 17 6H5M       *  0   0 TAGGC             * SA:Z:ref,9,+,5S6M,30,1;
r001  147 ref 37 30 9M         =  7 -39 CAGCGGCAT         * NM:i:1
```

Ilustrācija 18 SAM datnes paraugs (9)

4.6.4. BAM

BAM datne ir saspiests BGZF formāts. BGZF formāts nodrošina bloku saspiešanu virs standarta gzip datnes formāta. Katrs BGZF bloks satur standarta gzip datnes galveni ar sekojosiem standart-compliant paplašinājumiem:

- F.EXTRA bits galvenē parāda ekstra laukus tālāk.
- Ekstra lauks izmantojot BGZF, izmanto divas apakšlauka ID vērtības 66 un 6 (ascii "BC")
- BGZF ekstra lauka garuma payload (lauka LEN gzip specifikācijā) ir 2 (divi baiti)
- Payload BGZF ekstra laukam ir 16 bitu neatzīmēts vesels skaitlis endian formātā. Šis veselais skaitlis parāda izmēru, kas ir vienāds ar BGZF bloka izmēru no kā atņem 1 veselu.

Field	Description	Type	Value
magic	BAM magic string	char[4]	BAM\1
l_text	Length of the header text, including any NUL padding	int32_t	
text	Plain header text in SAM; not necessarily NUL-terminated	char[l_text]	
n_ref	# reference sequences	int32_t	
	List of reference information (n=n_ref)		
l_name	Length of the reference name plus 1 (including NUL)	int32_t	
name	Reference sequence name; NUL-terminated	char[l_name]	
l_ref	Length of the reference sequence	int32_t	
	List of alignments (until the end of the file)		
block_size	Length of the remainder of the alignment record	int32_t	
refID	Reference sequence ID, $-1 \leq$ refID $<$ n_ref; -1 for a read without a mapping position.	int32_t	[-1]
pos	0-based leftmost coordinate (= POS − 1)	int32_t	[-1]
bin_mq_nl	bin<<16|MAPQ<<8|l_read_name; bin is computed from the mapping position;[17] l_read_name is the length of read_name below (= length(QNAME) + 1).	uint32_t	
flag_nc	FLAG<<16|n_cigar_op;[18] n_cigar_op is the number of operations in CIGAR.	uint32_t	
l_seq	Length of SEQ	int32_t	
next_refID	Ref-ID of the next segment ($-1 \leq$ mate_refID $<$ n_ref)	int32_t	[-1]
next_pos	0-based leftmost pos of the next segment (= PNEXT − 1)	int32_t	[-1]
tlen	Template length (= TLEN)	int32_t	[0]
read_name	Read name,[19] NUL-terminated (QNAME plus a trailing '\0')	char[l_read_name]	
cigar	CIGAR: op_len<<4|op. 'MIDNSHP=X'→'012345678'	uint32_t[n_cigar_op]	
seq	4-bit encoded read: '=ACMGRSVTWYHKDBN'→ [0, 15]; other characters mapped to 'N'; high nybble first (1st base in the highest 4-bit of the 1st byte)	uint8_t[(l_seq+1)/2]	
qual	Phred base quality (a sequence of 0xFF if absent)	char[l_seq]	
	List of auxiliary data (until the end of the alignment block)		
tag	Two-character tag	char[2]	
val_type	Value type: AcCsSiIfZHB[20,21]	char	
value	Tag value	(by val_type)	

Ilustrācija 19 BAM datnes formāta raksturojoši elementi (9)

4.6.5. VCF

VCF datne ir teksta formāta datne. Tā satur meta datu informāciju rindiņās. Tai ir galvenes rinda un tālāk daru rindas, kas satur informāciju par genomu.

Ilustrācijā (20) redzams VCF datnes paraugs.

```
##fileformat=VCFv4.0
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=1000GenomesPilot-NCBI36
##phasing=partial
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=.,Type=Float,Description="Allele Frequency">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FILTER=<ID=q10,Description="Quality below 10">
##FILTER=<ID=s50,Description="Less than 50% of samples have data">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##FORMAT=<ID=HQ,Number=2,Type=Integer,Description="Haplotype Quality">
#CHROM POS     ID        REF ALT    QUAL FILTER INFO                              FORMAT      NA00001
     NA00002        NA00003
20     14370   rs6054257 G      A       29   PASS   NS=3;DP=14;AF=0.5;DB;H2           GT:GQ:DP:HQ 0|0:4
8:1:51,51 1|0:48:8:51,51 1/1:43:5:.,.
20     17330   .         T      A       3    q10    NS=3;DP=11;AF=0.017               GT:GQ:DP:HQ 0|0:4
9:3:58,50 0|1:3:5:65,3   0/0:41:3
20     1110696 rs6040355 A      G,T     67   PASS   NS=2;DP=10;AF=0.333,0.667;AA=T;DB GT:GQ:DP:HQ 1|2:2
1:6:23,27 2|1:2:0:18,2   2/2:35:4
20     1230237 .         T      .       47   PASS   NS=3;DP=13;AA=T                   GT:GQ:DP:HQ 0|0:5
4:7:56,60 0|0:48:4:51,51 0/0:61:2
20     1234567 microsat1 GTCT   G,GTACT 50   PASS   NS=3;DP=9;AA=G                    GT:GQ:DP    0/1:3
5:4       0/2:17:2       1/1:40:3
```

Ilustrācija 20 VCF datnes paraugs (10)

5. Pētījuma daļa

5.1. GemSIM izmantošana

Lai veiktu pētījumus, autore izmanto peles genoma datus. Sākotnēji tiek izmantota programma GemSIM. Programma darbojas ar komandrindas palīdzību Ubuntu vidē. Ar GemReads.py simulācijas programmas palīdzību tiek ģenerēts kļūdains datu modelis references genoma datnēm Mus_musculus.GRCmXX.dna.chromosome. Tiek izveidotas single-end datnes, simulējot 300000 – 6000000 100bp garus Illumina ierakstus, lietojot kļūdas modeli.

Izmantojamā komandrinda:

./GemReads.py –r musf.fasta –n 600000 –l 100 -m ill100v5_s.gzip –q 64 –o musf

Izmantotās FAST datnes paraugu var paskatīt 1.pielikumā un simulēto datu paraugu var apskatīt 2.pielikumā.

5.2. Bowtie2 izmantošana

5.2.1. Bowtie2-build

Bowtie2-build veido bowtie indekxusno DA sekvencēm, k;a izvasdes datnes veidojot sešas datnes ar suffiksiem .1.bt2, .2.bt2, .3.bt2, .4.bt2, .rev1.bt2, .rev2.bt2. šis datnes kopā veido indeksu un tiek izmantoti references atlasē. Oriģinālais FASTA fails tālāk netiek izmantots pēc tam, kad indeks ir gatavs.

Izmantojamā komandrinda:

bowtie2-build musf.fa index

Tālāk tiek izveidota SAM datne, izmantojot komandrindu:

bowtie2 -x index -U musf_single.fastq -S musf.sam

Lai apskatītu izveidotās datnes sākumu, izpilda komandrindu:

head musf.sam

Var apskatīt datnes sākumu, līdzīgu paraugam, ilustrācijā (21)

```
@HD VN:1.0  SO:unsorted
@SQ SN:gi|9626243|ref|NC_001416.1|  LN:48502
@PG ID:bowtie2  PN:bowtie2  VN:2.0.1
r1  0   gi|9626243|ref|NC_001416.1| 18401   42  122M    *   0   0   TGAATGCGAACTCCGGGACGCTCAGTAATGTGACGATAGCTGA
r2  0   gi|9626243|ref|NC_001416.1| 8886    42  275M    *   0   0   NTTNTGATGCGGGCTTGTGGAGTTCAGCCGATCTGACTTATGT
r3  16  gi|9626243|ref|NC_001416.1| 11599   42  338M    *   0   0   GGGCGCGTTACTGGGATGATCGTGAAAAGGCCCGTCTTGCGCT
r4  0   gi|9626243|ref|NC_001416.1| 40075   42  184M    *   0   0   GGGCCAATGCGCTTACTGATGCGGAATTACGCCGTAAGGCCGC
r5  0   gi|9626243|ref|NC_001416.1| 48010   42  138M    *   0   0   GTCAGGAAAGTGGTAAAACTGCAACTCAATTACTGCAATGCCC
r6  16  gi|9626243|ref|NC_001416.1| 41607   42  72M2D119M   *   0   0   TCGATTTGCAAATACCGGAACATCTCGGTAACTGCATAT
r7  16  gi|9626243|ref|NC_001416.1| 4692    42  143M    *   0   0   TCAGCCGGACGCGGGCGCTGCAGCCGTACTCGGGGATGACCGG
```

Ilustrācija 21 SAMm datnes paraugs (11)

Pirmajas rindās, kas sākas ar simbolu “@” ir datnes galvenes rindas, pārējās rindas ir SAM sakārtotas pa vienai rindai katram sakārtojumam.

5.2.2. Samtools

Samtools tiek izmantots, lai pārveidotu SAM datni par BAM datni. BAM ir binārs formāts atbilstošs SAM teksta formātam.

Izmanto komandrindu:

samtools view -bS musf.sam > musf.bam

Lai sagatavoto BAM datni sakārtotu, izmanto komandrindu:

samtools sort musf.bam -o musf.sorted.bam

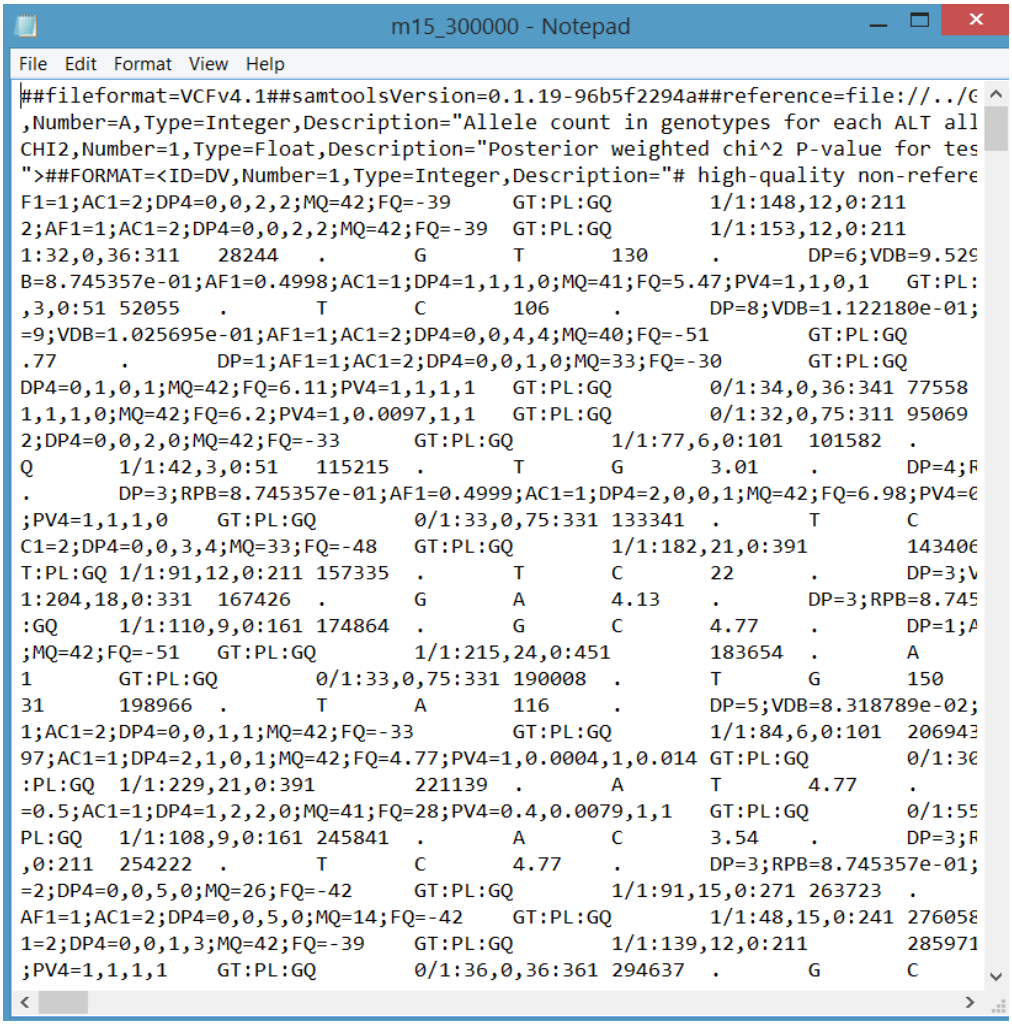
Sorted.bam datne ir formāts, kuru var izmantot, lai izveidotu iespējamoVCF formātu. Izmanto komandrindu:

samtools mpileup -uf musf.fa musf.sorted.bam | bcftools view -Ov - > musf.raw.bcf

Lai apskatītu iegūtos variantus, izmanto komandrindu, lai pārveidotu izveidoto BFC datni par VFC datni.

bcftools view –vcg musf.raw.bcf > musf.vcf

Iegūtais VCF formāts var tikt izmantots salīdzināšanai ar pārējiem datiem.



Ilustrācija 22 VCF datnes formāts

5.3. Gatk izmantošana

Gatk ir iespēja ievadīt FASTA datni un galarezultātu iegūt VCF formātā ar pāris jar komandām. Tomer ir iespēja arī ievadīt ar citām programmām iegūtos SAM vai BAM formātus VCF datņu izvadei. Rezultāts praktiski ir līdzvērtīgs abos gadījumos.

Komandrinda GATK izmantošnai:

```
java -jar GenomeAnalysisTK.jar \
   -T HaplotypeCaller \
   -R musf.fa \
   -I musf_reads.bam \
   -L 20 \
   --genotyping_mode DISCOVERY \
   -stand_emit_conf 10 \
   -stand_call_conf 30 \
   -o raw_variants.vcf
```

Vai izmantojama java -Xmx4G -jar GenomeAnalysisTK.jar -R paraugs.fasta -T HaplotypeCaller -I paraugs.bam -o raw_variants.vcf

6. Iegūto datu salīdzinājums

6.1. Mus_musculus.GRCm38.dna.chromosome.2 coverage 5 salīdzinājums ar simulatoriem

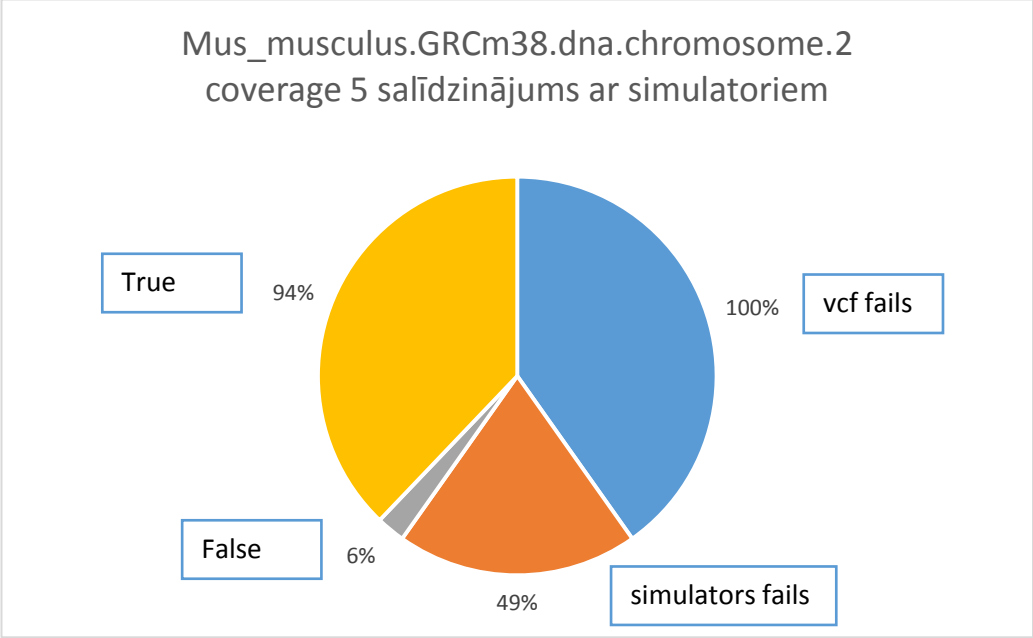
Salīdzinot datus no dažādiem genomiem, un dažādiem datu simulatoriem, vidējā datu atšķirība ir 1% – 40%.

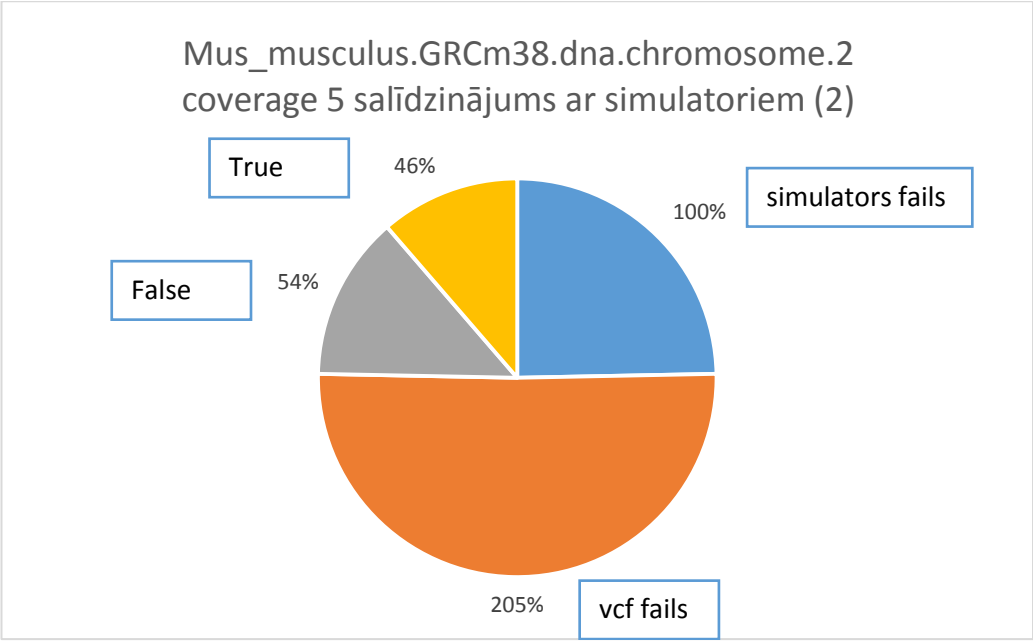
Tālāk esošajās diagrammās var redzēt, ka salīdzinot vienus un tos pašus izejas datus, rezultāti atšķiras, sākot ar dažiem procentiem līdz pat vairākiem desmitiem procentu. Tas ļauj secināt, ka katrā programmā atrastais SNP skaits ir atšķirīgs, jo katrā programmā esošais algoritms arī ir atšķirīgs.

Katrā diagramā ir parādīti abu izmantoto datņu lielumi pēc rindu skaita, kas pārveidoti salīdzinošos procentos. Abi datņu apzīmējumi visās diagrammās ir vienādās krāsās – zilā ir VCF datne, kura ir galarezultāts no divām programmām – Samtools vai GATK. Un otrā krāsa, kas diagrammā paredzēta datnei ir oranža un šī datne iegūta no simulatora teksta datnes veidā. Katrai diagrammai pie šīm krāsām ir iespēja redzēt datnes formātu.

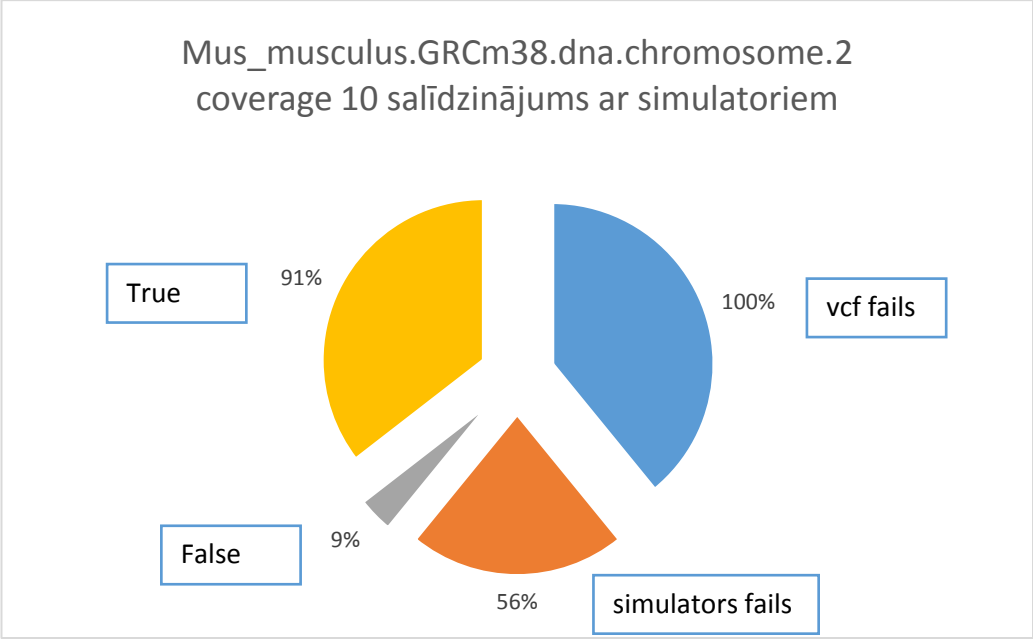
Diagrammā dzeltenai krāsai paredzēts procentuālais daudzums SNP, kas abās datnēs sakrīt. Pelēkā krāsa apzīmē tos SNP, kas abās datnēs nesakrīt.

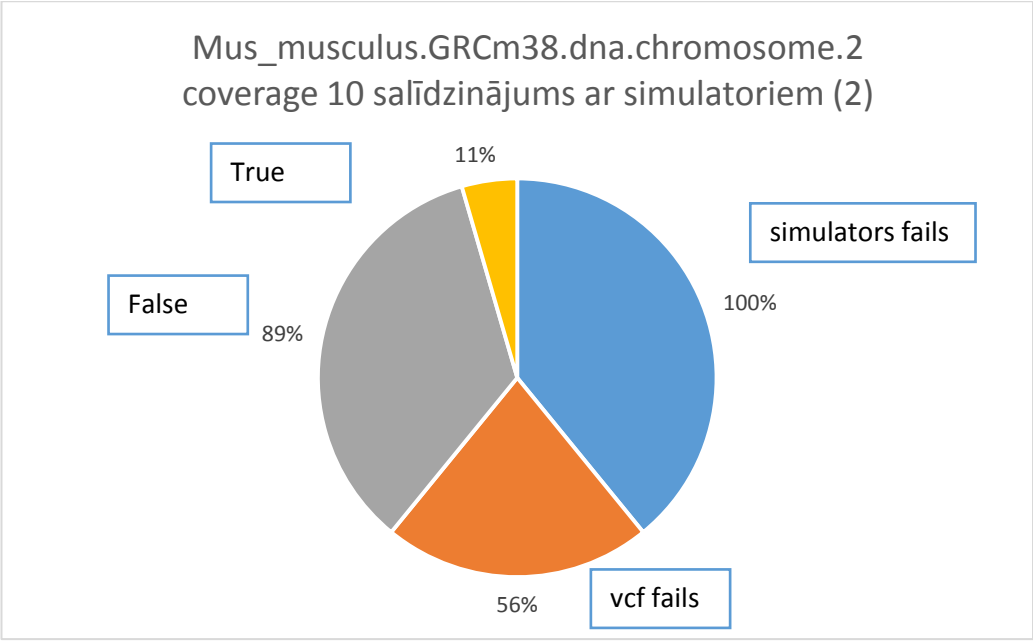
Tādā veidā vienā diagrammā ir iespēja redzēt salīdzināmās datnes un SNP salīdzinājumu procentos.





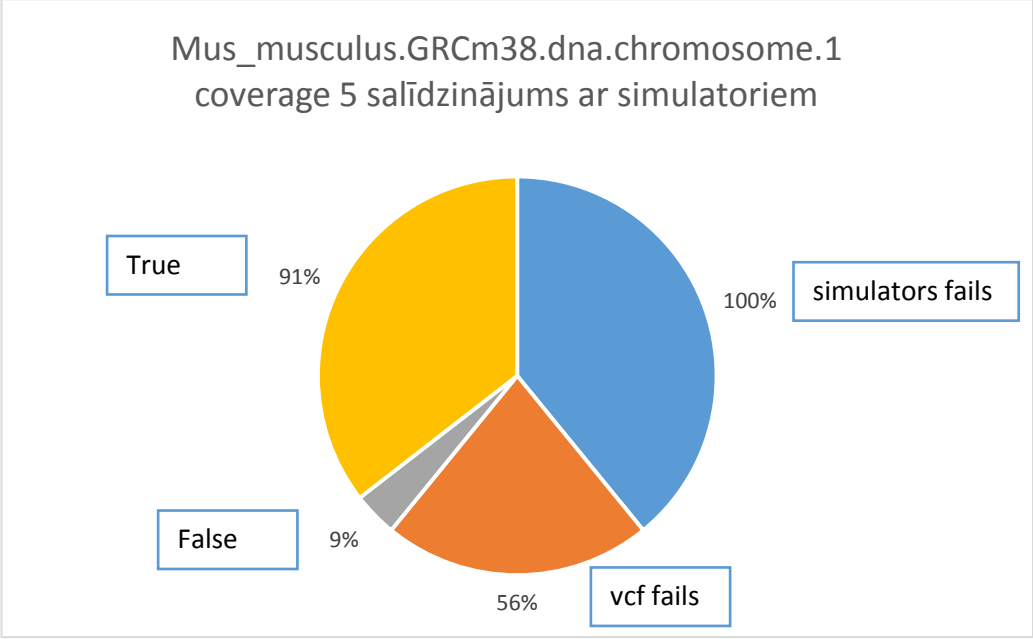
6.2. Mus_musculus.GRCm38.dna.chromosome.2 coverage 10 salīdzinājums ar simulatoriem

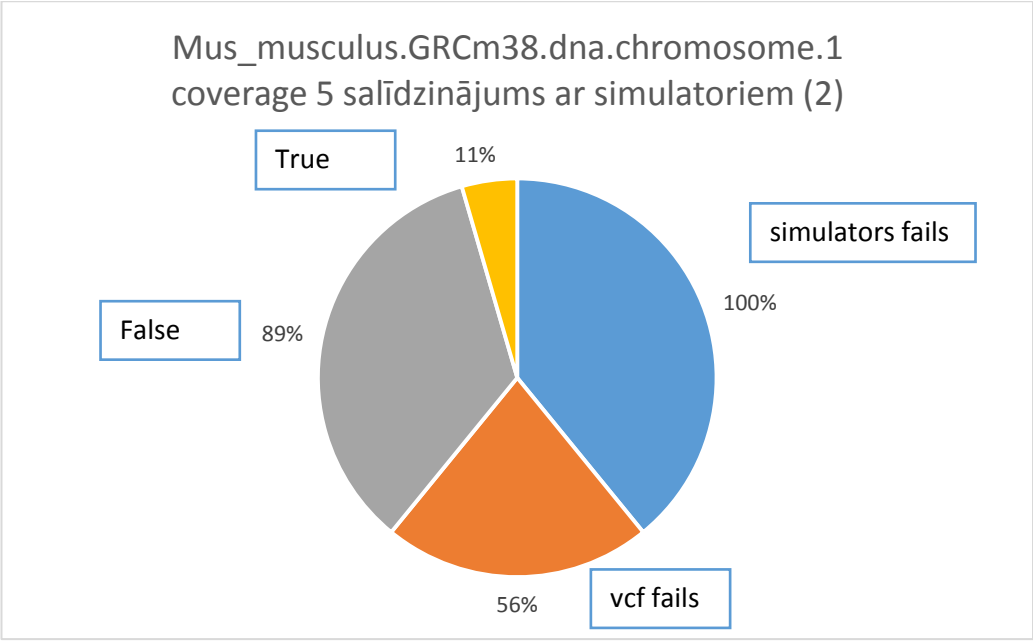




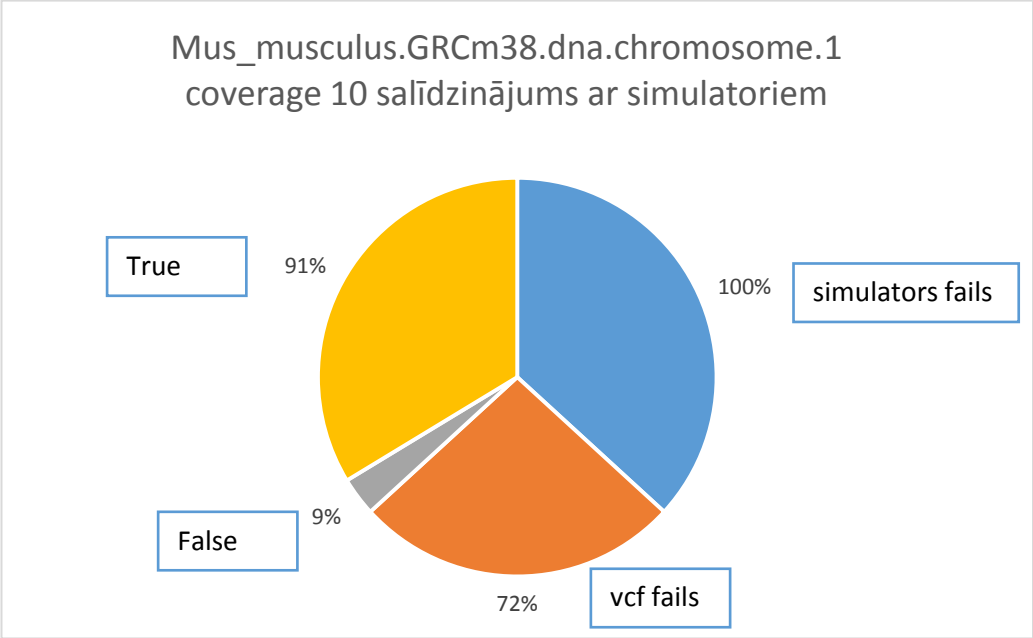
6.3.

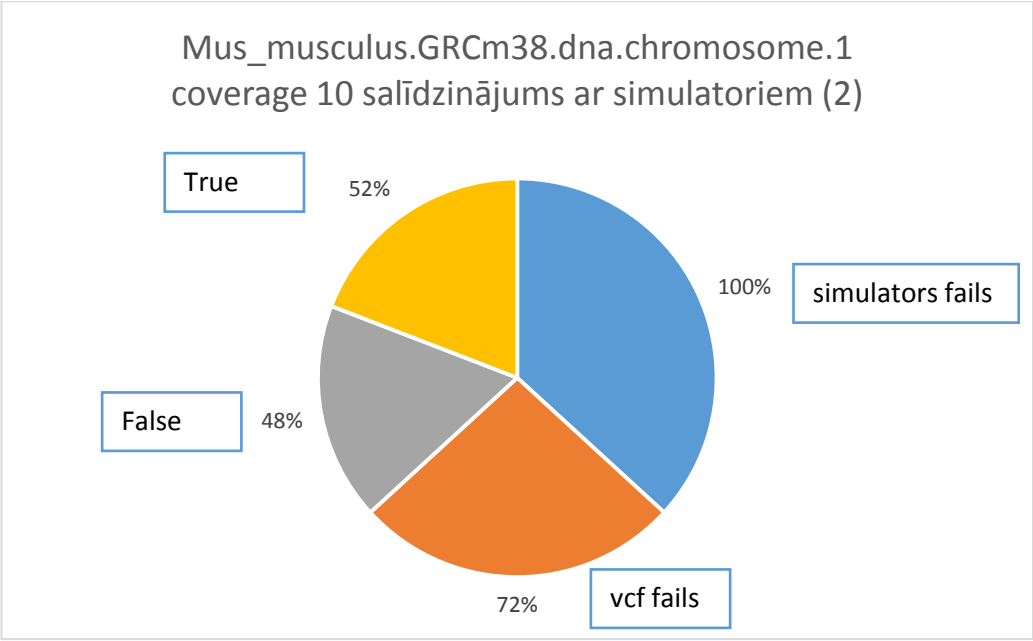
6.4. Mus_musculus.GRCm38.dna.chromosome.1 coverage 5 salīdzinājums ar simulatoriem



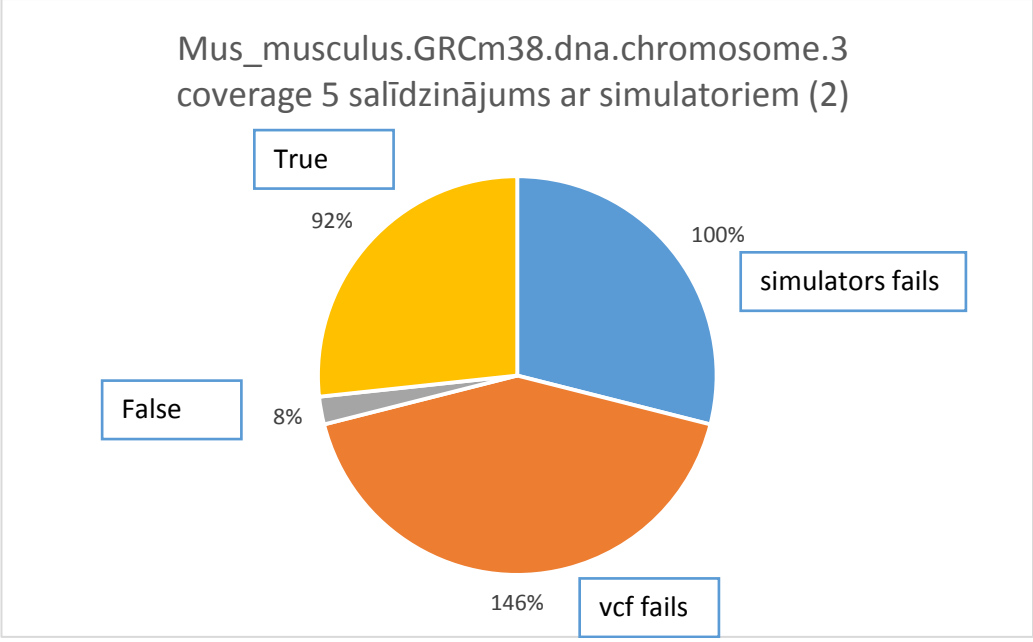


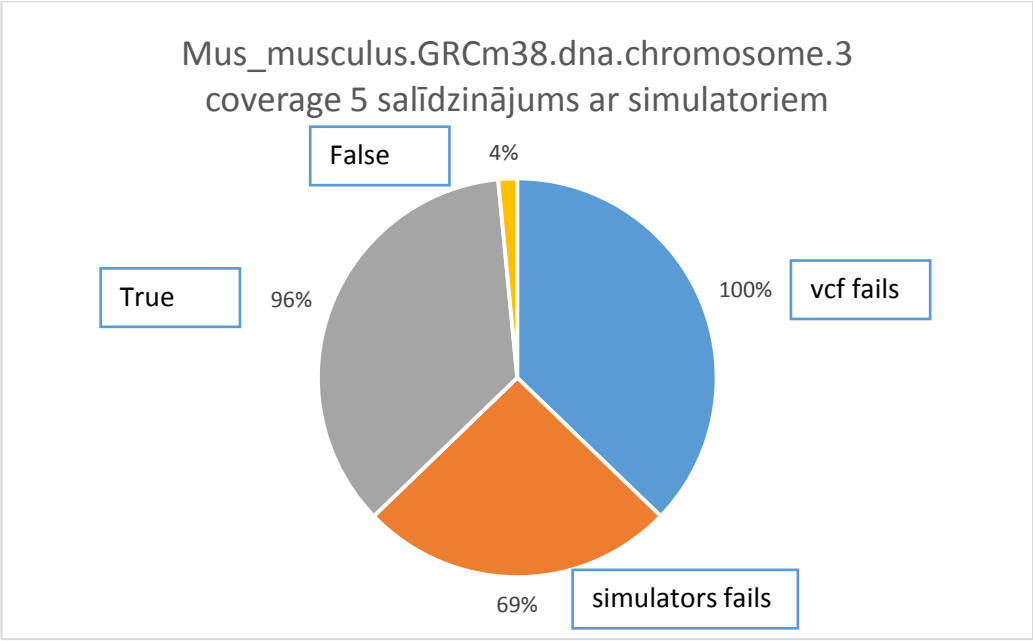
6.5. Mus_musculus.GRCm38.dna.chromosome.1 coverage 10 salīdzinājums ar simulatoriem



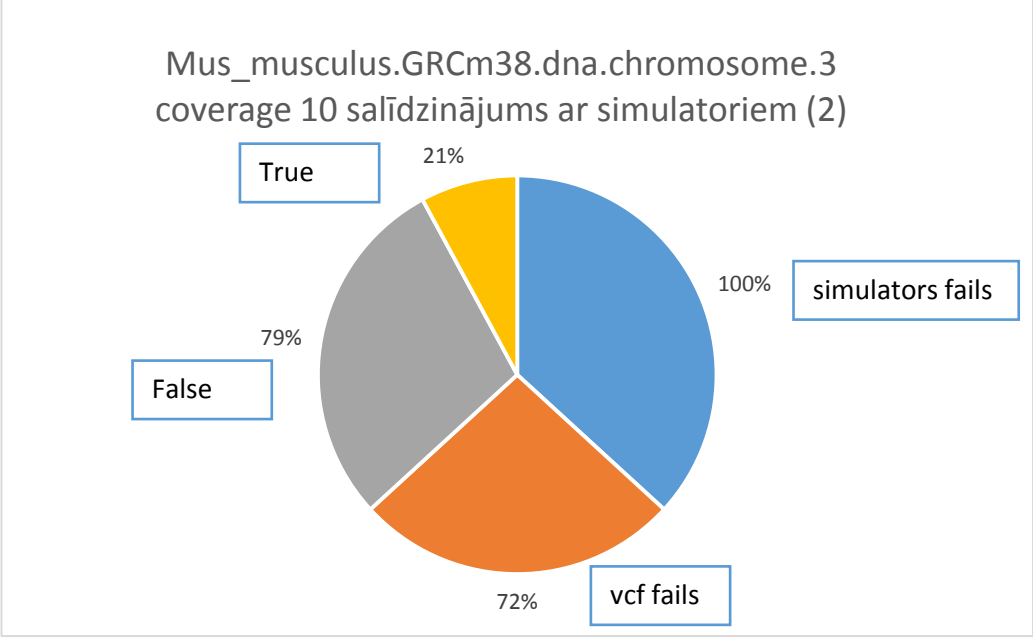


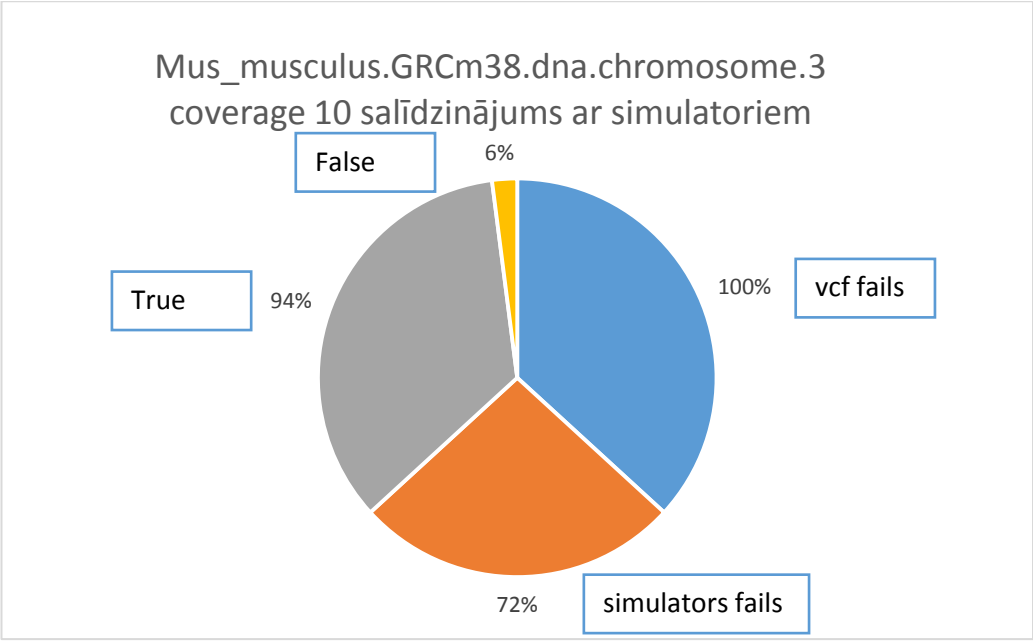
6.6. Mus_musculus.GRCm38.dna.chromosome.3 coverage 5 salīdzinājums ar simulatoriem





6.7. Mus_musculus.GRCm38.dna.chromosome.3 coverage 10 salīdzinājums ar simulatoriem




Mus_musculus.GRCm38.dna.chromosome.3
coverage 10 salīdzinājums ar simulatoriem
False
6%
True
94%
100%
vcf fails
72%
simulators fails

7. Secinājumi

Darbā apskatīti pamata statistikas modeļi un četru genotipu algoritmu darbība: GenoSNP, Illuminus, CRLMM, un GenCall.

Visas apskatītās metodes darbojas ar līdzīgiem pamatprincipiem, atšķiras tikai pielietojums. Katrā no programmām

Dažādos mērījumos, algoritmu rezultāti ir dažādi. Atkarībā no izmantoto datu apjoma atšķiras gan precizitātes mērījumi, gan rezultātu kvalitāte gan izmantoto datu drošība.

Izmantojot vienus un tos pašus gēnu paraugus programmu salīdzināšanai, rezultāti atšķiras vidēji sākot no dažiem procentiem līdz pat vairākiem desmitiem procentu.

Izmantotā literatūra un avoti

1. Vikipēdija. [Tiešsaiste] [Citēts: 2016. gada 05. 01.] https://lv.wikipedia.org/wiki/%C4%A2en%C4%93tika.

2. medicine.lv. *Latvijas medicīnas portāls.* [Tiešsaiste] 2012. gada 02. 04. [Citēts: 2016. gada 09. 01.] http://www.medicine.lv/raksti/dezoksiribonukleinskabe_pme.

3. medicine.lv. *Latvijas veselības portāls.* [Tiešsaiste] 2012. gada 02. 04. [Citēts: 2016. gada 09. 01.] http://www.medicine.lv/raksti/ribonukleinskabe_pme.

4. Liu, Cynthia Ruijie. Comparison of statistical models for genotype calling algorithms. *School of Mathematics and Statistics.* [Tiešsaiste] [Citēts: 2015. gada 23. 11.] http://www.ms.unimelb.edu.au/documents/thesis/(Cynthia)RuijieLiu.pdf.

5. McElroy, Kerensa. http://gensoft.pasteur.fr/docs/GemSIM/v1.6/Manual.pdf. *http://gensoft.pasteur.fr/.* [Tiešsaiste] 2012. gada 06. 07. [Citēts: 2016. gada 13. 12.] http://gensoft.pasteur.fr/docs/GemSIM/v1.6/Manual.pdf.

6. *Vikipēdija.* [Tiešsaiste] [Citēts: 2016. gada 09. 01.] https://lv.wikipedia.org/wiki/Olbaltumvielas.

7. Nature reviews immunology. [Tiešsaiste] [Citēts: 2016. gada 11. 01.] http://www.nature.com/nri/journal/v5/n1/fig_tab/nri1532_F2.html.

8. Ben Langmead, Cole Trapnell, Mihai Pop and Steven L Salzberg. http://genomebiology.com/2009/10/3/R25. [Tiešsaiste] 2009. gada 04. 03. [Citēts: 2017. gada 12. 01.] http://genomebiology.com/2009/10/3/R25.

9. Sequence Alignment/Map Format Specification. *https://samtools.github.io/.* [Tiešsaiste] 2016. gada 28. 04. [Citēts: 2017. gada 25. 04.] https://samtools.github.io/hts-specs/SAMv1.pdf. 494628a.

10. IGSR: The International Genome Sample Resource. *IGSR: The International Genome Sample Resource.* [Tiešsaiste] The International Genome Sample Resource (IGSR) has been established at EMBL-EBI to continue supporting data generated by the 1000 Genomes Project, , 2016. gada. [Citēts: 2017. gada 16. 03.] http://www.internationalgenome.org/wiki/Analysis/Variant%20Call%20Format/vcf-variant-call-format-version-40/.

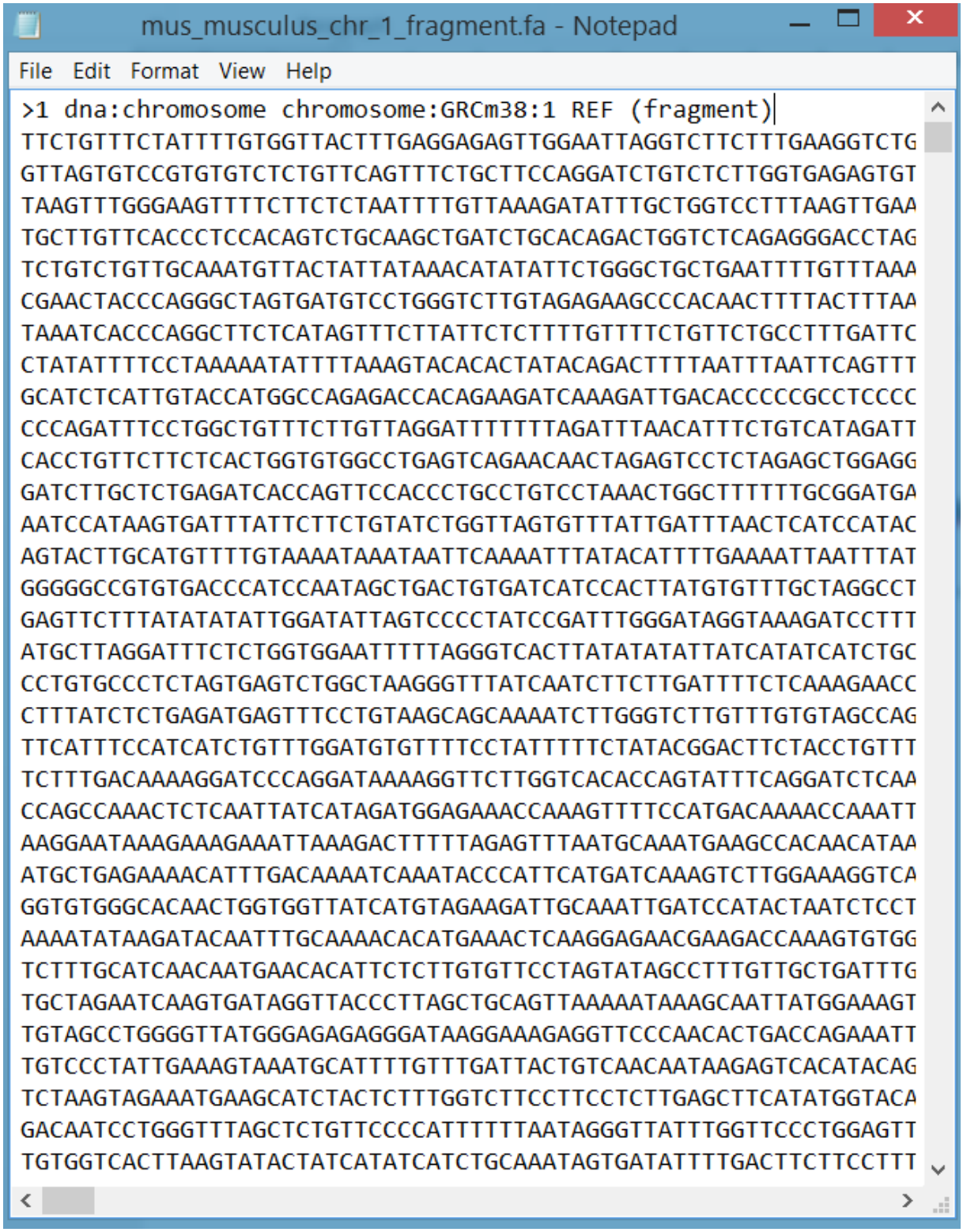
11. Bowtie 2. *sourceforge.net.* [Tiešsaiste] Johns Hopkins University, 2017. gada 05. 05. [Citēts: 2017. gada 10. 05.] http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml.

12. Genome Analysis Toolkit. *https://software.broadinstitute.org/gatk/.* [Tiešsaiste] Broad Institute, 2016. gada. [Citēts: 2017. gada 06. 03.] https://software.broadinstitute.org/gatk/documentation/topic?name=tutorials#1.

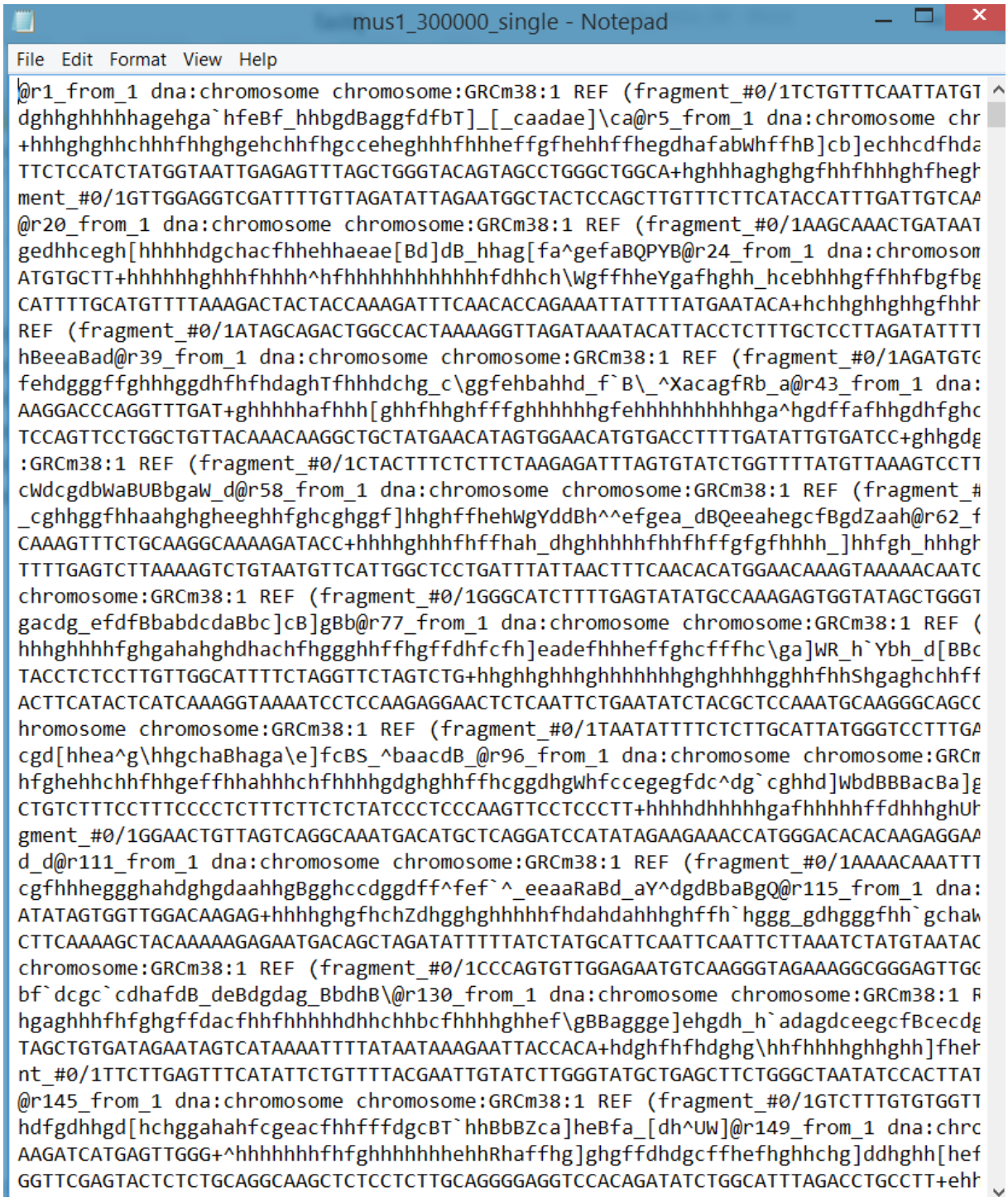
13. Xiaoqing Yu, Shuying Sun. BMC Bioinformatics. *https://bmcbioinformatics.biomedcentral.com.* [Tiešsaiste] 2013. gada. [Citēts: 2017. gada 15. 05.] https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-14-274.

PIELIKUMI

1. pielikums FASTA formāta datnes paraugs



2. pielikums FASTQ formāta datnes paraugs


mus1_300000_single - Notepad

File Edit Format View Help

```
@r1_from_1 dna:chromosome chromosome:GRCm38:1 REF (fragment_#0/1TCTGTTTCAATTATGT
dghhghhhhhagehga`hfeBf_hhbgdBaggfdfbT]_[_caadae]\ca@r5_from_1 dna:chromosome chr
+hhhghghhchhhfhhghgehchhfhgcceheghhhfhhheffgfhehhffhegdhafabWhffhB]cb]echhcdfhda
TTCTCCATCTATGGTAATTGAGAGTTTAGCTGGGTACAGTAGCCTGGGCTGGCA+hghhhaghghgfhhfhhhghfhegh
ment_#0/1GTTGGAGGTCGATTTTGTTAGATATTAGAATGGCTACTCCAGCTTGTTTCTTCATACCATTTGATTGTCAA
@r20_from_1 dna:chromosome chromosome:GRCm38:1 REF (fragment_#0/1AAGCAAACTGATAAT
gedhhcegh[hhhhhdgchacfhhehhaeae[Bd]dB_hhag[fa^gefaBQPYB@r24_from_1 dna:chromosom
ATGTGCTT+hhhhhhghhhfhhhh^hfhhhhhhhhhhhhhfdhhch\WgffhheYgafhghh_hcebhhhgffhhfbgfbg
CATTTTGCATGTTTTAAAGACTACTACCAAAGATTTCAACACCAGAAATTATTTTATGAATACA+hchhghhghhgfhhh
REF (fragment_#0/1ATAGCAGACTGGCCACTAAAAGGTTAGATAAATACATTACCTCTTTGCTCCTTAGATATTTT
hBeeaBad@r39_from_1 dna:chromosome chromosome:GRCm38:1 REF (fragment_#0/1AGATGTC
fehdgggffghhhggdhfhfhdaghTfhhhdchg_c\ggfehbahhd_f`B\_^XacagfRb_a@r43_from_1 dna:
AAGGACCCAGGTTTGAT+ghhhhhafhhh[ghhfhhghfffghhhhhhgfehhhhhhhhhhhga^hgdffafhhgdhfghc
TCCAGTTCCTGGCTGTTACAAACAAGGCTGCTATGAACATAGTGGAACATGTGACCTTTTGATATTGTGATCC+ghhgdg
:GRCm38:1 REF (fragment_#0/1CTACTTTCTCTTCTAAGAGATTTAGTGTATCTGGTTTTATGTTAAAGTCCTT
cWdcgdbWaBUBbgaW_d@r58_from_1 dna:chromosome chromosome:GRCm38:1 REF (fragment_#
_cghhggfhhaahghgheeghhfghcghggf]hhghffhehWgYddBh^^efgea_dBQeeahegcfBgdZaah@r62_f
CAAAGTTTCTGCAAGGCAAAAGATACC+hhhhghhhfhffhah_dhghhhhhfhhfhffgfgfhhhh_]hhfgh_hhhgh
TTTTGAGTCTTAAAAGTCTGTAATGTTCATTGGCTCCTGATTTATTAACTTTCAACACATGGAACAAAGTAAAAACAATC
chromosome:GRCm38:1 REF (fragment_#0/1GGGCATCTTTTGAGTATATGCCAAAGAGTGGTATAGCTGGGT
gacdg_efdfBbabdcdaBbc]cB]gBb@r77_from_1 dna:chromosome chromosome:GRCm38:1 REF (
hhhghhhhfghgahahghdhachfhggghhffhgffdhfcfh]eadefhhheffghcfffhc\ga]WR_h`Ybh_d[BBc
TACCTCTCCTTGTTGGCATTTTCTAGGTTCTAGTCTG+hhghhghhhghhhhhhhghghhhhgghhfhhShgaghchhff
ACTTCATACTCATCAAAGGTAAAATCCTCCAAGAGGAACTCTCAATTCTGAATATCTACGCTCCAAATGCAAGGGCAGCC
hromosome chromosome:GRCm38:1 REF (fragment_#0/1TAATATTTTCTCTTGCATTATGGGTCCTTTGA
cgd[hhea^g\hhgchaBhaga\e]fcBS_^baacdB_@r96_from_1 dna:chromosome chromosome:GRCm
hfghehhchhfhhgeffhhahhhchfhhhhgdghghhffhcggdhgWhfccegegfdc^dg`cghhd]WbdBBBacBa]g
CTGTCTTTCCTTTCCCCTCTTTCTTCTCTATCCCTCCCAAGTTCCTCCCTT+hhhhdhhhhhgafhhhhhffdhhhghUh
gment_#0/1GGAACTGTTAGTCAGGCAAATGACATGCTCAGGATCCATATAGAAGAAACCATGGGACACACAAGAGGAA
d_d@r111_from_1 dna:chromosome chromosome:GRCm38:1 REF (fragment_#0/1AAAACAAATTT
cgfhhheggghahdghgdaahhgBgghccdggdff^fef`^_eeaaRaBd_aY^dgdBbaBgQ@r115_from_1 dna:
ATATAGTGGTTGGACAAGAG+hhhhghgfhchZdhgghghhhhhfhdahdahhhghffh`hggg_gdhgggfhh`gchaW
CTTCAAAAGCTACAAAAAGAGAATGACAGCTAGATATTTTTATCTATGCATTCAATTCAATTCTTAAATCTATGTAATAC
chromosome:GRCm38:1 REF (fragment_#0/1CCCAGTGTTGGAGAATGTCAAGGGTAGAAAGGCGGGAGTTGG
bf`dcgc`cdhafdB_deBdgdag_BbdhB\@r130_from_1 dna:chromosome chromosome:GRCm38:1 F
hgaghhhfhfghgffdacfhhfhhhhhhdhhchhbcfhhhhghhef\gBBaggge]ehgdh_h`adagdceegcfBcecdg
TAGCTGTGATAGAATAGTCATAAAATTTTATAATAAAGAATTACCACA+hdghfhfhdghg\hhfhhhhghhghh]fheh
nt_#0/1TTCTTGAGTTTCATATTCTGTTTTACGAATTGTATCTTGGGTATGCTGAGCTTCTGGGCTAATATCCACTTAT
@r145_from_1 dna:chromosome chromosome:GRCm38:1 REF (fragment_#0/1GTCTTTGTGTGGTT
hdfgdhhgd[hchggahahfcgeacfhhfffdgcBT`hhBbBZca]heBfa_[dh^UW]@r149_from_1 dna:chrc
AAGATCATGAGTTGGG+^hhhhhhhfhfghhhhhhhhehhRhaffhg]ghgffdhdgcffhefhghhchg]ddhghh[hef
GGTTCGAGTACTCTCTGCAGGCAAGCTCTCCTCTTGCAGGGGAGGTCCACAGATATCTGGCATTTAGACCTGCCTT+ehh
```

Dokumentārās lapas forma

Maģistra darbs "METODES UN PROGRAMMATŪRA GENOMA DATU ANALĪZEI" izstrādāts LU Datorikas fakultātē.

Darba teksta galīgā versija izgatavota 22.05.2017

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: _____

(Autora paraksts un datums)

Ar savu parakstu apliecinu, ka esmu lasījis augstāk minēto maģistra darbu un atzīstu to par p i e m ē r o t u / n e p i e m ē r o t u (nevajadzīgo svītrot) aizstāvēšanai Latvijas Universitātes datorzinātņu maģistrantūrā.

Darba vadītājs: _____

(Vadītāja paraksts un datums)

Darbs iesniegts maģistratūras sekretariātā _____.

(Iesniegšanas datums)

Ar šo es apliecinu, ka darba elektroniskā versija ir augšupielādēta LU informatīvajā sistēmā.

Studiju metodiķe: _____.

(Metodiķes paraksts)

Recenzents: _____

(Dr.dat., Aleksandrs Rivošs)

Darbs aizstāvēts maģistra gala pārbaudījuma komisijas sēdē

_____ prot. Nr. _____

(Darba aizstāvēšanas datums)

Komisijas sekretārs: _____

(Sekretāra paraksts)