LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

VAIRĀKU SEKVENČU IZLĪDZINĀŠANAS METOŽU SALĪDZINĀJUMS

MAĢISTRA DARBS

Autors: **Valdis Gavars**
Stud. apl. Nr.: vg11059
Darba vadītājs: Prof. Juris Vīksna

Rīga 2021

ANOTĀCIJA

Šajā maģistra darbā paredzēts izpētīt, aprakstīt un salīdzināt dažādas praksē pieejamas vairāku sekvenču izlīdzināšanas metodes. Darbā tiek aprakstīti vairāku sekvenču izlīdzināšanas metožu galvenie pielietojumi bioinformātikā, biežāk sastopamie algoritmi, kuri tiek izmantoti darbā tālāk apskatītajās programmā.

Īsi aprakstītas atvērtā koda programmas, kuras industrijā tiek izmantotas visbiežāk. Maģistra darba ietvaros veikts praktisks pētījums par šo metožu priekšrocībām un trūkumiem. Salīdzinājums veikts gan uz reāliem datu masīviem, gan simulētiem, lai spētu pēc iespējas daudzpusīgāk salīdzināt pieejamo programmatūru. Veikta iegūto rezultātu grafiska atspoguļošana un analīze par novērojamajām tendencēm un programmu īpašībām, kā arī veikts salīdzinājums ar citu pētnieku iepriekš publicētajiem rezultātiem.

ATSLĒGAS VĀRDI: VAIRĀKU SEKVENČU IZLĪDZINĀŠANA, BIOINFORMĀTIKA, SEKVENCES

ANNOTATION

TITLE: COMPARISON OF MULTIPLE SEQUENCE ALIGNMENT METHODS

This master's thesis is intended to study, describe, and compare various methods of Multiple Sequence Alignment available. The paper describes the main applications of several Multiple Sequence alignment software in bioinformatics, the most common algorithms used in the following software.

Shortly described the open-source software most used in the industry. Within the framework of the master's thesis, a practical study of the advantages and disadvantages of these methods has been carried out. The comparison is made on both real data and simulated data to be able to compare the available software as much as possible. Shown graphical presentation of the obtained results and analysis of the observed trends and characteristics of the programs and comparison made with other researcher results published before.

KEYWORDS: MULTIPLE SEQUENCE ALIGNMENT, BIOINFORMATICS, SEQUENCES

AUTOREFERĀTS

Šī darba ietvaros tika salīdzinātas bioinformātikas industrijā bieži izmantotas un brīvi pieejamas vairāku sekvenču izlīdzināšanas programmas. Veicot šo teorētisko un praktisko pētījumu, iegūti dati par jaunākajām programmu versijām un to veiktspēju pie dažādiem datiem. Iepriekš publikācijās veiktie līdzīga veida pētījumi ir novecojuši, jo programmas regulāri tiek atjaunotas (Kalign jaunākā versija izdota 2019. gadā), kā arī šāda veida salīdzinājumus visbiežāk veic šo pašu programmu autori, kas var šo pētījumu padarīt subjektīvu.

Darbā lasītājs tiek iepazīstināts ar vairāku sekvenču izlīdzināšanu, tās nozīmi bioinformātikā, kā arī pastāstīts katras apskatītās programmas darbības princips un veikts svarīgāko algoritmu apskats. Autora prāt izklāsts ir veikts pietiekoši detalizēti, lai lasītājs varētu bez iepriekšējām zināšanām par šo jomu orientēties darba nozīmīgumā un lietderīgumā.

Autors šī darba ietvaros ir iepazinies ar vairāku sekvenču izlīdzināšanas programmām, to darbības principiem, lai tās varētu lietot un izmantot paredzētajiem nolūkiem, kā arī praktiski izstrādāts skripts, kas nodrošina pēc iespējas automatizētu programmu testēšanu un rezultātu dokumentāciju pētījuma veikšanai. Kā arī atlasīti nepieciešamie dati priekš programmu salīdzināšanas, ņemot vērā uzdevuma specifiku.

Salīdzinot darbā iegūtos rezultātus ar citās publikācijās atrodamiem ir redzams, ka darbā iegūtie rezultāti pamatā sakrīt ar citu veiktajiem novērojumiem, taču iegūti arī secinājumi, kuri citu veiktajos pētījumos nav novēroti, kurus būtu iespējams pētīt tālāk.

Darba teksta kvalitāte ir pārbaudīta manuāli, kā arī ar Tildes latviešu valodas rīku, lai darba teksts būtu pēc iespējas lasāmāks. Pareizrakstības kļūdām ir iziets cauri ar visiem pieejamajiem rīkiem. Darbā lietotie termini ir latviskoti pēc oficiālās terminoloģijas vai arī atveidoti pēc iespējas tuvāk pieejamajai informācijai. Autors ir izgājis cauri "Darba noformējuma kontrolsarakstam" un pārliecinājies par darba atbilstību tam.

Darbā izmantotās citu autoru idejas un teksti ir atzīmētas ar attiecīgajām atsaucēm, katram attēlam arī ir pievienots tā avots.

SATURS

APZĪMĒJUMU SARAKSTS 7

IEVADS 8

1. SEKVENCES UN TO IZLĪDZINĀŠANA 10

1.1. Sekvences 10

1.2. Sekvenču izlīdzināšana 11

1.3. MSA pielietojumi 12

1.3.1. Filoģenētika 12

1.3.2. RNS un olbaltumvielu struktūras prognozēšana 13

1.3.3. Praimeru izstrāde 14

1.3.4. Salīdzinošā genomika 15

2. ALGORITMI 17

2.1. K-kortežu attālums 17

2.2. Nīdlmena-Venša algoritms (Smita-Vatermana uzlabojums) 18

2.3. Substitūciju matricas 19

2.4. Kaimiņu-apvienošanas algoritms 21

2.5. UPGMA 22

2.6. Slēptie Markova Modeļi 23

3. PROGRAMMU APSKATS 26

3.1. Clustal Omega 26

3.2. Kalign 27

3.3. MAFFT 27

3.4. MUSCLE 28

4. METOŽU SALĪDZINĀJUMA PARAMETRI 30

4.1. BAliBASE 30

4.2. Pāru summas novērtējums 31
4.3. Kopējais kolonnu novērtējums 31
4.4. Skaitļošanas ilgums 32
5. PROGRAMMU IZVĒRTĒJUMS 33
5.1. BAliBASE references kopas 33
5.1.1. Pirmā kopa 33
5.1.2. Otrā kopa 35
5.1.3. Trešā, ceturtā un piektā kopa 37
5.1.4. Septītā un astotā kopa 39
5.1.5. Devītā kopa 41
5.1.6. Desmitā kopa 42
5.1.7. Kopvērtējums 44
5.2. Reāli dati 51
REZULTĀTI 56
SECINĀJUMI 58
IZMANTOTĀ LITERATŪRA UN AVOTI 59
PIELIKUMI 63

APZĪMĒJUMU SARAKSTS

MSA – Vairāku sekvenču izlīdzinājums *(Multiple Sequence Alignment)*

Insercija – Iespraudums bioloģiskajā sekvencē

Delēcija – Dzēšana bioloģiskajā sekvencē

PCR – Polimerāzes ķēdes reakcija *(Polymerase chain reaction)*

DNS – dezoksiribonukleīnskābe

RNS - ribonukleīnskābe

HMM – Slēptie Markova Modeļi *(Hidden Markov Models)*

IEVADS

Šī darba tēma ir aktuāla, jo mūsdienās sekvencēšanas datu apjoms palielinās ļoti strauji, ja pirmā pilnā cilvēka genoma sekvencēšanas izmaksas tika mērītas miljardos, tad šobrīd jau tiek lēsts, ka to var izdarīt lētāk nekā 1000 EUR. Līdz ar to datu apjoms ar ģenētiskajiem datiem, sekvencēm pieaug ar katru minūti, kā arī to iespējamais pielietojums dažādās nozarēs mainās, tiek atklāti, izdomāti jauni un inovatīvi veidi, kā ar ģenētiskajiem datiem padarīt cilvēku dzīves kvalitatīvākas. Vairāku sekvenču izlīdzināšanas tēma ir aktuāla arī autoram personīgi – vairāku sekvenču izlīdzināšana tika izmantota, lai palīdzētu izstrādāt praimerus vīrusa testu diagnostikai, palīdzot atrast stabilākās vīrusa RNS sekvenču daļas.

Ņemot vērā, ka vairāku sekvenču izlīdzināšana (turpmāk tekstā minēts arī kā MSA* – **Multiple Sequence Alignment**) ir nozīmīga ģenētisko datu apstrādāšanas daļa, kuru izmanto dažādiem ļoti svarīgiem mērķiem, tad ir noderīgi un aktuāli pārbaudīt dažādu pieejamo vairāku sekvenču līdzināšanas pamatdarbības principus, salīdzināt to darbības spējas, iespējamo datu ievadi un izvadi.

Darba mērķis ir iegūt informāciju par brīvi pieejamu programmatūru veiktspēju, priekšrocībām un trūkumiem, strādājot ar dažāda tipa un daudzuma sekvencēm. Uzdevums ir izpētīt literatūru par to, kā darbojas šīs programmas, kādus pamatprincipus tās izmanto. savu individuālu praktisko salīdzinājumu MSA programmatūru starpā, izmantojot gan šim nolūkam speciāli modelētus datus ar dažādu bioloģisko saturu, kā arī reālas datu kopas.

Darba uzdevumi ir iepazīties un aprakstīt nozares populārāko programmu darbības principus, lai lasītājs spētu izprast programmu atšķirības nianses. Tālāk - izstrādāt plānu programmu salīdzināšanai, izvēlēties parametrus un datu kopas, ar kurām programmas tiek salīdzinātas, paskaidrot to izvēli un parametru nozīmi, veikt praktisku rezultātu ieguvi, apkopot un atrādīt iegūtos rezultātus uzskatāmā veidā, kā arī veikt šo rezultātu analīzi.

Nodaļā SEKVENCES UN TO IZLĪDZINĀŠANA tiek paskaidrots, kas ir sekvences, sekvenču izlīdzināšana, to vērtība bioinformātikā, un tiek apskatīti populārākie, aktuālākie MSA pielietojumi šobrīd. Kā arī tas, kādu tieši lomu spēlē MSA šajos uzdevumos un lasītājs tiek iepazīstināts detalizētāk ar šīs tēmas aktualitāti. Iepazīstoties ar šo nodaļu tiek iegūts priekšstats par to, kāpēc MSA ir tik svarīgs.

Tālāk sadaļā ALGORITMI tiek apskatīti svarīgākie algoritmiskie principi, kurus izmanto programmatūras, lai veiktu vairāku sekvenču izlīdzināšanu. Lasītājs tiek iepazīstināts ar pavisam īsiem pašu svarīgāko algoritmu aprakstiem un shēmām. Kā arī par to kādi ir iespējamie kritēriji, lai salīdzinātu programmatūru darbības kvalitāti, kā arī novērtēt tās ātrdarbību, kuri tiks, izmantoti vēlāk, veicot praktisko salīdzinājumu.

PROGRAMMU APSKATS tiek veikts apkopojums šobrīd populārākajām vairāku sekvenču izlīdzināšanas programmatūrām, kuras ir atvērtā koda programmas ar brīvpieejas licencēm. Tiek apskatīts, kādus algoritmus izmanto šīs programmas, kā arī literatūrā atrodamie pētījumi, izmantojot šīs programmas.

Nodaļā METOŽU SALĪDZINĀJUMA PARAMETRI tiek raksturoti parametri, pēc kuriem, veicot maģistra darba praktisko daļu, apskatītās metodes tiks praktiski salīdzinātas savā starpā, izmantojot gan reālus, gan simulētus datus.

PROGRAMMU IZVĒRTĒJUMS ir nodaļa, kurā parādīts veiktais praktiskais darbs, iegūtie detalizētie darba rezultāti, veicot vairāku sekvenču izlīdzināšanas programmu salīdzinājumu. Atspoguļoti iegūtie rezultāti pārskatāmā veidā, apkopotas svarīgākās atziņas, kuras iegūtas, veicot salīdzinājumu. Iegūtie rezultāti noformēti grafiku veidā ar autora komentāriem par rezultātu interpretāciju.

REZULTĀTI – nodaļa, kurā autors atskatās uz paveikto un veic apkopojumu par izpildītajiem vai neizpildītajiem uzdevumiem.

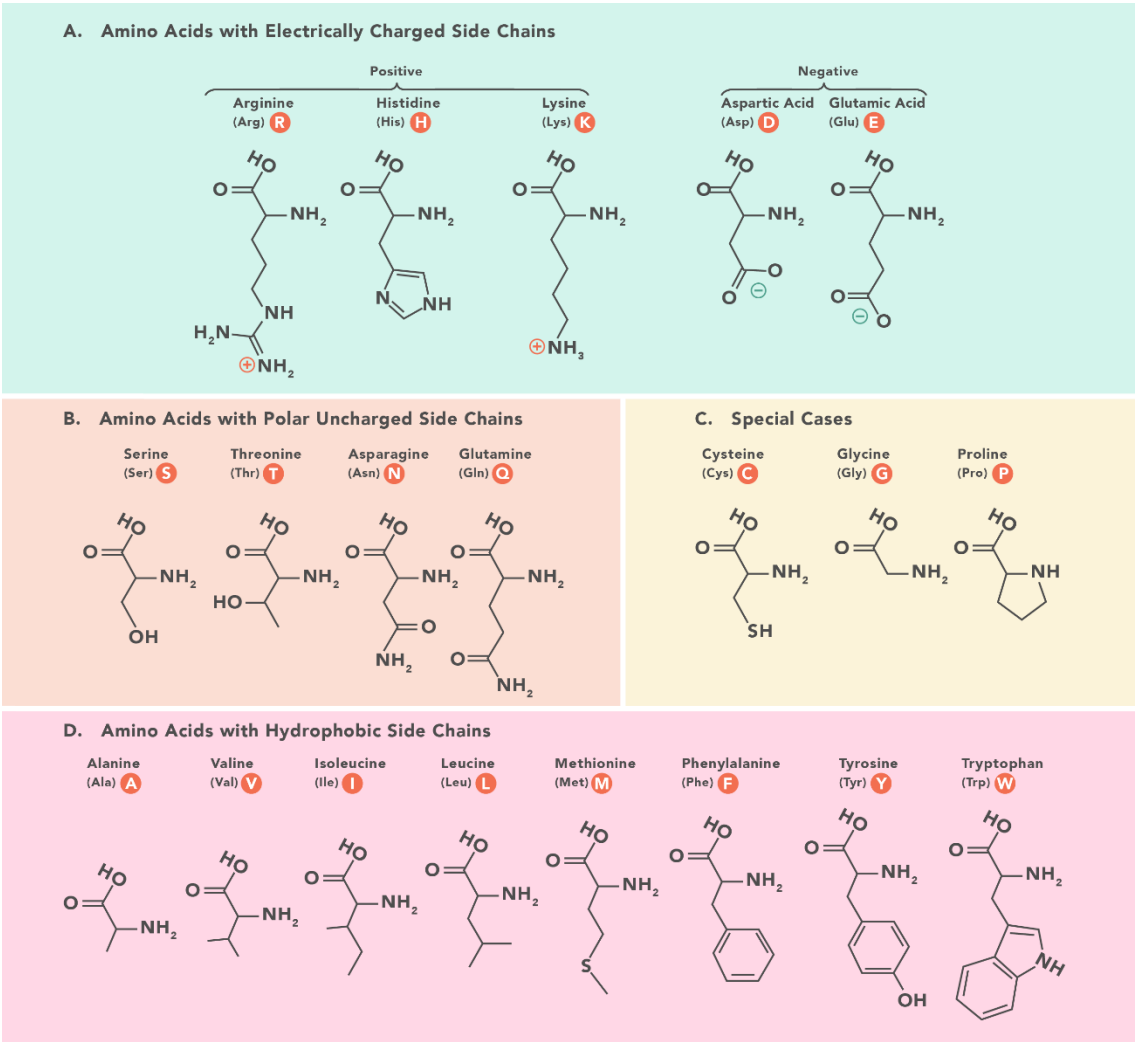
Nodaļā SECINĀJUMI autors veic īsu atskatu un pašvērtējumu veiktajam darbam un iezīmē nākotnes pētniecības plānu.

1. SEKVENCES UN TO IZLĪDZINĀŠANA

Šajā nodaļā sniegts ieskats par to, kas ir sekvences, to izlīdzināšana, lai sniegtu lasītājam zināšanas, kas nepieciešamas turpmākā darba lasīšanā

1.1. Sekvences

Sekvence šajā darbā ir domāta kā bioloģiskā sekvence. Bioloģiskās sekvences ir divu veidu – olbaltumvielu un nukleotīdu. Nukleotīdu sekvences pārsvarā sastāv no 4 simboliem {A, G, C, T}, kuri apzīmē četras bāzes – adenīnu, guanīnu, citozīnu un timīnu. Olbaltumvielu sekvences sastāv no 20 aminoskābēm – {A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V, O, U, B, Z, X, J}. Redzamas attēlotas arī att. 1.1.



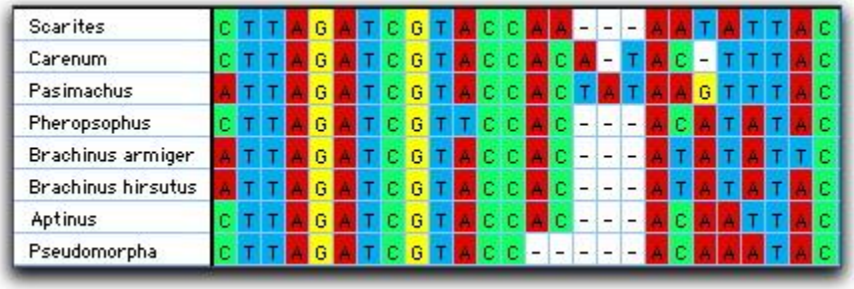
att. 1.1 Aminoskābes [1]

“Bioloģisko sekvenču analīze salīdzina, izlīdzina, indeksē un analizē bioloģiskās sekvences un tādejādi spēlē kritisku lomu bioinformātikā un modernajā bioloģijā” [2]

1.2. Sekvenču izlīdzināšana

Sekvenču izlīdzināšana balstās uz faktu, ka visi organismi ir radnieciski evolūcijas gaitā. Tas nozīmē, ka, jo līdzīgāki organismi, jo līdzīgākām būtu jābūt to sekvencēm (gan nukleotīdu, gan olbaltumvielu). Sekvenču izlīdzinājums ir sekvenču kārtošanas process, lai sasniegtu pēc iespējas lielāku līdzību, kas tad ataino arī šo sekvenču savstarpējo līdzības pakāpi. [2]

Izlīdzinājums atspoguļo sekvenču kopu, izmantojot viena burta kodu katrai aminoskābei (priekš olbaltumvielu sekvencēm) vai katram nukleotīdam (priekš DNS/RNS sekvencēm). Katra rinda līdzinājumā parasti atspoguļo vienu sekvenci, un strukturāli, funkcionāli vai evolucionāri līdzvērtīgi posmi ir līdzināti vertikāli. Ja sekvenču garumi atšķiras, tad tiek pievienoti sekvences plaisas apzīmējoši simboli, kas apzīmē insercijas* (iespraudums) vai delēcijas* (dzēšanas) notikumu. [3] Zemāk redzamajā att. 1.2 parādīts, kā izskatās sekvences līdzinājums, kurā dažādos grafiskos veidos tiek atspoguļotas – insercijas, delēcijas un mutācijas pēc sekvenču izlīdzinājuma datiem.



att. 1.2 Sekvences izlīdzinājuma piemērs [4]

Runājot par sekvenču izlīdzināšanu var runāt par divu veidu izlīdzināšanu – globālo un lokālo. Lokālā sekvenču izlīdzināšana dažādās sekvencēs mēģina atrast lokālus reģionus ar augstāko līdzību starp sekvencēm – attiecīgi, tā neizmanto pilnīgi visu sekvenci, bet gan apakškopu no tās. Globālā sekvenču izlīdzināšana izlīdzina sekvences visā garumā, izmantojot visu sekvences garumu, lai savietotu to pēc iespējas precīzāk ar citām sekvencēm. Šajā darbā tiek apskatīts globālais sekvenču izlīdzinājums.

Vēl sekvenču izlīdzināšanu var iedalīt kategorijās atkarībā no sekvenču skaita, kas jāizlīdzina – pāriska sekvenču izlīdzināšana un vairāku sekvenču izlīdzināšana. Pārisku sekvenču

izlīdzināšana salīdzina divas sekvences savā starpā un parasti tiek izmantota datubāžu meklēšanas programmās, lai atrastu radnieciskas sekvences jaunai sekvencei. Ja ir vairāk par divām sekvencēm, tad tas jau ir vairāku sekvenču izlīdzinājums, par kura pielietojumiem vairāk tiek pastāstīts nākamajā nodaļā. Šajā darbā tiek apskatītas tehniski sarežģītākā un skaitļošanas jaudas pieprasošākā - MSA.

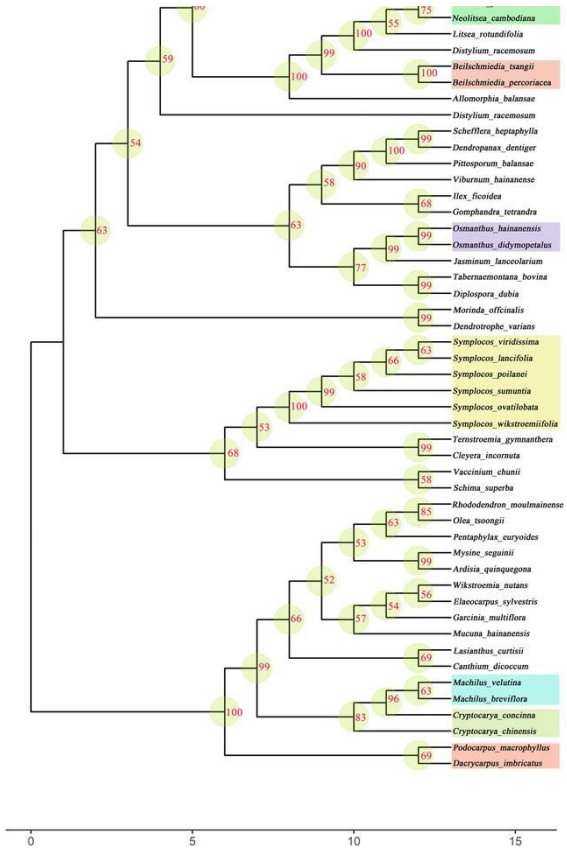
Ja sekvenču izlīdzināšanu veic tiešā veidā, tad tās algoritma sarežģītība, jeb O notācija ir $O(L^N)$ priekš N skaita sekvencēm L garumā, kas padara MSA sarežģītu pat nelielam daudzumam sekvenču, līdz ar to ir nepieciešams izmantot heiristiskas metodes. [5] Modernajā molekulārajā bioloģijā MSA spēlē fundamentālu lomu un tiek izmantota dažādos svarīgos veidos, piemēram, filoģenētika, RNS un olbaltumvielu struktūras prognozēšana, praimeru izstrāde, genoma savietošana un anotācija, salīdzinošā genomika.

1.3. MSA pielietojumi

Sekvenču izlīdzināšanai ir liela nozīme bioinformātikas nozarē, jo tā palīdz analizēt sekvenču savstarpējo radniecību, evolūciju un palīdz sniegt ieskatu dažādās bioloģijas nozarēs. Tālāk par populārākajiem sekvenču izlīdzināšanas pielietojumiem.

1.3.1. Filoģenētika

"Filoģenētika ir ģenētikas nozare, kas pēta visas ģenētiskā materiāla un tā elementu pārvērtības, sākot ar ģenētiskā materiāla pārkombinēšanos dzimumvairošanās procesā, mutācijām un beidzot ar ģenētiskā sastāva pārmaiņām populācijās." [6] Sugu koks tipiski tiek noteikts pēc gēnu kolekcijas (vai citiem genomikas reģioniem) veicot sekvenču līdzināšanu un tad no šiem līdzinājumiem tiek veidots filoģenētikas koks. Kādā 2011. gada zinātniskajā publikācijā [7] minēta problēma, ka, pieaugot sekvenču daudzumam, krietni sarūk izmantojamo programmatūru skaits.

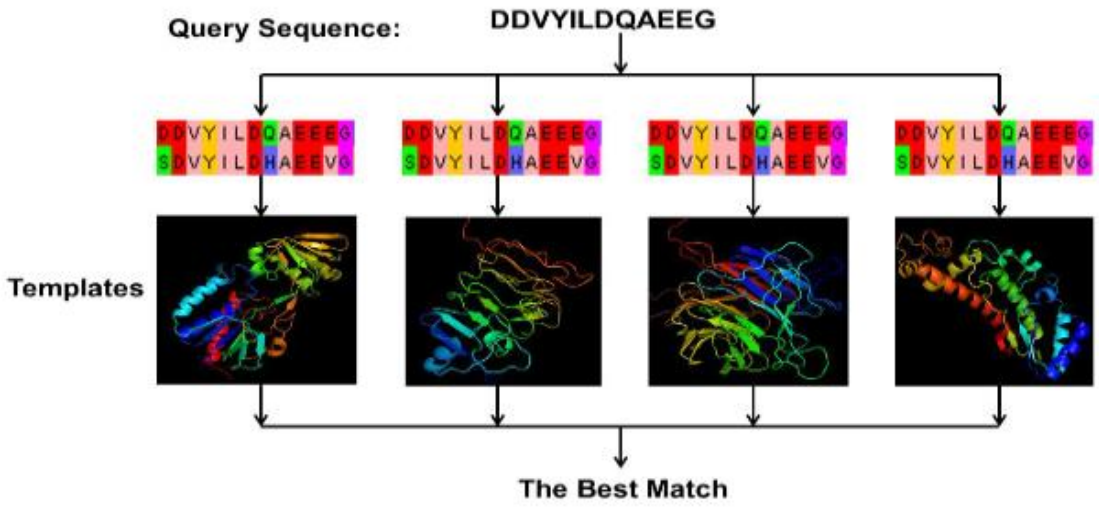


att. 1.3 Filoģenētikais koks. [8]

Redzams arī att. 1.3 kā izskatās filoģenētiskā koka piemērs, kurš tiek radīts no MSA rezultātiem.

1.3.2. RNS un olbaltumvielu struktūras prognozēšana

Olbaltumvielu struktūras prognozēšanā MSA rezultāts kā ievade priekš struktūras prognozētāja ir vissvarīgākais solis, kas palīdz uzlabot prognozes precizitāti. Kā minēts 2017. gada pētījumā, kurā tiek salīdzinātas MSA metodes pēc to rādītājiem olbaltumvielu struktūras prognozēšanas vajadzībām: "Pēdējā laika atklājumi genoma sekvencēšanā pieprasa arvien vairāk un vairāk vajadzību pēc uzticamām un ātrām MSA metodēm, kas attiecas gan uz sekvenču skaitu, gan katras sekvences garumu." [9] Kā parādīts att. 1.4 , kā MSA strādā olbaltumvielu struktūras noteikšanai.



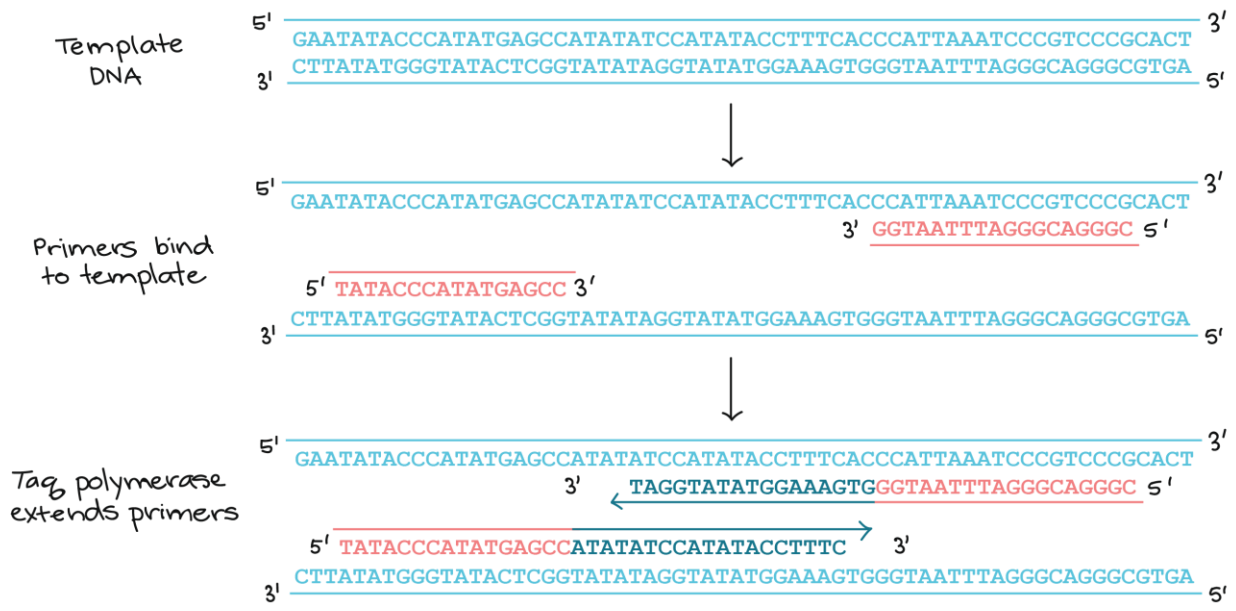
att. 1.4 Olbaltumvielu struktūras prognozēšana [10]

Pavisam procesā ir 3 soļi, 1) vaicājuma olbaltumvielu (vai RNS) sekvences līdzināšana pret šablonu datubāzi, 2) Viena vai vairāku šablonu izvēle, kas balstīta uz evolucionāru un strukturālu īpašību kalkulāciju no attiecīgajiem līdzinājumiem un 3) Būvēt 3D struktūru priekš mērķa olbaltumvielas (vai RNS) ņemot vērā ierobežojumus, ko dod līdzinātie reģioni, tajā pašā laikā minimizējot daļiņu enerģiju nelīdzināto cilpu reģionos un pievienot sānu ķēdes atomus. [10]

1.3.3. Praimeru izstrāde

Polimerāzes ķēdes reakcija ir process, kurā tiek pavairots paraugā esošais RNS vai DNS. Praimeris ir DNS vai RNS sekvences daļa, kuras īpašības ļauj pievienoties īpaši noteiktām sekvences daļām paraugā un cikliski kāpinot un pazeminot temperatūru šis RNS vai DNS tiek pavairots.

Polimerāzes ķēdes reakcijas *(Polymerase chain reaction – PCR)* praimeri ir ķīmiski sintezētas nukleotīdu praimeru molekulas, kuras ir svarīgas PCR amplifikācijas komponentes. Praimeri ir galvenie noteicēji PCR specifikā. Lai amplificētu specifiska DNS sekvence jāzina praimera norūdīšanas vietas sekvence mērķa DNS. Lai nodrošinātos pret iespēju, ka praimers pievienojas nespecifiskai vietai vai arī līdzīgai sekvencei citur mērķa DNS, tad praimerus parasti taisa 18-22 nukleotīdu garumā. [11] Vizuāls attēlojums - att. 1.5



att. 1.5 Praimeru darbības princips [12]

Mūsdienās visiem zināmais SARS-CoV-2 vīrusa testēšana arī balstās uz PCR metodi, un tā kā vīrusiem ir strauji mutējošs RNS, tad MSA tiek izmantota uz lieliem datu apjomiem, lai atrastu tos sekvenču rajonus, kuri ir stabili un gandrīz nemutē.

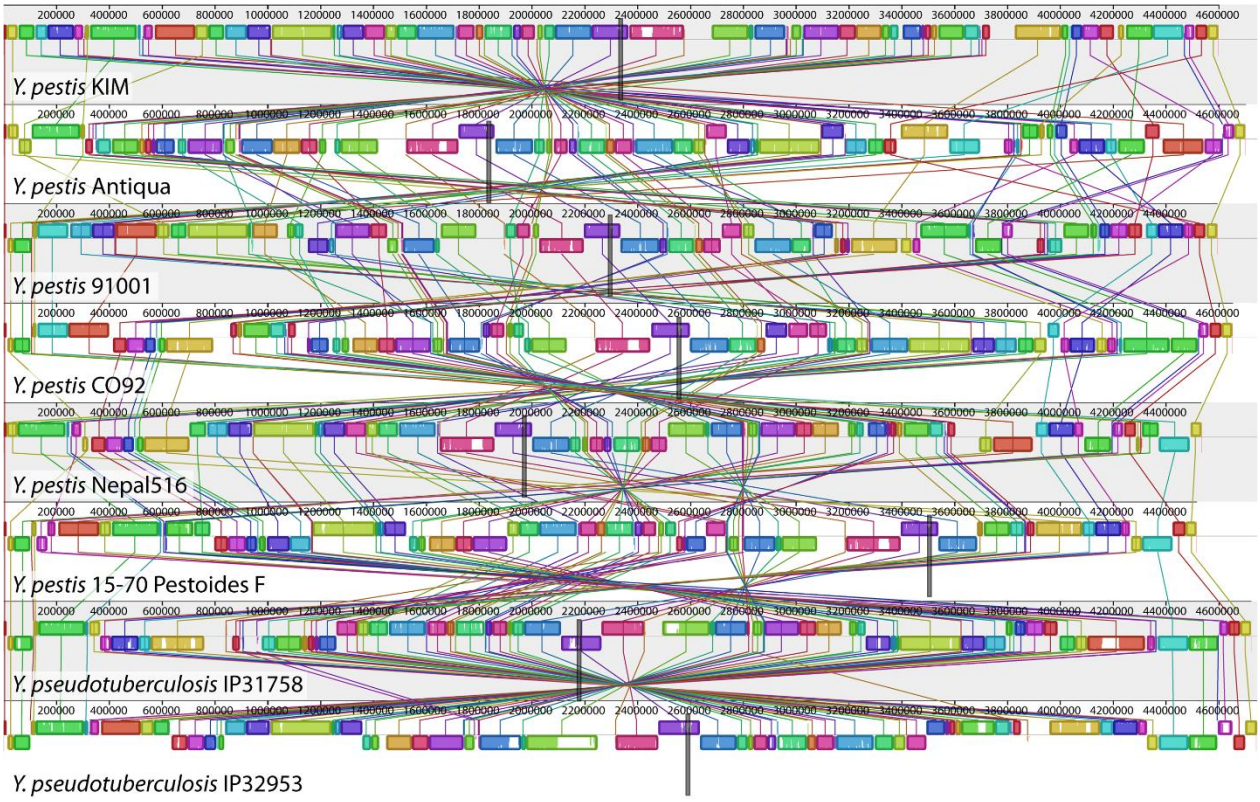
1.3.4. Salīdzinošā genomika

Sekvencēšanas tehnoloģijas attīstās ļoti strauji, kā arī visa genoma sekvencēšanas izmaksas krīt vēl straujāk. 2001. gadā Human Genome projekta ietvaros tika nosekvencēts pirmais pilnais genoms, kas izmaksāja 2.7 miljardus ASV dolāru. Šobrīd daži komerciāli uzņēmumi, kas nodarbojas ar sekvencēšanu, izsakās, ka spēj to izdarīt zem 1000 ASV dolāriem. Tātad var cerēt, ka tuvā nākotnē genoma sekvencēšana būs ierasta prakse klīniskajā medicīnā, kas padara personīgās genomikas un salīdzinošās genomikas pētījumus vēl svarīgākus.

Personīgā genomika ietver sevī sekvencēšanu, analīzi un indivīda genoma interpretāciju. Šī nozares spēj sniegt dažādus klīniskus pielietojumus, it īpaši ģenētisku problēmu un dažādu slimību diagnosticēšanā.

Salīdzinošā genomika ir cits pētniecības lauks, kas pēta genomiskās īpašības dažādos organismos. Tā mērķis ir saprast struktūru un genoma funkciju, identificējot reģionus ar līdzīgām sekvencēm starp aprakstītiem organismiem.

Gan personīgai, gan salīdzinošai genomikai nepieciešamā sekvenču izlīdzināšana, lai atklātu sekvenču konservāciju un variāciju. Sekvenču konservācijas struktūra var būt noderīga, lai paredzētu funkcionālas kategorijas, bet variācija – lai pierādītu attiecības starp organismiem vai populācijām dažādās vietās. Pētījumi parāda, ka variācija ir ļoti svarīga cilvēka veselībai un biežām ģenētiskām slimībām. Izlīdzināšanas ātrums ir svarīgs jautājums, jo genoma sekvence parasti sastāv no vairākiem miljoniem nukleotīdu. Viens no svarīgiem genoma salīdzināšanas pielietojumiem ir sekvences variācijas identifikācija starp genomiem, kuru var atrast lineāri skenējot tā izlīdzinājuma rezultātu. [13]



att. 1.6 Salīdzinošā Genomika [14]

2. ALGORITMI

Šajā nodaļā tiks apskatīti svarīgākie algoritmi, kuri tiek pārsvarā izmantotas nodaļas PROGRAMMU APSKATS aptvertajās programmās, lai saprastu pamatprincipus, kas ir kopīgi vai arī atšķirīgi šīm programmām, un tādejādi pēc tam arī uzsvērt šo programmu atšķirīgās daļas, paverot plašākas iespējas programmatūras darbības salīdzināšanai.

2.1. K-kortežu attālums

K-kortežu attāluma metodes algoritms priekš filoģenētisko koku veidošanas:

1) Aprēķināt attālumu starp katru sekvenču pāri izmantojot k-kortežu distanci
2) Izveidot pārisku k-kortežu attāluma matricu D priekš sekvenču kopas
3) Veidot filoģenētisko koku balstoties uz D un uz attāluma balstītu koku veidošanas metodi

K-kortežu attālums, kas ir starpība starp visu iespējamo *k* garuma kortežu frekvenci, var izvairīties no lielas skaitļošanas sarežģītības un tiek izmantots filoģenētisko koku rekonstrukcijā.

Sekvence *S* ar garumu *l* tiek definēta kā lineāra secība no *l* simboliem no ierobežota alfabēta *A* ar garumu *n*. *A* segments no *k* simboliem, kur k <= *l*, tiek apzīmēts kā k-kortežs, jeb k-vārds. Sanāk, ka kopā ir n^k iespējami k-korteži alfabētam *A*. K-korteža *w* parādīšanās skaits – N_w, tiek skaitīts bīdot slīdošo logu ar garumu *k* pār sekvenci ar soli 1 bāzu pāris (bp). Frekvence f_w no korteža *w* tiek iegūta no izdalot N_w ar kopējo kortežu skaitu. Respektīvi, f_w tiek definēts kā:

$$f_w = \frac{N_w}{l - k + 1} \qquad (2.1.)$$

K-kortežu distance tiek aprēķināta kā frekvenču starpība no visiem iespējamajiem k-kortežiem. Priekš DNS sekvencēm *A = {A, C, G, T}*. Katra DNS sekvence var tikt reprezentēta kā vektors, kas satur 4^k skaitļus, katrs no kura atspoguļo frekvenci attiecīgajam kortežam sekvencē. Priekš katrām divām sekvencēm *X* un *Y*, k-korteža attālums tiek aprēķināts, izmantojot šo formulu:

$$d(X,Y) = \sum_{i=1}^{4^k} \left| f_{w_i}^X - f_{w_i}^Y \right|^2 \qquad (2.2.)$$

, kur $f_{w_i}^X$ un $f_{w_i}^Y$ atbilst priekš i-tā k-korteža sekvencēs X un Y.

K-korteža attālums ņem starpību summu frekvencēs, lai izmērītu attālumu starp divām DNS sekvencēm. Kaut arī šis attālums nav jutīgs uz niecīgām starpībām starp gandrīz identiskām DNS sekvencēm, līdz ar to tas nav efektīvs, analizējot tuvu radnieciskas sugas. Kā arī, k-korteža attālums neņem vērā sekvenču struktūru, kura var saturēt svarīgu informāciju. Tātad k-korteža attālums var zaudēt informāciju, ko nes sekvenču struktūra. [15]

2.2. Nīdlmena-Venša algoritms (Smita-Vatermana uzlabojums)

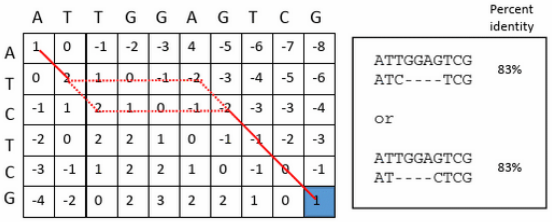
Nīdlmena-Venša *(Needleman Wunsch)* algoritms tādām pašām divām sekvencēm *X* un *Y*, kuras tika pieminētas, aprakstot k-korteža attālumu.

$$H_{i,j} = MAX\begin{cases} H_{i-1,j-1} + S_{i,j} \\ MAX\left(H_{i-k,j} - g - hk\right) \\ MAX\left(H_{i,j-1} - g - hl\right) \end{cases} \quad (2.3.)$$

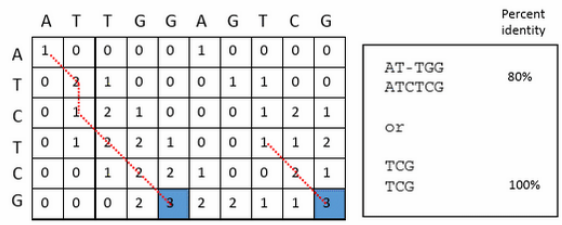
, kur $S_{i,j}$ ir novērtējums priekš x_i un y_i līdzināšanas, $H_{i,j}$ ir novērtējums optimālajam apakšsekvenču $x_1, \ldots, x_i$ un $y_1, \ldots, y_i$ izlīdzinājumam, g ir sods par to, ja tiek atvērta starpa; h – sods par starpas palielināšanu par vienu simbolu; k, l ir starpu garumi sekvencēs X un Y.

Optimālais lokālais izlīdzinājums starp divām sekvencēm, starp kurām visaugstāk novērtētie apakšsegmenti tiek izlīdzināti, iekļauj nelielu modifikāciju šajā algoritmā (to sauc par Smita-Vatermana algoritmu). Tiek ieviests papildu ierobežojums $H_{i,j} \geq 0$, to iekļauj rekursīvā algoritmā, lai izlīdzinājums varētu sākties vai beigties jebkurā virknes vietā.

Abus šos algoritmus var vizualizēt veidojot divdimensionālu līdzinājumu matricu ar pagaidu izlīdzinājuma rezultātiem, kā redzams - att. 2.1 un att. 2.2., kurā punkti ir ņemti no substitūciju matricām, kas aprakstītas nākamajā nodaļā.



att. 2.1 Nīdlmena-Venša algoritms [3]



att. 2.2 Smita-Vatermana algoritms [3]

Procentu identitātes rezultāts katram izlīdzinājumam tiek aprēķināts, izdalot identisko simbolu izlīdzinājumu ar kopējo simbolu skaitu. Katra pozīcija matricā satur rezultātu priekš

labākā pagaidu līdzinājuma, kas beidzas tajā pozīcijā. Visaugstākais rezultāts var tikt paplašināts uz sekojošām matricas pozīcijām vai nu izlīdzinot vienu simbolu no katras sekvences vai, ievietojot starp vienā vai otrā sekvencē. Šādā veidā visi iespējami līdzinājumi tiek apskatīt un beigu izlīdzinājums ir ar visaugstāko iespējamo rezultātu. [3]

2.3. Substitūciju matricas

Kā jau iepriekšējās nodaļās minēts – tad attiecības starp DNS vai olbaltumvielu sekvenču pāriem bieži attēlo ar līdzinājumu. Globāls līdzinājums attiecās pret visu sekvenci, bet lokāls līdzinājums tikai pret katras sekvences segmentu. Līdzinājumiem parasti piešķir rezultātus, gan lai izvēlētos starp daudzām dažādām līdzinājuma variācijām, gan lai salīdzinātu līdzinājumus starp dažādiem sekvenču pāriem. Augstāki vērtējumi tiek uzskatīti par labākiem, un augstākais līdzinājuma vērtējums priekš sekvenču pāra tiek saukts par optimālo līdzinājumu. Šo vērtējumu bieži izmanto arī kā sekvenču līdzīguma parametru.

Līdzinājuma vērtējums bieži tiek definēts kā summa no substitūciju rezultāta nukleotīdu vai aminoskābju pāru novietojuma attiecībā pret līdzinājumu un spraugu rezultāta priekš katras atlikumu virknes vienā sekvencē pret nulles simboliem ievietotiem otrā sekvencē. Substitūciju matrica ir kolekcija no rezultātiem, līdzinot visus iespējamos atlikumu pārus. [16]

Ir vairākas pieejamas substitūciju matricas, gan nukleotīdiem, gan aminoskābēm. Nukleotīdu vērtību matricas gan ir stipri vienkāršākas, tā iemesla dēļ, ka ir tikai četri simboli, līdz ar to tālāk tiks apskatīta plašāk izmantotā olbaltumvielu vērtību matrica BLOSUM62, kas tiek izmantota arī novērtējumā šajā darbā.

Tātad, lai noskaidrotu, vai divas sekvences ir evolucionāri radnieciskas vai nē, ir nepieciešams izlīdzinājuma rezultāts. Lai salīdzinātu divas hipotēzes, var izmantot log-varbūtības mēru – divu hipotēžu varbūtību attiecību logaritms. Ja tiek pieņemts, ka katrs izlīdzinātais atlikumu pāris ir statistiski neatkarīgs no citiem, tad izlīdzinājuma novērtējums ir individuālā log-varbūtības novērtējuma summa, katram pārim. Šie individuālie rezultāti veido 20 x 20 rezultātu matricu. Formula, lai aprēķinātu S(j, k) rezultātu, divām aminoskābēm j un k ir:

$$S(j,k) = \frac{1}{\lambda} \log \frac{P(jk)}{f(j)f(k)} \qquad (2.4)$$

, kur P(jk) ir pārbaudāmo hipotēžu varbūtība (ka šīs divas aminoskābes korelē, jo ir radnieciskas), P(jk) ir varbūtība, ka tiek sagaidīs novērot aminoskābes j un k radnieciskos sekvenču izlīdzinājumos; f(j)f(k) ir nulles hipotēzes varbūtība, ka šīs divas olbaltumvielas ir nesaistītas, un parādās neatkarīgi. Tātad – f(j) un f(k) ir vidējā aminoskābju novērošanas varbūtība olbaltumvielu sekvencē; λ – koeficients, kas parasti tiek uzstādīts, lai varētu noapaļot vērtību matricu uz veseliem un jēgpilniem skaitļiem. Ja tiek sagaidīts, ka j un k izlīdzināts kopā radnieciskā sekvencē atrodas biežāk nekā netīšām (P(jk)>f(j)f(k)), tad varbūtības attiecība ir lielāka nekā viens un novērtējums ir pozitīvs. [17]

Ala	A	4																	
Arg	R	-1	5																
Asn	N	-2	0	6															
Asp	D	-2	-2	1	6														
Cys	C	0	-3	-3	-3	9													
Gln	Q	-1	1	0	0	-3	5												
Glu	E	-1	0	0	2	-4	2	5											
Gly	G	0	-2	0	-1	-3	-2	-2	6										
His	H	-2	0	1	-1	-3	0	0	-2	8									
Ile	I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4								
Leu	L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4							
Lys	K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5						
Met	M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5					
Phe	F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6				
Pro	P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7			
Ser	S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4		
Thr	T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	
Trp	W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11
Tyr	Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7
Val	V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4
		A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V

att. 2.3 BLOSUM62 novērtējuma matrica [18]

att. 2.3 var redzēt BLOSUM62 novērtējuma matricu ar visiem novērtējumiem, kā arī zemāk – att. 2.4 var redzēt divus piemērus substitūciju matricu nukleotīdiem – vienkāršu līdzības matricu un vienkāršu attāluma matricu.

Simple similarity matrix for DNA:

	A	*T*	*G*	*C*
A	1	0	0	0
T	0	1	0	0
G	0	0	1	0
C	0	0	0	1

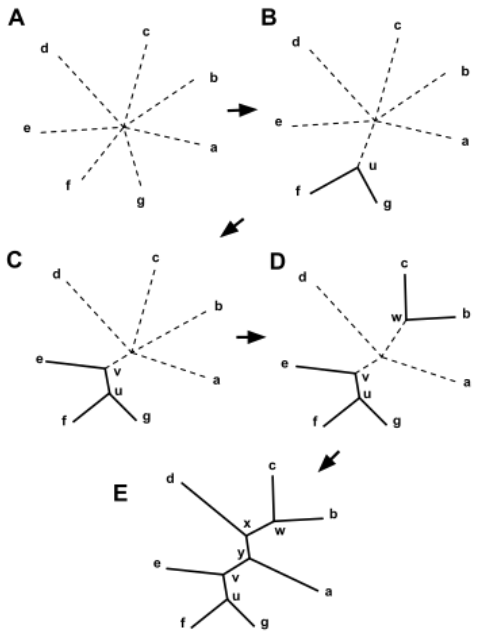
Simple distance matrix for DNA:

	A	*T*	*G*	*C*
A	0	5	5	1
T	5	0	1	5
G	5	1	0	5
C	1	5	5	0

att. 2.4 Nukleotīdu substitūciju matricu piemēri (http://biosiva.50webs.org/alignment.htm)

2.4. Kaimiņu-apvienošanas algoritms

Kaimiņu-apvienošanas metode ir hierarhisks klasteru algoritms. Algoritms sāk ar attāluma matricu *D,* kā ievadi, kur $D_{i,j}$ ir attālums starp klasteriem *i* un *j*. Tad iteratīvi tiek apvienoti algoritmi, izmantojot algoritmu, kas minimizē kopēju zaru garuma summu rekonstruētajā kokā. Vizuāls attēlojums redzams att. 2.5.



att. 2.5 Kaimiņu apvienošanas algoritms [19]

Tātad algoritms izmanto *n* iterācijas, kur divi klasteri (i un j) tiek izvēlēti un apvienoti jaunā klasterī. Klasteri tiek izvēlēti minimizējot:

$$Q(i,j) = D(i,j) - u(i) - u(j) \qquad (2.5.)$$

, kur

$$u(l) = \sum_{k=0}^{r-1} \frac{D(l,k)}{(r-2)} \qquad (2.6.)$$

r – atlikušais klasteru skaits. Kad minimālā Q vērtība tiek atrasta, tad D (attālumu matrica) tiek atjaunota, izņemot no tās i-tās un j-tās rindas un kolonnas. Jauna rinda un kolonna ar jauno

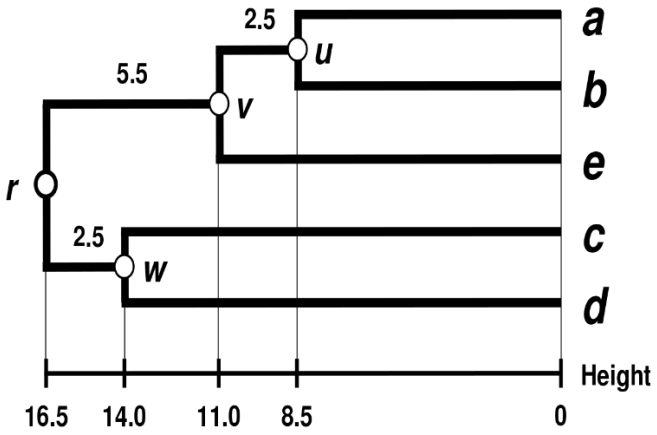
klasteri tiek pievienota. Attālumi starp jauno klasteri a = i ∪ j un vecais klasteris k tiek aprēķināts kā:

$$D(a,k) = \frac{D(i,k) + D(j,k) - D(i,j)}{2} \qquad (2.7.)$$

Algoritma rezultāts ir divu virzienu koks bez saknes, kurā katrs sākotnējais klasteris atbilst lapai un katrs apvienojums veido iekšēju mezglu. Atrast klasteru pāri, ko apvienot katru iterāciju paņem sarežģītību O(n^2). Tātad šī algoritma kopējā sarežģītība ir O(n^3). [20] Arī šim algoritmam ir dažādi iespējami uzlabojumi, šajā nodaļā aprakstīta tā klasiskā versija.

2.5. UPGMA

Viens no vienkāršākajiem koku veidošanas algoritmiem – UPGMA (nesvērtu pāru grupu metode ar aritmētisku vidējo jeb *Unweighted Pair Group Method with Arithmetic mean*). Tā ir vienkārša, hierarhiska klasterēšanas metode. Uzskatīta par vienkāršāko un ātrāko metodi priekš koka ar sakni veidošanas filoģenētikai. Taču galvenā problēma ir tā, ka šī metode pieņem vienādus evolucionāros tempus pilnīgi visām līnijām, jebšu pieņem, ka mutācijas temps šajās līnijās ir konstants laikā. Sanāk, ka algoritms veido zarus ar ļoti līdzīgiem attālumiem. Taču tā kā realitāte mutācijas temps atšķiras gan laikā, gan dažādās sugu līnijās, bieži rezultāti nav uzticami. [21]



att. 2.6. UPGMA koks [22]

Pirmajā solī divi beigu taksoni ar mazāko ģenētisko distanci (piemēram, i un j) tiek klasterizēti kopā, veidojot takosonomisku vienību k. Tad jauna mazāka attāluma matrica tiek aprēķināta, kas iekļauj k, nevis i un j – līdzīgi kā kaimiņu apvienošanā. Šajā procesā vidējie tiek

izmantoti, lai atvasināti attālumus starp jaunajām taksonomiskajām vienībām un atlikušajām beigu taksoniem. Šo distanci aprēķina:

$$D(i,j)x = \frac{D(i,x) + D(j,x)}{2} \qquad (2.8)$$

Katram taksonam X. Nākamajā iterācijā atkal divi taksoni ar mazāko attālumu tiek klaserizēti, un šis process tiek pabeigts, līdz brīdim, kad tikai divas taksonomiskās vienības ir palikušas.

2.6. Slēptie Markova Modeļi

Slēptie Markova Modeļi (*Hidden Markov Models,* jeb *HMM*), ir formāls pamats, lai veiktu varbūtības modeļus lineāru sekvenču apzīmēšanas problēmām. Tie dod konceptuālu rīku, lai būvētu sarežģītus modeļus, uzzīmējot intuitīvu bildi. Tie ir pamatā lielam apjomam programmu, kuras risina gēnu atrašanas, profilu meklēšanas un MSA problēmas. [23]

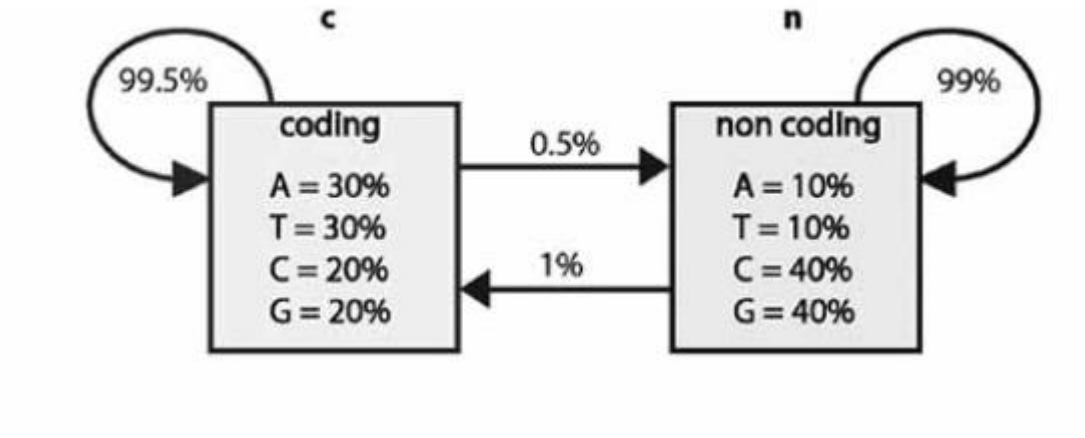
HMM ir statistisks modelis, kas var tikt izmantots, lai aprakstītu novērojamu notikumu evolūciju, kas ir atkarīga no iekšējiem parametriem, kurus nevar tieši novērot. Novēroto notikumu sauc par simbolu un neredzamo faktoru, kas ir novērojuma pamatā – par stāvokli. HMM sastāv no diviem stohastiskiem procesiem – no slēptu stāvokļu neredzama procesa un novērojamu simbolu redzama procesa. Slēptie stāvokļi veido Markova ķēdi un novēroto simbolu varbūtības sadalījumu, kas ir atkarīgs no tā stāvokļa.

Modelēt novērojumus šajos divos slāņos – redzamajā un neredzamajā ir ļoti noderīgi tāpēc, ka daudzas reālās problēmas risina neapstrādātu novērojumu klasificēšanu dažādās kategorijās jeb klašu apzīmējumiem, kas ir jēgpilnāki cilvēkiem. Piemēram, runas atpazīšanas problēmā, kurai HMM tika izmantoti vairākas desmitgades. Ideja ir paredzēt, kāds ir izrunātais vārds no ierakstītā runas signāla. Priekš šī nolūka runas atpazinējs mēģina atrast stāvokļu secību, kura radīja celšanos reālajā skaņā (novērojumā). Tā kā var būt ļoti dažādas izrunas variācijas, oriģinālie stāvokļi un attiecīgi izrunātie vārdi nevar tikt tieši novēroti un tos ir nepieciešams prognozēt.

Šis piegājiens ir noderīgs modelējot bioloģiskas - gan olbaltumvielu, gan DNS sekvences. Tipiski bioloģiska sekvence sastāv no mazākām apakšstruktūrām ar dažādām funkcijām un dažādiem funkcionālajiem reģioniem, kuri bieži attēlo īpašas statistiskas īpašības. Piemēram ir zināms, ka olbaltumvielas parasti sastāv no vairākiem domēniem. Ja tiek dota jauna olbaltumviela,

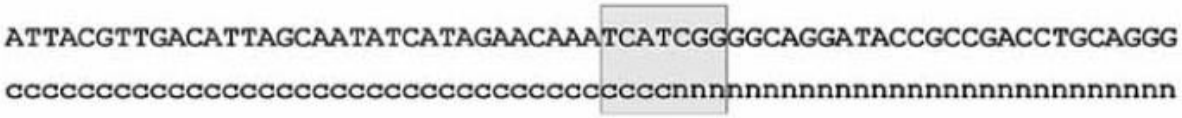
tad būtu interesanti paredzēt tā sastāvošos domēnus (kuri atbilst vienam vai vairākiem stāvokļiem HMM) un tā atrašanās vietas aminoskābju sekvencē (novērojumos) vai arī tai radnieciskās olbaltumvielas. [24]

Pavisam triviāls bioloģisks piemērs HMM ar tikai diviem iekšējiem stāvokļiem bet ar eksponenciālu sarežģītību. Modelis redzams - att. 2.7. Attēlā redzamas katras emisijas (A, T, C un G) varbūtības katram stāvoklim. Ārpus kastes četras bultas ir apzīmētas ar pārejas varbūtību.



att. 2.7 HMM piemērs [25]

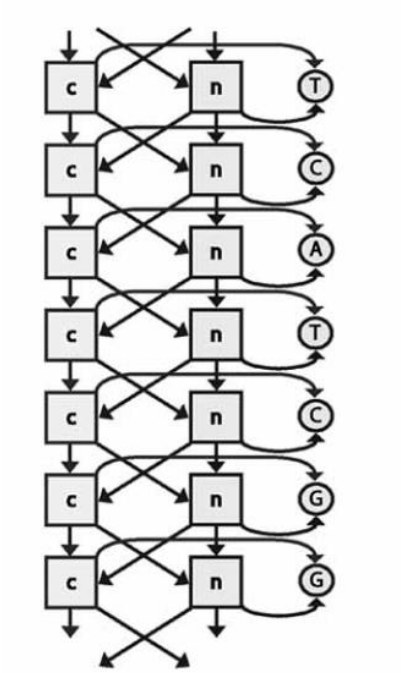
Iekšējo stāvokļu kopa U={‘c’, ‘n’}, kur c un n ir kodējošie un nekodējošie iekšējie stāvokļi un emisijas kopa ir kopa no četrām DNS bāzēm X = {‘A’, ‘T’, ‘C’, ‘G’}. Kā emisijas sekvenci mēs pieņemam sekvenci ar 65 bāzēm - att. 2.8. Pirmā rinda ir parauga sekvence ar 65 novērotām emisijām un otrā rinda ir viena no iekšējo stāvokļu paredzamajām sekvencēm. Apvilktā sekvences daļa paskaidrota - att. 2.9 un att. 2.10.



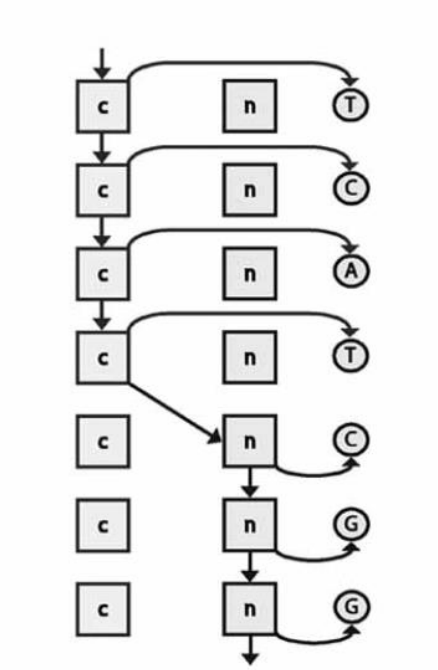
att. 2.8. Emisijas sekvence [25]

Svarīgi pieminēt, ka vairākumā HMM pielietojumu bioinformātikā norisinās simulēta inversija starp sekām un izraisītājiem. Piemēram, var tikt sintezēta zināma polimēru sekvence, kurai ir dažādas nezināmas īpašības sekvencē. HMM nepieciešams izvēlēties emisijas kā sekvences monomerus tāpēc, ka tikai tie ir zināmie dati un kā iekšējie stāvokļi to īpašības jānovērtē. Šādā

veidā parāda, ka sekvence ir sekas un īpašības ir izraisītājs, kaut gan realitātē ir pretēji. Lielisku piemēru dod polipeptīdi, kuriem tikai aminoskābju sekvence izraisa sekundāro struktūru, kaut gan iekš HMM aminoskābes tiek uztvertas kā emisijas un sekundārā struktūra – iekšējais stāvoklis.



att. 2.9. [25]



att. 2.10. [25]

Labajā kolonnā attēlotas apvilktās bāzes iekš att. 2.8, pārējās kolonnas attēlo katrai apvilktajai bāzei divas alternatīvas iekšējam stāvoklim (c vai n), kas emitēja bāzi. Katra rinda atsaucas uz to pašu pozīciju sekvencē, bultas reprezentē visas iespējamās pārejas un emisijas.

att. 2.10 – Attēlā redzams scenārijs ar lielu varbūtību starp izvēlēm starp alternatīviem iekšējiem stāvokļiem radot iekšējo stāvokļu sekvenci. [25]

3. PROGRAMMU APSKATS

Šajā nodaļā tiek apskatītas šobrīd praksē pielietotas programmas, iztirzāti to darbības principi, pieejamība, ātrdarbība, kā arī ievades un izvades informācijas iespējas. Darbā apskatītās metodes ir atvērtā koda programmatūras ar licencēm, kas ļauj tās brīvi izmantot, kā arī bieži izmantotas praksē. Šo programmu pamatā, tiek lietoti iepriekšējā nodaļā ALGORITMI aprakstītie algoritmi. Apraksts par katru programmu ir īss, lai sniegtu ieskatu par būtiskākajiem aspektiem, kas atrodami literatūrā, un tiek izcelti no programmu autoru puses.

Šīs četras programmas tika izvēlētas tāpēc, ka tās ir atvērta koda un brīvpieejas programmas, kuras visbiežāk tiek izmantotas, lai veiktu vairāku sekvenču izlīdzināšanu.

3.1. Clustal Omega

Clustal Omega ir pagaidām jaunākais MSA algoritms Clustal ģimenē. Sākotnēji tas bija tikai olbaltumvielu sekvencēm, taču kopš 2019. gada tika papildināts, nodrošinot arī nukleotīdu sekvenču izlīdzināšanu. Šī programmatūra ir rakstīta C un C++ valodās.

Kā paši Clustal Omega autori raksta savā publikācijā, kurā iepazīstina ar savu gara darbu: “Lielākā daļa automātisko MSA metožu izmanto progresīvās izlīdzināšanas heiristiku, kura izlīdzina sekvences lielākos un lielākos apakšlīdzinājumos, sekojot zarošanās kārtībai kokā. Ar sarežģītību $O(N^2)$ šis piegājiens var padarīt līdzinājumus, kas ir lielāki par pāris tūkstošiem sekvenču un nav liela garuma, taču, kad apjomi paliek lielāki, rodas problēmas. Progresīvais piegājiens ir ‘alkatīgs algoritms’, kur kļūdas, kas ir notikušas sākotnējās līdzināšanas soļos, nevar tikt izlabotas vēlāk. Lai ar to cīnītos, tikai izveidots konsistences princips. Tas ir atļāvis radīt jaunu ģenerāciju ar precīzākiem līdzinātājiem, taču uz skaitļošanas jaudas rēķina. Šīs metodes piedāvā 5-10% precizitātes uzlabojumu, taču tiek ierobežotas uz pāris simtiem sekvenču.” [5]

Autori apgalvo, ka Clustal Omega ir tikpat precīzs, taču atļauj izlīdzinājumus gandrīz jebkādā izmērā, pat 190 000 sekvenču uz viena procesora pāris stundās. Novērtējuma testos Clustal Omega esot daudz akurātāks, kā izmantotās “ātrās” metodes, un salīdzināms precizitātē ar lēnajām metodēm.

Clustal Omega atšķirīgais piegājiens ir metode, kura izmantota ceļveža koka veidošanā. Parasti tā sarežģītība gan laikā, gan atmiņas komponentēs ir $O(N^2)$, kur N – sekvenču skaits. Tiek izmantots mBed algoritms, lai veidotu ceļveža koku, kura sarežģītība ir $O(N*\log(N))$ un koki esot

tikpat precīzi, cik citām metodēm. mBed algoritms iegulda katru sekvenci *n* dimensijās, kur *n* ir proporcionāls log N. Katra sekvence tiek aizvietota ar *n* elementu vektoru, kur katrs elements ir attālums līdz vienai no *n* references sekvencēm. Šie vektori tiek klasterizēti, izmantojot standarta metodes (piemēram, iepriekš apskatītā UPGMA). [5]

3.2. Kalign

Kalign ir visnesenāk atjaunotā no visām šajā darbā apskatītajām programmatūrām, publicēts raksts 2019. gada Oktobrī. Šī programmatūra ir rakstīta valodā C ar GNU publisko licenci.

Kalign ir progresīvā līdzinājuma metode, kas autora vārdiem "uztur labu balansu starp precizitāti un ātrumu, salīdzinot ar citām izlīdzināšanas programmām". Tā kā iepriekšējā Kalign versija nespēja apkalpot desmit tūkstošus sekvenču, kas nereti ir sastopams mūsdienās, tika radīta jauna.

Jaunā Kalign versija izmanto Džīna Maijersa (*Gene Myers)* aptuveno virknes savietošanas algoritmu. Algoritms aprēķina precīzu attālumu starp divām virknēm, izmantojot paralēlu bitu instrukcijas. Standarta implementācijā maksimālais vaicājuma garums ir vienāds ar datora vārdu (64 simboli), taču algoritms izmantojot tālāku paralelizāciju, izmantojot SIMD instrukcijas, iekļaujot AVX un AVX2 instrukcijas, kas ir pieejamas uz visiem modernajiem datoriem. Izmantojot šīs instrukcijas, kļūst iespējams salīdzināt sekvences ar garumu 256.

Lai novērtētu aptuvenu pārisko sekvenču attālumu Kalign skenē pirmos 256 simbolus no īsākās sekvences pār garāko sekvenci. Attālums tiek definēts kā skaits, cik reižu ir nepieciešams labot vienu sekvenci, lai padarītu to par precīzu otras sekvences kopiju, Priekš attāli radnieciskām olbaltumvielu sekvencēm sekvenču līdzība ir pārāk zema, lai algoritms varētu noteikt jēgpilnus attālumus. Līdz ar to Kalign pārvērš visas olbaltumvielu sekvences samazinātā alfabētā. [26]

Tālāk Kalign izmanto ceļvežu koku konstrukcijas metodes tādas pašas kā Clustal Omega.

3.3. MAFFT

MAFFT *(Multiple sequence Alignment Fast Fourier Transform)* izmanto divas tehnikas, kas atšķiras no citām metodēm – pirmkārt, homologus reģionus identificē ar ātrajām Furjē transformācijām *(Fast Fourier transform)*. Šajā metodē aminoskābju sekvences pārveido uz sekvencēm, kuras satur tilpuma un polaritātes vērtības katram aminoskābes atlikumam. Otrkārt,

vienkāršota punktu sistēma tiek pielietota, kas samazina skaitļošanas laiku un palielina līdzinājumu precizitāti. MAFTT izmanto divu ciklu heiristiku – progresīvo metodi – FFT-NS-2 un iteratīvo precizēšanas metodi – FFT-NS-i. Progresīvajā metodē zemas kvalitātes pāriskās distances tiek ātri izskaitļotas, tiek konstruēts pagaidu MSA, precizētas distances tiek aprēķinātas no MSA. FFT-NS-i metode ir viena cikla progresīva metode, tā ir ātrāka un mazāk precīza kā FFT-NS-2. [27, 28]

3.4. MUSCLE

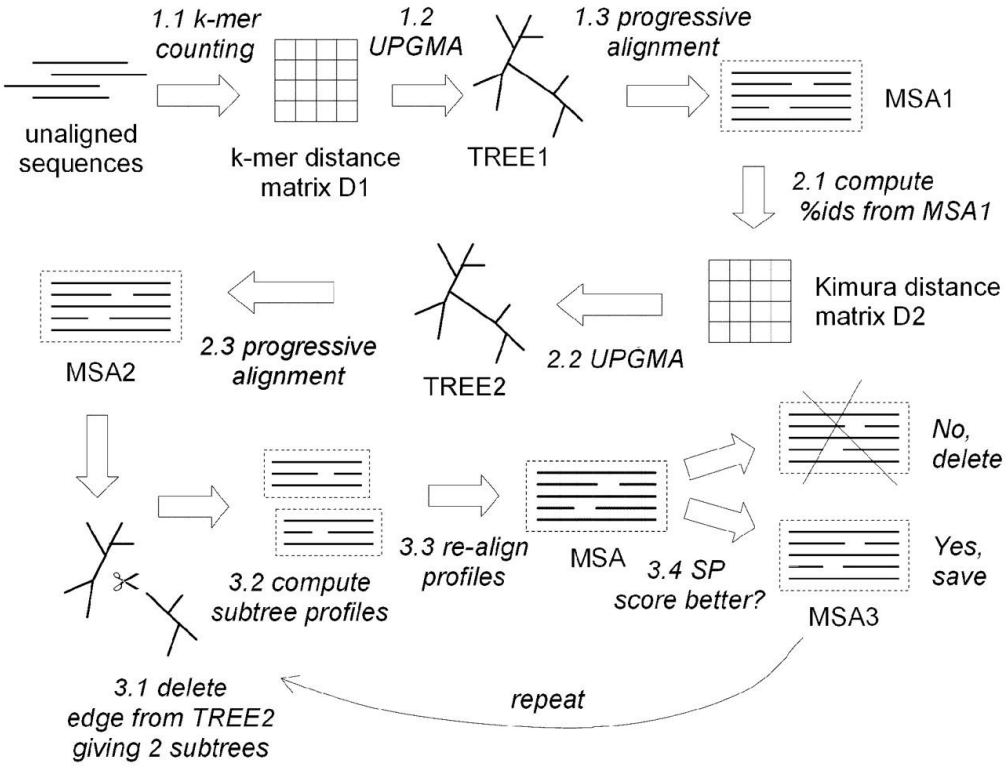
MUSCLE programmatūra arī ir rakstīta C++ programmēšanas valodā, neierobežotu licenci un ar atvērto kodu.

"Algoritms ir sekojošs:

1) Pirmā etapa mērķis ir veikt MSA, izvēloties ātrumu virs precizitātes:
    a. Tiek aprēķināts k-meru attālums katram ievades sekvenču pārim, iegūstot attālumu matricu D1
    b. D_1 tiek klasterizēts izmantojot UPGMA, radot bināro koku K_1
    c. Tiek veikts progresīvais līdzinājums, sekojot zarošanās secībai K_1. Katrai lapai tiek veidots profils no ievades sekvences. Mezgli kokā tiek apmeklēti prefiksa kārtībā (apakšmezgls pirms virsmezgla). Katrā iekšējā mezglā tiek veikts pāriskais līdzinājums no diviem apakšējiem profiliem, radot jaunu profilu, kuru pievieno attiecīgajam mezglam. Tas rada MSA visām ievades sekvencēm, ko sauksim par MSA_1.
2) Galvenais kļūdas cēlonis pirmajā solī ir aptuvenā k-meru attāluma mērīšana, kas rezultējas neprecīzā kokā. MUSCLE tāpēc pārrēķina koku, izmantojot Kimura attālumu, kas ir precīzāks, taču tam ir nepieciešams līdzinājums.
    a. Tiek aprēķināta Kimura distance katram ievades sekvenču pārim no MSA_1, radot jaunu attāluma matricu D_2.
    b. Matrica D_2 tiek klasterizēta, izmantojot UPGMA, radot koku K_2.
    c. Tiek veidots progresīvais līdzinājums tāpat kā solī 1.c. Šis tiek optimizēts, aprēķinot līdzinājumus tikai tiem apakškokiem, kuru zarošanās secība mainījās salīdzinot ar K_1
3) Pārskatīšana (iteratīvs solis, kas atkārtojas vairākas reizes, līdz konverģencei vai arī lietotāja definētam limitam)

a. Mala no K_2 tiek izvēlēta (malas tiek apmeklētas dilstošā attāluma secībā, sākot no saknes)
b. K_2 tiek sadalīts divos apakškokos, izdzēšot malu. Katra apakškoka izlīdzinājuma profils tiek aprēķināts
c. Jauns MSA tiek radīts atkal, izlīdzinot divus profilus.
d. Ja rezultāts ir uzlabojies, tad jaunais līdzinājums tiek atstāts, ja nē, tad izdzēsts."

[29]



att. 3.1 MUSCLE algoritma vizualizācija [29]

MUSCLE algoritma vizualizācija ir redzama - att. 3.1.

4. METOŽU SALĪDZINĀJUMA PARAMETRI

Lai varētu salīdzināt vairāku sekvenču izlīdzināšanas metodes, nepieciešams definēt parametrus, pēc kuriem šīs metodes tiks vērtētas, kuras arī šajā nodaļā tiek apskatītas. Literatūrā sastopami ļoti daudz un dažādi veidi, kā iespējams šos izlīdzinājumus novērtēt. Šajā darbā tiek izvēlēti visbiežāk sastopamie no kritērijiem.

4.1. BAliBASE

Vēsturiski jaunu vairāku sekvenču izlīdzinājuma programmatūru novērtējumu veica ar mazu apjomu programmas autora izvēlētu testu kopu. Nedaudz vēlāk, pašās 20. gs. beigās sāka izmantot izlīdzinājumu kopas, no strukturālām datubāzēm, taču tās datubāzes, kas tajā brīdī bija pieejamas, apvienoja olbaltumvielas radnieciskās kopās. Izlīdzinājuma netika strukturēti un klasificēti specifiski priekš sistemātiskas MSA programmatūras novērtēšanas.

Lai veiktu šo novērtējumu korekti, nepieciešams liels daudzums precīzu references izlīdzinājumu, kurus var izmantot kā testa kopas. Izlīdzināšanas efektivitāte mēdz būt atkarīga no daudz un dažādiem faktoriem, piemēram, sekvenču skaita, sekvenču garuma, sekvenču līdzīguma, inserciju skaita līdzinājumā, u.tml. Tāpēc sākotnēji tika radīta datu bāze BAliBASE (Novērtējuma Izlīdzinājuma datubāze – *Benchmark Alignment dataBASE*), kas satur augstas kvalitātes, manuāli veidotus, dokumentētus izlīdzinājumus, lai identificētu stiprās un vājās puses pieejamajām izlīdzināšanas programmām. Izlīdzinājumi ir balstīti uz trīs dimensiju strukturālajām superpozīcijām (izņemot transmembrānu sekvences).

Pirmā BAliBASE versija satur 142 references izlīdzinājumus, kas satur vairāk nekā 1000 sekvenču. Izlīdzinājumi tiek sadalīti četrās hierarhiskās referenču kopās, katra no šīm kopām var tikt sadalīta vēl mazākās grupās, atkarībā no sekvenču garuma un līdzīguma. Pirmajā versijā references izlīdzinājumi risina problēmas ar augstu mainību, nevienādu pārdalīšanu un iekšējās insercijas.

Otrajā datubāzes versijā tiek pievienotas trīs jaunas referenču kopas ar izlīdzinājumiem, kas satur strukturālus atkārtojumus, transmembrānu sekvences un cirkulāras permutācijas, lai novērtētu atrašanas / prognozēšanas precizitāti un šo komplekso sekvenču izlīdzināšanu.

Trešā versija, kas, izdota 2005. gadā, iekļauj jaunas, izaicinošākas testa kopas, kas precīzāk ataino reālās problēmas, ar kurām sastopas, kad tiek līdzinātas lielas un sarežģītas sekvenču kopas.

Izmantojot jaunu un pusautomātisku atjaunošanas protokolu, olbaltumvielu skaits BAliBASE datubāzē tiek palielināts līdz 6255 sekvencēm. Papildus tiek pievienotas pilna garuma sekvences priekš visiem testu gadījumiem, kas atspoguļo grūtākos gadījums priekš globāla un lokāla izlīdzinājuma programmām. Ceturtā versijai tiek papildināta ar jaunām sekvencēm, kas vairāk atbilst mūsdienu sekvenču izlīdzināšanas prasībām. [30, 31, 32, 33]

4.2. Pāru summas novērtējums

Pāru summas rezultāts (*Sum-of-Pairs score – SP*) ir viens no parametriem, ar kuru ir iespējams novērtēt vairāku sekvenču izlīdzināšanas kvalitāti. Tā ir plaši izplatīta metode, kas aprēķina novērtējumus visiem pāru izlīdzinājumiem no n sekvencēm un papildus var novērtēt arī vairāku sekvenču izlīdzinājumu ar SP novērtējumu, kura formula ir:

$$SP\ novērtējums = \sum_{j=1}^{n-1} \sum_{k=j+1}^{n} S(j,k) + GP \qquad (4.1)$$

, kur S(j, k) atgriež vērtību no vērtību matricas priekš atlikumu pāra j un k pozīcijā un GP – iepriekš definēts spraugas "sods". [34]

4.3. Kopējais kolonnu novērtējums

Kopējais kolonnu novērtējums (*Total Column score – TC score*) ir parametrs, kurš savā vērtējumā sniedz informāciju par to, cik precīzu kolonnu ir šajā līdzinājumā. Kolonna ir līdzinājumā viena simbola atrašanās vieta visās sekvencēs.

Kopējais kolonnu novērtējums ir pavisam vienkārša vērtēšanas sistēma priekš MSA, kas katru kolonnu, kas satur vienādus atlikumus (attiecībā pret references izlīdzinājuma kolonnu) novērtē ar 1 un katru kolonnu, kurā tā nav – ar 0, tādejādi atgriežot precīzo kolonnu skaitu.

$$TC\ novērtējums = \sum_{i=1}^{d} \begin{cases} 0, & ja\ atlikumi\ ir\ vienādi \\ 1, & ja\ atšķirīgi \end{cases} \qquad (4.2)$$

, kur d = kolonnu skaits. [34]

4.4. Skaitļošanas ilgums

Skaitļošanas ilgums ir ļoti svarīgs parametrs, pēc kura izvērtēt MSA programmas, jo palielinoties sekvenču skaitam, un lielumam šī problēma sāk kļūt ļoti aktuāla. Kā arī nākotnē bioloģisko datu apjoms tikai palielināsies un algoritma efektivitāte kļūst svarīgāka. Lai objektīvi varētu novērtēt parametrus, kas attiecināmi uz skaitļošanas ilgumu un jaudu, visi izlīdzinājumi tiek veikti uz viena datora ar parametriem:

- Procesors: Intel® Core™ i7-8750H CPU @ 2.20 GHz
- Atmiņa: DDR4 24 GB

Visas programmas tiek izpildītas Ubuntu 18.04 operētājsistēmā.

5. PROGRAMMU IZVĒRTĒJUMS

Šī darba rezultāti ir uz dažādām datu kopām veikti izlīdzinājumi ar visām darbā apskatītajām programmām un to novērtējums, izmantojot iepriekšminētos parametrus. Sākotnēji tika sagrupētas visas pieejamās un iepriekš apskatītās datu kopas un to datnes nosauktas atbilstoši to pārstāvētajai kategorijai, lai atvieglotu rezultātu analizēšanas procesu. Lai veiktu izlīdzinājumus, tika izmantots Python skripts, kas atrodams sadaļā PIELIKUMI, kurā tiek veikti izlīdzinājumi, katrai datnei ar katru programmatūru, kā arī pēc katras komandas izpildes tiek pierakstīts komandas izpildes laiks, lai varētu novērtēt katras programmas veiktspēju pēc tās izpildes ilguma.

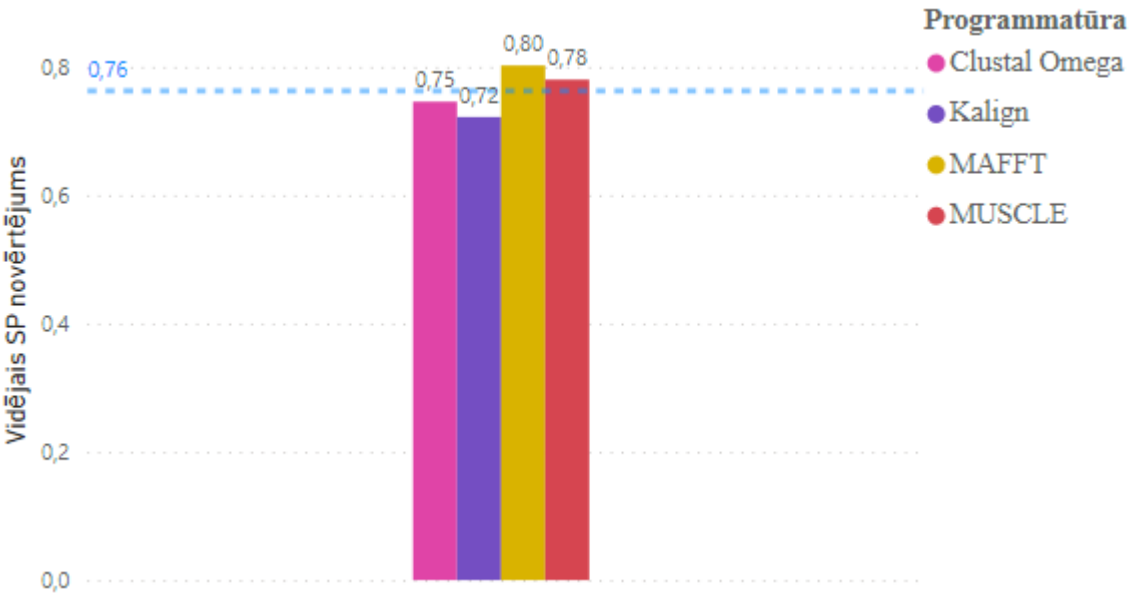
Šajā nodaļā rezultāti tiks sagrupēti pēc tā, kādas grupas dati tiek analizēti, un pastāstīts vairāk par datiem, kuri tiek izlīdzināti, lai vērstu uzmanību uz katras programmatūras "stiprajām" vai "vājajām" pusēm, atkarībā no to pielietojuma veida, kopā tiks izvērtētas 10 dažādas kopas.

5.1. BAliBASE references kopas

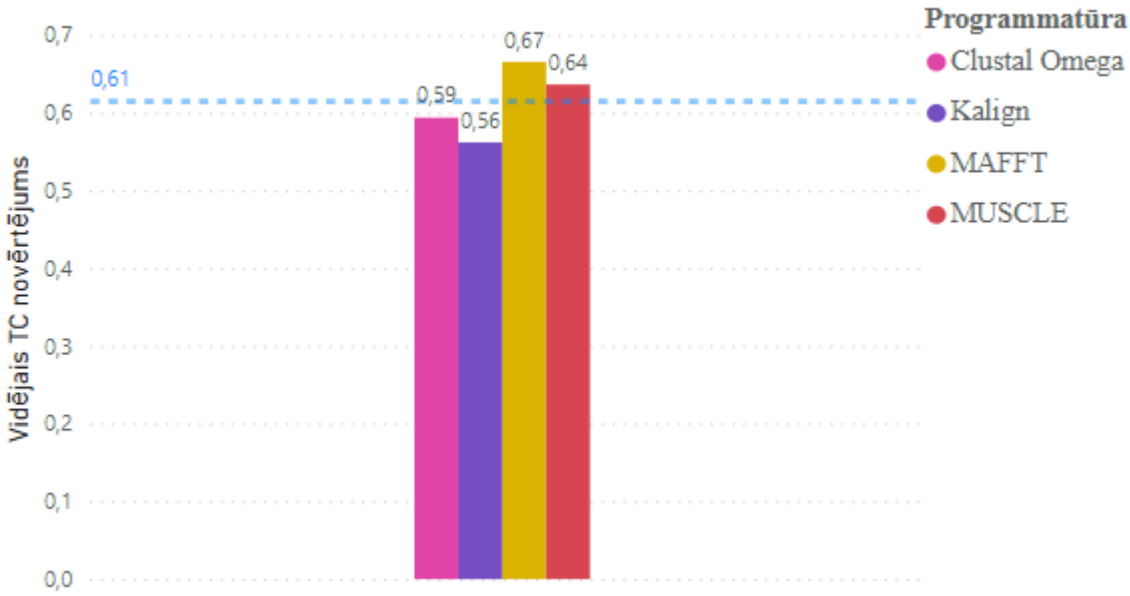
Šajā nodaļā tiks izklāstīts grafisks iegūto rezultātu detalizēts apskats, sākotnēji par katru (vai arī dažās nodaļās kopas ir apvienotas, ja nav iemesla izdalīt atsevišķi) funkcionālo bioloģisko datu kopu atsevišķi. Sarakstā pēc secības iztrūkst 6 kopa, kura programmu izpildīšanas brīdī bija brāķētā stāvoklī un netika izpildīta.

5.1.1. Pirmā kopa

Pirmā BAliBASE kopa ir sekvenču kopums, kas satur 164 FASTA failus, kuros katrā ir 6 vai mazāk vienāda attāluma sekvences, šajā kopā ir divas apakškopas, kuras atsevišķi neizdalīsim, taču var minēt, ka vienā ir sekvences ar lielu atšķirību (mazāk par 20% identitāte), otrā ir vidēja atšķirība (20 – 40 % identitāte). [30]

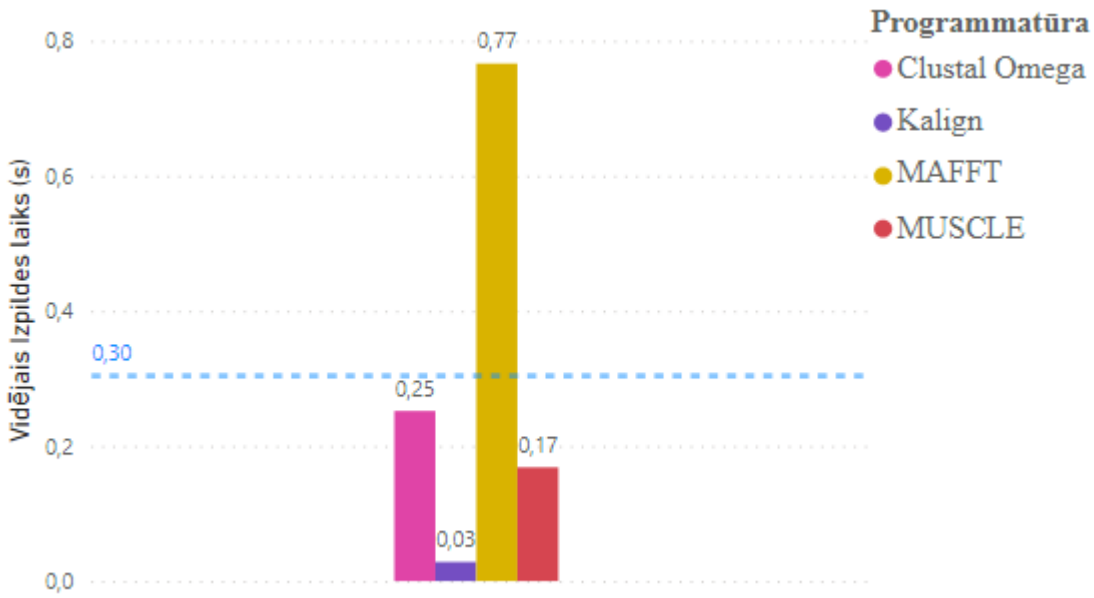


att. 5.1 SP novērtējums 1. kopai



att. 5.2 TC novērtējums 1. kopai

Pēc att. 5.1 un att. 5.2 datiem var redzēt, ka MAFFT programma ir ar visaugstāko rezultātu gan pēc SP rezultāta, gan pēc TC rezultāta, taču starpība starp otro augstvērtīgāko rezultātu, kas ir MUSCLE programmai, vai pat zemāko rādītāju Kalign nav liela, bet visi rezultāti ir 10% robežās no vidējās vērtības.

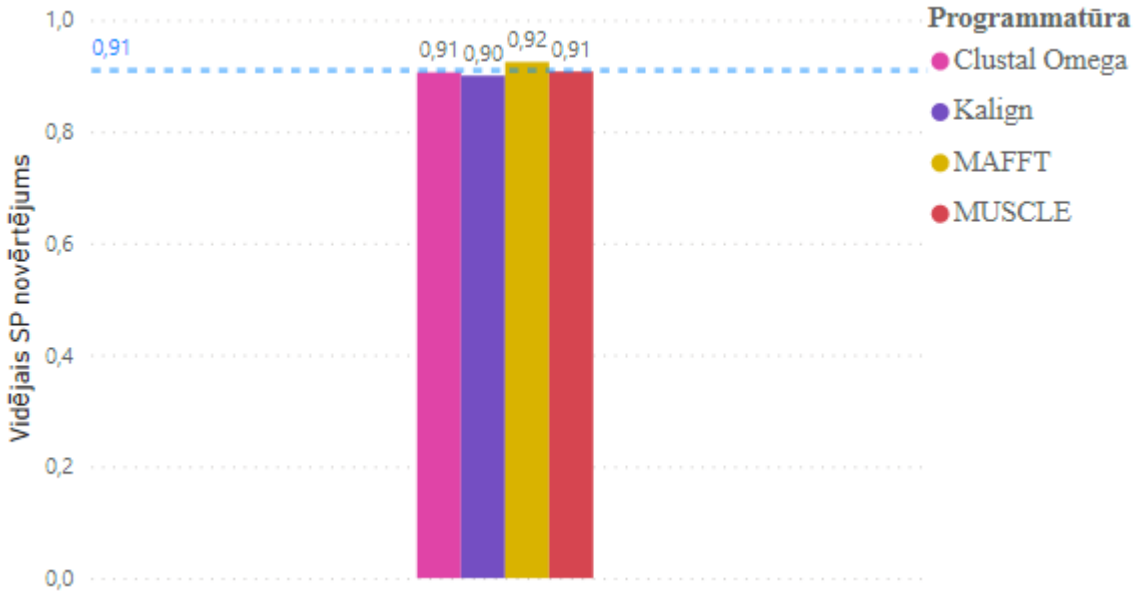


att. 5.3 Vidējais izpildes laiks 1. kopai

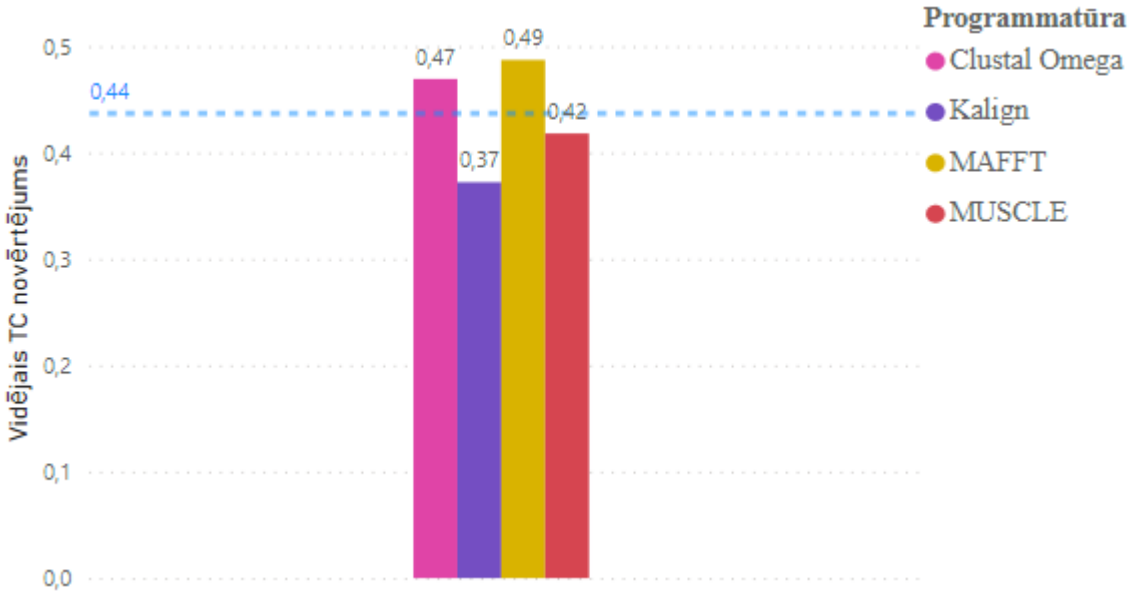
Izvērtējot izpildes laiku, var novērot lielu starpību starp ātrāko (Kalign) un lēnāko (MAFFT), kurā MAFFT programmas izpildes laiks no pārējiem rezultātiem ir ar 2.5 reizes ilgāku vidējo izpildes laiku kā vidējais izpildes laiks. Toties var novērtēt arī Kalign ātrdarbību, kur rezultāts ir 10 reizes īsāks, kā vidējais lielums, tomēr arī jāpiemin, ka pēc TC un SP rādītājiem Kalign bija ar viszemāko vērtību.

5.1.2. Otrā kopa

Otrā kopa ir radnieciskas sekvences un atšķirīga "bāreņu sekvence". Kopa satur 82 datnes. [30]

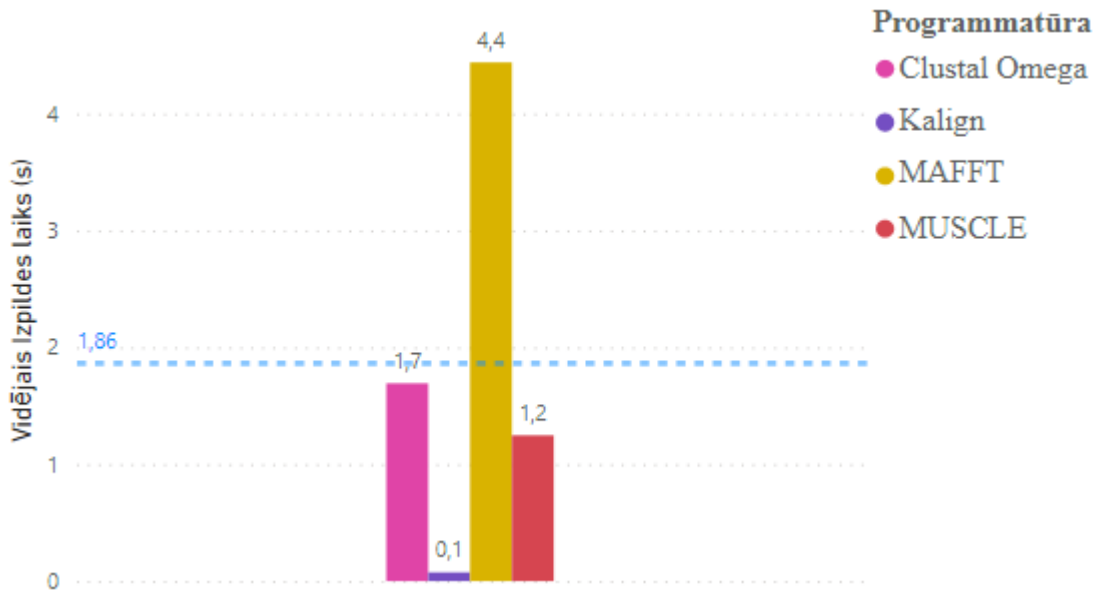


att. 5.4 SP novērtējums 2. grupai



att. 5.5 TC novērtējums 2. grupai

Pēc 2. grupas datiem - att. 5.4 un att. 5.5 ir ļoti labi novērojams, kāpēc ir nepieciešami vairāki parametri, lai izvērtētu vairāku sekvenču izlīdzināšanas metodes, pēc SP novērtējuma rezultāti ir praktiski vienādi un ļoti tuvu 1, kas liecinātu par to, ka izlīdzinājumi visi ir vienādi un augsti kvalitatīvi, taču, ņemot vērā datu kopas specifiku, ka tajā ir iekļauta šī viena "bāreņu sekvence", kura ir atšķirīga no citām, tad ir nepieciešams arī otrs novērtējums, kurā varam redzēt, ka starpība starp novērtējumiem ir. Tāpat kā pirmajai grupai, augstākais rezultāts – MAFFT programmai, zemākais Kalign, taču šai grupai Clustal Omega rezultāts ir ļoti tuvs MAFFT.

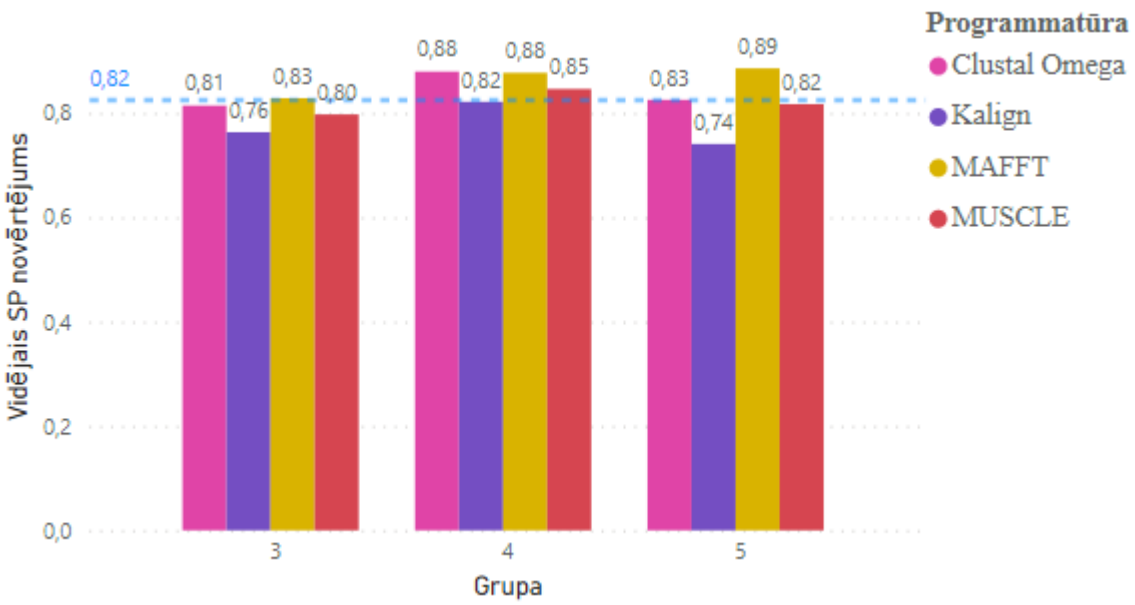


att. 5.6 Vidējais izpildes laiks 2. grupai

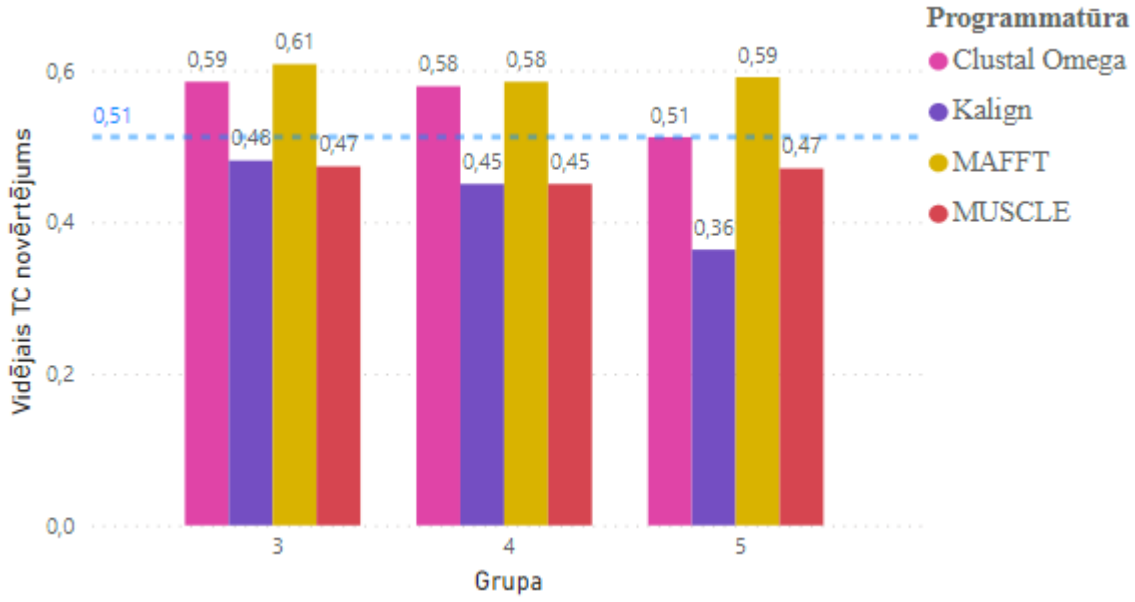
Vidējais izpildes laiks att. 5.6 uzrāda praktiski identisku ainu iepriekšējai grupai, kur visātrākais ar lielu starpību ir Kalign, lēnākais – MAFFT un Clustal Omega nedaudz lēnāka nekā MUSCLE.

5.1.3. Trešā, ceturtā un piektā kopa

Trešās, ceturtās un piektās kopas rezultāti savā starpā ir ļoti līdzīgi, tādēļ tie tiek apvienoti vienos grafikos. Trešās kopas dati – 60 datnes, kuros ir sekvences ar apakšgrupām, kuru identitāte starp grupām ir mazāka par 25%. Ceturtās kopas dati – 49 datnes, sekvences ar N/C beigu paplašinājumiem. Piektā kopa – 31 datne, sekvences ar iekšējām insercijām. [30]

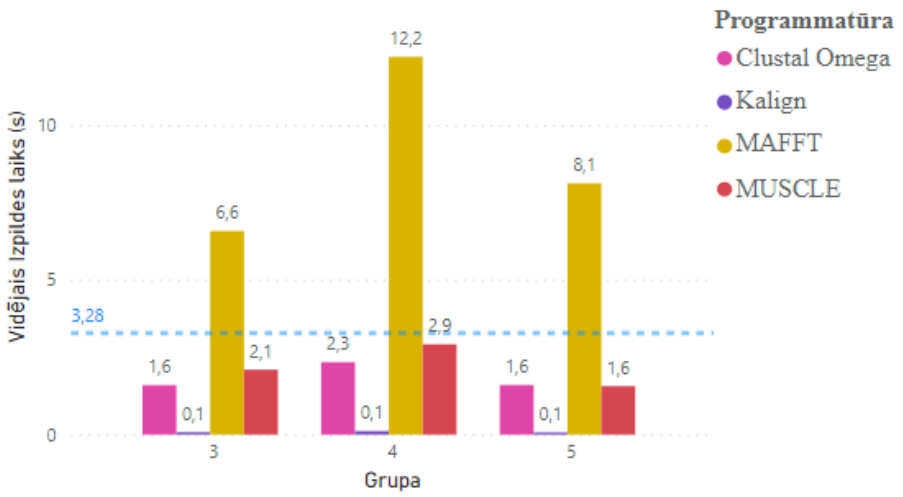


att. 5.7 SP novērtējums 3., 4. un, 5. grupai



att. 5.8 TC novērtējums 3., 4., un 5. grupai

3., 4. un 5. grupas dati att. 5.7 un att. 5.8 uzrāda tādu pašu tendenci, kā ir novērots jau iepriekšējās sekvenču kopās – augstākie rezultāti MAFFT programmai, kurai šajās kopās ļoti tuvu seko Clustal Omega.

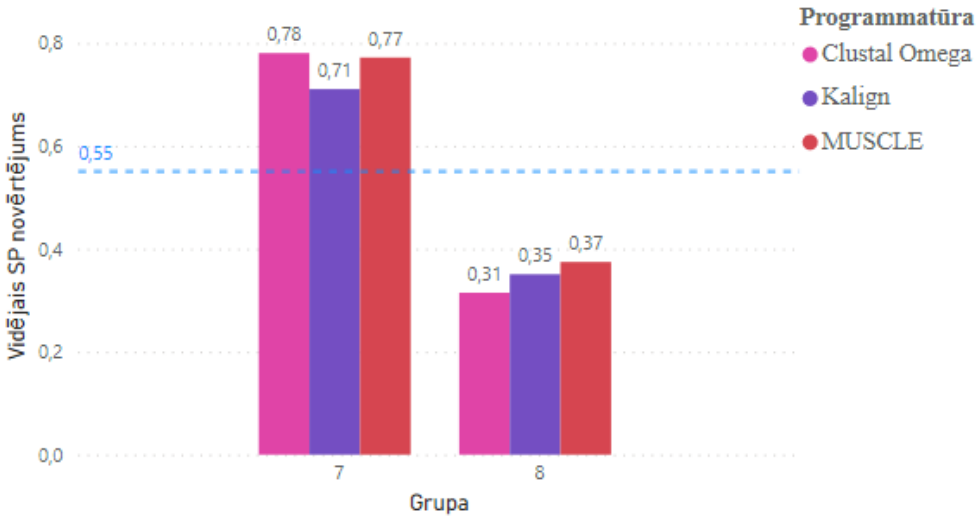


att. 5.9 Vidējais izpildes laiks 3., 4. un 5. grupai

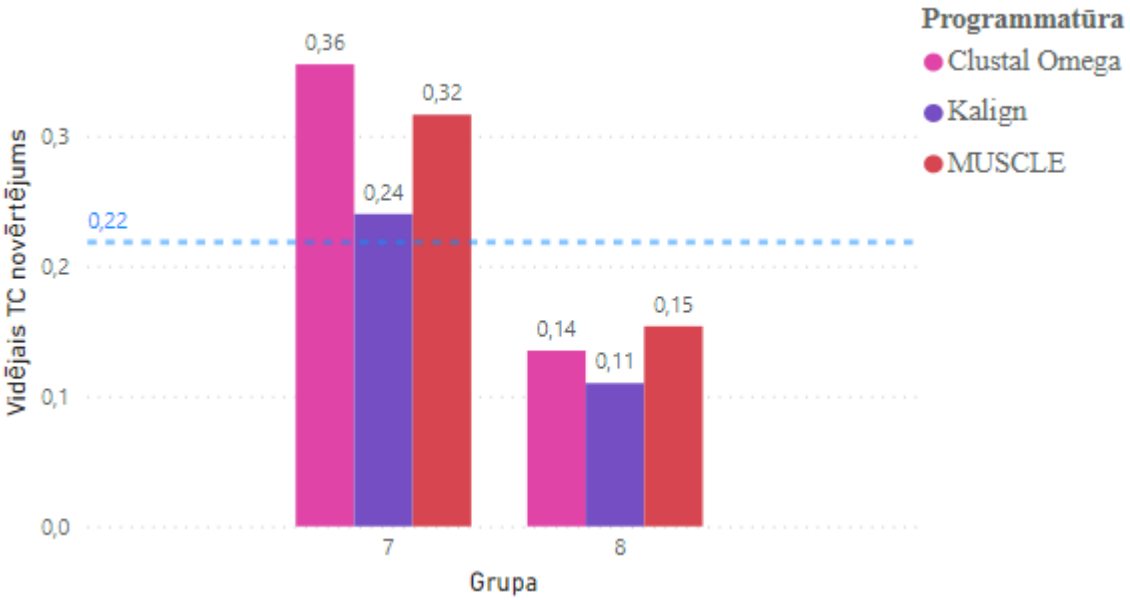
Tāpat kā ar kvalitatīvajiem rādītājiem arī izpildes laiks att. 5.9 uzrāda tādu pašu tendenci, kāda novērota iepriekšējos divos datu apskatos.

5.1.4. Septītā un astotā kopa

Septītā un astotā kopa jau ir no otrās BAliBASE versijas, izmēri: septītā – 8 datnes, astotā 10 datnes. Septītā kopa sastāv no transmembrānu olbaltumvielu kopām, apmēram no 400 olbaltumvielu sekvencēm. Astotā kopa satur olbaltumvielas, kurās domēnu sekojošā kārtība nav saglabāta.

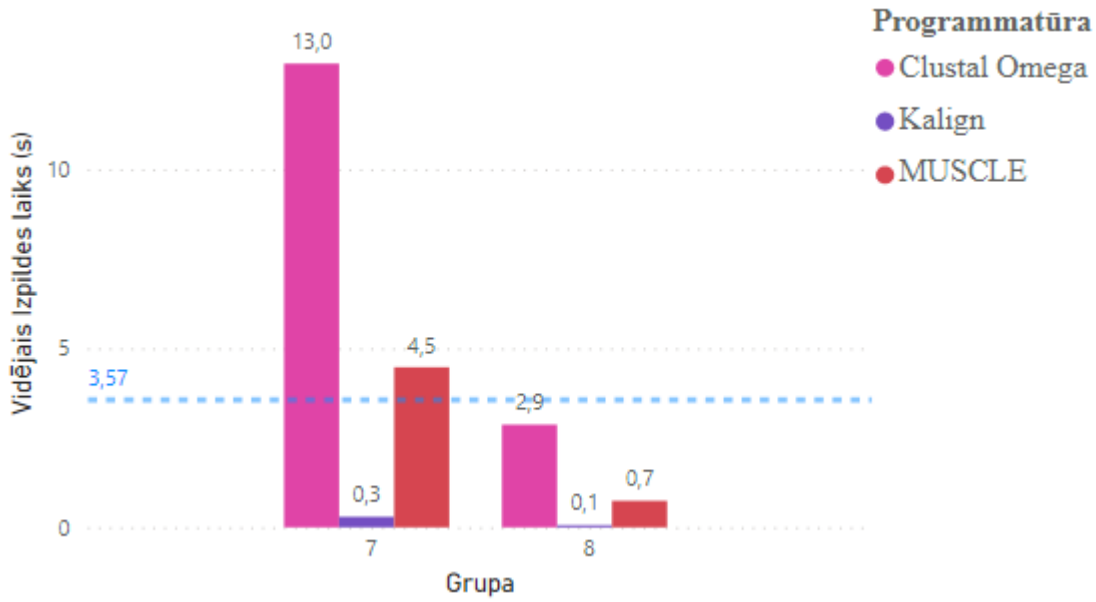


att. 5.10 SP novērtējums 7. un 8. kopai



att. 5.11 TC novērtējums 7. un 8. kopai

Šiem datiem, kas redzami att. 5.10 un att. 5.11 specifika ir tāda, ka trūkst datu par MAFFT programmu, jo šī programma atteicās apstrādāt šo kopu datus. Līdz ar to varam novērtēt to, ka 7. kopai vislabākie kvalitatīvie rādītāji ir Clustal Omega programmai, taču 8. kopai, kurai rezultāti ir salīdzinoši ļoti vāji kopumā, vislabākais rādītājs ir MUSCLE programmai.

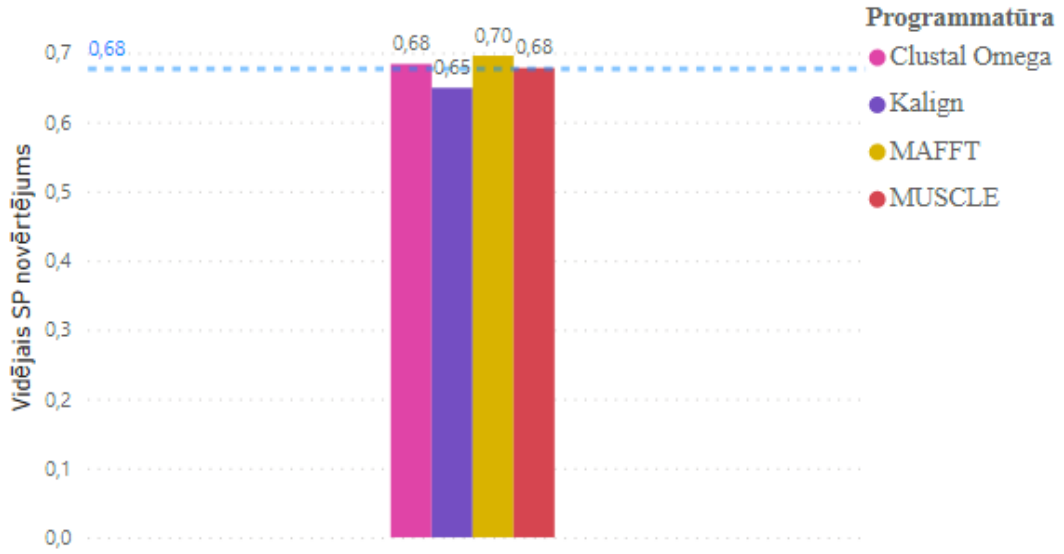


att. 5.12 Vidējais izpildes laiks 7. un 8. grupai

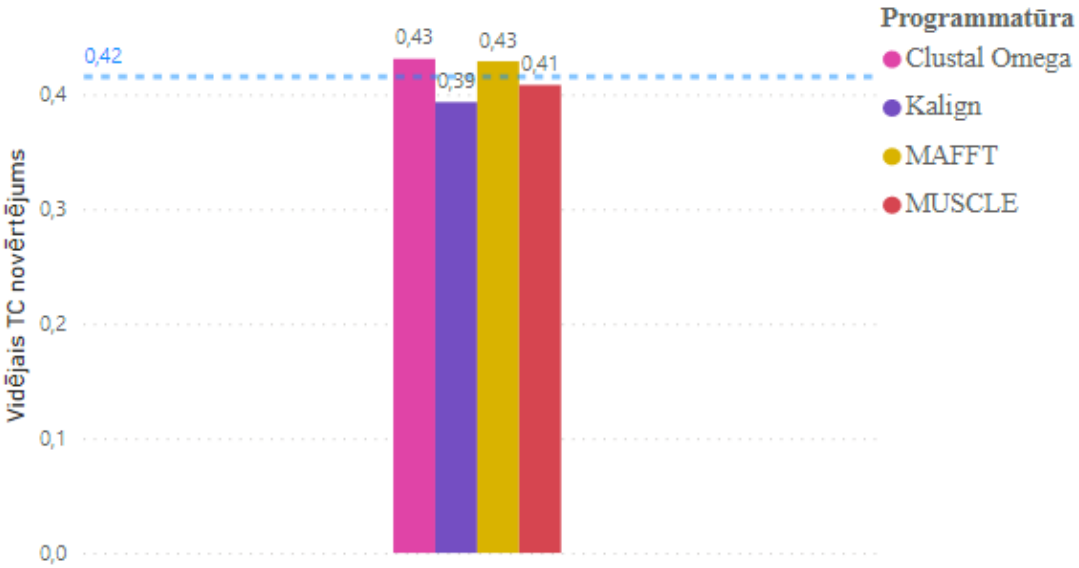
Vidējā izpildes laikā izmaiņu nav, tik cik tas, ka trūkst MAFFT programmas, kas iepriekšējos novērojumos ir vislēnākā.

5.1.5. Devītā kopa

Visa devītā kopa ir BAliBASE 3. versija, kura sastāv no 224 datnēm iekš četrām apakškopām, kurām kopā vēl ir deviņas apakšapakškopas, taču tās atsevišķi neizdalīsim. Tā satur olbaltumvielu kopas ar lineāriem motīviem. Lineāri motīvi iekļauj svarīgas funkcionālas vietas, kā olbaltumvielu iedarbības vietas, šūnu daļu mērķa signālus, pēctranslācijas modifikācijas vietas vai šķelšanās vietas. Šīs vietas bieži ir sastopamas nesakārtotos reģionos, kurus ir sarežģīti izlīdzināt ar klasiskām vairāku sekvenču izlīdzināšanas metodēm. Vairums lineāro motīvu ir 3 līdz 10 aminoskābju garumā, un vairums satur nezināmus atlikumus. [32]

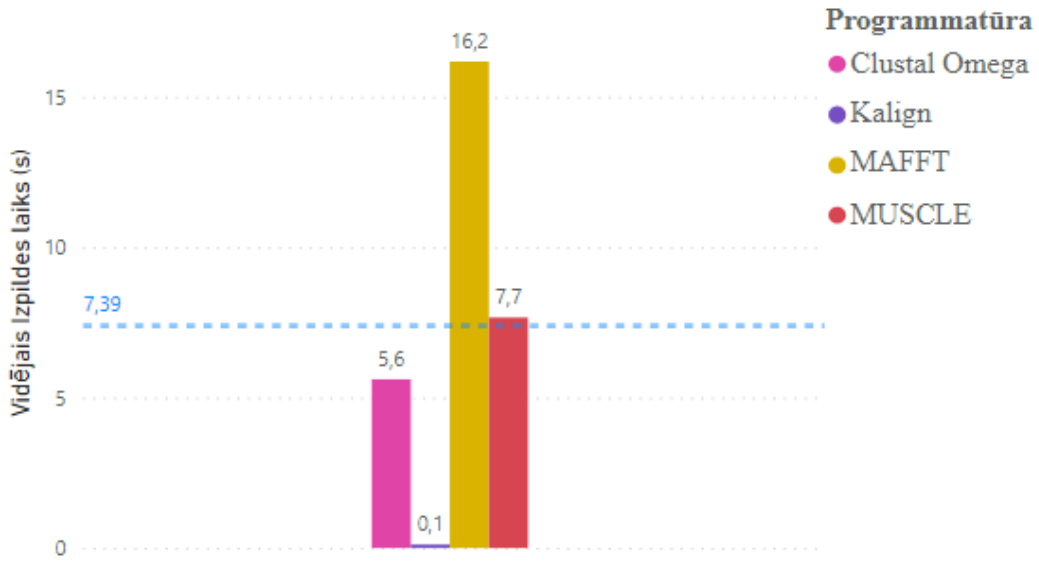


att. 5.13 SP novērtējums 9. grupai



att. 5.14 TC novērtējums 9. grupai

9. grupas datiem att. 5.13 un att. 5.14 redzami ļoti līdzīgi rezultāti starp visām programmām, principā pat varētu teikt, ka būtiskas starpības nav.

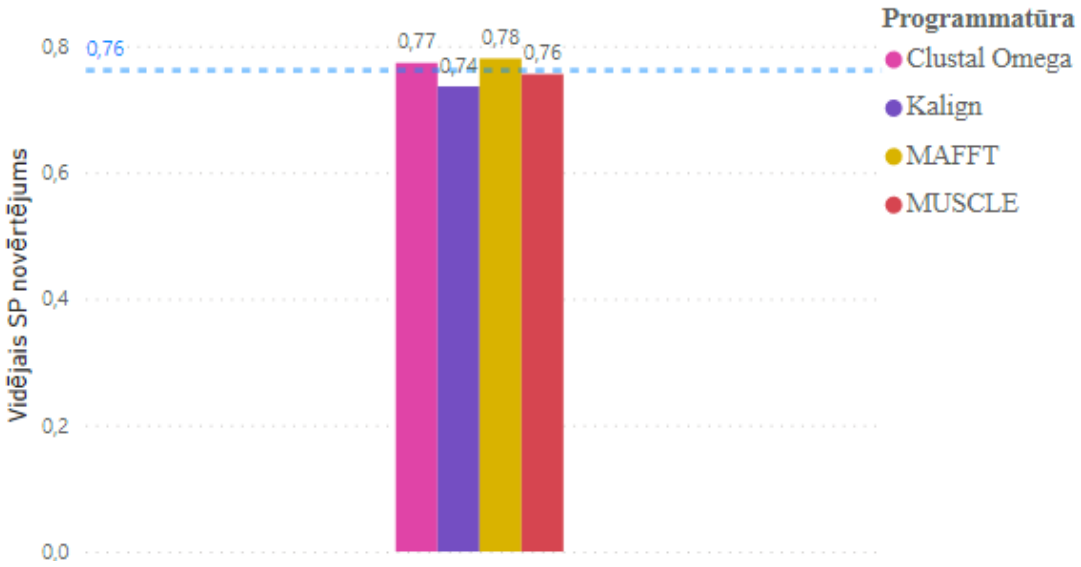


att. 5.15 Vidējais izpildes laiks 9. grupai

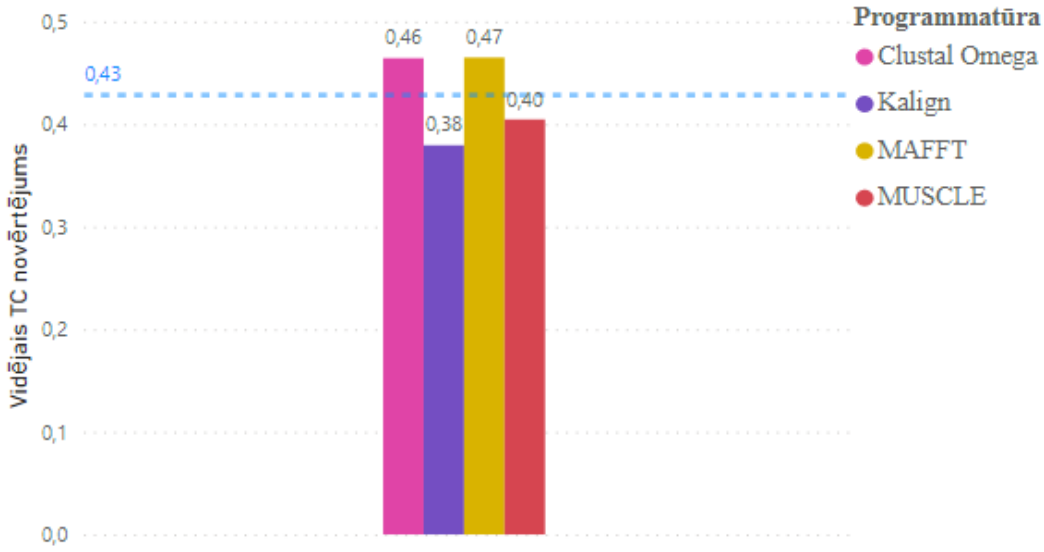
Skatoties datus att. 5.15 varam redzēt, ka tendence vidējā izpildes laikā ir tāda pati, kā iepriekšējās datu grupās, taču šoreiz, ņemot vērā to, ka kvalitātes novērtējumos starpības nav, tad šīs grupas novērtējums parāda to, ka šajos gadījumos varētu būtu optimāli izvēlēties ātrāko no programmām, kas šajā (un arī visos pārējos gadījumos) ir Kalign.

5.1.6. Desmitā kopa

Desmitā kopa ir ceturtās versijas papildinājums. Pievienotas 218 lielas, sarežģītas olbaltumvielu kopas, veidotas, lai atainotu šodienas sekvenču izpētes prasības. Kā arī balstās uz trīs premisēm. Pirmkārt, daudzas no eksistējošām vairāku sekvenču izlīdzināšanas novērtējuma sistēmām, un līdz ar to arī MSA programmas, pastiprināti pievērš uzmanību uz šabloniem, kas ir saglabāti lielākajā daļā sekvenču un nepietiekoši pievērš uzmanību retākiem šabloniem, kas varētu norādīt uz apakš sugu specifiku vai arī specifiku pēc konteksta. Otrkārt, esošās MSA programmas pārsvarā modelē lodveida domēna struktūru un evolūciju. Kaut gan daudzas olbaltumvielas, it īpaši eikariotos ir nestrukturētas vai satur lielus nestrukturētus reģionus. Šie reģioni bieži satur motīvus, piemēram, signalizējošās sekvences vai arī pēctranslācijas modifikācijas, kuras iesaistās šūnas regulējošās funkcijās. Visbeidzot, lielas caurlaidības sekvencēšanas tehnoloģijas ir radījušas milzīgus apjomus ar trokšņainiem datiem, iekļaujot fragmentāras vai citādi kļūdainas sekvences, kuras iespaido MSA programmu veiktspēju. [33]

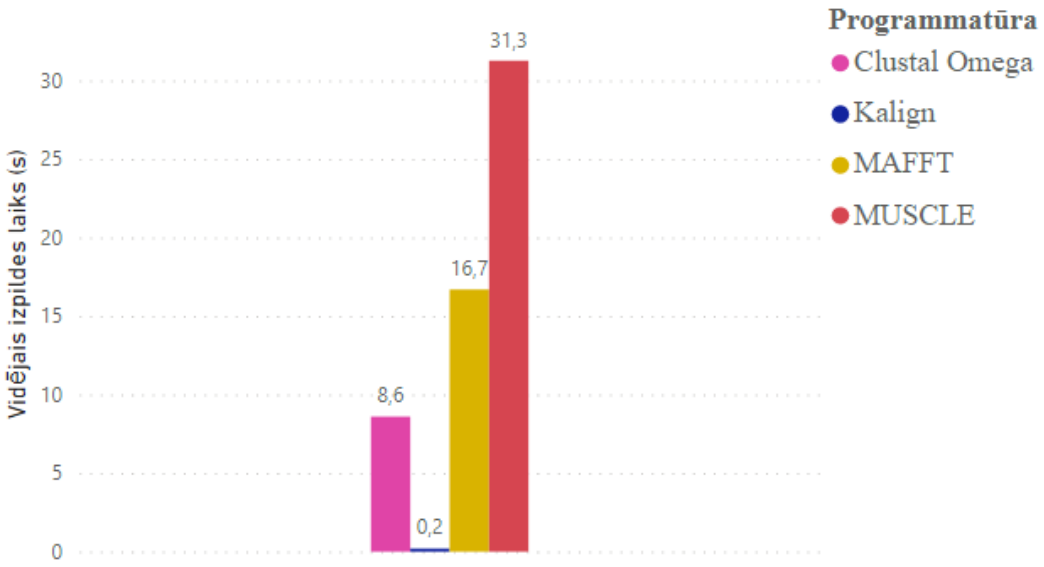


att. 5.16 SP novērtējums 10. grupai



att. 5.17 TC novērtējums 10. grupai

Rezultāti, kas redzami att. 5.16 un att. 5.17 no visiem iepriekš apskatītajiem ir vissvarīgākie, jo datu kopas, kas izmantotas, radītas visnesenāk, līdz ar to vairāk pielāgotas šodienas problēmu risināšanai un šodienas reālajām situācijām. Taču aina būtiski nav mainījusies, MAFFT programma ir ar visaugstāko novērtējumu, kurai ļoti cieši seko Clustal Omega programma.



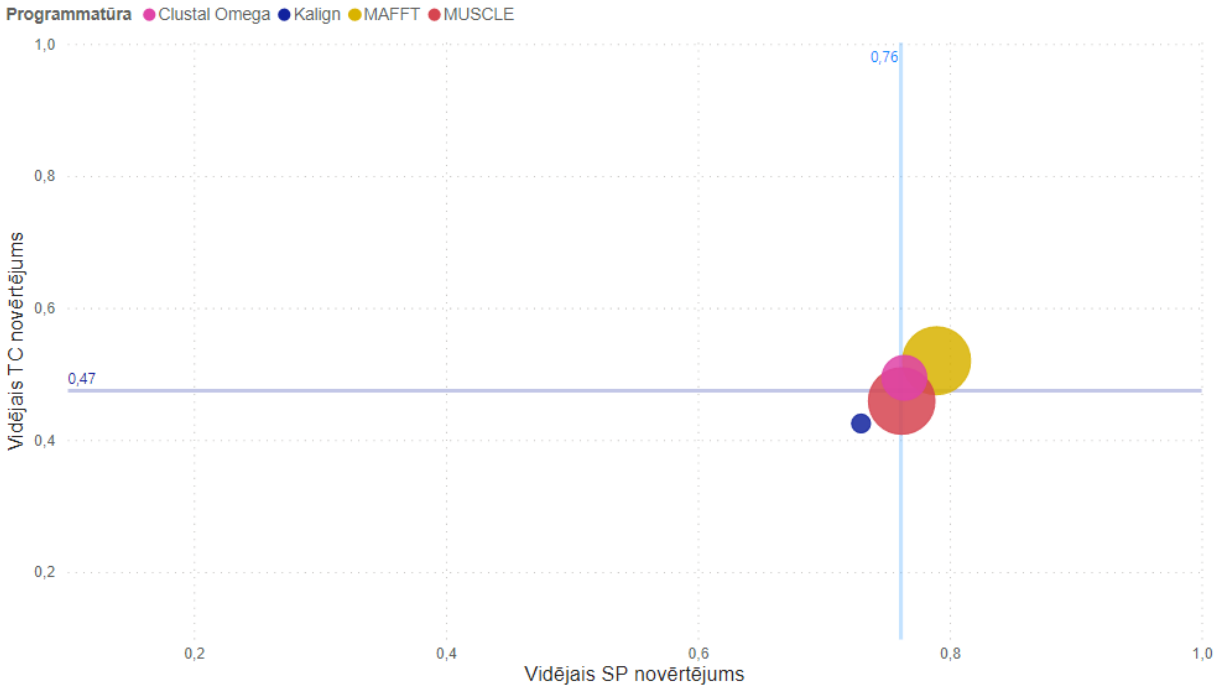
att. 5.18 Vidējais izpildes laiks 10. grupai

Apskatot att. 5.18 redzamos datus par izpildes vidējo izpildes laiku, redzams, ka neraksturīgi iepriekš redzētajam, daudz lēnāka nekā pārējās ir programma MUSCLE, precīzākā MAFFT programmatūra vēl aizvien ir ātrāka par to, taču jūtami lēnāka nekā pārējās, Kalign darbojas ļoti ātri, apstrādājot visas datnes praktiski momentāni. Taču, kam būtu jāpievērš uzmanība, ir Clustal Omega ātruma un kvalitāšu novērtējums kopumā.

5.1.7. Kopvērtējums

Iepriekš atsevišķi izdalītās grupas tiek apvienotas un izvērtēti rezultāti. Kopumā tika analizētas 859 datnes, kas tika sadalītas 9 grupās no BAliBASE datubāzes.

Sākotnēji tiek analizēti rezultāti, ignorējot grupas, bet apskatot vērtējuma parametrus par visām datnēm kopumā.



att. 5.19 Izvērtējums par visām BAliBASE datnēm

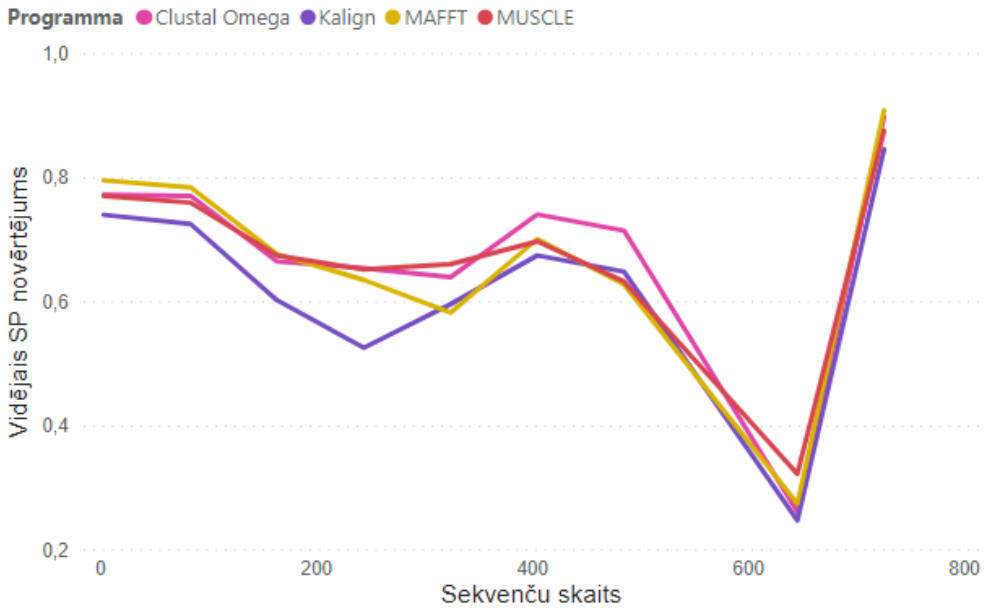
Skatoties att. 5.19, kur x ass ir vidējais SP novērtējums, y ass – vidējais TC novērtējums un burbuļa izmērs – vidējais izlīdzinājuma izpildes laiks, var novērot gan iepriekš redzamās tendences, gan arī atklājās varbūt iepriekš nepieminēta informācija. Kā jau iepriekš, apskatot rezultātus atsevišķi pa grupām, kvalitātes rādītājos augstākais novērtējums vienmēr bija programmai MAFFT, kas redzams arī šajā kopējā ainā, kā arī tas, ka vidējais izpildes laiks šai programmai ir lielāks, kā pārējām. Līdzīgi novērojams ir arī tas, ka programma Kalign izceļas starp citām ar visātrāko izpildes laiku. Taču, kas varbūt iepriekš, analizējot datus detalizēti nebija tik viegli novērojams – Clustal Omega, kuras kvalitātes rādītāji ir tuvu MAFFT uzrādītajiem, ir caurmērā gan ātrāka, gan kvalitatīvāka par programmu MUSCLE.

Šajā att. 5.19 gan zināma ietekme varētu būt pēdējās grupas lielajam skaitliskajam īpatsvaram attiecībā pret citām grupām, tādejādi savā ziņā neprecīzi atainojot proporcionālo veiktspēju dažāda tipa datos, taču jāņem vērā arī tas, ka pēdējās grupas dati, ir veidoti visnesenāk, līdz ar to ar pēc iespējas aktuālāku skatu uz risināmām problēmām

Vēl iespējams salīdzināt programmu veiktspēju atkarībā no sekvenču daudzuma, sekvenču garuma un simbolu skaita, kas tad ir nākamais analīzes objekts. Šie dati tika iegūti, no katras BAliBASE datnes izgūstot tajā atrodamo sekvenču skaitu un sekvenču vidējo garumu, salīdzinot ar SP un TC novērtējumiem, un šīs datnes izlīdzināšanas izpildes laiku. Lai dati nebūtu tik saraustīti

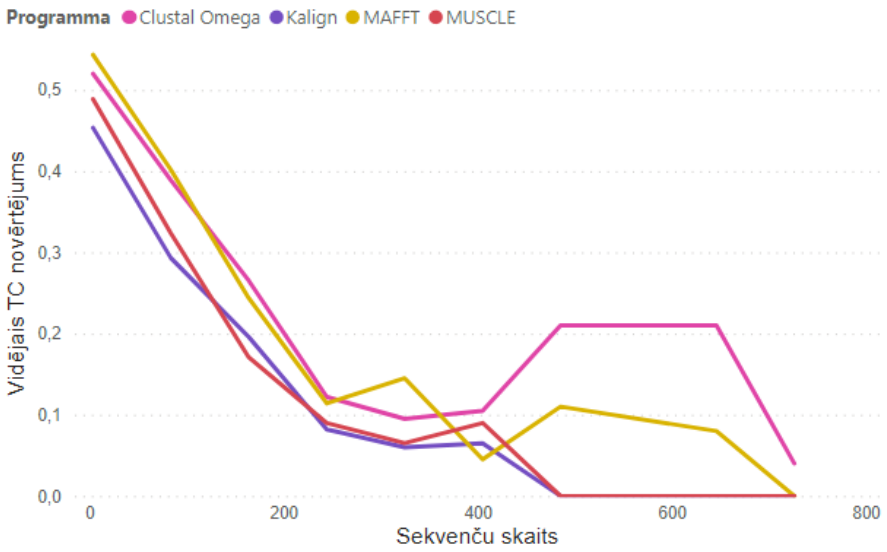
tie tiek apvienoti grupās, lai kopumā katrā grafikā nebūtu vairāk pat 10 grupām (piemēram, sekvenču garums 1-10, 11-20, u.t.t).

5.1.7.1. Sekvenču skaits



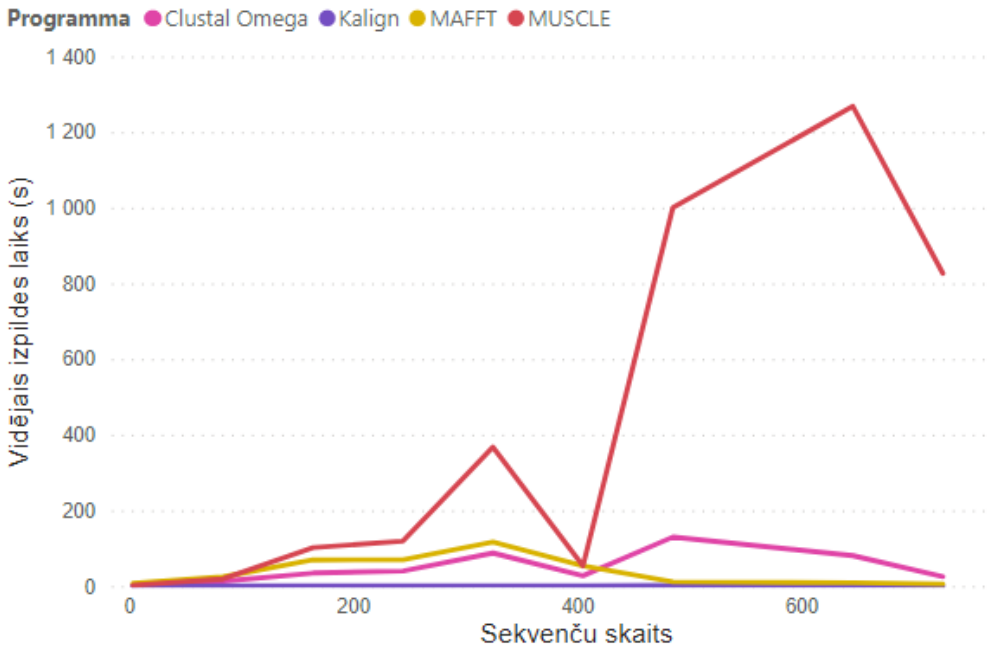
att. 5.20 Sekvenču skaita un SP novērtējuma grafiks

att. 5.20 redzamajā grafikā, kur uz X-ass redzams sekvenču skaits un uz Y-ass vidējais SP novērtējums grūti saskatīt kādu reālu likumsakarību, kas ļautu domāt, ka kādai no programmām ir priekšrocības vai tieši otrādi mīnusi atkarībā no sekvenču skaita, vērtējot pēc SP novērtējuma parametra.



att. 5.21 Sekvenču skaita un TC novērtējuma grafiks

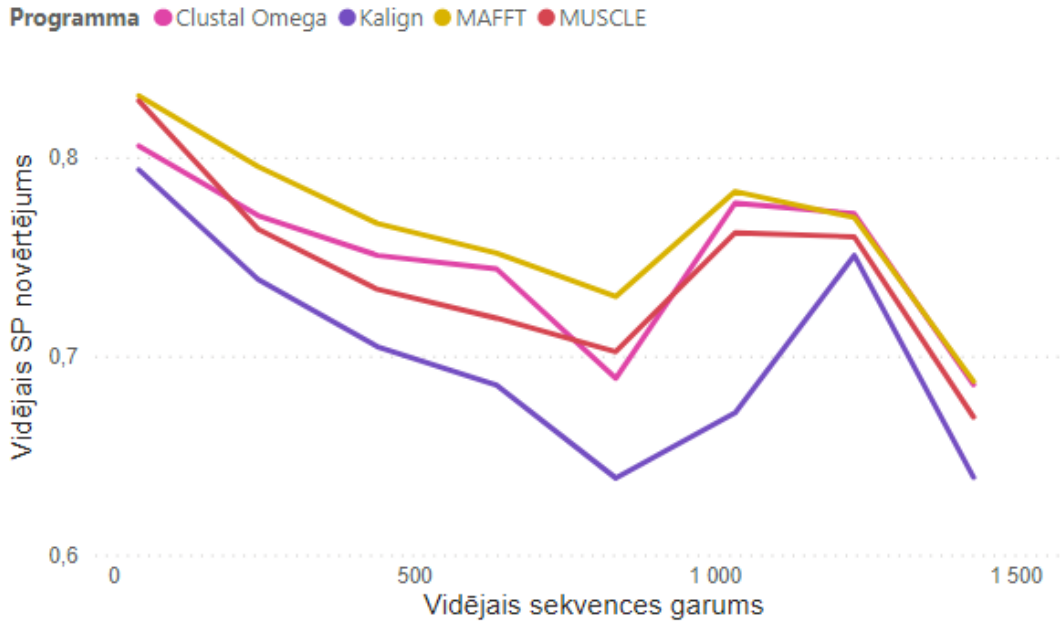
Pēc att. 5.21 redzamajiem datiem jau parādās aina, kurā visas programmu līknes nekustās simetriski, bet gan ir arī izlēcēji, kas atbalsta iepriekš jau novēroto situāciju. Iepriekš, analizējot 10. kopas datus (kuros ir sarežģītākās sekvences, lielāki datu apjomi, kā citās) Clustal Omega uzrādīja labākus rezultātus, kā iepriekš novērots. Arī šajā bildē redzams, ka palielinoties sekvenču skaitam TC novērtējuma starpība starp Clustal Omega un citām programmām ir ar krietni lielāku plaisu, kā pie zemāka sekvenču skaita.



att. 5.22 Sekvenču skaita un izpildes laika grafiks

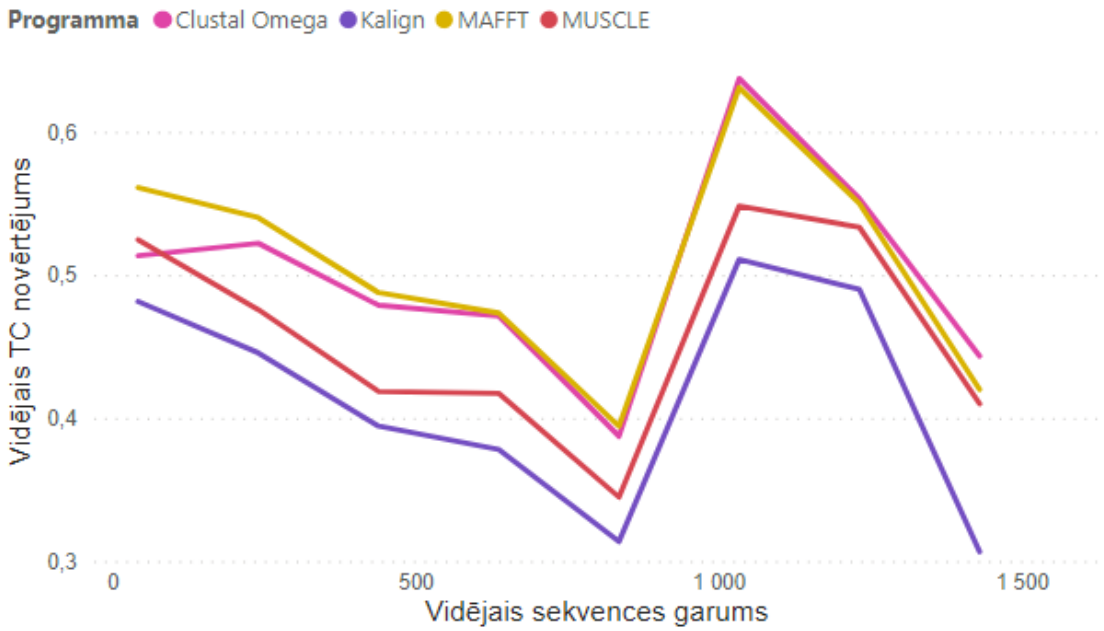
att. 5.22 redzamais grafiks diezgan nepārprotami liecina par MUSCLE programmas grūtībām, kas rodas palielinoties sekvenču skaitam – ļoti strauji pieaug vidējais izpildes laiks.

5.1.7.2. Vidējais sekvences garums



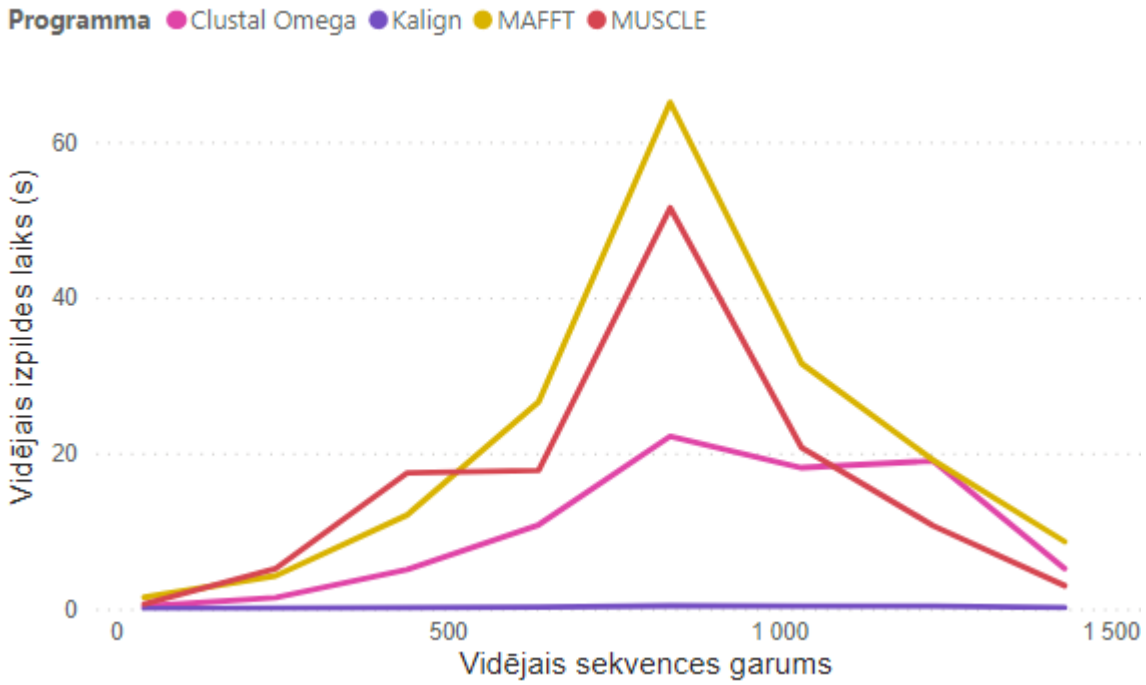
att. 5.23 Vidējā sekvences garuma un SP novērtējuma grafiks

Līdzīgu ainu kā iepriekš apskatītajā att. 5.20, tāpat arī att. 5.23 nav redzamas stingras likumsakarības starp vidējo sekvences garumu un SP novērtējumu.



att. 5.24 Vidējā sekvences garuma un TC novērtējuma grafiks

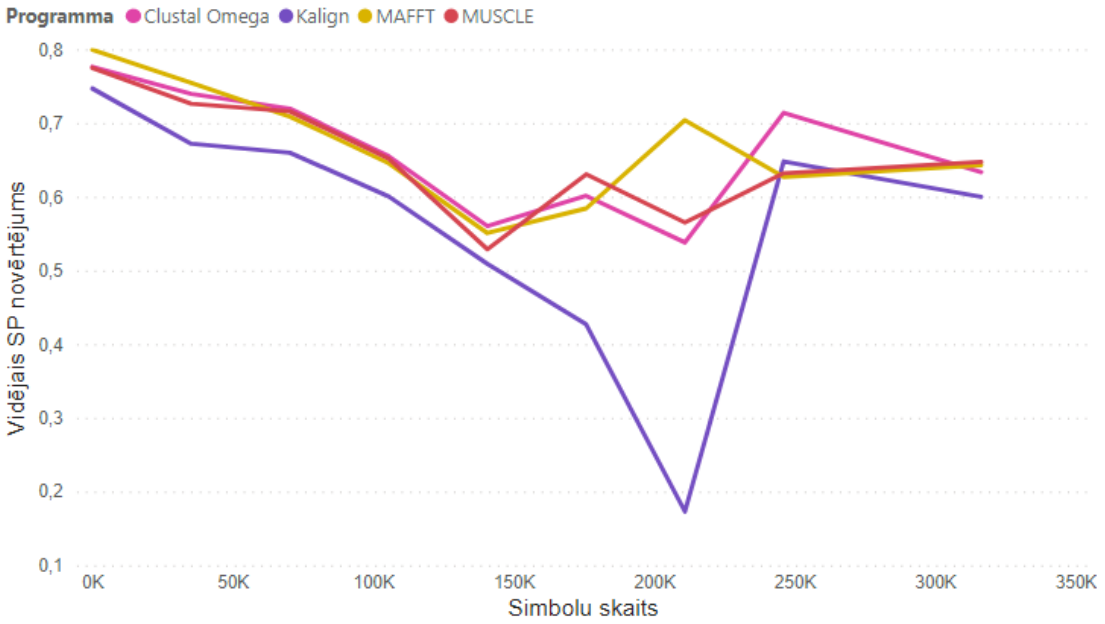
att. 5.24 arī neuzrāda acīmredzamas likumsakarības, lai varētu sākt domāt par vidējās sekvences garuma ietekmi pret TC novērtējumu šajā darbā apskatītajās programmās.



att. 5.25 Vidējā sekvences garuma un izpildes laika grafiks

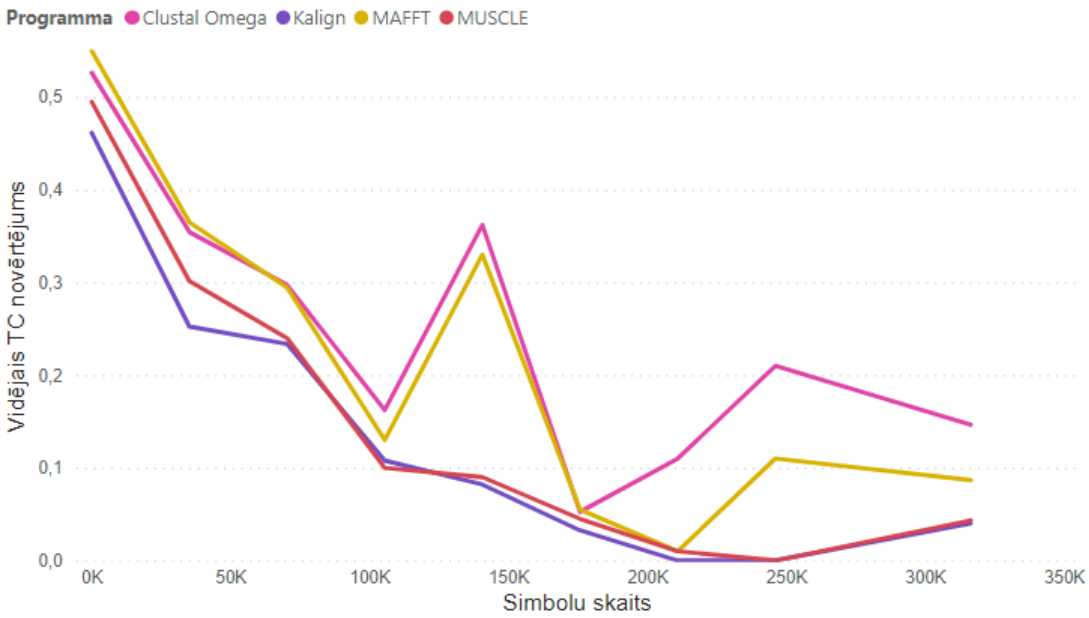
Ļoti interesantu ainu uzrāda att. 5.25, kurā izpildes ilgums ilgākais visām programmām ir sekvencēm apmēram 850 simbolu garumā. Kā arī interesanti, ka Clustal Omega piedzīvo ļoti lēzenu kāpumu pret sekvenču garumu, atšķirībā no MAFFT un MUSCLE, kur var novērot lielu starpību. Kā arī jau iepriekš novērots, tad programmas Kalign izpildes ātrums ir gandrīz absolūti neatkarīgs no ievades datiem (vismaz tik detalizēti, cik šajā darbā tiek apskatīts).

5.1.7.3. Simbolu skaits



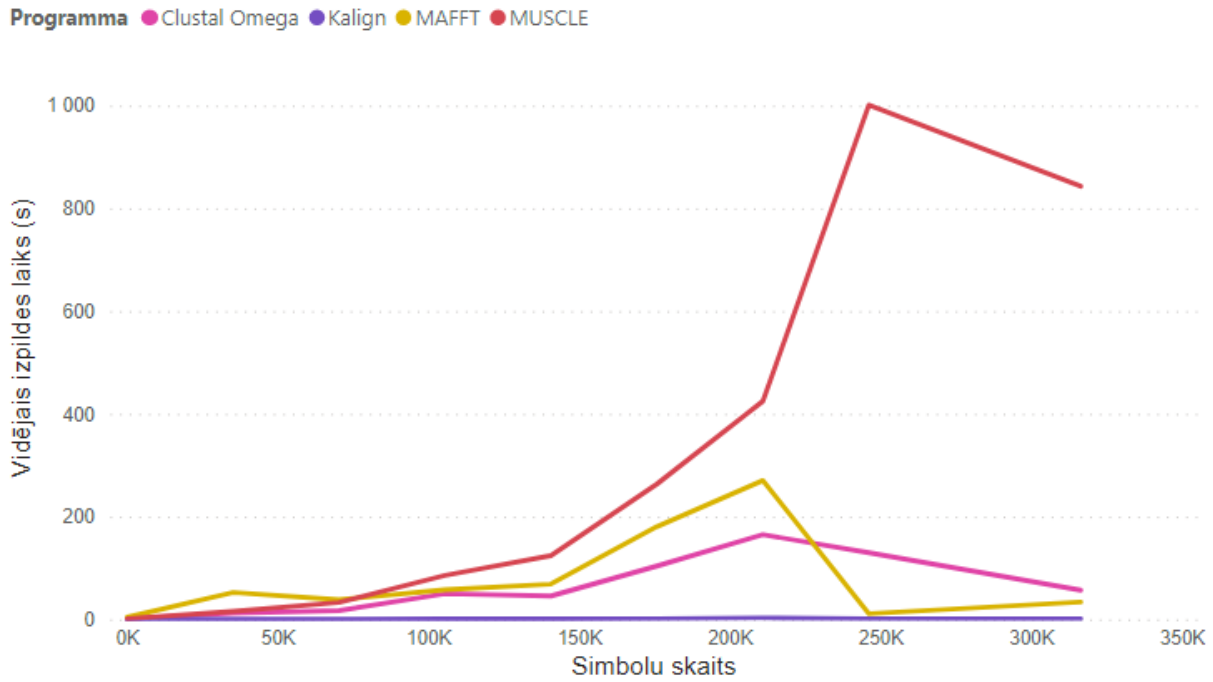
att. 5.26 Simbolu skaita un vidējā SP novērtējuma grafiks

Īpatnēja aina novērojama att. 5.26, jo Kalign kritums pie 180 - 230 tūkstoš simbolu skaita, ir inverss MAFFT un arī Clustal Omega kāpumam attiecībā pret citām sekvencēm. Skatoties datos, redzams, ka šajā amplitūdā ir sekvences, kuras bioloģiski raksturotas kā sekvences ar pozitīviem lineāriem motīviem.



att. 5.27 Simbolu skaita un vidējā TC novērtējuma grafiks

Arī šajā att. 5.27 ir redzams MAFFT un Clustal Omega kāpiens pie reģioniem 100 – 150 un 210 – 250 tūkstoši simboli. Ja 210 – 250 tūkstoš simbolu reģions tika apskatīts jau pie iepriekšējā grafika, tad 100 – 150 ir jaunums, kur atlasot datus, redzams, ka tās ir pārsvarā 10. grupas sekvences, kurās redzams liels apjoms un trokšņaini dati, reproducējot mūsdienu sekvencēšanas izaicinājumus.



att. 5.28 Simbolu skaita un vidējā izpildes laika grafiks

Šajā att. 5.28 uzskatāmi parādās programmas MUSCLE lielākā problēma – pie lielākiem datu apjomiem programmai strauji palielinās izpildes laiks, kā arī kā nākamajā nodaļā būs redzams pie reāliem datiem, ka programma nespēj apstrādāt lielus datu apjomus. Pārējās programmas komentējot, var pieminēt, ka īpatnēji izskatās MAFFT un Clustal Omega grafiki, kuru izpildes laiki, līdzīgi kā iepriekšējā nodaļa sarūk pie lielākiem apjomiem. Autoraprāt, tas, visticamāk, ir datu kopas īpatnības rezultātā.

5.2. Reāli dati

Lai pārbaudītu, kā darbojas algoritmi uz reālām un apjomīgām datu kopām, tika izgūti dati no NCBI datubāzes. Tika izvēlēti vīrusu RNS dati šādu iemeslu dēļ: Pirmkārt, iepriekš apskatītie salīdzinājumi norisinājās uz olbaltumvielu sekvencēm, līdz ar to, lai tiktu apskatīts pēc iespējas plašāks darbības loks, tad tika izvēlētas sekvences ar nukleotīdu datiem. Otrkārt, pirms šī darba izstrādes, darba ideja radās, strādājot pie vīrusu RNS sekvenču izlīdzināšanas praimeru izstrādei,

tādejādi, salīdzinot algoritmu darbību uz šīm datu kopām, tikt pie praktiski izmantojamiem rezultātiem.

Zemāk 5.1. tabulā redzams izvēlēto reālo datu raksturojums, kas tas ir par organismu (ja nav īpaši minēts, tad tā saimniekorganisms ir cilvēks), no kura pasaules reģiona, cik liels ir sekvenču skaits un kāds, ir vidējais sekvences garums, kā arī analizējamās datnes izmērs.

5.1 tabula

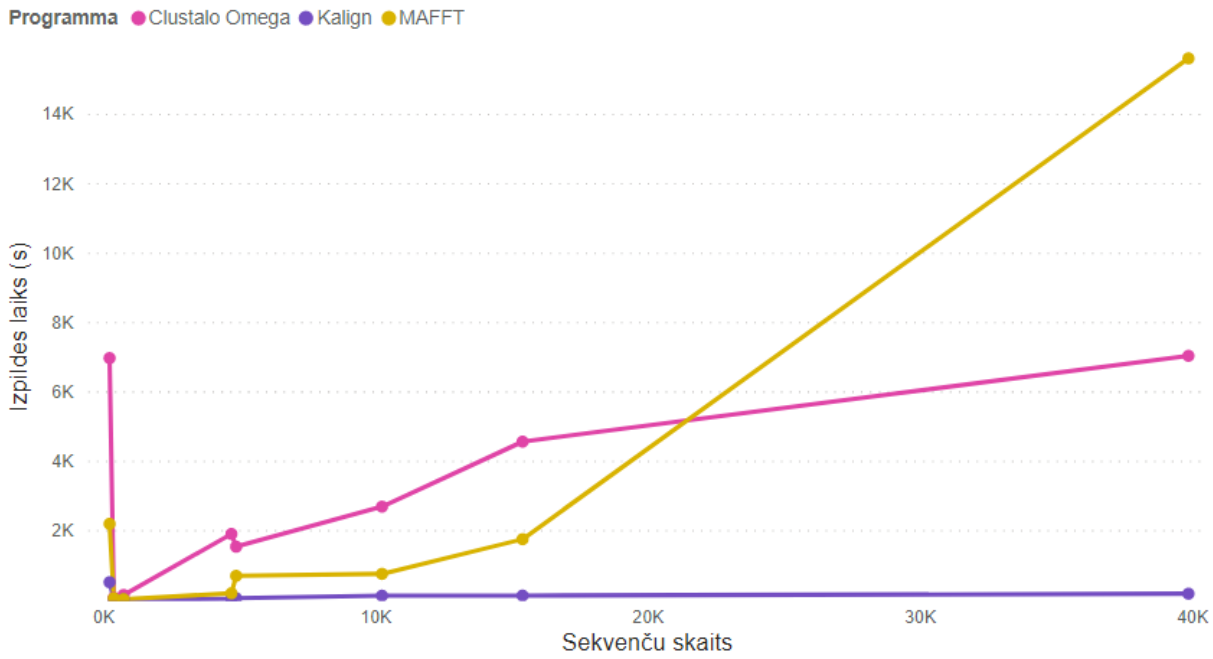
Izmantoto datu īpašības

Sekvenču informācija	Sekvenču skaits	Vidējais sekvences garums	Datnes izmērs
HIV vīrusa RNS sekvences, Eiropā	399	516	238 KB
HIV vīrusa RNS sekvences, pasaulē	758	707	595 KB
SARS-CoV-2 vīrusa RNS sekvences, Ķīnā	245	15780	3873 KB
C Hepatīta vīrusa sekvences, Lielbritānijā	4894	836	4459 KB
HIV vīrusa RNS (visos saimniekorganismos), pasaulē	4721	951	4903 KB
Gripas RNS sekvences, Dienvidamerikā	10252	1561	17069 KB
Gripas sekvences, Austrālijā	15416	1646	26947 KB
C Hepatīta vīrusa sekvences, Eiropā	39877	706	31400 KB

[35]

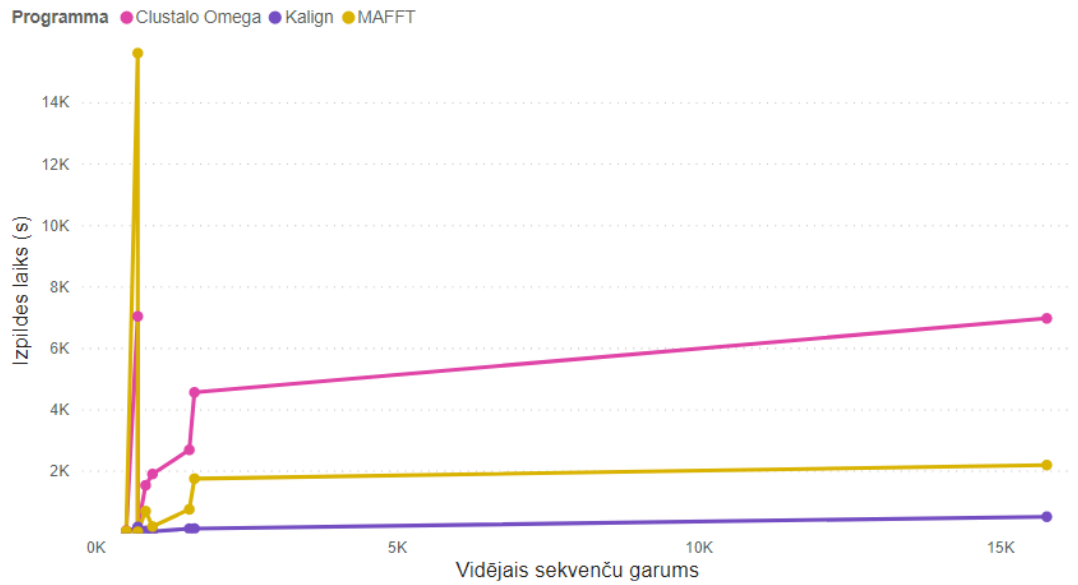
Diemžēl uz reālām datu kopām nav iespējams veikt SP vai TC novērtējumu, jo to iegūšanai ir nepieciešams references izlīdzinājums, pēc kura tiek novērtēts jaunais izlīdzinājums, līdz ar to reālās datu kopās vērtēsim tikai izpildes laiku atkarībā no tā, cik daudz sekvenču, cik garas ir sekvences un cik daudz simbolu sekvenču datnē.

Kā arī tālāk esošajos grafikos tiks apskatītas tikai 3 programmas – MAFFT, Clustal Omega, Kalign, tādēļ, ka, MUSCLE nav paredzēta nukleotīdu sekvenču izlīdzināšanai (ir opcijas to veikt, bet tās netiek atbalstītas un izpildes laikā rodas kļūdas).



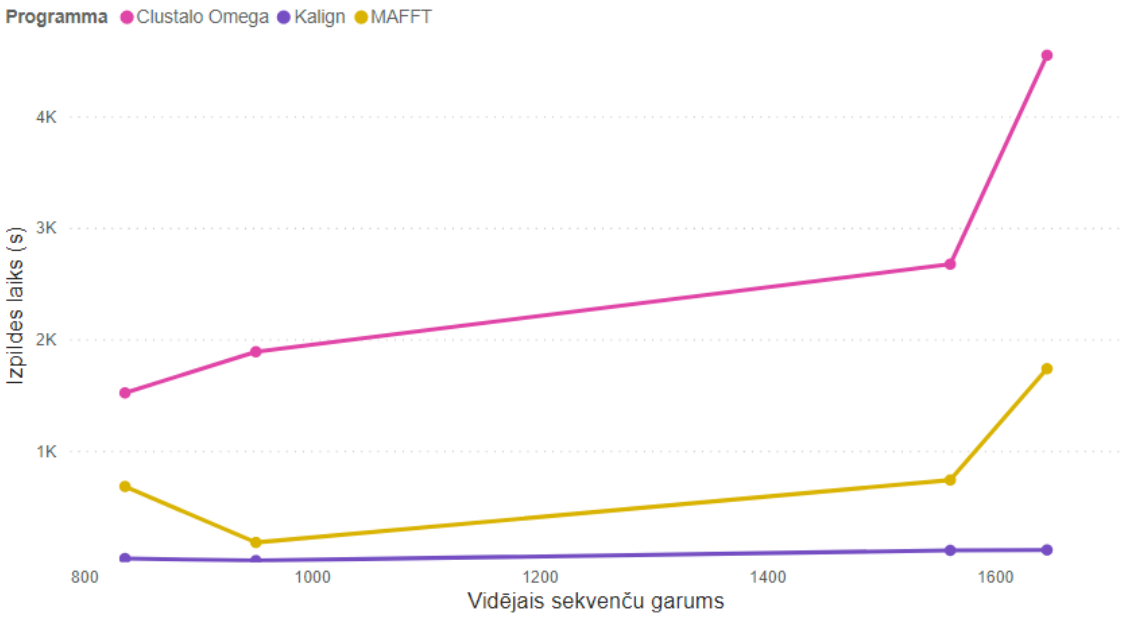
att. 5.29 Sekvenču skaita un izpildes laika grafiks

Grafikā, kas redzams att. 5.29 ļoti uzskatāmi redzama MAFFT un Clustal Omega izpildes laika palielināšanās sekvenču skaitam palielinoties. Taču jau sākot ar šo grafiku varam novērot atšķirību starp rezultātiem, ko iegūstam no reāliem datiem un iepriekš iegūtajiem BAliBASE kopu datiem. Clustal Omega izpildes laiks lēnāks nekā MAFFT (izņemot pēdējo, apjomīgāko datni), kaut gan iepriekš iegūtie rezultāti rādīja tieši pretējo. Tālākos pētījumos būtu interesanti izpētīt, vai tas ir saistīts ar to, ka šajā nodaļā tiek izmantotas nukleotīdu sekvences, bet iepriekš – olbaltumvielu sekvences.



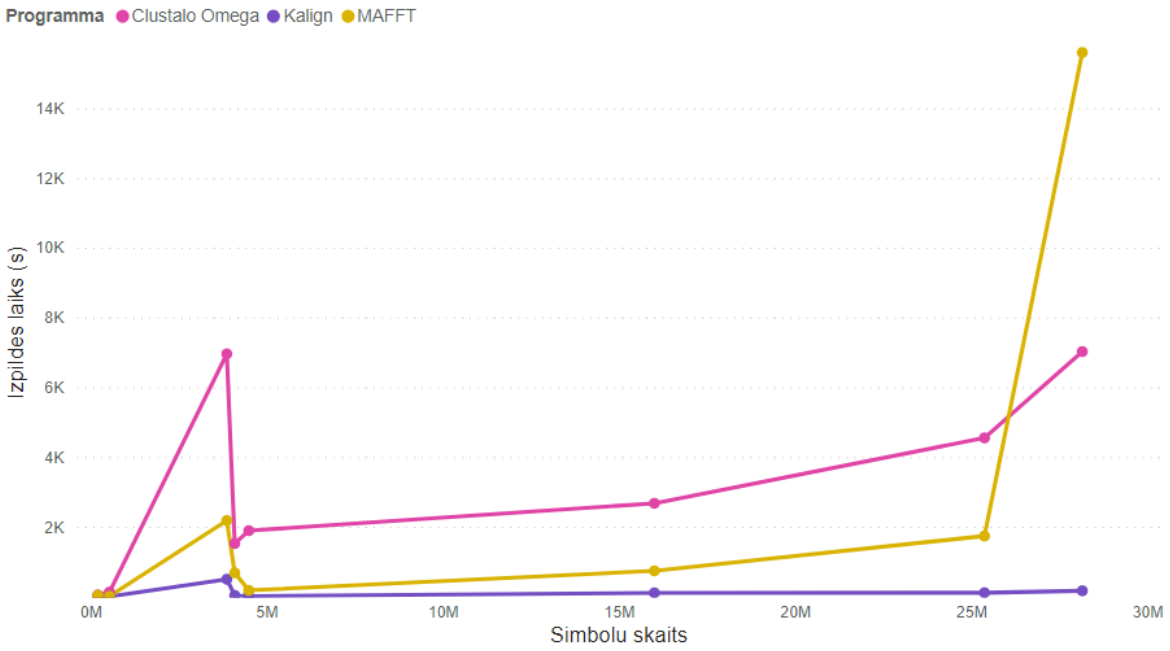
att. 5.30 Vidējā sekvenču garuma un izpildes laika grafiks

att. 5.30 redzamais grafiks ir izdevies diezgan neveiksmīgs, jo tam sanāk diezgan neproporcionāli dati, starp sekvenču garumu 3000 un 15000 ir milzīga sprauga bez datiem, līdz ar to vizuālais attēls uzrāda neadekvātu bildi, kā arī pirmais pīķis rodas no tā, ka ir 40 tūkstošu sekvenču liels fails, kurā sekvenču vidējais garums ir ~700 simbolu. Lai padarītu bildi saprotamāku tiks nofiltrēti izlēcēji.



att. 5.31 Vidējā sekvenču garuma un izpildes laika grafiks (bez izlēcējiem)

att. 5.31 jau ir saprotamāks un ir redzama sakarība, starp vidējo sekvenču garumu un izpildes laiku.



att. 5.32 Simbolu skaita un izpildes laika grafiks

att. 5.32 redzamā aina apstiprina att. 5.29 novēroto, ka kopumā uz šiem datiem Clustal Omega ir bijis lēnāks nekā MAFFT, izņemot lielāko no datnēm.

REZULTĀTI

Šī darba ietvaros tika salīdzinātas četras programmas, kuras veic vairāku sekvenču izlīdzināšanu. Šajā nodaļā tiks apkopoti iegūtie rezultāti un salīdzināti ar šo programmu autoru veiktajiem salīdzinājumiem, kā arī veikts atskats uz darba mērķu un uzdevumu izpildi.

Vērtējot programmas pēc to izlīdzinājumu kvalitātes rādītājiem – SP un TC novērtējumiem, stabili augstākie rezultāti tika sasniegti ar MAFFT programmu, kas sakrīt arī ar citu pētījumu rezultātiem. Pašu MAFFT autoru publikācijā, kurā tiek prezentēta šī programma, gan nav salīdzinājums ne ar vienu no šajā darbā apskatītajām programmām, jo tas publicēts 2002. gadā. [28] Līdzvērtīgi rezultāti redzami arī Clustal Omega publicētajā rakstā, ja runa ir par BAliBASE datu kopu, MAFFT rezultāti no šīm 4 programmām arī šajā publikācijā ir visaugstākie (šajā 2011. gada publikācijā gan nav iekļautas jaunākās datu kopas, kas iekļautas pēc tam). Interesanti, ka arī *Prefab* referenču kopai, kas, minēta šajā pašā publikācijā sakrīt novērtējums, taču veicot novērtējumu pret *HomFam* referenču kopu MAFFT rezultāti ir zemāki gan par Clustal Omega, gan Kalign (salīdzināts gan ar novecojušu Kalign versiju) rezultātiem (taču augstāki par MUSCLE). [5]

Viszemākos kvalitātes vērtējumus uzrādīja Kalign programma, kas gan sakrīt, gan nesakrīt ar šīs programmas autoru publicēto pētījumu 2019. gadā. Autori uzrādījuši rezultātus kopā caurmērā pret vairākām referenču kopām, izvērtējot pēc SP novērtējuma, kurās kopumā bija līdzīgs, kādu uzrādīja Clustal Omega un MUSCLE (šajā pētījumā MAFFT programma netika iekļauta). Autori arī savā publikācijā min, ka tieši BAliBASE referenču kopai Kalign ir uzrādījusi sliktākus rezultātus. [26]

Ja tiek apskatīts programmu novērtējums pēc izpildes laika, tad visātrākā programma ir Kalign, kas atbilst arī pašu autoru uzstādījumam. Šo salīdzināt ar citiem pētījumiem ir grūti, jo šī jaunākā versija ir iznākusi 2019. gadā, līdz ar to pārējo publikācijās nav bijusi iekļauta. Par pārējām 3 programmām šajā darbā nav viennozīmīgu secinājumu – vislielākās problēmas, izpildot programmas, bija ar MUSCLE, kuras izpildes laiki bija nestabili, pret referenču kopu lēnākā bija programma MAFFT, bet pret reālajiem datiem – Clustal Omega, līdz ar to šis ir jautājums, kurš, visticamāk, būtu jāpēta tālāk ar lielāku datu variāciju, un stabilākiem skaitļošanas apstākļiem. Clustal Omega publicētajā pētījumā starp šīm 3 programmām kā vislēnākā ir uzrādīta – MUSCLE, tad Clustal Omega un kā ātrākā MAFFT. [5]

Pēc šajā darbā iegūtajiem rezultātiem, autoraprāt, Kalign ir efektīva programma, kā iegūt pirmreizējus rezultātus, lai intuitīvi varētu izvērtēt sekvences un, iespējams, no šiem rezultātiem veikt lēmumus par tālāko plānu, ja nepieciešami pēc iespējas precīzāki rezultāti, tad izvēle būtu par labu programmai MAFFT.

Veicot programmu salīdzinājumu, tika iegūti 3 veidu dati, kvalitātes 4 – SP un TC novērtējumu rezultāti par katru datni, kā arī izpildes laiks katrai datnei. Šie rezultāti tika apkopoti, un attēloti grafikos, lai varētu uzskatāmi uzrādīt šī darba rezultātus.

SECINĀJUMI

Veicot literatūras izpēti par darba tēmu, tika iegūts papildu apstiprinājums bioloģisko datu analīzes problēmas aktualitātei, kur daļa no risinājuma ir tieši efektīvas vairāku sekvenču izlīdzināšanas metodes.

Šī darba ietvaros tika apskatītas četras industrijā plaši izmantotas vairāku sekvenču izlīdzināšanas programmas, par kurām darbā veikts izklāsts gan par pašām programmām, gan galvenajiem algoritmiem, kas šajās programmās ir kopīgi vai tieši otrādi atšķiras.

Lai veiktu programmu salīdzinājumu, autors izpētīja literatūru un citu iepriekš paveikto, lai izvēlētos datu kopas un parametrus, pēc kuriem tika veikta programmu salīdzināšana. Programmas tika salīdzinātas gan pret simulētām datu kopām, kurām pieejami arī references izlīdzinājumi, gan arī pēc programmu izpildes laikiem gan šim pašām simulētajām kopām, gan reāliem datiem.

Darba mērķis, autoraprāt, ir izpildīts, jo par programmām ir iegūta gan iepriekšējo pētījumu apstiprinoša informācija, gan jauni rezultāti, kuri iepriekš, veicot literatūras izpēti, netika atrasti.

Tālāk šo virzienu varētu pētīt, gan papildinot programmu klāstu ar kādu no industrijas profesionālajām maksas programmām, gan paplašinot references datu apjomu. Doktorantūrā iespējams turpināt darba tēmu, izstrādājot jaunu vairāku sekvenču izlīdzināšanas programmu.

IZMANTOTĀ LITERATŪRA UN AVOTI

[1] K. Steward, «Amino Acids – the Building Blocks of Proteins,» Technology Networks, 2019. [Tiešsaiste]. Pieejams: https://www.technologynetworks.com/applied-sciences/articles/essential-amino-acids-chart-abbreviations-and-structure-324357.

[2] J. Han, M. Kamber un J. Pei, «Data Mining,» *Data Mining Trends and Research Frontiers*, 2012, pp. 585-631.

[3] J. Thompson, Statistics for Bioinformatics: Methods for Multiple Sequence Alignment, Saint Louis: Elsevier, 2016.

[4] «DNA Sequence Alignment,» [Tiešsaiste]. Pieejams: http://www.sequence-alignment.com/.

[5] F. Sievers, A. Wilm, D. Dineen, T. J. Gibson, K. Karplus, W. Li, R. Lopez, H. McWilliam, M. Remmert, J. Söding, J. D. Thompson un D. G. Higgins, «Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega,» Mol Syst Biol, 2011.

[6] «Tēzaurs,» [Tiešsaiste]. Pieejams: https://tezaurs.lv/.

[7] K. Liu, R. C. Linder un T. Warnow, «Multiple sequence alignment: a major challenge to large-scale phylogenetics,» 2011.

[8] Y. Kang, Z. Deng, R. Zang un W. Long, «DNA barcoding analysis and phylogenetic relationships of tree species in tropical cloud forests,» *nature research,* 2017.

[9] Q. Le, F. Sievers un D. G. Higgins, «Protein multiple sequence alignment benchmarking through secondary structure prediction,» *Bioinformatics,* sēj. 33, nr. 9, pp. 1331-1337, 2017.

[10] J. Ma, «Protein Structure Prediction by Protein Alignments,» Chicago, 2015.

[11] E. van Pelt-Verkuil, A. van Belkum un J. P. Hays, «PCR Primers,» *Principles and Technical Aspects of PCR Amplification*, Dordrecht, Springer, 2008, pp. 63-90.

[12] «Polymerase chain reaction (PCR),» [Tiešsaiste]. Pieejams: https://www.khanacademy.org/science/ap-biology/gene-expression-and-regulation/biotechnology/a/polymerase-chain-reaction-pcr.

[13] H. Lin un W. Hsu, «GSAlign: an efficient sequence alignment tool for intra-species genomes.,» *BMC Genomics,* sēj. 21, nr. 182, 2020.

[14] A. E. Darling, I. Miklós un M. A. Ragan, «Dynamics of Genome Rearrangement in Bacterial Populations,» *PLOS Genetics,* 2008.

[15] D. Wei un Q. Jiang, «A DNA sequence distance measure approach for phylogenetic tree construction.,» *IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications,* 2010.

[16] S. F. Altschul, «Substitution Matrices,» *ENCYCLOPEDIA OF LIFE SCIENCES*, 2008.

[17] S. R. Eddy, «Where did the BLOSUM62 alignment score matrix come from?,» *NATURE BIOTECHNOLOGY,* sēj. 22, nr. 8, pp. 1035-1036, 2004.

[18] G. Guidi, «Distributed Many-to-Many Protein Sequence Alignment using Sparse Matrices,» [Tiešsaiste]. Pieejams: https://www.researchgate.net/figure/The-BLOSUM62-scoring-matrix-for-proteins_fig3_344436929.

[19] «Wikipedia. Neighbour Joining.,» 2014. [Tiešsaiste]. Pieejams: https://en.wikipedia.org/wiki/Neighbor_joining#/media/File:Neighbor_joining_7_taxa_start_to_finish_diagram.svg.

[20] M. Simonsen, T. Mailund un P. C.N.S., «Rapid Neighbour-Joining,» *Algorithms in Bioinformatics,* Berlin, Springer, 2008.

[21] «Sequentix. UPGMA,» 2000. [Tiešsaiste]. Pieejams: https://www.sequentix.de/gelquest/help/upgma_method.htm.

[22] E. Douzery, «Wikipedia. UPMGA,» [Tiešsaiste]. Pieejams: https://en.wikipedia.org/wiki/UPGMA#/media/File:UPGMA_Dendrogram_5S_data.svg .

[23] S. R. Eddy, «What is a hidden Markov model?,» *Nature,* p. 1315–1316, 2004.

[24] B.-J. Yoon, «Hidden Markov Models and their Applications in Biological Sequence Analysis,» *Current Genomics,* pp. 402-415, 2009.

[25] V. De Fonzo, F. Aluffi-Pentini un V. Paris, «Hidden Markov Models in Bioinformatics,» *Current Bioinformatics,* pp. 49-61, 2007.

[26] T. Lassmann, «Kalign 3: multiple sequence alignment of large datasets,» *Bioinformatics,* pp. 1928-1929, 2020.

[27] J. Daugelaite, A. O'Driscoll un R. D. Sleator, «An Overview of Multiple Sequence Alignments and Cloud Computing in Bioinformatics,» *International Scholarly Research Notices,* sēj. 2013, p. 14, 2013.

[28] K. Katoh, K. Misawa, K. Kuma un T. Miyata, «MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform,» *Nucleic Acids Research,* sēj. 30, nr. 14, 2002.

[29] R. C. Edgar, «MUSCLE: multiple sequence alginment with high accuracy and high throughput,» *Nucleic Acids Research,* pp. 1792-1797, 2004.

[30] J. D. Thompson, P. Frederic un P. Oliver, «BAliBASE: a benchmark alignment database forthe evaluation of multiple alignment programs,» *Bioinformatics,* pp. 87-88, 1999.

[31] A. Bahr, J. D. Thompson, J.-C. Thierry un O. Poch, «BAliBASE (Benchmark Alignment dataBASE): enhancements for repeats, transmembrane sequences and circular permutations,» *Nucleic Acids Research,* pp. 323-326, 2001.

[32] J. D. Thompson, P. Koehl, R. Ripp un O. Poch, «BAliBASE 3.0: Latest Developments of the MultipleSequence Alignment Benchmark,» *PROTEINS: Structure, Function, and Bioinformatics,* pp. 127-136, 2005.

[33] J. Thompson, «BAliBASE Reference Set 10 : benchmark alignments containing sequences with subfamily specific features, motifs in disordered regions and fragmentary/erroneous sequences,» [Tiešsaiste]. Pieejams: http://www.lbgi.fr/balibase/BAliBASE_R10/. [Piekļūts 2020].

[34] A. Mishra, B. K. Tripathi un S. S. Soam, «A Quality Assessment of Protein Multiple Sequence,» *2020 International Conference on Contemporary Computing and Applications*, Lucknow, 2020.

[35] «National Center for Biotechnology Information,» [Tiešsaiste]. Pieejams: https://www.ncbi.nlm.nih.gov/.

PIELIKUMI

1. PIELIKUMS

Kods sekvenču izlīdzinājuma veikšanai un izpildes laika mērīšanai (Python)

```
import os
import time
import csv
from functools import wraps
from subprocess import run, PIPE

fieldnames = ['no', 'software', 'filename', 'execution_time']
execution_data = []

def timefunc(func):
    @wraps(func)
    def time_closure(*args, **kwargs):
        global counter
        start = time.perf_counter()
        result = func(*args, **kwargs)
        time_elapsed = time.perf_counter() - start
        execution_data.append(
            {'no': counter, 'software': func.__name__, 'filename': args[0], 'execution_time':
time_elapsed})
        counter += 1
        return result
    return time_closure

@timefunc
def clustalo(filename):
    new_filename = filename.split(".")[0] + '.msf'
    os.system(f"clustalo -i final/data/{filename} -o final/outputs/clustalo/{new_filename} --
outfmt=msf")

@timefunc
def kalign(filename):
    new_filename = filename.split(".")[0] + '.msf'
    os.system(f"./kalign-3.3.1/src/kalign -i final/data/{filename} -o
final/outputs/kalign/{new_filename} --format msf")

@timefunc
def mafft(filename):
    os.system(f"mafft --auto final/data/{filename} > final/outputs/mafft/{filename}")

@timefunc
def muscle(filename):
    new_filename = filename.split(".")[0] + '.msf'
    os.system(f"muscle -in final/data/{filename} -out final/outputs/muscle/{new_filename} -msf")

for filepath in new_filepaths:
    for software in (kalign, clustalo, muscle):
        software(filepath)

with open('elapsed_time_data.csv', 'w') as f:
    writer = csv.DictWriter(f, fieldnames=fieldnames)
    writer.writeheader()
    for data in execution_data:
        writer.writerow(data)
```

2. PIELIKUMS

Kods izlīdzinājumu novērtējuma iegūšanai, izmantojot BALIBASE rīku (Python)

```
from subprocess import run, PIPE


def baliscore(software, filename):
    fn = filename.split('.')[0]
    output = run(["./BAliBASE_R9/src/bali_score",
f"final/references/{fn}.msf", f"final/outputs/{software}/{fn}.msf"],
                 stdout=PIPE)
    return output


if __name__ == '__main__':
    with open("baliscore_5.csv", 'w') as f:
        f.write(f"software,filepath,tcscore,spscore\n")
    for index, filepath in enumerate(new_filepaths):
        for software in ['clustalo', 'muscle', 'kalign', 'mafft']:
            output = baliscore(software, filepath)
            try:
                tc_score = output.stdout.decode('utf-8').strip().split("TC
score=")[1].split("\n")[0].strip()
                sp_score = output.stdout.decode('utf-8').strip().split("SP
score=")[1].split("\n")[0].strip()
            except IndexError:
                continue
            with open("baliscore_5.csv", 'a') as f:
                f.write(f"{software},{filepath},{tc_score},{sp_score}\n")
```

DOKUMENTĀRĀ LAPA

Maģistra darbs "VAIRĀKU SEKVENČU IZLĪDZINĀŠANAS METOŽU SALĪDZINĀJUMS" izstrādāts LU Datorikas fakultātē.

Darba teksta galīgā versija izgatavota 23.05.2021.

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: _____doc. Jevgēnijs Vihrovs_____
(Akad.amats, zin.grāds, vārds, uzvārds)

Darbs aizstāvēts maģistra gala pārbaudījuma komisijas sēdē

_____ prot. Nr. _____

(Darba aizstāvēšanas datums)

Komisijas sekretārs: _____
(Sekretāra paraksts)