Norges miljø- og biovitenskapelige universitet
Fakultet for veterinærmedisin og biovitenskap
Institutt for kjemi, bioteknologi og matvitenskap

Masteroppgave 2015
30 stp

Fylogenetisk analyse av helgenom NGS-data fra Bacillus cereus-gruppen

Phylogenetic Analysis of Whole Genome NGS Data from the Bacillus cereus Group

Benedicte Been Simensen

Forord

Denne masteroppgaven ble utført ved Institutt for Kjemi, Bioteknologi og Matvitenskap ved Norges Miljø og Biovitenskapelige universitet og i samarbeid med Forsvarets Forskningsinstitutt. I løpet av det halve året jeg har jobbet med oppgaven har jeg fått forståelse for helt nye ting samt kunnskap jeg håper jeg får brukt senere.

Lars Gustav Snipen, førsteamanuensis ved biostatistikkgruppen ved IKBM har vært min hovedveileder og til han vil jeg rette en stor takk. Han har engasjert seg i oppgaven min, hjulpet meg når det trengtes og alltid vært tilgjengelig for spørsmål. Lars har gitt meg utrolig god veiledning. I tillegg må jeg få takke Jaran Strand Olsen, forsker ved FFI, som min andre veileder. Han har bistått med rådata, og hele veien vist interesse og gitt oppmuntring, i tillegg til å bidra med synspunkter fra biologisk ståsted. Jeg er svært takknemlig for begges innsats i oppgaven min.

Videre må jeg få takke mamma, Marianne, for innsatsen i korrekturlesning og engasjement i oppgaven min, og pappa, Kjell, for oppmuntring og kloke ord. Min lillesøster Iselin fortjener også en stor takk for korrekturlesing og støtte, og det samme fortjener min venninne Elin som i tillegg til korrekturleser har vært en god turkamerat.

Ås, mai 2015

Benedicte Been Simensen

Sammendrag

Sammenlikning av bakteriers genom er en effektiv måte å kartlegge hva som skiller arter og stammer fra hverandre. For å sammenlikne hele genomer fra bakterier er det vanlig å sekvensere. Utfordringen er dagens sekvenseringsteknologi, som ikke klarer å lese av hele lengden til genom-sekvensen, men kutter DNA-molekylene i kortere fragmenter før avlesning. Resultatet etter sekvensering er et stort antall korte fragmenter som etter sekvensering må pusles sammen til et fullstendig genom. Til dette finnes det flere programmer det er mulig å benytte seg av.

I denne oppgaven ble det undersøkt om ulike bioinformatiske verktøy har ulik effekt på det endelige fylogenetiske resultatet, i hovedsak i form av fylogenetiske trær. Det ble plukket ut tre programmer som forbehandler rådata-filene etter sekvensering (preprosessering): Quake, Timmomatic og Nesoni. Videre ble det plukket ut tre programmer som setter sammen de korte fragmentene til lengre sekvenser (assemblering): SPAdes, Velvet og Celera. Det ble gjennomført et faktorialt forsøk der alle tre nivåer av faktoren preprosessering er kombinert med alle nivåer av faktoren assemblering. Hvor vellykket resultatet hver kombinasjon ga, ble evaluert med sammenstilling mot referansegenom og N50.

For videre fylogenetiske analyser ble det brukt ”Multilocus Sequence Typing” (MLST) for bestemmelse av allelvariasjoner i utvalgte gener, og hvert bakterieisolat fikk en sekvenstype. Basert på variasjon mellom isolatene ble fylogeni kartlagt med ”neigbor joining” trær, og eBURST.

Abstract

In order to reveal what genes separates bacterial species and strains, a comparison of genomes are an effective method. The usual starting point with an unknown genome is the sequencing of the genetic material. The challenge is todays sequencing technology, which are unable to sequence the whole genome length, but cuts the DNA-molecules into shorter fragments before sequencing. The result of sequencing is a great number of short fragments that have to be put together in the right order to a complete genome. For this task there are several programs available.

In this thesis different bioinformatical tools were tested to se if they led to different phylogenetic results, mainly looking at phylogenetic trees. Three programs that process the raw data files after sequencing (preprocessing) was chosen: Quake, Trimmomatic and Nesoni. There was also chosen three programs that put the short fragments together into longer sequences (assembly): SPAdes, Velvet and Celera. The experiment was a factorial experiment where all three levels of the preprocessing factor were combined with all levels of the factor assembly. The results from each combination were evaluated using alignment to a reference genome and N50-values.

”Multilocus Sequencing Typing” (MLST) was used to determinate the isolates individual sequence type using variation in alleles in chosen genes. This was used in phylogenetic analyzes. Basted on the differences between the sequence types of the isolates, it was created neighbor joing trees and the program eBURST was used to cluster related isolates.

Innholdsfortegnelse

1. Innledning

1.1 Mikroorganismenes rolle

Mikroorganismer er organismer som består av en eller få celler, slik som bakterier og sopp. Disse organismene finnes over alt, og har opp gjennom hatt en stor betydning for jordens utvikling. Lenge dominerte de som eneste organismer, og flere av jordens naturlige prosesser lar seg gjennomføre på grunn av deres virke. Blant annet er nitrogenfikserende bakterier sentrale i nitrogenkretsløpet, og vi er avhengige av bakteriene i tarmene for en velfungerende fordøyelse. I dag har vi også lært å dra nytte av deres egenskaper både innen medisin, industri og forskning.

Det finnes et mangfold av bakteriearter og vi mennesker lever fra fødsel til død med bakterier både inni og rundt oss. De fleste er helt ufarlige, og mange er nødvendige for en normalt fungerende hverdag. Derimot finnes det også patogene bakterier. Patogene bakterier er bakterier som er i stand til å forårsake sykdom hos andre organismer. For å unngå store epidemier, og for å kunne forhindre, forebygge og kurere sykdom er det viktig med informasjon om de patogene bakteriene, og forstå hva som gjør dem patogene. Dette kan gjøres ved å undersøke deres genetiske kode, for å kartlegge hva som skiller de patogene fra de harmløse bakteriene (Tortora, 2010).

1.2 Den genetiske koden

Alle levende organismer er bærere av en genetisk kode som inneholder informasjon om organismens oppbygning, utseende og egenskaper. Denne genetiske koden finnes i arvestoffet, eller DNA-molekylene. Deoksyribonukleinsyre (DNA) har en struktur som minner om en stige kveilet rundt i en spiral, der de to langsgående ryggradene i stigen er kjeder bestående sukkermolekyler og fosfatgrupper. Trinnene i stigen sitter på sukkermolekylene og består av nitrogenholdige baser. I DNA finnes de fire basene adenin (A), tymin (T), cytosin (C) og guanin (G). Hvert trinn i stigen består av to baser i et par, og det er alltid de samme basene som danner et basepar. A danner alltid basepar med T, og C danner alltid med G. Vi sier at disse basene er komplementære. Monomeren i DNA-molekylet består av en base, et sukkermolekyl og en fosfatgruppe, og kalles et nukleotid.

Figur 1.1. Illustrasjon av DNA som viser molekylets oppbygning. (http://www.fhi.no/tema/gener-og-dna/dna-analyser-og-slektskap*)*

Til sammen kan tre baser utgjøre en kode som koder for en aminosyre, og en enkelttrådet DNA-sekvens koder dermed for en kjede med aminosyrer. Et gen på DNA-tråden er en sekvens som koder for et protein. Alt arvemateriale, både kodende og ikke-kodende kalles organismens genom. Hos dyr og planter er det kun en liten del av hele genomet som utgjør den kodende delen, mens hos eksempelvis bakterier er så å si alt kodende. Et lokus (fl. loci) er et spesifikt sted i genomet, og ulike varianter av baserekkefølgen i et lokus kalles alleler. Variasjon i alleler er det som skaper genetisk diversitet.
DNA er organisert i kromosomer. Mennesker har normalt 23 par kromosomer, til sammen 46, og kromosomene befinner seg i cellens kjerne. Bakterier er prokaryoter, noe som innebærer at deres arvemateriale ikke befinner seg inni en cellekjerne. De er encellede og har som regel kun ett kromosom, som inneholder informasjon om cellens oppbygning og funksjon. I tillegg til dette kromosomet kan bakterier ha plasmider. Plasmider er små sirkulære DNA-molekyler som inneholder genetisk informasjon. Denne informasjonen er ikke nødvendig for cellens overlevelse, men som kan gi den bakterien ekstra egenskaper, eksempelvis resistens mot antibiotika.

1.3 Slekten *Bacillus*

Bakterier som tilhører slekten *Bacillus* er stavformede grampositive bakterier med sporedannende egenskaper (Økstad, 2012). *Bacillus*-slekten er stor, da den rommer over 60 ulike arter med stor genetisk diversitet. Slekten er mye forsket på da den kan brukes i industrielle sammenhenger og grunnet dens egenskaper som modellorganisme. Den er også et interessant objekt innen forskning grunnet noen av artenes patogene egenskaper, og deres

opphav til menneskelige sykdommer (Økstad, 2011). Den mest kjente er miltbrann-bakterien, *Bacillus anthracis.*

1.3.1 *Bacillus cereus*

De fleste patogene *Bacillus* artene tilhører undergruppen *Bacillus cereus gruppen* (B. *cereus sensu lato)* som inkluderer *B. anthracis, B. cereus (sensu stricto), B. thuringiensis, B. mycoides, B. pseudomycoides og B. weihenstephanensis*. Av disse er det *B. anthracis, B. cereus (sensu stricto)* og *B. thuringiensis* som er mest studert. *B. thuringiensis* er en bakterie som kan forårsake infeksjoner hos insekter, og er derfor mye brukt kommersielt som insektdrepende midler. Den finnes vanligvis i jorden, og kan klassifiseres ved krystalltoksinene som produseres under sporedannelse. *B. cereus* er også primært en jordbakterie, og kan forårsake matforgiftning hos mennesker. *B. anthracis* er også en art med patogene egenskaper overfor pattedyr. Denne bakteriearten gir opphav til anthrax, eller miltbrann, som kan føre til infeksjoner i huden, mage/tarm, eller lunger. Hudanthrax er lett å kurere, mens de to sistnevnte er mer alvorlige, og det er påvist dødsfall hos mennesker etter inntak av dyr som har dødd av sykdommen (Økstad, 2011).

Flere arter av *B. cereus*-gruppen er svært utbredte og det er stor variasjon i naturlige habitater. Ettersom de er sporedannende bakterier har de evne til å overleve i vanskelige omgivelser med lavt næringsinnhold. De kan lett spres til mat og drikkevarer, som melkeprodukter, ris, og andre råvarer som er rike på karbohydrater og forårsake fordøyelsesinfeksjoner (Økstad, 2011).

Å undersøke slektskap i *B. cereus* gruppen er utfordrende av mange ulike grunner. Gruppen inneholder for det første stammer som er svært nært beslektet, og nesten umulig å skille fra hverandre. Nære slektskap mellom stammer kan også gå på tvers av art. Dette har spesielt blitt observert hos *B. cereus* og *B. thuringiensis,* der det er umulig å skille de to artene fra hverandre uavhengig av typingsmetodik. Generelt er genomene hos de ulike artene i *B. cereus* gruppen komplekse med tanke på at antall og størrelse på plasmidene varierer mellom stammene. Det spesielt interessante er at de genetiske forskjellene som er avgjørende for definering av artstilhørighet kan ligge nettopp i plasmidene. Hos *B. anthracis* ligger virulensegenskapene på de to plasmidene pXO1 og pXO2, som koder for toksiner og kapselelementer (Økstad, 2012). Den store variasjonen innen plasmidinnhold i gruppen indikerer at det gjennom evolusjonen har vært en utveksling av genetisk informasjon gjennom

horisontal genoverføring, det vil si overføring av arvestoff mellom organismer av samme generasjon (Tortora, 2010). Det kan være flere grunner til at nettopp *B. anthracis* har utviklet seg til en patogen bakterie, og en teori er at denne bakteriearten er et såkalt "hopeful monster". Det antas da at potensialet for virulens var tilstede hos opprinnelsesorganismen og at potensialet ble oppnådd ved en horisontal genoverføring av plasmidene pXO1 og pXO2, som inneholder de nødvendige genene. Dette ble bekreftet gjennom studier av Zwick *et al*. (2012).

1.4 16S rRNA hos bakterier

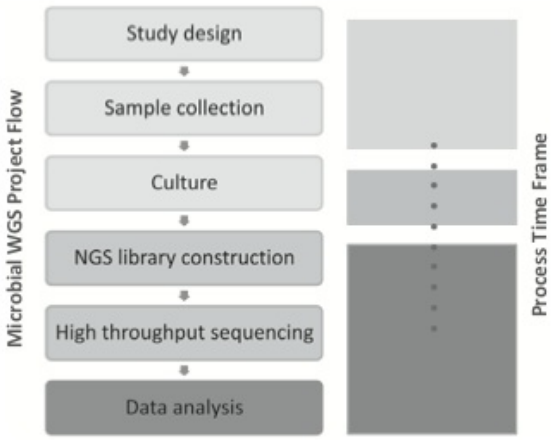
16S-genet koder for 16S rRNA, en komponent i den lille subenheten (30S) i organellen ribosomet hos prokayoter (organismer uten cellekjerne). Dette genet er mye brukt i identifisering og klassifisering av bakterier da alle kjente bakterier har genet samt at det er svært konservert. Det betyr at genet varierer lite fra art til art, men likevel så mye at det kan benyttes til å skille mange arter fra hverandre. Det evolusjonære slektskapet mellom arter kalles fylogeni, og avhengig av celletype kan organismer klassifiseres inn i et av de tre domenene *Eukarya*, *Bacteria*, eller *Archaea* (Tortora, 2010). Med 16S rRNA som markør i genomet er det mulig å klassifisere bakterier (Woese, 1987). En utfordring er derimot at 16S-genet ikke varierer tilstrekkelig mellom nært beslektede arter, slik at disse blir vanskelig å skille. Særlig vanskelig er det å skille organismer av samme art. For å kunne studere slektskap innad i nært beslektede arter, er det tatt i bruk nye metoder som benytter seg av flere molekylære markører i arvestoffet. Til grunn for metodene ligger teknologier som sekvensering og typebestemmelse.

1.5 Neste generasjons sekvensering

Det har lenge blitt brukt fenotypiske egenskaper hos organismer for å skille arter fra hverandre. Hos høyerestående arter kan en eksempelvis bruke tilstedeværelse av ryggrad i skjelettet for å klassifisere, mens det hos bakterier har vært vanlig å bruke gramfarging (Bartholomew, 1952). I løpet av de siste 20 årene har det imidlertid blitt mulig å kartlegge baserekkefølgen i organismers DNA. Dette kalles sekvensering, og det har blitt mulig å sammenlikne, ikke bare på utseende og egenskaper, men også det som ligger til grunn for dette. Med informasjon fra sekvensering kan gen-sekvensene brukes for å fastslå fylogenetiske sammenhenger (Bishop, 2014).

Sekvensering er en prosess som kartlegger baserekkefølgen i et gen eller et helt genom. Den første metoden brukt var Sanger sekvensering på 1970-tallet, der det ble brukt terminerende nukleotider for å danne DNA-tråder av ulik lengde. Disse ble deretter analysert med gelelektroforese for å få frem base-rekkefølgen i sekvensen (Sanger, 1977). Det har skjedd mye innen utvikling av sekvenseringsmetoder, og mer effektive og mindre kostbare metoder er tatt i bruk, såkalt "Next generation sequencing" (NGS). På markedet finnes flere plattformer som benytter seg av ulike teknologier, for eksempel Illumina MiSeq, Ion Torrent og PacBio RS II[c]. Det er mulig å sekvensere store menger DNA på kortere tid enn før, noe som resulterer i en økende mengde genetisk informasjon som må behandles. Det å kunne behandle informasjonen raskt, og å kunne utføre data analyser har som en følge av dette, blitt svært viktig.

Et sekvenseringsprosjekt for mikrobielle organismer kan organiseres i seks ulike deler i (figur 1.2). I trinnet som omhandler "sample collection" er det vesentlig å tenke gjennom hva som er formålet med prosjektet. Dersom målet er å kartlegge genetisk diversitet mellom de ulike isolatene, er det avgjørende med isolater fra ulike miljøer. Dette øker sannsynligheten for å fange opp variasjoner i gener og loci, da isolater tilpasset ulike miljø fører til ulike genetiske tilpasninger. Prosessen skjer ved påføring av DNA-løsning til apparatet, og det første som gjennomføres er en konstruering av NGS-bibliotek ("NGS library construction") av DNA-molekylene, som klargjør den kjemiske og fysiske prosessen i selve sekvenseringen. Siste steg i prosessen er data analysen. Til høyre i figuren vises en fremstilling av tidsbruken på de forskjellige stegene, der det kommer klart frem at denne siste analysen tar mye tid.



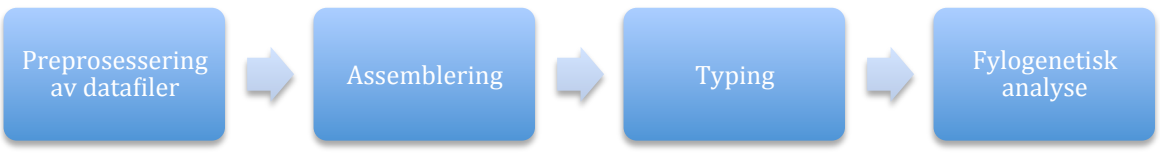
Figur 1.2. Organisering av sekvenseringsprosjekt for mikrobielle organismer (Bishop, 2014).

1.6 Data analysen

Resultatet av en sekvensering er store datafiler, gjerne med en størrelse på flere gigabytes. Filene inneholder blant annet en bokstavrekkefølge som representerer den avleste nukleotidsekvensen, samt en tilhørende score til hver bokstav, som indikerer kvaliteten på det avleste nukleotidet. Til tross for dagens gode sekvenseringsteknologi, er det ikke mulig å sekvensere mer enn noen hundre basepar før kvaliteten på avlesningen blir dårlig. Dette er grunnet kjemien og fysikken i instrumentene, og også avhengig av hvilken teknologi som brukes. Løsningen er å sekvensere korte fragmenter av genomet mange ganger, som senere kan samles sammen til et fullstendig genom. Disse korte fragmentene kalles ”reads”.

Prosessen fra datafiler med korte sekvenser til et resultat med biologisk betydning i form av komplette genom-sekvenser og fylogenetiske illustrasjoner, innebærer flere steg. I hvert steg er det flere valgmuligheter i form av innstillinger og programmer. Målet er å komme frem til et fornuftig resultat som samsvarer med virkeligheten, i form av korrekte sammensatte genomer og riktige fylogenetiske fremstillinger. Det kan da være avgjørende å ta de riktige valgene i hvert trinn. Hvilke metoder som benyttes og hvilke verdier som blir gitt til parametere, kan være med på å påvirke det endelige resultatet. Det er også essensielt at de valgene en tar i de tidlige stegene, kan påvirke kvaliteten i de påfølgende steg.

Stegene fra datafiler til fremstillinger med en overordnet biologisk betydning, som eksempelvis slektskap, kan illustreres som i figur 1.3. Disse innebærer en forbehandling av datafilene med korte DNA-sekvenser, en samling av alle sekvensfragmenter til fullstendig genom (assemblering), identifisering av ulike typer av en art (typing), og til slutt en fylogenetisk analyse.

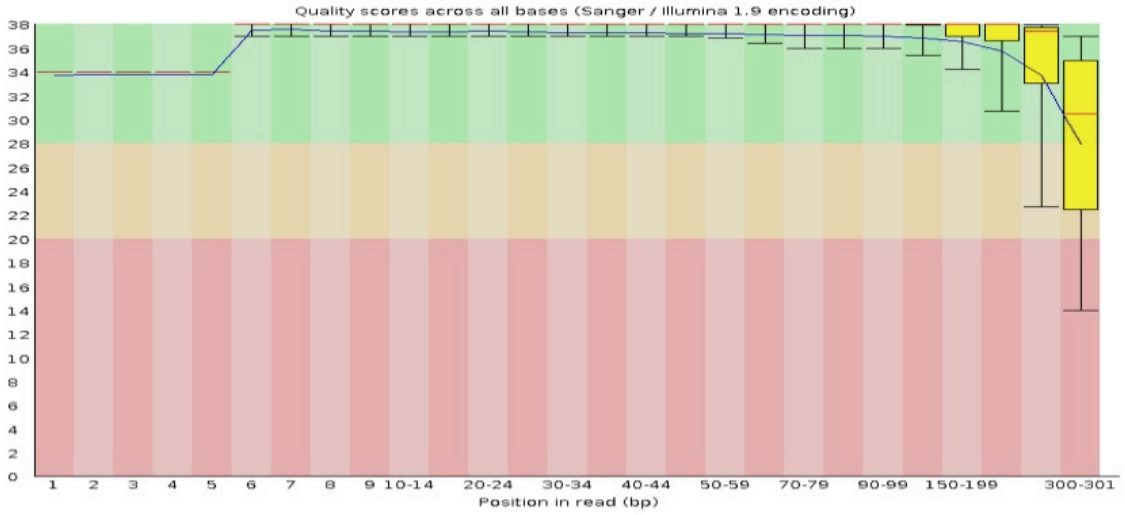


Figur 1.3. Oversikt over stegene i data analysen etter sekvensering av DNA.

1.6.1 Preprosessering av datafiler med read-sekvenser

Det er viktig for videre analyser at dataene med read-sekvenser som jobbes med er av god kvalitet. God kvalitet vil si få sekvenseringsfeil, noe som undersøkes i en preprosessering. De

nevnte resultat-filene fra sekvenseringen er *fastq*-filer, som i tillegg til bokstavsekvensen til nukleotidene inneholder respektive kvalitetsverdier for hver base. Ofte plottes disse verdiene i et Score-Quality(SQ)-plot, som viser en fordeling av basenes kvalitetsverdi over sekvensen. Et slik plot vises i figur 1.4. Her kommer det frem at basene i enden fragmentet får lavest verdi.



Figur 1.4. QS-plot for et sekvensert read på omtrent 300bp. Horisontalt til høyre er lengden på fragmentet, og verdien fastslås av den loddrette aksen. Grønt område indikerer høy kvalitetsverdi, gult en noe mer usikker, mens rødt indikerer en dårlig verdi.

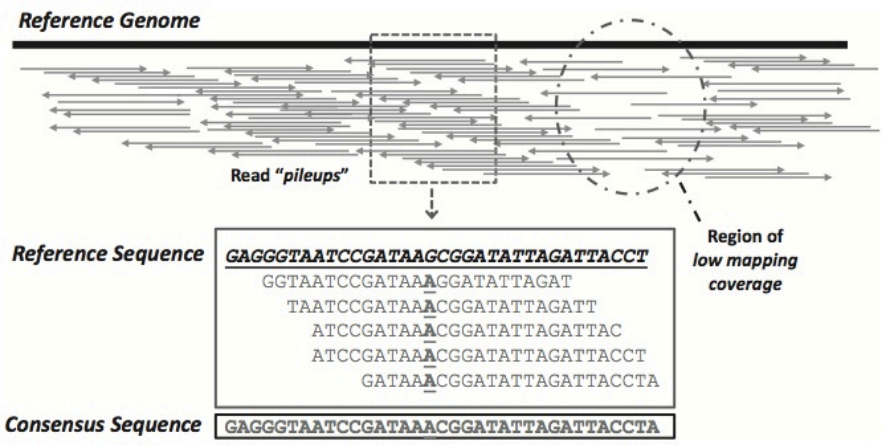
Verktøyene som utfører preprosesseringsjobben fjerner hele eller deler av reads med dårlig kvalitet, og prøver også å identifisere og fjerne sekvenseringsfeil. I tillegg fjernes sekvenser som ikke er en del av selve genomet, men som ble satt på DNA-tråden for at sekvenseringen skulle kunne gjennomføres, såkalte ”adapter-sekvenser”. Flere programmer finnes for å gjennomføre denne prosessen, og eksempler er og måten de operer på varierer, noe som også kan påvirke resultatet. Avgjørende for alle programmer som bearbeider slike datafiler, er at de ikke påvirker sekvensbitene i den grad at videre analyser blir påvirket i gal retning. Det er viktig at riktige sekvenser blir fjernet, og at sekvenser som tilhører i det opprinnelige genomet ikke blir gjernet. Målet er å trimme bort data av dårlig kvalitet, og samtidig ikke miste viktige opplysninger. Feil i preprosesseringen kan gi opphav til feilaktige resultater i form av eksempelvis et feil sammensatt genom.. Det er også optimalt om verktøyet brukt i preprosesseringen er fleksibelt, ettersom det i dag eksisterer flere sekvenseringsplattformer som leverer ulikt format på sine datafiler. Noen sekvenseringer leverer fra seg datafiler i med reads i par, det vil si reads sekvensert over et visst område i motsatt retning av hverandre med en kjent avstand mellom seg. Det er også ønskelig at parene blir opprettholdt gjennom denne bearbeidingen.

1.6.2 Kartlegging av reads og assemblering

Etter preprosessering, skal reads pusles sammen til lengre sekvenser, og i beste fall til et fullstendig genom. I første omgang samles reads til lengre sekvenser, såkalte ”contigs”, og så eventuelt til enda lengre sekvenser kalt ”scaffolds”. Også i dette steget er det mulig å velge blant mange programmer, slik at det også her kan forekomme varierende resultater, avhengig av hva slags verktøy som benyttes. Avhengig av hvilken sekvenseringsplattform som er benyttet, er det viktig å velge et program som er tilpasset den aktuelle plattformen.

Programmene som brukes til å samle reads til scaffolds, kan hovedsakelig skilles på om det benyttes et referansegenom som hjelpemiddel, eller om det gjøres kun med informasjonen som finnes sekvensfragmentene, det vil si uten en referanse. Da kalles metoden *de novo*, som betyr ”from the beginning”. Begge metoder har flere alternativer i valg av programvare.

Ved bruk av et referansegenom sammenstilles de korte reads mot referansen, og de vil dermed orienteres under referansegenomet der baserekkefølgen er lik (figur 1.4). Sekvenseringsplattformen kan ha sekvensert et område på DNA´et flere ganger, men oftest litt forskjøvet i forhold til hverandre. Bildet av sammenstillingen vil derfor fremstille de ulike read-sekvensene som multiple streker under en heltrukken linje som representerer referansegenomet (figur 1.4). Basert på sammenstillingen mot referansegenomet dannes det en lengre sammenhengende konsensus-sekvens.



Figur 1.4: Fremstilling av hvordan ”reads” mapper til et referansegenom, og hvordan en kommer frem til en konsensussekvens i enden ((Bishop, 2014).

Ved å gjennomføre assembleringen *de novo* blir genomet forsøkt rekonstruert basert kun på read-sekvensene. Dette kan gjøres på ulike måter, men metoden går ut på å bruke overlappende sekvenser i reads i mellom, til å sette dem sammen til lengre, kontinuerlige sekvenser (contigs). Med ”paired-end” sekvensering sekvenseres DNA-tråden fra begge sider. Produktet er to reads med en kjent avstand mellom seg, og disse to sies å være et par. Dersom reads fra et par befinner seg i to ulike contigs er det mulig å beytte seg av denne informasjonen til å danne lengre scaffold-sekvenser.

Fordelen med sammenstilling av reads mot et referansegenom er at det er relativt enkelt, da en allerede har et genom som utgangspunkt for puslespillet. Man har en tilnærmet fasit, og unngår dermed de store feilene. Ulempen er imidlertid at referansegenomet kan hindre identifisering av nye sekvenser som finnes i det ukjente genomet, og ikke i referansegenomet. En kan risikere å sitte igjen med sekvenser som ikke matcher referansegenomet, men som er viktige for den sekvenserte organismen. Dette vil altså medføre tap av informasjon. Med *de novo* assemblering vil en ha muligheten til å sette sammen reads uavhengig av en referanse, og vil dermed kunne inkorporere all tilgjengelig sekvens. Dette gjør at reads kan bli satt sammen måter som ikke er sett tidligere, noe som kan avdekke uoppdagede sekvensrekkefølger, og potensielt nye alleler. Ulempen er at jobben med å pusle sammen reads til lengre sekvenser blir større, tyngre og vanskeligere da en ikke har andre holdepunkter enn likheter mellom reads.

1.7 Typingsmetoder

Det å type innebærer å identifisere ulike typer av organismer innenfor en art, og kan dermed brukes til å kartlegge den genetiske variasjonen innenfor arten. Det å kunne skille ulike bakterietyper fra hverandre basert på typer eller subtyper, er viktig i studier av sykdomsutbredelse ettersom menneskelige patogener innenfor en art gjerne har svært ulike egenskaper basert på deres type. Det er gjerne ønskelig å kartlegge disse egenskaper for å hurtig kunne behandle sykdommen og på best mulig måte. Flere typingsmetoder er tilgjengelige, slik at det stilles krav til hvordan de skal kunne fungere på en tilfredsstillende måte. Det er viktig at typingsmetoden er kapabel til å typebestemme alle isolater, og at den må kunne skille på isolater som ikke er relaterte. Den må også kunne skille tett beslektede isolater. Metoden må kunne reproduseres, og det er en fordel med lav kostnad og at metoden

er rask og enkel å bruke. I tillegg er det ideelt om dataene som registreres kan deles og er overførbare mellom ulike systemer, og at de er lett tilgjengelige i en felles database. En standard nomenklatur er da å foretrekke i metoden (Sabat, 2013).

Tidligere har det vært vanlig å bruke typingsmetoder som baserer seg på fenotypiske egenskaper, slik som for eksempel serotype (bruker et serum for bestemmelse) og paghe-type (bruk av virus for bestemmelse) (Sabat, 2013). Dette er metoder som kan gi misledende resultater, da fenotypiske egenskaper kun gir en indikasjon på genotypen, men kan like gjerne være et resultat av miljø. Nyere sekvenseringsteknologi har gjort det mulig å gjøre studier på selve genomet til bakteriene. Dette er molekylære typingsmetoder som bruker DNA-sekvensen som utgangspunkt. Med bakgrunn i forskjeller og likheter i DNA-sekvensen (allelene) er det muligheter for å skille bakterieisolatene fra hverandre. Når en genvariant til et isolat er fastslått, er det mulig å sammenlikne allelvariantene til isolatene, og med dette bestemme den fylogenetiske sammenhengen mellom isolatene. Tidligere har det blitt brukt metoder som pulse-felt gel elektroforese (PFGE) og multilocus enzyme elektroforese der for eksempel metabolske enzymer skilles på bakgrunn av elektroforetiske mobiliteter (Maiden, 1998). Variasjonen i enzymenes mobilitet elektroforesen danner et mønster som vil utgjøre isolatets elektroforetiske type. Dette mønsteret kan sammenlignes med mønsteret til andre isolater, og dermed får man en høyoppløselig sammenligning av ulike isolater. Dette konseptet er videreutviklet i multilocus sequencing typing (MLST) der det ikke lenger er enzymene som skilles på bakgrunn av fysiske egenskaper, men det kartlegges forskjeller i DNA-kodene som ligger til grunn for enzymene (Maiden, 1998).

1.8 Fylogenetisk analyse basert på helgenomsekvensering

Fylogenetiske analyser basert på helgenomsekvensering gjøres vanligvis for å kartlegge slektskap mellom organismene, ved å studere hvilke gener som deles mellom arter og hvilke gener som skiller dem fra hverandre. Det er også av interesse å finne ut hvilke stammer som til sammen utgjør en art, og på hvilke områder stammene er forskjellige innad i arten. For å gjøre slike sammenlikninger er man avhengig av gode sekvenseringsteknikker. Dette er løst med NGS, og det legges mye arbeid i å kunne sette sammen så fullstendige genomer som mulig. For å gjøre fylogenetiske undersøkelser, er en heldigvis ikke avhengig av å ha komplette sekvenser, men det holder med contigs eller scaffolds som dannes under en assemblering. Fylogenien eller slektskapet til en samling organismer, fremstilles gjerne i et

fylogenetisk tre, med et utgangspunkt i en felles stamfar. Det er også mulig å gjøre grupperinger der en legger organismer som er nært beslektet nær hverandre i grupper.

1.9 Problemstilling

I denne oppgaven skal betydningen av valg verktøy og parametere i de tidlige stegene i data analysen undersøkes. Effekten av preprosesseringsverktøy i kombinasjon med valg av assembleringsmetode skal undersøkes og evalueres. De tre preprosesseringsprogrammene Quake, Trimmomatic og Nesoni i kombinasjon med de tre assembleringsprogrammene SPAdes, Velvet og Celera skal sammenliknes. Preprosesseringsprogrammene har en noe forskjellig tilnærming til hvordan fjerning av baser og sekvenser skal gjennomføres, mens assembleringsprogrammene bruker i alle hovedsak forskjellige algoritmer. Vil det være effekt av kombinasjon av valgte programmer?

Videre skal MLST-markører brukes til å beregne avstand mellom et sett av bakterieisolater, og det skal bygges fylogenetiske trær på bakgrunn av disse. Vil det være forskjell i trærne avhengig av hvilke verktøy som er benyttet? Er metodene i hvert steg de ideelle for å undersøke slike datasett?

2 Materialer og Metoder

2.1 Datasett

Data brukt i denne oppgaven er hovedsakelig sekvenser fra *Bacillus cereus* gruppen, mottatt fra Forsvarets Forskningsinstitutt (FFI). Isolatene er hentet fra forskjellige miljøer samt fra ulike geografiske områder. En oversikt over hvilke isolater datasettet inneholder, er å finne i tabell 2.1. Datasettet består av 24 isolater som alle er sekvensert på Illumina Miseq, med en read length på 300bp, og coverage på 50-100. Sekvenseringen ble gjennomført paired end, slik at det for hvert isolat ble dannet to sekvens-filer, en for forover-sekvensen og en for revers-sekvensen. I tillegg ble det ferdig assemblerte genomer fra NCBI inkludert i defylogenetiske analysene for å få et større perspektiv på hvor i gen-treet de 24 isolatene hørte til.

Tabell 2.1. Oversikt over isolatene analysert i oppgaven. Art og stamme er gitt, og der navnet er cereus/thuringiensis er det ikke klart å bestemme hvilken art det er. Sekvensmengde tilsvarer antall reads i rådatasettet.

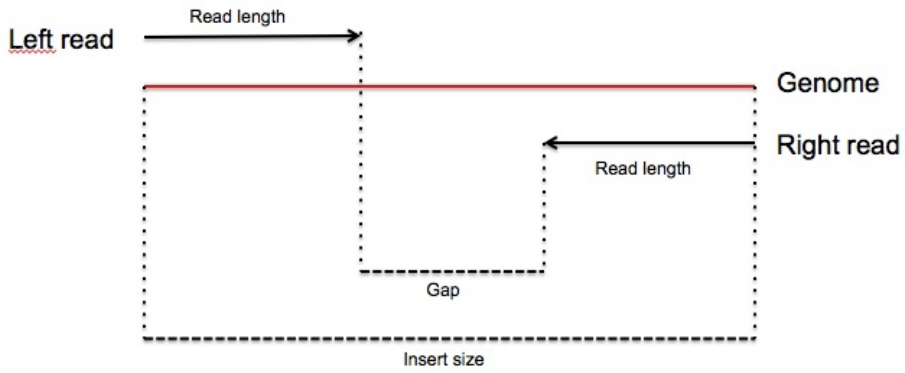
Art	**Stamme**	**Sekvensmengde (ant reads)**
B. cereus	2000031002	1 224 354
B. cereus	2002734520	3 769 352
B. cereus	ATCC14579	1 198 052
B. cereus	B275	1 230 658
B. cereus	BGSC6E1	1 103 300
B. cereus	DSM318	1 533 006
B. cereus	DSM336	1 280 380
B. cereus	NVH0597-99	1 512 456
B. cereus	NVH1518-99	1 240 146
B. cereus/thuringiensis	FFIBCgr36	1 391 996
B. cereus/thuringiensis	FFIBCgr44	1 729 422
B. cereus/thuringiensis	FFIBCgr46	1 298 280
B. cereus/thuringiensis	FFIBCgr113	1 222 116
B. anthracis	FFIBCgr114	1 518 190
B. cereus/thuringiensis	FFIBCgr115	1 450 010
B. cereus/thuringiensis	FFIBCgr116	1 883 970
B. cereus/thuringiensis	FFIBCgr119	4 179 232
B. cereus/thuringiensis	FFIBCgr121	1 574 284
B. cereus/thuringiensis	BGSC4AJ1	1 230 658
B. cereus/thuringiensis	BGSC4AS1	871 822
B. cereus/thuringiensis	BGSC4AU1	1 211 346
B. cereus/thuringiensis	BGSC4AY1	1 330 214
B. cereus/thuringiensis	BGSC4BA1	1 801 156
B. cereus/thuringiensis	BGSC4CC1	1 462 612

2.2 Illumina Miseq

Blant flere av dagens NGS teknologier finnes Illumina. Illumina bruker ”sequencing by synthesis” for å kartlegge baserekkefølgen. Det vil si at sekvenseringen skjer ved en syntese av DNA-tråden, og benytter seg av fluorescerende nukleotider for å kartlegge . Først må

DNA-sekvensene bearbeides noe, det må fragmenteres og settes på adapter-sekvenser; det blir dannet et ”NGS-bibliotek”. Alle sekvenser får satt på universelle adaptere som er komplementære til en slags anker-sekvens. Denne anker-sekvensen sitter på en glassoverflate. På denne måten kan sekvensene festes i ankeret. DNA-molekylene amplifiseres ved at den frie enden av DNA-molekylet festes til en ny adaptersekvens som også er festet på glassplaten, og fra primer-adapteren blir tråden utvidet til en hel dobbelttrådet DNA-sekvens. Videre denatureres sekvensen, og steget gjentas på nytt til det etter hvert dannes store grupper av enkelttrådet DNA. Selve sekvenseringen utføres ved at fluorescensmodifiserte nukleotider tilføres, og det komplementære nukleotidet til enkelttråden inkorporeres på adaptersekvensen. Hvert nukleotid har hver sin farge. Dette nukleotidet terminerer også videre utvidelse av tråden. Nukleotidene som ikke er inkorporert vaskes bort, og det registreres hvilket nukleotid som er satt på. Den terminerende delen av det inkorporerte nukleotidet fjernes, og prosessen gjentas til trådens ende. Fordelen ved denne typen sekvensering er at det vanligvis registreres færre feil vedrørende bestemmelsen av hvilket nukleotid som blir inkorporert, såkalt base gjenkjenningsfeil. Ulempen derimot, sammenliknet med andre teknologier, er at sannsynligheten for substitusjonsfeil, det å utvide tråden med feil nukleotid, er høyere.

Resultatet av en Illumina Miseq sekvensering er et stort sett av korte reads-sekvenser på omtrent 250-300bp i lengde. I datasettet i denne oppgaven hadde alle reads en lengde på 301bp. Dette er paired-end reads, der det produseres to korte reads, med informasjon om en eventuell avstand mellom dem, et gap, som vist i figur 2.2.1



Figur 2.2.1: Sekvensering med paired end av genomet (i rødt), der en får et par med reads, såkalte mate-pairs.

2.3 Preprosessering

Feil i sekvenseringen kan ha betydning for resultatet av assemblering av reads. Assembleringer bruker overlappende sekvenser i reads til å lage lengre fragmenter, slik at feil

i sekvenseringen også kan introdusere feil i assembleringen. Feil som fører til manglende overlapp der det egentlig skulle ha vært innført, kan forårsake åpne rom (gaps) i det assemblerte genomet, mens feil som fører til falske overlapp, kan resultere i utvetydige sammensetninger av reads eller feilaktige koblinger. En viktig faktor i korreksjon og fjerning av baser er ”coverage”, som er det gjennomsnittlige antall ganger en enkelt base er avlest i en sekvenseringsrunde. Coverage kan beregnes, som i likning (2.1)

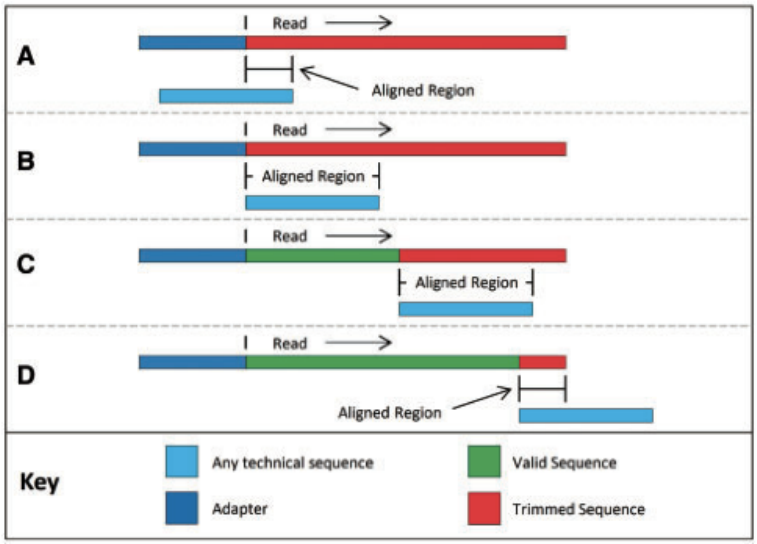
$$C = LN/G \qquad (2.1)$$

der C er coverage-verdien, L er gjennomsnittlig lengde på reads, N er antall reads og G representerer total lengde på genomet. Det optimale er en dyp sekvensering, der coverage-verdien er høy for alle baseplasser. Kvaliteten på hver enkelt baseavlesning er gitt i en phred-score. Phred-scorene er et symbol gitt til hver base, der hvert symbol har en tallverdi. Disse symbolene kan være både tall og store og små bokstaver med hver sin verdi. Det er disse phred-scorene som er utgangspunktet for utlukingen av baser og sekvenser med lav kvalitet. Etter preprosessering vil en sitte igjen med et sett av reads. Antallet kan ha blitt noe redusert, noen av read-sekvensene kan være kortere enn andre, og alle er ikke nødvendigvis paired-end lenger.

Dette steget i data analysen ble gjennomført med tre forskjellige preprosesseringsprogrammer; Trimmomatic, Nesoni og Quake.

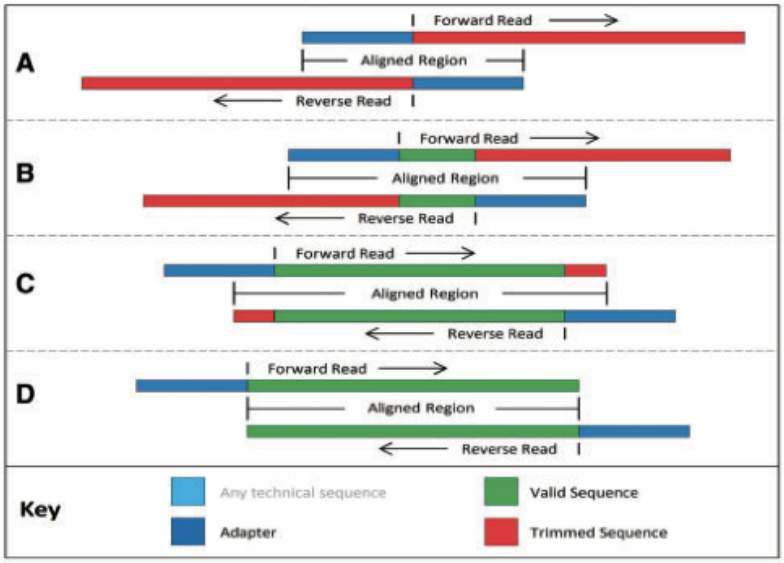
2.3.1 Trimmomatic

Trimmomatic har flere funksjoner i sin bearbeidelse av reads. Blant annet identifiserer og fjerner den det man kan kalle tekniske sekvenser, som for eksempel rester av adaptersekvenser. Dette er avgjørende da de ikke er en del av genomet og kan introdusere feil i videre analyser. Trimmomatic har to fremgangsmåter som kan brukes for å fjerne tekniske sekvenser; ”simple mode” og ”palindrome mode”. Simple mode benytter seg av en tilnærmet match mellom read og en allerede eksisterende teknisk sekvens som leveres av bruker. Alle reads sammenstilles etter den tekniske sekvensen, og ved et treff fjernes read-fragmentet, se figur 2.3.1.



Figur 2.3.1: Figuren illustrerer simple mode adaptertrimming med Trimmomatic. A: noe overlapp med teknisk sekvens i starten av read, og hele readet kastes. B: hel overlapp med teknisk sekvens i starten av read, og hele readet kastes. C: overlapp med hel teknisk sekvens ut i read, resterende sekvens kastes. D: noe overlapp med teknisk sekvens i slutten av read, denne delen av readet kastes. (Bolger et al., 2014)

Palindrome mode bruker read par som utgangspunkt for sammenlikning, og det faktum at par vil ha samme antall read-baser samt en eventuell kontaminering av adaptersekvensen i enden. Adapterne vil da være omvendte av hverandre, og baserekkefølgen til read-parene vil være omvendt komplementær og vil matche ved en sammenstilling. Adapter-sekvensene plasseres foran alle reads, og de sammenstilles. En sammenstilling med en revers komplementær overlapp vil gi sammenstillingen en høy score, og alle sammenstillinger som overstiger en øvre grense fører til trimming. Testen utføres helt til enden av readet, som vist i figur 2.3.2. Det er gitt at palindrome mode kun kan brukes på paired-end data. (Bolger et al., 2014)



Figur 2.3.2: Figuren illustrer palindrome mode adaptertrimming med Trimmomatic. A: overlap mellom adapter og starten av motsatt read, og hele readen kastes. B: overlapp med adapter noe ut i read, og resterende read-sekvens kastes. C: siste del av read overlapper noe med adapter-sekvensen og denne delen av readet kastes. D: ingen overlapp med adapter og trimmingen er fullført.

Trimmomatic har en ”glidende vindu” kvalitetssikring (”sliding window quality filtering”) som fjerner sekvenser med lav kvalitet. Reads gjennomgås fra den ene enden til den andre, og dersom gjennomsnittlig kvalitetsscore til en gruppe baser faller under en satt grense, trimmes resterende del av read-et bort.

Korte reads er ofte lite informativt da disse ved en tilfeldighet kan forekomme flere steder i sekvensen. Likevel kan det å beholde lengden til et read også være skadelig da sannsynligheten for sekvenseringsfeil øker desto lengre sekvensen er. For å finne en balanse bruker Trimmomatic ”maksimum informasjon” kvalitetssikring (”Maximum Information quality filtering”) der kvalitetskravet blir strengere utover i hvert read. Tre verdier for tre faktorer kombineres for å finne den perfekte lengden på et read; lengdegrense (”length treshold”) (LT) som bestemmes av en gitt mållengde (t) samt en antatt gjenværende readlengde (l), ”coverage” (Cov) som er bevart readlengde, og feilrate (”error rate”) (Err) der sannsynligheten for feil fra readets kvalitetsverdi blir brukt for å finne akkumulert sannsynlighet av feil. Beregningene av scoren til de tre faktorene finnes henholdsvis i likning (2.2), (2.3) og (2.4).

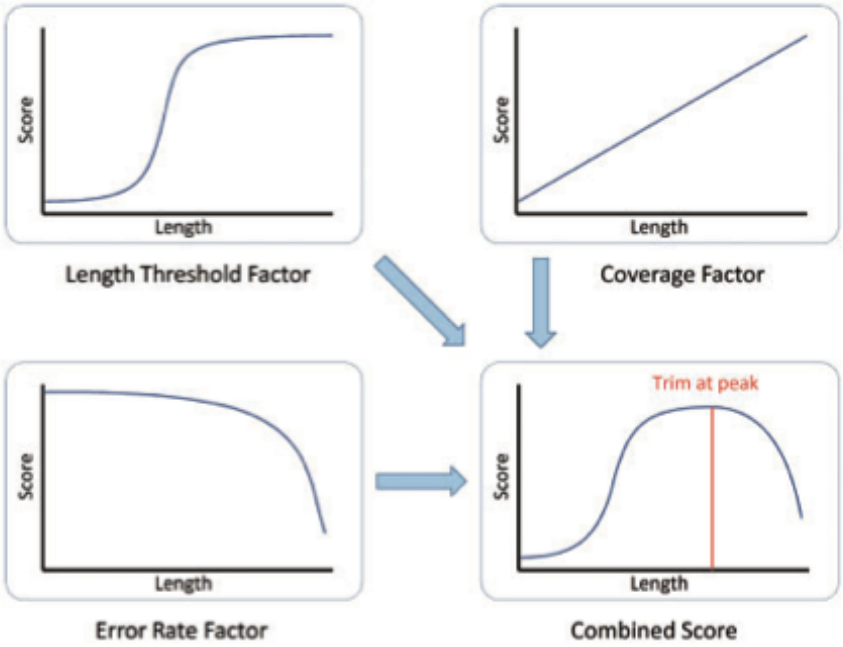
$$\text{Score}_{LT}(l) = \frac{1}{1+e^{t-l}} \qquad (2.2)$$

$$\text{Score}_{Cov}(l) = l \qquad (2.3)$$

$$\text{Score}_{Err}(l) = \prod_{i=l}^{l} P_{Corr}[i] \qquad (2.4)$$

De tre faktorene kombineres til likning (2.5), der det settes en verdi s av bruker som kontrollerer balansen mellom coverage (s=0 gir maks) og feilrate (s=1 gir maks). Kombinasjonen kan fremstilles grafisk som vist i figur 2.3.3.

$$\text{Score}(l) = \frac{1}{1+ e^{t-l}} * e^{l-s} * (\prod_{i=l}^{l} P_{Corr}[i])^{s} \qquad (2.5)$$



Figur 2.3.3: Grafisk fremstilling av de tre funksjonene som beregner score for length treshold, coverage og error rate, samt en kombinasjon av de tre. s er grensen for trimming og vises i plotet nederst til høyre med en oransje linje.(Bolger et al., 2014)

2.3.2 Quake

Preprosesseringsprogrammet Quake tar i bruk coverage fra sekvenseringen og kombinerer den med maksimal sannsynlighet for kvalitetsverdier, og rater for feilavlesninger av nukleotider til å oppdage og korrigere for sekvenseringsfeil. Quake deler opp reads i kortere sekvenser, k-mers. K-mers med høy coverage refereres til som pålitelige, da de mest

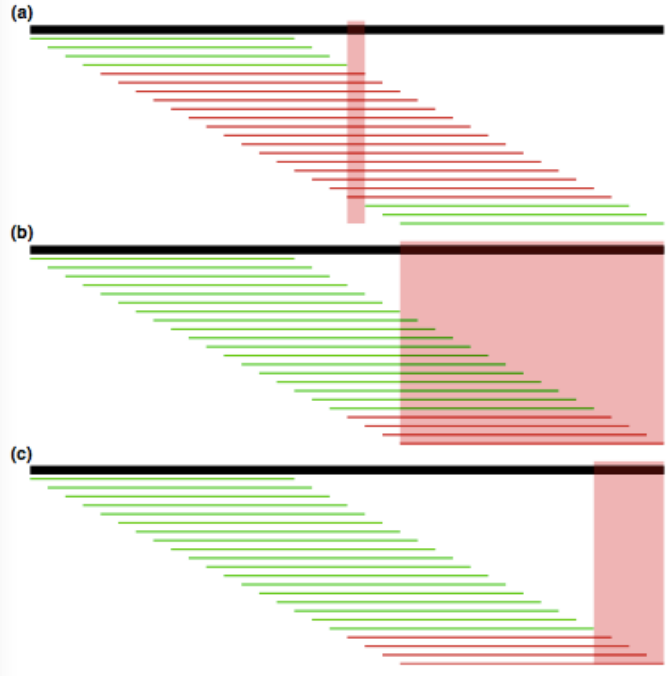
sannsynlig er å finne i genomet. Man kan kalle k-mers med lav coverage for upålitelige i denne sammenhengen.

Valg av k-verdi er essensiell og må gjøres med omhu. En lav verdi av k vil medføre at algoritmen kjøres hurtig. Det er også lettere å finne feil ettersom sammenlikning av korte sekvenser gjør det lettere å utpeke forskjeller. Med en *for* kort k-verdi derimot, vil sannsynligheten for å finne sekvensen igjen et annet sted i genomet ved en tilfeldig nukleotidsubstitusjon øke, noe som fører til forvirringer i søket etter feilsekvensering. En generell anbefaling er å sette en k-verdi slik at likning (2.6) tilfredsstilles, der G er størrelsen på genomet. For *Bacillus,* som har en genomstørrelse på omtrent 5,5millioner basepar, blir k-verdien tilnærmet lik 15. Opptelling av antall forekomster av alle k-mers er det første som skjer i algoritmen, og k-mer coverage blir dermed kartlagt.

$$k \cong \log_4 200G \qquad (2.6)$$

I sin algoritme implementerer Quake en spesiell metode for å bedre kunne skille mellom sanne og feilaktige k-mers, der en med sanne k-mers mener riktig sekvenserte k-mers. Sanne k-mers med lav forekomst og feilaktige k-mers med høy forekomst kan likne hverandre i verdi i k-mer-coverage. Derimot regner man med at de vil ha forskjellig kvalitet i baseavlesning, og dermed brukes kvalitetsscoren til å utføre separeringen. Kvalitetsverdien, som for hvert nukleotid angir sannsynligheten for at rett base er avlest i sekvenseringen, blir brukt for å beregne et sannsynlighetsprodukt, slik at snarere enn å øke k-merens coverage med én for hver observasjon av den gitte sekvensen, økes coverage med sannsynlighetsproduktet. Dette er hva Quake kaller ”q-mer-telling”. Q-mer telling brukes videre for å finne en coverage-grense som skiller sanne k-mers fra feilaktige. Ved hjelp av et histogram dannes to fordelinger med coverage tilhørende de to gruppene. Fordelingene vil gå i hverandre, og målet er å finne en fornuftig coverage-verdi som skiller dem. Dette gjøres ved å beregne sannsynlighetsraten for at en k-mer ved en gitt coverage tilhører den ene fordelingen eller den andre. Grensen for tilhørighet i de to gruppene settes med en tilpasning til formålet med videre analyser. Eksempelvis er det i *de novo* assemblering vesentlig at det å kaste sanne k-mers med lav coverage kan medføre gaps og feil-assembleringer rundt repetitive sekvenser, slik at det kan være en fordel med en mindre streng grense. Alle reads som inneholder k-mers som defineres som usikre i forhold til den nevnte grenseverdien, er utsatt for korreksjon.

En usikker k-mer med en potensiell substitusjonsfeil vil kun angi et område på read-et der feilen befinner seg, slik at det fremdeles vil være usikkerhet tilknyttet eksakt lokasjon. Ved å bruke alle k-mers klassifisert som usikre i read-et og finne krysningspunktet mellom dem, er det mulig å lokalisere regionen der feilen mest sannsynlig befinner seg. Dette kan illustreres i figur 2.3.4a. Ved lokalisering av usikre k-mers nær enden av read-et må det legges til grunn at det ofte forekommer flere sekvenseringsfeil nær enden, og den usikre regionen utvides, som vist i figur 2.3.4b. Ettersom en gjerne går ut fra at de fleste sekvenseringsfeil forekommer nær enden av et read, kan det antas at k-merene lengst til høyre inneholder feilen, og kun trimme bort denne delen av sekvensen, som vist i figur 2.3.4c. Sistnevnte gjennomføres først, og dersom det trengs videre trimming, gjennomføres prosessen i 2.3.4b.



Figur 2.3.4: k-mers definert som sanne (grønne) og usikre (røde) sammenstilles mot readet, og det feilaktige området lokaliseres. (a) området med feil lokaliseres gjennom krysningspunktet mellom usikre reads, her markert med søylen. (b) nær enden av readet er det funnet en feil, og sekvensen frem til endes vurderes som usikker. (c) grunnet at det typisk er feil i enden, antas det først at det er basene her som er feil, og disse trimmes bort.

Når regionen som trolig inneholder sekvenseringsfeil er lokalisert brukes sannsynlighetsberegning for å korrigere i read-et, slik at alle overlappende k-mers i området stemmer. En sannsynlighet for et sett av korreksjoner beregnes først, og med Bayes teorem og en antakelse om uavhengighet i sekvenseringsfeilene kan sannsynligheten beregnes som gitt i

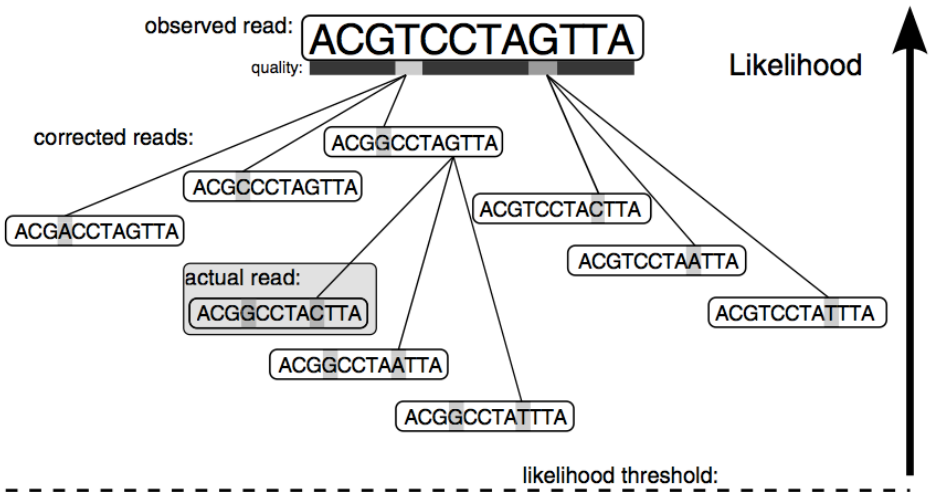
likning (2.7). B = B_1, B_2, …, B_N utgjør observerte nukleotider i readet, og A = A_1, A_2, …, A_N er de faktiske nukleotidene på det sekvenserte DNA-fragmentet. Gitt det observerte nukleotidet finnes sannsynligheten for at det tildeles basebokstaven A.

$$P(A=a \mid B=b) = \prod_{i=1}^{N} \frac{P(B_i=b_i \mid A_i=a_i)P(A_i=a_i)}{P(B_i=b_i)} \qquad (2.7)$$

$P(B_i = b_i)$ vil være den samme for alle baser i et read, og ettersom reads evalueres enkeltvis kan denne ses bort fra. $P(A_i = a_i)$ er definert som GC% til genomet. Den betingede sannsynligheten kan oppsummeres i likning (2.8), der $p_i = 1\text{-}10^{\frac{-q_i}{10}}$ er sannsynligheten for at det avleste nukleotidet i posisjon i stemmer og q_i tilsvarer nukleotidets kvalitetsverdi. Eq(x,y) er en matrise med sannsynlighetene for at base y er kalt (sekvensert/avlest) for nukleotid x med en kvalitet q, gitt at det har forekommet en sekvenseringsfeil.

$$P(B_i = b_i \mid A_i = a_i) = \begin{cases} p_i & hvis\ b_i=a_i \\ (1-p_i)E_{q_i}(a_i, b_i) & ellers \end{cases} \qquad (2.8)$$

Avslutningsvis bruker Quake videre sannsynlighetsberegning for å søke etter et sett med korreksjoner som gjør at alle overlappende k-mers stemmer overens. Dette kan fremstilles som et tre, som i figur 2.3.5, der det søkes etter korreksjoner som endrer det observerte readet med feil slik at det det blir riktig. Her representerer nodene i treet mulige korrigerte reads, mens grenene representerer korreksjonene. Hver node kan tildeles en sannsynlighet, og det settes en gitt sannsynlighetsgrense som antall korreksjoner ikke kan overstige(Kelley et al., 2010).



Figur 2.3.5: Illustasjon som viser søket etter rett read-sekvens fremstilt som et søk gjennom et tre med noder og grener.(Kelley et al., 2010)

2.3.3 Nesoni Clip

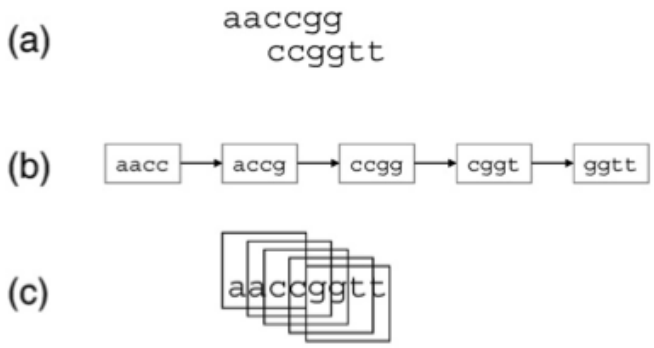
Nesoni er et program utviklet for å prosessere data produsert av NGS-teknikker. En modul av Nesoni er Nesoni Clip som utfører en preprosessering av reads før assemblering. Nesoni Clip trimmer vekk eventuelle adaptersekvenser og fjerner baser av dårlig kvalitet. Adaptersekvenser for Illumina importeres automatisk dersom andre sekvenser ikke blir gitt manuelt, og det søkes gjennom alle reads etter disse sekvensene. Deretter gjennomføres et søk etter baser med lav kvalitet, der det på forhånd er satt en nedre grense for trimming. Baser med lav kvalitet fjernes uten videre vurdering. Programmet er lite dokumentert, og fremdeles under utvikling.

2.4 Assemblering

For å sette sammen reads til lengre contigs og scaffolds ble det brukt tre ulike *de novo* assembleringsprogrammer. Den store forskjellen i programmene er typen algoritme som blir brukt, og her deles det inn i to klasser: overlap-layout-consensus (OLC) og de-brujin-graph (DBG). Felles for dem begge er at de bruker overlappende informasjon i reads. Assembleringsverktøyene som ble brukt her var SPAdes (http://bioinf.spbau.ru/spades), Velvet (https://www.ebi.ac.uk/~zerbino/velvet/) og Celera (http://wgs-assembler.sourceforge.net/wiki/index.php?title=Main_Page).

2.4.1 Graf-dannelse

Grafer brukes ofte i assembleringer. En graf er et sett av noder med kanter mellom seg, der hver trukne linje representerer en forbindelse mellom de to nodene. Er kantene piler kan en sti med en retning formes i grafen. I assemblering er overlappende sekvenser en kilde til forbindelse, og to sekvenser med en overlappende del kunne tilsvart to noder med en link mellom seg (Miller et al., 2010). Et eksempel vises i figur 2.4.1. Ideelt ville man i en assemblering kun hatt en forbindelse mellom hver node, og det ville vært mulig å sette sammen hele sekvensen i genomet kun ved å følge stien gjennom grafen.



Figur 2.4.1. Eksempel på grafdannelse i assemblering. To overlappende sekvenser (a) som ved hjelp av noder og piler (b) kan settes sammen til en fullstendig sekvens (c). (Miller et al., 2010)

2.4.2 De Brujin Graf (DBG)

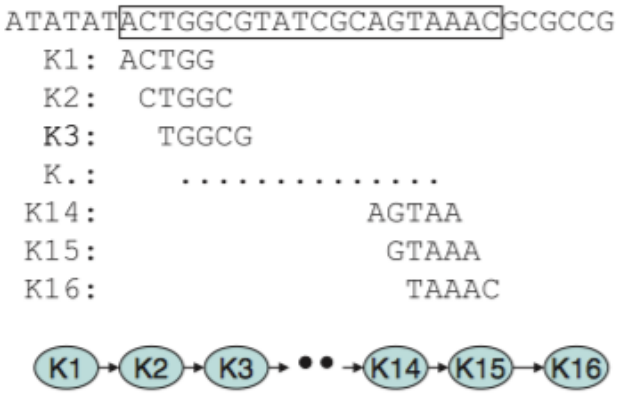
Denne algoritmen forsøker å samle alle reads ved først å kutte dem i enda kortere sekvenser, kalt ”*k-mers*”. K gir lengden på sekvensen og er en parameter som kan varieres og settes manuelt. Disse k-mers blir igjen stykket opp i k-1-mers, der k-mers stykkes opp i en høyre og venstre del, som vist i figur 2.4.2.

Sekvens:	AGTGTTCA
k-mers, k=4:	AGTG, GTGT, TGTT, GTTC, TTCA
k-1-mers:	AGT & GTG, GTG & TGT, TGT & GTT, GTT & TTC, TTC & TCA

Figur 2.4.2: Illustrasjon av hvordan DBG deler opp sekvenser i k-mers og k-1-mers

En de brujin graf opererer med en såkalt ”hash tabell” (Miller et al., 2010), som er en organisert oversikt og oppsummering av alle k-mers. Hver k-mer får en indeks i tabellen, og gjennom en hash-funksjon får hver unike k-mer et tall. Dermed blir det enkelt å slå opp i tabellen dersom det skal undersøkes om liknende element har blitt oppdaget før. Hash-tabeller er gode dersom man er ute etter raske oppslag, og her gir den assembleringsmetoden en fordel

i at algoritmen ikke behøver noen verktøy for å sammenstille alle sekvenser med hverandre. Ulempen er at slike tabeller bruker mye minne.
I grafen tilsvarer hver node en k-1-mer, og kanter med retning trekkes mellom dem basert på de overlappende sekvensene. Pilene indikerer et naboforhold mellom k-mers, og reads som inneholder identiske k-mers settes sammen til lengre contig-sekvenser, som vist i figur 2.4.3.



Figur 2.4.3: Enkel oppbygning av DBG; reads kuttet i k-mers og det dannes en sti mellom dem som til sammen utgjør en lengre sekvens. (Li et al., 2012)

Det er flere utfordringer ved denne metoden, og flere faktorer som gjør at assembleringen blir mer komplisert. Blant flere ting inneholder DNA sekvenser som gjentas, såkalte repeterende områder. I en DBG vil slike repeats føre til en sammenslåing av to stier i grafen, slik at det dannes ”bobler”, det vil si at grafen ikke kun følger en rett linje, men kan deles og ved en senere node igjen slåes sammen. Det blir da uvisst hvilken av stiene grafen bør ”velge” som sin vei. Palindromsekvenser er sekvenser som er identiske uavhengig av hvilken vei den leses, og disse vil folde tilbake på seg selv i en slik graf. Dersom sekvensen inneholder feil vil dette kunne gi en deling av stien i grafen, og det vil dannes utstikkere som ikke fører noen vei videre. Dette medfører huller i assemblert genom, og oppstykkede sekvenser. Med et perfekt genom vil man kunne lage en graf med en vei som er innom alle kanter mellom nodene kun en gang. Dette kalles for en Euler-sti. Med naturlige DNA-sekvenser blir dette svært vanskelig med sekvenseringsmetodenes mange feil, slik at målet vil derfor være å komme frem til stien som med høyest sannsynlighet er riktig. Den store fordelen med denne algoritmen er at den bruker kort tid.

2.4.2.1 SPAdes

Et av programmene brukt til assemblering i denne oppgaven er SPAdes assembler. Den bruker DBG-algoritmen. Programmet ble kjørt med standard innstillinger, der det er de to modulene BayesHammer, som korrigerer feil på reads, og SPAdes som iterativt assemblerer reads i genomet. SPAdes går gjennom følgende steg:

1) Graf-konstruksjon
2) Justeringer og beregninger i forhold til avstand mellom parede k-mers i read-par.
3) Ny grafdannelse med hensyn til read-par.
4) Konstruksjon av contigs

(Bankevich et al., 2012)

SPAdes assemblerer reads i flere omganger med k-mers med ulik k-verdi, og velger avslutningsvis den k-meren som ga best resultat i form av færrest contigs, til å danne scaffolds. Verdier av K velges automatisk basert på read-lengden og typen datasett (SPAdes, 2014).

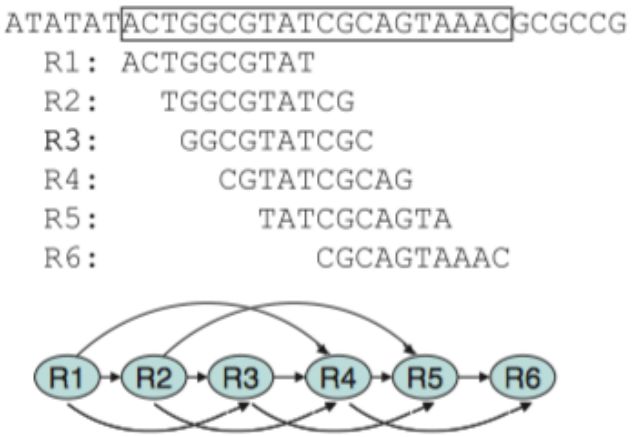
2.4.2.2 Velvet

Velvet er også et program som benytter seg av DBG-algoritmen. Det består av to programmer som alltid brukes sammen, *velveth* og *velvetg*. Førstnevnte leser inn alle read-filer, og bygger et bibliotek besteånde av k-merer og konstruere sammenstillingene mellom dem. *Velvetg* leser sammenstillingene og bygger en så enkel graf som mulig. I velvet settes k-verdien (eller hash-verdien) manuelt av bruker, men dersom noe ikke spesifiseres velges automatisk 31bp (Zerbino, 2010). Velvet starter med byggingen av en hash-tabell, bygger så en DBG og prøver deretter å simplifisere den og fjerne eventuelle feil og feilaktige elementer, som utstikkere og bobler i grafen. Assembleringen skjer ved å finne den beste veien gjennom grafen (Zerbino and Birney, 2008). I denne oppgaven ble Velvet re-kompilert slik at det var mulig å sette k-verdien mye høyere enn standarden på 31.

2.4.2 Overlap-layout-consensus (OLC)

Algoritmen bruker grafer til å modellere parvise forhold mellom read-sekvensene (Miller et al., 2010). Her representerer nodene hele reads, og linkene mellom dem representerer likheter mellom sekvenser. OLC-algoritmen kartlegger forholdet mellom reads ved å sammenlikne

alle reads med hverandre for å kunne identifisere overlappende baser i sekvensene. Dette gjøres med en ”seed-and-extend” – tilnærming (Miller et al., 2010). Linkene i grafen dannes mellom overlappende sekvenser, og reads knyttes sammen, som vist i figur 2.4.4. Til slutt vil veien gjennom linkene samles til en konsensus sekvens, og herfra er det mulig å bygge contigs. Utfordringen er å finne en sti som går gjennom alle noder kun en gang, noe som ofte omtales som en Hamilton-sti. Noe som bidrar til å gjøre dette vanskelig er falske positive veier/linker som tilfeldig dannes ved at ulike deler av genomet likner. Sekvenseringsfeil vil kunne virke i motsatt vei, og hindre eventuelle riktige linker, ved å introdusere en større ulikhet. Det er per i dag ikke påvist at det er mulig å bestemme en Hamilton-vei gjennom en graf, noe som gjør at denne algoritmen får betegnelsen NP-hard. Det finnes ingen løsning. Dette gjør også at algoritmen bruker lang tid for å komme så nær en optimal løsning som mulig Hvor godt resultatet blir, avhenger av ulike parametere deriblant minimum overlappingslengde, samt antall mismatcher som tillattes i en overlappende sekvens (Miller et al., 2010).



Figur 2.4.4: Fremstilling av oppbygningen til OLC- algoritmen. Reads samles i multiple sammentillinger og på bakgrunn av linker mellom nodene finnes det en konsensus sekvens og det dannes contigs.(Li et al., 2012)

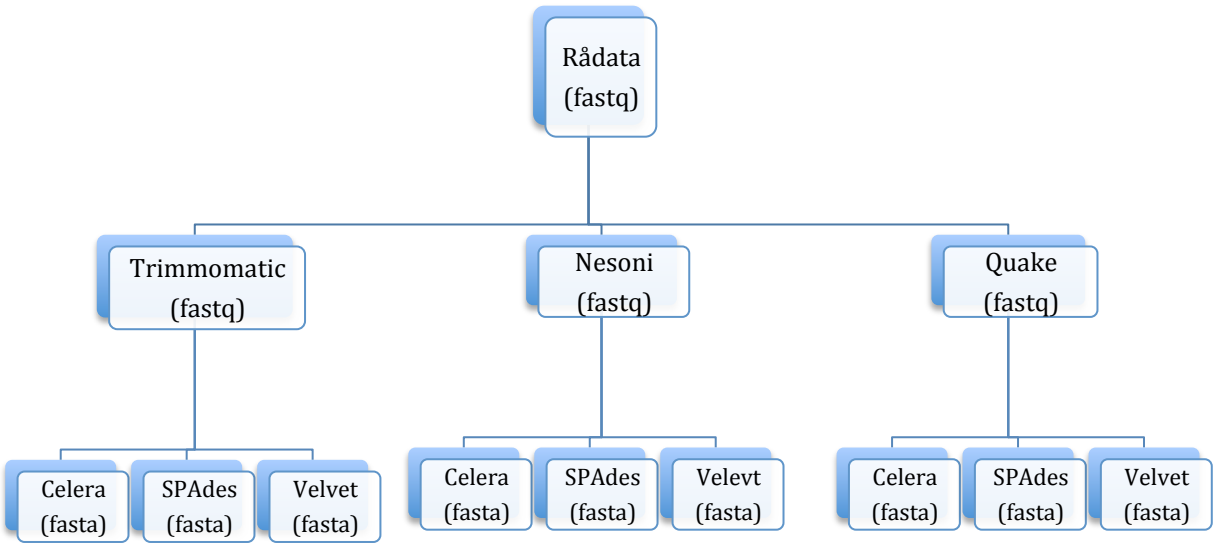
2.4.2.1 Celera

Celera assembler baserer sin assemblering på OLC-algoritmen. Den sender read-data gjennom en prosess av flere trinn, der hvert steg gir et resultat som sendes videre til neste steg. For å utelate repetitive sekvenser, sjekkes først alle fragmenter etter kjente repetitive sekvenser, og områdene merkes, eller fjernes for videre behandling. Alle fragmenter sammenstilles på jakt etter overlappende sekvenser. ”Unitigs” dannes ved sammenslåing av fragmenter med overlappende sekvenser, med et hensyn til repetitive områder. Videre dannes contigs ved å slå sammen *unitigs.* Contigene bygges blant annet ved å slå sammen unitigs som

konsekvent inneholder en del hver fra minst to read par, overlappende sekvenser av read-par og ved bruk av kvalitetsverdier for å sette fragmenter sammen. Av contigs dannes scaffolds, og til slutt dannes en konsensusberegning, og en konsensus baserekkefølge konstrueres.

2.4.4 Innstillinger og bruk

Både Quake, Trimmomatic og Nesoni ble i denne oppgaven brukt med standard innstillinger. Alle preprosesseringsprogram ble kjørt gjennom R ved bruk av R-pakken *IKBMassembly* som er utviklet av ansatte ved Institutt for Kjemi, Bioteknologi og Matvitenskap ved NMBU. Programmene ble utført på regneclusteret her på campus. Alle tre preprosesseringene ble kjørt på alle 24 genomer ved hjelp av første del av scriptet *assembly_pipeline.R*. Her sendes reads inn som fastq-filer, med en lengde på 301bp. Fra preprosesseringen kommer det ut igjen nye fastq-filer som fremdeles inneholder reads, men noen er nå kortere etter kvalitetssikring og trimming. Noen parede reads er også blitt splittet opp på dette punktet. Det var tre sett med preprosseserte reads, et fra hvert program. Alle tre sett med fastq-filer ble brukt til assemblering, alle med tre ulike assembleringsprogram. Fastq-filene fra de tre preprosesseringsprogrammene ble brukt som input til de tre scriptene *assembly_pipeline_celera.R, assembly_pipeline_spades.R* og *assembly_pipeline_velvet.R.* For hver preprosessering med hver av de tre assembleringsprogrammene, ble det produsert filer med scaffold-sekvenser, men også filer med logg-informasjon. Her ble fokuset videre på fasta-filene med assemblerte scaffolds. En fullstendig oversikt over gangen i prosjektet er gitt i figur 2.4.5. Dette er et faktorialt forsøk der alle tre nivåer av faktoren preprosessering er kombinert med alle nivåer av faktoren assemblering.



Figur 2.5.1: Oversikt over alle stegene fra rådata til assembleringsprogrammene som gir scaffold-filer.

For isolatet *Bacillus cereus* ATCC14579 ble det i tillegg gjennomført en Velvet-assemblering med tre ulike k-verdier på 31, 91 og 151. Dette isolatet ble også forsøkt assemblert uten noen form for preprosessering. Grunnen til at dette isolatet ble behandlet på denne måten var tilgjengeligheten på et allerede assemblert og publisert fullstendig genom fra denne stammen. Dette kunne dermed fungere som en slags fasit ved sammenlikning.

2.4.4.1 Satte parametere

Da programmene ble kjørt ble følgende parametere satt:

For preprosessering:

- laveste phred-verdi for trimming: 20
- minimum read-lengde: 36
- k-lengde i Quake: 15

For assemblering:

- sekvenseringsteknologi: ”illumina-long” (Celera)
- insert lengde mellom parede reads: 500-200 (Celera og Velvet)
- bruk av den beste overlappende grafen for å bygge unitigs: ”bogart” (Celera)
- brukersatt k-verdi i k-mers: 151 (Velvet)
- minimum coverage: 2 (SPAdes og Velvet)
- minimum lengde på contigs: 500 (Velvet)

2.5 Evaluering

Resultatene av videre analyser avhenger av kvaliteten på assembleringen. Det er derfor viktig å evaluere assembleringen da assembleringsmetoden i varierende grad kan være vellykket for isolatene. Den ideelle måten å evaluere graden av suksess er ikke lett å fastslå, men flere fremgangsmåter er mulige. Det kan være lurt å kombinere disse for å få best innsikt.

2.5.1 N50

N50-verdi er en vanlig måte å evaluere asembleringskvalitet på. Denne lengden tilsvarer den korteste contigen slik at 50% av den totale contiglengden er å finne i contiger med N50 eller lenger. Et eksempel er assemblering A som inneholder fire contiger med lengdene 70bp, 60bp, 40bp og 20bp. Totalt vil lengden på contigene være 190bp. N50-verdien vil da være 60bp ettersom 70+60 = 130, noe som er mer enn 50% av den totale contig-lengden (Bishop, 2014). N50 ble beregnet for alle isolater for alle programkombinasjoner.

2.5.2 Antall scaffolds

Ved å telle antall scaffolds, kan en få et visst inntrykk av hvor god assembleringen har vært. En perfekt assemblering vil klare å samle alle reads til en lang sekvens uten gaps som da tilsvarer det fullstendige genomet. Assemblering til kun en lang sekvens forekommer svært sjeldent, men verdien til antall scaffolds kan gi en indikator på hvor god prosessen har vært. Et stort antall scaffold-sekvenser betyr at få reads er satt sammen, noe som indikerer dårlig assemblering. Antall scaffold-sekvenser ble talt for alle isolater med alle programkombinasjoner.

2.5.3 BLAST mot kjent genom

Ettersom de fleste bakteriegenom innenfor samme art er langt på vei like, er det mulig å sammenlikne de assemblerte isolat-sekvensene mot et allerede kjent helgenom av samme art. En vellykket assemblering vil ha stor likhet med dette genomet. For å gjøre dette ble Basic Logical Alignment Sequencing Tool (BLAST) benyttet. Dette er et verktøy som brukes til å søke med en sekvens (query-sekvens) i en database etter en liknende sekvens. BLAST kan brukes til å søke med og mot både nukleotidsekvenser og proteinsekvenser. I dette tilfellet ble det opprettet en database med nukleotidsekvensen til genomene *cereus_ATCC_14579_PRJNA57975.fsa* og *anthracis_Ames_Ancestor_PRJNA58083.fsa.* Scaffold-sekvensene ble brukt som query-sekvenser. Query-sekvensene blir sammenstilt mot det ønskede genomet og den får en score basert på antall mutasjoner (Altschul, 1990).

Avhengig av hva en er interessert i hentes den ønskede informasjonen ut etter sammenstillingen. For å få en formening om hvor vellykket assembleringen hadde vært ble det beregnet en dekningsgrad mellom referansegenom og scaffolds. Denne forteller hvor stor andel av referansegenomet som dekkes av scaffold-sekvensene. Et perfekt assemblert genom vil få en tilnærmet hundre prosent dekning ved blasting mot et genom fra samme stamme. Her var det dekning i forhold til bakterienes kromosom samt plasmider som ble beregnet. Først ble sammenstillingen som ga den beste "bit scoren" som ble brukt i beregningen av dekning. En bit score er et mål på antall match minus antall mismatch i en sammenstilling, der det gis ulik score for ulike matcher og mismatcher avhengig av hvor lett de oppstår ved en tilfeldighet. Det beste er en høy-bit score. Start og stop-posisjonene til scaffold-sekvensene med best bit score ble brukt til å beregne andelen som dekket det fullstendige genomet. Det ble først kjørt analyser på isolatet *B cereus* ATCC14579 ettersom denne stammen allerede har et fullstendig genom publisert, som ville fungere som en tilnærmet fasit. Deretter ble dekning basert på beste bit score beregnet for alle 24 isolater mot databasegenomet *anthracis_Ames_Ancestor_PRJNA58083.fsa.* Dette genomet ble valgt for å kunne identifisere om noen av isolatene i datasettet kunne klassifiseres som *B. anthracis.*

Dekningsgrad ble i tillegg beregnet ved å bruke sammenstillinger over en viss lengde. Scaffold-sekvenser for alle tjuefire genomer ble blastet mot *anthracis_Ames_Ancestor_PRJNA58083.fsa,* og dekningsgrad beregnet på sammenstillinger med lengder over 1000bp, 5000bp og 10000bp ble beregnet. Gjennomsnittlig dekning for alle kombinasjoner for hvert isolat ble beregnet.

Til å sammenstille genomene mot en BLAST-database ble scriptet *script_blast_genomes.R* brukt. Dette gjør den igjen ved BLAST+ programvaren fra NCBI (http://blast.ncbi.nlm.nih.gov/Blast.cgi).

2.5.4 Mixed model

Når alle dekningsgradene fra isolatene skal sammenliknes for å avgjøre om dekningsgrad signifikant varierer avhengig av hvilke programmer som er valgt i prosessen, er det mulig å gjøre en statistisk analyse med modellen gitt i (2.9).

$$y_{ij} = \mu + \rho_i + \alpha_j + e_{ij} \qquad (2.9)$$

der y_{ij} tilsvarer observert dekningsgrad for preprosesseringsprogram *i* og assembleringsprogram *j,* μ er forventet dekningsgrad uansett metode, ρ_i er effekt av preprosesseringsprogram, α_j er effekt av assembleringsprogram og $e_{ij} \sim N(0,\sigma^2)$ beskriver den tilfeldige variasjonen i observert dekningsgrad y_{ij}.

Problemet med denne modellen er antakelsen om uavhengighet mellom observasjonene y_{ij}. Alle isolater med alle kombinasjoner av programmene blir tatt med i analysen. Hvor god dekningsgraden er avhenger av isolatet. Det betyr at hvilket isolat dekningsgraden er beregnet for, gir store mengder informasjon, og antakelsen om uavhengighet blir veldig feil. Løsningen er å bruke en "mixed model" der vi innfører isolat som en tilfeldig effekt. Dette betyr at feil-leddet e_{ij} fra (2.8) splittes i to komponenter, en som skyldes hvilket (tilfeldig valgt) isolat vi observerer, og en som skyldes alle andre tilfeldige variasjoner. Den nye modellen er gitt i (2.10)

$$y = \mu + \rho_i + \alpha_j + G_k + e_{ijk} \qquad (2.10)$$

der en nå antar $e_{ijk} \sim N(0,\sigma_e{}^2)$ og $G_k \sim N(0,\sigma_G{}^2)$.

Med denne modellen blir det nå tatt hensyn til hvilke isolater dekningsgraden er beregnet for, og antakelsen om uavhengighet kan igjen innføres i modellen. I praksis er den store forskjellen mellom en vanlig to-veis variansanalyse som i (2.9) og en mixed model i (2.10) i vårt tilfelle være at for sistnevnte får vi langt mer presise estimater av de fast effektene ρ_i og α_j, og selv ganske små, men systematiske, effekter blir detektert.

2.6 Bestemmelse av sekvenstype

2.6.1 Multilocus sequence typing (MLST)

Multilocus sequence typing (MLST) angir genotypen til bakterien basert på sekvenssammenstilling i ulike husholdningsgener (gener som ofte brukes i MLST er husholdningsgener). I prinsippet er husholdningsgener gener som skal finnes i alle isolater til den aktuelle arten som skal studeres, men dette er ikke alltid tilfelle. Genene koder for de mest primære oppgavene og er ofte viktige for cellers funksjon. En kan til en viss grad vite at genet en leter etter skal være der, og det er mulig å registrere ulike varianter. Figur 2.6.1 viser to allel-varianter i et lokus der det kun skiller en base.

Isolat a	AGTCGTCAAGTTGG	allel 1
Isolat b	AGTCGTCTAGTTGG	allel 2

Figur 2.6.1: To sekvenser der det skiller en base i samme lokus. Dette gir ulike allel-typer hos de to isolatene.

For hver variasjon vil dette allelet tildeles et nummer, ovenfor vil det da henholdsvis være 1 og 2. Avhengig av antall isolater som studeres og hvor heterogent genomet er, kan det etter hvert være registrert hundrevis av ulike alleler. DNA-sekvenser fra isolater sekvenseres og sammenliknes med allerede eksisterende markører. Hvert lokus i DNA-sekvensen får dermed tildelt et nummer tilsvarende det allelet som er funnet i gitte lokus. Et isolat vil få tildelt en rekke med tall som tilsvarer allelprofilen. Disse tallene utgjør sekvenstypen til isolatet, og kan fremstilles i en vektor som vist i (2.11).

ST = [1 7 19 5 8 25 3] (2.11)

Ved sekvensering av flere genomer kan alle sekvenstypene samles i en matrise, som vist i figur 2.6.2

ST1	8	5	9	12	1	5
ST2	8	2	3	12	1	5
ST3	8	2	9	12	1	5

Fugur 2.6.2. Illustrasjon av en sekvensmatrise med sekvenstyper for tre isolater.

I et slikt datasett er det enkelt å avgjøre hvor isolater skiller i de utplukkede genmarkørene. Det er lurt å ta med i betraktningen hvilke gener som velges som markørgener. Derimot er det ikke like lett å skille nært beslektede stammer fra hverandre, da de er svært like, også på sekvensnivå. I tillegg er denne typingsmetoden noe dyr sammenliknet med andre typingsmetoder. Til gjengjeld oppfyller den alle andre krav, og er spesielt hendig med hensyn til at det er enkelt å distribuere og bruke tabellene med sekvenstyper videre. (Sabat, 2013, Maiden, 1998)

2.6.2 BLAST med MLST-markører

For å kunne bestemme sekvenstypen til hvert isolat ble det brukt 14 kjente MLST-markører for *B. cereus*. Syv av dem hentet fra NCBI (http://pubmlst.org/bcereus/, 2015), og de resterende syv ble hentet fra Helgasons artikkel der de utviklet et MLST-skjema for *B. cereus* (Helgason et al., 2004). Sekvenstype ble bestemt ved bruk av BLAST, der markørene ble sammenstilt med hvert av isolatenes scaffold-sekvenser, og det ble registret hvilket allel som eventuelt matchet. Dette ble gjort med de 24 genomene i datasettet sammen med syv ekstra genomer hentet fra NCBI, for sammenlikning senere i en fylogenetisk analyse En oversikt over de stammene lagt til i datasettet vises i tabell 2.4.1. Markørene brukt i sammenstillingen finnes i tabell 2.4.2.

Tabell 2.4.1: Oversikt over de genomsekvenser hentet inn fra NCBI med tilhørende accession nr.

Art	**Stamme**	**Accession number**
cereus	AH676	NZ_CM000738.1
cereus	B4264	NC_011725.1
cereus	Rock1-15	NZ_CM000729.1
weihenstephanensis	KBAB4	NC_010184.1
cereus	AH1134_PRJNA54485	NZ_ABDA00000000.2
cereus	VD166_PRJNA181702	NZ_AHFI00000000.1
cereus	VD184_PRJNA203487	NZ_AHFK00000000.1

Tabell 2.4.2: Oversikt over MLST-markørene brukt for å finne sekvenstypen til hvert isolat, deres lengde samt gen-funksjon. Markørene som er hentet fra NCBI er merket med rødt.

MLST-markører	**Lengde**	**Gen-funksjon**
adk	450bp	adenylate kinase
ccpA	418bp	catabolite control protein A
ftsA	401bp	cell division protein
glp	372bp	alpha-glycerophosphate oxidas
glpT	330bp	glycerol-3-phosphate permease
gmk	504bp	guanylate kinase
ilv	393bp	dihydroxy-acid dehydratase
pta	414bp	phosphate acetyltransferase
pur	348bp	phosphoribosylaminoimidazolecarboxamide
pyc	363bp	pyruvate carboxylase
pyrE	404bp	orotate phosphoribosyltransferase
recF	470bp	DNA replication and repair protein
sucC	504bp	succinyl-CoA synthetase subunit beta
tpi	435bp	triosephosphate isomerase

Til å kjøre BLAST med MLST-markørene, ble *micropan*-pakken i R benyttet (Snipen and Liland, 2015). R-scriptet *script_mlst.r* ble brukt til å sammenstille genomene og MLST-sekvensene. Ved oppdagelse av MLST-sekvensen i genomet, fikk isolatet en indeksering avhengig av hvilken allel-type som ble oppdaget. Dersom programmet oppdaget en ny alleltype ble denne lagt inn i filen med allerede eksisterende alleler for den aktuelle markøren. Dette gjorde at om denne markøren skulle oppdages igjen ved en senere sammenstilling, ville den få den tilsvarende indeksering. I funksjonen *mlstScan*, som er utviklet av hovedveileder Lars Snipen, ble informasjonen fra BLAST for hvert genom lagt inn i en tabell med tre kolonner og en rad for hvert genom. Allel-typen i kolonne en, matchende MLST-sekvens funnet i genomet i kolonne to, og en kommentar i siste kolonne. Det var BLASTn (n→nukleotid) som ble benyttet i funksjonen, og det ble ikke tillatt gaps i sammenstillingen (”ungapped”). Det vil si at lengden på MLST-sekvensen måtte være helt lik lengden på sekvensen funnet i genomet. Videre ble det laget en fullstendig matrise med alle sekvenstyper ved å bruke funksjonen *appendAlleles*. Her ble informasjonen i tabellen med alleltyper brukt til å lage en vektor med sekvenstypen til hvert isolat. Vektorene ble lagt sammen til en fullstendig matrise.

2.7 Avstandsberegning

For å videre kunne gjøre fylogenetiske analyser må det ligge til grunn et utgangspunkt for den relative slektskaps-avstanden mellom isolatene. Dette kan gjøres på ulike måter, og i denne oppgaven ble det brukt en Hammingavstand.

2.7.1 Hammingavstand

Hammingavstanden mellom to vektorer av lik lengde, kan defineres som antallet plasser der de to vektorene er ulike. (Fiedler, 2004) Et eksempel er

A = [4 3 9 2]
B = [4 1 9 5]

Sekvens A og B er ulike i to posisjoner, og hammingavstanden blir dermed 2. Dette kan oppsummeres i en formel, (2.12).

$$D_{ij} = \sum_{k=1}^{K} I(X_{ik} \neq X_{jk}) \qquad (2.12)$$

D_{ij} representerer avstanden mellom de to objektene X_{ik} og X_{jk}, der begge objekter har en lengde K. k er posisjonen(e) der de to objektene skiller seg fra hverandre (Pinheiro et al., 2005). I denne oppgaven ble hammingavstanden beregnet med kommandoen *distHamming* i R.

2.8 Fylogenetiske analyser

Fylogenetiske trær bygges opp via clustring, det vil si gruppering av data basert på likheter og forskjeller, der like data plasseres i samme gruppe. Grupperingen er som regel avhengig av en avstandsmatrise, som er beskrevet i avsnittet ovenfor. Det er vanlig å fremstille slike analyser som et tre, eller i grupperinger.

2.8.1 eBURST

eBURST er en grupperingsalgoritme som prøver å gruppere slektende grupper av organismer med et utgangspunkt i det den finner som ”grunnlegger-organismen”. Ut fra denne stamfaren linkes det til etterkommere ved å trekke en linje mellom individene. Ut fra disse igjen trekkes det nye linjer til neste generasjon, og så stopper algoritmen. Hvor stor forskjellen skal være på

grunnlegger og etterfølger er en parameter som innstilles av bruker. Ved bruk av sekvenstyper er en vanlig innstilling at seks av syv loci skal ha samme indeksering for å kunne trekke en parallell mellom dem, og dermed fastslå nært slektskap. Ingen sekvenstyper fra individene kan være tilstede i mer enn en gruppe. De sekvenstypene som ikke kan plasseres i en gruppe kalles singletons. Sekvenstyper i en eBURST-gruppe med den strengeste defineringen av gruppen er nært beslektet og regnes som et klonalt kompleks. Dette er noe av ulempen med eBURST, da den ikke viser noen sammenheng mellom de klonale kompleksene, selv om de skulle være beslektet(Feil et al., 2004). eBURST ble brukt gjennom en R-pakke som er utviklet på IKBM ved NMBU. De 24 isolatene samt de syv ekstra genomene ble analysert med eBURST der avstandsmatrisen inneholdt avstander beregnet med Hamming-metoden.

2.8.2 Neighbor joining trær

Neighbor joining (NJ) trær bruker evolusjonære avstander for å konstruere trær. Med utgangspunkt i et stjerneformet tre, er målet å gruppere slik at for hver steg minimeres den totale grenlengden i treet. Naboer i et slikt tre vil være to par med enheter som henger sammen gjennom en node. Ut fra det opprinnelige datasettet beregnes lengden til alle grenene mellom alle enheter, og der avstanden er kortest knyttes de to enhetene sammen med en node. Videre beregnes nye avstander, og nå regnes de to sammenslåtte enhetene som en. Dette gjentas til det er dannet et fullstendig sammensatt tre (Saitou, 1987). Her ble de beregnede hammingavstandene som nevnt ovenfor, benyttet som avstander for å sette sammen grenene i treet, som bestod av isolatene i datasettet samt de ekstra genomene.

3. Resultater

3.1 Preprosessering

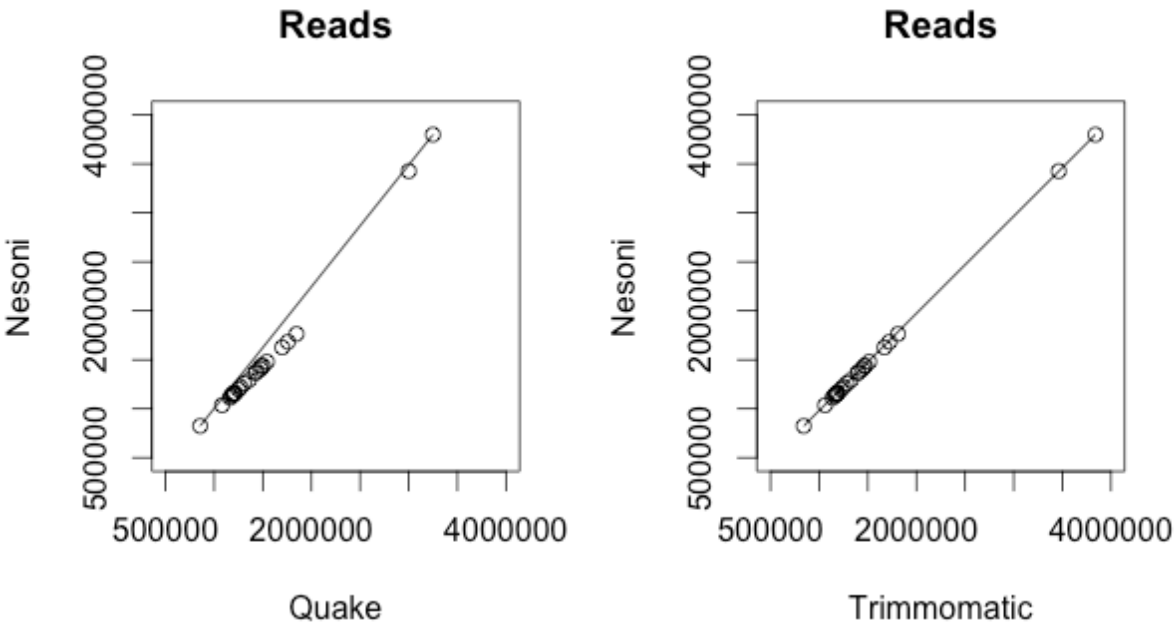
Hvert av de tre preprosesseringsprogrammene ga 24 nye fastq-filer med trimmede reads. Antall reads produsert for hver genom ble sammenliknet, og tabell 3.1 viser en oversikt over gjennomsnittlig antall reads per isolat for hvert preprosesseringsverktøy og før preprosessering, samt det totale antallet.

Tabell 3.1: Gjennomsnittlig og totalt antall reads for hvert isolat etter prosessering av de ulike programmene, samt før preprosessering

	Nesoni	Quake	Trimmomatic	Før prepross.
Gjennomsnittlig ant. reads per isolat	1 485 019	1 495 031	1 515 782	1 592 008
Totalt antall reads	35 640 458	35 880 746	36 378 767	38 208 204

Figur 3.1 viser to plot der read-tallet for hvert isolat ble satt opp i forhold til hverandre. Hver sirkel i plotet tilsvarer et isolat, mens størrelsen på x- og y-akse representerer antall reads for det gitte genomet. Her er Quake sammenliknet med Nesoni (a) og Trimmomatic sammenliknet med Nesoni (b). Det ble også kalkulert korrelasjonskoeffisienter for de to plottene, som ble på henholdsvis R= 0.9944275 og R= 0.9999651. Denne korrelasjonen forteller om samsvaret i antall reads for hvert isolat for to programmer. Om de to programmene hadde fått likt antall reads i hvert isolat, ville korrelasjonskoeffisienten vært 1. De fleste isolatene har et sted mellom en og to millioner reads, men to skiller seg ut med over tre millioner. Antall reads i isolatene er omtrent likt, uavhengig av brukte preprosesseringsprogram.

(a) **(b)**



Figur 3.1. Sammenheng mellom antall reads funnet i alle genomer med henholdsvis Quake og Nesoni (a) og Trimmomatic og Nesoni (b).

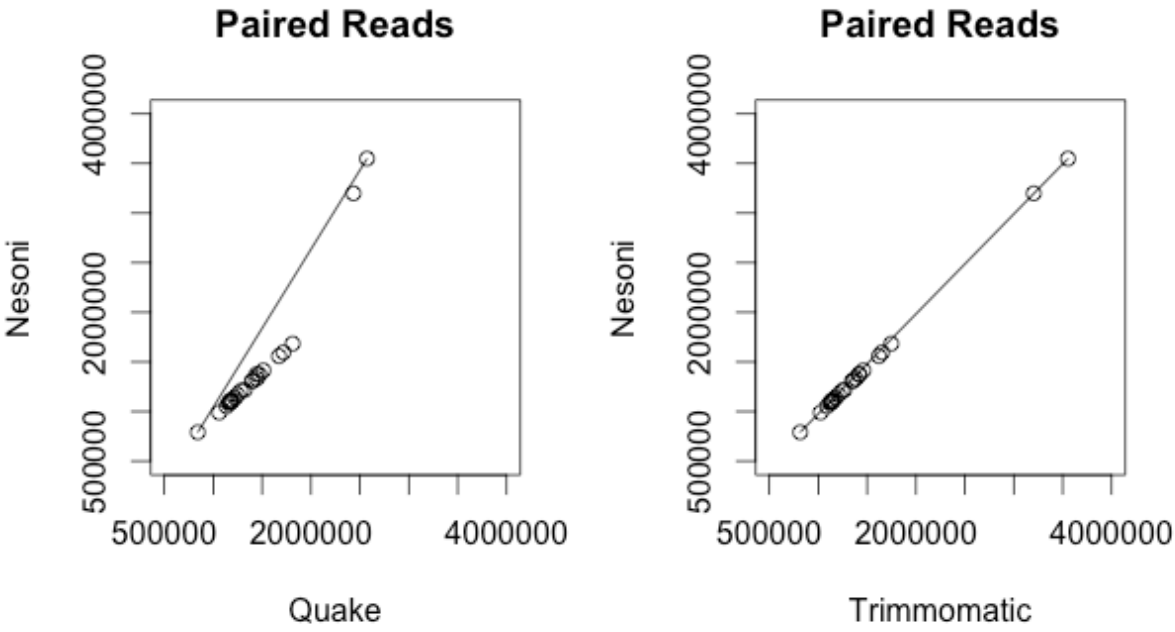
Videre ble det talt opp antall reads de tre preprosesseringsprogrammene hadde klart å beholde read-parene for. Et gjennomsnittlig antall parede reads per isolat, samt totalt antall er gitt i tabell 3.2

Tabell 3.2. Oversikt over antall parede reads for hvert isolat etter prosessering av de ulike programmene.

	Nesoni	Quake	Trimmomatic	Før prepross.
Gjennomsnittlig ant. parede reads per isolat	1 411 334	1 419 161	1 450 648	1 592 008
Totalt antall parede reads per isolat	33 872 012	34 059 858	34 815 544	38 208 204

Dette er også fremstilt i figur 3.2, der to og to programmer er satt opp mot hverandre, og antall parede reads for hvert isolat fremkommer. Sirklene representerer igjen hvert isolat. På y-aksene leses det av antall parede reads for Nesoni, mens det på x-aksene er antall parede reads for Quake (a) og Trimmomatic (b). Korrelasjonskoeffisientene ble beregnet også her, og fikk verdier på henholdsvis R= 0.9712856 og R= 0.9998203.

(a) **(b)**



Figur 3.2. Sammenheng mellom antall parede reads funnet i alle genomer med henholdsvis Quake og Nesoni (a) og Trimmomatic og Nesoni (b).

3.2 Vurdering av assemblering

Etter assembleringen var resultatet 24 fasta-filer med scaffold-sekvenser behandlet på til sammen ni forskjellige måter; alle kombinasjoner av de tre preprosesseringsprogrammene Trimmomatic, Quake og Nesoni, og assembleringsprogrammene Spades, Velvet og Celera.

3.2.1 Sammenlikning av assemblerte sekvenser med eksisterende genom fra samme stamme

Et av genomene i datasettet var *Bacillus cereus* ATCC14579. Dette genomet er tidligere assemblert, og hele genom-sekvensen er kartlagt og publisert. For å vurdere effekten av programvalg var det naturlig å gjøre en sammenlikning av assemblerte scaffold-sekvenser fra stammen i det opprinnelige datasettet, med det ferdig kartlagte genomet. Dette ble gjort ved bruk av blast der *Bacillus cereus* ATCC14579 ble hentet fra NCBI og sekvensene ble sammenstilt. Det ble utført en sammenstilling der det ble tatt utgangspunkt kun i sammenstillingen som ga beste bit-score, og resultatet finnes i tabell 3.3. Her er det en tydelig sammenheng mellom valg av assemblerer og preprosesseringsprogram. Dekningsgrad

(cov.chr og cov.psd for hhv kromosom og plasmid) angir hvor mye av genomet i databasen som dekkes av query-sekvensen, noe som her er noe varierende.

Tabell 3.3: Oversikt over antall scaffolds produsert, genomstørrelse, antall fremmede bokstaver i sekvensen (ikke A, T, C, G), kromosom-coverage og plasmid-coverage. Tabellen gjelder for blasting kun med stammen ATCC14579.

Preproc	Assembler	Scaffolds	Size	Aliens	Cov.chr	Cov.psd
Trimmomatic	Celera	29	4927349	320	0.6061480	0
Trimmomatic	Spades	50	5367952	400	0.8485562	1
Trimmomatic	Velvet31	401	5291132	62	0.9884462	1
Trimmomatic	Velvet91	104	5347340	222	0.9902966	1
Trimmomatic	Velvet151	84	5357929	124	0.9905459	1
Nesoni	Celera	6	4727262	530	0.9907064	1
Nesoni	Spades	52	5367881	405	0.9907473	1
Nesoni	Velvet31	525	5291674	103	0.9911835	1
Nesoni	Velvet91	108	5348795	133	0.9912035	1
Nesoni	Velvet151	96	5359446	123	0.9912955	1
Quake	Celera	25	5367869	155	0.9916453	1
Quake	Spades	50	5369768	399	0.9916568	1
Quake	Velvet31	499	5291155	80	0.9916630	1
Quake	Velvet91	118	5347839	90	0.9916630	1
Quake	Velvet151	84	5357929	124	0.9916630	1
NoPreproc	Celera	26	5384926	191	0.9929720	1
NoPreproc	Spades	50	5369719	399	0.9929722	1
NoPreproc	Velvet31	1168	5219487	151	0.9943655	1
NoPreproc	Velvet91	127	5345126	0	0.9943719	1
NoPreproc	Velvet151	88	5357576	8	0.9943719	1

3.2.2 Evaluering av assembleringer ved sammenstilling mot referansegenom

Tilsvarende metode ble brukt på alle 24 isolater i datasettet, og det ble nå brukt et ferdig kartlagt genom fra *Anthracis Ames Ancestor PRJNA58083* til bruk i sammenlikningen. Dette genomet er spesielt interessant, da det vil avsløre om noen av isolatene er bærere av plasmidene pXO1 og pXO2 og dermed kan defineres som en *Bacillus anthracis,* samtidig

som stammen er beslektet de i datasettet. Igjen ble kun sammenstillingene med best score brukt til dekningsberegning, noe som generelt ga lave dekningsverdier. Derimot var det noe variasjon mellom kombinasjonene. Dette ble undersøkt med en statistisk analyse ved brukt av en ”mixed model”. Resultatet finnes i tabell 3.4. Den forventede dekningsgraden er gitt under faste effekter, under estimater. Dette er den forventede dekningsgraden ved bruk av Trimmomatic etterfulgt av Celera er å finne i første rad, og denne kombinasjonen fungerer som en referanse til de andre kombinasjonene. Det forventes her en dekning på omtrent 23,4%. Forventet økning i dekningsgrad ved å bytte ut Trimmomatic med Nesoni er med 0,101092, og tilsvarende ved å skifte til Quake øker forventet dekningsgrad med 0,130839. Ved å skifte ut Celera med SPAdes eller Velevet øker forventet dekningsgrad med henholdsvis 0,130839 og 0.011471. Det skal påpekes at byttet til SPAdes ikke gir signifikant endring i forventet dekning. Det vil si at til tross for at det er en forskjell i dekning ved bruk av SPAdes i forhold til Celera, kan en likevel ikke påstå at denne forskjellen er større enn man kan forvente ved en tilfeldig variasjon i data.

Tabell 3.4: Oversikt over data ved bruk av ”mixed model” statistisk metode for å vurdere variasjon i dekning når kun sammenstillingen med beste bit-score er brukt i beregningen, for alle verktøy-kombinasjoner for alle genomer. Antall obeservasjoner er 216 (24 isolater, alle med ni kombinasjoner av verktøy)

Tilfeldige effekter			
	Varians	Standardavvik	
Grupper	0.014312	0.11963	
Residualer	0.002044	0.04521	
Faste effekter			
	Estimat	Standardfeil	t-verdi
Trimmomatic og Celera	0.234182	0.025370	9.231
Nesoni	0.101092	0.007535	13.416
Quake	0.130839	0.007535	17.363
SPAdes	0.011471	0.007535	1.522
Velvet	0.062863	0.007535	8.342

For å undersøke om det var mulig å oppnå en bedre dekning ble selekteringen av scaffold-sekvenser brukt i beregning av dekning, endret til å inkludere alle sammenstillinger innen en viss lengde. Denne lengden ble satt til å være 5000bp da dette tilsvarer omtrent størrelsen til et bakterie-operon. Alle sammenstillinger lenger enn 5000bp ble nå tatt med i beregningen av

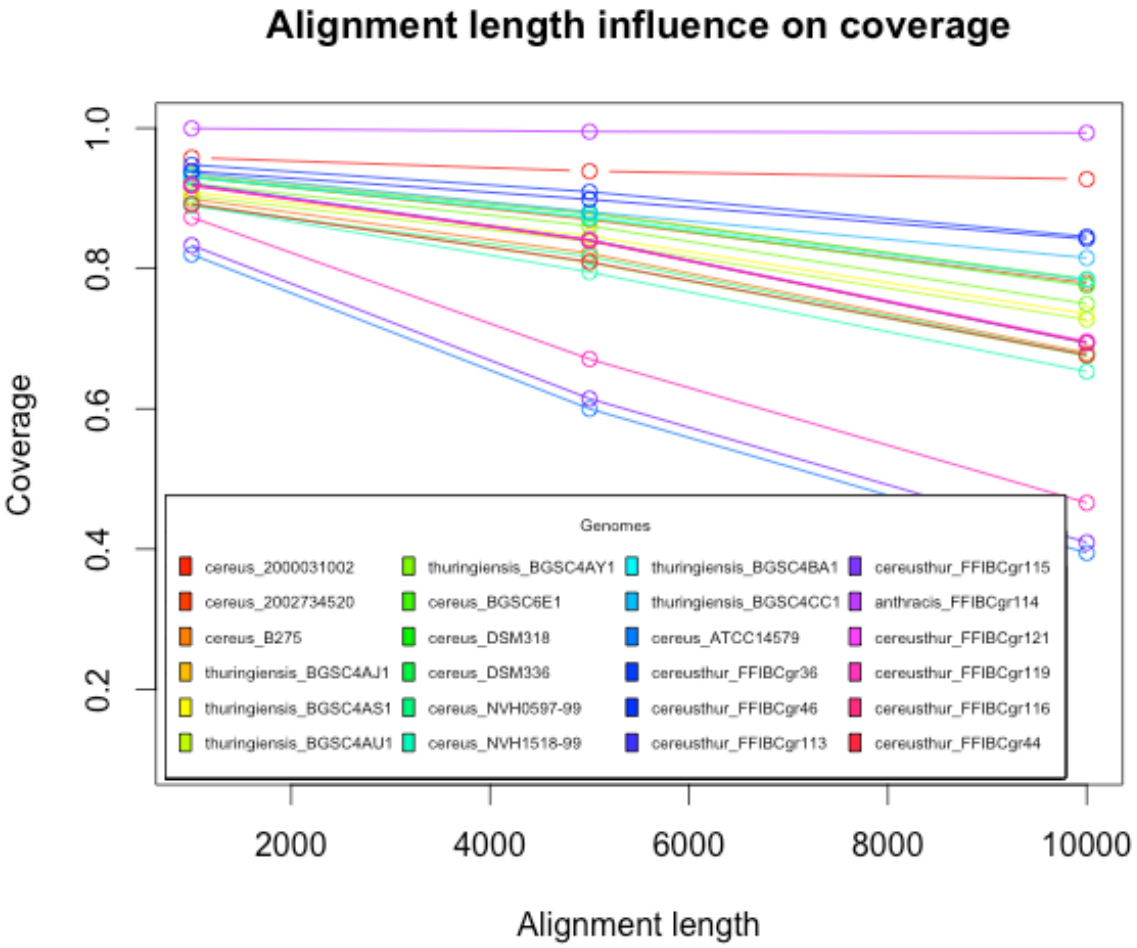
dekning, og det ble observert en generell økning. Igjen ble det utført en ”mixed model” statistikk, og resultatene vises i tabell 3.5. Ved å bruke sammenstillinger over en viss lengde blir forventet dekningsgrad med Trimmomatic etterfulgt av Celera 82,4%. En utskiftning av preprosesseringsprogram gir en økning i forventet dekningsgrad med henholdsvis 0,0029848 og 0,0033092 med Nesoni og Quake. Et bytte i assembleringsprogram fra Celera til SPAdes gir en økning i forventet dekningsgrad med 0,0022933, mens et bytte til Velvet fører til en økning med 0,0022954. T-verdiene viser at endringen i forventet dekning som følge av endring i program, er signifikant.

Tabell 3.5: Oversikt over data gitt ved bruk av ”mixed model”, statistisk metode for å vurdere variasjon i coverage når sammenstillinger med lengde lenger enn 5000bp ble brukt i beregningen, for alle kombinasjoner for alle genomer.

Tilfeldige effekter			
	Varians	Standardavvik	
Grupper	7.816e-03	0.088411	
Residualer	2.001e-05	0.004473	
Faste effekter			
	Estimat	Standardfeil	t-verdi
Trimmomatic og Celera	0.8242236	0.0180596	45.64
Nesoni	0.0029848	0.0007455	4.00
Quake	0.0033092	0.0007455	4.44
SPAdes	0.0022933	0.0007455	3.08
Velvet	0.0022954	0.0007455	3.08

3.2.3 Dekning som et resultat av lengden på sammenstillingen

En forventet dekningsgrad på omtrent 82,4% (fra tabell 3.5) er fremdeles ganske lavt. Det ble beregnet dekningsgrad for sammenstillinger på over 1000bp, 5000bp og 10000bp for alle isolater, og sammenhengen fremstilles i figur 3.3. Det er tydelig at for de fleste isolatene vil scaffold-sekvensene gi lavere dekningsgrad når sammenstillingskravet blir strengene.



Figur 3.3. Grafen viser sammenheng mellom dekningsgrad og kravet om lengden på sammenstillingen ved blasting av scaffolds fra alle isolater mot referansegenomet Anthracis Ames Ancestor PRJNA58083. Dekningsgraden er et gjennomsnitt fra alle programkombinasjoner.

Isolatet FFIBCgr114 får svært høy dekningsgrad. Den fikk også opp mot 100% dekning mot de to plasmidene i referansegenomet, noe som gjør at den kan antas å være en *B. anthracis*.

3.2.4 N50

Det ble i tillegg beregnet N50-verdier for alle isolater for alle programkombinasjoner. Dette ble også sammenliknet med en ”mixed model” analyse, og resultatet vises i tabell 3.6.

Tabell 3.6. ”Mied model” analyse av N50-data for alle isolater behandlet med alle kombinasjoner av preprosessering og assemblering.

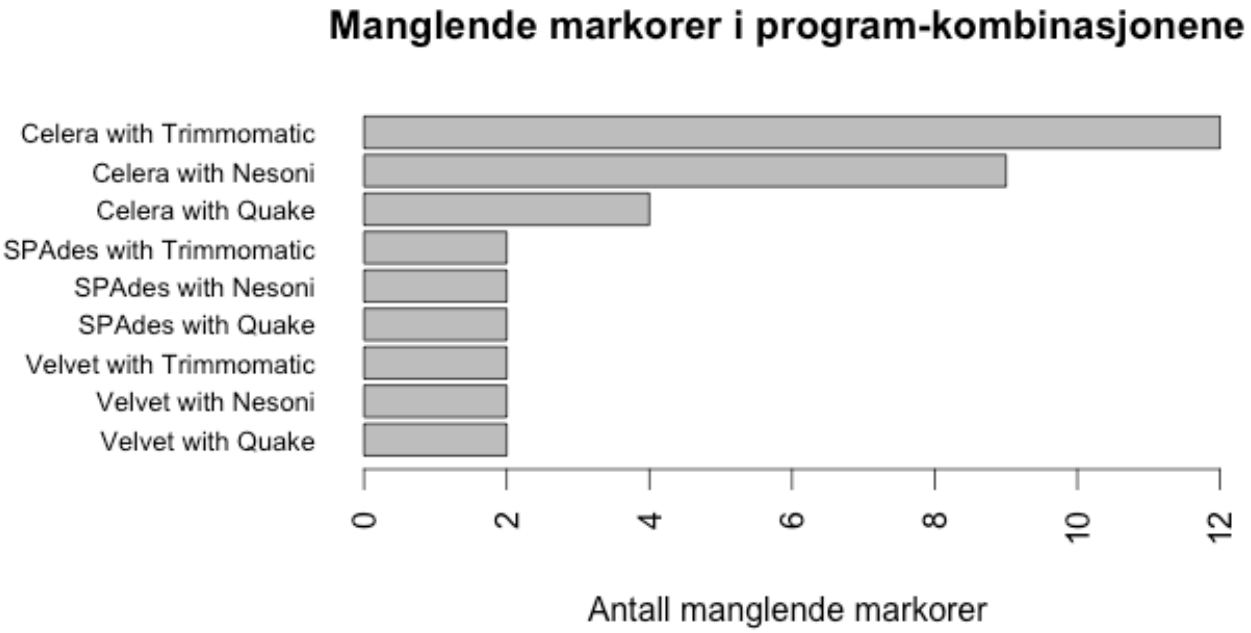
Tilfeldige effekter			
	Varians	Standardavvik	
Grupper	4.226e+10	205566	
Residualer	2.034e+11	450999	
Faste effekter			
	Estimat	Standardfeil	t-verdi
Trimmomatic og Celera	823710	80430	10.241
Nesoni	10180	75167	0.135
Quake	-5842	75167	-0.078
SPAdes	-164502	75167	-2.188
Velvet	-626541	75167	-8.335

Forventet N50-verdi ved bruk av kombinasjonen Trimmomatic-Celera gir en forventet N50-verdi på 823 710bp. En signifikant reduksjon i N50-verdi får en ved å bytte til SPAdes eller Velvet som har henholdsvis 164 502 og 626 541 basepar færre i sine forventede N50-verdier. Forskjellen ved å bytte preprosesseringsprogram er ikke signifikant. En lavere N50-verdi indikerer lengre scaffold-sekvenser og dermed også et lavere antall scaffolds.

3.3 Søk etter MLST-markører

Videre var målet å søke gjennom de assemblerte genomene etter gitte MLST-markører. Dette ble gjort ved en sammenstilling av genom og MLST-sekvenser ved bruk av blast. Det ble søkt gjennom alle alleler for hver markør, og det eventuelle allelet som ble funnet igjen ble lagret i en vektor med et tall tilsvarende allelet for hver markør. Dette er sekvenstypen til hvert isolat basert på de fjorten markørene. Alle vektorene ble samlet i en matrise. De stedene der R-funksjonen ikke klarte å finne igjen et allel for en markør i et genom, ble det stående ”NA” der det ellers er satt inn allel-typen. Antall NA, altså manglende markører, var noe varierende for de ni kombinasjonene av preprosessering og assembleringsprogrammer, og en oversikt over antall manglende markører observert i allel-matrisen finnes i figur 3.4. De

kombinasjoner av isolater og markører der det ikke ble funnet noen markør er angitt i tabell 3.7. For noen isolater er det flere markører som ikke blir funnet.



Figur 3.4: Oversikt over antall manglende markører i sekvensmatrisene for de ulike kombinasjonene av preprosesserings- og assembleringsprogrammer

Tabell 3.7: Oversikt hvilke programkombinasjoner som gir manglende markør-sekvenser (Trim. for Trimmomatic).

	Manglende markører kombinasjonene								
	Velvet			SPAdes			Celera		
Isolat	Quake	Nesoni	Trim.	Quake	Nesoni	Trim.	Quake	Nesoni	Trim.
cereus FFIBCgr46	*pyc*	*pyc*	*pyc*	*pyc*	*pyc*	*pyc*	*pyc*	*pyc*	*pyc*
cereus AH676	*sucC*	*sucC*	*sucC*	*sucC*	*sucC*	*sucC*	*sucC*	*sucC*	*sucC*
cereus 2000031002							*adk*	*adk, pta, pur, recF*	*adk, glpT, pta, pur, recF, tpi*
cereusthur FFIBCgr115							*glp*	*glp*	*glp*
cereus BGSC6E1								*adk, glpT*	*glpT*
thuringiensis BGSC4AU1									*pta, recF*

Fravær av allel-type gjør at det ikke er mulig å danne en avstandsmatrise med isolatet eller allelet som inneholder manglende markører, ettersom det ikke er noe tall å beregne avstand til i den aktuelle posisjonen. Dette gjør at enten genomet eller markøren må tas ut av matrisen, for så å beregne avstander basert på de resterende. De kombinasjonene med færrest manglende markører hadde mangler i samme posisjon i matrisen; genomet gr46 og markøren *pyc* samt genomet AH676 og markøren *sucC*. Celera kan sies å være den som jevnt over ga dårligst resultat da det var opptil elleve markører den ikke klarte å finne fordelt på scaffold-sekvenser fra fem ulike isolater. Det ble besluttet å fjerne isolater med manglende markører. For noen programkombinasjoner medførte det et tap av opptil flere isolater.

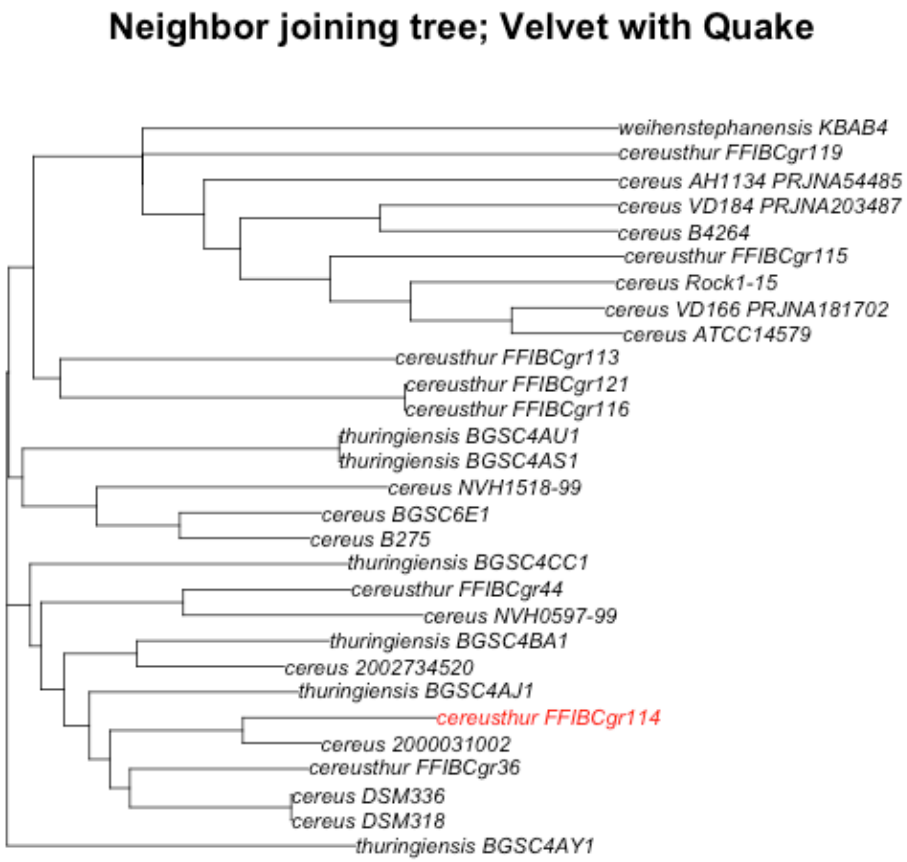
3.4 Hammingavstand

Hammingavstand ble beregnet, og resultatet var ni avstandsmatriser, en for hver kombinasjon av programmer, med tall fra 1 til 14. De isolatene der det ikke ble funnet markører i ble som nevnt fjernet fra datasettet, og hamming-avstand ble derfor kun beregnet for de resterende. Matrisene med sekvenstyper fra alle kombinasjoner med SPAdes og Velvet ble helt identiske, noe som førte til identiske avstandsmatriser. Dette betyr at hvilken av disse som ble brukt videre ikke spilte noen rolle. En av dem ville være representativ for de resterende.

3.5 Fylogenetisk analyse

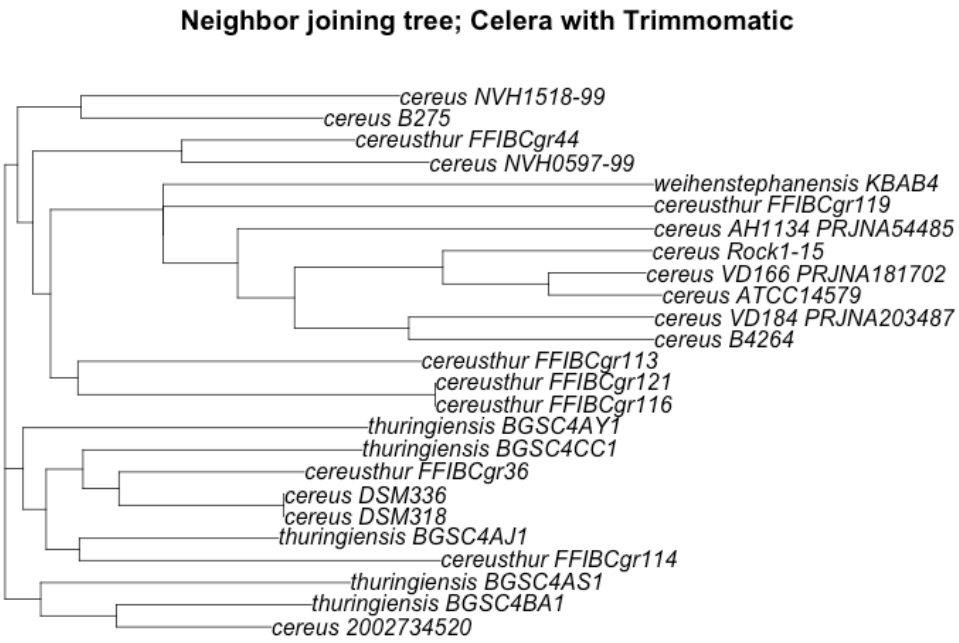
2.5.1 Neighbor joining trær

Ut fra avstandene beregnet med hamming-metoden ble det produsert trær. Ettersom matrisene var helt like for alle kombinasjonene med både SPAdes og Velvet ble også disse trærne seende identiske ut. Figur 3.5 viser treet produsert med ”neigbour joining” for Velvet og Quake, som også er representativ for resterende trær produsert med Velvet og SPAdes. Treet er rotet, og har en felles stamfar i endenoden.



Figur 3.5: Fylogenetisk tre som viser slektskap mellom 29 ulike *B. cereus* gruppe isolater . Isolatet merket i rødt antas å være *B. anthracis.*

Celera ga en redusert avstandsmatrise da flere av isolatene ble fjernet grunnet mangler på markører. Dermed ble også treet bestående av færre grener, og kombinasjonen som ga lavest antall var Celera og Trimmomatic. Treet for denne kombinasjonen fremstilles i figur 3.6.

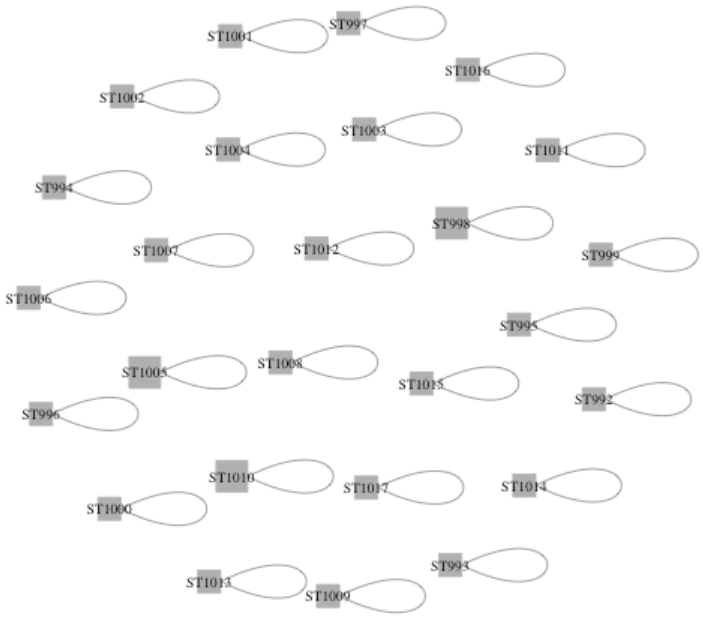


Figur 3.6. Fylogenetisk fremstilling av isolater med fullstendige markør-typer med Celera og Trimmomatic.

Den store forskjellen mellom de to trærne er grenantallet, og ved å fjerne tilsvarende gener som ble fjernet fra Trimmomatic-Celera-treet fra Quake-Velvet-treet, ble det seende identisk ut.

3.5.2 eBURST

Fylogeni ble også kartlagt med eBURST basert på Hammingavstander. eBURST ble kjørt med den sekvensmatrisen med flest markører, noe som gjaldt alle behandlet med SPAdes og Velvet. I og med at disse matrisene var like, kunne hvilken som helst være representativ også for resten av kombinasjonene med disse to assemblerings-verktøyene. Det ble valgt å bruke kombinasjonen Quake og Velvet. Her ble kommandoen kjørt med default-verdier, det vil si at cutoff var satt til 2. Dette betyr at størrelsen på grupperingene kunne ha en størrelse på maksimalt to ulike alleler, altså etterkommer(e) av grunnlegger samt etterkommers etterkommer(e). Resultatet ble ingen klonale komplekser, men heller bare singletons, som vist i figur 3.5.



Figur 3.6. Resultat fra eBURST på sekvensmatrisen for programkombinasjonen Quake og Velvet. Avstand beregnet med hammingavstand.

4. Diskusjon

Gjennom alle steg fra rådata i fastq-filer til fylogenetiske fremstillinger er det flere faktorer som kan bidra til å påvirke det endelige resultatet. I alle trinn kan parametere settes, metoder kan velges, og vurderinger tas.

4.1 Preprosessering

Gjennom preprosessering av rådata-reads fjernes reads som anses å ha dårlig kvalitet. Opptelling av antall reads etter trimming kan være en god indikator på hvor god eller hvor strengt preprosesseringsprogrammet er. I tabell 3.1 vises gjennomsnittlig antall reads per isolat etter preprosessering med hver av de tre programmene. Her ser man at Trimmomatic resulterer i flest reads i gjennomsnitt.

Da isolatene har ulik størrelse på sekvensene var det ikke hensiktsmessig kun å telle totalt antall reads og sammenlikne antallet. Ettersom utgangspunktet for antall reads i isolatene var forskjellig, ble det undersøkt samsvaret mellom antall reads i hvert isolat mellom to programmer (Quake og Nesoni samt Trimmomatic og Nesoni). I figur 3.1 fremstilles dette i et punktdiagram, og det kommer frem at gjennsomsnittlig antall reads kan være misvisende. Figuren viser en samsvarende effekt mellom programmene der man etter preprosessering, sitter igjen med omtrent likt antall reads for hvert isolat med de ulike programmene. Det er variasjon i antall reads mellom isolater, men alle tre programmer ser ut til å fjerne omtrent like mange reads i forhold til hverandre. Hvert isolat har et tilnærmet likt antall reads etter preprosessering med de tre programmene. Korrelasjonskoeffisienten, som ligger tett opp til 1, bekrefter dette og indikerer et lineært forhold mellom de to variablene, som her er programmer. For Trimmomatic og Nesoni ser det ut til å være omtrent like mange reads tilstede i hvert isolat (figur 3.1b), mens Quake ser ut til å beholde flere reads enn Nesoni (figur 3.1a). Det å forkaste mange reads kan være en god ting, da det er viktig å luke ut feil, samt usikre reads for ikke å få disse med i videre assemblering. Reads med feil kan forstyrre assembleringen og føre til sammensetninger som egentlig ikke finnes i det komplette genomet. På den andre siden er det også viktig å få med all informasjon som er nødvendig for å kunne bygge et så fullstendig genom som mulig. Det å forkaste read-sekvenser som egentlig ikke inneholder feil, og som hører til i det komplette genomet, gjør at assembleringen blir ufullstendig. Fravær av viktige read-sekvenser kan føre til huller i de assemblerte scaffold-sekvensene, og videre innføre et større antall scaffolds, ettersom sekvensene som knytter dem

sammen kan være fjernet. Huller i assemblerte sekvenser kan føre til at gener blir stykket opp, slik at det ikke er mulig å bruke disse til fylogenetiske analyser senere. Her gjelder det å finne balansen mellom det å forkaste for få reads, som vil gi feilinformasjon, og for mange reads, som vil gi for lite informasjon.

Antall parede reads for hvert preprosesseringsverktøy ble også talt opp, og gjennomsnittet samt totalt antall er å finne i tabell 3.2. Sammenliknes denne tabellen med tabell 3.1 ser man at Nesoni og Quake har omtrent like stort antall parede reads som totalt antall reads. Det betyr at disse programmene har klart å holde på read-parene. Trimmomatic derimot har et lavere antall parede reads enn det totale antallet. Trimmomatic har dermed forkastet reads som er deler av par, slik at flere reads har mistet sin partner. Det kan være en fordel å beholde flest mulig reads som par, da informasjonen kan brukes når det skal dannes scaffolds fra contig-fragmenter. Med bakgrunn i disse resultatene, er det rimelig å anta at assembleringer gjort med Quake og Nesoni vil ha en fordel da disse har så å si alle sine reads i par og kan benytte seg av denne informasjonen. Ser en videre på figur 3.2 kommer det tydeligere frem for hvilke isolater de ulike preprosesseringsprogrammene har klart å beholde de parede reads-sekvensene. Figur 3.2a viser en sammenlikning av Quake og Nesoni, og det viser seg at Quake ikke i like stor grad klarer å ta vare på read-par i isolater med mange reads. Dette bekreftes av R-verdien som er noe lavere for dette plottet. Det viser seg i figur 3.2b at Trimmomatic og Nesoni er omtrent like i antall read-par ettersom totalt antall reads for isolatet økes. Trimmomatic tok totalt sett vare på flest reads etter trimming, men det viste seg at mange av dem ikke lenger var en del av et par, noe som kan være en ulempe senere.

4.2 Vurdering av assembleringskvalitet

Etter assemblering måtte kvaliteten på dette steget for alle kombinasjoner av preprosesserings- og assembleringsprogrammer sjekkes. Dette ble gjort på flere forskjellige måter, men i hovedsak ved sammenlikning til referansegenom.

4.2.1 Sammenlikning av assemblerte sekvenser med eksisterende genom fra samme stamme

For å få en god oversikt over hvor vellykket de ni forskjellige programkombinasjonene gjorde det, ble det først tatt utgangspunkt i det isolatet der det allerede var et fullt assemblert genom tilgjengelig. For stammen *Bacillus cereus* ATCC14579 er et fullstendig genom publisert, noe som her kunne brukes til å evaluere egne resultater for nettopp dette isolatet. Scaffold-filene

fra alle ni kombinasjoner i programmet ble derfor blastet mot det fullstendige genomet for denne stammen. Dekningsgraden, det vil si hvor stor andel av referansegenomet som dekkes av scaffold-sekvensene, vil dermed kunne gi en god indikasjon på hvor vellykket assembleringen var. Dette ble gjort ved å bruke de scaffold-sekvensene med best bit score. En perfekt assemblering ville gitt fullstendig dekning av referansegenomet, og en dekningsgrad på 100%. Som vist i tabell 3.3 er dette ikke tilfelle. Alle assembleringer gjort med preprosesseringsprogrammet Trimmomatic ga jevnt over lavest dekningsgrad. Dette kan bety at det å ta vare på parede reads spiller en rolle for assembleringen. Det er også mulig at Trimmomatic har tatt med for mange reads, og vært for snill i trimmingen, slik at ikke alle read-sekvenser med lav kvalitet og feil har blitt fjernet. Sekvenser med lav kvalitet, som burde vært fjernet fra read-sekvensene kan ha gitt opphav til feilaktige koblinger mellom reads og contigs. Resultatene kan bety at de read-sekvensene Trimmomatic tok vare på sammenliknet med Quake og Nesoni (som ikke lenger tilhører noe par), fører til mer rot i assembleringen enn de er til nytte.

Ser man på dekningsgrad sammenliknet med økt verdi av k i k-merene for kombinasjonene som innvolverer Velvet, er det en økning i dekningsgrad med økt k-mer-lengde. Generelt ser det ut til at økt k-mer-lengde gir økt dekning, og antall scaffold-sekvenser blir færre. Dette indikerer både en mer riktig, og en mer fullstendig assemblering, da dekningen viser at sekvensen er riktig, og et lavt antall scaffolds betyr at flere reads har blitt satt sammen.

Det kommer også frem at lavt antall scaffolds ikke nødvendigvis betyr at assembleringen har vært mer vellykket. Celera har i kombinasjon med alle preprosesseringsprogrammene færrest scaffold-sekvenser, men er også den som gjør det dårligst med tanke på dekningsgrad. Dette kan bety at, til tross for en sammenhengende sekvens, kan det være feil på visse områder i scaffold-sekvensen, noe som vil vise seg når den sammenstilles med et fullstendig genom fra samme stamme. Derimot er det viktig å huske på at en må kreve en viss reduksjon i antallet fra reads til scaffolds. Dersom antall scaffolds er svært stort, vil det si at svært få reads i det hele tatt er satt sammen til lengre sekvenser. En kan tenke seg at dersom ingen reads er assemblert sammen, vil en sammenstilling mot et referansegenom fremdeles gi høy dekning, da de fleste de små fragmentene vil finne et matchende sted i genomet. Dekningen vil dermed bli misvisende høy, og vurderingsmetoden blir ikke pålitelig. Det beste resultatet vil en trolig oppnå med en balanse mellom høy dekning og lavt antall scaffols-sekvenser.

Det er forventet at SPAdes gir god dekning selv uten et preprosesseringsprogram ettersom den har en innebygget kvalitetssikring. Videre gjør også Celera og Velvet det bedre uten noen form for forbehandling. Dette kan bety at preprosesseringsprogrammene trimmer vekk viktig informasjon som kunne ha blitt brukt i assembleringen. Samtidig er det viktig å ta antall scaffold-sekvenser med i betraktningen, og en ser at Velvet uten preprosessering med en k-verdi lik 31 gir et svært stort antall. Dette kan, som diskutert tidligere, føre til misvisende dekningsgrad. Derimot reduseres antall scaffold-sekvenser ved økt k-verdi. Dette kan bety at lengre k-mers veier opp for effekt av ingen preprosessering.

Celera i kombinasjon med Trimmomatic resulterer i lavest dekningsgrad. Det kan skyldes at Trimmomatics mange reads byr på utfordringer når Celera sammenstiller alle reads med hverandre. Generelt er det Celera som kommer dårligst ut av assembleringsprogrammene i følge disse resultatene. Dette tyder på at OLC algoritmen passer dårligst til denne typen datasett. Historisk er OLC algoritmen best egnet til lengre sekvenser, noe en får ved Sanger-sekvensering. Derimot kan det se ut til at den nye NGS-teknologien drar mer nytte av DBG-baserte algoritmer i og med at lengden på reads er kortere.

4.2.2 Sammenstilling av assembleringer mot referansegenom

Ettersom det ikke finnes fullstendige assemblerte genomer publisert for de resterende isolatene ble det brukt et referansegenom, som i utgangpunktet var tenkt at skulle være svært likt. Til dette ble *Anthracis Ames Ancestor PRJNA58083* benyttet, og dekningsgrad ble beregnet for alle isolater mot dette genomet. I første omgang ble dekningsgraden beregnet ut fra det relativt strenge kravet der kun scaffold-sekvenser med best bit score fra sammenstillingen ble tatt med. Dette førte totalt sett til lave dekningsverdier. Likevel var det interessant å undersøke om noen av kombinasjonene gjorde det bedre enn andre, til tross for lav dekningsgrad. Det ble utført en mixed model statistikk der resultatet er presentert i tabell 3.4. Den forventede dekningsgraden er svært lav, som det fremgår av referanse-raden i første kolonne under faste effekter, her med en verdi på 23,4%. Dette er forventet dekning for Trimmomatic i kombinasjon med Celera, som her fungerer som en slags referanseverdi før bytte til et annet program. Resultatet betyr at de scaffold-sekvensene som ga best resultat fra blastingen kun dekker 23,4% av hele genom-sekvensen når de sammenstilles. Den forventede dekningsgraden øker mest ved bytte til et annet preprosesseringsprogram. At endringen i den forventede dekningsgraden skyldes endring i preprosessering, er også svært signifikant, noe som fremkommer av den høye t-verdien (signifikant ved $t>3$). Å benytte seg av

assembleringsprogrammet SPAdes gir en endring i forventet dekning, men den kan ikke sies å være signifikant. Dette betyr at det er en forskjell i dekning ved bytte fra Celera til SPAdes, men man kan ikke sikkert påstå at forskjellen skyldes valg av program. Et bytte til Velvet derimot ser ut til å gi økning i forventet dekningsgrad, der en med bakgrunn i den signifikante t-verdien, kan si at økningen skyldes nettopp endring i program. Denne jevnt over svært lave dekningsgraden kan skyldes det strenge kriteriet med at kun scaffold-sekvensene med best bit score blir tatt med i beregningen. Likevel skal det også nevnes at til tross for at bakteriegenomene i prinsippet skal være svært like, er dette nødvendigvis ikke tilfellet. Den lave dekningsgraden kan også ha biologiske forklaringer i at genomene rett og slett er ulike. Bakterier har en rask evolusjon, og deres tilpasning til miljøet skjer hurtig. *B. cereus*-gruppen kan derfor ha ulike tilretteleggingen i genomene sine avhengig av hvilke miljøer de er hentet fra. Ettersom isolatene i dette datasettet var hentet fra ulike geografiske steder og fra ulike kilder er det ingen stor overraskelse at dekningsgraden varierer. Ikke engang et perfekt assemblert genom ville nødvendigvis fått 100% dekningsgrad mot referansegenomet. Det kan derfor være vanskelig å trekke sikre slutninger når et referansegenom fra en annen stamme er benyttet til sammenlikning.

Med det strenge kravet som ga opphav til svært lav dekningsgrad tatt i betraktning, ble det besluttet å senke kravet noe og selektere ut de scaffold-sekvenser som skulle være med i beregningen på en annen måte. Det ble bestemt å inkludere alle scaffold-sekvenser der sammenstillingen med referansegenomet oversteg en viss lengde. Ettersom et bakterie-operon har lengde på omtrent 5000bp var det nærliggende å velge en minimumslengde på sammenstillingen med denne lengden. På denne måten kan man håpe at dekningen får med seg et helt sett av gener. Igjen ble det beregnet dekningsgrad for alle isolater og det ble forsøkt å finne en forskjell i programkombinasjonene med en mixed model analyse. Tabell 3.5 viser at den forventede dekningsgraden ved bruk av Trimmomatic og Celera nå har økt til 82,4%. Økningen er forventet da man nå har åpnet for at et større antall sekvenser i hvert scaffold er med på å beregne dekningsgraden. Her vil bytter til både Nesoni og Quake samt SPAdes og Velvet gi signifikante endringer i forventet dekningsgrad, selv om endringen i praksis er så liten at den ikke betyr stort. Ut fra akkurat disse resultatene kan det tenkes at valg av program ikke har den store innvirkningen på den endelige sekvensen.

4.2.3 Dekning som et resultat av lengden på sammenstillingen

Ettersom en forventet dekningsgrad på omtrent 82% fremdeles er ganske lav ble det besluttet å variere kravet om lengden på sammenstillingen. Lengden ble prøvd utvidet til 10 000bp for å undersøke i hvor stor grad dekningsgraden ble redusert. Det ble også gjennomført med et mildere krav på 1000bp. Sammenhengen mellom dekningsgrad og lengde på sammenstilling illustreres av figur 3.3. Der kommer det tydelig frem at for de fleste isolater vil økt krav til sammenstillingslengde også føre til lavere dekningsgrad. Dette betyr at færre scaffold-sekvenser har en fullstendig lengde som kan finnes igjen i referansegenomet når lengdekravet økes, slik at mindre deler av dette genomet dekkes av disse sekvensene. Igjen kan dette skyldes biologiske faktorer da scaffold-sekvensene fra isolatene og genomet som ble brukt til sammenlikning ikke er fra samme stamme. Nok en gang kan det være at scaffold-sekvensene ikke klarer å sammenstilles med en høyere prosent grunnet faktiske ulikheter i genomene, og at det ikke skyldes dårlig preprosessering og feil i assembleringen. Som forventet ble dekningsgraden høyere da grensen gikk ved 1000bp, noe som betyr at flere sammenstillinger har en lengde på mellom 1000 og 5000 basepar. Videre ser man at det å flytte grensen fra 1000 til 10 000 gir en effekt som er omtrent parallell for alle isolater. Dette kan bety at de fleste isolater har lengre eller flere sekvenser som likner hverandre. Når en øker lengdekravet vil isolatene bli for ulike, og dekningen reduseres. Man kan tro at det er de samme sekvensene som får treff ved kortere sammenstillinger.

Et av isolatene derimot, får svært god dekningsgrad med referansegenomet. Stammen *Bacillus cereus* FFIBC114 får tett opptil 100%, med krav om sammenstilling på over både 1000bp, 5000bp og 10 000bp. Dette må bety at de sammenstillingene som finnes for dette isolatet er lange sammenstillinger som finnes igjen i store deler av referansegenomet. I blastingen fikk også dette isolatet svært høy dekningsgrad med referansegenomets to plasmider, noe som tyder på at isolatet er bærer av tilsvarende plasmider. Dette er plasmidene pXO1 og pXO2, og kan dermed antas å være en *B. anthracis.* Det som også observeres i figur 3.3 er at *B. cereus* ATCC14579 får lav dekning sammenliknet med referansegenomet. Dette er ikke uventet, da det i tidligere fylogenetiske undersøkelser har vist seg at denne stammen har en lengre evolusjonær avstand til *B. anthracis*-stammene (Olsen et al., 2007). God dekning får isolatet *B. cereus* 2000031002, noe som også stemmer godt overens med litteraturen, der denne stammen viser seg å være nært beslektet med *B anthracis* (Kolsto et al., 2009).

4.2.4 N50

N50 verdien skal si noe om lengden på de lengste scaffold-sekensene, og en lang scaffold-sekvens er det ønskelige. Dette betyr at flere reads er samlet i lengre sekvenser, og med et ideelt sammensatt genom ville denne verdien vært lik antall basepar totalt i genomet. Som vist i tabell 3.6 gir kombinasjonen Trimmomatic-Celera høyest N50. Dette samsvarer med tabell 3.3, der det også viser seg at denne kombinasjonen produserer lavest antall scaffold-sekvenser. En signifikant reduksjon forekommer i forventet N50-verdi dersom en bytter assembleringsprogram. Ut fra kun N50-verdien ville det vært naturlig å anta at Celera er det programmet som utførte den beste assembleringen. Likevel, som det har kommet frem i tabell 3.4 og 3.5, gir Celera lave deknings-verdier slik at kun en N50-vurdering i seg selv er ikke nok til å evaluere graden av suksess for assembleringen. Til tross for at dette programmet gir lengre scaffold-sekvenser, kan det hende den assemblerer feil, noe som vil komme frem i dekningsgraden.

4.3 Søk etter MLST-markører

MLST-markører er husholdningsgener som skal finnes i alle stammer innen den aktuelle arten. Det å evaluere hvor vellykket assembleringen har vært med bakgrunn i hvor mange markører som kan finnes igjen i scaffold-sekvensene er derfor en effektiv metode. Med riktig assemblerte scaffold-sekvenser vil gen-sekvensen ha blitt puslet sammen fra read-sekvensene, og markør-sekvensen vil kunne finnes igjen. Ved å sammenstille scaffold-sekvensene med markør-sekvensene vil det dermed komme frem om markørene finnes i de assemblerte sekvensene. Som vist i figur 3.4 er det ingen av programkombinasjonene som er kapable til å assemblere scaffold-filer som inneholder alle de utplukkede MLST-markørene. Eventuelt kan markør-genene mangle i de opprinnelige genomene. Likevel er det tydelig at noen av kombinasjonene gjør det bedre enn andre når det gjelder å assemblere sammen disse sekvensene. Alle kombinasjoner med Velvet og SPAdes får samme resultat, der to isolater mangler hver sin markør. Dette er samme isolater og samme markører for alle preprosesseringsprogrammer for disse to assembleringsprogrammene, noe som kan tyde på at isolatet ikke var bærer av disse markørene i utgangspunktet. Ettersom gode assemblerte scaffold-sekvenser skal inneholde markørene vil en med disse resultatene kunne si at Celera, med alle sine kombinasjoner av preprosesseringsprogram kom dårligst ut. Dette assemblerings-verktøyet mangler 4, 9 og 12 markører til sammen for alle isolater med henholdsvis Quake, Nesoni og Trimmomatic. I henhold til tabell 3.7 er det er flere isolater som mangler flere markører. Spesielt stammen *Bacillus cereus* 2000031002 har på det meste

seks manglende markører når Celera er brukt i kombinasjon med Trimmomatic. Det at denne kombinasjonen gjør det såpass dårlig samsvarer til en viss grad med tidligere resultater, da Trimmomatic etterfulgt av Celera har vist seg å jevnt over være den dårligste kombinasjonen. En mulig forklaring er manglende evne til å ta vare på par for Trimmomatic sin del, som kan ha innvirkning hos Celera. Det kan også være at det store antallet reads Trimmomatic sitter igjen med etter assemblering fører med seg feil som videreføres i assembleringen. Mangel på markør gjør at sekvenstypen for det aktuelle isolatet ikke blir komplett, noe som gjør det umulig å beregne avstand til andre sekvenstyper. Dette ble løst ved å ta ut de isolatene med manglende markører, før det ble beregnet avstand. Det betyr følgelig at for de ulike kombinasjonene sitter en igjen med et varierende antall isolater å utføre fylogenetiske analyser på.

Metoden brukt for å finne igjen markører i scaffold-sekvensene kan videre diskuteres, ettersom den kan sies å være relativt streng. Det ble brukt en ungapped blast-sammenstilling i søket etter markørene, noe som betyr at hele lengden til markør-sekvensen skal matche eksakt med en tilsvarende lengde i scaffold-sekvensen. På mange måter kan dette være et noe strengt krav da det, til tross for mangel på hele lengden til markøren, kan være at store deler av markøren finnes igjen i scaffold-sekvensene. Det kan også forekomme enkle substitusjoner i enkelte posisjoner, noe en kan vurdere om til en viss grad kan tillates. Kun raske manuelle undersøkelser av disse manglende markørene ble gjennomført. Det viste seg at for de manglede markørene var det kun en del av markøren ble funnet igjen, mens den resterende var ikke å finne. Dette kan skyldes mislykket assemblering, og dette kan være noe å arbeide mer med. Videre kan det vurderes om indekseringen av alleltype kunne blitt bestemt på en annen måte. I metoden med indeksering av tall for hvert ulikt allel, vil tallverdiene bli satt uavhengig av hvor stor forskjellen er mellom sekvensene. Dette betyr i praksis at to allel-sekvenser som kun er ulike i en base kan få indeksene 1 og 2. En tredje sekvens som skiller seg fra de to andre i ti baser vil kunne få indeksen 3. Denne indekseringen forteller ingen ting om *hvor* ulike allel-sekvensene er. En fornuftig tanke ville være at to isolater med alleltypene 1 og 2 ville være likere hverandre enn to isolater med eksempelvis alleltypene 1 og 3. Dette er informasjon en ikke kan benytte seg av ved å bruke denne metoden. En mulighet er å bruke allel-sekvensen til sammenlikning, istedenfor kun allelprofilen. Da vil graden av ulikhet i sekvensene komme tydelig frem, og en kan undersøke slektskap mer nøyaktig på denne måten. Likevel behøver ikke variasjon i kun et basepar bety nærere slektskap enn variasjon i en lengre sekvens. Rekombinasjon kan introdusere flere punktmutasjoner, som ville resultert i

en endring i en lengre sekvens. Dette kan ved å sammenlikne hele allel-sekvenser oppfattes som flere biologiske hendelser, selv om det i realiteten kun er en biologisk hendelse. Dermed kan den evolusjonære avstanden feilberegnes dersom en ukritisk antar at flere mutasjoner er ekvivalent med lengre evolusjonær avstand.

4.4 Fylogenetisk analyse

4.4.1 Fylogenetiske trær

Figur 3.5 viser det fylogenetiske treet produsert etter bruk av programmene Quake og Velvet. Dette treet er representativt for alle kombinasjoner med SPAdes og Velvet, ettersom de alle hadde identiske avstandsmatriser. Dette tilsier at forskjellen i forventede verdier med de ulike programmene brukt, mest sannsynlig ikke har noen innvirkning på det endelige resultatet. I forhold til kombinasjonene til Celera er det færre grener i treet grunnet færre isolater med markører funnet i sekvensene. Dersom isolatene med manglende markører i Trimmomatic-Celera-kombinasjonen også fjernes fra de andre kombinasjonene vil dette videre føre til identiske trær. Dette tyder på at avstandsmatrisene er like, og at det sånn sett ellers er likhet i assembleringen på disse områdene.

4.4.2 eBURST

Resultatet av eBURST analysen på isolatene behandlet med Quake og Velvet ga, som vist i figur 4.5, kun singletons. For at en parallell skal kunne trekkes mellom to sekvenstyper i en slik analyse kan de respektive sekvenstypene til isolatene kun være ulike ved ett allel. Dersom flere alleler er forskjellige vil ikke eBURST trekke en fylogenetisk sammenheng mellom de to isolatene. Dette kan man på mange måter si er en streng grense å sette. Her blir det sett på hele fjorten markører, slik at det kunne vært mulig å regulere antall tillatte ulikheter i allel-matrisene. Vanligvis linkes to isolater dersom de har maksimalt en ulikhet, men i dette tilfellet kunne det blitt vurdert å tillatt flere. Også her må en ta i betraktning at det ikke finnes noen informasjon om *hvor* ulike allel-sekvensene er. Det er også viktig å huske på at fjorten gener er et svært lite antall i forhold til det totale antallet hos bakterier. Gjennomsnittlig antall gener hos en bakterie er 5000. Det å se kun på forskjellen i fjorten av dem vil ikke nødvendigvis gi all informasjon, da de kan være like i disse fjorten markørene, men skilles fra hverandre i andre deler av genomet.

4.4.3 Fylogeni hos *B. cereus*

Dendrogrammet produsert av MLST-markørene i figur 3.5 viser at stammene DSM336 og DSM318 har identiske allelprofiler, noe som stemmer godt overens med resultater fra liknende analyser (Olsen et al., 2007). I nevnte artikkel kommer det frem at disse stammene er nærest beslektet *B. anthracis*. Som figur 3.5 viser opptrer de også i denne oppgaven nær stammen som kan antas å være en *B. anthracis*, FFIBCgr114. I denne oppgaven havner stammen BGSC4AJ også nær den antatte *B. anthracis* stammen, noe som stemmer med tidligere undersøkelser (Kim et al., 2005) der denne stammen ble plassert som en nær nabo av *B. anthracis.* Videre plasseres stammen 2000031002 som den aller nærmeste naboen til *B. anthracis*-stammen FFIBCgr114. Dette er også observert i tidligere studier der det kommer frem at dette isolatet er et av de nærest beslektede stammene til *B. anthracis* og dermed en viktig bidragsyter til å forstå denne artens evolusjon (Kolsto et al., 2009). I samme artikkel bekreftes også naboforholdene mellom stammene DSM336, DSM318 og 2002734520. Stammen BGSC4AY1 får også en plassering som tilsvarer hva som er funnet av Kolsto et. al. (2009). FFIBCgr36 havner her som nærmeste nabo til DSM336, DSM3 og FFIBCgr114. Dette kan bety at også denne er nært beslektet *B anthracis*. I figur 3.5 ser en en gruppering av blant annet KBAB4, AH3411, B4264 og ATCC14579, noe som også er tilfelle i studier av Olsen et al. (2007). I tillegg er det samsvar med nært slektskap mellom BGSC4AU1 og BGSC4AS1. I denne oppgaven ser det ut til at disse to har identiske alleltyper, noe som ikke stemmer overens med tidligere studier. Dette kan skyldes metodene benyttet i denne oppgaven.

Resultatene i oppgaven tilsier at metodene brukt her kan være brukbare verktøy for å kartlegge slektskap hos *B. cereus* gruppen. De ulike verktøyene brukt ga ikke store utslag i forhold til ulike grupperinger. Den største forskjellen i trærne er som tidligere nevnt antall grener, som følge av antall manglende markører funnet i isolatene.

4.5 Konklusjon

Gjennom arbeidet i denne oppgaven ble det undersøkt om valg av programmer og innstillinger hadde innvirkning på det endelige resultatet i prosessen fra rådatafiler til fylogenetisk fremstilling. Det viste seg at valg av preprosesseringsprogram kunne for noen analyser føre til feilaktige og ubrukelige resultater. Programmene SPAdes og Velvet som ble testet fikk omtrent identiske resultater, og kan dermed sier å være jevngode. Begge er programmer som benytter seg av DBG. Celera som bruker en OLC-aloritme har jevnt over

gjort det noe dårligere, og det kan på dette grunnlaget konkluderes at, det er DBG-assemblering som passer best til data som er fremstil ved hjelp av Illumina Miseq platformen. Videre er det erfart at det å bestemme graden av suksess for en assemblering er svært vanskelig, da det finnes mange kriterier som kan brukes i evalueringen. Den beste metoden vil mest sannsynlig være å kombinere flere. MLST er en velfungerende metode dersom en skal skissere evolusjonært slektskap. Derimot tar denne metoden kun for seg et lite antall gener, slik at en ikke får kartlagt forskjeller i resten av genomet. Det endelige produktet, som her var de fylogenetiske trærne, var like resultater ved bruk av SPAdes og Velvet, og noe mer manglende hos Celera. Preprosesseringsprogram hadde noe innvirkning på resultatet, men det var assembleringsprogrammene som hadde størst grad påvirket. Dette betyr i sin tur at valg av programmer ikke nødvendigvis trenger å være avgjørende for hvordan resultatet blir, men om man skulle velge en kombinasjon som ikke fungerer optimalt, kan dette få konsekvenser for resultatene.

Dette kan kort oppsummeres i følende punkter:

- valg av preprosesseringsprogram kan ha betydning for videre analyser
- det å ikke gjennomføre preprosessering kan gi god assemblering
- assembleringer med DBG-programmer er mer egnet til data med korte reads enn assembleringer med OLC-programmer
- MLST fungerer godt når evolusjonært fylogeni skal kartlegges, men fremdeles er det biologiske forskjeller i genomer som ikke blir registrert
- evolusjonære avstander varierer ikke stort ved bruk av de ulike bioinformatiske verktøyene

4.6 Videre arbeid

I denne oppgaven er det flere ting det er mulig å jobbe videre med. I alle steg i prosessen fra rådata til en fylogenetisk fremstilling av isolatene er det mulig å variere både parametere og behandlingsmetoder av data. I forhold til preprosesseringen av read-filene er det en rekke parametere som kan endres på. Valg av kvalitetsgrense på base-sekvensen for klipping av reads kan justeres både strengere og mildere, noe som videre kan ha innvirkning på assemblering. I denne oppgaven ble alle programmer kjørt med standard (default) innstillinger og disse kan enkelt endres og sammenliknes med resultatene funnet her. Blant annet ble Trimmomatic i denne oppgaven kjørt med standardinnstillingen som bruker funksjonen ”glidende vindu”. En mulighet er å kjøre programmet med spesifikasjonen ”maksimal

informasjon", som er mer tilpasningsdyktig i forhold til å balansere lengde på read og sannsynlighet for feil. I denne funksjonen settes parameteren s, slik at brukeren selv kan bestemme hva som er viktigst; en lang read eller en tilnærmet feilfri read. For å bestemme best mulig s, kan Trimmomatic kjøres flere ganger med ulike s-verdier.

Videre kan det gjøres justeringer i assembleringssteget. Her opererte Velvet med den satte k-verdien 151. Da Velvet ble kjørt med de tre k-verdiene 31, 91 og 151 så en tydelig forbedring i antall scaffolds ved økning av k, slik at det kunne vært interessant å prøve med enda høyere k-verdi.

For å vurdere kvaliteten på assembleringene ble det blant annet sammenstilt scaffold-sekvenser fra *Bacillus cereus* ATCC14579 mot det ferdig assemblerte genomet fra denne stammen. Dette er en ideell måte å undersøke effekten av selve metoden, uten store påvirkninger av biologiske forskjeller. De andre tjuefire isolatene ble sammenliknet mot en annen bakteriestamme, noe som kan være egnet dersom en ønsker å kartlegge likheter mellom disse. I forhold til metodetesting er dette derimot ikke helt ideelt. Genetiske variasjoner mellom stammene er mest sannsynlig grunnen til lave dekningsverdier. For å på best mulig måte unngå de biologiske effektene burde scaffold-sekvensene fra hvert isolat sammenstilles med et genom som er mest mulig likt seg selv. En større jobb ville da blitt å søke gjennom alle *Bacillus*-stammer publisert hos NCBI, og for hvert isolat velge det genomet med best match. På denne måten blir det lettere å se metodeeffekten, når alle isolater sammenstilles mot et eget genom med allerede store likheter. Som en tilnærming til dette kunne en sammenstilt isolater en forventer at vil gruppere seg i nærheten av *B anthracis* med *anthracis*-genomet, og fjernere slektninger kunne blitt sammenstilt mot stammen ATCC14579.

Ved å legge sammen kortere sammenstillinger for å regne ut den totale dekningsgraden bringer med seg noen usikkerheter. Selv om dekningen totalt i en slik sammenstilling er høy, betyr ikke dette at sekvensene nødvendigvis ligger i riktig rekkefølge. En måte å sikre dette på er ved å sjekke start- og slutt-posisjon til hit-sekvensene og se til at de ligger etter hverandre med stigende rekkefølge. Eventuelt om det er match på den omvendte tråden i synkende rekkefølge. Noe som eventuelt bør sjekkes før dette gjennomføres er at alle match-sekvenser befinner seg på den samme tråden, slik at ikke noen havner på forover-tråden, og andre på revers. Det gir misvisende dekningsresultater.

I søket etter MLST-markører ble det tidligere nevnt at en mulighet var å kjøre blast med et mildere krav, som åpner for at sekvensen som sammenstilles markøren ikke behøver å ha nøyaktig samme lengde, og at det tillates gaps. Dersom dette hadde blitt gjort på alle scaffold-sekvenser mot markørene er det mulig at sekvenstypene hadde sett noe annerledes ut. I tillegg er det mulig at sekvenser som forble uoppdagede i denne oppgaven hadde kommet frem ved bruk av en slik tilnærming.

Reference list

ALTSCHUL, S. F. G., WARREN; MILLER, WEBB; MYERS, EUGENE; LIPMAN, DAVID 1990. Basic Local Alignment Search Tool. *Journal of Molecular Biology*, 403-410.

BANKEVICH, A., NURK, S., ANTIPOV, D., GUREVICH, A. A., DVORKIN, M., KULIKOV, A. S., LESIN, V. M., NIKOLENKO, S. I., PHAM, S., PRJIBELSKI, A. D., PYSHKIN, A. V., SIROTKIN, A. V., VYAHHI, N., TESLER, G., ALEKSEYEV, M. A. & PEVZNER, P. A. 2012. SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. *J Comput Biol,* 19, 455-77.

BARTHOLOMEW, J. M., TODD 1952. The Gram Strain. *Bacteriological Reviews,* 16, 1-29.

BISHOP, Ö. T. 2014. *Bioinformatics and Data Analysis in Microbiology,* Norfolk, UK, Caister Academic Press.

BOLGER, A. M., LOHSE, M. & USADEL, B. 2014. Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics,* 30, 2114-20.

FEIL, E. J., LI, B. C., AANENSEN, D. M., HANAGE, W. P. & SPRATT, B. G. 2004. eBURST: Inferring Patterns of Evolutionary Descent among Clusters of Related Bacterial Genotypes from Multilocus Sequence Typing Data. *Journal of Bacteriology,* 186, 1518-1530.

FIEDLER, J. 2004. HammingCodes.

HELGASON, E., TOURASSE, N. J., MEISAL, R., CAUGANT, D. A. & KOLSTO, A. B. 2004. Multilocus Sequence Typing Scheme for Bacteria of the Bacillus cereus Group. *Applied and Environmental Microbiology,* 70, 191-201.

HTTP://BIOINF.SPBAU.RU/SPADES.

HTTP://BLAST.NCBI.NLM.NIH.GOV/BLAST.CGI.

HTTP://PUBMLST.ORG/BCEREUS/. 2015.

HTTP://WGS-ASSEMBLER.SOURCEFORGE.NET/WIKI/INDEX.PHP?TITLE=MAIN_PAGE.

HTTPS://http://WWW.EBI.AC.UK/~ZERBINO/VELVET/.

KELLEY, D. R., SCHATZ, M. C. & SALZBERG, S. L. 2010. Quake: quality-aware detection and correction of sequencing errors. *Genome Biol,* 11, R116.

KIM, K., SEO, J., WHEELER, K., PARK, C., KIM, D., PARK, S., KIM, W., CHUNG, S. I. & LEIGHTON, T. 2005. Rapid genotypic detection of Bacillus anthracis and the Bacillus cereus group by multiplex real-time PCR melting curve analysis. *FEMS Immunol Med Microbiol,* 43, 301-10.

KOLSTO, A. B., TOURASSE, N. J. & OKSTAD, O. A. 2009. What sets Bacillus anthracis apart from other Bacillus species? *Annu Rev Microbiol,* 63, 451-76.

LI, Z., CHEN, Y., MU, D., YUAN, J., SHI, Y., ZHANG, H., GAN, J., LI, N., HU, X., LIU, B., YANG, B. & FAN, W. 2012. Comparison of the two major classes of assembly algorithms: overlap-layout-consensus and de-bruijn-graph. *Brief Funct Genomics,* 11, 25-37.

MAIDEN, M. B., JANE; FEIL, EDWARD; MORELLI, GIOVANNA; RUSSEL, JOANNE; URWIN, RACHEL; ZHANG, QING; ZHOU, JIAJI; CAUGANT, DOMINIQUE; FEAVERS, IAN; ACHTMAN, MARK; SPRAT, BRIAN 1998. Multilocus sequencing typing:A portable approach to the identification of clones within populations of patogenic microorganisms. *Proceedings of the National Academy of Sciences of the United States of America,* 95, 3140-31-45.

MILLER, J. R., KOREN, S. & SUTTON, G. 2010. Assembly algorithms for next-generation sequencing data. *Genomics,* 95, 315-27.

OLSEN, J. S., SKOGAN, G., FYKSE, E. M., RAWLINSON, E. L., TOMASO, H., GRANUM, P. E. & BLATNY, J. M. 2007. Genetic distribution of 295 Bacillus cereus group members based on adk-screening in combination with MLST (Multilocus Sequence Typing)

used for validating a primer targeting a chromosomal locus in B. anthracis. *J Microbiol Methods,* 71**,** 265-74.
PINHEIRO, H. P., DE SOUZA PINHEIRO, A. & SEN, P. K. 2005. Comparison of genomic sequences using the Hamming distance. *Journal of Statistical Planning and Inference,* 130**,** 325-339.
SABAT, A. J. B., A; NASHEV, D; SÁ-LEÃO, R; VAN DIJL, J M; GRUNDMANN, H; FRIEDRICH, A W; 2013. Overview og molecular typing metods for outbreak detection and epidemiological surveillance. *Euro Surveill,* 18.
SAITOU, N. N., MASATOSHI 1987. The neighbor-joining Method: A New Method for Reconstructing Phylogenetic Trees. *Mol. Biol. Evol,* 4**,** 425.
SANGER, F. N., S; COULSON, AR 1977. DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences of the United States of America,* 74**,** 5463-5467.
SNIPEN, L. & LILAND, K. H. 2015. micropan: an R-package for microbial pan-genomics. *BMC Bioinformatics,* 16**,** 79.
SPADES 2014. SPAdes de novo assembler: User manual.
TORTORA, G. J. F., BERDELL R.; CASE, CHRISTINE L 2010. *Microbiology - An Introduction*, Pearson Education.
WOESE, C. R. 1987. Bacterial Evolution. *Microbiological Reviews,* 51**,** 221-271.
ZERBINO, D. R. 2010. Using the Velvet de novo assembler for short-read sequencing technologies. *Curr Protoc Bioinformatics,* Chapter 11**,** Unit 11 5.
ZERBINO, D. R. & BIRNEY, E. 2008. Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome Res,* 18**,** 821-9.
ØKSTAD, O. A. K., ANNE-BRIT. 2011. *Genomics of Foodborne Bacterial Pathogens*.
ØKSTAD, O. A. K., ANNE-BRIT. 2012. Chapter 6, Evolution of the Bacillus cereus Group. *In:* SANSINENEA, E. (ed.).

Norges miljø- og
biovitenskapelige
universitet

Postboks 5003
NO-1432 Ås
67 23 00 00
www.nmbu.no