TARTU ÜLIKOOL

LOODUS- JA TÄPPISTEADUSTE VALDKOND
ÖKOLOOGIA JA MAATEADUSTE INSTITUUT
BOTAANIKA OSAKOND

Rauno Kaiv

NCBI BLAST ja BLAST+ pakettide programmide blastn ja megablast ülevaade, versioonide võrdlus ja analüüs

Bakalaureusetöö

Juhendaja: Kessy Abarenkov, PhD

TARTU 2016

NCBI BLAST ja BLAST+ pakettide programmide blastn ja megablast ülevaade, versioonide võrdlus ja analüüs

Lühikokkuvõte:

Uurimistöö annab ülevaate BLAST pakettidest ja nende abil DNA järjestustele homoloogsete järjestuste otsimisest. Tutvustatakse BLAST algoritmi tööpõhimõtet, vajalikke kontseptsioone ja mõisteid ning selgitatakse, kuidas muuta otsinguparameetreid, ja mida see endaga kaasa toob. Lisaks tutvustatakse alternatiivseid programme, millega saab homoloogiaotsinguid sooritada.

Töö praktiline osa keskendub elurikkuse infosüsteemide PlutoF ja UNITE analüüsimoodulis kasutatavale BLAST paketi megablast otsingukäsule. Töös leiti, mida on vaja teha üleminekuks uuele BLAST+ homoloogiaotsinguprogrammile blastn, ja võrreldi katseliselt nende programmide päringute ressursikasutust (aeg, mälu). Lisaks analüüsiti päringuid maskeeritud ja indekseeritud andmebaasiga.

Märksõnad: BLAST, megablast, blastn, homoloogiaotsinguprogramm.

CERCS: B110 Bioinformaatika, meditsiiniinformaatika, biomatemaatika, biomeetrika.

NCBI BLAST and BLAST+ toolkits' applications blastn and megablast; overview, comparison and analysis

Abstract:

The thesis gives an overview of BLAST toolkits and how these are used for conducting DNA sequence searches. BLAST's algorithm, importantconcepts, core principlesand terms about BLASTare explained. In addition to that, alternative programs for querying DNA sequences are introduced.

The experimental part of the thesis examines a BLAST package program used by biodiversity information systems PlutoF and UNITE. Thesis results show the needed steps for migrating to a newer BLAST toolkit and what do new available features offer.

Keywords: BLAST, megablast, blastn, sequence similarity search program.

CERCS: B110 Bioinformatics, medical informatics, biomathematics, biometrics.

Sisukord

1. Sissejuhatus 4
2. Kirjanduse ülevaade 6
2.1. Ajalugu 6
2.2. Programmid BLAST+ paketis 7
2.3. Joonduse tüübid 8
2.4. BLAST-i algoritm 8
2.4.1. Häälestus 9
2.4.2. Eelotsing 9
2.4.3. Tagasijälitus 9
2.5. Päringu sooritamine tekstiterminalis 10
2.6. BLAST tulemuste hindamine 11
2.7. Järjestuste maskeerimine 12
2.8. BLAST+ uuendused, kasutamisvõimalused 13
2.9. Programmi blastn alternatiivid 15
2.9.1. HS-BLASTN 15
2.9.2. G-BLASTN 15
3. Praktiline osa 17
3.1. Materjal ja metoodika 17
3.2 Tulemused ja arutelu 19
Kokkuvõte 22
Summary 23
Kasutatud kirjandus 24
Lisad 26

1. Sissejuhatus

Uue tundmatu järjestuse (DNA, RNA või valk) avastamise korral on võimalik saada homoloogiaotsinguprogrammidest esmane informatsioon vastava polümeeri kohta. Leides tundmatule järjestusele andmebaasist sarnaseid, ühise evolutsioonilise päritoluga vasteid ehk homoloogseid järjestusi, on võimalik saada aimdust avastatud järjestuse funktsiooni ja fülogeneesi kohta.

Uue põlvkonna sekveneerimismeetodid on bioloogiliste järjestuste sekveneerimise muutnud teadlastele üha odavamaks ja kättesaadavamaks. Bioinformaatikas on seetõttu toimunud plahvatuslik kasv järjestusandmebaaside mahus. See on omakorda muutnud huvipakkuvate järjestuste leidmise suurenevatest andmestikest üha keerukamaks. Selleks, et probleemi leevendada, tuleks päringuid sooritada kas võimsama riistvara või kiirema tarkvaraga. Kuigi arvutite võimekus on pidevalt kasvanud, nullib andmehulga kasv riistvara uuenemisest tuleneva tulu. Seega jääb suur surve tarkvarapoolsetele lahendustele, mis peaksid leidma viise otsingutulemuste kiirendamiseks. (Remm 2015)

Üheks laialtkasutatavaimaks homoloogiaotsinguprogrammiks on saanud BLAST, millega on võimalik leida mistahes uuritavale järjestusele DNA- või valguandmebaasidest homoloogseid järjestusi. Nagu ka teised otsingutarkvarad, uueneb ka BLAST pidevalt, et leida uusi viise otsingute kiirendamiseks.

BLAST programmid on väga paindlikud, lubades kasutajatel parameetrite kaudu seadistada otsingu algoritmi vastavalt oma spetsiifilistele vajadustele. Tihtipeale on vaja leida kompromiss otsingu kiiruse ning täpsuse vahel. Kui programmi kasutaja soovib otsinguparameetreid ise muuta, peaks ta olema teadlik BLAST-i tööpõhimõtetest.

Uurimistöö eesmärkideks on:

1. anda ülevaade BLAST+ programmipaketist ja sellest, mida peaks teadma tehes päringuid nukleotiidjärjestustega;
2. näidata, kuidas saavutada kiiremaid homoloogiaotsingu tulemusi ja tutvustada mõningaid alternatiive BLAST+ paketi programmile blastn;
3. leida, millised on vajalikud sammud üleminekuks nukleotiidsete järjestuste homoloogiaotsinguprogrammilt megablast BLAST paketis programmile blastn BLAST+ paketis, ja mida see endaga kaasa toob;

4. kuna megablast ja blastn asuvad erinevates BLAST paketi versioonides, siis tuleks leida ka, mil viisil erineb paketi uus versioon vanast.

BLAST paketiga on võimalik sooritada otsinguid nii valgu kui ka DNA järjestustega ning paketti on võimalik kasutada nii eraldiseisva programmina (*stand-alone program*) kui ka veebirakendusena. Käesolev uurimistöö keskendub BLAST paketi osale, mis tegeleb nukleotiidjärjestustega (programmid blastn, megablast) ning mida kasutatakse personaalarvutites ja serverites.

2. Kirjanduse ülevaade

Bioinformaatikas on sekveneerimisandmete analüüsimiseks kõige sagedamini kasutatav Riikliku Biotehnoloogia Infokeskuse (*National Center for Biotechnology Information*, NCBI) programmipakett BLAST. Paketi nimi on akronüüm ingliskeelsest nimetusest *Basic Local Alignment Tool* ning see koosneb paljudest programmidest, millega on võimalik sooritada päringuid nii nukleotiidide kui ka valkude järjestustega. Programmiga on võimalik uuritavale järjestusele (*query sequence*) leida referentsandmebaasist vasteid (*subject sequence*) ning hinnata, kui suure kindlusega võib väita, et päringujärjestus ning tulemuseks saadud järjestused on homoloogid ehk neil on ühine evolutsiooniline päritolu.

BLAST päringuid on võimalik sooritada mitmel erineval viisil. Lihtsaim viis on BLAST-i kasutada veebirakendusena NCBI ametlikul kodulehel (http://blast.ncbi.nlm.nih.gov), kus kasutajad saavad otsinguid sooritada läbi graafilise kasutajaliidese. See kasutusviis on küll mugav, kuid teatud piirangutega. Nimelt ei ole veebis kasutatav BLAST sobilik sooritamaks mahukaid päringuid. Suuremahuliste päringute tegemiseks on mõeldud BLAST-i eraldiseisev pakett personaalarvuti jaoks, mida kasutatakse läbi tekstiterminali. Viimase puhul peab kasutajal arvutis olema ka andmebaas, millest järjestusi hakatakse otsima. Uurimistöö kirjandusülevaade käsitlebki just BLAST-i eraldiseisvat programmi.

2.1. Ajalugu

1981. aastal loodi Smith-Watermani algoritm (Smith 1981), mis erinevalt eelkäijatest suutis päringujärjestusele leida andmebaasist täieliku joonduse asemel osalisi joondusi (kohalikke joondusi). See on oluline näiteks domeense ehitusega valgujärjestuste puhul, sest kui varasemalt tundmatule proteiinile puudub andmebaasist täielik vaste, kuid leidub domeen selle otsitavast valgust, siis need vasted võivad anda aimdust tundmatu proteiini funktsioonist. Uus algoritm garanteeris suurima skooriga kohaliku joonduse leidmise, kuid see tagatis muutis päringute kestvuse liiga pikaks. Seetõttu pole Smith-Watermani algoritm enam laialdaselt kasutusel (Remm 2015).

Kiiremaid päringuid pakkus 1985. aastal publitseeritud FASTP heuristiline algoritm (hiljem FASTA), mis kasutas erinevaid otseteid, et kiiremini tulemusteni jõuda. FASTA programmi järgi sai nimetuse ka failivorming, mida kasutatakse laialdaselt bioloogiliste järjestuste salvestamiseks (Remm 2015).

Aastal 1990 avaldas NCBI FASTA-le alternatiivse homoloogiotsinguprogrammi BLAST. Uus programm oli FASTA-st *ca* 10 korda kiirem (Remm 2015). Just tänu oma töökiirusele on

BLAST tänaseni laialdaselt kasutatav bioinformaatika töövahend. Programm on läbinud mitmeid arenguetappe, millest olulisemad on kolm:

- 1990 – publitseeriti esimene BLAST programmi tutvustav teadusartikkel (Altschul *et al.* 1990). Uudne algoritm sooritas päringuid kiiremini kui varasemad vahendid, nagu näiteks FASTA. Just tänu kiiretele päringutulemustele sai uus programm teadlaste seas populaarseks. Paketi oluliseks puuduseks oli võimetus sooritada vahedega joondusi (*gapped alignment*). Seepärast ei suutnud BLAST otsing kõiki bioloogilise tähtsusega joondusi leida;

- 1997 – avaldati BLAST edasiarendus (Gapped BLAST, ka BLAST 2.0), mis suutis sooritada ka vahedega joondusi. Uus versioon oli varasemast keskmiselt kolm korda kiirem. (Altschul *et al.* 1997; Madden 2013);

- 2009 – publitseeriti hetkel kõige uuem pakett BLAST+. Uue paketi jaoks kirjutati BLAST-i lähtekood täielikult ümber, st algoritmi koostamist alustati puhtalt lehelt ning vana koodi ei kasutatud. Uus algoritm on kiirem tänu mitmetele mälu ning protsessori kasutust optimeerivale muudatusele (vt ptk 2.8.) Kuna vana paketi arendamine lõpetati, siis soovitati edaspidi tungivalt kasutada BLAST+ paketti. (Camacho e*t al.* 2009)

2.2. Programmid BLAST+ paketis

BLAST+ pakett koosneb programmidest, mida on vaja järjestuste otsimiseks andmebaasist. Lisaks otsinguprogrammidele on paketis ka andmebaaside- ning maskeerimisprogrammid. Andmebaasiprogrammidega sooritatakse näiteks andmestike vormindamist ning indekseerimist. Maskeerimistarkvaraga peidetakse järjestustes alad, mis võivad põhjustada eksitavate tulemuste saamist.

DNA järjestuste otsimiseks DNA andmebaasist on BLAST+ paketis programm blastn. Vaikimisi teeb see programm andmebaasist vähetundlikke ehk laiahaardelisi, ning seetõttu ka aeglaseid, päringuid. See tähendab, et blastn-ga on võimalik leida järjestusi, mis ei pea lähedalt suguluses olema. Kui tavaliselt peaks kasutaja otsingu tundlikkuse muutmiseks ise parameetreid muutma, siis blastn programmil on olemas neli parameetritekogu erinevate otsingu tundlikkustega, mida saab valida parameetriga *task* (vt Tabel 1). Vaikimisi kasutatakse megablast otsingutüüpi. (Madden 2013)

Kui otsingutüübid muudavad otsinguid tundlikumaks peamiselt sõnade pikkuse (*word size*) suurendamisega, siis kitsamaid otsinguid põhjustavad ka järgmised faktorid:

- maskeerimise kasutamine (vt ptk 2.7.);

- väiksema E-väärtuse (vt ptk 2.6.) kasutamine;
- vähesemate jooduste meelde jätmine otsingul (*Maximum target sequences*). (NCBI Handout Series 2016)

Tabel 1. blastn otsingutüübid, mida saab valida parameetriga *task* (NCBI 2016a, d).

blastn	blastn	Otsingutüüp, mis on mõeldud leidmaks kaugemalt suguluses järjestusi, nt järjestusi eri liikidest. Neljast otsingutüübist aeglaseim. Sõnade pikkus vaikimisi 11.
	megablast	Vaikimisi valitud otsingutüüp, mis on optimeeritud leidma väga sarnaseid, tavaliselt samast liigist järjestusi. Kiireim otsingutüüp. Sõnade pikkus vaikimisi 28.
	dc-megablast	Väiksema tundlikkusega megablast otsing. Sõnade pikkus vaikimisi 11.
	blastn-short	Optimeeritud leidma järjestusi, mis on lühemad kui 50 aluspaari. Sõnade pikkus vaikimisi 7.

2.3. Joonduse tüübid

Järjestuste joondus (*sequence alignment*) aitab kindlaks teha homoloogsed piirkonnad järjestuste vahel. BLAST algoritm on loodud leidma kohalikke joondusi järjestuste vahel. Erinevaid joondusi järjestuste vahel võib jagada kolmeks alamtüübiks (vt joonis 1). Need on kohalik joondus (*local alignment*), ülekattega joondus (*semiglobal alignment, overlap alignment*) ja täielik joondus (*global alignment*). Täieliku joonduse korral üritatakse leida kattuvus kogu järjestuse ulatuses. Ülekattega joonduses kattuvad ühe järjestuse algus ning teise lõpp. Kohaliku joonduse korral leitakse kattuvus vaid teatud osast kogu järjestuses. Bioinformaatikas kasutatakse kõige laialdasemalt kohalikku joondust, sest see on sobilik uuritavale järjestusele homoloogsete lõikude leidmiseks nukleotiidi või valgu järjestusest andmebaasist. (Remm 2015)

CATGCATGCATGCA ||| |||| CAT_CATG	CATGCATGCATGCA ||||| | TGCATGC_T	CATGCATGCATGCA ||||| | | |||| CATGCCT_C_TGCA
Kohalik joondus	Ülekattega joondus	Täielik joondus

Joonis 1. Joonduste kolm alamtüüpi.

2.4. BLAST-i algoritm

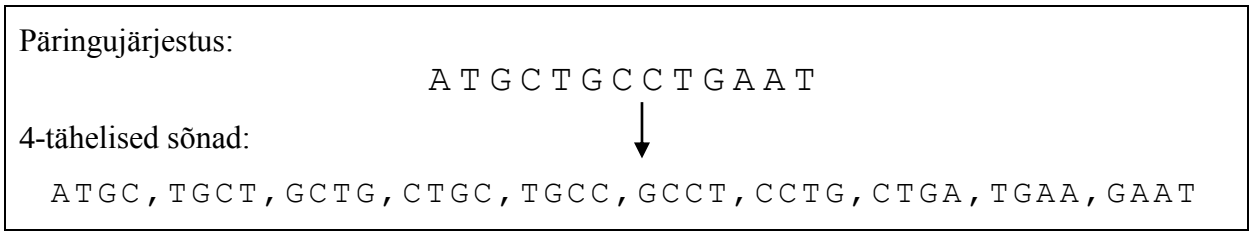
BLAST-iga spetsiifiliste päringute sooritamiseks on vaja muuta otsingu parameetreid. Kui sätestada päringut kiiremaks, tähendab see tihtipeale väiksemat täpsust otsingutulemustes ehk osa tulemusi ei pruugita üles leida. Seega peaks kasutaja teadma, kui kaugelt suguluses olevaid järjestusi soovitakse leida, et saavutada aktsepteeritav kompromiss kiiruse ning otsingutäpsuse vahel. Näiteks kui soovitakse teha päringuid järjestuste vahel, mis on pärit eri liikidest, tuleb arvestada aeglasema päringuga.

BLAST-i heuristiline otsing kasutab tulemuste kiiremaks leidmiseks otseteid ning seetõttu

pole garanteeritud parima joonduse leidmine. Päringu võib jagada kolmeks etapiks: häälestus (*setup*), eelotsing (*preliminary search, scanning*) ja tagasijälitus (*traceback*). **(**Madden 2013)

2.4.1. Häälestus

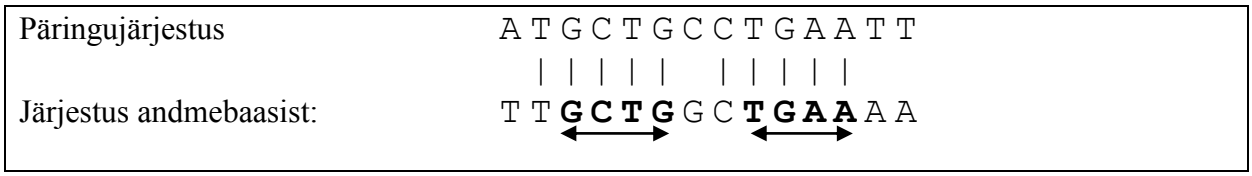
Häälestusetapis loetakse operatiivmällu päringujärjestused, otsingu parameetrid ning andmebaas. Otsingujärjestusest luuakse kindla pikkusega lõigud, mida kutsutakse sõnadeks (*words*) (vt Joonis 2). Sõnadest luuakse otsingutabel (*lookup table*). Selles faasis toimub vajadusel ka päringujärjestuse maskeerimine (vt ptk 2.7.). (Altschul *et al.* 1997)



Joonis 2. Näidisjärjestus ning kõik selle järjestuse neljatähelised sõnad.

2.4.2. Eelotsing

Selles faasis hakatakse andmebaasist otsingutabelis olevatele sõnadele vasteid otsima (*seeding*). Saadud vasted on päringujärjestusest palju lühemad ning järgmise sammuna hakatakse neid mõlemas suunas pikendama niikaua, kuni skoor enam ei suurene (vt Joonis 3). Vähendamaks pikendamise faasi jõudvate vastete arvu, peavad vastel olema vähemalt kaks lähestikku asutavad sõna. Skoor muutub vastavalt sellele, kas pikendatud ja otsingujärjestuse vahel on paardumine (*match*) või mitte (*mismatch*). Pikendamisel ei võeta arvesse deletsioone ega insertsioone, st toimub vahedeta pikendamine. Ainult joondused, mille skoor ületab lävendiparameetrit T, jäetakse järgmise etapi jaoks mällu. (Altschul *et al.* 1997)

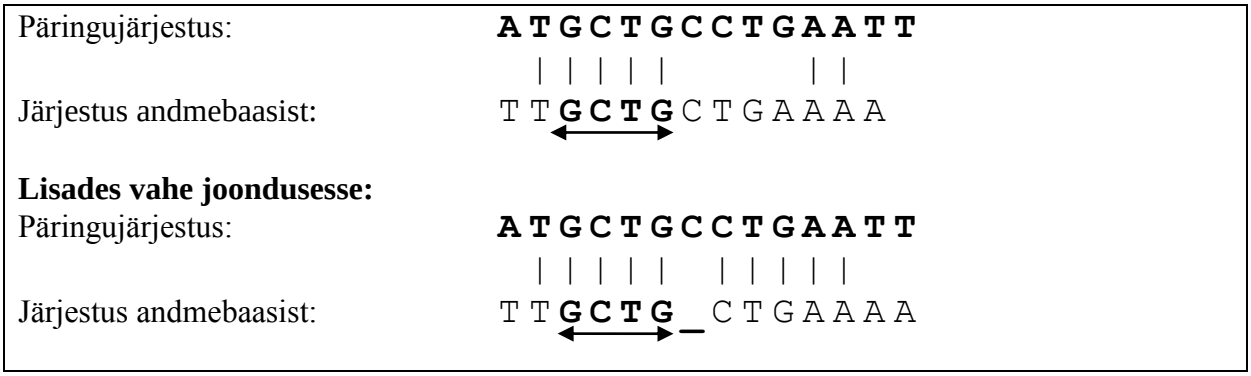


Joonis 3. Vahedeta pikendamine. Algoritm hakkab joonisel olevat kahte neljatähelist sõna pikendama, et leida kõrgeim skoor. Päringujärjestuse ja andmebaasist järjestuse vahel on selles näites 10 paardumist. Kui paardumise eest oleks näiteks määratud 4 punkti, siis selle joonduse skoor oleks pikendamise lõpuks 40p (10 x 4p).

2.4.3. Tagasijälitus

Kui eelnevad pikendamised toimusid vahedeta, siis selles etapis proovitakse vahedega pikendamisega joonduse skoori suurendada ehk veel täpsemaid vasteid leida (vt Joonis 4). Ajaliselt kõige nõudlikum ongi pikendamisprotsess, millele kulub vähemalt 90% päringu

kestvusest. Päringute kiirendamiseks tuleks suurendada algoritmi lävendiparameetrit T, mis vähendab pikendamisele kuuluvate järjestuste arvu ehk nii vähendatakse otsinguruumi (*search space*). Viimase sammuna väljastatakse tulemused kasutaja poolt määratud vormingus. (Altschul *et al.* 1997)



Joonis 4. Vahedega pikendamine. Joonisel on näha, kuidas joondusesse vahe sisse toomine võib parandada joonduse skoori. Kuigi vahe eest joonduses antakse trahvipunkte, võib vahede lisamine uusi paardumisi luua ja skoori parandada.

2.5. Päringu sooritamine tekstiterminalis

Päringu sooritamiseks tekstiterminalis valitakse esmalt sobilik programm. Näiteks nukleotiidse järjestuse otsimiseks DNA andmebaasist kasutatakse blastn-i. Järgmisena peab kasutaja arvutis olema andmebaas BLAST formaadis. Tihtipeale võivad andmebaasid olla aga laialtlevinud FASTA vormingus. Olemasoleva FASTA vormingus andmebaasi sobivasse vormingusse viimiseks on BLAST+ paketis programm makeblastdb.

Valitud programm käivitatakse koos kindlate märksõnade ehk parameetritega. Otsingukäsus peab parameetritele eelnema sidekriips, et programm need ära tunneks. Parameetrile järgneb selle väärtus. Näide otsingukäsust on toodud Joonisel 5, uurimistöös kasutatud parameetrite selgitused on Lisas 1.

Kõige lihtsam on järjestust andmebaasist otsida täpsustades vaid kaks parameetrit: päringujärjestus ning andmebaasi asukoht arvutis. Kui tulemuste väljundfaili asukohta otsingukäsus ei määrata, väljastatakse tulemus ekraanile. Reeglina oleks vajalik otsingutulemused salvestada aga faili. Kui varasemas BLAST paketis oli otsingutulemusi võimalik salvestada vaid *.txt* ja *.xml* formaadis, siis BLAST+ tulemusi on võimalik väljastada ka *.json* vormingus. (NCBI 2016d)

Päringu kiirust ning tundlikkust mõjutavad kõige rohkem pa-rameetrid, mis määravad sõnade pikkuse ning karistuse vahede eest joonduses (*gap penalty,* vt ptk 2.6.) (McGinnis *et al.* 2004). Kiireimate tulemuste saamiseks on soovituslik sooritada väikseima tundlikkusega

otsinguid, millega veel leiab huvipakkuvaid järjestusi.

```
$blastn -query sisendfail.fasta -db andmebaas.fasta -w 10 -out väljundfail.txt
```

Joonis 5. Näide otsingukäsust blastn programmiga. Parameetrite selgitused on toodud Lisas 1.

Päringuid tundmatute järjestustega ei sooritata tavaliselt ühekaupa. Korraga on võimalik otsida vasteid isegi sadadele või tuhandetele järjestustele. Selleks peavad päringujärjestused olema ühes failis FASTA formaadis. Iga järjestus failis peab algama uuelt realt sümboliga „>“, millele järgneb järjestuse nimi, kirjeldus (mittekohustuslik), ja alles siis, uuelt realt, järjestus ise. Näide FASTA vormingust on Joonisel 6.

```
>gi|1015806892|emb|LN876644.1| Cladosporium halotolerans genomic DNA sequence contains ITS1, 5.8S
rRNA gene and ITS2, isolate PMES_TB3
GGCCTGGATGTTCACAACCCTTTGTTGTCCGACTCTGTTGCCTCCGGGGCGACCCTGCCTCCGGGCGGGGGCC
CCGGGTGGACATCTCAAACTCTTGCGTAACTTTGCAGTCTGAGTAAATTTAATTAATAAATTAAAACTTTCAA
CAACGGATCTCTTGGTTCTGGCATCGATGAAGAACGCAGCGAAATGCGATAAGTAATGTGAATTGCAGAATT
CAGTGAATCATCGAATCTTTGAACGCACATTGCGCCCCCTGGTATTCCGGGGGGCATGCCTGTTCGAGCGTCA
TTTCACCACTCAAGCCTCGCTTGGTATTGGGCGACGCGGTCCGCCGCGCGCCTCAAATCGACCGGCTGGGTCT
TTCGTCCCCTCAGCGTTGTGGAAACTATTCGCTAAAGGGTGCCGCGGGAGGCCACGCCGTAAAACAACCCCA
TTTCTAAGGTTGACCTCGGATCAGGTAGGGATACCCGCTGAACTTAAGCATATCAAAAGTCGGAGGAAGTAG
GAATACCCGCTGAAATTAAGCATATCAATAAGTCGGA
```

Joonis 6.DNA järjestus FASTA formaadis. Kahel esimesel real asub järjestuse kirjeldus ning nimi, mis on kujutatud paksus kirjas. GI (*GenInfo Identifier*) number on NCBI poolt määratud identifikaator järjestusele. Kolmandast reast kujutatakse *C. halotolerans*-i ühte genoomse DNA järjestust. Oluline on mainida, et DNA järjestus asub antud joonisel kolmandal real, sest tekstiredaktor poolitab esimest rida. FASTA formaadis failides algab järjestus tegelikult teisest reast. Järjestus on võetud GenBank andmebaasist.

2.6. BLAST tulemuste hindamine

BLAST tulemustes on mitmeid arvulisi näitajad, mis aitavad hinnata, kui suur on tõenäosus, et kattuvused joondustes pole juhuslikud, vaid tegemist on homoloogiaga. Näidis kirjeldustega BLAST päringu tulemusest on nähtav Joonisel 7. Kõige olulisemad näitajad BLAST tulemuste hindamiseks on järgmised:

- skoor – väljendab kvantitatiivselt kahe järjestuse sarnasust (*sequence similarity*) – mida kõrgem skoor, seda sarnasemad järjestused. Skoori saamiseks liidetakse kõigepealt kokku punktid paarduvate tähtede eest ning siis lahutatakse skoorist maha trahvipunktid, mis antakse mittepaarduvate tähtede ning vahede eest joonduses.
- vahed joondustes (*alignment gap*) – näitavad, et ühel järjestusel joonduses on evolutsiooni käigus toimunud insertsioon või deletsioon. Lisaks mutatsioonidele võivad vahed tekkida ka sekveneerimisel tekkinud vigade tõttu. Uue põlvkonna sekveneerimismeetodeid kasutades võib veaprotsent olla kuni 1% (Fox *et al.* 2014).

Joonduse tulemustes on näha vaid vahede arvu ning asukohta, aga tasub ka teada, mil määral vahed joonduses skoori mõjutavad.

Mutatsioonid DNA-s ei toimu tavaliselt ühe, vaid mitme nukleotiidi kaupa. Seda eripära on vaja arvesse võtta, kui arvutatakse trahvipunkte vahede eest. Seetõttu kasutatakse kaheosalist vahe karistuse süsteemi (*affine gap alignment*), mille puhul antakse esimese vahe tekkimise eest joonduses (n-ö vahe avamine) tunduvalt suurem skoor kui sellele järgnevate vahede eest. Näiteks määrab blastn vahe avamise trahviks 5 ning iga järgneva vahe eest joonduses 2 punkti. Seega, kui joonduses oleks 4 nukleotiidi pikkune vahe, annaks blastn selle eest 13 trahvipunkti (5p + 4 x 2p). Kui aga joonduses oleks neli eraldi asetsevat ühe-nukleotiidilist vahet, oleks summaarne trahvipunktide arv 20 (4 x 5p).

- E-väärtus – järjestuste vahel võib kattuvusi esineda ka läbi juhuse. Sellisel juhul ei oma joondus mingisugust bioloogilist tähtsust. Skoori bioloogilist olulisust aitab kirjeldada E-väärtus (*expected value, E-value*), mis näitab, mitu joondust, sama või suurema skooriga, on võimalik kasutatud andmebaasist saada juhuslikult. E-väärtus on suur näiteks selliste järjestuste puhul, millel esineb väikese komplekssusega piirkondi (*low-complexity sequence region*). Kui kahe järjestuse vahel esineb suuremat sarnasust, kui võiks eeldada juhuslikust kattuvusest, siis võib eeldada, et järjestused on ühise evolutsioonilise päritoluga ehk tegu on homoloogsete järjestustega (homoloogidega).

```
>Cenococcum_sp|AB839377|SH214461.07FU|reps|k__Fungi;p__Ascomycota;c_
          _Dothideomycetes;o__Hysteriales;f__Gloniaceae;g__Cenococ
          cum;s__Cenococcum_sp
          Length = 475
 Score =  262 bits (152), Expect = 2e-69
 Identities = 164/170 (96%), Gaps = 1/170 (0%)
 Strand = Plus / Plus

Query: 584 ctgcggaaggatcattacagaaagtaaacgcggatcaaaccgcgaacttttaaacctttg 643
           |||||||||||||||||||||||||||||||| |||| ||||||||||||||||||||
Sbjct: 1   ctgcggaaggatcattacagaaagtaaacgcgggtcaagccgcgaacttttaaacctttg 60
```

Joonis 7. Näidislõik päringutulemuste failist. Esimesed kolm rida on järjestuse nimetus. *Length* näitab päringujärjestuse pikkust, S*core* on joonduse skoor, *Expect* on E-väärtus, *Identities* on identsusprotsent, mis kirjeldab järjestuse kattuvust, *Gaps* näitab vahede arvu joonduses, *Strand* näitab mõlema järjestuse suunda (*plus* – 5' -> 3' suund, *minus* – 3' -> 5' suund).

2.7. Järjestuste maskeerimine

Nukleiinhapete järjestustes pole nukleotiidid juhuslikult jaotunud. Tihtipeale esineb DNA-s regioone, kus mõned nukleotiidid esinevad keskmiselt sagedamini. Näiteks esineb nii

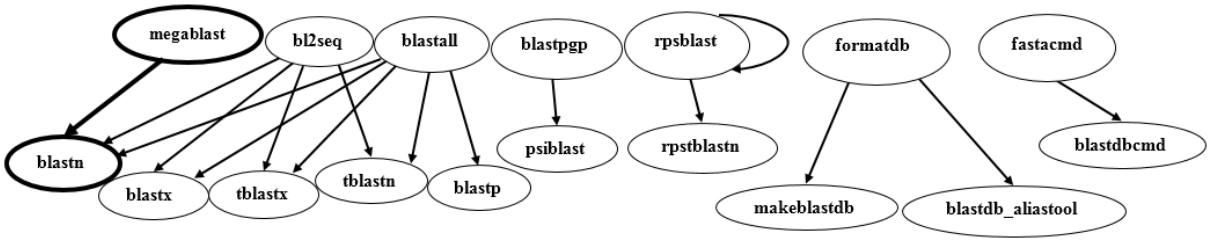
bakterite, arhede kui ka eukarütootide genoomis A-T rikkaid piirkondi, kuhu kinnituvad replikatsiooni initsieerivad valgud (Rajewska *et al.* 2012).

Seda on BLAST-i kasutades oluline teada, sest väikese komplekssusega järjestused, kus üks nukleotiid või nukleotiidide kombinatsioon esineb keskmiselt sagedamini, saavad algoritmilt ekslikult kõrge joonduse skoori. St kõrget skoori on võimalik saada ka läbi juhuse ning sel juhul ei oma joondus bioloogilist tähtsust. Väikese komplekssusega regioonid oleks seetõttu vajalik ära peita ehk maskeerida.

Väikese keerukusega piirkondade varjamiseks on kaks meetodit. Jäiga maskeerimise (*hard-masking*) puhul asendatakse maskeeritava piirkonna tähed selleks ette nähtud tähtedega. Nukleiinhappe järjestuste puhul N-idga, aminohappe järjestuste puhul X-idega. Paindliku maskeerimise (*soft-masking*) korral vahetatakse tavapäraselt suurtähtedega kujutatud monomeerid väiketähtedega, sellisel juhul jääb maskeeritud ala sisuline pool alles. BLAST+ paketiga on võimalik maskeerimist rakendada nii päringujärjestustele kui ka tervele andmebaasile. (Remm 2015)

2.8. BLAST+ uuendused, kasutamisvõimalused

Võrreldes uut BLAST+ paketti vanaga, on esmalt märgata, et enamikke programme paketis enam ei ole. Kaotatud programmide funktsioonid on ümber jaotatud uute programmide vahel. Ümberstruktureerimist illustreerib Joonis 8. Selle ümberkorraldusega püüti luua uus loogilise tööjaotusega struktuur, mis sarnaneks ka BLAST veebiversiooniga.



Joonis 8. Ülevaade BLAST+ restruktureerimisest. Joonisel on näha, kuidas BLAST paketi programmide (ülemised) funktsioonid on jaotatud BLAST+ programmide (alumised) vahel. Paksudes ringides on uurimistöö fookuses olevad programmid. (NCBI 2016c)

Teiseks suureks muutuseks on otsinguparameetrite uued pikemad nimed. Et uues BLAST+ paketis kasutada vana paketi otsingukäske, tuleb käskudele leida uute parameetritega vasted. Uues paketis on selleks skript *legacy_blast.pl,* mis küll teisendab parameetrid, kuid ei kontrolli parameetrite väärtusi. See võib olla problemaatiline, sest näiteks blastn programmi puhul peavad osade parameetrite väärtused olema antud kindlas vahekorras. Nimelt igale lubatud paardumise/mittepaardumise parameetri paarile peavad vastama kindla suhtega vahe

karistuse väärtused. Seetõttu ei pruugi kõik BLAST käsud BLAST+ paketis töötada, ilma et parameetrite väärtusi muudetaks. (NCBI 2016b)

Lisaks programmide funktsioonide ümberjaotamisele ja otsinguparameetrite nimede muutmisele tutvustati BLAST+ uuendustena ka järgmisi olulisi muudatusi:

- päringute aheldamine – BLAST+ paketis kasutatakse mitme päringujärjestusega otsingu korral päringute kiirendamiseks andmebaasi ainult ühe korra. Seda funktsiooni nimetatakse päringute aheldamiseks (*concatenation of queries*). Eriti suure mõjuga on see mahukate megablast otsingute ja vähesel määral ka blastn otsingute puhul; (NCBI 2016a)
- järjestuste poolitamine – BLAST+ kasutab päringujärjestuste poolitamist (*query splitting*), mis aitab otsinguid kiirendada just pikemate järjestuste (alates 500 tähest) puhul. See muudatus võimaldab häälestusetapis kasutada väiksemat otsingutabelit. Järjestuste poolitamine muudab otsinguid kiiremaks just pikemate järjestuste puhul, lühemate järjestuste puhul ei muutnud poolitamine päringute jõudlust ei paremaks ega halvemaks; (Camacho *et al.* 2009)
- kaugteenus – BLAST+ paketis on nüüd võimalus saata oma arvutist päringuid NCBI serveritesse. Sellisel juhul ei toimu otsinguarvutused kasutaja arvutis, vaid NCBI serveris. Kaugteenus (*remote service*) on eriti kasulik juhtudel, kui oleks vaja sooritada mahukas päring ning kasutaja arvuti pole piisavalt võimas. Selle kasutamiseks on otsingukäsule vaja lisada parameeter *remote* ja täpsustada NCBI serveris olev andmebaas, mida soovitakse kasutada. Kasutaja enda andmebaasi ei ole võimalik selle teenusega kasutada; (NCBI 2016a)
- WindowMasker – WindowMasker on programm BLAST+ paketis, mis identifitseerib ja peidab sageli korduvad ning väikese komplekssusega alad andmebaasi järjestustes, et vältida liiga paljude tabamuste tekkimist sõnade otsimise etapis. Vähem tabamusi pikendamise faasis muudab järjestuste otsimise andmebaasist küll kiiremaks, aga see võib põhjustada ka osade järjestuste leidmata jätmist. Alade varjamiseks kasutatakse paindlikku maskeerimist. Võrreldes varasemate maskeerimisprogrammidega, on WindowMasker kiirem ning ei vaja maskeeritavate alade identifitseerimiseks andmestikku maskeerimisinfoga; (Morgulis *et al.* 2006)
- megablasti indekseeritud otsing – tavaliselt alustab otsimisalgoritm andmebaasi läbivaatamist rea kaupa algusest lõpuni. Suurte andmestike puhul on see ajakulukas

ning ebaefektiivne. Palju tõhusam oleks andmestikku lugema hakata kohe huvipakkuva järjestuse juurest. Seda on võimalik teha indekseeritud andmebaasidega. Andmebaasi indekseerimise käigus luuakse sorteeritud indeksfail, milles on kirjeldatud kõigi andmebaasis olevate kirjete algus- ja lõppkoordinaadid. Luuakse justkui aadressiraamat, milles on kõik kirjed tähestikulises järjekorras ning mille järgi on hõlbus huvipakkuv järjestus üles leida. BLAST+ paketis on andmebaaside indekseerimiseks programm makembindex. Kasutades megablasti indekseeritud andmebaasiga, tuleks arvestada teatud piirangutega. Kui otsingul tekib liiga palju tabamusi, siis indekseeritud megablast (*indexed megablast*) kiiremaid tulemusi ei anna. Selle vältimiseks tuleks referentsandmebaas esmalt maskeerida näiteks WindowMasker programmiga. Indeksfail on andmebaasist *ca* neli korda suurem ning kui see ei mahu arvuti operatiivmällu, kaovad indekseeritud megablast otsingul kiiruse eeliseid. Lisaks pole võimalik otsinguid sooritada väiksema sõna pikkusega kui 16. (Morgulis *et al.* 2008)

2.9. Programmi blastn alternatiivid

DNA homoloogiaotsinguteks on lisaks BLAST+ programmile blastn mitmeid alternatiivseid programme, nagu näiteks kiiremaid otsingutulemusi lubavad HS-BLAST ja G-BLASTN. Mõlema programmi kasutamise muudab mugavaks asjaolu, et paljud nende parameetrid kattuvad blastn parameetritega.

2.9.1. HS-BLASTN

HS-BLASTN (Chen *et al.* 2015) on mõeldud leidmaks väga sarnaseid järjestusi. Programmi algoritm sarnaneb NCBI BLAST omaga, modifitseeritud on algoritmi esimesi etappe – sõnadest otsingutabeli koostamist ja sõnade otsimist andmebaasist. HS-BLASTN on sobilik just paljude päringujärjestuste otsimiseks mahukatest referentsandmebaasidest ja väljastab samu joondusi mis NCBI megablast. Programmi loojad leidsid oma katsetes, et nende programm on megablastist kuni 20 korda kiirem. Katses otsiti liigi *Homo sapiens* geene inimgenoomi andmebaasist (hg38).

2.9.2. G-BLASTN

Kui eelnev programm lubas kiiremaid tulemusi tänu otsingualgoritmi modifitseerimisele, siis uudset lähenemist otsingute kiirendamiseks pakub NCBI BLAST baasil loodud G-BLASTN (Zhao *et al.* 2014), mis kasutab otsingute kiirendamiseks graafikaprotsessorit (GPU-d). Kuigi GPU-ga töötavaid valkude homoloogiaotsinguprogramme oli juba varasemalt kasutusel (nt

GPU-BLAST), siis G-BLASTN-iga sai esmakordselt sooritada otsinguid nukleotiidsete järjestustega. GPU tehnoloogia muudab ahvatlevaks hinna ja võimsuse suhe. Vouzis ja Sahinidis (2011) hinnangul võimaldab GPU tehnoloogia kasutamine personaalarvutites saavutada superarvutite võimsust. Kuid tuleb arvestada, et algoritmid, mis töötavad hästi CPU-l (keskprotsessoril) ei pruugi seda teha GPU-l. Seetõttu tuli ka G-BLASTN autoritel muuta esmalt NCBI-BLAST algoritmi enne kui nad jõudsid kiiremate otsingutulemusteni. Programmi võimekus sõltub suuresti kasutatava GPU võimekusest – päringu referentsandmebaas peab mahtuma GPU mällu.

3. Praktiline osa

Uurimistöös selgitati välja, mida on vaja teha selleks, et vanalt megablast programmilt üle minna uuele blastn tarkvarale. Ehk kuidas saaks kasutada megablastis kasutatavat otsingukäsku blastn-is. Uuele paketile üle minnes tuleb arvestada sellega, et BLAST+ paketis megablast programmi ei ole. Megablast päringute tegemiseks on ette nähtud otsinguprogramm blastn. Lisaks võrreldi kui palju ressurssi (aega ja operatiivmälu) megablast ja blastn programmil päringute tegemiseks kulub. BLAST+ uutest võimalustest prooviti indekseeritud megablasti ja WindowMaskerit. Võrdlemaks indekseeritud ja indekseerimata megablasti, tuli esmalt referentsandmebaas WindowMasker programmiga maskeerida. Kirjanduse põhjal võib eeldada, et uus blastn on megablastist märkimisväärselt kiirem ja ka indekseeritud otsing peaks andma kiiremaid tulemusi indekseerimata otsingust.

3.1. Materjal ja metoodika

Megablast ja blastn võrdlemiseks kasutati megablast otsingukäsku, mida kasutavad elurikkuse infosüsteemi PlutoF (Abarenkov *et al.* 2010*)* ja seente molekulaarseks määramiseks kasutatava UNITE andmebaasi (Kõljalg *et al.* 2013) analüüsimoodulid. Valitud otsingukäsk on mõeldud väga sarnaste, samast liigist pärit, järjestuste leidmiseks. Megablast otsingukäsku ei saa kasutada otse blastn-is, sest uues paketis on uued parameetrite nimed. Seega, vanade megablast otsingukäskude kasutamiseks uues paketis tuleb neile leida uute parameetritega vasted. Megablast otsingukäsu teisendamiseks kasutati BLAST+ paketis olevat skripti *legacy_blast.pl*. Teisenduse tulemus on näidatud Tabelis 2.

Tabel 2. Üleval on analüüsimooduli megablast otsingukäsk väga sarnaste järjestuste leidmiseks, all on selle käsu vaste blastn-s kasutamiseks.

BLAST	$**megablast -a** 4 **-F** F **-d** <*andmebaas*> **-b** 1 **-r** 1 **-q** -2 **-i** <*päringujärjestused*> **-v** 30 **-W** 28 **-G** -1 **-E** -1 **-o** <*väljundfail*>
BLAST+	$**blastn -num_threads** 4 **-dust** no **-db** <andmebaas> **-num_alignments** 1 **-reward** 1 **-penalty** -2 **-query** <*päringujärjestused*> **-num_descriptions** 30 **-word_size** 28 **-gapopen** -1 **-gapextend** -1 **-out** <*väljundfail*>

Töö käigus selgus, et valitud otsingukäsu parameetrite väärtused ei ole blastn-is kasutuskõlblikud. *legacy_blast.pl* n-ö tõlgib küll vanad otsingukäsud uueks, kuid ei kontrolli, kas parameetrite väärtused on vahekorras, mida blastn programm aktsepteerib. Problemaatiliseks osutusid paardumise/mittepaardumise (*reward/penalty*) eest punkte määravate parameetrite ja vahe karistuse (*gapopen/gapextend*) parameetrite suhe. Selleks, et

mõlemat otsingukäsku saaks võrrelda võrdsete parameetrite väärtustega, muudeti konfliktsete parameetrite väärtusi järgnevalt:

- trahvipunktid vahe avamise eest joonduses (*gapopen*) – 0 (algselt -1);
- trahvipunktid vahe pikendamise eest (*gapextend*) – 2 (algselt -1).

Indekseeritud ja indekseerimata megablast võrdlemiseks otsustati kasutusele võtta suurem hulk päringujärjestusi (6000) kui kasutati megablast ja blastn võrdluses (maksimaalselt 500). Põhjuseks oli vajalik samm maskeerida referentsandmebaas WindowMaskeriga, mis muutis päringute kestvuse märkimisväärselt lühemaks ja kitsamaks (pikendamise faasi jõudis vähem järjestusi ja osadele päringujärjestustele ei leitud enam vasteid). Et saada võrdluseks pikemaid päringukestvusi, võeti kasutusele võimalikult suur järjestuste hulk. Piiravaks teguriks osutus sel juhul katsete läbiviimiseks kasutatud arvuti operatiivmälu. Otsingu muutmiseks laiahaardelisemaks, kasutati otsingukäsus väikseimat sõna pikkust, mida indekseeritud megablastiga on võimalik kasutada (16). Indekseeritud megablast päringute sooritamiseks indekseeriti andmebaas programmiga makembindex ning blastn käsule lisati parameeter -*use_index* väärtusega *true.*

Päringutulemuste faile võrreldi töö autori kirjutatud skriptiga (https://github.com/RaunoKa/BLAST), mis kontrollis, kas iga järjestuse parim tulemus eri failides kattub. Uurimistöö läbiviimiseks kasutati programme blastn ja WindowMasker BLAST+ paketist (versioon 2.3.0) ning megablasti BLAST paketist (2.2.26). Testimiseks kasutatud referentsandmebaas, kuupäevaga 31. jaanuar 2016, laaditi alla UNITE kodulehelt (https://unite.ut.ee/repository.php) ning selles oli 43 293 järjestust. Järjestused, mida andmebaasist otsima hakati, laaditi alla NCBI kodulehelt (http://www.ncbi.nlm.nih.gov/nuccore), GenBank andmebaasist. Järjestuste otsimisel kasutatud päring on näidatud Joonisel 9.

```
(("Fungi"[Organism] AND (00000000140[SLEN] : 00000003000[SLEN])) AND (((ITS1[titl] OR ITS2[titl]) OR 5.8S[titl]) OR "internal transcribed spacer"[titl] OR "internal transcribed spacers"[titl] OR "ITS 1"[titl] OR "ITS 2"[titl])) AND ("2016/02/02"[PDAT] : "2016/02/03"[PDAT])
```

Joonis 9. Päring NCBI kodulehelt päringujärjestuste saamiseks. Selle päringu esimest 500 järjestust kasutati megablast ja blastn võrdlemiseks. Päringus on määratud järjestuste taksonoomiline kuuluvus, lubatud järjestuste pikkuste vahemik, molekuli tüüp ja avaldamise kuupäev. 6000 päringujärjestuse saamiseks muudeti päringus järjestuste avaldamise kuupäeva vahemikku ("2016/04/02"[PDAT] : "2016/05/03"[PDAT]) ja kasutati 6000 esimest järjestust tulemustes.

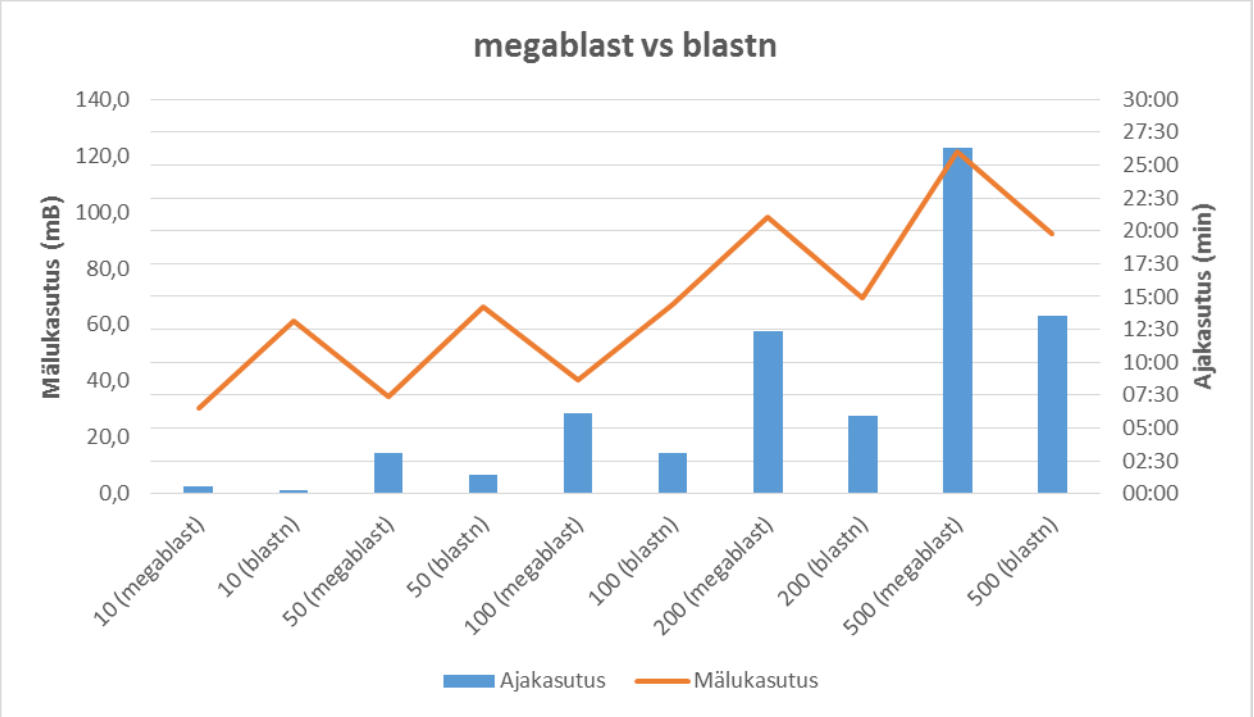
Nii andmebaasi- kui ka päringujärjestused olid seente ribosomaalse DNA ITS *(Internal Transcribed Spacer)*, ametliku seente triipkoodi, järjestused (Schoch et al. 2012). Päringute

aja- ja mälukasutuse mõõtmiseks kasutati terminalikäsku *time* koos täpsustavate parameetritega (-f "%E %M"). Testid viidi läbi sülearvutiga, millel oli 64-bitine Linux operatsioonisüsteem, 4GB operatiivmälu (RAM) ja 4-tuumaline protsessor Intel® Core™ i5-2410M.

3.2 Tulemused ja arutelu

Megablast ja blastn võrdluse tulemused olid ootuspärased ja näitasid, et päringud uue BLAST+ paketiga on keskmiselt 2 korda kiiremad kõigi katsetatud järjestushulkadega (vt Joonis 10). Mälu kasutab BLAST+ väiksemate järjestushulkade puhul rohkem kui BLAST. Alles suuremate järjestushulkade puhul (200 ja 500 järjestust) muutub BLAST+ mälukasutus BLAST paketist optimaalsemaks.

Võrreldes megablast ja blastn 500 järjestusega päringu tulemuste faili, selgus, et ainult kolme järjestuse parimad tulemused ei kattunud. Nende kolme järjestuse puhul olid esimene ja teine parim tulemus vahetusse läinud. Seega on tulemused sisuliselt samad, kuid tuleks arvestada, et megablast ja blastn võivad samade parameetrite väärtuste korral anda tulemusi, milles parimate vastete edetabelis võivad üksikute järjestuste kohad vahetusse minna.



Joonis 10. Megablast ja blastn ressursikasutuse võrdlus eri järjestushulkadega (10, 50, 100, 200, 500). Kasutatud andmed on kolme korduskatse keskmised tulemused. Tabel katse tulemustega on esitatud Lisas 2.

WindowMaskeriga referentsandmebaasi maskeerimine muutis päringute sooritamise andmebaasist kitsamaks ja märkimisväärselt kiiremaks. Kui maskeerimata andmebaasiga

kulus 500 järjestusega päringuks keskmiselt 13,5 minutit, siis maskeeritud andmebaasiga kulus keskmiselt ligikaudu 3 sekundit. Küll aga ei leitud maskeeritud andmebaasist, sõna pikkusega 28, enam osadele päringujärjestustele vasteid (64-le 500st). Alles sõna pikkusega 13 leiti kõigile järjestustele vasted, päringute kestvus oli keskmiselt 12 sekundit. Pärast kõigile 500 päringujärjestusele tulemuste leidmist, võrreldi päringutulemusi maskeeritud ja maskeerimata andmebaasiga ning leiti, et parimad tulemused ei kattunud vaid üheksa järjestuse puhul.

Indekseeritud megablasti päring indekseerimata päringust kiiremaid tulemusi ei näidanud (vt Tabel 3). Võib oletada, et indekseeritud megablast otsing leidis andmebaasist, maskeerimisele vaatamata, siiski liiga palju vasteid pikendamise etapiks, mistõttu kiiremaid tulemusi ei saadud. Kuigi kasutati võimalikult väikest sõnade pikkust (16), ei leitud 6000 päringujärjestusest 158 järjestusele vastet. Seega on näha, et indekseeritud megablast ei sobi kõikideks töödeks, sest seda ei saa kasutada väiksemate sõnade pikkustega kui 16.

Tabel 3. Korduskatsed ressursikasutuse mõõtmiseks indekseeritud ja indekseerimata megablast päringuga 6000 päringujärjestusega ja sõna pikkusega 16.

Indekseeritud		Indekseerimata	
Aeg (min)	Mälu (mB)	Aeg (min)	Mälu (mB)
01:49	1372	02:00	1283
02:00	1378	01:59	1281
01:59	1372	01:44	1282
01:47	1373	01:50	1283
01:46	1376	01:48	1282

BLAST paketti kasutavad infosüsteemid, nagu PlutoF ja UNITE analüüsimoodul, peaksid kasutama võimalikult uut versiooni homoloogiaotsingu tarkvarast. Vanemalt versioonilt üleminek uuele võib olla aeganõudev, sest uue funktsionaalsuse kasutamine nõuab lisateadmisi. Lisaks peab uue paketi kasutusele võtmiseks muutma uute programmide jaoks otsingukäske, mis võib põhjustada otsingu tundlikkuse muutumist - enam ei leita uue käsuga samu järjestusi ja vajalik oleks otsingukäsu kalibreerimine. Töö autori arvates pakub BLAST+ piisavalt palju eeliseid, et tülikas üleminekuprotsess ette võtta. Uuendatud pakett sooritab päringuid kiiremini ning pakub uusi võimalusi, nagu näiteks kaugteenus ja indekseeritud megablast. Lisaks soodustab BLAST+ kasutamist NCBI põhjalik dokumentatsioon.

Edasiste samateemaliste uuringute puhul tasuks praktilised katsed läbi viia mahukamate andmetega (rohkemate päringujärjestuste ning suuremate andmebaasidega) ja võimekamal riistvaral (nt arvutusserveril), et suurte andmehulkade kasutamise valukohad selgemalt kajastuksid. Lisaks tuleks kasutada otsingutulemusi võrdlevat automatiseeritud süsteemi, mis

suudaks võrrelda väga paljusid joonduseid ja millega saaks võrrelda iga päringujärjestuse kõiki vasteid korraga (mitte ainult esimest). See aitaks leida parimaid otsinguparameetrite väärtuseid, mida oleks vaja teatud otsinguülesande sooritamiseks võimalikult kiirelt ja piisavalt täpselt. Käesolev töö annab edasiste katsete disainimiseks ja väljatöötamiseks hea aluse.

Kokkuvõte

Sekveneerimisandmete eksponentsiaalne kasv muudab andmebaasidest info kättesaamise üha ajakulukamaks. Selleks, et otsingud ei muutuks liialt ressursinõudlikuks, tuleb kasutada võimalikult kiireid viise ja vahendeid päringute sooritamiseks. Homoloogiaotsinguprogrammid, nagu BLAST, üritavad andmetulvaga sammu pidada ning pakuvad mitmeid uusi võimalusi otsingutulemuste kiirendamiseks. Programmi kasutajad peaksid üha enam olema teadlikud päringuprogrammide tööpõhimõtetest, kasutamisviisidest.

Käesolev uurimistöö andis ülevaate BLAST programmipaketist, selle ajaloost, arengust, tööpõhimõtetest, kasutusvõimalustest ja alternatiividest, keskendudes just DNA järjestuste otsingutele. Lisaks selgitati, mida peaks kasutaja teadma, et ise muuta otsinguparameetrite vaikeväärtusi, ja mida see kaasa toob.

Uurimistöö praktilises osas selgus, et megablast otsingukäske saab muuta uue blastn jaoks sobilikuks paketis oleva skriptiga *legacy_blast.pl*, aga sellisel juhul tuleb kasutajal veenduda, et parameetrite väärtused on antud sellises vahekorras, mida blastn toetab. St kõiki otsingukäske ei saagi blastn programmis kasutada ilma parameetrite väärtuseid muutmata. Muuta tuli ka uurimistööks valitud otsingukäsu parameetrite väärtusi, et sama käsu ressursikasutust eri programmides võrrelda saaks. Võrreldes megablast ja blastn programmide ressursikasutust oli näha, et uus blastn suutis päringuid sooritada keskmiselt kaks korda kiiremini kui megablast ja suuremate järjestushulkade puhul oli uuem programm mälukasutuse poolest optimaalsem.

Kasutades töös andmebaasi maskeerimiseks WindowMaskerit, selgus, et päringud selle andmebaasiga muutuvad märkimisväärselt kiiremaks ja ka kitsamaks. Seega, kasutades maskeeritud andmebaasi kiiremate tulemuste saavutamiseks, tuleb arvestada ka võimalusega, et osadele päringujärjestustele ei leita võib-olla enam vasteid. Peale selle leiti, et uurimistöös kasutatud andmetega (päringujärjestuste ja andmebaasiga) indekseeritud megablast kiiremaid tulemusi ei andnud.

NCBI BLAST and BLAST+ toolkits' applications blastn and megablast; overview, comparison and analysis

Rauno Kaiv

Summary

Finding information from the sequence databases is getting more difficult due to the exponential growth of sequencing data. In order to keep the query times reasonable, one should use the most efficient ways and means for querying sequences. Sequence similarity search program BLAST tries to be one step ahead from the ever-growing data floods by offering its users new ways for accelerating their sequence searches. BLAST users should be aware how the program works and what are the program's features. This might not always be the case, and because of that, BLAST might not be used to its fullest potential.

This thesis gives an overview of BLAST's stand-alone program, its history, development, algorithm, features and alternatives while focusing on the programs' parts which are used for querying DNA sequences. It is also discussed how and when the search parameters should be modified and how this affects the results.

In the experimental part of the thesis, where tests with different programs and features were conducted, it was concluded that when migrating from the old BLAST toolkit's program megablast to the new BLAST+ toolkit's blastn, it might not always be possible to use the same search commands as the ones used in the older toolkit, because BLAST+ might not support a set of search parameters. When megablast's search speeds were compared with blastn's, the results showed that the new program was usually twice as fast as the old one.

Also, when using the feature of masking a database with WindowMasker, the tests showed that using a masked database makes queries significantly faster, but also makes queries miss more hits as compared to a same search with an unmasked database. Testing the indexed megablast did not show any faster queries than non-indexed search.

Kasutatud kirjandus

- Abarenkov K, Tedersoo L, Nilsson RH, *et al*. PlutoF—a Web Based Workbench for Ecological and Taxonomic Research, with an Online Implementation for Fungal ITS Sequences. *Evolutionary Bioinformatics Online*. 2010; 6: 189-196.
- Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. *Journal of Molecular Biology*, 1990; 215: 403–10.
- Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, *et al*. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*, 1997; 25: 3389–402.
- Camacho C, Coulouris G, Avagyan V, Ma N, Papadopoulos J, Bealer K, *et al*. BLAST+: architecture and applications. *BMC Bioinformatics*, 2009; 10: 421.
- Chen Y, Ye W, Zhang Y, Xu Y. High speed BLASTN: an accelerated MegaBLAST search tool. *Nucleic Acids Research*, 2015; 43: 7762-7768.
- Fox EJ, Reid-Bayliss KS, Emond MJ, Loeb LA. Accuracy of Next Generation Sequencing Platforms. *Next generation, sequencing & applications*, 2014; 1: 1000106.
- Kõljalg U, Nilsson RH, Abarenkov K, Tedersoo L, Taylor AFS *et al.* Towards a unified paradigm for sequence-based identification of fungi. *MolecularEcology*, 2013; 22: 5271–5277.
- Madden T. The BLAST Sequence Analysis Tool. http://www.ncbi.nlm.nih.gov/books/NBK153387/ viimati uuendatud 15. märts 2013.
- Morgulis A, Gertz EM, Schäffer AA, Agarwala R. WindowMasker: window-based masker for sequenced genomes. *Bioinformatics*, 2006; 22: 134-141.
- Morgulis A, Coulouris G, Raytselis Y, Madden TL, Agarwala R, Schäffer AA. Database indexing for production MegaBLAST searches. *Bioinformatics*, 2008; 24: 1757-1764.
- NCBI ametlik dokumentatsioon (a). BLAST+ features. http://www.ncbi.nlm.nih.gov/books/NBK279668/ viimati alla laetud 01.05.2016.
- NCBI ametlik dokumentatsioon (b). BLASTN reward/penalty values. http://www.ncbi.nlm.nih.gov/books/NBK279678/ viimati alla laetud 01.05.2016.

- NCBI ametlik dokumentatsioon (c). Conversion from C toolkit applications. http://www.ncbi.nlm.nih.gov/books/NBK279683/ viimati alla laetud 01.05.2016.
- NCBI ametlik dokumentatsioon (d). Options for the command-line applications. http://www.ncbi.nlm.nih.gov/books/NBK279675/ viimati alla laetud 01.05.2016
- NCBI Handout Series. BLAST homepage & search pages. ftp://ftp.ncbi.nlm.nih.gov/pub/factsheets/HowTo_BLASTGuide.pdf viimati uuendatud 18.03.2016
- Rajewska M, Wegrzyn K, Konieczny I. AT-rich region and repeated sequences – the essential elements of replication origins of bacterial replicons. *FEMS Microbiology Reviews*, 2012; 36: 408-434;
- Remm M. *Bioinformaatika*. Tartu Ülikooli Kirjastus, Tartu 2015.
- Schoch CL, Seifert KA, Huhndorf S, et al. Nuclear ribosomal internal transcribed spacer (ITS) region as a universal DNA barcode marker for Fungi. Proceedings of the *National Academy of Sciences of the United States of America*, 2012;109: 6241-6246.
- Smith TF, Waterman MS. Identification of Common Molecular Subsequences. *Journal of Molecular Biology* 1981. 147: 195-197.
- Zhao K ja Chu X. G-BLASTN: accelerating nucleotide alignment by graphics processors. *Bioinformatics*, 2014; 30: 1384-1391.
- Vouzis PD, Sahinidis NV. GPU-BLAST: using graphics processors to accelerate protein sequence alignment. *Bioinformatics*, 2011; 27: 182-188.

Lisad

Lisa 1. BLAST pakettide käsureaparameetrid.

BLAST	BLAST+	Kirjeldus
-w	-word_size	Sõnade pikkus
-g	-gapopen	Trahvipunktid vahe avamise eest joonduses
-e	-gapextend	Trahvipunktid vahe pikendamise eest
-r	-reward	Punktid paardumise eest joonduses
-q	-penalty	Trahvipunktid mittepaardumise eest
-a	-num_threads	Kasutatavate protsessorite arv
-f	-dust	Järjestuse maskeerimise
-v	-num_descriptions	Joonduste kirjelduste arv tulemustes
-b	-num_alignments	Joonduste arv tulemustes
-m	-outfmt	Tulemuste formaadi valik
-i	-query	Päringujärjestuste faili asukoht
-d	-db	Kasutatav andmebaasi asukoht
-o	-out	Tulemuste faili asukoht

Lisa 2. Programmide megablast ja blastn päringute ressursikasutus eri järjestushulkadega.

Järjestust	Ajakasutus (min)	Mälukasutus (mB)
10 (megablast)	00:34	30,5
10 (blastn)	00:17	61,5
50 (megablast)	03:03	34,6
50 (blastn)	01:25	66,2
100 (megablast)	06:08	40,2
100 (blastn)	03:03	67,2
200 (megablast)	12:24	98,2
200 (blastn)	05:58	69,7
500 (megablast)	26:17	121,6
500 (blastn)	13:30	92,2

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, Rauno Kaiv

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose
NCBI BLAST ja BLAST+ pakettide programmide blastn ja megablast ülevaade, versioonide võrdlus ja analüüs

mille juhendaja on Kessy Abarenkov.

1.1. reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
1.2. üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace´i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.

3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, **19.05.2016**