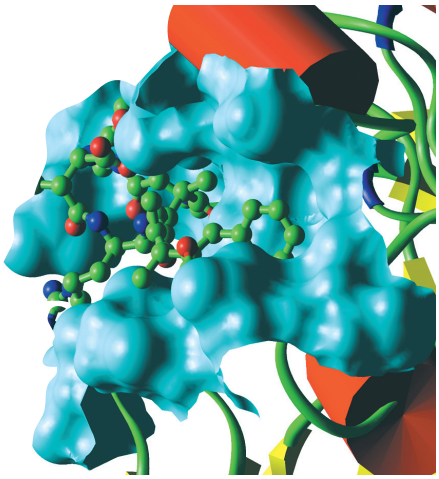


SEKVENSSIANALYYSIOPAS

KIMMO MATTILA, ARI-MATTI SARÉN, JARNO TUIMALA JA EIJA KORPELAINEN
CSC

Sekvenssianalyysiopas

Kimmo Mattila
Ari-Matti Sarén
Jarno Tuimala
Eija Korpelainen

Tieteen tietotekniikan keskus CSC

Tämän teoksen tekijänoikeudet kuuluvat CSC – Tieteellinen laskenta Oy:lle. Teoksen tai osia siitä voi kopioida ja tulostaa vapaasti henkilökohtaiseen käyttöön sekä Suomen yliopistojen ja korkeakoulujen kurssikäyttöön edellyttäen, että kopioon tai tulosteeseen liitetään tämä ilmoitus teoksen tekijästä ja tekijänoikeuksista. Teosta ei saa myydä tai sisällyttää osaksi muita teoksia ilman CSC:n lupaa.

© Kimmo Mattila, Ari-Matti Sarén, Jarno Tuimala, Eija Korpelainen ja
CSC – Tieteellinen laskenta Oy
2006

1. painos

ISBN 952-5520-21-8

http://www.csc.fi/csc/julkaisut/oppaat/

Painopaikka:
Picaset Oy
Helsinki 2006

Lukijalle

Tämä opas on tehty johdannoksi CSC:n sekvenssianalyysiohjelmien ja tietokantojen käyttöön. Oppaassa esitellään CSC:n www:n ja unix-ympäristöt sekä käydään läpi CSC:n käytetyimpien sekvenssianalyysiohjelmien alkeet. Opas ei sisällä kattavaa kuvausta mistään biolaskennan aihepiiristä, vaan sen tarkoitus on auttaa uutta käyttäjää alkuun ja ohjata hänet yksityiskohtaisemman tiedon lähteille.

Tämä Sekvenssianalyysioppaan ensimmäinen painos perustuu vanhempaan Bio-oppaaseen, joka poistuu käytöstä. Oppaassa tapahtuneet muutokset ovat lähinnä päivityksiä, sillä CSC:n laskentaympäristö on muuttunut niiden kolmen vuoden aikana, jotka edellisen Bio-oppaan julkaisemisesta on kulunut.

CSC:n palvelimien ja niissä olevien ohjelmien käyttöä kuvaavassa tekstissä `teletype`-kirjasintyyppi tarkoittaa, että kyseessä on komennon tai tiedoston nimi. Yleiset nimet, esimerkiksi tiedostojen ja hakemistojen nimet, jotka vaihtelevat tapauskohtaisesti, on kirjoitettu *kursiivilla*:

```
rm tiedosto
```

Joissakin esimerkeissä on haluttu selvästi erottaa, mikä osa on käyttäjän antamia komentoja ja mikä on tietokoneen tulostusta. Tällöin on käyttäjän antamat komennot **`lihavoitu`**. Uusien termien esittelyyn ja tekstin korostamiseen on käytetty *kursiivia*. Erikoisnäppäimet on esitetty laatikoituina, esimerkiksi: Ctrl ja Return. Valikkoja käyttävien graafisten käyttöliittymien komentoihin on tekstissä viitattu tyyliin *Valikko/Komento*. Esimerkiksi merkintä `File/Open` tarkoittaa File-valikosta löytyvää Open-komentoa.

Opas esittelee CSC:n biopalveluja siinä muodossa, jossa ne olivat vuoden 2006 joulukuussa. Palveluissa ja käyttöliittymissä on saattanut tämän jälkeen tapahtua muutoksia.

Kiitämme oppaan tekoa edesauttaneita ja teosta eri vaiheissa kommentoineita: Minna Laine, Vesa Ollikainen, Timo Kervinen ja Juha Haataja.

Otamme mielellämme vastaan kaikki tähän kirjaan liittyvät kommentit. Ne voi osoittaa Kimmo Mattilalle sähköpostiosoitteeseen `Kimmo.Mattila@csc.fi`.

Espoossa 1.12.2006

Tekijät

Sisältö

Lukijalle **5**

1 Johdanto **10**

2 Tutkijan käyttöliittymä **11**
2.0.1 Tiedostojen hallinta ja siirto 12
2.0.2 Ohjelmat 14
2.0.3 Salasanan vaihtaminen ja tietojen päivittäminen . . . 15

3 Unix-käyttöjärjestelmä **16**
3.1 Pääteyhteydet 16
3.1.1 Pääteyhteyden luominen 16
3.1.2 X-pääteyhteyden luominen 17
3.1.3 Yhteyksien sulkeminen 18
3.1.4 Lisätietoja yhteyksistä 18
3.2 CSC:n tiedostorakenne 18
3.2.1 Kotihakemisto 19
3.2.2 Metatyöhakemisto ja muut väliaikaishakemistot . . . 20
3.2.3 Arkistohakemisto 20
3.2.4 Projektihakemisto 20
3.2.5 Hakemistorakenteen käyttö 20
3.3 Unix-komennot 21
3.3.1 Hakemistojen käyttö 21
3.3.2 Tiedostojen käsittely 23
3.3.3 Opassivut 25
3.4 Unix-käytön niksejä 25
3.4.1 Aiempien komentojen selaus 25
3.4.2 Komentojen täydennys 26
3.4.3 Ohjaus- ja erikoismerkit 26
3.4.4 Tausta-ajot ja ohjelmien keskeyttäminen 27
3.5 Teksti- ja kuvaeditorit 28
3.5.1 Nano 28
3.5.2 GNU emacs 29
3.5.3 Kuva- ja PostScript-tiedostojen tarkastelu 30
3.6 Eräajot 31

3.7 Unix-esimerkki: Arkistohakemiston käyttö 33
3.8 Lisätietoa CSC:n Unix-ympäristöstä 35

4 Tietokantahakuohjelma SRS 36
4.0.1 Tietokantojen valinta 37
4.0.2 Hakuehtojen määrittely 37
4.0.3 Hakutulokset ja tallennus 39
4.0.4 SRS:n komentorivikäyttö 42

5 Sekvenssianalyysiohjelmisto EMBOSS 44
5.1 Sekvenssitiedostot 45
5.2 USA-merkintätapa 45
5.3 EMBOSS:in käyttöliittymät 47
5.4 Www-käyttö Tutkijan käyttöliittymässä 47
5.4.1 Ohjelman suoritus ja tulosten tallentaminen 48
5.5 EMBOSS-ohjelmiston komentorivikäyttö 51
5.5.1 Alustus, tietokannat ja ohjelman valinta 51
5.5.2 Ohjelman suoritus 52
5.5.3 Tulostiedostot 53
5.5.4 Kuvatiedostot 54
5.5.5 Syötteet komentorivillä 54
5.5.6 Ohjelmakohtaiset valitsimet 54
5.5.7 Yleiset valitsimet 58
5.5.8 Listatiedostot 59
5.5.9 EMBOSS-ohjelman parametritiedostot 61
5.6 Jemboss 61
5.6.1 Tiedostojen hallinta 61
5.6.2 Ohjelman suoritus 63
5.7 Lisätietoa EMBOSS ohjelmistosta 63

6 Kahden sekvenssin vertailu ja rinnastus 65
6.1 Johdanto 65
6.2 Pistekartat (dotplot) 65
6.3 Sekvenssirinnastukset 68
6.3.1 Johdanto 68
6.3.2 Pisteytysmatriisit 68
6.3.3 Aukkosakot 71
6.4 Kahden sekvenssin rinnastus EMBOSS-ohjelmilla 72
6.4.1 Paikallinen rinnastus water- ja matcher-ohjelmilla . . 72
6.4.2 Kattava rinnastus needle- ja stretcher-ohjelmilla . . . 76
6.4.3 Rinnastusten jatkokäsittely 78
6.5 Nukleotidi-proteiini-rinnastukset Wise2-ohjelmalla . . . 79

7 Samankaltaisuushaut 82
7.1 Johdanto 82
7.2 BLAST 83

7.2.1 BLAST-algoritmi 83
7.2.2 BLAST-hakujen eri tyypit 84
7.2.3 BLAST-haku Tutkijan käyttöliittymässä 85
7.2.4 BLAST-haun parametrit 87
7.2.5 BLAST-ajon tulokset 88
7.2.6 Tulosten tarkastelu BioView Viewer -ohjelmalla . . 89
7.2.7 BLAST:in komentorivikäyttö 91
7.2.8 Parallel BLAST 93
7.2.9 Omien BLAST-tietokantojen käyttö 94
7.2.10 Lisätietoa BLASTista 95

8 Usean sekvenssin analyysi 96
8.1 Usean sekvenssin rinnastus Clustal-ohjelmilla 98
8.1.1 ClustalX Coronalla 98
8.1.2 ClustalW Tutkijan käyttöliittymässä 100
8.2 Usean sekvenssin rinnastus T-Coffeeta käyttäen 100
8.2.1 T-Coffee Coronalla 101
8.3 Rinnastusten editointi- ja visualisointiohjelmia 101
8.3.1 Rinnastus- ja sekvenssieditori Seaview 101
8.3.2 Jalview rinnastuseditori 102
8.4 Kätketyt Markovin mallit 104
8.4.1 HMMER-ajon vaiheet 104
HMM-profiilin luominen 104
Kalibrointi 105
Haku 106
8.4.2 Haut Pfam-tietokannoista 106
8.4.3 Omien HMM-profiilien käyttö Coronassa 108
8.4.4 Lisätietoa HMMERista 110
8.5 Muita monen sekvenssin analyysiohjelmia 110

9 Molekyylisystematiikka 112
9.1 Molekyylisystematiikan taustaa 112
9.2 Menetelmät 112
9.2.1 Etäisyysmenetelmät 113
9.2.2 Parsimonia 113
9.2.3 Suurimman uskottavuuden menetelmät 113
9.2.4 Ulkoryhmä ja juuri 113
9.2.5 Evoluutiomalli 114
9.2.6 Heuristinen haku 114
9.2.7 Tulosten arviointi 114
9.3 Ohjelmat 115
9.3.1 PHYLIP Coronalla 115
9.3.2 PHYLIP Tutkijan käyttöliittymässä 121
9.3.3 Muut ohjelmat 122

Liite 124

A Tärkeimmät CSC:llä ylläpidetyt tietokannat 125
A.1 Sekvenssitietokannat 125
A.1.1 EMBL 125
A.1.2 RefSeq 126
A.1.3 UniProt 126
A.2 Tunnistetietokannat 127
A.2.1 PROSITE 127
A.2.2 PRINTS 127
A.2.3 Pfam 128
A.2.4 InterPro 128
A.3 PDB-rakennetietokanta 129

B Yleisimmistä sekvenssitiedostomuodoista 130
B.1 FASTA 130
B.2 RAW/PLAIN 130
B.3 SWISS-PROT, EMBL ja NCBI 131
B.4 Clustal ohjelmien aln-muoto 133
B.5 MSF eli Multiple Sequence File format 134
B.6 Phylip-tiedostomuodot 135

Hakemisto 136

1 Johdanto

Tieteen tietotekniikan keskus CSC on opetusministeriön omistama voittoa tavoittelematon yhtiö, joka tarjoaa tietotekniikkapalveluita suomalaiselle tutkijayhteisölle.

Sekvenssianalyysiohjelmistojen käyttö CSC:llä on pääsääntöisesti ilmaista yliopisto- ja korkeakoulututkijoille. SRS-tietokantapalvelinta lukuun ottamatta tässä oppaassa käsiteltävien ohjelmien käyttöön tarvitaan kuitenkin CSC:n käyttäjätunnus. Ohjeet CSC:n käyttäjätunnuksen hakemiseksi löytyvät osoitteesta

```
http://www.csc.fi/asiakkaaksi/uusi_asiakas
```

CSC:n biopalveluiden käyttö perustuu Internet-yhteyteen käyttäjän oman tietokoneen ja CSC:n tietokoneiden välillä. CSC:llä olevien ohjelmien käyttöön on kaksi vaihtoehtoa: Verkkoselaimella toimiva Tutkijan käyttöliittymä tai pääteyhteyden kautta tapahtuva komentorivikäyttö. Useimpien ohjelmien kohdalla tässä oppaassa esitellään molemmat käyttötavat.

Tutkijan käyttöliittymä on www-ympäristö, jonka avulla voidaan helposti käyttää yleisimpiä sekvenssianalyysimenetelmiä. Tutkijan käyttöliittymän kautta tehtävät analyysit eivät vaadi Unix-käyttöjärjestelmän tuntemista. Jos aiot käyttää vain Tutkijan käyttöliittymää voit jättää Unix-johdannon sisältävän kappaleen huomiotta.

Pääteyhteyttä käytettäessä yhteys CSC:n palvelimiin otetaan pääteohjelman (esim. ssh tai putty) avulla. Pääteyhteyttä käytettäessä työskentely tapahtuu käyttäjän kirjoittamien Unix-komentojen avulla. Vaikka Tutkijan käyttöliittymässä toimivien sovellusten määrä kasvaa jatkuvasti, monet CSC:n ohjelmista ovat edelleen käytettävissä vain pääteyhteyden kautta. Samoin käsiteltäessä suuria aineistoja tai raskasta laskentaa vaativia analyysejä on pääteyhteys usein Tutkijan käyttöliittymää tehokkaampi käyttötapa.

CSC:n bioinformatiikkapalveluita kehitetään jatkuvasti. Kaikkein ajantasaisin tieto sekvenssianalytiikan ohjelmistoista ja tietokannoista CSC:llä löytyy CSC:n biotieteiden sivuilta osoitteesta:

```
http://www.csc.fi/tutkimus/alat/biotieteet/
```

Ohjelmakohtaisilla sivuilla on linkit ohjelmien varsinaisiin käyttöoppaisiin joissa ohjelmat ja menetelmät on kuvattu yksityiskohtaisemmin kuin tässä oppaassa. Kaikissa CSC:n biopalveluita käsittelevissä kysymyksissä voit myös ottaa yhteyttä suoraan CSC:n asiantuntijoihin.

2 Tutkijan käyttöliittymä

Tutkijan käyttöliittymä (Scientist's interface) on www-palvelu, jonka kautta asiakkaat voivat käyttää CSC:n palvelimilla olevia käyttää ohjelmia, käsitellä ja muokata CSC:llä olevia tiedostojaan ja siirtää tiedostoja oman koneen ja CSC:n välillä. Käyttöä varten tarvitaan verkkoselain[1] sekä CSC:n käyttäjätunnus ja salasana.

Tutkijan käyttöliittymään pääsee kaikilta CSC:n verkkosivuilta mustassa yläpalkissa olevan linkin *Extranet* avulla. Suora osoite tutkijan käyttöliittymän aloitussivulle on

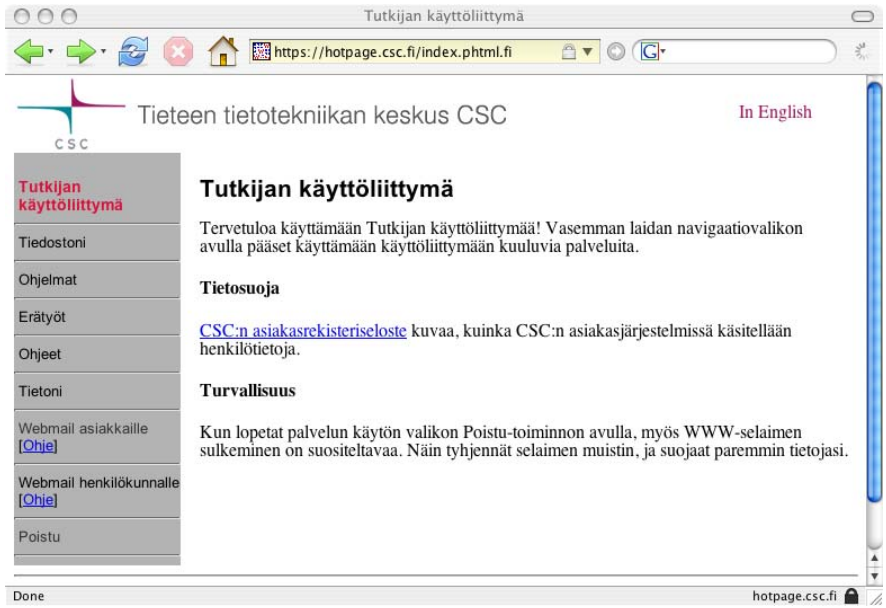
```
https://hotpage.csc.fi/log/in.phtml
```

Käyttöliittymään kirjaudutaan sisään käyttämällä CSC:n käyttäjätunnusta ja salasanaa. Mikäli paikallinen käyttöympäristösi kuuluu haka-luottamusverkostoon, riittää, että käytät CSC:n salasanaa kirjautuessasi ensimmäistä kertaa Tutkijan käyttöliittymään. Tämän jälkeen voit jatkossa käyttää kirjautumisessa haka-järjestelmää ja paikallista käyttäjätunnusta ja salasanaa. Haka-verkosto kattaa mm. useimmat suomalaiset yliopistot.

Tutkijan käyttöliittymän eri toiminnot löytyvät harmaasta sivupalkista. Mustan yläpakin oikeassa laidassa olevasta linkistä voit valita käyttöliittymän kieleksi suomen tai englannin. Jatkossa tarkastellaan tarkemmin tiedostojen käsittelyä (Vasemman sivupalkin linkki: Tiedostoni) ja ohjelmien käyttöä (Ohjelmat). Näiden toimintojen lisäksi Tutkijan käyttöliittymässä voit seurata erätöiden etenemistä (Erätyöt), tarkastella ja päivittää projektisi tietoja CSC:llä (Tietoni), tai käyttää sähköpostia (Webmail-asiakkaille).

Kun lopetat työskentelyn, sulje Tutkijan käyttöliittymä linkin *Poistu/Logout* avulla. Tämä toimenpide poistaa koneestasi kirjautumisen yhteydessä koneellesi lähetetyn tunnistusevästeen. Pelkkä verkkoselaimen sulkeminen ei siis riitä, sillä poistamaton eväste säilyy koneellasi vuorokauden ajan mahdollistaen käytön ilman salasanakyselyä.

[1] Käyttöliittymä vaatii, että selaimesi sallii käyttöliittymään kirjauduttaessa lähetettävän evästeen (cookie) vastaanoton. Jotkin selaintyypit esim. Safari, eivät pysty käyttämään Tutkijan käyttöliittymää.



Kuva 2.1: *Tutkijan käyttöliittymä.*

2.0.1 Tiedostojen hallinta ja siirto

Linkki *Tiedostoni* avaa Tutkijan käyttöliittymän tiedostonhallintaosan, jonka avulla voit tarkastella ja muokata CSC:llä olevia tiedostojasi. Voit myös kopioida, siirtää ja poistaa tiedostoja sekä siirtää tiedostoja oman koneesi ja CSC:llä olevan kotihakemistosi välillä.

Tiedostoni-toiminto avaa näkyviin listan CSC:n kotihakemistossa olevista tiedostoista ja kansioista. Listassa näkyy kunkin tiedoston nimi, omistaja, koko, käyttöoikeudet sekä tiedoston viimeisimmän muutoksen päivämäärä. Tiedostolistaa voidaan käyttää kahdella tavalla. Ensimmäinen tapa on valita ensin tiedosto klikkaamalla tiedoston nimen edessä olevaa ruutua ja tämän jälkeen valitsemalla haluttu toiminto tiedostolistauksen päällä olevilla painikkeilla (Imuroi PC:llesi, Muokkaa, Poista, Siirrä tai Kopioi).

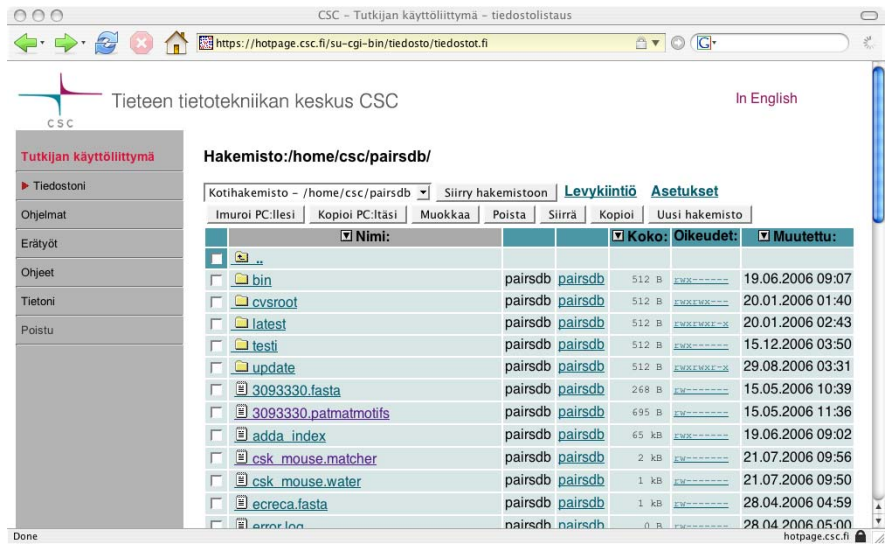
Toinen tapa on on klikata suoraan tiedoston nimeä. Tällöin tutkijan käyttöliittymä suorittaa tiedostolle toiminnon, joka on valittuna Tutkijan käyttöliittymän *Asetukset* kohdassa. Vaihtoehtoina ovat Muokkaa, Kopioi omalle koneelle tai näytä.

Muokkaa-toiminto aukaisee tiedoston selaimessa näkyvässä editorissa. Tiedoston sisältöä, nimeä ja lukuoikeuksia voidaan muuttaa. Muutokset jäävät voimaan painamalla sivun alareunassa olevaa *Tallenna muutokset* painiketta.

Imuroi PC:llesi eli **Kopioi omalle koneelle** tallentaa tiedoston oman koneesi kovalevylle. Varsinainen kopiointiprosessi riippuu käytettävästä selaimesta ja käyttöjärjestelmästä. Useimmissa tapauksissa selain kysyy, minne ja minkä

nimiseksi tiedosto kopioidaan.

Näytä-toiminto lähettää tiedoston sisällön sellaisenaan selainohjelmalle. Tätä toimintoa kannattaa käyttää tutkijan käyttöliittymässä olevien html-tiedostojen, kuvien sekä esimerkiksi pdf ja .doc muotoisten tiedostojen tarkasteluun. Unix-ohjelmien tekemiä tekstimuotoisia tiedostoja ei yleensä kannata tarkastella näytä-toiminnon vaan muokkaa-toiminnon avulla.

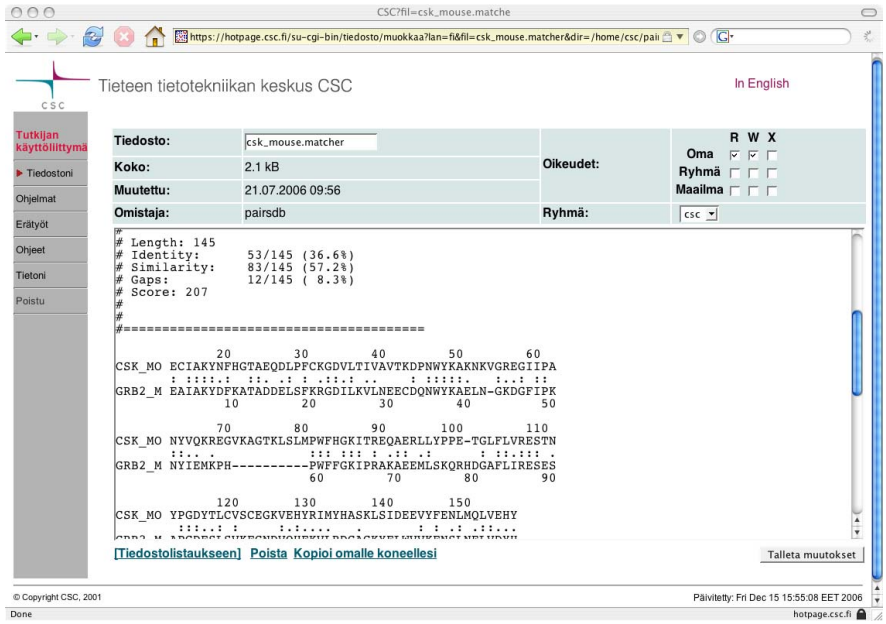


Kuva 2.2: *Tiedostonhallinta Tutkijan käyttöliittymässä.*

Tiedoston siirto omalta koneelta CSC:lle onnistuu linkin **Kopioi PC:ltäsi** avulla. Tämä avaa uuden ikkunan, jossa voit nappia *Browse* painamalla etsiä siirrettävän tiedoston omalta koneeltasi. Jos siirrät tekstitiedostoja Windows-koneelta, ruksaa kohta *Muunna unix-muotoon* ennen napin *Kopioi tiedosto* painamista.

Kotihakemistossa oleviin alihakemistoihin voi siirtyä osoittamalla hakemistojen nimiä. Kotihakemiston koko on 200 MB (käyttämäsi levytila näkyy linkistä *levykiintiö*). Lisäksi käytettävissäsi on suuria väliaikaistiedostoja varten `verkkolevy` (`$METAWRK`) jossa ei ole kokorajoitusta. Verkkolevy ei kuitenkaan sovellu tiedostojen pitkäaikaiseen säilytykseen, sillä kuukauden käyttämättömänä olleet tiedostot poistetaan sieltä automaattisesti (katso kappale:3.2.2). Siirtyminen kotihakemiston ja verkkolevyn välillä tapahtuu *Siirry hakemistoon*-napista, kun oikea hakemisto on pudotusvalikosta valittu.

Tiedostoja ja hakemistoja voi siirtää, kopioida ja poistaa ruksaamalla ne ja valitsemalla hakemistolistauksen päällä olevista painikkeista haluttu toiminto (*Siirrä, Poista, Kopio, Uusi hakemisto*). Kopioitaessa ja siirrettäessä tiedostoja avautuu uusi sivu, jolla kohdehakemiston voi valita pudotusvalikosta tai hakemistojen nimeä osoittamalla.



Kuva 2.3: *CSC:llä sijaitseva tiedosto avattuna Tutkijan käyttöliittymässä.*

2.0.2 Ohjelmat

Tutkijan käyttöliittymässä oleviin ohjelmiin pääsee harmaassa sivupalkissa olevan *Ohjelmat*-linkin avulla. Bioinformatiikkasovellusten pääsivulle siirrytään valitsemalla *Selaimella käytettävissä olevat sovellukset* -listasta kohta: *Biotiede*. Suora osoite biosovellusten pääsivulle on:

```
https://hotpage.csc.fi/appl/molbio/index.phtml.fi
```

Biotieteiden ohjelmasivu on saatavilla vain englanniksi. Käytettävissä olevat ohjelmat on luokiteltu käyttötarkoituksen mukaan. Kuhunkin analyysityyppiin liittyvät ohjelmat saat näkyviin klikkaamalla kyseisen analyysiryhmän nimeä. Esimerkiksi *Restriction sites* linkin avulla löytyvät restriktioanalyyseihin liittyvät ohjelmat. Ohjelman käyttöliittymä aukeaa klikkaamalla ohjelman nimeä (esim. remap). Suurin osa käytettävistä ohjelmista kuuluu EMBOSS-sekvenssianalyysiohjelmistoon. EMBOSS-ohjelmien lisäksi käytettävissä ovat sekvenssirinnastusohjelma ClustalW, sekvenssihakuohjelma BLAST, fylogeniaohjelmisto PHYLIP sekä geenikartoitusohjelma Merlin. EMBOSS-, BLAST-, ClustalW ja Phylip-ohjelmien käyttöä käsitellään tämän oppaan muissa kappaleissa. Merlin ohjelmasta kerrotaan tarkemmin CSC:n Merlin-ohjesivulla.

```
http://www.csc.fi/molbio/progs/linkage/merlin/
```

Tutkijan käyttöliittymää käytetään myös muutamien paikallisesti asennettavien ohjelmistojen jakeluun. Nämä ohjelmat löytyvät Ohjelmat-pääsivulta

kohdasta *Omalle koneelle ladattavat sovellukset*. Paikallisesti asennettavia biotieteiden ohjelmia ovat BLAST-tulosten tarkasteluohjelma BioView Viewer (katso kappale 7.2.6) ja molekyylimallinnusohjelmisto Discovery Studio.

2.0.3 Salasanan vaihtaminen ja tietojen päivittäminen

Harmaan sivupalkin linkistä *Tietoni* avautuvalla sivulla voit tarkistaa ja päivittää yhteys- ja projektitietosi sekä tarvittaessa vaihtaa salasanan.

3 Unix-käyttöjärjestelmä

CSC:n sovelluspalvelin, corona.csc.fi, käyttää Unix- käyttöjärjestelmää (Solaris 9). Toisin kuin graafiset Windows- ja Mac-käyttöjärjestelmät, Unix- ja Linux-käyttöjärjestelmissä työskentely perustuu edelleen paljolti komentorivikäyttöön, jossa käyttäjä kirjoittaa komentojen ja tiedostojen nimet. Suurissa palvelinkoneissa Unix-käyttöjärjestelmiä pidetään graafisia käyttöjärjestelmiä vakaampina ja joustavampina. Tämän takia useat monen käyttäjän koneet käyttävät jotakin Unix- tai sitä suuresti muistuttavan Linux-käyttöjärjestelmän versiota.

Tämän kappaleen tarkoituksena on auttaa uutta käyttäjää ottamaan ensiaskeleensa CSC:n Unix-ympäristössä. Esiteltävät Unix-käskyt ja -ohjelmat edustavat vain murto-osaa Unixin toiminnoista. Kattavampi ohjeistus työskentelyyn CSC:n palvelimien Unix-ympäristössä löytyy kirjoista *Corona-käyttöopas* ja *CSC:n Palvelinympäristö*. Oppaat löytyät CSC:n Oppaat-sivulta:

`http://www.csc.fi/csc/julkaisut/oppaat`

3.1 Pääteyhteydet

3.1.1 Pääteyhteyden luominen

Yhteydenotto CSC:n palvelimille tehdään salausta käyttävällä pääteohjelmalla. Linux- ja Mac OSX -koneissa on yleensä valmiiksi asennettuna tähän tarkoitukseen sopiva ohjelma `ssh`. Tällöin yhteydenotto esimerkiksi Corona-palvelimelle tehdään yksinkertaisesti avaamalla omassa koneessa uusi pääteohjelma (terminal) ja antamalla komento:

`ssh corona.csc.fi -l` *käyttäjätunnus*

Jos haluat käyttää myös X-pääteominaisuuksia anna komento

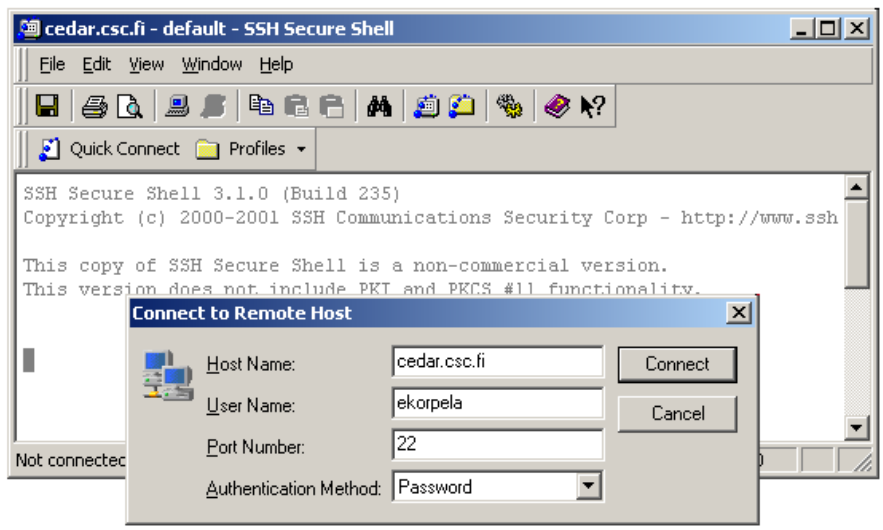
`ssh corona.csc.fi -X -l` *käyttäjätunnus*

Joissakin tapauksissa -X valitsin ei toimi, jolloin kannattaa kokeilla valitsinta -Y (huom. -X ja -Y kirjoitetaan isoilla kirjaimilla).

`ssh corona.csc.fi -Y -l` *käyttäjätunnus*

Windows-koneissa voidaan pääteyhteyden luomiseen käyttää esimerkiksi kau-

pallista F-Secure SSH tai ilmaista PuTTY-ohjelmaa. F-Secure SSH -ohjelmassa yhteydenluonti käynnistetään valitsemalla *File/Connect*. Ohjelma kysyy palvelinkoneen nimen (*Host Name*) ja asiakkaan käyttäjätunnuksen (*User Name*). Painettaessa nappia *Connect* ohjelma kysyy vielä käyttäjän salasanan.



Kuva 3.1: *Yhteydenotto SSH:lla.*

Kohdasta *Edit/Settings* voit muuttaa pääteyhteyden asetuksia. Muuttaminen ei ole välttämätöntä, mutta se voi helpottaa päätetyöskentelyä. Muutokset on tehtävä ennen yhteyden avausta, jotta ne ovat voimassa istunnon aikana.

Settings-ikkunan kohdasta *Profile Settings/Tunneling* kannattaa ruksata *Tunnel X11 connection*, jolloin X- ikkunayhteys ohjautuu näytöllesi automaattisesti. Ssh:n kautta ohjattua X-yhteyttä voi nopeuttaa kohdassa *Profile Settings/Connection* poistamalla asetus *Compression* ja muuttamalla salausmenetelmän (*Encryption Algorithm*) Blowfish-tyyppiseksi.

Käyttäjällä voi olla auki useita pääteyhteyksiä yhteen tai useampaan palvelimeen. Uuden pääteikkunan avaaminen onnistuu SSH- ohjelman kohdasta *Window/NewWindow*.

3.1.2 X-pääteyhteyden luominen

X-pääteyhteys ei ole välttämätön CSC:n palvelimien käyttämiseen, mutta graafisen käyttöliittymän sisältävät ohjelmat (esim. ClustalX) vaativat sen toimiakseen. Jotta X-pääteyhteys olisi mahdollinen, täytyy päätekoneessasi olla X-ikkunointiohjelma eli X-emulaattori. Linux-koneissa X-ikkunointiohjelma on yleensä automaattisesti käytössä. Mac OSX koneissa käytettävä ilmainen X11-pääteohjelma on ladattavissa Applen omilla sivuilla.

Windows koneisiin on saatavissa useita kaupallisia X-pääteohjelmia (mm. XVision ja Hummingbird eXceed). Näiden ohjelmien käynnistys ja oikeat asetukset vaihtelevat kone- ja ohjelmakohtaisesti. X-pääteohjelmien toiminnasta ja asetuksista kannattaa kysyä paikallisilta mikrotukihenkilöiltä.

Yksinkertaisin tapa käyttää X-pääteyhteyttä on ohjata se SSH-ohjelman kautta edellä olleiden esimerkkien mukaisesti (valitsimet -X ja -Y tai Tunnel X11 asetus).Huomaa että Windows- ja Mac OSX- koneissa X-pääteohjelma on käynnistettävä ennen ssh-yhteyden avaamista. Kun oman koneesi X-pääteohjelma on käynnissä ja X-pääteyhteyden salliva yhteys Corona-palvelimelle on aukaistu, voit testata yhteyden toiminnan esimerkiksi avaamalla uuden X-ikkunan komennolla:

```
xterm &
```

tai käynnistämällä taskulaskimen

```
xcalc &
```

X-ikkuna olettaa, että käyttäjällä on kolminäppäiminen Unix-hiiri. Tämä saattaa tuottaa hankaluuksia joidenkin ohjelmien käytössä, sillä joissakin PC- ja Mac-koneiden hiirissä on vain yksi tai kaksi näppäintä. Useimmat X-emulaattorit sisältävät kuitenkin toiminnon, jossa painamalla samanaikaisesti jotain näppäinyhdistelmää sekä hiiren näppäintä saadaan puuttuvien hiirinäppäinten toiminnot suoritettua. Käytettävät näppäinyhdistelmät vaihtelevat ohjelmakohtaisesti.

3.1.3 Yhteyksien sulkeminen

Kun työskentely CSC:n palvelimella lopetetaan, suljetaan ensin CSC:n palvelimissa käynnissä olevat ohjelmat. Tämän jälkeen sekä pääte- että X-pääteyhteydet katkaistaan komennoilla `exit` tai `logoff`. Pääteohjelma ja X-emulaattori suljetaan päätekoneessa vasta yhteyden sulkemisen jälkeen.

3.1.4 Lisätietoja yhteyksistä

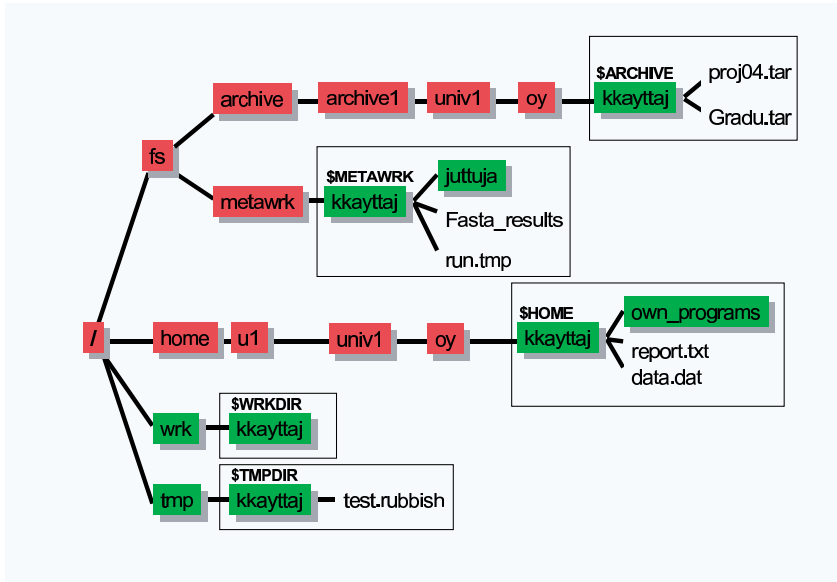
Yksityiskohtaisempaa tietoa pääte- ja X-pääteyhteyksistä löytyy CSC:n palvelinympäristö -oppaan luvuista 6.8. Oppaan voi tilata CSC:stä tai hakea PDF-muodossa CSC:n kotisivuilta:

```
http://www.csc.fi/oppaat/palvelimet/
```

3.2 CSC:n tiedostorakenne

Tiedostojen hallinta Unixissa perustuu hierarkkiseen puumaiseen hakemistorakenteeseen. Jokaisessa koneessa on juurihakemisto (/), jonka alaisuudessa

muut hakemistot ja tiedostot ovat. Jokainen tiedosto sijaitsee jossakin hakemistossa. Samoin käskyt suoritetaan jossakin hakemistossa. Hakemistoa, jossa käyttäjä on antaessaan komennon, kutsutaan tässä oppaassa nykyhakemistoksi.



Kuva 3.2: *Kaavakuva käyttäjän kkayttaj hakemistojen sijainnista Coronassa.*

Coronan käyttäjällä on useita henkilökohtaisia hakemistoja, joissa omia tiedostoja voidaan säilyttää, suorittaa komentoja ja ajaa ohjelmia. Tärkeimmät näistä ovat

- kotihakemisto(`$HOME`)
- metatyöhakemisto (`$METAWRK`)
- arkistohakemisto (`$ARCHIVE`)
- projektihakemisto

3.2.1 Kotihakemisto

Kun käyttäjä avaa pääteyhteyden Coronaan, hän siirtyy automaattisesti omaan kotihakemistoonsa. Kotihakemisto on käyttäjäkohtainen perushakemisto, jonne voidaan pysyvästi varastoida aineistoa ja luoda alihakemistoja. Kotihakemiston koko on 200 MB, mikä riittää useimmille CSC:n biopalvelujen käyttäjille, mutta saattaa rajoittaa hyvin suurten tiedostojen käsittelyä. Kotihakemisto näkyy samanlaisena kaikkiin CSC:n palvelimiin ja tutkijan

käyttöliittymään. Kotihakemistossa olevia tiedostoja voidaan siis muokata ja käyttää sekä pääteyhteyden että Tutkijan käyttöliittymän kautta. Kotihakemisto varmuuskopioidaan päivittäin.

3.2.2 Metatyöhakemisto ja muut väliaikaishakemistot

Metatyöhakemisto (`$METAWRK`) on tarkoitettu suurille väliaikaisille tiedostoille. Metatyöhakemistossa ei ole kokorajoitusta, mutta sinne ei voi varastoida tiedostoja pysyvästi. Tiedostot, joita ei ole muokattu yli kuukauteen, poistetaan automaattisesti. Metatyöhakemistoa voidaan käyttää myös Tutkijan käyttöliittymän kautta. Tutkijan käyttöliittymässä metatyöhakemistosta käytetään nimeä *verkkolevy*.

Metatyöhakemiston lisäksi jokaisella käyttäjällä on myös kaksi muuta väliaikaishakemistoa, `$WRKDIR` ja `$TMPDIR`. Näissä hakemistoissa tiedostojen säilytysajat ovat kuitenkin lyhemmät: `$WRKDIR`:ssä yksi viikko ja `$TMPDIR`:ssä noin yksi päivä.

Väliaikaishakemistoja ei varmuuskopioida, joten vahingossa poistettuja tai levyrikossa tuhoutuneita tiedostoja ei voida palauttaa.

3.2.3 Arkistohakemisto

Arkistohakemisto `$ARCHIVE` on tarkoitettu harvoin käytettävien tiedostojen pidempiaikaiseen säilytykseen. Toisin kuin työ- ja kotihakemistoissa, arkistohakemistossa ei voi muokata tiedostoja tai ajaa ohjelmia. Arkistohakemistoon siirrettyjä tiedostoja ei varastoida levylle, vaan ne kopioituvat CSC arkistopalvelimen nauhakaseteille. Arkistohakemistoon varastoitavan aineiston määrällä ja säilytysajalla ei ole rajoituksia, kun CSC- käyttäjätunnus on voimassa. Kappaleessa 3.7 on esimerkki tiedon tallentamisesta arkistohakemistoon

3.2.4 Projektihakemisto

CSC:n asiakkaat voivat anoa käyttöönsä ns. projektihakemiston, joka on tutkimusryhmän yhteiskäytössä olevaa pysyvää varmuuskopioitua levytilaa. Projektihakemiston koko, maksimissaan jopa 1000 GB, määräytyy käyttötarpeen mukaan.

3.2.5 Hakemistorakenteen käyttö

Coronan hakemistorakenteen tarkoitus on välttää levytilan turhaa käyttöä. Monet ohjelmat tuottavat ajon aikana suuria väliaikaistiedostoja, joita ei enää tarvita tulosten valmistuttua. Suorittamalla ajonsa metatyöhakemistossa ja

siirtämällä vain varsinaiset syöttötiedot ja lopputulokset kotihakemistoon käyttäjä välttyy varastoimasta tarpeettomia tiedostoja. Myöhemmin, esimerkiksi tutkimusprojektin päätyttyä, projektiin liittyvät tiedostot voidaan pakata yhdeksi tiedostoksi ja varastoida pysyvästi arkistohakemistoon.

Kaikki käyttäjät pystyvät liikkumaan laajalti Coronan hakemistoissa ja lukemaan mm. ohjelma- ja tietokantahakemistoissa olevia tiedostoja. Ohjelmien suorittaminen ja uusien tiedostojen luominen onnistuu kuitenkin vain käyttäjän henkilökohtaisissa hakemistoissa (esitetty kuvassa 3.2 vaaleanharmaana). Muut käyttäjät eivät pysty lukemaan henkilökohtaisiin hakemistoihin varastoitua aineistoa.

3.3 Unix-komennot

Kun pääteyhteys CSC:lle (esim. Corona-palvelimeen) on aukaistu, voidaan palvelinta käyttää Unix-komentojen avulla. Unix-käyttöjärjestelmässä useimpien komentojen perusrakenne on seuraava:

```
komento -valitsimet argumentti1 argumentti2 ...
```

Valitsimet ja argumentit vaihtelevat komentokohtaisesti. Monissa tapauksissa valitsimien ja argumenttien antaminen ei ole välttämätöntä. Valitsimien tehtävä on yleensä muuttaa komennon toimintaa oletusarvosta, joka suoritetaan, jos valitsimia ei määritetä. Argumenteilla taas määritetään, mitä tiedostoja, merkkijonoja tai hakemistoja käsitellään tai mihin komennon tulokset ohjataan.

Esimerkiksi komennolle `ls` voidaan antaa useita valitsimia ja argumentteja. Pelkkä komento `ls` tulostaa hakemiston sisällön aakkosjärjestyksessä. Komennon toimintaa voidaan muuttaa esimerkiksi valitsimella `-t`, joka tulostaa hakemiston sisällön tiedostojen iän mukaisessa järjestyksessä: `ls -t`. Ilman argumenttia `ls` tulostaa näytölle sen hakemiston sisällön, jossa käyttäjä on antaessaan komennon. Komennolle voidaan kuitenkin antaa argumentti, joka kertoo, minkä hakemiston sisältöä halutaan tarkastella. Esimerkiksi komento:

```
ls -t /fs/metawrk/kkayttaj
```

tulostaa hakemiston `/fs/metawrk/kkayttaj` sisällön. Jos komento, valitsimet tai argumentit on annettu väärin, tulostuu ruudulle Return-näppäimen painamisen jälkeen virheilmoitus. Komentojen toimintaa ja käyttöä esitellään tarkemmin seuraavissa kappaleissa.

3.3.1 Hakemistojen käyttö

Taulukossa 3.1 on esitetty tärkeimmät komennot, joilla voit liikkua Unixin hakemistojärjestelmässä ja muokata sitä. Alla on esimerkkejä komentojen käytöstä.

Taulukko 3.1: *Unix-komentoja.*

Komento	*Selitys*
`cd hakemisto`	Siirtyminen toiseen hakemistoon (change directory)
`ls`	Hakemiston sisällön listaus (list)
`pwd`	Nykyhakemiston nimen tulostaminen (print working directory)
`mkdir hakemisto`	Uuden hakemiston luominen (make directory)
`rmdir hakemisto`	Hakemiston poisto (remove directory)

Kun olet ottanut pääteyhteyden Coronaan, olet kotihakemistossasi. Voit tarkistaa hakemistosi sijainnin komennolla `pwd`. Kotihakemiston sijainnin muistaminen ei kuitenkaan ole välttämätöntä (katso `cd`-komento sivu 23 ja erikoismerkit sivu 27).

Hakemiston sisällön näet komennolla `ls`. Se tulostaa nykyhakemistossa olevien tiedostojen ja alihakemistojen nimet. Enemmän tietoa hakemistosi sisällöstä saat näkyviin komennolla `ls -la`. Listauksessa näkyvät tällöin myös pisteellä (`.`) alkavat asetustiedostot sekä tiedostojen ja hakemistojen käyttöoikeudet, koot ja viimeisimmän muutoksen ajankohta. Komennot `ls` ja `pwd` eivät muuta tiedostoja tai hakemistoja, joten voit halutessasi käyttää niitä tämän oppaan myöhempien esimerkkien lomassa tarkistaaksesi, missä hakemistossa olet ja mitä tiedostoja kyseisessä hakemistossa on.

Hakemistosta toiseen siirrytään komennolla `cd` *hakemistopolku*. Esimerkiksi metatyöhakemistoon siirrytään komennolla

```
cd /fs/metawrk/käyttäjätunnus
```

tai

```
cd $METAWRK
```

Jälkimmäisessä tapauksessa käytetään hakemistopolun sijasta automaattisesti määritettyä `$METAWRK`-ympäristömuuttujaa, joka sisältää metatyöhakemiston sijainnin.

Uusia alihakemistoja voi Unix-ympäristössä luoda käskyllä `mkdir` *hakemiston_nimi*. Esimerkiksi komento

```
mkdir sekvenssit
```

luo uuden hakemiston nimeltä `sekvenssit`. Komennolla `ls` voit tarkistaa, että hakemisto syntyi. Voit siirtyä luomaasi alihakemistoon komennolla

```
cd sekvenssit
```

Voit palata `sekvenssit`-hakemistosta takaisin ylemmän tason hakemistoon komennolla (huomaa cd-komennon jälkeen tuleva välilyönti):

```
cd ..
```

Alihakemisto poistetaan komennolla `rmdir` *hakemisto*, esimerkiksi:

```
rmdir sekvenssit
```

Komentoa `rmdir` voit käyttää ainoastaan silloin, kun poistettava hakemisto on tyhjä. Riippumatta siitä, missä hakemistossa olet, takaisin kotihakemistoon pääset aina antamalla komennon `cd` ilman hakemistonnimeä:

```
cd
```

3.3.2 Tiedostojen käsittely

Unix-käyttöjärjestelmän ja -ohjelmien käyttämät tiedostot voidaan luokitella kahteen pääluokkaan, binääriaineistoa sisältäviin tiedostoihin ja merkkiaineistoa sisältäviin tekstitiedostoihin. Binääritiedostoissa tieto on binäärikoodattuna, jolloin tiedoston datatiheys on suurempi ja käsittely nopeampaa. Käyttäjän kannalta varjopuolena on se, että binääritiedostoa on hyvin hankala lukea ja muokata. Yleensä binääritiedostot sisältävätkin ohjelmien käyttämää aineistoa, jota käyttäjän ei tarvitse käsitellä.

Teksti- eli ASCII-tiedostoissa tieto on esitetty merkkeinä, joten käyttäjä voi helpommin muokata ja tarkastella tiedostoa. Tekstitiedostot ovat huomattavasti binääritiedostoja yleisempiä ja miltei kaikki tässä oppaassa käsiteltävät tiedostot ovat erityyppisiä tekstitiedostoja.

Jokaisella tiedostolla on nimi. Nimi voi olla periaatteessa miltei mikä tahansa merkkiyhdistelmä. Joitakin merkkejä ei kuitenkaan ole järkevää käyttää tiedostonimissä, koska niillä on erityismerkitys Unix-ympäristössä. Turvallisia merkkejä ovat pienet ja isot kirjaimet sekä numerot, piste, tavuviiva ja alaviiva. Esimerkiksi välilyöntien käyttö tiedostonimissä tuottaa ongelmia ja yleensä niiden sijasta käytetään alaviivaa tyyliin `uusi_tiedosto.txt`.

Huomaa että Unix:issa isot ja pienet kirjaimet tulkitaan aina eri merkeiksi. Nimet `uusi_tiedosto.txt` ja `Uusi_Tiedosto.txt` viittaavat siis eri tiedostoihin. Tiedostojen nimissä ei yleensä tarvitse käyttää Windows-tyylistä tyyppimäärittelyä (esim. .txt, .seq tai .fasta) mutta pitkällä aikavälillä johdonmukainen nimeämistapa helpottaa tiedostojen hallintaa.

Taulukossa 3.2 on esitelty yleisimmät tiedostojen käsittelyssä käytettävät Unix-komennot.

Tekstitiedostojen, esimerkiksi sekvenssitiedostojen, sisältöä voi tarkastella komennolla `cat` sivutusohjelmilla `more` ja `less`. Komentoja `cat`, `more` ja `less` käytetään kirjoittamalla komento ja tarkasteltavan tiedoston nimi. Esimerkiksi kotihakemistossa olevan tiedoston `.cshrc` sisältöä voi tarkastella komennoilla:

```
cat .cshrc
```

tai

```
more .cshrc
```

tai

```
less .cshrc
```

Taulukko 3.2: *Komentoja tiedostojen käsittelyyn.*

Komento	*Selitys*
`cat`	Tiedostojen tulostus näytölle
`cp`	Tiedostojen kopiointi (copy)
`grep`	Merkkijonon poimiminen tiedostosta
`gzip`	Tiedoston kompressointi
`gunzip`	Kompressoinnin purkaminen
`head`	Tiedoston alun tulostus
`less`	Tiedoston tulostus kuvaruutu kerrallaan
`more`	Tiedoston tulostus kuvaruutu kerrallaan
`mv`	Tiedoston uudelleennimeäminen tai siirto (move)
`rm`	Tiedoston tuhoaminen (remove)
`tail`	Tiedoston lopun tulostus
`tar`	Hakemistojen pakkaaminen tar-tiedostoiksi

Cat ja sivutusohjelmat ovat turvallisia työkaluja tiedostojen tarkasteluun, sillä niillä ei voi tuhota tai muuttaa tarkasteltavaa tiedostoa.

Cat-komento tulostaa yhden tai useamman tiedoston sisällön näytölle sellaisenaan. Sivutusohjelmat more ja less ovat usein käyttökelpoisempia ohjelmia sillä ne tulostavat tiedoston sisällön pääteikkunaan näytönpituus kerrallaan. Sekä `more` että `less` ohjelmilla voit lukea tiedostoa eteenpäin rivi kerrallaan painamalla Return tai kuvaruudullinen kerrallaan painamalla Space. Tiedoston katselun voi lopettaa painamalla `q`.

`less` sivutusohjelmassa tiedostoa voi selata myös takaisinpäin riveittäin painamalla `k` tai kuvaruudullinen kerrallaan painamalla `b`. Tekstistä voi myös etsiä merkkijonoa kirjoittamalla */merkkijono*.

Tiedoston alkua voit tarkastella komennolla `head` ja loppua komennolla `tail`. Oletusarvoisesti komennot tulostavat kymmenen ensimmäistä tai viimeistä riviä, mutta rivien määrää voi muuttaa tyyliin `tail` *-rivimäärä tiedosto*, esimerkiksi 30 viimeistä riviä tiedostosta `output.seq` tulostuvat komennolla

```
tail -30 output.seq
```

Komentoa `grep` käytetään merkkijonojen etsimiseen tekstitiedostosta. Komento annetaan muodossa `grep` *merkkijono tiedosto* ja se tulostaa näytölle ne tiedoston rivit, joista haettu merkkijono löytyi. Esimerkiksi komento

```
grep CC sekvenssi.seq
```

tulostaa tiedostosta `sekvenssi.seq` rivit, joilta löytyy kirjainyhdistelmä `CC`.

Tiedoston kopiointi tapahtuu Unixissa käskyllä `cp` (copy), jota voi käyttää vaikkapa seuraavasti:

```
cp output.seq kopio.seq
cp output1.seq output2.seq sekvenssit/
```

Ensimmäisessä tapauksessa tehdään tiedostosta `output.seq` kopio, jonka ni-

meksi tulee `kopio.seq`. Toisessa esimerkissä kopioidaan hakemistoon `sekvenssit` tiedostot `output1.seq` ja `output2.seq` nimiä muuttamatta.

Tiedoston uudelleennimeäminen tai siirto tehdään käskyllä `mv` (move):

```
mv output.seq kopio.seq
mv output1.seq output2.seq sekvenssit/
```

Esimerkeissä tapahtuvat samat siirrot kuin `cp`-käskyn esimerkeissä, mutta tällä kertaa alkuperäiset tiedostot `output.seq`, `output1.seq` ja `output2.seq` häviävät nykyhakemistosta.

Tiedosto tuhotaan käskyllä `rm` *tiedosto*, esimerkiksi

```
rm kopio.seq
```

minkä jälkeen rm-komento kysyy varmistuksen sille, että tiedosto tosiaankin halutaan poistaa:

```
rm: remove kopio.seq (yes/no)?
```

Vastaukseksi riittää `y` (yes) tai `n` (no).

Huomaa, että rm-komennon yhteydessä tehtävä varmistus ei välttämättä ole käytössä CSC:n ulkopuolisissa unix-koneissa.

3.3.3 Opassivut

CSC-palvelimiin on asennettu yksityiskohtaiset englanninkieliset kuvaukset Unix-käyttöjärjestelmän komennoista. Kunkin komennon kuvauksen saat näkyviin komennolla `man` *komento*, esimerkiksi `man ls`.

Man-komennon tulostamaa tekstiä voi selata `less`-komennon tapaan nuoli- ja välilyöntinäppäimillä. Ohjeiden lukemisen voi lopettaa valitsemalla `q`.

3.4 Unix-käytön niksejä

Komentojen syöttämistä pääteikkunassa ohjaa komentotulkki, joka määrittelee Unix-ikkunan käyttöympäristön toiminnan ja käytettävissä olevat komennot. CSC:n koneissa käyttäjien oletuskomentotulkki on tcsh. Jäsennetyn kuvauksen tcsh-komentotulkin toiminnasta löydät CSC:n palvelinympäristö - oppaasta. Alle on koottu muutamia tcsh:n ominaisuuksia, jotka helpottavat Unix-ympäristön käyttöä.

3.4.1 Aiempien komentojen selaus

Nuolinäppäimillä ↑ ja ↓ voit selata aiemmin annettuja komentoja ja suorittaa ne uudelleen painamalla Return.

3.4.2 Komentojen täydennys

Tab-näppäimellä voit täydentää komennon tai tiedostonnimen komennon syöttämisen yhteydessä. Esimerkiksi, jos olet hakemistossa, jossa on vain yksi s-alkuinen tiedosto, `sekvenssi1.seq`, voit käyttää täydennystä kirjoittamalla

```
less s
```

ja painamalla tämän jälkeen Tab, komento täydentyy automaattisesti muotoon

```
less sekvenssi1.seq
```

Jos hakemistossa on muita samanalkuisia tiedostoja, täydennys tehdään vain niin pitkälle kuin vaihtoehtoja on vain yksi. Jos edellisen esimerkin tapauksessa hakemistossa on myös tiedosto `sekvenssi2.seq`, pystyy komentotulkki täydentämään komennon vain osittain muotoon

```
less sekvenssi
```

Tämän jälkeen voit kirjoittaa komennon loppuun, mutta riittää myös, jos jatkat komennon muotoon

```
less sekvenssi1
```

ja täydennät komennon jälleen painamalla Tab.

3.4.3 Ohjaus- ja erikoismerkit

Unixissa on joukko merkkejä, joita käytetään apuna komennoissa. Näitä merkkejä ei kannata käyttää tiedostojen ja hakemistojen nimissä. Yleisimmin käytettyjä erikoismerkkejä ovat *, |, > ja ~.

Merkki * (asteriski) tarkoittaa mitä tahansa merkkiyhdistelmää. Merkkiä voi käyttää sinällään tai muiden merkkien seassa. Esimerkiksi komento `ls *` listaa kaikki nykyhakemistossa olevat tiedostot ja hakemistot, mutta `ls *.fasta` vain ne tiedostot, jotka loppuvat päätteeseen `.fasta`.

Merkki | (putki, pystyviiva) ohjaa komennon tulosteen seuraavalle komennolle. Esimerkiksi komennossa

```
head -100 sekvenssi.ent | grep CC
```

Merkki | ohjaa komennon `head -100 sekvenssi.seq` tulosteen (sata ensimmäistä riviä tiedostosta sekvenssi.ent) näytön sijasta `grep`-komennolle, joka tässä tapauksessa poimii tulosteesta näytölle vain ne rivit, joissa esiintyy merkkiyhdistelmä `CC`.

Merkki > ohjaa komennon tulosteen uuteen tiedostoon. Esimerkiksi komento

```
ls *.fasta > fasta_sekvenssit.lista
```

kirjoittaa `ls *.fasta`-komennon tuottaman listan .fasta-loppuisten tiedostojen nimistä tiedostoon `fasta_sekvenssit.lista`.

Merkki ~ (aaltoviiva, tilde) toimii tcsh-komentotulkissa käyttäjän kotihakemiston symbolina. Jos esimerkiksi haluat kopioida työhakemistosta tiedoston kotihakemistoosi, voit hakemistopolun sijasta viitata kotihakemistoon ~-merkillä seuraavasti:

```
cp tiedosto.txt ~/tiedosto.txt
```

Merkki \ (kenoviiva) merkkiä käytetään kertomaan komennolle, että kenoviivan jälkeen tulevaa merkkiä ei tulkita erikoismerkiksi. Tässä oppaassa \-merkki esiintyy usein pitkien komentorivien yhteydessä, tarkoittamassa sitä, että komento jatkuu \-merkin jälkeen tulevalla rivillä (T.s. \-merkin jälkeen annettua Return lyöntiä ei tulkita kehoitteeksi suorittaa kirjoitettu komennon osuus).

3.4.4 Tausta-ajot ja ohjelmien keskeyttäminen

Graafista käyttöliittymää käyttävät ohjelmat, sekä ohjelmat ja komennot, jotka eivät ole vuorovaikutteisia, voidaan haluttaessa ajaa Coronassa tausta-ajona. Tausta-ajossa pääteikkuna ei jää odottamaan ohjelman tai komennon suorittamisen loppumista, vaan pääteikkunassa voidaan käynnistää uusia komentoja ja ohjelmia, vaikka aiemmin annettua komentoa yhä suoritettaisiin. Komento ohjataan tausta-ajoksi lisäämällä komennon perään &-merkki.

Esimerkiksi käynnistämällä ggv-ohjelma tausta-ajoksi komennolla

```
ggv kuva.ps &
```

voidaan pääteikkunaa, josta ggv käynnistettiin, käyttää muuhun työskentelyyn ggv-ohjelman ollessa auki. Tausta-ajo saadaan muutettua tavalliseksi ajoksi komennolla `fg`.

Käynnissä oleva ohjelma tai komento voidaan haluttaessa keskeyttää. Vuorovaikutteisessa käytössä, jossa komennon jälkeen ohjelma kysyy seuraavaa tarvitsemaansa tietoa (esimerkiksi valitsinta tai argumenttia), ohjelma keskeytetään valitsemalla Ctrl-`c` (eli painetaan control ja `c` samanaikaisesti). Jos ohjelma halutaan pysäyttää vain väliaikaisesti, valitaan Ctrl-`z`. Näin pysäytetty ohjelma voidaan käynnistää uudelleen komennolla `fg`, ohjata tausta-ajoksi komennolla `bg` tai poistaa `kill`-komennolla.

Keskeytetyn tai taustalla ajettavan komennon tai ohjelman voi lopettaa etsimällä kyseisen prosessin tunnuksen (PID) ja antamalla sen `kill`-komennolle. Toiminnassa olevat prosessit voi listata käyttämällä komentoa `ps` seuraavasti:

```
corona1 ~>ps
PID    TTY  TIME CMD
  61226  pts/6  0:00 ps
 554282  pts/6  0:00 -csh
```

Komento `ps` näyttää vain kyseisen pääteyhteyden aikana käynnistetyt ohjelmat. Jos haluat tarkistaa kaikki käynnistämäsi prosessit, täytyy komentoa muuttaa seuraavasti:

```
ps -fu käyttäjätunnus
```

Tämä komento antaa listan, josta näkyvät käynnissä olevien prosessien tunnukset. Haluttu prosessi keskeytetään komennolla `kill` *PID-tunnusluku*, esimerkiksi:

```
kill 123456
```

3.5 Teksti- ja kuvaeditorit

Tekstitiedostojen sisällön muokkaus ja uusien tekstitiedostojen luominen tapahtuu Unix-ympäristössä tekstieditoreilla. Niitä käytetään tekstinkäsittelyohjelmien tapaan. Ne ovat tarpeellisia myös esimerkiksi muokattaessa parametri, lista- ja sekvenssitiedostoja.

CSC:n palvelimilla on käytössä useita erilaisia editoreita. Käytetyimmät niistä ovat nano ja emacs. Muita yleisesti käytettyjä editoreita ovat pico, vi, xemacs, nedit ja gedit. Sekvenssitiedostojen muokkausta varten on lisäksi erityisiä sekvenssieditoreita, kuten Seaview ja MSE (katso kappale 8.3)

3.5.1 Nano

Nano on niin sanottu pikkueditori. Se on helppokäyttöinen ja hyvä aloittelevalle käyttäjälle pienten tiedostojen tuottamiseen ja muokkaamiseen. Nanoa voidaan käyttää myös ilman X-pääteyhteyttä. Nanon ulkoasu ja komennot muistuttavat paljolti Pine-sähköpostiohjelmaa.

CSC:n palvelimilla Nano käynnistyy komennolla `nano` *tiedosto*. Esimerkiksi komento

```
nano tekstitiedosto.txt
```

avaa tiedoston `tekstitiedosto.txt`, jos kyseinen tiedosto on olemassa. Jos tiedostoa ei ole, ohjelma avaa uuden tyhjän tiedoston ja ehdottaa komennossa annettua nimeä tallennettaessa.

Kun Nano-editori on käynnistynyt, sen peruskomennot näkyvät pääteikkunan alareunassa. Komentoja käytetään painamalla samanaikaisesti Ctrl-näppäintä (johon ohjelma viittaa merkillä ^) ja jotakin kirjainta. Paljon käytettyjä Nano-editorin käskyjä ovat help-valikkoon siirtyminen Ctrl-g, Nanosta poistuminen Ctrl-x, tiedoston tallettaminen Ctrl-o sekä tiedoston luku ohjelmaan Ctrl-r. Jos ruutu sisältää Ctrl-r-käskyä käytettäessä muuta tekstiä, liitetään toinen tiedosto aiemman tekstin jatkoksi. Nano-editori mahdollistaa tekstin selaamisen sivuittain käskyillä Ctrl-y (edellinen sivu) ja Ctrl-v (seuraava sivu). Sanoja voi hakea tekstin seasta käskyllä Ctrl-w ja kursorin sijainnin voi tarkistaa käskyllä Ctrl-c. Tekstin leikkaamiseksi valitaan ensin tekstialue kursorilla, valinta aloitetaan painamalla Ctrl-^, minkä jälkeen kursori siirretään valitun alueen loppuun. Valitun tekstin leikkaaminen tapahtuu Ctrl-k

Kuva 3.3: *nano-editorin käyttöliittymä.*

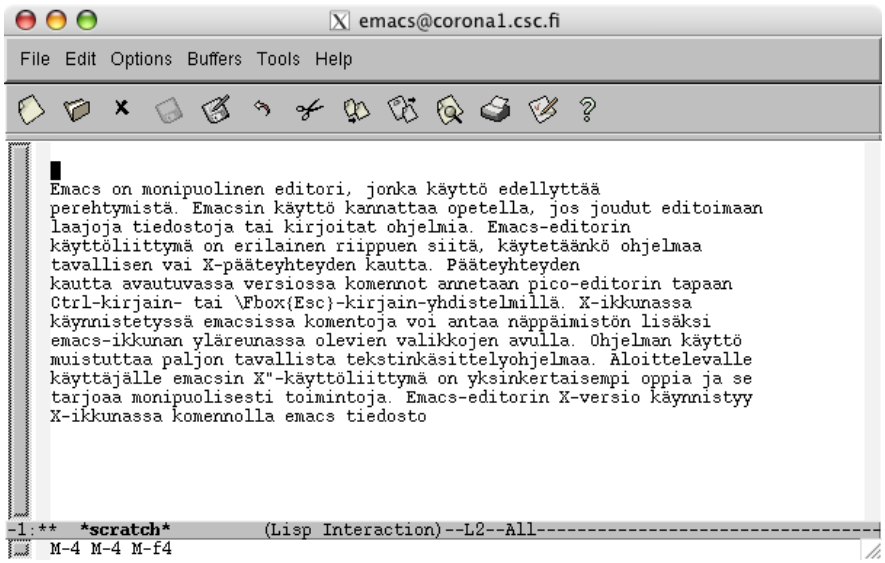
komennolla, minkä jälkeen teksti voidaan liittää haluttuun kohtaan komennolla Ctrl-u.

3.5.2 GNU emacs

Emacs on monipuolinen editori, jonka käyttö edellyttää perehtymistä. Emacsin käyttö kannattaa opetella, jos joudut editoimaan laajoja tiedostoja tai kirjoitat ohjelmia. Emacs-editorin käyttöliittymä on erilainen riippuen siitä, käytetäänkö ohjelmaa tavallisen vai X-pääteyhteyden kautta. Pääteyhteyden kautta avautuvassa versiossa komennot annetaan Nano-editorin tapaan Ctrl-kirjain- tai Esc-kirjain-yhdistelmillä. X-ikkunassa käynnistetyssä emacsissa komentoja voi antaa näppäimistön lisäksi emacs-ikkunan yläreunassa olevien valikkojen avulla. Ohjelman käyttö muistuttaa paljon tavallista tekstinkäsittelyohjelmaa. Aloittelevalle käyttäjälle emacsin X-käyttöliittymä on yksinkertaisempi oppia ja se tarjoaa monipuolisesti toimintoja. Emacs-editorin X-versio käynnistyy X-ikkunassa komennolla `emacs` *tiedosto*.

```
emacs tekstitiedosto &
```

Files-valikon kautta avataan ja talletetaan tiedostoja sekä poistutaan emacsista.*Edit*-valikko sisältää editointitoimintoja, mm. kopioi-, liitä-, etsi- ja korvaatoiminnot. *Buffers*-valikosta voi valita tarkasteltavan tiedoston auki olevien tiedostojen joukosta (samanaikaisesti voi olla auki useita tiedostoja). *Tools*-valikko sisältää sekalaisia toimintoja, kuten tiedostojen vertailun ja yhteen liittämisen. Emacsin ominaisuuksia on käsitelty tarkemmin Metakoneen käyttöoppaassa (GNU-Emacsia käsittelevä kappale).



Kuva 3.4: *Emacs-editorin X-käyttöliittymä.*

3.5.3 Kuva- ja PostScript-tiedostojen tarkastelu

Monet tässä oppaassa käsiteltävät ohjelmat tuottavat tekstitiedostojen lisäksi myös kuvatiedostoja. Coronassa olevia kuvatiedostoja voi tarkastella Tutkijan käyttöliittymässä tai X-pääteyhteyden yli toimivilla `eog` ja `ggv`- ohjelmilla. Eog on yksinkertainen kuvankäsittelyohjelma, joka pystyy lukemaan yleisimpiä kuvatiedostotyyppejä kuten png, bmp jpg, ja tiff. Ohjelma ei kuitenkaan osaa lukea PostScript tai PDF tiedostoja.

Coronasta avatussa X-ikkunassa eog käynnistyy komennolla `eog` *tiedosto*. Esimerkiksi:

```
eog kuva1.png
```

PostScript (.ps) ja PDF-tiedostoja voidaan Coronassa tarkastella `ggv`-ohjelmalla.

```
ggv kuva1.ps
```

PostScript-muotoisia kuvia voidaan Coronassa muuntaa PDF-muotoon komennolla `ps2pdf`. Ohjelma lukee annetun PostScript-tiedoston ja tuottaa saman nimisen PDF-tiedoston. Esimerkiksi komento:

```
ps2pdf kuva1.ps
```

tuottaa PDF-tiedoston nimeltä kuva1.pdf.

3.6 Eräajot

Coronassa voidaan normaalin pääteyhteyden kautta suorittaa töitä, jotka kestävät enintään kaksi prosessorituntia. Tätä pidemmät työt on lähetettävä ajoon niin sanottuina eräajoina. Eräajoissa käyttäjä kokoaa suoritettavat komennot tekstimuotoiseen eräajotiedostoon. Tiedoston alussa määritellään myös, miten paljon laskenta-aikaa ja muistikapasiteettia työ enimmillään tulee tarvitsemaan. Valmis eräajotiedosto lähetetään eräajojärjestelmälle, joka käynnistää työn kun pyydetyt resurssit ovat saatavilla. Alla olevat esimerkit esittelevät Coronassa on käytössä olevan Sun N1 Grid Engine eräajojärjestelmän peruskomennot. Esimerkit eivät siis toimi koneissa, joissa on jokin muu eräajojärjestelmä (esim. LSF tai PBS).

Coronassa eräajotiedoston voi luoda esimerkiksi nano-editorilla. Eräajotiedoston alkaa komentotulkin määrittelyrivillä (CSC:llä yleensä tcsh):

```
#!/usr/bin/tcsh
```

Tämän jälkeen määritellään työn vaatimat resurssit. Nämä määrittelyrivit alkavat merkinnällä `#$` minkä jälkeen tulevat eräajomäärittelyt. Coronassa ainoa pakollinen määrittely on varattava laskenta-aika, joka määritellään valitsimella `-l h_rt`. Aikamäärittely annetaan muodossa:

`#$ -l h_rt=`*tunnit*:*minuuttit*:*sekunnit*

Coronassa pisin sallittu laskenta-aika on 168 h.

Eräajomäärittelyjen jälkeen tulevat suoritettavat unix-komennot. Komentojen yhteydessä on annettava kaikki tarvittavat parametrin, sillä erätöissä parametreja ei voida syöttää interaktiivisesti. Eräajotiedostoon on sisällytettävä myös mahdollisten ohjelmien alustuskomennot sekä siirtyminen kotihakemistosta siihen hakemistoon jossa työ suoritetaan. Alla esimerkki eräajotiedostosta.

```
#!/usr/bin/tcsh
#$ -l h_rt=06:30:00
cd $METAWRK
use emboss
ehmmpfam /qfs2/biodb/pfam/Pfam_ls sekvenssi1.fasta -a 0  \
 -e 0.001 -outfile Pfam_osumat
```

Tiedoston ensimmäisellä rivillä määritellään työn vaatimaksi laskenta-ajaksi 6h 30 min. Seuraavat rivit sisältävät komennot, joilla: 1. siirrytään metatyöhakemistoon, 2. alustetaan EMBOSS-ohjelmisto, 3. käynnistetään ehmmpfam ajo. Valmis eräajotiedosto lähetetään ajoon komennolla `qsub`. Esimerkiksi eräajotiedosto nimltä `era.cmd` lähetetään ajoon komennolla:

```
qsub era.cmd
```

Työ etenemistä voidaan seurata komennoilla

```
qstat
```

ja

```
qstat -u käyttäjätunnus
```

Qstat komenon tulosteesta näet mm. työn tunnusnumeron (job-ID) ja tilan (State). Tila-sarakkeen arvo kertoo onko työsi käynnistynyt (r) vai onko se edelleen jonossa (qw). Virheellinen työ voidaan poistaa tai keskeyttää komennolla

```
qdel tunnusnumero
```

Joillekin ohjelmille kaikkien ohjelman tarvitsemien parametrien syöttäminen ei onnistu komentorivillä (esim. Phylip, TNT ja MrBayes). Interaktiivinen parametrien syöttäminen voidaan eräjoissa korvata <<EOF -rakenteella, jossa ohjelman pyytämien parametrien arvot listataan *komento«EOF* -merkintää seuraavilla riveillä. Parametrit tai komennot on syötettävä täsmälleen siinä järjestyksessä, jossa ohjelma ne kysyy. Parametrilista päätetään rivillä EOF.

Mikäli edellä käytetty komento ehmmpfam annetaan ilman parametrejä se kysyy tarvitsemansa tiedot interaktiivisesti (Pfam-tietokannan sijainti, hakusekvenssi, tulostustapa, e-arvo ja tulostiedoston nimi.) Eräajotiedostossa parametrit voitaisiin siis syöttää myös käyttäen <<EOF -rakennetta seuraavasti:

```
#!/usr/bin/tcsh
#$ -l h_rt=06:30:00
cd $METAWRK
use emboss
ehmmpfam<<EOF
/qfs2/biodb/pfam/Pfam_ls
sekvenssi1.fasta
0
0.001
Pfam_osumat
EOF
```

Edellä olleet esimerkit esittelevät vain aikavarauksen käyttöä eräajotöiden yhteydessä, mutta ajoa on mahdollista säädellä myös monilla muilla tavoilla. Tauluko 3.3 sisältää yleisimmin Coronassa käytettyjä eräajomäärittelyjä. Määrittelyt voidaan lisätä komentotiedostoon #$ alkuisille riveille tai ne voidaan antaa qsub-koennon yhteydessä.

Esimerkiksi alla oleva eräajotiedosto suoritaa estwisedb vertailun proteiinisekvenssi ja nukleotidisekvenssijoukon välillä. Työ kirjoittaa sekä normaali tulostuksen (-o) että vikailmoitukset (-e) tiedostoon estwise_ajo.log. Työn maksimikesto on 8 tuntia ja se varaa käyttöönsä 1GB muistia. Työ ajetaan käyttäen 4 prosessoria. Työ käynnistetään siinä hakemistossa, josta se on lähetetty (-cwd), Kun työ päättyy se lähettää sähköpostia osoitteeseen kkayttaj@yliopisto.fi.

```
#!/usr/bin/tcsh
#$ -o estwise_ajo.log
#$ -e estwise_ajo.log
#$ -cwd
#$ -l h_rt=08:00:00
#$ -l h_vmem=1G
#$ -pe cre 4
#$ -M kkayttaj@yliopisto.fi
#$ -m e
```

Taulukko 3.3: *Coronan eräajojärjestelmän tärkeimmät valitsimet.*

Valitsin	*Selitys*
`-N` *työn_nimi*	Työn nimi
`-o [`*/polku/*`]`*tiedosto*	Eräajon normaali tulostustiedosto (standard output)
`-e [`*/polku/*`]`*tiedosto*	Eräajon virheiden tulostustiedosto (standard error)
`-l h_rt=`*h:min:sek*	Työlle pyydettävä laskenta-aika. (Huom. valitsimessa on pieni L-kirjain.)
`-l v_mem=`*muisti*`M`	Työlle varattavan muistin koko (M = megatavu, G = gigatavu)
`-cwd`	Asetetaan eräajon käynnistyshakemistoksi kotihakemiston sijaan se hakemisto, jossa lähetyskomento (`qsub`) annetaan.
`-pe cre` *pros.lkm*	Käytetään cre-rinnakkaislaskentaympäristöä tietyllä prosessorimäärällä.
`-t 1-`*n*	Asetaan joukko yhtäaikaisia peräkkäistöitä (n.s. taulukkoerätyöt, array jobs), joilla on tehtävätunnukset (`$SGE_TASKID`) 1-*n*.
`-M` *sähköpostiosoite*	Sähköpostiosoite tai osoitteet, joihin eräajotyö lähettää -m parametrilla määritetyt ilmoitukset.
`-m b ja/tai e`	Määritellään, milloin eräajojärjestelmä lähettää ilmoituksen: työn alussa (b) lopussa (e). Ilmoitusta ei lähetetä, mikäli valitsinta ei käytetä.

```
use hmmer
estwisedb -pthread -pthr_no 4 -kbyte 1000000 -prodb \
proteiinit.fasta estit.fasta > uniprot.osumat3
```

Lisätietoja Coronan eräajojärjestelmästä löytyy Coronan käyttöoppaan luvusta 3. Laajojen aineistojen analysoinnissa hyödyllisten taulukkoerätöiden käyttöä esitellään @CSC-lehden artikkelissa @CSC 4/2005 s.9.

3.7 Unix-esimerkki: Arkistohakemiston käyttö

CSC:n arkistopalvelin on tarkoitettu suurten tiedostojen ja hakemistojen pitkäaikaiseen säilytykseen. Tallennettavan datan määrällä ei ole kokorajoitusta. Arkistoitu data voidaan ottaa takaisin käyttöön niin kauan, kuin käyttäjätunnus on voimassa.

Alla on esimerkki tiedon tallentamisesta arkistopalvelimelle. Siirrytään aluksi hakemistoon, jossa arkistoitavat tiedostot ovat. Tässä tapauksessa tallennettavat tiedostot ovat $METAWRK-hakemistossa.

```
corona1 >cd $METAWRK
```

Tarkistetaan ensin hakemiston sisältö ls-komennolla. Tämän jälkeen luodaan uusi hakemisto, johon arkistoitavat tiedostot siirretään:

```
corona1 /fs/metawrk/kkayttaj> ls
input.dat  sekvenssi1.fasta  sekvenssi2.fasta  tulos.ps
corona1 /fs/metawrk/kkayttaj> mkdir arkisto1
```

Kopioidaan arkistoitavat tiedostot uuteen hakemistoon:

```
corona1 /fs/metawrk/kkayttaj> cp input.dat arkisto1
corona1 /fs/metawrk/kkayttaj> cp *.fasta arkisto1
corona1 /fs/metawrk/kkayttaj> cp tulos.ps arkisto1
```

Tarkistetaan, että kopiointi onnistui komennolla:

```
corona1 /fs/metawrk/kkayttaj> ls arkisto1
input.dat  sekvenssi1.fasta  sekvenssi2.fasta  tulos.ps
```

Hakemisto arkisto1 pakataan `tar`-komennolla yhdeksi *arkisto1.tar*-nimiseksi tiedostoksi. Tämä tiedosto kompressoidaan `gzip`-komennolla niin, että se vie vähemmän tilaa.

```
corona1 /fs/metawrk/kkayttaj> tar cvf arkisto1.tar arkisto1

corona1 /fs/metawrk/kkayttaj> gzip arkisto1.tar
```

Tuloksena saatu *arkisto1.tar.gz*-tiedosto siirretään arkistohakemistoon komennolla

```
corona1 /fs/metawrk/kkayttaj> cp arkisto1.tar.gz $ARCHIVE
```

Siirron onnistumisen voi tarkistaa komennolla

```
corona1 /fs/metawrk/kkayttaj> ls $ARCHIVE
arkisto1.tar.gz
```

Tämän jälkeen arkisto1-hakemisto ja arkisto1.tar.gz tiedosto voidaan poistaa työskentelyhakemistosta:

```
corona1 /fs/metawrk/kkayttaj> rm arkisto1.tar.gz
corona1 /fs/metawrk/kkayttaj> rm -r arkisto1
```

Arkistoidun tiedoston palauttaminen työskentelyhakemistoon tapahtuu seuraavasti. Kopioidaan arkistoitu tiedosto työskentelyhakemistoon.

```
corona1 /fs/metawrk/kkayttaj> cp $ARCHIVE/arkisto1.tar.gz ./
```

Puretaan kompressointi gunzip-komennolla ja aukaistaan tiedostopaketti hakemistoksi tar-komennolla.

```
corona1 /fs/metawrk/kkayttaj> gunzip arkisto1.tar.gz
corona1 /fs/metawrk/kkayttaj> tar xvf arkisto1.tar
```

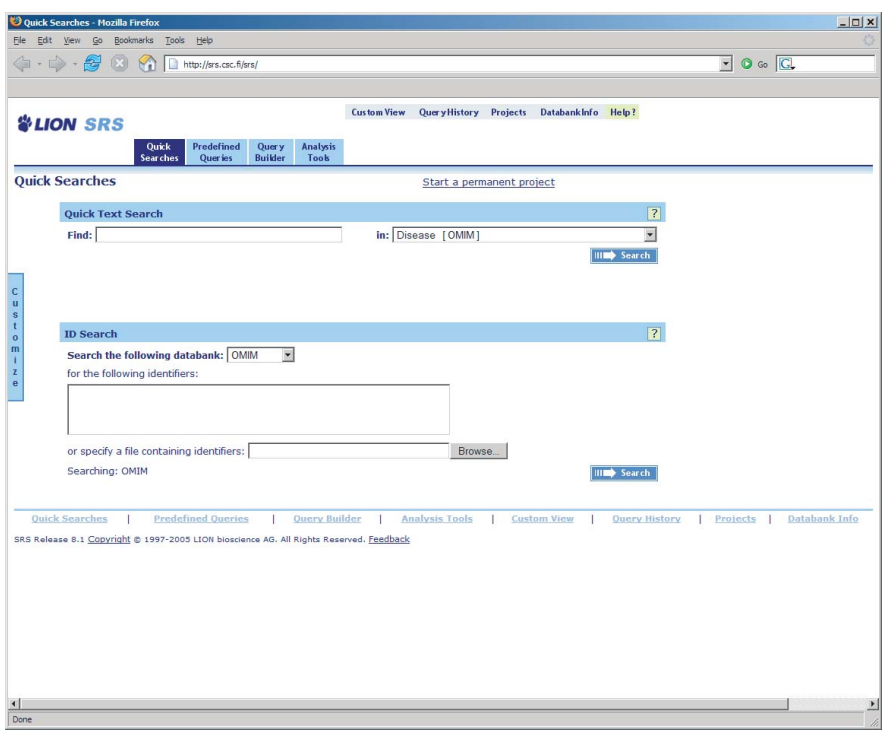
3.8 Lisätietoa CSC:n Unix-ympäristöstä

Enemmän tietoa Unix-käyttöjärjestelmästä ja CSC:n palvelinten käytöstä löytyy osoitteesta:

```
http://www.csc.fi/tutkimus/Laskentapalvelut/
```

4 Tietokantahakuohjelma SRS

Verkkoselaimella toimiva SRS (Sequence Retrieval System) on hyödyllinen apuväline etsittäessä sekvensejä tai muita tietoja erilaisista biotietokannoista. SRS-haut ovat hakutermeihin perustuvia tekstihakuja. Sillä voi siis esimerkiksi hakea sekvenssejä tietokantakoodin, geenin nimen, geenituotteen, organismin nimen jne. avulla. Ohjelma soveltuu hyvin myös tilanteisiin, joissa etsittävän sekvenssin tietokantakoodia tai tarkkaa nimeä ei tunneta. Esimerkiksi EMBOSS:issa käytettävät sekvenssitunnukset voi helposti selvittää SRS:n avulla.



Kuva 4.1: *SRS:n aloitussivu.*

SRS-ohjelmalla on myös mahdollista hakea tietoja useasta tietokannasta samanaikaisesti. On kuitenkin syytä huomata, että tällöin ovat käytössä vain ne hakukentät, jotka ovat kaikille haettaville tietokannoille yhteisiä. Hakuehtoja yhdistelemällä haut voidaan rajata hyvinkin tarkasti. Hakutulokset voidaan

esittää ja tallettaa eri muodoissa.

CSC:n ylläpitämä SRS löytyy osoitteesta `http://srs.csc.fi/`. Ohjelma aukeaa *Quick Searches* -sivulle (kuva 4.1). Tältä sivulta on mahdollista tehdä yksinkertaisia kyselyjä joihinkin tietokantoihin. Yleensä tosin parempiin tuloksiin pääse varsinaisten hakusivujen kautta. Sivun yläreunasta löytyy linkit mm. SRS:n käyttöohjeisiin (*Help*), sekä käytettävissä olevien tietokantojen tietoihin (*DataBankInfo*).

4.0.1 Tietokantojen valinta

Kysely aloitetaan siirtymällä sivulle *Query Pages* (kuva 4.2). Käytettävissä olevat tietokannat on listattu kohdassa *Available Data Sources*. Ne on jaettu sisällön mukaan luokkiin. Kukin luokka sisältää edelleen alaluokkia. Esimerkiksi luokka *Sequence* sisältää alaluokat *Nucleid Acid Sequence* ja *Protein Sequence*. Alaluokat saa näkyviin klikkaamalla + merkkiä luokan nimen edessä tai valitsemalla optio *Expand All*.

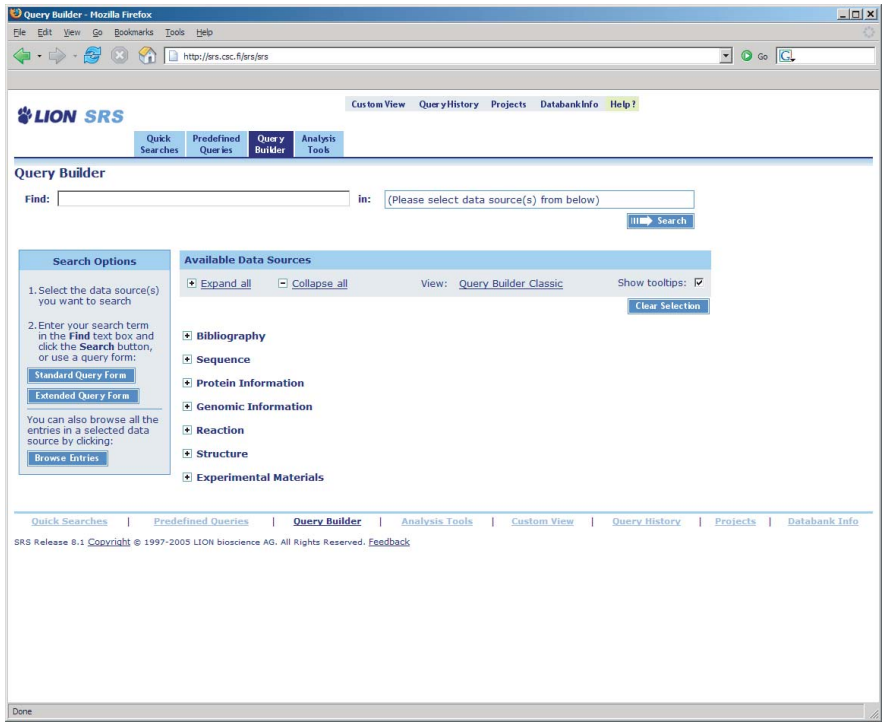
Alaluokkien nimen jäljessä on laatikko, jossa teksti *DB*. Tätä laatikkoa klikkaamalla saa esiin listauksen itse tietokannoista. Tietokannat valitaan käyttöön nimen edessä olevalla laatikolla. Tietokannan nimi toimii linkinä tietokannan kuvaukseen. Tietokantoja voi valita niin monta kuin on tarpeen, myös useista luokista. Vanhaan SRS-versioon tottuneet voivat valita optiot *Expand all* ja *Query Builder Classic*, jolloin tietokantojen valintanäkymä vastaa vanhaa versiota.

CSC:n SRS sisältää mm. keskeisimmät sekvenssi-, tunniste- ja rakennetietokannat (liite A). Laajempi tietokantavalikoima löytyy mm. EBI:n SRS-palvelimelta (`http://srs.ebi.ac.uk`).

4.0.2 Hakuehtojen määrittely

Hakuehtojen määrittelyyn on useita vaihtoehtoja. *Query Builder* -sivun yläreunassa on laatikko, johon hakutermin voi syöttää pikahakua varten. Usein tällä tavalla on kuitenkin vaikea tehdä riittävän spesifistä hakua, joten on suositeltavaa käyttää yksityiskohtaisempia hakusivuja. Sivun vasemmassa reunassa on *Search Options* -kenttä, josta edelleen löytyy valinnat *Standard Query Form* ja *Extended Query Form*. Hakulomakkeiden ero on lähinnä siinä, että *Extended* versiossa kaikki mahdolliset kentät on näkyvissä samanaikaisesti, kun taas *Standard* versio on hieman yksinkertaisempi. Seuraavassa käsittelemme lähinnä *Standard*-versiota.

Standard Query Form (Kuva 4.3) on jaettu alueisiin, joihin on keskitetty eri tyyppisiä toimintoja. Vasemmalla ylhäällä on *Search Options* -alue, josta löytyy hakuehtoihin liittyviä valintoja. Voit esimerkiksi valita, millä operaattorilla yhdistät hakusanat. AND-operaattori vaatii, että molemmat/kaikki hakusanat löytyvät. OR-operaattorille riittää, että joku hakusanoista löytyy.



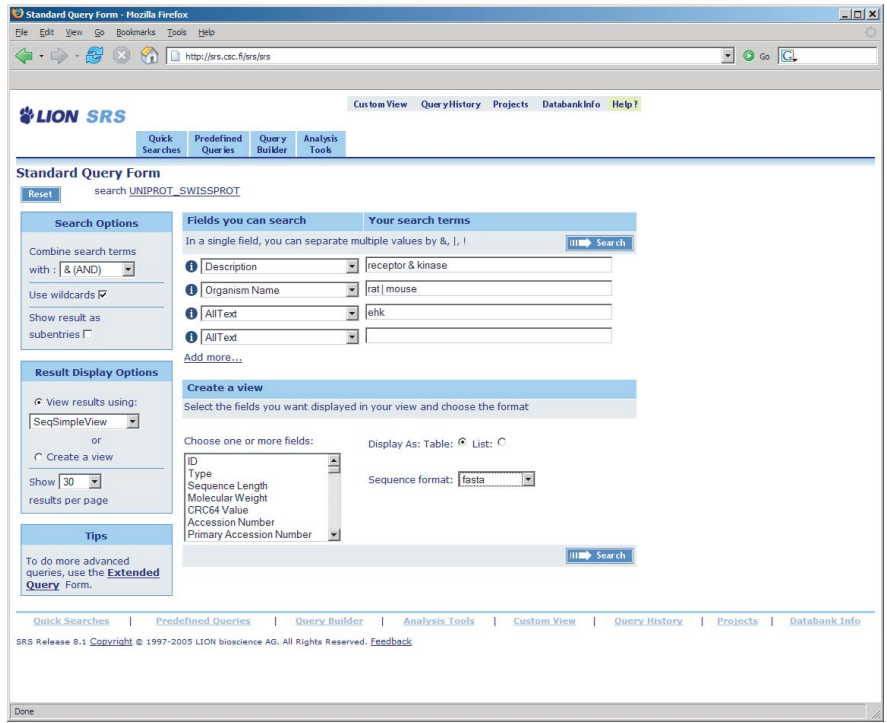
Kuva 4.2: *SRS:n tietokantavalintasivu.*

BUTNOT-operaattori taas edellyttää, että toinen, mutta ei molemmat hakusanat löytyvät. Tästä kohdasta voit myös valita (*Use wildcards*) etsitäänkö tarkkaa vastinetta hakusanoillesi vai lisätäänkö hakutermeihin ns. jokeri-merkki, jolloin etsitään kaikki sanat, jotkavat alkavat hakusanallasi.

Hakusanat syötetään hakukenttiin, joita on aluksi neljä, mutta niitä voi tarvittaessa lisätä valitsemalla *Add more.*. kenttien alapuolelta. Kunkin kentän vasemmassa sarakkeessa on alasvetovalikko, josta voit valita mistä tietokannan kentästä hakusanaasi etsitään. Oikeassa sarakkessa on tekstilaatikko, johon itse hakusana syötetään. Yhteen tekstilaatikkoon voi myös syöttää useamman sanan ja ne voi erottaa haluamillaan operaattoreilla: `&` (AND), `|` (OR) tai `!` (BUTNOT).

Oletusarvoisena hakukenttänä on *AllText*, jolloin haku suoritetaan kaikkiin kenttiin. Usein pääsee kuitenkin parempiin tuloksiin, kun haut keskitetään spesifisiin kenttiin. Esimerkiksi EMBL tietokannassa organismin nimi kannattaa hakea *Organism name* -kentästä, koska muuten löytyy myös suuri joukko sekvenssejä, joiden *Description* -kentässä mainitaan hakuorganismin nimi (`"similar to..."`), vaikkei sekvenssi olekaan haettavasta organismista.

Kuvan 4.3 esimerkissä haetaan UniProt/Swissprot-tietokannasta tietueita, joiden *Description*-kentässä esiintyy sanat `kinase` ja `receptor`. Toisessa hakukentässä hakuehto kohdistetaan sekvenssien lähdeorganismiin (*Organism name*) ja määritellään, että sekvenssi saa olla joko hiirestä tai rotasta (`mouse`



Kuva 4.3: *SRS:n hakusivu.*

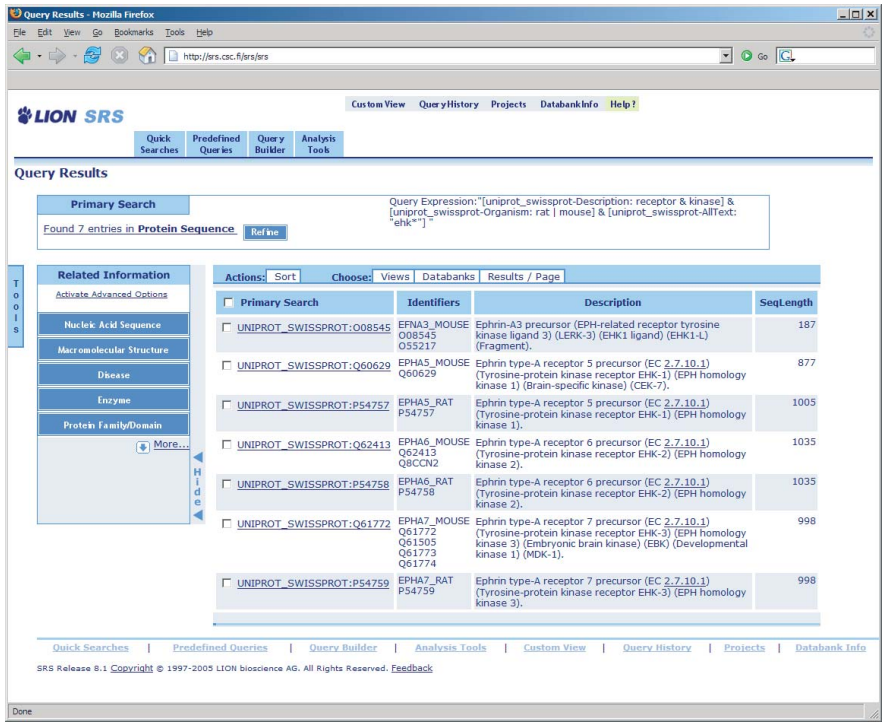
| `rat`). Lisäksi jostain tekstikentästä (*AllText*) täytyy löytyä lyhenne `ehk`.

Hakuja tehtäessä kannattaa muistaa, että terminologia vaihtelee eri tietokannoissa. Eri hakukenttien sisällöstä saa tietoja valisemalla kentän alasvetovalikosta ja klikkaamala *i*-painiketta sen vieressä. Tietosivulta voi myös tarkistaa, kuinka monta kertaa hakusana esiintyy valitun tietokannan tietyssä kentässä. Esimerkisi etsittäessä EMBL-tietokannan *Organism name* -kentästä *mus*-alkuiset sanat (List values that match *mus**) havaitaan, että *mus musculus* esiintyy tietokannassa miljoonia kertoja. Tietokannassa esiintyy myös muita *mus*-alkuisia lajeja, esimerkiksi *Musca domestica*.

Hakutulokset voidaan esittää erilaisissa lista- ja taulukkomuodoissa. *Results Display Option* kohdasta voit valita jonkin valmiiksi määritellyistä esitystavoista tai määritellä oman esitystapasi valitsemalla ensin optio *Create View* ja sitten valitsemalla haluamasi kentät *Create a View* alueelta.

4.0.3 Hakutulokset ja tallennus

Hakutulossivun (kuva 4.4) ylälaidassa on *Primary Search* -alue, jossa kerrotaan löydettyjen tulosten lukumäärä ja näytetään SRS-palvelimelle lähetetty kysely. Mikäli haku ei ollut tarpeeksi spesifinen, sitä on mahdollisuus tarkentaa alueella olevasta *Refine*-napista.



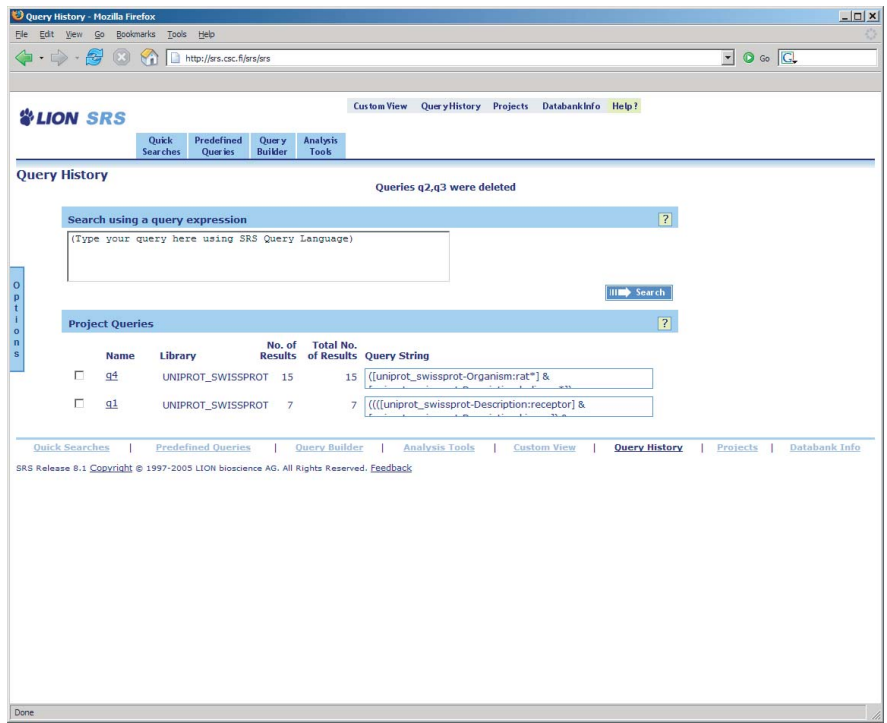
Kuva 4.4: *Esimerkkihaulla saadut tulokset.*

Sivulta on myös mahdollista vaihtaa tulosten esitystapaa (*Views*), valita vain yhdestä tietokannasta tulleet tulokset (*Databanks*) ja määrittää montako tulosta yhdellä sivulla esitetään (*Results/Page*). Itse hakutulokset esitetään valitussa muodossa ja yksittäisen tietueen saa avattua klikkaamalla sen nimeä. Valitsimesta *Sort* tulokset voidaan järjestää erilaisten ominaisuuksien mukaan (esimerkiksi sekvenssin pituus tai tietueen luomisajankohta). Järjestäminen ei kuitenkaan ole mahdollista, kun on käytetty useampaa kuin yhtä tietokantaa.

Tulostietueista voi etsiä niihin liittyviä tietoja *Related Information* -alueen toiminoilla. Esimerkisi Swissprot-haun tuloksista voi tarkistaa, löytyykö jollekin sekvenssille rakenne PDB-tietokannasta. Tämä tapahtuu valitsemalla *Related Information* -alueelta *Macromolecular Structure* ja edelleen *Primary Search -> Related Information*. Sivun vasemmassa laidassa on lisäksi *Tools*-valikko, joka aukeaa klikkaamalla. Siinä on työkaluja hakutolosten tallentamiseen ja siirtämiseen Excel-muotoon.

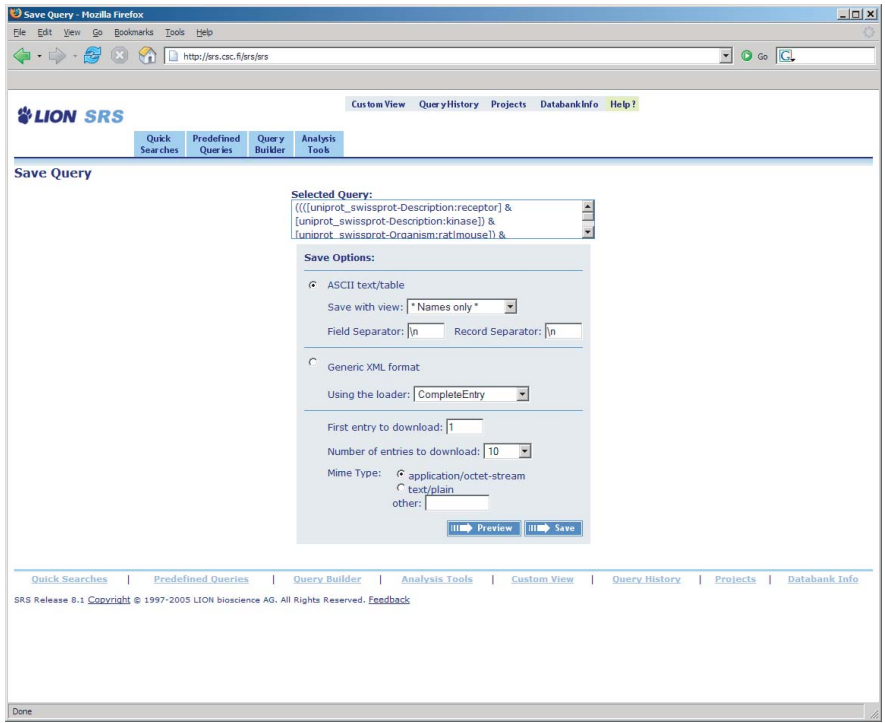
Uusi SRS-haku kannattaa käynnistää WWW-sivun yläreunassa olevasta *Query Builder* -linkistä. Näin edellinen haku säilyy tallessa ja sen voi tarvittaessa yhdistää uusiin hakuihin tulostenhallintasivulla, joka aukeaa *QueryHistory* -linkistä. Toiminot löytyvät *Options*-valikosta sivun vasemmasta reunasta. SRS-istunnon voi tarvittaessa keskeyttää ja tallentaa omalle koneelle *Projects*-sivun *Options*-valikosta klikkaamalla *Save*. Kun istuntoa jatketaan, se ladataan napilla *Open*.

Kuvassa 4.5 on esimerkki tulostenhallintasivusta. Istunnon aikana on ajettu kaksi SRS-hakua, *q1* ja *q4*. Hakutulokset voidaan yhdistää *Options*-valikon painikeella *Combine*. Jos yhdistämistavaksi valitaan AND, poimitaan ne sekvenssit, jotka olivat molempien hakujen tuloksissa. OR-vaihtoehto valitsee kaikki hauissa 1 ja 4 saadut sekvenssit. Kolmas vaihtoehto BUTNOT valikoi ne sekvenssit, jotka olivat haun 1 tuloksissa, mutta eivät haun 4 tuloksissa.



Kuva 4.5: *SRS:n tulossivu.*

SRS-haun tulokset voi tallettaa omalle koneelle *Options*-valikon valitsimella *Save*, joka aukaisee tallennussivun (kuva 4.6). Mikäli sekvenssejä halutaan jatkokäsitellä Tutkijan käyttöliittymässä tai CSC:n laskentapalvelimilla, kannattaa yleensä valita muoto *ASCII text/table*. Kohdasta *Save with view* valitaan tulosten tallennustapa, esimerkiksi tietueiden koko sisältö (*Complete enries*), sekvenssi fasta-muodossa (*FastaSeqs*) tai pelkkä nimilista (*Names only*), joka voidaan helposti muokata EMBOSSin listatiedostoksi. Käyttäjä voi myös valita sarakkeen- ja rivinvaihtoa tarkoittavat merkit. Valitsimilla *First entry to download* ja *Number of entries to download* voidaan valita tallennettavat tietueet. Kannattaa huomata, että valitsimen *Number of entries to download* oletusarvo on 10. Useimmiten tähän kuitenkin kannattaa valita *All*. Tallennettava tieto voidaan avata esikatseluun selaimeen (*Preview*) tai tallentaa suoraan oman koneen kovalevylle (*Save*). Mikäli tallennettava tiedosto aukaistaan ensin näkyviin selaimeen, se voidaan tallentaa selaimen omalla tallennuskomennolla (esim. *File/Save as*). Tiedostotyypiksi kannattaa yleensä valita `.txt`, jolloin tuloksia voi tarkastella esim. WordPad-ohjelmalla.



Kuva 4.6: *SRS:n tallennussivu.*

4.0.4 SRS:n komentorivikäyttö

SRS-hakuja voidaan tehdä Coronan komentorivikäytössä `getz` -komennon avulla. SRS:n komentorivikäyttö tarjoaa nopean tavan hakea SRS:n www-käytöliittymässä määritelty sekvenssijoukko analysoitavaksi esimerkiksi Coronan BLAST- tai EMBOSS-ohjelmilla.

Komento `getz` tulee käyttöön komennolla `use emboss` tehtävän

EMBOSS-alustuksen yhteydessä. Komennon perussyntaksi on:

`getz` 'genkysely-lauseke'

Esimerkiksi alla oleva komento tulostaa kaikkien niiden UniProt-tietokannan sekvenssien tunnusnumerot, joiden Description-kentässä on merkkijono: prion.

```
getz '[uniprot-Description: "Prion*"]'
```

Huomaa että kyselylauseke tulee yksinkertaisten lainausmerkkien sisään. Lisäämällä komentoon valitsimet `-d -sf fasta`, ohjelma tulostaa löytyneet sekvenssit fasta-muodossa. Tuloksena saatava sekvenssidata voidaan ohjata uuteen tiedostoon »-merkillä. UniPriot-tietokannan prioni-sekvenssit voidaan siis poimia yhdeksi fasta-muotoiseksi sekvenssitiedostoksi komennolla:

```
getz -d -sf fasta '[uniprot-Description: "Prion*"]'
> prionit.fasta
```

Käyttämällä muita valitsimia, voidaan tulostettavaksi valita myös esimerkiksi koko tietue (-e) tai löydettyjen sekvenssien lukumäärä (-c). Getz-komennon muut valitsimet saat näkyviin komennolla:

```
getz -help
```

Käytännössä monimutkaisempia kyselylausekkeita ei kannata kirjoittaa käsin, vaan käyttää SRS:n normaalia www-käyttöliittymää haun määrittelyyn. Kun sopiva hakuehto on löydetty, se on helppoa kopioida SRS:stä ja getz-komennon syötteeksi. Käytetty kyselylauseke näkyy SRS:n tulossivun oikeassa yläreunassa *Query Expression:* merkinnän vieressä. Esimerkiksi aiemmin tehdyssä haussa löydetyt sekvenssit voitaisiin hakea Coronassa komennoilla.

```
corona1 ~>  use emboss

corona1 ~>  getz -d -sf fasta ' kopioi tähän hakulauseke'
> srs_tulos.fasta
```

5 Sekvenssianalyysiohjelmisto EMBOSS

Tässä luvussa käsitellään EMBOSS-ohjelmiston yleisiä piirteitä sekä ohjelman käyttöä CSC:llä. EMBOSS on asennettu CSC:n `corona.csc.fi` -palvelimelle ja sitä voidaan käyttää Tutkijan käyttöliittymän tai pääteyhteyksien avulla. EMBOSS sisältää yli 200 nukleotidi- ja proteiinisekvenssien käsittely- ja analysointiohjelmaa. Ohjelmien tarkat kuvaukset löytyvät CSC:n EMBOSS-sivuilta sekä aakkosellisena listana että toiminnan mukaan luokiteltuna. Yleisiä EMBOSS-ohjelmien käyttökohteita ovat mm.

- sekvenssien haku tietokannoista
- sekvenssien vertailut ja rinnastukset
- restriktioanalyysi
- alukkeiden suunnittelu
- sekvenssien ominaisuuksien laskeminen ja kuvantaminen
- sekvenssimotiivien etsintä
- sekvenssitiedostojen esittäminnen ja formaattimuutokset

Varsinaisten EMBOSS-ohjelmien lisäksi EMBOSS pakettiin kuuluu myös joukko n.s. EMBASSY-ohjelmia, jotka ovat EMBOSS-muotoon muutettuja ulkopuolisia ohjelmistoja. CSC:llä on käytettävissä EMBASSY-versiot seuraavista ohjelmista

- HMMER, proteiinen HMM-analyysi
- MEME, sekvenssimotiivien tunnistutsMEME)
- MSE, sekvenssieditori
- PHYLIP, fylogenetiikka
- sim4, mRNA:n genomiset rinnastuksetsim4)
- Vienna, RNA sekundäärirakenneanalyysi.

Käyttäjä voi halutessaan asentaa EMBOSS-paketin myös omalle koneelleen. Ohjelmisto on ilmainen ja sen voi hakea osoitteesta

```
http://emboss.sourceforge.net/
```

5.1 Sekvenssitiedostot

EMBOSS:illa analysoitavien sekvenssien on oltava jossain ohjelmiston ymmärtämässä tiedostomuodossa. Tiedostomuoto (esim. fasta, gcg tai embl) tarkoittaa sekvenssin esitystapaa tiedostossa. Keskeisimmät tiedostomuodot on esitelty liitteessä B. Kaikki EMBOSS:in ymmärtämät sekvenssimuodot ovat paljaita tekstitiedostoja, joissa on vain kirjoitettua tekstiä eikä esimerkiksi kirjasin- tai tyylimäärittelyjä. Täten esimerkiksi Microsoft Word-muodossa talletettu tiedosto ei toimi, vaan tallennus on tehtävä muodossa *Text only*. Useimmat sekvenssieditorit (esimerkiksi seaview tai mse) samoin kuin unixin tekstieditorit (esimerkiksi Nano ja Emacs) tallentavat tiedostot automaattisesti tekstitiedostoina.

Yleisimmin käytetty sekvenssitiedostotyyppi on fasta-muotoinen sekvenssitiedosto. Alla on esimerkki fasta-muotoisesta sekvenssitiedostosta.

```
>Sekvenssi1 Esimerkkisekvenssi fasta-muodossa
ATGGCCAGAATCATGATCAACAACGCCCTGGCCATCAACGAG
CCCCAGAATCATGATAGAATCACCAGAATCATGATTGAT
```

Tiedoston alussa on merkillä > alkava kommenttirivi, jota ei tulkita sekvenssiksi. EMBOSS ja monet muut ohjelmistot olettavat, että fasta-tiedoston kommenttirivin ensimmäinen sana on sekvenssin nimi tai tunniste, joten kommenttirivin alkuun kannattaa kirjoittaa jokin sopiva tunnistetermi.

Kommenttirivin jälkeen tulee varsinainen sekvenssidata. Sekvenssi voidaan kirjoittaa yhdelle tai useammalle riville. Rivien pituudella ei ole merkitystä. Kaikki kommenttirivin jälkeiset merkit tulkitaan kuuluvaksi samaan sekvenssiin. Fasta-muotoinen sekvenssitiedosto voi sisältää useita kommenttiriveillä erotettuja sekvenssejä.

```
>Sekvenssi1 Esimerkkisekvenssi fasta-muodossa
ATGGCCAGAATCATGATCAACAACGCCCTGGCCATCAACGAG
CCCCAGAATCATGATAGAATCACCAGAATCATGATTGAT
>Sekvenssi2 toinen sekvenssi
CCAGAATCATGATCAACAACGCCCTGGCCATCAACGAG
CCCAACAGAATCATGATAGAATCACCAGAATCATGATTGAT
```

5.2 USA-merkintätapa

Käyttäjä voi kertoa EMBOSS-ohjelmalle halutun sekvenssin sijainnin ja tiedostomuodon ns. USA-merkintätavan (Uniform Sequence Address) avulla.

USA voi viitata paitsi sekvenssitiedostoon myös suoraan tietokannassa olevaan sekvenssiin. USA:n perusrakenne on:

```
sekvenssimuoto::sijainti:nimi
```

Huomaa että sekvenssimuodon jäljessä on kaksi kaksoispistettä ja sijainnin (tietokannan tai tiedoston nimen) jäljessä vain yksi.

Sekvenssimuoto

EMBOSS tunnistaa yleensä tiedoston sekvenssimuodon automaattisesti, joten muotoa ei yleensä tarvitse sekvenssiä luettaessa määritellä. Muotomäärittelyä käytetäänkin lähinnä silloin, kun tulossekvenssi halutaan jossain oletusarvosta poikkeavassa muodossa. EMBOSS pystyy käyttämään lukuisia sekvenssimuotoja (mm. genbank, nbrf, embl, gcg, dnastrider, fitch, fasta, phylip, pir, msf, asn.1, paup ja clustalw). Kattava lista EMBOSS:in käyttämistä tiedostomuodoista löytyy seqret-ohjelman ohjesivulta.

Sijainti

Sijainti voi viitata tiedostoon tai EMBOSS:ssa määritettyihin tietokantoihin, jotka saa näkyviin showdb-ohjelmalla. Coronassa käytettävissä olevia tietokantoja ovat EMBL, IMGT, UniProt/Swiss-Prot, UniProt/TrEMBL ja PDB.

Nimi

Nimi voi olla sekvenssin nimi tai tunnusnumero tietokannassa tai monen sekvenssin tiedostossa. Verkon kautta käytettävien tietokantojen tapauksessa nimi pitää tietää täsmällisesti, mutta Coronassa olevien tietokantojen sekvensseihin voidaan viitata pelkän nimen tai tunnusnumeron osan perusteella käyttäen merkkiä *. Merkki * tarkoittaa USA-määrittelyssä mitä tahansa merkkijonoa.

Esimerkiksi USA määrittely

```
embl::swiss:myg_human
```

viittaa ihmisen myoglobiinisekvenssiin (ID-koodi myg_human) joka on embl-muodossa UniProt/Swiss-Prot-tietokannassa. Myös määrittely `swiss:myg_human` riittäisi, sillä EMBOSS tunnistaa embl-muodon automaattisesti. Swiss-Prot-tietokannan kaikkiin myg-alkuisiin sekvensseihin voitaisiin viitata merkinnällä:

```
swiss:myg*
```

Kaikkiin tietokannassa oleviin sekvensseihin voidaan viitata antamalla pelkkä tietokannan nimi ja kaksoispiste. Esimerkiksi:

```
swiss:
```

Mikäli tulossekvenssi haluttaisiin sekvenssi1-nimisenä tiedostona pir-muodossa, olisi USA määrittely:

```
pir::sekvenssi1
```

Myöhemmissä esimerkeissä käytetään myös harvinaisempaa asis-sijaintimäärettä, jonka avulla sekvenssi voidaan syöttää suoraan komentorivillä:

```
asis:AATCACCATGATCAACAACGCCCTGGCCATCAACGAG
```

5.3 EMBOSS:in käyttöliittymät

Corona-koneelle asennettua EMBOSS-ohjelmistoa voidaan käyttää kolmella eri tavalla:

1. www-käyttö Tutkijan käyttöliittymän avulla
2. Komentorivikäyttö pääteohjelman (esim. SSH) avulla
3. Jemboss-käyttö X-pääteohjelman avulla

Tutkijan käyttöliittymä on kätevin tapa yksittäisten analyysien tekoon. Syöte voidaan antaa käyttäjän omalta koneelta ja tulostiedosto voidaan myös tallentaa suoraan omalle koneelle. Kaikki EMBOSS-ohjelmat eivät kuitenkaan ole käytetävissä Tutkijan käyttöliittymässä.

Komentorivikäyttö Unix-ympäristössä tarjoaa www:tä laajemman ohjelmavalikoiman sekä enemmän mahdollisuuksia ohjelmaparametrien säätöön. Se mahdollistaa rutiiniluonteisten analyysien automatisoinnin sekä sekvenssilistojen käytön. Komentorivillä EMBOSS:ia voi myös käyttää tehokkaasti muiden analyysiohjelmien (esim. BLAST, ClustalX ja PHYLIP) kanssa.

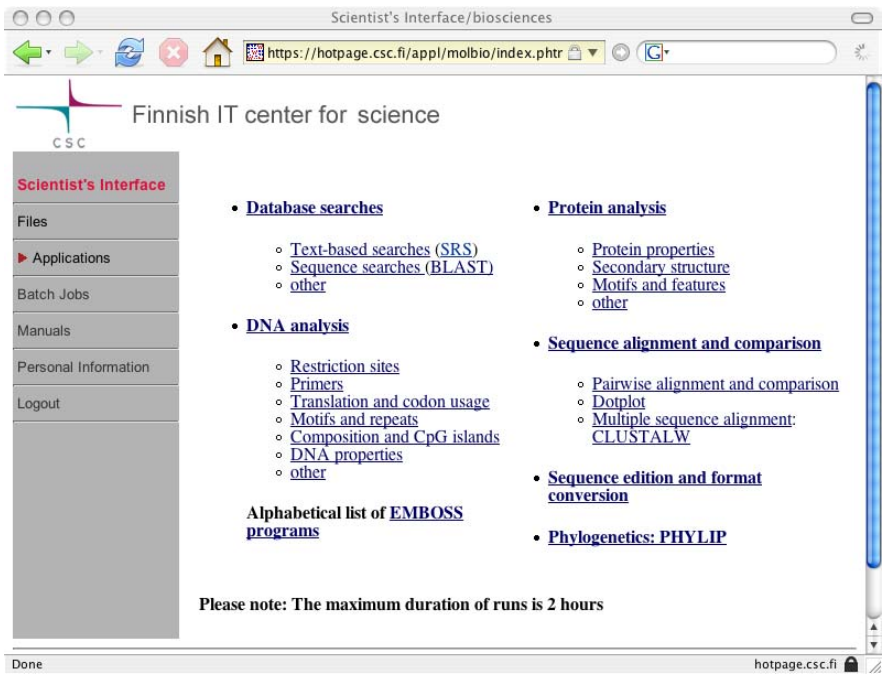
Graafinen Jemboss-käyttöliittymä tarjoaa komentorivikäytön monipuolisuuden ja se sisältää myös DNA-editorin ja Jalview-sekvenssirinnastuseditorin. Jembossin käyttö vaatii toimivan X-pääteyhteyden käyttäjän koneen ja CSC:n välillä.

5.4 Www-käyttö Tutkijan käyttöliittymässä

Tutkijan käyttöliittymä on www-palvelu, jonka avulla voi käyttää CSC:llä olevia ohjelmia ja siirtää tiedostoja oman koneen ja CSC:n välillä. Tutkijan käyttöliittymästä kerrotaan tarkemmin kappaleessa 2. Biotieteiden ohjelmistovalikoima sisältää yli 150 analyysiohjelmaa (kuva 5.1). Sopivan työkalun löytämiseksi ohjelmat on ryhmitelty aihealueittain ja sen mukaan, analysoidaanko DNA- vai proteiinisekvenssiä. Molemmille sekvenssityypeille yhteiset menetelmät kuten rinnastus ja editointi ovat omina ryhminään. Ryhmä-

kohtaisilla sivuilla on lista aiheeseen kuuluvista ohjelmista lyhyine kuvauksineen ja manuaalilinkkeineen. EMBOSS-ohjelman voi myös etsiä nimen perusteella aakkosellisesta listasta.

EMBOSS-ohjelmien lisäksi Tutkijan käyttöliittymän kautta voi tehdä mm. BLAST-hakuja, monen sekvenssin rinnastuksia ClustalW:llä, fylogenteiikkaa PHYLIP-ohjelmilla ja geenikartoitusta Merlin-ohjelmalla. Navigoinnin helpottamiseksi sivulta on linkki myös CSC:n SRS-palvelimelle.

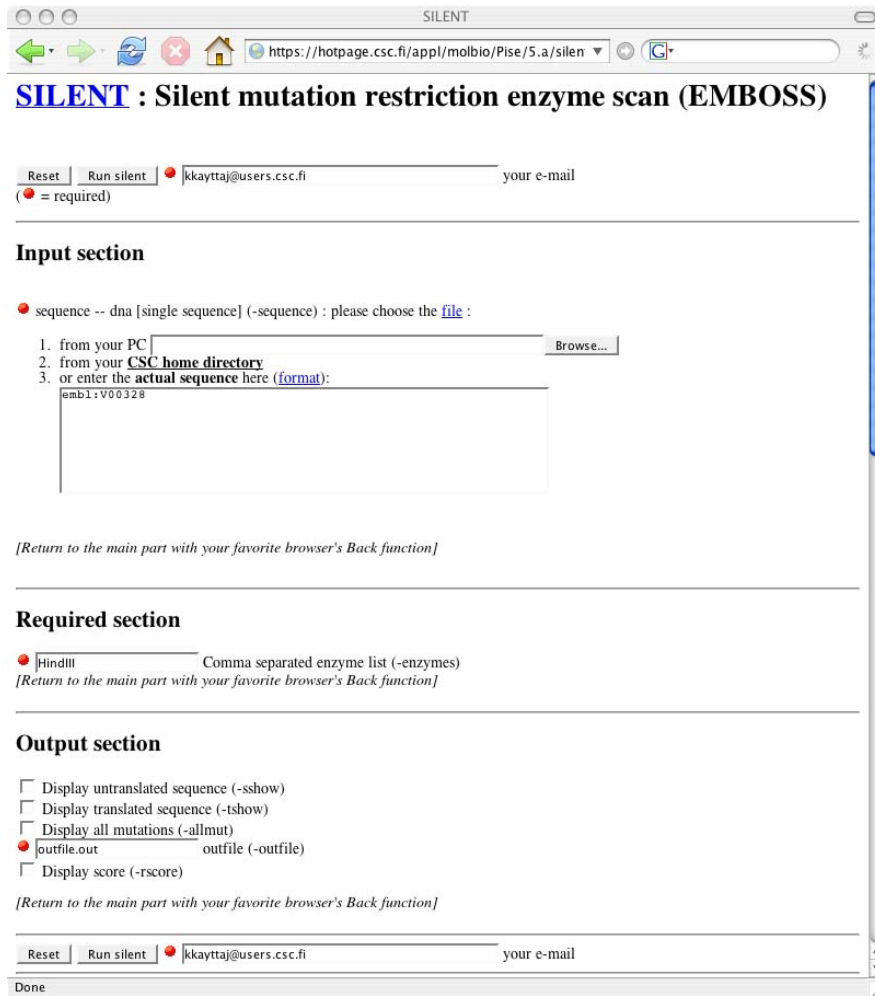


Kuva 5.1: *Tutkijan käyttöliittymä sisältää yli 150 biotieteiden ohjelmaa. Biotiedesovellusten pääsivulla ohjelmat on ryhmitelty aihepiireittäin.*

5.4.1 Ohjelman suoritus ja tulosten tallentaminen

EMBOSS:in ohjelmasivu avautuu osoittamalla ohjelman nimeä ryhmäkohtaisella sivulla. Kaikkien EMBOSS-ohjelmasivujen rakenne on samanlainen (kuva 5.2).

Analysoitavan sekvenssin voi ladata omalta koneelta *Browse*-napin avulla tai käyttäjän omasta CSC-kotihakemistosta (linkki *CSC home directory*), jolloin tiedoston sisältö kopioituu alla olevaan tekstikenttään klikkaamalla tiedoston nimeä. Tekstikenttään voi myös kirjoittaa sekvenssin tai kopioida sen leikepöydältä. Kaikissa vaihtoehdoissa sekvenssin on oltava EMBOSS:in ymmärtämässä tiedostomuodossa. EMBOSS-ohjelmat osaavat hakea syötesekvenssin suoraan tietokannastakin, mikäli sekvenssin nimi tai hakukoodi (acces-

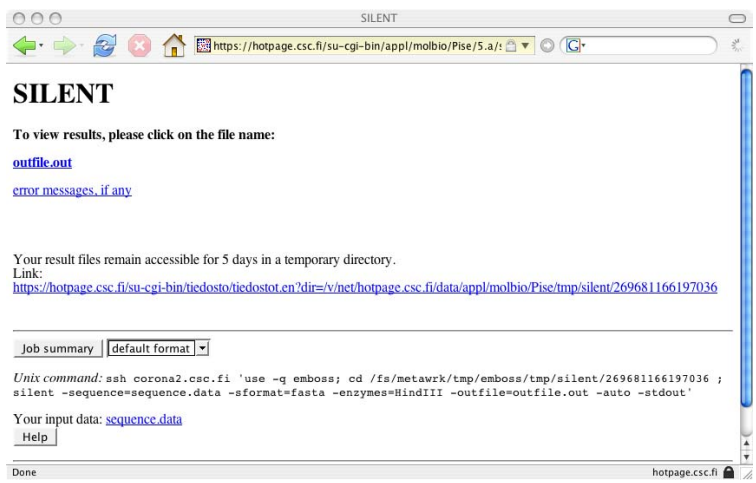


Kuva 5.2: *EMBOSS-ohjelman syötesekvenssit voidaan sijoittaa käyttöliittymään fasta-muodossa tai antaa tietokantaviittauksena USA-merkintätavan mukaisesti. Vaihtoehtoisesti sekvenssin voi myös ladata joko omalta koneelta tai CSC-kotihakemistosta.*

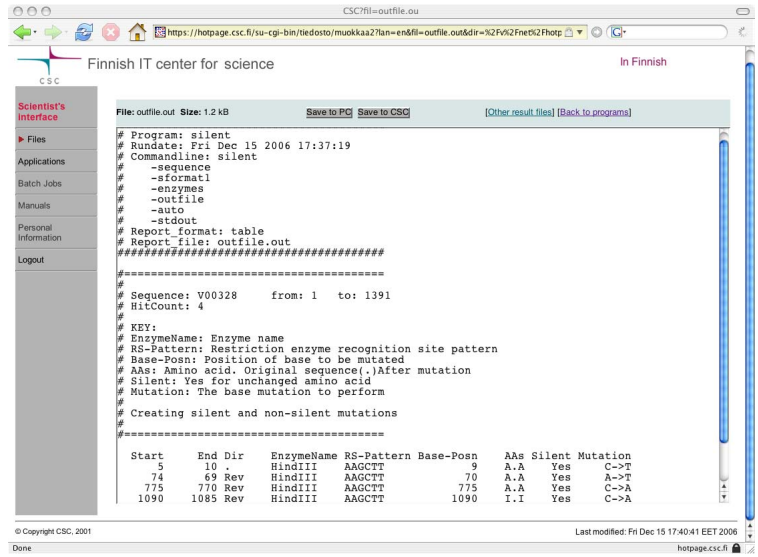
sion number) annetaan tekstikenttään USA-merkintätavan mukaisesti (esimerkiksi embl:V00328).

Ohjelman nimi sivun ylälaidassa on linkki manuaalisivuille, ja parametrien yhteydessä olevista linkeistä pääsee sivun lopussa olevaan pienempään ohjeistukseen. Sähköpostikenttään kirjautuu automaattisesti käyttäjän sähköpostiosoite CSC:llä, mutta osoitteen voi tarvittaessa muuttaa. Tulokset lähetetään sähköpostitse, jos ohjelman suorittaminen kestää yli minuutin. Ohjelma käynnistyy nappia *Run* painamalla.

Tulosten valmistuttua käyttäjä saa välisivun (Kuva 5.3), jossa on linkki tulostiedostoon sekä mahdollisiin virheilmoituksiin. Tulostiedostoon väliaikaishakemistossa, jossa se säilyy noin viiden päivän ajan. Tulosten tarkastelusivulla EMBOSS:in antaman tuloksen voi tallentaa joko omalle koneelle (*Save to PC*) tai käyttäjän omaan CSC-kotihakemistoon (*Save to CSC*). (Kuva 5.4.) Jotkut EMBOSS-ohjelmat tuottavat useampia tulostiedostoja, joihin pääsee linkistä *Other result files*.



Kuva 5.3: *EMBOSS-ohjelman välisivulla on linkki tulostiedostoon sekä mahdollisiin virheilmoituksiin.*



Kuva 5.4: *EMBOSS-ohjelman tulostiedoston voi tallentaa omalle koneelle tai käyttäjän omaan CSC-kotihakemistoon.*

5.5 EMBOSS-ohjelmiston komentorivikäyttö

5.5.1 Alustus, tietokannat ja ohjelman valinta

Komentorivillä tarkoitetaan tilaa, johon komennot annetaan sen jälkeen kun pääteyhteys palvelimeen (corona.csc.fi) on luotu. Ohjeet pääteyhteyden luomiseksi löytyvät tämän oppaan luvusta 3.1.1 Yhteyden luomisen jälkeen EMBOSS alustetaan Coronassa komennolla.

```
use emboss
```

Alustus tarvitsee tehdä vain kerran yhteyden avaamisen jälkeen. Alustuksen jälkeen unix-ympäristö tunnistaa EMBOSS-komennot ja käytössä olevat tietokannat. EMBOSS:in käyttämän tietokantavalikoiman saa näkyviin komennolla `showdb`.

```
corona1 ~> use emboss
[emboss is now in use]
corona1 ~> showdb
corona1 ~> showdb
Displays information on the currently available databases
# Name          Type  ID  Qry All Comment
# ============ ==== ==  === === =======
pdb_seq         P    OK  OK  OK  PDB database (sequences)
swiss           P    OK  OK  OK  Alias to UniProt/SwissProt
trembl          P    OK  OK  OK  UniProt/TrEMBL database
uniprot_swiss   P    OK  OK  OK  UniProt/SwissProt database
uniprot_trembl  P    OK  OK  OK  UniProt/TrEMBL database
embl            N    OK  OK  OK  Alias to EMBLRELEASE database
emblnew         N    OK  OK  OK  EMBLNEW database
```

```
emblrelease     N    OK  OK  OK  EMBLRELEASE database
emblwgs         N    OK  OK  OK  EMBL WGS database
imgt            N    OK  OK  -   IMGT via SRS
```

Taulukon ensimmäinen sarake (Name) kertoo, mikä on tietokannan nimi käytettäessä USA-määrittelyä. Joihinkin tietokantoihin voi viitata useammalla eri nimellä (esim. uniprot_swiss ja swiss).

Coronassa sopivaa EMBOSS-ohjelmaa voi etsiä ohjelmalla *wossname*. Wossname kysyy käyttäjältä sanan, jota se etsii ohjelmistojen kuvauksista. Alla olevassa esimerkissä etsitään EMBOSS-ohjelmia, jotka liittyvät restriktioanalyyseihin (restriction).

```
corona1 ~> wossname
Finds programs by keywords in their one-line documentation
Keyword to search for, or blank to list all programs:restriction
Finds programs by keywords in their one-line documentation
SEARCH FOR 'RESTRICTION'
frestboot        Bootstrapped restriction sites algorithm
frestdist        Distance matrix from restriction sites or fragments
frestml          Restriction site maximum Likelihood method
recoder          Remove restriction sites but maintain same translation
redata           Search REBASE for enzyme name, references, suppliers etc
remap            Display sequence with restriction sites, translation etc
restover         Find restriction enzymes producing specific overhang
restrict         Finds restriction enzyme cleavage sites
showseq          Display a sequence with features, translation etc
silent           Silent mutation restriction enzyme scan
```

Jos wossname-ohjelmalle ei anneta etsittävää sanaa, se tulostaa kaikkien EMBOSS-ohjelmien nimet ja lyhyet kuvaukset toiminnan mukaan järjestettynä.

Kun sopivalta vaikuttava ohjelma on löytynyt, sen tarkempaan kuvaukseen voi tutustua joko CSC:n www-sivuilla tai antamalla Coronassa komennon `tfm` *ohjelman_nimi*. Esimerkiksi restrict-ohjelman käyttöoppaan voi tulostaa ruudulle komennolla:

```
tfm restrict
```

5.5.2 Ohjelman suoritus

EMBOSS-ohjelma käynnistyy kirjoittamalla sen nimi komentoriville. Tulostettuaan näytölle lyhyen kuvauksen ohjelma kysyy syötetietoja, kuten analysoitavan sekvenssin USA-määrittelyn tai sekvenssitiedoston nimen, muut tarvittavat parametrit ja lopuksi tulostiedoston nimen. Mikäli ohjelmalla on kysytylle parametrille jokin oletusarvo, se esitetään kyselyn yhteydessä hakasulkujen sisällä.

Alla on esimerkki *seqret*-ohjelman komentorivikäytöstä. Seqretin avulla voidaan lukea haluttu sekvenssi tiedostosta tai tietokannasta ja tulostaa se uuteen tiedostoon. Seqretillä voi myös muuttaa sekvenssin tiedostomuotoa (esimerkiksi gcg-muodosta fasta-muotoon). Alla olevassa esimerkissä haetaan SwissProt-tietokannasta sekvenssi, jonka tunnusnumero on P12067.

```
corona1 ~> seqret
```

```
Reads and writes (returns) sequences
Input (gapped) sequence(s): swiss:P12067
output sequence(s) [lysc1_pig.fasta]:
corona1 ~>
```

Tässä tapauksessa ohjelman ehdottama tulostiedoston nimi (lysc1_pig.fasta) hyväksyttiin painamalla Return-näppäintä. Tuloksena saatiin fasta- muodossa oleva sekvenssitiedosto, jonka sisältö voidaan katsoa esimerkiksi `less`-komennolla.

```
corona1 ~> less lysc1_pig.fasta

>LYSC1_PIG P12067 Lysozyme C-1 (EC 3.2.1.17) (1,4-beta-N-acetylmuramidase C).
KVYDRCEFARILKKSGMDGYRGVSLANWVCLAKWESDFNTKAINRNVGSTDYGIFQINSR
YWCNDGKTPKAVNACHISCKVLLDDDLSQDIECAKRVVRDPQGIKAWVAWRTHCQNKDVS
QYIRGCKL
```

Mikäli tulos haluttaisiin esim. gcg-muodossa, se voidaan määritellä Output sequence-kohdassa käyttäen USA-merkintää:

```
corona1 ~> seqret
Reads and writes (returns) sequences
Input (gapped) sequence(s): swiss:P12067
output sequence(s) [lysc1_pig.fasta]:gcg::lysc1_pig.seq
```

Nyt tuloksena on gcg-muodossa oleva tiedosto nimeltä lysc1_pig.seq.

Jos SwissProt-tietokannasta haluttaisiin hakea kaikki lysc1-alkuiset sekvenssit, niihin voitaisiin viitata käyttäen merkintätapaa `swiss:lysc1*`.

```
corona1 ~>seqret
Reads and writes (returns) sequences
Input (gapped) sequence(s): swiss:lysc1*
output sequence(s) [lysc1_anapl.fasta]: lysc1.fasta
corona1 ~>
```

Tuloksena on nyt tiedosto lysc1.fasta, jossa on kahdeksan fasta-muotoista sekvenssiä. Voit tarkistaa tämän komennolla `infoseq lysc1.fasta`.

5.5.3 Tulostiedostot

Normaalisti EMBOSS-ohjelman tulostiedosto tulee siihen hakemistoon, jossa ohjelma on suoritettu. Tekstimuotoisten tulostiedostojen tarkastelu Coronassa onnistuu esim. less-komennolla:

```
less tiedoston_nimi
```

Mikäli tulostiedosto ei ole sekvenssitiedosto, sen alussa on tiedot siitä, millä ohjelmalla ja milloin tulostiedosto on luotu. Lisäksi tulostiedoston alkuun kirjautuu tietoa analyysissä käytetyistä syötteistä ja parametreistä.

Tulostuksen voi ohjata tiedoston sijaan suoraan näytölle antamalla tulostiedoston nimeksi *stdout*. Tämä tulostusvaihtoehto on käytännöllinen esim. ohjelmiin tutustuttaessa tai eri parametrien vaikutusta testattaessa.

5.5.4 Kuvatiedostot

Monet EMBOSS-ohjelmat tuottavat kuvatiedostoja. Jotta kuvia voi katsella Corona-koneesta pääteyhteyden kautta, täytyy oman koneen välillä olla toimiva x-pääteyhteys (katso kappale 3.1.2).

Kuvia tuottavat EMBOSS-ohjelmat kysyvät käyttäjältä kuvan esitystavan. Oletusarvo on X11, eli kuva avautuu X-yhteyden kautta päätekoneen näytölle. Kuva ei tallennu, vaan se häviää käyttäjän painaessa Return-näppäintä uudelleen. Tallennettavan kuvan voi tuottaa antamalla `Graph Type` -parametrin arvoksi `cps`,`ps` tai `png`, jotka tuottavat värillisen tai mustavalkoisen PostScript-kuvan tai PNG-tyyppisen kuvan. PostScript kuvia voit katsella Coronassa mm. `ggv`-ohjelmalla (kappale 3.5.3). PNG-kuvia voit tarkastella ohjelmalla `eog`. Mikäli käytössäsi ei ole X-pääteyhteyttä, voit käyttää Tutkijan käyttöliittymää kuvien tarkasteluun.

Muokkausta ja tulostusta varten kuvat kannattaa siirtää omalle koneelle esim. Tutkijan käyttöliittymän avulla. Alla on esimerkki PostScript-kuvan tuottamisesta avoimia lukukehyksiä etsivällä plotorf-ohjelmalla.

```
corona1 ~> plotorf
Plot potential open reading frames
Input nucleotide sequence: embl:V00328
Graph type [x11]: cps
Created plotorf.ps
corona1 ~> ggv plotorf.ps
```
(ggv ohjelmalla aukaistu tiedosto on esitetty kuvassa 5.5)

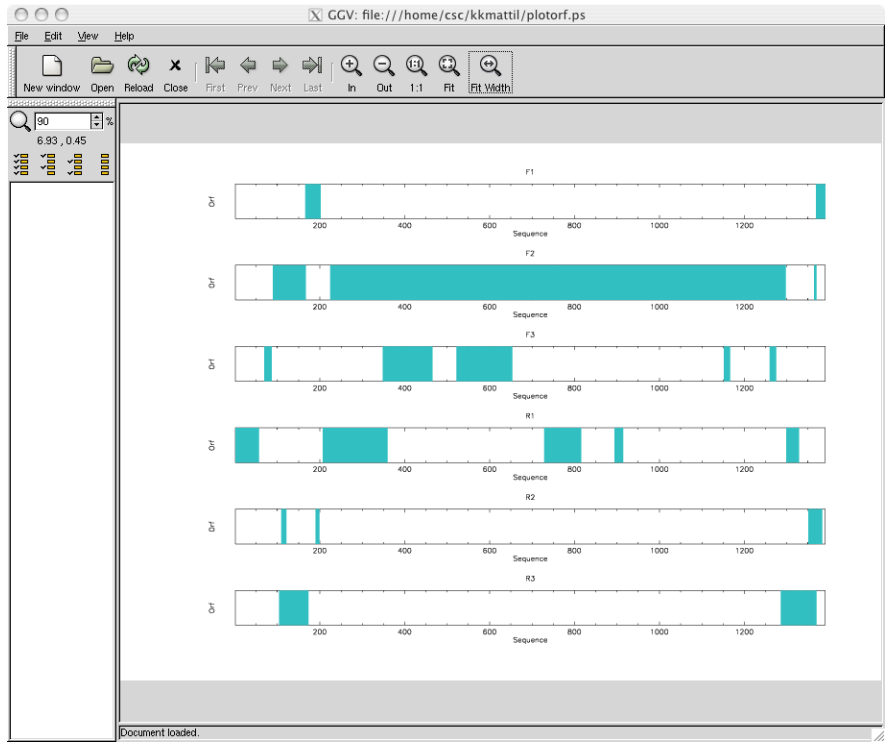
5.5.5 Syötteet komentorivillä

Edellä olleissa esimerkeissä ohjelmia ajettiin interaktiivisesti, jolloin ohjelma kysyi käyttäjältä tarvittavat tiedot. EMBOSS-ohjelmalle voi antaa komentorivillä valmiiksi tietoa syöttö- ja tulostustiedostojen nimistä sekä muista parametreista. Edellisen esimerkin kaltaisen `seqret`-komennon voisi suorittaa antamalla syöttö- ja/tai tulostiedostojen nimet komennon yhteydessä. Huomaa, että *-merkin sisältävä USA-määrittely on sijoitettava komentorivillä lainausmerkkeihin.

```
corona1 ~> seqret "swiss:lysc1*" lysc1.fasta
Reads and writes (returns) sequences
```

5.5.6 Ohjelmakohtaiset valitsimet

Jos EMBOSS-ohjelma käynnistetään ilman muita määrittelyjä, se kysyy vain keskeisimmät tiedot ja käyttää muissa tapauksissa oletusarvoja. Usein on kuitenkin tarpeen muuttaa ohjelman oletusarvoja tai tarkistaa, mitkä ne ovat. Oletusarvoja voidaan muuttaa ohjelmakohtaisilla valitsimilla (Optional qualifiers), jotka saa näkyviin valitsimella `-help`.



Kuva 5.5: *Tulostiedosto plotorf.ps aukaistuna ggv-ohjelmalla. Kuvan asemointi on korjattu valikon View asetuksella Orientation:seascape*

Alla olevassa esimerkissä käännetään lyhyt DNA-sekvenssi *transeq*-ohjelmalla aminohapposekvenssiksi. Mikäli transeq käynnistetään ilman muita määrittelyjä, se kääntää annetun nukleotidisekvenssin vain ensimmäisessä lukukehyksessä.

```
corona1 ~> transeq
Translate nucleic acid sequences
Input nucleotide sequence(s): asis::TGCAGCTGCTGCAGCTGCTGCAGCTGC
protein output sequence(s) [asis.pep]: csc.pep
corona1 ~> less csc.pep
>asis_1
CSCCSCCSC
```

Transeq-ohjelmaa voidaan tarvittaessa käyttää myös monipuolisemmin. Komento `transeq -help` tulostaa tiedot transeq-ohjelman valitsimista. (Alla olevaa transeq -help komennon tulostusta on karsittu taittoteknisistä syistä).

```
corona1 ~> transeq -help
   Standard (Mandatory) qualifiers:
  [-sequence]          seqall     Nucleotide sequence(s) filename and optional
                                  format, or reference (input USA)
  [-outseq]            seqoutall  [<sequence>.<format>] Protein sequence
                                  set(s) filename and optional format (output
```

```
                                   USA)

   Additional (Optional) qualifiers:
   -frame              menu       [1] Frame(s) to translate (Values: 1 (1); 2
                                  (2); 3 (3); F (Forward three frames); -1
                                  (-1); -2 (-2); -3 (-3); R (Reverse three
                                  frames); 6 (All six frames))
   -table              menu       [0] Code to use (Values: 0 (Standard); 1
                                  (Standard (with alternative initiation
                                  codons)); 2 (Vertebrate Mitochondrial); 3
                                  (Yeast Mitochondrial); 4 (Mold, Protozoan,
                                  Coelenterate Mitochondrial and
                                  Mycoplasma/Spiroplasma); 5 (Invertebrate
                                  Mitochondrial); 6 (Ciliate Macronuclear and
                                  Dasycladacean); 9 (Echinoderm
                                  Mitochondrial); 10 (Euplotid Nuclear); 11
                                  (Bacterial) (jatkuu...)

   -regions            range      [Whole sequence] Regions to translate.
                                  If this is left blank, then the complete
                                  sequence is translated.
                                  A set of regions is specified by a set of
                                  pairs of positions. (jatkuu...)

   -trim               boolean    [N] This removes all 'X' and '*' characters
                                  from the right end of the translation. The
                                  trimming process starts at the end and
                                  continues until the next character is not a
                                  'X' or a '*'
   -clean              boolean    [N] This changes all STOP codon positions
                                  from the '*' character to 'X' (an unknown
                                  residue). This is useful because some
                                  programs will not accept protein sequences
                                  with '*' characters in them.

   Advanced (Unprompted) qualifiers:
   -alternative        boolean    [N] The default definition of frame '-1' is
                                  the reverse-complement of the set of codons
                                  used in frame 1. (Frame -2 is the set of
                                  codons used by frame 2, similarly frames -3
                                  and 3). (jatkuu...)
   General qualifiers:
   -help               boolean    Report command line options. More
                                  information on associated and general
                                  qualifiers can be found with -help -verbose
```

(jatkuu)

Listassa on ensin ohjelman pakolliset parametrit (Standard qualifiers), tässä tapauksessa käännettävä nukleotidisekvenssi (-sequence) ja tulostiedoston nimi (-outseq). Seuraavaksi listassa näkyvät ohjelmakohtaiset lisävalitsimet (Additional qualifiers). Käyttämällä valitsinta -opt ohjelma ei automaattisesti käytä lisävalitsimille oletusarvoja, vaan kysyy niiden arvot ajon aikana. Esimerkiksi transeq-ohjelmassa kysytään käännettävän sekvenssin lisäksi käytettävä lukukehys, kodonitaulukko sekä käännettävä sekvenssialue. Lisäksi voidaan valita poistetaanko X- ja *- merkit tulossekvenssin lopusta tai muutetaanko *-merkit X-merkeiksi:

```
corona1 ~>transeq -opt
Translate nucleic acid sequences
Input nucleotide sequence(s): asis::TGCAGCTGCTGCAGCTGCTGCAGCTGC
Translation frames
         1 : 1
         2 : 2
```

```
          3 : 3
          F : Forward three frames
         -1 : -1
         -2 : -2
         -3 : -3
          R : Reverse three frames
          6 : All six frames
Frame(s) to translate [1]: 6
Genetic codes
          0 : Standard
          1 : Standard (with alternative initiation codons)
          2 : Vertebrate Mitochondrial
          3 : Yeast Mitochondrial
          4 : Mold, Protozoan, Coelenterate Mitochondrial ...
          5 : Invertebrate Mitochondrial
          6 : Ciliate Macronuclear and Dasycladacean
          9 : Echinoderm Mitochondrial
         10 : Euplotid Nuclear
         11 : Bacterial
         12 : Alternative Yeast Nuclear
         13 : Ascidian Mitochondrial
         14 : Flatworm Mitochondrial
         15 : Blepharisma Macronuclear
         16 : Chlorophycean Mitochondrial
         21 : Trematode Mitochondrial
         22 : Scenedesmus obliquus
         23 : Thraustochytrium Mitochondrial
Code to use [0]: 11
Regions to translate (eg: 4-57,78-94): 2-25
Trim trailing X's and *'s [N]:Y
Change all *'s to X's [N]:
protein output sequence(s) [asis.pep]: csc.pep
```

Esimerkissä syötesekvenssi transloitiin kaikissa kuudessa lukukehyksessä. Translointi tehtiin vain osalle syötettyä sekvenssiä ja siinä käytettiin bakteerien kodonitaulukkoa.

Osa tai kaikki valitsimet voidaan syöttää myös suoraan komentorivillä. Valitsimien järjestyksellä ei ole väliä. Valitsimet, joilla ohjelmalle välitetään jokin lukuarvo tai merkkijono, kirjoitetaan muotoon *-valitsin=arvo* tai *-valitsin arvo*.

Boolen-tyyppiset valitsimet, jotka määrittävät onko jokin toiminto käytössä vai ei, voidaan antaa muodossa: *-valitsin* (toiminto käytössä) tai *-novalitsin* (toiminto ei ole käytössä).

Edellä tehty translointi voitaisiin siis tehdä myös seuraavalla komennolla:

```
corona1 ~> transeq asis::TGCAGCTGCTGCAGCTGCTGCAGCTGC csc.pep \
 -frame 6 -table 11 -regions 2-25 -trim -noclean
```

Tulostiedoston csc.pep sisältö on kummassakin tapauksessa sama.

```
corona1 ~> less csc.pep
>asis_1
SCCSCC
>asis_2
AAAAA
```

```
>asis_3
LLQLL
>asis_4
AAAAAA
>asis_5
SSCSS
>asis_6
QQLQQL
```

5.5.7 Yleiset valitsimet

Ohjelmakohtaisten valitsinten lisäksi EMBOSS:issa on joukko yleisiä valitsimia, jotka ovat käytössä soveltuvin osin kaikissa EMBOSS-ohjelmissa. Näitä valitsimia ei mainita valitsimen `-help` tulostuksessa eikä niitä kysytä, vaikka ohjelma käynnistetään valitsimen `-opt` kanssa. Yleisvalitsinten käyttö ei ole välttämätöntä, mutta usein niillä voi nopeuttaa työskentelyä. Keskeisimmät yleisvalitsimet on esitetty taulukossa 5.1. Listan kaikista tietyn ohjelman käyttämistä valitsimista saa näkyviin valitsimilla `-help -verbose`, esimerkiksi

```
seqret -help -verbose
```

Alla on muutama esimerkki sekvenssien lukuun ja kirjoitukseen liittyvien valitsimien käytöstä.

Käytettäessä valitsinta `-sask` EMBOSS-ohjelma kysyy, mistä kohtaa syötettyä sekvenssiä lukeminen aloitetaan ja mihin se lopetetaan. Tätä valitsinta käyttäen voidaan esimerkiksi tietokannasta lukea vain haluttu osa sekvenssiä:

```
corona1 ~> seqret swiss:P12067 -sask
Reads and writes (returns) sequences
     Begin at position [start]: 21
       End at position [end]: 40
output sequence(s) [lysc1_pig.fasta]: lysc1_pig21-40.pep

corona1 ~> less lysc1_pig21-40.pep

>LYSC1_PIG P12067 Lysozyme C-1 (EC 3.2.1.17) (1,4-beta-N-acetylmuramidase C).
RGVSLANWVCLAKWESDFNT
```

Yllä ollut esimerkki voitaisiin tehdä myös komennolla:

```
seqret swiss:P12067 lysc1_pig21-40.pep -sbegin=21 -send=40
```

Sekvenssien tulostusta säätelevistä valitsimista `-ossingle` on usein käyttökelpoinen. Jos haluaisimme esimerkiksi hakea SwissProt-tietokannasta kaikki lysc1-alkuiset sekvenssit, mutta tallentaa ne yhden tiedoston sijasta erillisiksi tiedostoiksi, se onnistuisi komennolla:

```
corona1 ~>seqret "swiss:lysc1*" -ossingle -auto
corona1 ~>ls
```

Taulukko 5.1: *Esimerkkejä EMBOSS:in käyttämistä yleisvalitsimista.*

Valitsin	*Toiminta*
`-auto`	Käytetään oletusarvoja, ellei komentorivillä muuta määritetä
`-help`	Tulostaa tiedot ohjelmakohtaisista valitsimista
`-opt`	Kysyy ohjelmakohtaisten valitsinten arvot
`-sbegin`	Analyysin aloituskohta sekvenssissä
`-send`	Analyysin lopetuskohta sekvenssissä
`-sreverse`	Käytetään käänteistä sekvenssiä
`-sask`	Kysyy -sbegin, -send ja -sreverse ohjelmaa ajettaessa
`-sformat`	Luettavan tiedoston muoto
`-osformat`	Kirjoitettavan tiedoston muoto
`-osname`	Kirjoitettavan tiedoston nimi
`-ossingle`	Kirjoita tulostettavat sekvenssit erillisiin tiedostoihin
`-rformat`	Analyysin antaman raportin kirjoitusmuoto
`-filter`	Lue oletussyöte (stdin) ja kirjoita tulos oletustulosteeksi (stdout)

```
lysc1_anapl.fasta   lysc1_horse.fasta   lysc1_tacac.fasta
lysc1_bovin.fasta   lysc1_pig.fasta
lysc1_canfa.fasta   lysc1_rat.fasta
lysc1_caphi.fasta   lysc1_sheep.fasta
```

Edellisessä esimerkissä valitsin `-auto` käyttää oletusarvoja muille parametreille.

Monet EMBOSS-ohjelmat tunnistavat *-rformat* valitsimen, jolla ohjelman antaman tekstimuotoisen tulosraportin esitystapaa voidaan muuttaa. Esimerkiksi valitsimella `-rformat excel` tuloksena on taulukkomuotoinen tiedosto joka voidaan lukea sisään esimerkiksi Microsoft Excel -taulukkolaskentaohjelmaan. Toinen yleisesti käytetty raporttimuoto on EMBOSS-listatiedosto (`-rformat listfile`). Tarkemmat kuvaukset erilaisista tulosraporteista löytyvät EMBOSS-projetin sivuilta:

`http://emboss.sourceforge.net/docs/themes/ReportFormats.html`

5.5.8 Listatiedostot

USA-merkintätavan lisäksi EMBOSS pystyy käyttämään myös listatiedostoja, joilla käyttäjä viittaa haluamiinsa sekvensseihin. Listat ovat käytännöllisiä käytettäessä samaa sekvenssijoukkoa useaan analyysiin tai haluttaessa määritellä tietokannasta vain tietty osajoukko. Rakenteeltaan EMBOSSin listatiedosto on tekstitiedosto, joka sisältää joukon USA-muotoisia määrittelyjä.

Listatiedoston voi luoda tekstieditorilla tai esimerkiksi SRS:n avulla. Joissakin tapauksissa myös EMBOSS-ohjelman tulosteen voi ohjata listatiedostoksi.

Alla on esimerkki listatiedostosta `sekvenssit.lista`, joka viittaa joukkoon EMBL-tietokannassa olevia sekvenssejä.

```
embl:AB044571
embl:AB062904
embl:AB072269
embl:AB072311
```

Kun listatiedostoa käytetään EMBOSS-ohjelman syötteenä, täytyy tiedoston nimen edessä olla @-merkki. Alla olevassa esimerkissä tarkistetaan tiedoston `sekvenssit.lista` sekvenssien pituudet ja GC-pitoisuudet *infoseq*-ohjelmalla.

```
corona1 ~> infoseq @sekvenssit.lista

Displays some simple information about sequences
USA                 Name        Accession    Type Length %GC    Descript
embl-id:AB044571    AB044571    AB044571     N    663    52.49  Mus ...
embl-id:AB062904    AB062904    AB062904     N    831    41.28  Mus ...
embl-id:AB072269    AB072269    AB072269     N    4065   51.05  Mus ...
embl-id:AB072311    AB072311    AB072311     N    1106   63.02  Mus ...
```

Listatiedostossa olevaa USA-määrittelyä voidaan täydentää sekvenssin lukemisen aloitus- ja lopetuskohdan määrittelyllä [*alku*:*loppu*]. Nukleotidisekvenssille voidaan määritellä käytettäväksi myös käänteinen sekvenssi termillä *:r*. Esimerkiksi alla olevan listatiedoston avulla voidaan lukea tiedostoista sekvenssi1.fasta alue 4-60, tiedostosta sekvenssi2.fasta alue 33-74 ja tiedostosta sekvenssi3.fasta alueelle 27-44 käänteinen sekvenssi.

```
sekvenssi1.fasta[4:60]
sekvenssi2.fasta[33:74]
sekvenssi3.fasta[27:44:r]
```

Listatiedostossa voidaan käyttää sekaisin erilaisia USA-määrittelyjä. Alla olevassa esimerkissä ensimmäinen rivi viittaa suoritushakemistossa olevaan tiedostoon sekvenssit.fasta. Toinen rivi viittaa gcg-muotoiseen sekvessi2.seq-tiedostoon, kolmas rivi UniProt/TrEMBL-tietokannassa olevaan sekvenssiin q12334 ja neljäs rivi SwissProt-tietokannan prio-alkuisiin sekvensseihin. Viimeistä edellinen rivi alkaa # merkillä, mikä tarkoittaa kommenttiriviä, jota ei lueta. Viimeinen rivi viittaa toiseen listatiedostoon.

```
sekvenssi.fasta
gcg::sekvessi2.seq
trembl:q12334
swiss:prio*
# Tuloslista edellisestä analyysistä
@toinen_lista
```

5.5.9 EMBOSS-ohjelman parametritiedostot

Useissa ohjelmissa käyttäjä valitsee jonkin parametrin EMBOSS-ohjelmiston mukana tulevien valmiiden parametritiedostojen joukosta. Tällaisia parametrejä ovat mm. kahden sekvenssin rinnastuksessa käytettävät pisteytysmatriisit, restriktioentsyymien nimet ja kodonitaulukot. Tutkijan käyttöliittymässä valittavissa olevat vaihtoehdot näkyvät suoraan käyttöliittymässä. Komentorivi- ja Jemboss-käyttöliittymissä käyttäjän on etukäteen tiedettävä haluamansa parametritiedoston nimi, mikäli hän haluaa käyttää muuta kuin oletusarvoa.

Listan EMBOSSin mukana tulevista parametritiedostoista saa näkyviin komennolla:

```
embossdata -showall
```

tai

```
embossdata -showall | less
```

Jos haluat tarkastella parametritiedostoja tarkemmin, siirry EMBOSSin datahakemistoon komennolla

```
cd /p/appl/molbio/emboss/latest/share/EMBOSS/data
```

Hakemiston sisältöä voi tarkastella ls ja less komennoilla. Jos haluat muokata oman versiosi jostakin parametritiedostosta, kopioi se ensin kotihakemistoosi ja muokkaa kopioitua versiota.

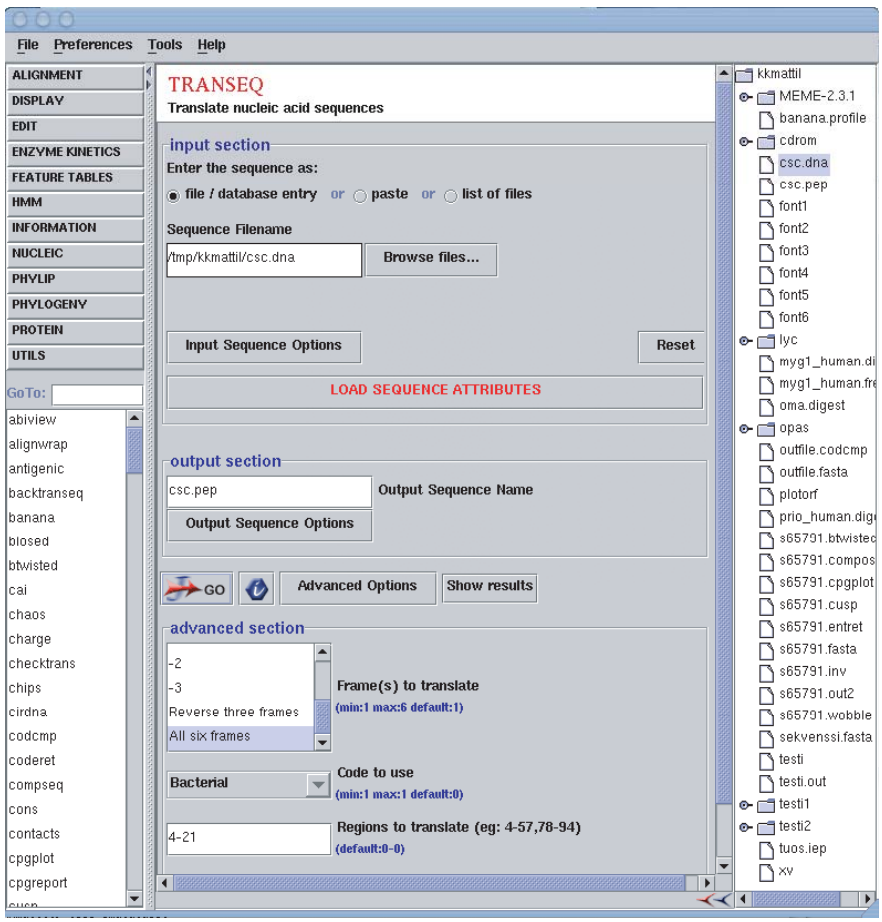
5.6 Jemboss

Mikäli X-pääteyhteyden luominen oman koneen ja Coronan välille onnistuu, EMBOSS- ohjelmia voi käyttää graafisen Jemboss-käyttöliittymän kautta. Kun X-pääteyhteys Coronaan on aukaistu (katso kappale 3.1.2) ja EMBOSS alustettu komennolla `use emboss`, Jemboss käynnistyy komennolla

```
jemboss
```

5.6.1 Tiedostojen hallinta

Käyttöliittymän oikeassa alareunassa olevaa kaksoisnuolta klikkaamalla avautuu näkyviin kotihakemiston tiedostot. Mikäli halutaan käyttää jotakin muuta hakemistoa, se voidaan määritellä valikon *Preferences/Advanced options* avulla. Tiedostoja voi avata tarkasteltavaksi ja muokattavaksi kaksoisklikkaamalla niiden nimeä.



Kuva 5.6: *Transeq-ohjelma Jemboss-käyttöliittymässä*

5.6.2 Ohjelman suoritus

Jemboss-käyttöliittymässä ohjelmat valitaan joko toiminnan mukaan ryhmitellyistä valikoista (vasemman reunan yläosa) tai aakkosellisesta ohjelmistolistasta (vasemman reunan alaosa). Ohjelman nimen voi myös suoraan kirjoittaa *Go To* -kohtaan.

Kun EMBOSS-ohjelma on valittu, sen parametrivalikot tulevat näkyviin käyttöliittymän keskiosaan. Kohdassa *Input selection* valitaan ohjelman syötetiedot. Syötesekvenssi voidaan valita *Browse*-napin avulla tai vetämällä tiedostolistassa oleva sekvenssitiedosto hiirellä kenttään *Sequence Filename*. Kohtaan *Sequence Filename* voidaan myös kirjoittaa USA-määrittely. Muuttamalla asetus *Enter the sequence as* arvoon *paste* korvautuu kenttä *Sequence Filename* kentällä *Sequence Cut and Paste*, johon sekvenssidata voidaan kopioida suoraan oman koneen leikepöydältä.

Käyttöliittymässä on oletusarvoisesti esillä valitsimet vain pakollisille parametreille. Mahdolliset erityisvalitsimet saa näkyviin sivun alaosaan klikkaamalla alhaalla olevaa nappia *Advanced Options*.

Kun kaikki parametrit on annettu, painetaan *GO*-nappia, jolloin ohjelman suoritus käynnistyy. Tulosten valmistuttua ilmestyy näkyviin nappi *Show Results*, jolla tulostiedoston voi aukaista tutkittavaksi. Tulostiedostot tulevat myös näkyviin oikean reunan tiedostolistaan. Jembossia käytettäessä tulokset siis tallentuvat Corona-koneen hakemistoon aivan kuten komentorivikäytössäkin.

Tavallisten EMBOSS-komentojen lisäksi voidaan Jemboss ohjelman Tools-valikosta Jalview- tai Jemboss-sekvenssirinnastuseditori tai Jemboss DNA editori jolla voidaan luoda kaaviokuvia sirkulaarisista tai lineaarisista DNA-sekvensseistä.

5.7 Lisätietoa EMBOSS ohjelmistosta

CSC:llä EMBOSS-ohjelmiston pääsivu löytyy osoitteesta:

```
http://www.csc.fi/english/research/sciences/bioscience/programs/emboss/
```

Tällä sivulla ovat tiedot mm. ohjelman versiosta sekä ohjeet eri käyttöliittymien käynnistämiseksi. Sivulta löytyvät myös linkit EMBOSSin ohjelmakohtaiseen dokumentaatioon. Ohjelmien käyttöoppaat on järjestetty sekä aakkoselliseksi listaksi että toiminnan mukaan jaotelluksi listaksi.

```
https://extras.csc.fi/emboss/doc/programs/html/index.html
https://extras.csc.fi/emboss/doc/programs/html/groups.html
```

Tällä sivulla on myös erillinen linkki EMBASSY-ohjelmien opassivuille.

```
https://extras.csc.fi/emboss/doc/programs/embassy/index.html
```

EMBOSS-projektin kotisivu löytyy osoitteesta:

```
http://emboss.sourceforge.net/
```

Tällä sivulla on runsaasti tietoa EMBOSS:ista. Sivulta voi myös ladata EMBOSS- ohjelmiston asennuksessa tarvittavat tiedostot. Lisäksi sivulta voi hakea mm. EMBOSS- alkeisoppaan.

EMBOSS-sivuston Themes-osuudesta löytyy kattavat kuvaukset mm.USA-formaatista sekä raportti- ja rinnstusformaateista, joita tässä oppaassa käsitellään vain lyhyesti.

6 Kahden sekvenssin vertailu ja rinnastus

6.1 Johdanto

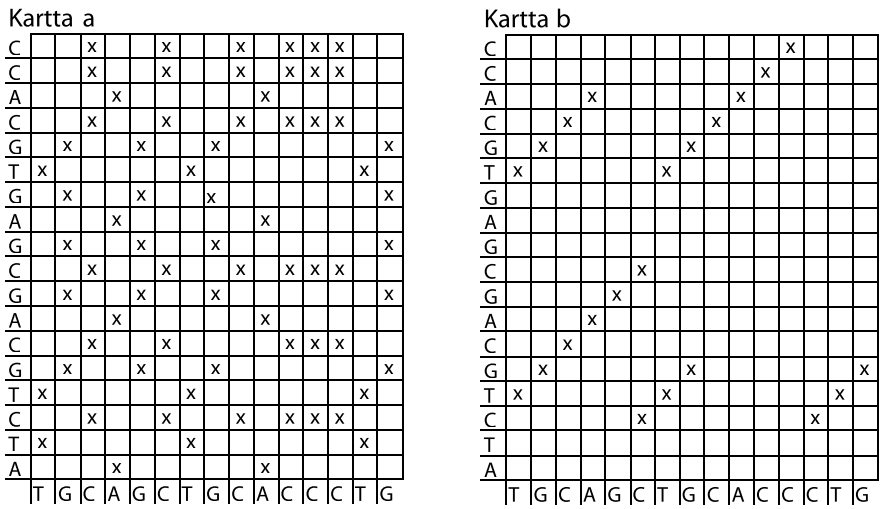
Kahden sekvenssin vertailu on eräs bioinformatiikan keskeisimmistä analyysityypeistä. Sekvenssejä voidaan vertailla monin eri tavoin. Joissakin tapauksissa halutaan löytää vain yhden emäksen eroja miltei identtisten sekvenssien välillä, kun taas joissakin tapauksissa halutaan tutkia, onko kahdella sekvenssillä mitään yhteisiä piirteitä. Monet bioinformatiikan menetelmät, kuten monen sekvenssin rinnastus ja samankaltaisuushaut tietokannoista, perustuvat kahden sekvenssin vertailuun.

Tässä kappaleessa tutustutaan lyhyesti kahden sekvenssin vertailumenetelmiin ja niiden käyttöön EMBOSS:issa. Esimerkit on tehty käyttäen EMBOSS:ia komentorivillä, mutta samat analyysit voitaisiin suorittaa myös Tutkijan käyttöliittymässä tai Jemboss-ohjelmassa. Esitellyt ohjelmat soveltuvat sekä nukleotidi- että proteiinisekvensseille.

6.2 Pistekartat (dotplot)

Pistekartat (dotplot) ovat yksikertainen, mutta hyvin havainnollinen tapa verrata kahta sekvenssiä keskenään. Yksinkertaisimmillaan pistekartta on taulukko, jossa ensimmäinen sekvenssi on taulukon vaaka-akselilla ja toinen pystyakselilla. Mikäli jossakin taulukon ruudussa vaaka- ja pystyakselilla on sama kirjain (emäs- tai aminohappokoodi), laitetaan kyseiseen ruutuun merkki. Tuloksena saadussa kuvaajassa samankaltaiset sekvenssikohdat näkyvät lävistäjän suuntaisina viivoina. Yleensä näin piirretty pistekartta tuottaa kuvaajan, jossa on hyvin paljon merkittyjä kohtia. Vertaamalla sekvenssejä toisiinsa tietynmittaisina paloina saadaan merkittävästi samankaltaiset alueet usein erottumaan satunnaisesta samanlaisuudesta paremmin.

Dottup-ohjelma toimii näin. Se lukee kaksi sekvenssiä ja piirtää pistekartan vertailemalla sekvenssejä määrätyn pituisina ”sanoina” (wordsize). Taulukon ruutuun tulee merkintä vain jos vaaka- ja pystyakselilla olevat ”sanat” ovat samat. Pienillä sanapituuksilla dottup-kuvaajaan tulee runsaasti sattu-



Kuva 6.1: *Kahden sekvenssin vertaaminen pistekartoilla. Kartta a. Taulukossa on merkitty kohdat, joissa pysty- ja vaakarivillä on sama emäs. Kartta b. taulukkoon on merkitty vain ne kohdat, joissa pystyrivin ja vaakarivin kolme perättäistä emästä ovat samat.*

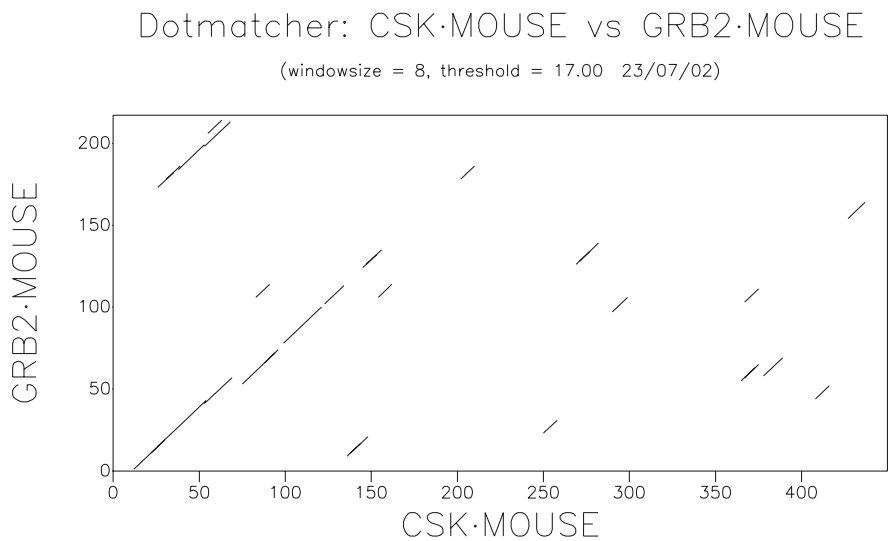
malta löytyneitä samankaltaisuuksia, kun taas suurilla sanapituuksilla osa merkittävistä samankaltaisuuden alueista saattaa jäädä löytymättä. Dottup-ohjelmaa kannattaakin käyttää useilla eri sanapituuden arvoilla, jolloin satunnaisten ja merkittävästi samankaltaisten sekvenssikohtien erottelu on helpompaa. Yleensä proteiineille kannattaa käyttää lyhyempiä (3-20) sanapituuksia ja nukleotideille pidempiä (10-50).

Dotmatcher on dottupia monipuolisempi ohjelma pistekarttojen laskemiseen. Siinäkin sekvenssejä verrataan tietyn mittaisina sanoina, mutta ohjelma ei tarkastele pelkästään sanojen identtisyyttä, vaan laskee pisteytysmatriisin (Matrix File) avulla verrattaville sanoille niiden samankaltaisuutta kuvaavan arvon (katso kappale 6.3.2.). Mikäli arvo ylittää raja-arvon (threshold), piirretään kyseiseen kohtaan viiva. Koska dotmatcher ei tarkastele vain identtisyyksiä, se pystyy paremmin havaitsemaan heikommin samankaltaisia sekvenssialueita.

Alla olevassa esimerkissä dotmatcherilla on verrattu hiiren Gbr2-proteiinia (grb2_mouse) hiiren csk-proteiiniin (csk_mouse). Csk-proteiini sisältää yhden SH3- ja yhden SH2-domeenin. Myös Grb2-proteiini sisältää yhden SH2-domeenin ja kaksi SH3-domeenia muistuttavaa aluetta. Sopivilla parametreillä tehty pistekartta antaa nopeasti yleiskuvan siitä, missä kohdin kahden sekvenssin välillä on samankaltaisuutta. Tässä tapauksessa sanapituutta ja kynnysarvoa oli muutettava, jotta samankaltaiset alueet saatiin näkymään selvästi.

```
corona1 ~>dotmatcher swiss:csk_mouse swiss:grb2_mouse -opt
Displays a thresholded dotplot of two sequences
```

```
Matrix file [EBLOSUM62]:
Window size over which to test threshhold [10]: 8
Threshold [23]: 17
Graph type [x11]:
```



Kuva 6.2: *Dotmatcher-ajon tulos. Ajossa verrattiin hiiren Grb2-proteiinia (grb2_mouse) hiiren csk-proteiiniin. Csk-sekvenssissä SH3-domeeni sijaitsee välillä 9-70 ja SH2-domeeni välillä 82-171. Grb2-sekvenssissä on kaksi SH3-domeenin tyyppistä aluetta väleillä: 1-58 ja 156-215 sekä yksi SH2-tyyppinen alue välillä 60-152. Samankaltaiset alueet näkyvät kuvassa kolmena lävistäjän suuntaisena viivaryhmänä pistekartan vasemmalla puoliskolla*

Pistekarttojen etu sekvenssirinnastuksiin verrattuna on se että ne vertaavat sekvenssien kaikkia kohtia toisiinsa eivätkä poimi esille vain yksittäisiä samankaltaisia alueita. Esim. sekvensseissä toistuvat alueet ja kaukana toisistaan olevat, mutta silti toisiinsa liittyvät samankaltaiset alueet on helpompaa havaita pistekartoissa kuin tavallisissa sekvenssirinnastuksissa. Pistekarttoja voidaan käyttää myös yhden sekvenssin analyysiin, sillä sekvenssissä olevat samankaltaiset alueet, kuten toistojaksot tai samantyyppiset domeenit, näkyvät pistekartassa.

Taulukko 6.1: *Pistekarttojen luomiseen käytettäviä EMBOSS-ohjelmia.*

Ohjelma	*Toiminta*
`dotmatcher`	Laskee pistekartan käyttäen samankaltaisuusvertailussa sanapituutta ja kynnysarvoa.
`dotpath`	Laskee pistekartan, jossa ei ole sekvenssien sisällä päällekäin meneviä diagonaalin suuntaisia viivoja.
`dottup`	Laskee pistekartan käyttäen samankaltaisuusvertailussa tietyn mittaisten sanojen identtisyyttä.
`polydot`	Lukee sisään joukon sekvenssejä ja tekee kaikille sekvenssipareille pistekartan.

6.3 Sekvenssirinnastukset

6.3.1 Johdanto

Yleisin tapa kuvata kahden sekvenssin samankaltaisuutta on sekvenssirinnastus (sequence alignment). Yksinkertaisimmillaan sekvenssirinnastus on taulukko, jossa verrattavat sekvenssit ovat eri riveillä aseteltuna siten, että toisiaan vastaavat sekvenssikohdat ovat samoilla sarakkeilla.

```
RINNASTASEKVENSSI
   KINNAS--SELVEN-SI
```

Hyvän rinnastuksen avulla saadaan tietoa sekvenssien samankaltaisuudesta, samankaltaisten osien sijainnista ja eroavuuksien laadusta. Esimerkiksi tuntemattoman sekvenssin funktiota voidaan arvioida rinnastamalla se toiminnaltaan tunnettujen sekvenssien kanssa. Usein sekvenssien rinnastaminen on vain ensimmäinen vaihe kahden tai useamman sekvenssin analysoinnissa.

Lyhyiden tai hyvin samankaltaisten sekvenssien rinnastus on mahdollista tehdä käsin sekvenssi- tai tekstieditorilla, mutta jos sekvenssit poikkeavat toisistaan enemmän kuin 5%, alkaa lyhyidenkin sekvenssien rinnastaminen kokonaan käsin olla hankalaa. Sekvenssien vertailuun onkin kehitetty useita täysin automaattisia rinnastusohjelmia. Vaikka nämä ohjelmat löytävät periaatteessa parhaan rinnastuksen, niiden tulokset ovat vain yksinkertaisen matemaattisen mallin mukaisia parhaita rinnastuksia. Monissa tapauksissa tutkija voi muun biologisen tiedon avulla muokata rinnastusta paremmaksi. On myös syytä muistaa, että rinnastukset pystyvät kuvaamaan sekvenssien välisiä biologisia suhteita ja samankaltaisuuksia vain vajavaisesti.

6.3.2 Pisteytysmatriisit

Jotta sekvenssejä voitaisiin verrata toisiinsa tietokoneohjelmalla, täytyy sekvenssien samankaltaisuutta pystyä mittaamaan numeerisesti. Yksinkertaisin

tapa mitata rinnastuksen laatua on laskea kohdakkain osuvien identtisten sekvenssikohtien määrä. Tätä tapaa käytetään joskus nukleotidisekvenssien vertailussa. Proteiineille tämä vertailutapa ei toimi hyvin, sillä proteiinien toiminnan kannalta eri aminohappojen muutosten merkittävyys tai yleisyys sukulaisproteiinien välillä vaihtelee suuresti. Esimerkiksi hydrofobisen valiinin korvautuminen toisella hydrofobisella aminohapolla, esimerkiksi isoleusiinilla, on huomattavasti yleisempää kuin valiinin korvautuminen hydrofiilisella aminohapolla, esim. arginiinilla.

Aminohappojen samankaltaisuuden astetta pyritään sekvenssirinnastusten yhteydessä kuvaamaan pisteytysmatriiseilla (scoring matrix). Pisteytysmatriisit ovat taulukkoja, joihin on koottu eri aminohappopareille rinnastuksessa käytettävät pisteet. Mitä korkeampi pistemäärä, sitä parempi aminohappopari on rinnastuksen kannalta.

Kuvassa 6.3 esitetty BLOSUM62-matriisi antaa valiinin (V) ja isoleusiinin (I) rinnastuksesta 3 pistettä, kun taas valiinin(V) ja arginiinin(R) rinnastus antaa -3 pistettä. Pisteytysmatriisin arvot eivät kuvaa pelkkää samankaltaisuuden astetta vaan ennemminkin tiettyjen aminohappoparien merkittävyyttä rinnastuksen kannalta tai tietyntyyppisen eron todennäköisyyttä. Tämä tulee esille esimerkiksi tutkimalla pisteytysmatriisin lävistäjällä olevia arvoja. Kahden alaniinin (A) rinnastuksesta BLOSUM62-matriisi antaa neljä pistettä, mutta kahden kysteiinin (C) rinnastus antaa yhdeksän pistettä. Tämä johtuu siitä että kysteiinit ovat yleensä proteiinirakenteen kannalta tärkeämpiä ja evoluution kuluessa hitaammin muuttuvia kuin alaniinit. Rinnastuksen kannalta on siis tärkeämpää saada kysteiinit kuin alaniinit osumaan kohdakkain.

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7	-1	-1	-4	-3	-2
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4	1	-3	-2	-2
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-2	-2	0
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	-3
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

Kuva 6.3: *BLOSUM62 matriisi.*

Pisteytysmatriisista voidaan lukea jokaiselle rinnastuksessa kohdakkain ole-

valle aminohappoparille pistemäärä, joiden summa kuvaa koko rinnastuksen hyvyyttä (mikäli rinnastuksessa ei ole aukkoja). Suurimman pistemäärän saanut rinnastus on ”paras”. Alla olevassa esimerkissä kolme rinnastusta on pisteytetty BLOSUM62-matriisin mukaisesti. Toinen rinnastus antaa korkeimman pistemäärän.

Rinnastus 1.

```
S E K V E N S S I
E R I D S S I
0+0-3-3+0+1-2    = -7
```

Rinnastus 2.

```
S E K V E N S S I
  E R I D S S I
  5+2+3+2+1+4-2  = 15
```

Rinnastus 3.

```
S E K V E N S S I
    E R I D S S I
    1-3-3+1+4+4+4= 8
```

Aminohappojen samankaltaisuudelle ei ole yhtä oikeaa asteikkoa, vaan tilanteesta riippuen rinnastuksen pisteytyksessä kannattaa painottaa eri tekijöitä. Tästä johtuen rinnastusta käyttäviä analyysejä varten on kehitetty suuri joukko erilaisia pisteytysmatriiseja, joista tunnetuimmat ovat BLOSUM- ja PAM-matriisisarjat.

BLOSUM (BLOcks SUbstitution Matrix) -matriisisarja on johdettu BLOCKS- tietokantaan kerätyistä, tietyille proteiiniperheille tyypillisten sekvenssialueiden rinnastuksista. Kullakin BLOSUM-matriisilla on oma tunnuslukunsa, esim. BLOSUM62 tai BLOSUM80. Luku kuvaa matriisin luomisessa käytettyjen sekvenssirinnastusten samankaltaisuutta. Esimerkiksi BLOSUM62-matriisin luomisessa sekvenssien identtisyys on ollut vähintään 62 %. Käytettävä matriisi kannattaa valita tutkittavien sekvenssien samankaltaisuuden perusteella. Jos on syytä olettaa, että vertailtavat sekvenssit ovat vain heikosti samankaltaisia, kannattaa käyttää vähemmän identtisistä sekvensseistä johdettua matriisia, esim. BLOSUM35. Jos taas tutkitaan hyvin samankaltaisia sekvenssejä voidaan käyttää enemmän identtisyyttä painottavia matriiseja kuten BLOSUM80.

PAM-matriisit (Percent Accepted Mutation) ovat rakenteeltaan hyvin samanlaisia kuin BLOSUM-matriisit, mutta niiden arvot on laskettu käyttäen toisenlaista lähestymistapaa. PAM-matriisit, joita kutsutaan niiden kehittäjän Margaret Dayhoffin mukaan usein myös Dayhoff-matriiseiksi, perustuvat tunnetuista proteiiniperheistä evoluutiomallin avulla määritettyihin aminohappojen muutostodennäköisyyksiin.

PAM-matriisisarjassa matriisin tunnusnumero viittaa oletettujen mutaatioiden prosenttiosuuteen. Esim. PAM120 olettaa, että vertailtavien sekvenssein välillä on 100 sekvenssikohtaa kohti tapahtunut 120 muutosta sen jälkeen, kun sukulaissekvenssit ovat lähteneet kehittymään yhteisestä kantamuodostaan. Numeroinnin suunta on siis päinvastainen BLOSUM-matriiseihin ver-

rattuna. Mitä suurempi numero, sitä erilaisemmille sekvensseille matriisi soveltuu. Esim. PAM250 soveltuu parhaiten noin 20 % samankaltaisille sekvensseille, PAM120 40 % samankaltaisille ja PAM60 60 % samankaltaisille sekvensseille.

Sekä PAM- että BLOSUM-matriisit löytyvät EMBOSSin asennushakemistosta (/p/appl/molbio/emboss/latest/share/EMBOSS/data). Tässä hakemistossa oleviin matriiseihin voi EMBOSS-ohjelmissa viitata suoraan niiden nimellä, esim. EPAM120. (matriisien nimien edessä oleva E-kirjain viittaa EMBOSS-ohjelmaan) Proteiinisekvensseille oletusarvona käytetään yleensä BLOSUM62-matriisia. DNA-sekvensseille käytetään oletusarvona EDNAMAT-matriisia, joka antaa identtisistä nukleotidipareista 5 pistettä ja ei identtistä -4 pistettä.

PAM- ja BLOSUM-matriisien lisäksi verkossa on saatavilla myös monia muita pisteytysmatriiseja, kuten PAM-matriisien kaltainen, mutta uudempi Gonnet- matriisisarja tai membraaniproteiineille suunniteltu PHAT-matriisisarja. Siirtämällä haluttu matriisi työskentelyhakemistoon Coronassa ja muokkaamalla se EMBOSS- matriisien malliseksi, sitä voidaan käyttää PAM- tai BLOSUM-matriisien sijaan.

6.3.3 Aukkosakot

Insertioita tai deleetioita kuvataan rinnastuksissa aukoilla (gap). Aukkojen lisäämisen ja koon vaikutusta rinnastuksen laatuun säädellään aukkosakkoparametreilla (gap penalty). Rinnastuksissa aukkoja pyritään välttämään, joten aukkosakkoparametrit asetetaan niin, että aukkojen lisääminen pienentää rinnastuksen samankaltaisuuspisteitä.

Yleisin aukkojen pisteytysmenetelmä on kaksivaiheinen aukkosakko (affine gap penalty). Se koostuu kahdesta parametrista: aukonmuodostussakosta (gap opening penalty, g_o) ja aukonlaajennussakosta (gap extension penalty, g_e). Aukonmuodostussakko kertoo kuinka monta virhepistettä uuden aukon lisääminen rinnastukseen aiheuttaa. Aukonlaajennussakko taas kertoo kuinka monta virhepistettä jo olemassa olevan aukon laajentaminen aiheuttaa. Aukon, jonka pituus on l, aiheuttama aukkosakko $g(l)$ on muodostussakon ja laajennussakon summa.

$$g(l) = g_o + g_e * (l - 1) \tag{6.1}$$

Jos lisätään edellä olleeseen rinnastukseen aukkoja käyttäen aukon muodostussakkona -4 ja aukonlaajennussakkona -1, pystytään virhepisteistä huolimatta löytämään aukollinen rinnastus, joka antaa korkeamman pistemäärän kuin aukottomat rinnastukset.

Rinnastus 4.

```
 S E K V E N S S I
 - E R I D - S S I
-4+5+2+3+2-4+4+4+4= 16
```

Rinnastus 5.

```
S E K V E N S S I
E R I D - - S S I
0+0-3-3-4-1+4+4+4= 2
```

Toisin kuin pisteytysmatriiseille, aukkosakkojen arvojen valinnalle ei ole selkeää käytäntöä, mutta yleensä suositellaan, että aukonmuodostussakko olisi hieman suurempi kuin suurin käytetyssä pisteytysmatriisissa identtisyydestä annettava pistemäärä. Aukonlaajennussakon arvoksi taas suositellaan arvoa, joka on noin 10% aukonmuodostussakosta.

EMBOSS-ohjelmien oletusarvot tai pisteytysmatriiseissa käytetyt aukkosakot on asetettu kokemuksen perusteella niin, että ne tuottavat keskimäärin hyviä tuloksia. Arvojen muuttamista kannattaa kuitenkin kokeilla. Yleensä aukonmuodostussakko on huomattavasti suurempi kuin aukonlaajennussakko. Esimerkiksi *matcher*-ohjelmassa proteiinisekvensseille käytettävät oletusarvot ovat: g_o=14 ja g_e=4 . Kasvatettaessa aukonmuodostussakkoa ja pienennettäessä aukonlaajennussakkoa, tulee rinnastukseen vähemmän, mutta laajempia aukkoja. Päinvastaisessa tilanteessa, laskettaessa aukonmuodostussakkoja ja kasvatettaessa aukonlaajennussakkoa rinnastukseen tulee useampia pienempiä aukkoja.

6.4 Kahden sekvenssin rinnastus EMBOSS-ohjelmilla

Käyttämällä pisteytysmatriisia ja aukkosakkoparametrejä voidaan mikä tahansa sekvenssirinnastus pisteyttää. Ongelmaksi jää löytää rinnastus, joka antaa suurimman pistemäärän ja on siis käytettyjen parametrien mukaan paras rinnastus. Yleisimmin paras rinnastus etsitään ohjelmilla jotka käyttävät *dynaamista optimointia* (dynamic programming).

Sekvenssejä voidaan rinnastaa usein eri tavoin riippuen siitä, millaisia asioita tutkitaan. Kaksi eniten käytettyä rinnastustapaa ovat kattava rinnastus (global alignment) ja paikallinen rinnastus (local alignmnet).

6.4.1 Paikallinen rinnastus water- ja matcher-ohjelmilla

Usein vertailtavat sekvenssit sisältävät samankaltaisia kohtia, mutta eivät kokonaisuutena ole rinnastettavissa. Tällaisia tilanteita ovat esimerkiksi aminohapposekvenssit, joissa on erilainen domeenijärjestys tai eksonirakenteeltaan erilaiset DNA-sekvenssit. Tällöin käytetään paikallista rinnastusta (local aligment). Paikallisessa rinnastuksessa sekvenssejä ei pyritä rinnastamaan koko pituudeltaan, vaan etsitään niistä merkittävästi samankaltaisia alueita ja rinnastetaan vain nämä alueet. Mikäli et ole varma, millainen rinnastus sopisi sekvensseillesi, kannattaa ensin tehdä paikallinen rinnastus.

Paikallisten rinnastusten tekemiseen EMBOSS-paketissa on kaksi ohjelmaa. *Water* käyttää Smithin ja Watermanin algoritmiä, joka etsii kahden sekvens-

Taulukko 6.2: *Kahden sekvenssin vertailuun käytettäviä EMBOSS-ohjelmia.*

Ohjelma	*Toiminta*
`diffseq`	Etsii pieniä eroja (SNP-kohtia) kahden hyvin samanlaisen sekvenssin välillä.
`esim4`	Sim4-ohjelman EMBOSS-versio. Ohjelma on suunniteltu rinnastamaan mRNA-sekvenssi genomiseen sekvenssiin.
`est2genome`	Rinnastaa EST-sekvenssejä genomisen DNA-sekvenssin kanssa.
`matcher`	Löytää parhaat paikalliset rinnastukset kahden sekvenssin välillä.
`needle`	Needleman-Wunsch-tyyppinen kattava rinnastus.
`seqmatchall`	Tekee parittaiset vertailut annetulle sekvenssijoukolle.
`supermatcher`	Rinnastaa kahden nukleotidisekvenssin koodaavat alueet vastaavan kahden proteiinisekvenssin rinnastuksen perusteella.
`stretcher`	Kahden sekvenssin kattava rinnastus.
`tranalign`	Rinnastaa cDNA-sekvenssit annetun proteiinisekvenssirinnastuksen mukaisesti.
`water`	Smith-Waterman-tyyppinen paikallinen rinnastus.
`wordmatch`	Etsii kahden sekvenssin välisiä tietyn mittaisia täsmällisiä osumia.

sin välillä parhaiten rinnastuvan alueen. *Matcher*-ohjelma käyttää samankaltaista, mutta vähemmän muistia vaativaa algoritmiä. Lisäksi matcherilla, toisin kuin water-ohjelmalla, voidaan etsiä useita hyvin rinnastuvia alueita. Yleensä rinnastuksia kannattaa tehdä useita, käyttäen erilaisia aukkoparametreja ja pisteytysmatriiseja ja verrata rinnastuksia saatavilla olevaan biologiseen dataan.

Alla esimerkki paikallisen rinnastuksen suorittamisesta water- ja matcher-ohjelmilla. Esimerkissä käytetään samoja proteiinisekvenssejä kuin sivulla 66 olleessa dotmacher-esimerkissä. Tehdään ensin paikallinen rinnastus `water`-komennolla:

```
orona1 ~> water swiss:csk_mouse swiss:grb2_mouse
Smith-Waterman local alignment.
Gap opening penalty [10.0]: 14
Gap extension penalty [0.5]: 4
Output alignment [csk_mouse.water]:
corona1 ~> less csk_mouse.water

########################################
# Program: water
# Rundate: Fri Jul 21 2006 09:50:50
# Commandline: water
#    [-asequence] swiss:csk_mouse
#    [-bsequence] swiss:grb2_mouse
#    -gapopen 14
#    -gapextend 4
# Align_format: srspair
```

```
# Report_file: csk_mouse.water
########################################

#=======================================
#
# Aligned_sequences: 2
# 1: CSK_MOUSE
# 2: GRB2_MOUSE
# Matrix: EBLOSUM62
# Gap_penalty: 14.0
# Extend_penalty: 4.0
#
# Length: 145
# Identity:      53/145 (36.6%)
# Similarity:    83/145 (57.2%)
# Gaps:          12/145 ( 8.3%)
# Score: 207.0
#
#
#=======================================

CSK_MOUSE         13 ECIAKYNFHGTAEQDLPFCKGDVLTIVAVTKDPNWYKAKNKVGREGIIPA     62
                     |.||||:|..||:.:|.|.:||:|.::....|.|||||:.. |::|.||.
GRB2_MOUSE         2 EAIAKYDFKATADDELSFKRGDILKVLNEECDQNWYKAELN-GKDGFIPK     50

CSK_MOUSE         63 NYVQKREGVKAGTKLSLMPWFHGKITREQAERLLYPP-ETGLFLVRESTN    111
                     ||::.:.          .|||.|||.|.:||.:|... ..|.||:|||.:
GRB2_MOUSE        51 NYIEMKP----------HPWFFGKIPRAKAEEMLSKQRHDGAFLIRESES     90

CSK_MOUSE        112 YPGDYTLCVSCEGKVEHYRIMYHASKLSIDEEVYFENLMQLVEHY    156
                     .|||::|.|.....|:|::::::.:.......|.|.:|.:||:::
GRB2_MOUSE        91 APGDFSLSVKFGNDVQHFKVLRDGAGKYFLWVVKFNSLNELVDYH    135


#---------------------------------------
#---------------------------------------
```

Vertailtavien sekvenssien välillä voi olla useita rinnastuvia kohtia, joista Water-ohjelma tulostaa vain eniten pisteitä saaneen paikallisen rinnastuksen. *Matcher*-ohjelmalla voidaan sen sijaan tulostaa parhaan rinnastuksen lisäksi myös muita korkean pisteytyksen saaneita paikallisia rinnastuksia.

Alla olevassa esimerkissä paikallinen rinnastus on tehty matcherilla. Tulokseen on valittu näkyviin myös toiseksi paras paikallinen rinnastus. Tämä toinen rinnastus vastaa csk-sekvenssin sh3-domeenia (9-70) ja grb2-sekvenssin toista sh3-domeenia (156-215).

```
corona1 ~> matcher swiss:csk_mouse swiss:grb2_mouse -opt
Finds the best local alignments between two sequences
Matrix file [EBLOSUM62]:
Number of alternative matches [1]: 2
Gap penalty [14]:
Gap length penalty [4]:
Output alignment [csk_mouse.matcher]:
corona1 ~> less csk_mouse.matcher
########################################
# Program: matcher
# Rundate: Fri Jul 21 2006 09:56:57
# Commandline: matcher
#    [-asequence] swiss:csk_mouse
#    [-bsequence] swiss:grb2_mouse
#    -options
#    -alternatives 2
```

```
# Align_format: markx0
# Report_file: csk_mouse.matcher
########################################

#=======================================
#
# Aligned_sequences: 2
# 1: CSK_MOUSE
# 2: GRB2_MOUSE
# Matrix: EBLOSUM62
# Gap_penalty: 14
# Extend_penalty: 4
#
# Length: 145
# Identity:      53/145 (36.6%)
# Similarity:    83/145 (57.2%)
# Gaps:          12/145 ( 8.3%)
# Score: 207
#
#
#=======================================

             20        30        40        50        60
CSK_MO ECIAKYNFHGTAEQDLPFCKGDVLTIVAVTKDPNWYKAKNKVGREGIIPA
       : :::::  ::. .: : .:::: ..    : ::::::   :..: ::
GRB2_M EAIAKYDFKATADDELSFKRGDILKVLNEECDQNWYKAELN-GKDGFIPK
              10        20        30        40         50

             70        80        90       100        110
CSK_MO NYVQKREGVKAGTKLSLMPWFHGKITREQAERLLYPPE-TGLFLVRESTN
       ::.. .             ::: ::: : .:: .:      : ::.::: .
GRB2_M NYIEMKPH----------PWFFGKIPRAKAEEMLSKQRHDGAFLIRESES
                          60        70        80        90

             120       130       140       150
CSK_MO YPGDYTLCVSCEGKVEHYRIMYHASKLSIDEEVYFENLMQLVEHY
        :::..: :     :.:....   .       : : .: .::...
GRB2_M APGDFSLSVKFGNDVQHFKVLRDGAGKYFLWVVKFNSLNELVDYH
              100       110       120       130


#=======================================
#
# Aligned_sequences: 2
# 1: CSK_MOUSE
# 2: GRB2_MOUSE
# Matrix: EBLOSUM62
# Gap_penalty: 14
# Extend_penalty: 4
#
# Length: 50
# Identity:      17/50 (34.0%)
# Similarity:    28/50 (56.0%)
# Gaps:           2/50 ( 4.0%)
# Score: 69
#
#
```

```
#=======================================

          20        30        40        50        60
CSK_MO AKYNFHGTAEQDLPFCKGDVLTIVAVTKDPNWYKAKNKVGREGIIPANYV
       : ..:    . .: : .:: . ..    :::::.:     :. :. : :::
GRB2_M ALFDFDPQEDGELGFRRGDFIHVMD-NSDPNWWKGACH-GQTGMFPRNYV
            170       180       190       200       210


#---------------------------------------
#---------------------------------------
```

Yllä olleissa esimerkkitapauksissa rinnastusten arviointia ja parametrien valintaa helpotti se, että sekvenssien domeenirakenne tunnettiin. Todellisessa analyysitilanteessa tällaista tietoa ei useinkaan ole saatavilla, jolloin erilaisten rinnastusvaihtoehtojen arviointi voi olla hyvin hankalaa. Tässä tapauksessa dotmacherin antama pistekartta olisi kuitenkin auttanut rinnastusten arvioinnissa.

6.4.2 Kattava rinnastus needle- ja stretcher-ohjelmilla

Kattavassa rinnastuksessa kahta sekvenssiä verrataan toisiinsa koko pituudeltaan. Tämä menetelmä soveltuu tilanteeseen, jossa tutkittavissa sekvensseissä oletetaan olevan samankaltaisia piirteitä koko sekvenssin matkalta (esimerkiksi sukulaisproteiineja koodaavat cDNA-sekvenssit). EMBOSS:issa on kaksi ohjelmaa kattavaa rinnastusta varten. *Needle*, joka käyttää Needlemanin ja Wunchin algoritmia sekä *stretcher*, joka käyttää Myersin ja Millerin versiota Needlemanin ja Wunchin algoritmista. Stretcher poikkeaa needlestä lähinnä teknisiltä ominaisuuksiltaan. Se vaatii vähemmän muistia kuin needle ja pystyy siten käsittelemään suurempia sekvenssejä. Kumpikin menetelmä löytää annettuja parametreja vastaavan laskennallisesti parhaan rinnastuksen. Tämä ei kuitenkaan välttämättä tarkoita, että rinnastus olisi biologisessa mielessä paras mahdollinen. Aivan kuten paikallisten rinnastusten yhteydessä, myös needle- ja stretcher-ohjelmia kannattaa ajaa erilaisilla parametriyhdistelmillä ja arvioida tuloksia muun biologisen tiedon perusteella.

Alla olevassa esimerkissä ihmisen ja hiiren kaseiini1-sekvenssit on rinnastettu stretcher-ohjelmalla. Mikäli `stretcher`-komennon yhteydessä ei käytetä `-opt` valitsinta, ohjelma kysyy vain syötesekvenssien ja tulostiedoston nimen, ja käyttää pisteytysmatriisina BLOSUM62-matriisia proteiineille ja EDNAMAT-matriisia nukleotidisekvensseille. Alla olevassa esimerkissä `-opt` valitsinta on käytetty. Pisteytysmatriisiksi on valittu BLOSUM35 ja aukonmuodostussakko (gap penalty) on kasvatettu arvoon 20.

```
corona1 ~> stretcher swiss:cas1_human swiss:cas1_mouse -opt
Finds the best global alignment between two sequences
Matrix file [EBLOSUM62]: EBLOSUM35
Gap penalty [12]: 20
Gap length penalty [2]:
Output alignment [cas1_human.stretcher]: kas_rinnastus.txt
```

Rinnastuksen tulos kirjoitettiin tiedostoon kas_rinnastus.txt. Tulostiedoston

alussa on tiedot rinnastetuista sekvensseistä, rinnastuksen saama pistemäärä sekä rinnastuksessa olevien identtisten ja samankaltaisten kohtien lukumäärä ja prosenttiosuus. Sekvenssirinnastuksessa identtiset kohdat on merkitty kaksoispisteellä ja samankaltaiset kohdat pisteellä.

```
corona1 ~> less kas_rinnastus.txt
########################################
# Program: stretcher
# Rundate: Fri Jul 21 2006 10:15:59
# Commandline: stretcher
#    [-asequence] swiss:cas1_human
#    [-bsequence] swiss:cas1_mouse
#    -options
#    -datafile EBLOSUM35
#    -gapopen 20
#    -outfile kas_rinnastus.txt
# Align_format: markx0
# Report_file: kas_rinnastus.txt
########################################

#=======================================
#
# Aligned_sequences: 2
# 1: CAS1_HUMAN
# 2: CAS1_MOUSE
# Matrix: EBLOSUM35
# Gap_penalty: 20
# Extend_penalty: 2
#
# Length: 319
# Identity:      83/319 (26.0%)
# Similarity:   113/319 (35.4%)
# Gaps:         140/319 (43.9%)
# Score: 68
#
#
#=======================================

               10        20        30               40
CAS1_H MRLLILTCLVAVALARPKLPLRYPERLQNPSESS-------EPIPLESRE
       :.:::::::: :.: :.:  :     :   . :       .:  :   .
CAS1_M MKLLILTCLVAAAFAMPRLHSRNAVSSQTQQQHSSSEEIFKQPKYLNLNQ
               10        20        30        40        50

            50        60        70        80        90
CAS1_H EYMNGMNRQRNILREKQTDEIKDTRNESTQNCVVAEPEKMESSISSSSEE
       :..:.::::: .: : : :::: : . ...   .:  .. .::::::::::
CAS1_M EFVNNMNRQRALLTE-QNDEIKVTMDAASEEQAMASAQE-DSSISSSSEE
               60         70        80         90

                           100       110
CAS1_H MS-----------------LSKCA-EQFCRLNEYNQL-------------
                          :  :  ::. :  .::::
CAS1_M SEEAIPNITEQKNIANEDMLNQCTLEQLQRQFKYNQLLQKASLAKQASLF
      100       110       120       130       140
```

```
CAS1_H ------------------------------------------------QLQ
                                                         ::
CAS1_M QQPSLVQQASLFQQPSLLQQASLFQQPSMAQQASLLQQLLLAQQPSLALQ
      150       160       170       180       190

         120                                        130
CAS1_H AAHAQ--------------------------------EQIRRMNENSHVQ
        . ::                                ::  :::  : ::
CAS1_M VSPAQQSSLVQQAFLAQQASLAQKHHPRLSQSYYPHMEQPYRMNAYSQVQ
      200       210       220       230       240

                           140       150       160
CAS1_H V-----------------PFQQLNQLAAYPYAVWYYPQIMQYVPFPPFSD
      .                  :: :. :  :.:    :.:: :::.    . .
CAS1_M MRHPMSVVDQALAQFSVQPFPQIFQYDAFPLWA-YFPQDMQYLTPKAVLN
      250       260       270       280        290

        170       180
CAS1_H ISNPTAHENYEKNNVMLQW
          :   .. :: ::   :
CAS1_M TFKPIVSKDTEKTNV---W
       300       310


#---------------------------------------
#---------------------------------------
```

6.4.3 Rinnastusten jatkokäsittely

Usein rinnastusohjelman antamaa tulosta halutaan muokata tai analysoida edelleen muilla ohjelmilla. Käyttämällä `-aformat`-optiota EMBOSS-ohjelman tekemä rinnastus voidaan tallentaa fasta- tai msf-muodossa, jolloin sitä voidaan käyttää muiden ohjelmien syötteenä.

Esimerkisi edellä tehty kaseiinisekvenssien rinnastus tallentuu fasta-muodossa seuraavasti

```
corona1 ~> stretcher swiss:cas1_human swiss:cas1_mouse -opt \
 -aformat fasta
Finds the best global alignment between two sequences
Matrix file [EBLOSUM62]: EBLOSUM35
Gap penalty [12]: 20
Gap length penalty [2]:
Output alignment [cas1_human.stretcher]: kas_rinnastus.fasta
```

Tulostiedosto (kas_rinnastus.fasta) on nyt kaksi sekvenssiä sisältävä fasta-tiedosto.

```
corona1 ~> less kas_rinnastus.fasta
>CAS1_HUMAN
MRLLILTCLVAVALARPKLPLRYPERLQNPSESS-------EPIPLESREEYMNGMNRQR
NILREKQTDEIKDTRNESTQNCVVAEPEKMESSISSSSEEMS-----------------L
SKCA-EQFCRLNEYNQL-----------------------------------------
-----------------QLQAAHAQ--------------------------------EQI
RRMNENSHVQV-----------------PFQQLNQLAAYPYAVWYYPQIMQYVPFPPFSD
```

```
ISNPTAHENYEKNNVMLQW
>CAS1_MOUSE
MKLLILTCLVAAAFAMPRLHSRNAVSSQTQQQHSSSEEIFKQPKYLNLNQEFVNNMNRQR
ALLTE-QNDEIKVTMDAASEEQAMASAQE-DSSISSSSEESEEAIPNITEQKNIANEDML
NQCTLEQLQRQFKYNQLLQKASLAKQASLFQQPSLVQQASLFQQPSLLQQASLFQQPSMA
QQASLLQQLLLAQQPSLALQVSPAQQSSLVQQAFLAQQASLAQKHHPRLSQSYYPHMEQP
YRMNAYSQVQMRHPMSVVDQALAQFSVQPFPQIFQYDAFPLWA-YFPQDMQYLTPKAVLN
TFKPIVSKDTEKTNV---W
```

Fasta- ja msf-muodossa olevia sekvenssirinnastuksia voidaan analysoida ja muokata useimmilla monen sekvenssin analyysiohjelmilla. Esimerkiksi Coronassa sekvenssirinnastusten muokkaamiseen voidaan käyttää komentorivillä toimivaa *mse*-sekvenssieditoria tai X-pääteyhteyden kautta toimivia *Seaview*- ja *JalView*-ohjelmaa. Sekvenssieditoreista on tarkemmat kuvaukset kappaleessa 8.3. Lisäksi EMBOSS-pakettiin kuuluu muita ohjelmia, joilla voidaan analysoida tai kuvantaa fasta- tai msf-muodossa olevia sekvenssirinnastuksia.

6.5 Nukleotidi-proteiini-rinnastukset Wise2-ohjelmalla

Edellä esitellyt rinnastusmenetelmät toimivat tapauksissa joissa nukleotidisekvenssiä verrattiin toiseen nukleotidisekvenssiin tai proteiinisekvenssiä toiseen proteiinisekvenssiin. EMBOSS-ohjelmat eivät suoraan osaa rinnastaa proteiinisekvenssiä nukleotidisekvenssiin. Periaatteessa ongelma voitaisiin ratkaista kääntämällä nukleotidisekvenssi kaikissa lukukehyksissä ja vertaamalla saatuja proteiinikäännöksiä kohdeproteiiniin. BLASTx-algoritmi käyttää tätä tapaa.

Tämän lähestymistavat käyttökelpoisuutta kuitenkin haittaa se, että se ei pysty huomioimaan tilannetta, jossa sekvenointivirhe jakaa todellisuudessa yhtenäisen proteiinisekvenssin eri lukukehyksissä esiintyviksi alisekvensseiksi. Toinen ongelmia aiheuttava ilmiö ovat aitotumallisten genomeissa esiintyvät intronit, joiden kohdalla tavallinen kaksivaiheinen aukkosakkomalli ei toimi hyvin.

Evan Birneyn ja Richard Copleyn EBI:ssä genomiprojektien tarpeisiin kehittämä Wise2-ohjelmisto pyrkii huomioimaan edellä mainitut ongelmatilanteet. Wise2-ohjelmisto yhdistää paikallisen rinnastusalgoritmin geenirakennemalliin, joka pyrkii ottamaan huomioon sekvenointivirheet ja intronit. Käytettävissä on ihmisen ja *C.elegans*:in genomien mukaa parametrisoituja malleja. Ohjelmistoa voidaan käyttää myös HMM-profiilien vertaamiseen genomisiin tai cDNA-sekvensseihin (katso kappale 8.4.2).

Wise2-ohjelmisto on käytettävissä Corona-palvelimella ja se alustetaan yhdessä HMMER-paketin kanssa komennolla `use hmmer`. Tämän jälkeen käytettävissä ovat kahden sekvenssin rinnastusohjelmat `genewise` (genominen sekvenssi) ja `estwise` (cDNA-sekvenssi), sekä vastaavat sekvenssijoukoille tai tietokannoille tarkoitetut ohjelmat `genewisedb` ja `estwisedb`. Rinnastettavien sekvenssien tulee olla fasta-muodossa.

Yksinkertaisimmillaan proteiinisekvenssin rinnastaminen genomisen nukleotidisekvenssin kanssa tehdään seuraavilla komennoilla:

```
corona1 >use hmmer
corona1 >genewise prio_human.fasta dna_pala.fasta
genewise $Name: wise2-1-22c $ (unreleased release)
This program is freely distributed under a GPL. See source directory
Copyright (c) GRL limited: portions of the code are from separate copyright

Query protein:       PRIO_HUMAN
Comp Matrix:         blosum62.bla
Gap open:            12
Gap extension:       2
Start/End            default
Target Sequence      DNA-pala
Strand:              forward
Start/End (protein)  default
Gene Paras:          human.gf
Codon Table:         codon.table
Subs error:          1e-05
Indel error:         1e-05
Model splice?        model
Model codon bias?    flat
Model intron bias?   tied
Null model           syn
Algorithm            623
GeneWise6 Matrix calculation: [ 197000] Cells 99%

genewise output
Score 147.28 bits over entire alignment
Scores as bits over a synchronous coding model

Warning: The bits scores is not probablistically correct for single seqs
See WWW help for more info

PRIO_HUMAN        19 LGLCKKRPKPG--GWNTGGSR-YPGQGSPGGNRYPPQGGGG--WGQPHG
                     +GLCKKR        NTGGSR YPGQGSPGGN YPPQGGGG  WGQPHG
                     VGLCKKRQNLDLEDENTGGSRLYPGQGSPGGN!YPPQGGGGGNWGQPHG
DNA-pala           2 ggctaaccacgtgggaaggacctcgcgacgga2tcccgggggatgcccg
                     tgtgaagaatataaaacggggtacgaggcgga accaggggggaggacag
                     gcccggcacgcggtgctgacaacaagcttacc tatgagtcctgtgcta


PRIO_HUMAN        63 GGWGQPHGGGWGQPHGGG-WGQPHGGGWGQGGG
                     GGWG PHGGGWGQPHGGG WG PHGGGWGQ  G
                     GGWG-PHGGGWGQPHGGGCWG!PHGGGWGQLHG
DNA-pala         148 ggtg ccgggtgcccgggttg2ccgggtgcccg
                     gggg caggggacaggggggg caggggatag
                     tcga ttatcgtgctatccgt ctttcgagttt


PRIO_HUMAN        95                          HSQWNKPSKPKTNMKHMAGAAAA
                                               +  +K SK  TNM   AGAAAA
                                 T:V[gtc]      SNCGSKLSK--TNMN!GAGAAAA
DNA-pala         243 GGTGGGAG  Intron 1   CAGTCaatgaacta  aaaa2gggggggg
                      <1-----[244  :  498]-1>  gagggatca  cata gcgcccc
                                               tctgtggaa  ccgt taattat


PRIO_HUMAN       119 GAVVGGLGGYMLGS                          MSRPIIHFG
                     GAVVGGLGGYMLGS                          +    +++
                     GAVVGGLGGYMLGS          A:G[ggc]        L----LYIL
DNA-pala         563 gggggggcggtacgaGGTGAGTG  Intron 2   CAGGCc    ttat
                     gcttggtggattgg <1-----[606  :  625]-1>  t    tatt
                     aagagcttccgggt                          g    actg


PRIO_HUMAN       143 SDYEDRYYRENMHRYPNQVYYRPMD--EYSNQNNFVHD
                     +  EDRYYR NM+RYPNQVYYR ++   YSNQNNFVHD
```

```
                    AIEEDRYYR!NMYRYPNQVYYR!QNICQYSNQNNFVHD
DNA-pala        643 gagggcttc2aatctcacgtta4caatctaacaatgcg
                    ctaaagaag atagacaataag aatgaagaaaattaa
                    atggctctg cgctcccagccg atatgttcgcctgtc

//
```

Genewise-ohjelman tulostuksen alussa on lyhyt yhteenveto käytetyistä parametreistä minkä jälkeen tulee varsinainen sekvenssirinnastus. Rinnastuksessa on proteiinisekvenssien lisäksi esitetty myös käännettyjä aminohappoja vastaavat kodonit. Rinnastukseen on myös merkitty introneiksi tulkittujen sekvenssialueiden sijainti ja kohdat joissa lukukehys on vaihtunut (!).

Genewise ja estwise ohjelmien valitsimet voi tulostaa käyttämällä valitsinta `-help`. Esimerkiksi: `estwise -help`

Valitsimilla voidaan vaikuttaa hakuparametreihin sekä tulostusformaattiin. Genewisedb- ja estwisedb-komennoilla voidaan verrata useita proteiinisekvenssejä annettuun nukleotidisekvenssiin tai sekvenssijoukkoon. Kokonaisia proteiinisekvenssitietokantoja käyttävät vertailut ovat kuitenkin hyvin hitaita. Yleensä nämä analyysit kannattaakin tehdä erätöinä (katso kappale 3.6). Erätöissä voidaan valisimilla `-pthread -pthr_no` *pros_lkm* ottaa käyttöön useampi prosesori, mikä nopeuttaa hakua.

Taulukko 6.3: *Wise2 ohjelmat Coronassa.*

Komento	*Selitys*
`genewise`	Vertaa genomisia nukleotidisekvenssiä proteiinisekvenssiin tai proteiini-HMM-profiiliin
`genewisedb`	Vertaa genomisia nukleotidisekvenssejä proteiinisekvenssijoukkoon (tietokantaan) tai proteiini-HMM-profiilijoukkoon (tietokantaan)
`estwise`	Vertaa cDNA-sekvenssiä proteiinisekvenssiin tai proteiini-HMM-profiiliin
`estwisedb`	Vertaa cDNA-sekvenssejä proteiinisekvenssijoukkoon (tietokantaan) tai proteiini-HMM-profiilijoukkoon (tietokantaan)

Lisätietoja Wise2-ohjelmista lötyy EBI:n Wise2-sivulta:

`http://www.ebi.ac.uk/Wise2`

7 Samankaltaisuushaut

7.1 Johdanto

Laboratoriossa tuotetun sekvenssin vertailu sekvenssitietokantoihin sekvenssin tunnistamiseksi tai sen tehtävän ennustamiseksi on eräs käytetyimpiä bioinformatiikan käytännön sovelluksia. Sekvenssihauissa tietokannasta halutaan poimia kaikki ne sekvenssit, jotka ovat hakusekvenssin kanssa merkittävästi samankaltaisia, ja jättää samalla huomiotta ne sekvenssit, joissa samankaltaisuus johtuu vain sattumasta. Tietokantojen jatkuvasti kasvaessa ja tietokantavalikoiman laajetessa hyvin suunniteltu haku voi antaa entistä enemmän tietoa tutkittavasta sekvenssistä.

Sekvenssihakuohjelmien hyviä ja helppokäyttöisiä WWW-käyttöliittymiä löytyy useilta avoimilta palvelimilta, esimerkiksi NCBI:n ja EBI:n sivuilta. CSC:llä on samankaltaisuushakuja voidaan tehdä Corona-palvelimella, joko käyttäen Tutkijan käyttöliittymää tai unix-komentoriviä.

Yksittäisillä sekvensseillä tehtävät haut eivät useinkaan ole Coronassa NCBI:n tai EBI:n hakupalveluja nopeampia. Sen sijaan suurten sekvenssijoukkojen analysoinnissa Corona on usein huomattavasti nopeampi. Coronassa tai tutkijan käyttöliittymässä voidaan tehdä satoja BLAST analyysejä yhtenä ajona. Hauissa voidaan käyttää CSC:n ylläpitämien julkisten sekvenssitietokantojen lisäksi myös omia BLAST-tietokantoja.

Riippumatta siitä missä haut tehdään, käyttäjän tulisi käyttöliittymän toiminnan lisäksi tuntea käyttämänsä menetelmän perusperiaatteet ja sen käyttöä tai tarkkuutta rajoittavat tekijät. Jotta hakua voidaan optimoida, pitäisi myös säädettävissä olevien parametrien vaikutus tuntea. Algoritmin lisäksi myös käytettävän tietokannan sisällön ja rakenteen tunteminen on yhtä tärkeää. Usein sopivan tietokannan, alajoukon tai oman sekvenssijoukon valitseminen suuren yleistietokannan sijaan on tehokkain tapa tarkentaa hakua.

7.2 BLAST

7.2.1 BLAST-algoritmi

Parittaisten sekvenssirinnastusten tekeminen tuntemattoman sekvenssin ja tietokannassa olevien sekvenssien välille on tehokas menetelmä samankaltaisten sekvenssien löytämiseksi. Vaikka yksittäisen paikallisen rinnastuksen laskeminen onkin suhteellisen nopeaa, on laajan tietokannan seulominen tällä tavoin monissa tapauksissa hyvin hidasta. Haun nopeuttamiseksi on kehitetty useita heuristisia menetelmiä, kuten BLAST ja FASTA, jotka poimivat ensin tietokannasta samankaltaisimmilta vaikuttavat sekvenssit ennen rinnastusten tekemistä (FASTA-ohjelmistoa ei käsitellä tässä oppaassa tarkemmin. Ohjelman käyttöohjeet löytyvät osoitteesta

```
http://www.csc.fi/english/research/sciences/bioscience/programs/fasta/
```

BLAST-algoritmi (Basic Local Alignment Search Tool) ei vertaa hakusekvenssiä kokonaisena kaikkiin tietokannan sekvensseihin, vaan ennen varsinaista sekvenssivertailua suoritetaan parittaisia sekvenssivertailuja nopeampi esikäsittely, jolla tietokannasta etsitään todennäköisimmin samankaltaiset sekvenssit ja niiden rinnastuvat alueet. Varsinaiset rinnastukset lasketaan vain näille valikoiduille sekvenssialueille.

Esikäsittelyvaiheen alussa BLAST- algoritmi paloittelee hakusekvenssin word-parametrilla määritetyn pituisiksi alasekvensseiksi. Toisessa vaiheessa luodaan joukko word-parametrin mittaisia sekvenssipaloja, joita hakusekvenssissä ei esiinny. Uusista sekvenssipaloista valitaan ne, joiden samankaltaisuus jonkin alasekvenssin kanssa ylittää asetetun kynnysarvon (T) (samankaltaisuus määritellään pisteytysmatriisilla). Lopputuloksena saadaan joukko sekvenssipaloja, jotka ovat joko identtisiä (sekvenssistä paloitellut) tai samankaltaisia (word-parametrin pituiset, keinotekoiset) hakusekvenssin jonkin osan kanssa.

Tämän jälkeen tietokannasta etsitään alasekvensseille täsmällisiä osumia. Osuman löydyttyä vertailua jatketaan löytynyttä sekvenssiä pitkin kumpaankin suuntaan, jotta saadaan selville, onko samankaltainen alue osumaa pidempi ja ylittääkö alue tietyn kynnysarvon (S). Vasta näin löytyneille sekvensseille tehdään paikallinen rinnastus. Rinnastuksen saamaa pistemäärää arvioidaan käytettyjen parametrien ja tietokannan suhteen. Vertailun perusteella ohjelma arvioi, onko tietokannasta löytynyt sekvenssi merkittävästi samankaltainen hakusekvenssin kanssa.

BLAST-algoritmista on tullut erittäin suosittu ennen kaikkea sen nopeuden johdosta. Ohjelma on myöskin melko herkkä ja spesifinen, eli se löytää yleensä useimmat merkittävästi samankaltaiset sekvenssit ja poimii vain harvoja vääriä osumia. BLAST:in heuristiikka kuitenkin rajoittaa sen tarkkuutta. Etenkin pitkät, heikosti mutta silti merkittävästi samankaltaiset alueet jäävät BLAST-hauissa helposti havaitsematta. Lisäksi hauissa oletetaan, että nukleotidien ja aminohappojen esiintymistaajuudet ovat samanlaiset läpi koko sekvenssin. Tapauksissa, joissa tämä ei pidä paikkaansa (esim. membraaniproteiinit), BLAST ei useinkaan löydä kaikkia merkittävästi samankaltai-

sia sekvenssejä. Tarkempien hakujen tekemiseen voidaan käyttää mm. PSI-BLAST-algoritmia tai Smith-Waterman-, Profiili- ja HMM-hakuja.

7.2.2 BLAST-hakujen eri tyypit

BLAST-algoritmista on viisi versiota, joita käytettään riippuen siitä, ovatko hakusekvenssi ja tarkasteltava tietokanta samaa vai eri tyyppiä (proteiini- tai nukleotidisekvenssejä).

- BLASTn vertaa nukleotidisekvenssiä nukleotidisekvenssitietokantoihin. Yksi vertailu/hakusekvenssi.
- BLASTp vertaa aminohapposekvenssiä aminohapposekvenssitietokantoihin. Yksi vertailu/hakusekvenssi.
- BLASTx lukee nukleotidisekvenssin, kääntää sen aminohapoiksi kaikissa kuudessa lukukehyksessä ja vertaa tuloksia aminohapposekvenssitietokantoihin. Kuusi vertailua/hakusekvenssi.
- TBLASTn lukee aminohapposekvenssin, kääntää nukleotidisekvenssitietokannan sekvenssit kaikissa kuudessa lukukehyksessä proteiineiksi ja vertaa hakusekvenssiä käännöksiin. Kuusi vertailua/hakusekvenssi.
- TBLASTx lukee nukleotidisekvenssin kuudessa lukukehyksessä aminohapoiksi ja vertaa niitä aminohapoiksi käännettyihin tietokantojen nukleotidisekvensseihin. 36 vertailua/hakusekvenssi.

Perus BLAST-hakujen lisäksi Coronassa on käytettävissä myös kolme muuta BLAST-algoritmia: PSI-BLAST, PHI-BLAST ja MegaBLAST. Näitä menetelmiä voidaan käyttää vain komentorivikäytössä.

- PSI-BLAST (Position Spesific Iterative BLAST). Tämä algoritmi toimii vain proteiinisekvensseille. PSI-BLAST-algoritmissä tehdään ensin normaali BLASTp-haku. Löytyneiden sekvenssien joukosta valitaan tietyn kynnysarvon ylittävät sekvenssit, joiden perusteella luodaan sekvenssiryhmää kuvaava sekvenssiprofiili. Sekvenssiprofiililla tehdään uusi tietokantahaku. Löytyneet uudet osumat lisätään profiiliin ja uusia hakuja tehdään niin kauan kunnes uusia osumia ei enää löydy tai asetettu määrä hakukierroksia on tehty. Sekvenssiprofiilin avulla PSI-BLAST pystyy huomioimaan eri aminohapposubstituutioiden merkittävyyden vaihtelua sekvenssien eri kohdissa. Tästä johtuen PSI-BLAST pystyy usein tunnistamaan heikompia homologioita kuin pelkkä BLASTp. Toisaalta, virheellisten sekvenssien kertyminen sekvenssiprofiiliin saattaa tuottaa virheellisiä osumia.
- PHI-BLAST (Pattern-Hit Initiated BLAST). Tämä algoritmi toimii vain proteiinisekvensseille. Algoritmissa käytetään haun syötteenä Prosite-tyyppisiä sekvenssihahmoja (pattern) ja jotakin hahmon sisältävää se-

kvenssiä. Pelkkään sekvenssihahmoon perustuvaan hakuun (esim. fuzzpro) verrattuna, sekä sekvenssiä että sekvenssihahmoa käyttävä PHI-BLAST on tarkempi.

- MegaBLAST on suurten nukleotidisekvenssiaineistojen vertailuun kehitetty hakuohjelma. Se käyttää tavallista BLASTn algoritmia epätarkempaa mutta nopeampaa hakumenetelmää (greedy algorithm). Ohjelmaa kannattaa käyttää mikäli haluttujen osumasekvenssin ja hakusekvensseihin välisten erojen tiedetään olevan pieniä (esimerkiksi sekvensointivirheet).

7.2.3 BLAST-haku Tutkijan käyttöliittymässä

Yksittäinen BLAST-haku on helpointa tehdä Tutkijan käyttöliittymän kautta. Tutkijan käyttöliittymän sekvenssihaut löytyvät biotieteiden sovellussivulta:

`https://hotpage.csc.fi/appl/molbio/index.phtml.en`,

kohdasta *Sequnce searches*. Valittavissa on kaksi BLAST-käyttöliittymää. BLAST-linkistä aukeava käyttöliittymässä voidaan säätää vain BLAST-ohjelman yleisimmin käytettyjä parametreja. BLASTadvnced-käyttöliittymässä voidaan säätää laajempaa joukkoa parametrejä.

BLAST-Käyttöliittymän rakenne muistuttaa EMBOSS-ohjelmien käyttöliittymiä. Pakolliset kentät, joihin on annettava jokin arvo, on merkitty punaisella pallolla. Etenkin BLAST-ajojen tapauksessa on syytä varmistaa, että ensimmäisessä kentässä oleva sähköpostiosoite on oikein, sillä suuriin tietokantoihin tehtävät haut saattavat kestää kauan jolloin käyttöliittymä lähettää viestin ajon valmistumisesta.

Haettava sekvenssin voidaan lukea omalta koneelta painamalla (Browse - nappi), CSC:n kotihakemistosta (linkki: *CSC home directory*) tai se voidaan kopioida suoraan käyttöliittymässä olevaa sekvenssiruutuun. Syötetiedosto voi sisältää useita hakusekvenssejä. Tällöin kaikille tiedostossa oleville sekvensseille tehdään erillinen haku. Kaikkien ajojen tulokset tulevat kuitenkin samaan tiedostoon.

Käyttöliittymä ei tunnista, onko haussa käytettävä sekvenssi nukleotidi- vai proteiinisekvenssi, joten käyttäjän on itse valittava *Blast program*-valitsimella käytettyyn sekvenssiin ja tietokantaan sopiva BLAST-ohjelma. Proteiini- ja nukleotiditietokannoille on erilliset valitsimet. Se, kumpaa tietokantavalitsinta käytetään määräytyy valitun BLAST-ohjelman mukaan: BLASTp- ja BLASTx-haut tehdään proteiinitietokantoihin, joten tietokannan määrää valitsin *Search protein database* ja vastaavasti BLASTn- tBLASTn- ja tBLASTx-hauissa vain valitsimella *Search nucleotide database* on merkitystä.

Käyttäjän on siis aina muistettava asettaa BLAST-ohjelma hakutietokannan ja hakusekvenssin tyypin mukaisesti. Oletusarvoilla käyttöliittymä tekee haun UniProt-tietokannasta olettaen että hakusekvenssi on proteiini.

Kuva 7.1: *Tutkijan käyttöliittymän BLAST-käyttöliittymä.*

Taulukko 7.1: *BLAST-tietokannat CSC:llä.*

Nimi komentorivikäytössä	*Nimi tutkijan käyttöliittymässä*
`Nukleotidisekvenssitietokannat`	
`arabidopsisN`	Arabidopsis nucleotide sequences
`embl_others`	EMBL (without EST GSS and HTG)
`embl_htg`	EMBL HTG sequences
`embl_est`	EMBL EST sequences
`embl_gss`	EMBL GSS sequences
`emblnew`	EMBL updates
`nt`	nt (NCBI's nonredundant nucleotide database)
`refseq_con`	RefSeq Human Contigs
`refseq`	RefSeq RNA Sequeces
`Proteiinisekvenssitietokannat`	
`uniprot`	UniProt (Swiss+TrEMBL)
`swiss`	UniProt/Swiss
`trembl`	UniProt/TrEMBL
`nr`	nr (NCBI's nonredundant protein database)
`pdb`	Protein data bank, 3D

7.2.4 BLAST-haun parametrit

BLAST-algoritmin tärkein parametri on E-arvo, joka asetetaan valitsimella *E-value threshold*. E-arvo määrää, miten suurta samankaltaisuutta pidetään merkittävänä. E-arvon laskemisessa käytetään Karlinin ja Altschulin kehittämää stokastista mallia, joka arvioi, kuinka suurella todennäköisyydellä yhtä hyvät samankaltaisuuspisteet tuottava sekvenssirinnastus syntyy sattumalta. Esimerkiksi jos löytyneen sekvenssin E-arvo on 0.1, tarkoittaa se sitä että todennäköisyys että osuma on virheellinen on yksi kymmenestä. Lopputulokseen valitaan vain ne sekvenssit joiden E-arvo alittaa annetun raja-arvon. Samankaltaisuuskriteerejä tiukennetaan pienentämällä E-arvoa lähemmäs nollaa (esimerkiksi 0.001 Huom. BLAST käyttää desimaalierottimena pistettä). Se millaisia E-arvoja osumat saavat riippuu sekä hakusekvenssistä että tietokannan koosta.

Valitsimesta *Word size:* voidaan muuttaa BLAST-haun ensimmäisessä vaiheessa käytettävää sanapituutta. BLASTn-hauille sanapituuden oletusarvo 11. Muilla BLAST-tyypeille oletusarvona käytetään kolmea.

Kohdasta *Matrix* valitaan sekvenssirinnastusten laskennassa käytettävä pisteytysmatriisi. Valittavana on kaksi PAM- ja kolme BLOSUM-matriisia. BLASTn ohjelma käsittelee vain nukleotidisekvenssejä joten sen yhteydessä tällä patametrillä ei ole merkitystä. Tarkempi kuvaus pisteytysmatriisien käytöstä löytyy kappaleesta 4.3.2

Jos valitsin *Filter query sequence* on voimassa, hakusekvenssit käsitellään suodatusalgoritmilla ennen hakua. Nukleotidisekvensseille käytetään DUST-algoritmia ja proteiineille SEG-algoritmia. Suodatusalgoritmit etsivät tutkittavasta sekvenssistä yksinkertaisia tai toistuvia kohtia kuten poly-A- ja LINE-sekvenssejä, jotka eivät yleensä ole haun kannalta merkittäviä. Nämä alueet merkitään nukleotidisekvensseissä *n*-kirjaimella ja proteiinisekvensseissä *X*-kirjaimella, eikä niitä huomioida varsinaisessa BLAST-haussa. Suodatus on

oletusarvoisesti käytössä, sillä useimmissa tapauksissa se tehostaa hakua ja vähentää vääriä osumia.

Parametrit *description* ja *alignments* määräävät näytettävien osumien ja niihin liittyvien sekvenssirinnastusten maksimimäärät.

BLASTadvanced käyttöliittymästä löytyy lisäksi suuri joukko muita parametreja. Näistä useimmin käytettyjä ovat aukonmuodostus- ja laajennussakot (*Cost to open a gap* ja *Costo to extend a gap*).

Kun parametrit on asetettu, haku käynnistetään napista *Run*, minkä jälkeen selain jää odottamaan tulostiedostojen valmistumista. BLAST-ajon kesto vaihtelee käytettävästä tietokannasta ja hakusekvenssien määrästä riippuen muutamasta sekunnista useisiin tunteihin. Mikäli työ kestää yli minuutin, näytölle aukeaa sivu joka ilmoittaa että työ on lähetty ajoon. Tämä välisivu sisältää myös linkin hakemistoon, jonne tulokset myöhemmin tulevat. Linkki tuloshakemistoon lähetetään käyttäjälle myös sähköpostilla ajon valmistuttua.

7.2.5 BLAST-ajon tulokset

Kun työ on valmis, BLAST-ajon tuloshakemisto sisältää useita tiedostoja. Varsinaisia tulostiedostoja on kolme. Kaikissa näissä on sama informaatio esitettynä hieman eri tavoin.

- blast2.txt sisältää tekstimuotoisen tulostiedoston. Tiedostoa voidaan tarkastella Tutkijan käyttöliittymässä kun tiedostojen käsittelyn oletustoiminnoksi on valittu: muokkaa (edit).
- blast2.html verkkoselaimelle sovitettu tekstimuotoinen tulostiedosto. Tiedostoa voidaan tarkastella Tutkijan käyttöliittymässä kun tiedostojen käsittelyn oletustoiminnoksi on valittu: näytä (show).
- blast_results.cma BioView Viewer ohjelmalle tarkoitettu tulostiedosto. BioView Viewerin käyttö esitellään kappaleessa 7.2.6.

Muut hakemistossa olevat tiedostot ovat ajon suorittamiseen liittyvä väliaikaistiedostoja.

Blast2.txt ja Blast2.html tiedoston alussa on tiedot käytetystä BLAST-ohjelmasta sekä tietokannan ja hakusekvenssin tiedot Tämän jälkeen listataan tietokannasta löytyneiden samankaltaisten sekvenssien nimet ja hakukoodit (accession code). Tuloslistassa on ensimmäisenä samankaltaisimman sekvenssin pisteytys. Bit score ilmoittaa sellaisen haun laajuuden, jolla saataisiin sattumalta yhtä hyvä tai parempi osuma tietokannasta kuin nyt saatu (jos arvo on 30, pitää tehdä 2^{30} parittaista vertailua). Sen jälkeen tulee odotusarvo E, joka on todennäköisyys sille, että tehdyn kokoisessa haussa tulisi yhtä hyvä tai parempi osuma sattumalta. Mitä pienempi odotusarvo E on, sitä samankaltaisempia hakusekvenssi ja osuma ovat.

```
                                                          Score     E
Sequences producing significant alignments:               (bits)  Value

GRB2_MOUSE Q60631 mus musculus (mouse). growt...   419  e-116
GRB2_HUMAN P29354 homo sapiens (human), and r...   418  e-116
GRB2_CHICK Q07883 gallus gallus (chicken). gr...   403  e-111
Q9PU11 Q9pu11 xenopus laevis (african clawed ...   399  e-110
GRB2_XENLA P87379 xenopus laevis (african cla...   398  e-110
GRAP_HUMAN Q13588 homo sapiens (human). grb2-...   258  6e-68
DRK_DROME Q08012 drosophila melanogaster (fru...   257  8e-68
Q9CX99 Q9cx99 mus musculus (mouse). 8430435n1...   252  4e-66
SEM5_CAEEL P29355 caenorhabditis elegans. sex...   239  2e-62
Q9TUB7 Q9tub7 ursus maritimus (polar bear) (t...   175  4e-43
```

Arvo Bit score ja odotusarvo E esitetään myös listan jälkeen tulevien sekvenssirinnastusten yhteydessä, joissa myös kerrotaan prosentteina aukkojen sekä identtisten ja samankaltaisten (positives) aminohappojen tai nukleotidien määrä. Löytyneet samankaltaiset sekvenssialueet esitetään rinnastuksena. Yllä on hakusekvenssi (Query) ja alla BLAST-algoritmin löytämä samankaltainen sekvenssi (Sbjct). Huomaa, että kyseessä on paikallinen rinnastus eli se kattaa yleensä vain osan haku- ja osumasekvensseistä.

Jos kohdakkain olevat aminohapposekvenssikohdat ovat identtiset, merkitään sekvenssien väliin kyseistä aminohappoa vastaava kirjain. Jos aminohapot ovat käytetyn pisteytysmatriisin mukaan huomattavan samankaltaisia, merkitään sekvenssien väliin +. Sekvenssien rinnastuvat alueet voivat olla eripituisia. Tällöin ohjelma sijoittaa aukon sekvenssiin merkiksi lisäykselle tai poistumalle. Tiedoston lopussa on listaus ohjelman käyttämistä parametreista sekä aminohappojen tapauksessa ohjelman käyttämä pisteytysmatriisi.

```
>Q9DDK6 Q9ddk6 salmo salar (atlantic salmon). src-family tyrosine
          kinase sck. 3/2002
         Length = 502

 Score = 88.5 bits (237), Expect = 8e-17
 Identities = 57/168 (33%), Positives = 94/168 (55%), Gaps = 14/168 (8%)

Query: 3   AIAKYDFKATADDELSFKRGDILKVLNEECDQNWYKAEL--NGKDGFIPKNYI---EMKP 57
           AIA YD++   + +L FK+GD LK+L E  +  W++A+    G++GFIP NY+    ++
Sbjct: 59  AIALYDYEGINEGDLGFKKGDKLKILQESGE--WWRAQSISTGQEGFIPSNYVAIDSLET 116

Query: 58  HPWFFGKIPRAKAE-EMLSKQRHDGAFLIRESESAPGDFSLSVK-----FGNDVQHFKV- 110
             WFF  + R  AE ++L+     G+F+IR+SE+  G +SLSV+      G+ V+H+K+
Sbjct: 117 EEWFFKGVSRKDAERQLLASGNKMGSFMIRDSETTKGSYSLSVRDSDSQSGDTVKHYKIR 176

Query: 111 LRDGAGKYFLWVVKFNSLNELVDYHRSTSVSRNQQIFLRDIEQMPQQP 158
             D  G Y    + F +L ELV +++       Q +    +   PQ+P
Sbjct: 177 TLDNGGFYISPRITFTTLQELVSHYKKLGDGLCQALTSPCLSPKPQKP 224
```

BLAST:in tavallinen tekstimuotoinen tulostiedosto ei useinkaan ole käytännöllisin tapa tarkastella haun tuloksia. Alla esiteltävä BioView Viewer -ohjelma mahdollistaa havainnollisemman ja joustavamman tavan tarkastella BLAST-tuloksia. Komentorivikäytössä BLAST-tulos voidaan esittää myös taulukko ja XML-muodossa.

7.2.6 Tulosten tarkastelu BioView Viewer -ohjelmalla

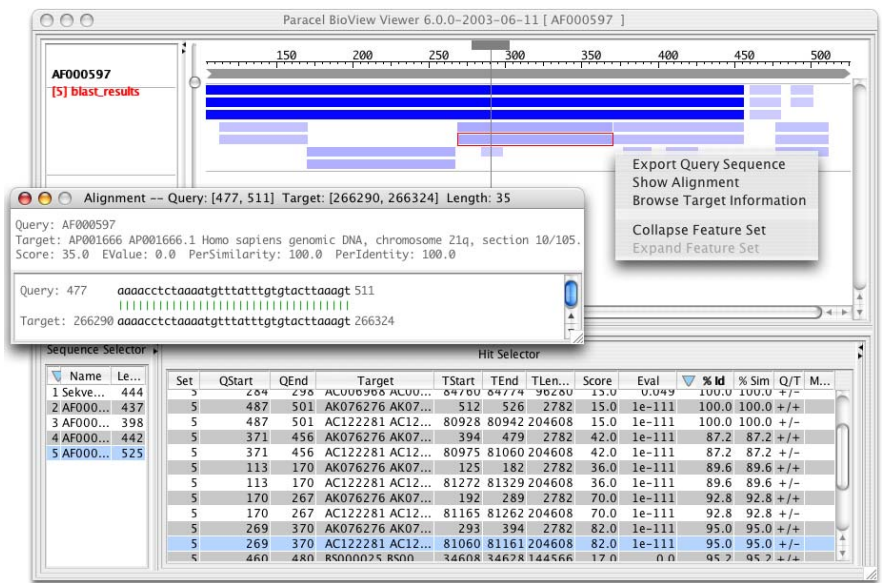
Valikkokäyttöisellä apuohjelmalla BioView Viewer voidaan tarkastella Tutkijan käyttöliittymän ja pb ohjelman tuottamia .cma-muotoisia BLAST:in tu-

lostiedostoja kuva- ja taulukkomuodossa. Tämä Java-ohjelma on asennettava käyttäjän omalle koneelle. Ohjelman voi ladata osoitteesta

```
https://hotpage.csc.fi/appl/molbio/dl.phtml
```

Saatavilla on valmiiksi käännetyt versiot Windows- ja Mac OS X -koneille sekä Java-tiedostot Linux- ja unix-koneita varten. Sivulta löytyy myös asennusohjeita. Ohjelman BioView Viewer asennus vaatii koneen ylläpitäjän oikeudet ja sen käynnistys vaihtelee koneen asetuksista riippuen.

BioView Viewer lukee vain .cma-muodossa olevia tulostiedostoja. Tutkijan käyttöliittymässä olevan .cma-tiedoston lukeminen vaihtelee selaimesta ja koneesta riippuen. Joissakin tapauksissa .cma-tiedosto pitää ensin tallentaa omalle koneelle ja avata erikseen ohjelmalla BioView Viewer. Joissakin tapauksissa tiedosto aukeaa automaattisesti oikein asennettuun BioView Viewer -käyttöliittymään.



Kuva 7.2: *BLAST-haun tulokset BioView Viewer-ohjelmassa*

Tutkijan käyttöliittymän tuottaman .cma-tiedoston aukaiseminen vaihtelee hieman käyttäjän omasta koneesta riippuen. Kaksoisklikkaa ensin linkkiä "BLAST results as a CMA file"tai tiedostoa blast_results.cma ja tallenna tiedosto omalle koneelle klikkaamalla *Save to PC* nappia tai linkkiä *Copy to your computer*. Tämän jälkeen kun selaimesi kysyy millä ohjelmalla tiedosto avataan, valitse ohjelmaksi omalle koneellesi asennettu Bioview Viewer.

Kuvassa 7.2 on esimerkki BioView Viewerissä aukaistusta BLAST-tuloksesta. Kutakin osumasekvenssiä vastaa palkki, joka kuvaa osuman sijaintia hakusekvenssin suhteen, sekä taulukko joka sisältää tiedot löytyneistä sekvensseistä. Palkkien sävy kertoo osuman laadusta. Mitä samankaltaisempi sekvenssialue, sitä tummempi sävy. Nukleotidisekvenssien tapauksessa palkin väri

kertoo, onko kyseessä alkuperäisen suuntainen (sininen) vai käänteinen (vihreä) juoste. Kilkkaamalla palkkia sitä vastaava rivi korostuu taulukossa ja päin vastoin. Klikkaamalla osumapalkkia hiiren oikealla näppäimellä, saat auki valikon, josta voit valita näkyviin osumaa vastaavan rinnastuksen (Show Alignment) tai tutkia osumasekvenssin tietoja www-selaimessa (Browse Target Information) .

Huomaa, että kuvassa olevat osumapalkit eivät ole paremmuusjärjestyksessä. Sen sijaan taulukossa esitettävä data voidaan järjestää minkä tahansa sarakkeen mukaan klikkaamalla sarakkeen otsikkoa. Klikkaamalla sarakkeen otsikkoa uudestaan järjestys muuttuu käänteiseksi. Osumien E-arvot sisältävän sarakkeen käsittelyssä on kuitenkin virhe, sillä BioView Viewer ei ymmärrä e-kirjaimeen perustuvia eksponenttimerkintää eikä osaa järjestää tätä saraketta oikein. Parhaimman osuman löytämiseksi kannattaakin yleensä järjestää osumat Score-sarakkeen mukaisesti.

Näkyvän sekvenssialueen laajuutta voidaan säätää näytön vasemmassa reunassa olevalla säätöjanalla. Näytön yläreunassa olevalla osoittimella valitaan sekvenssikohta, jonka suhteen näkyvää sekvenssialuetta muutetaan.

Mikäli BLAST-haku on tehty useampaa hakusekvenssiä käyttäen, on BioView Viewerin vasemmassa alakulmassa erillinen taulukko, jolla voidaan valita, minkä hakusekvenssein tuloksia tarkastellaan.

7.2.7 BLAST:in komentorivikäyttö

Edellä esitetyn www-käyttöliittymän lisäksi BLAST-ajoja voidaan tehdä myös Coronan komentorivitilassa. BLAST-ohjelmisto otetaan käyttöön komennolla `use blast`. Tämän jälkeen blastall ja muut NCBI-BLAST-komennot ovat käytettävissä. Alustuksen yhteydessä näytölle tulee lista käytössä olevista tietokannoista (katso taulukko 7.2.3).

Tavalliset BLAST-ajot suoritetaan `blastall`-komennolla. Ohjelmalle pitää komennon yhteydessä määritellä ainakin BLAST-ohjelman tyyppi (`-p`), tietokanta (`-d`), hakusekvenssit sisältävä tiedosto (`-i`). BLAST-ohjelma on syötesekvenssistä ja tietokannasta riippuen `blastn`, `blastp`, `blastx`, `tblastn` tai `tblastx` (katso sivu 7.2.2). Hakusekvenssin on oltava fasta-muotoisena tiedostona. USA-määrittelyä ei tässä tapauksessa voida käyttää, sillä BLAST ei ole EMBOSS-ohjelma. Yleensä haun tulokset kannatta ohjata uudeksi tulostiedostoksi valitsimen `-o` avulla. Mikäli tulostiedostoa ei olla määritelty, tulostus ohjautuu suoraan näytölle.

Esimerkiksi sekvenssihaku nukleotidisekvenssillä RefSeq-tietokantaa vastaan tehdään yksinkertaisimmillaan komennoilla:

```
corona1 ~>use blast
corona1 ~>blastall -p blastn -i sekvenssi.fasta -d refseq \
 -o tulokset
```

BLAST-ohjelmien parametrien valitsimet voidaan tulostaa käyttämällä valitsinta `--help` . Esimerkiksi `blastall --help`. Blastall-komennon tärkeim-

mät parametrit ja normaalin tulostiedoston rakenne on esitelty www-käyttöliittymän yhteydessä.

```
corona1 ~> blastall --help

blastall 2.2.13   arguments:

  -p  Program Name [String]
  -d  Database [String]
    default = nr
  -i  Query File [File In]
    default = stdin
  -e  Expectation value (E) [Real]
    default = 10.0
  -m  alignment view options:
0 = pairwise,
1 = query-anchored showing identities,
2 = query-anchored no identities,
3 = flat query-anchored, show identities,
4 = flat query-anchored, no identities,
5 = query-anchored no identities and blunt ends,
6 = flat query-anchored, no identities and blunt ends,
7 = XML Blast output,
8 = tabular,
9 tabular with comment lines
10 ASN, text
11 ASN, binary [Integer]
    default = 0
  -o  BLAST report Output File [File Out]  Optional
    default = stdout
  -F  Filter query sequence (DUST with blastn, SEG with others)
    default = T
(jatkuu...)
```

Esimerkiksi E-arvo asetetaan valitsimella `-e`, suodatusta säädellään valitsimella `-F` ja tulostiedoston muotoa valitsimella `-m`. Jos esimerkiksi haluaisimme tehdä edellä olleen BLAST ajon E-arvolla 0.001, ilman suodatusta ja saada tulokset tekstimuotoisena taulukkona olisi BLAST-komento muotoa:

```
corona1 ~>blastall -p blastn -i sekvenssi.fasta \
-d refseq -o tulokset -e 0.001 -F F -m 9
```

Tavallisia BLAST-hakuja suorittavan blastall-komennon lisäksi BLAST ohjelmistoon kuuluu myös joukko muita komentoja. Näistä yleisimmin käytettyjä ovat:

- `bl2seq` tekee paikallisen rinnastuksen kahdelle sekvenssille käyttäen BLAST-algoritmia.
- `megablast` on suurien nukleotidiaineistojen nopeaa analyysiin kehitetty BLAST-versio. Ohjelma käyttää tavallista blastn-ohjelmaa nopeampaa mutta epätarkempaa hakualgoritmia (greedy BLAST algorithm).
- `blastpgb` ohjelmalla voidaan tehdä proteiinisekvensseille BLASTp-hakuja herkempiä mutta hitaampia iteratiivisia PSI- ja PHI-BLAST hakuja.

- `blastclust` luokittelee annetun sekvenssijoukon. Proteiinijoukon vertailussa käytetään BLASTp algoritmiä ja nukleotidijoukon vertailussa MegaBLAST algoritmia.
- `fastacmd` hakee sekvenssin BLAST-tietokannasta tunnistenumeron tai nimen perusteella
- `formatdb` indeksointiohjelma. Tällä ohjelmalla käyttäjät voivat luoda omia BLAST-tietokantoja.

Nämä komennot toimivat samaan tapaan kuin blastall. Ne tulevat käyttöön `use blast` alustuksen yhteydessä ja niiden valitsimet saa näkyviin valitsimella `--help`. Alla esimerkki komennolla `blastpgb` tehtävässä PSI-BLAST hausta.

```
corona1 ~> blastpgp -i proteiini.fasta -d swiss -j 5 -h 0.001 -o tulos.txt
```

Haussa käytettävä tietokanta hakusekvenssi ja hakuparametrit määritetään samoilla valitsimilla kuin normaalissa BLAST-haussa. Näiden lisäksi on käytetty PSI-BLAST-algoritmiin liittyviä valitsimia `-j` jolla asetetaan yhdelle sekvenssille tehtävien iteraatiokierrosten maksimimääräksi 5 ja `-h` joka määrää kynnysarvon (tässä 0.01), jota paremmat osumat liitetään haussa käytettävään sekvenssiprofiiliin. Valitsinta `-p` ei tässä yhteydessä tarvitse käyttää.

7.2.8 Parallel BLAST

BLAST-ohjelmiston lisukkeeksi Coronaan on lisätty apuohjelma *pb* (Parallel BLAST). . Pb-ohjelma ohjaa BLAST-ajot automaattisesti Coronan eräajojärjestelmään, mikä nopeuttaa BLAST analyysiä jos hakusekvenssejä on useita. Suurille sekvenssimäärille nopeutus tavalliseen BLAST-komentoon verrattuna voi olla monikymmenkertainen. Tutkijan käyttöliittymässä toimiva BLAST käyttää aina pb-ohjelmaa. Pb-ohjelma tulee käyttöön normaalin BLAST-alustuksen yhteydessä ja sen syntaksi on: `pb` *blast komento*. Esimerkiksi

```
corona1 ~> pb blastall -p blastn -i sekvenssit.fasta \
-d refseq -o tulokset
```

Pb-ohjelman yhteydessä voidaan käyttää blastall-, blastpgp- ja megablast-komentoja ja kaikkia niiden valitsimia. Lisäksi pb-ohjelma tunnistaa muutaman ylimääräisen valitsimen. Mikäli pb-komennon yhteydessä käytetään valitsinta `-cma` , se tuottaa normaalin BLAST-tulostiedoston lisäksi BioView Viewer -ohjelmalle tarkoitetun .cma-muotoisen tulostiedoston. (Valitsinta -cma ei voida käyttää samanaikaisesti BLAST-ohjelmien `-m` valitsimen kanssa)

```
corona1 ~> pb blastall -p blastn -i sekvenssit.fasta  \
-d refseq -o tulokset.txt -cma
```

Yllä oleva komento tekee BLAST-ajon sekvenssit.fasta tiedoston sekvensseille ja tulostaa sekä teksti- että .cma-muotoisen tiedoston (tulokset.txt.cma). Mikäli käytössä on X-pääteyhteys, tuloksia voidaan tarkastella komennolla:

```
corona1 ~>bvviever tulokset.txt.cma
```

7.2.9 Omien BLAST-tietokantojen käyttö

Sekvenssihakuja tehtäessä on usein järkevää käyttää laajan yleistietokannan sijaan jotakin erikoistietokantaa tai kohdistaa haku johonkin määrättyyn sekvenssiryhmään, esim. vain tiettyihin eilöihin. CSC:n asiakkaat voivat tarvittaessa asentaa Coronan $METAWRK-hakemistoon omia väliaikaisia sekvenssitietokantoja täydentämään CSC:n ylläpitämää tietokantavalikoimaa. Tietokantana käytettävä sekvenssijoukko voidaan kopioda CSC:lle ulkopuoliselta palvelimelta tai se voidaan koota itse SRS-palvelimen ja `getz`-komennon avulla.

Helpoiten omien tietokantojen käyttö BLAST-hauissa onnistuu pb-ohjelman avulla. Käyttämällä valitsimia `-dbnuc` (nukleotidisekvensseille) tai `-dbprot` (proteiinisekvensseille) voidaan haku kohdistaa valmiin BLAST-tietotokannan sijasta tiedostoon, joka sisältää tietokantasekvenssit fasta-muodossa.

Oletetaan, että haluaisimme verrata BLAST-ohjelmalla suurta joukkoa sekvenssejä NCBI:n influenssatietokantaan, joka löytyy osoitteesta:

`ftp://ftp.ncbi.nih.gov/genomes/INFLUENZA/influenza.fna.`

Koska sekvenssitietokannat ovat usein suuria, niitä kannattaa käsitellä kotihakemiston sijaan $METAWRK-hakemistossa. Siirtyminen $METAWRK-hakemistoon tapahtuu komennolla

```
corona1 ~>cd $METAWRK
```

Tietokantana käytettävä fasta-muotoinen sekvenssitiedosto voidaan hakea NCBI:n ftp-palvelimelta komennolla `wget`:

```
corona1 >wget ftp://ftp.ncbi.nih.gov/genomes/INFLUENZA/influenza.fna
```

Oletetaan, että nukleotidisekvenssit, joilla haku halutaan tehdä on koottu kotihakemistossa olevaan tiedostoon omat.fasta. Kopioidaan tiedosto $METAWR-hakemistoon. Tämän jälkeen hakemistossa pitäisi olla tiedostot influenza.fna ja omat.fasta.

```
corona1 /fs/metawrk/kkayttaj> cp $HOME/omat.fasta ./
corona1 /fs/metawrk/kkayttaj> ls
influenza.fna   omat.fasta
```

Seuraavaksi käynnistetään BLAST-haku. Koska sekä tietokanta- että hakusekvenssit ovat nukleotideja käytetään BLASTn-ohjelmaa. lisäksi käytetään pb-ohjelman valitsinta `-dbnuc` määräämän hakutietokannaksi influenza.fna-tiedosto.

```
corona1 /fs/metawrk/kkayttaj> pb blastall -pblastn -i \
omat.fasta -dbnuc influenza.fna -e 0.01 -o inf_osumat.txt -cma
```

Ajon valmistuttua tuloksia voidaan tarkastella esimerkiksi komennoilla:

```
corona1 /fs/metawrk/kkayttaj> less inf_osmuat.txt
corona1 /fs/metawrk/kkayttaj> bvviewer inf_osmuat.txt.cma
```

7.2.10 Lisätietoa BLASTista

Lisätietoa BLAST-ohjelmien käytöstä CSC:n palvelimilla löytyy BLAST-sivulta:

```
http://www.csc.fi/english/research/sciences/bioscience/programs/blast/
```

8 Usean sekvenssin analyysi

Monen sekvenssin rinnastusta käytetään apuna monissa erityyppisissä tehtävissä. Tavallisia käyttökohteita ovat muun muassa sekvenssijoukkoa kuvaavien Markovin mallien luominen, proteiinirakenteiden homologiamallinnus ja sekvenssien fylogeneettinen analyysi. Biologisesti järkevän rinnastuksen aikaansaaminen onnistuu usein parhaiten rinnastusohjelmien ja käsityön yhdistämisellä.

Tavallisimmat usean sekvenssin rinnastusmenetelmät perustuvat samantyyppiseen lähestymistapaan kuin kahden sekvenssin rinnastus (kappale 6.3). Sekvenssejä verrataan käyttäen pisteytysmatriiseja ja aukkosakkoparametreja. Periaatteessa parhaimman monen sekvenssin rinnastuksen etsimiseen voitaisiin käyttää samankaltaista dynaamiseen optimointiin perustuvaa lähestymistapaa kuin kahden sekvenssin vertailussa, mutta käytännössä tällaisen analyysin laskenta-aika kasvaa hyvin suureksi jo muutaman sekvenssin tapauksessa.

Suurin osa usean sekvenssin rinnastusohjelmista käyttää niin sanottua parittain etenevää monen sekvenssin rinnastusta (pairwise progressive sequence alignment). Tämä algoritmi paloittelee monen sekvenssin rinnastustehtävän joukoksi kahden sekvenssin tai sekvenssiryhmän vertailuja. Algoritmi etenee pääpiirteittäin seuraavasti.

- Tehdään rinnastettaville sekvensseille kaikki parittaiset sekvenssirinnastukset.
- Lasketaan parittaisten rinnastusten avulla sekvenssien samankaltaisuutta kuvaava etäisyysmatriisi, jonka perusteella tehdään puukaavio. Puukaavio kuvaa sekvenssien samankaltaisuutta, ei fylogeneettisiä sukulaissuhteita. Yleensä puukaavion tekemiseen käytetään neighbor joining -algoritmia.
- Rinnastetaan sekvenssit puukaavion mukaisessa järjestyksessä niin, että kerrallaan rinnastetaan kaksi samankaltaisinta sekvenssiä tai sekvenssiryhmää. Parittaisia rinnastuksia tehdään puukaavion mukaisessa järjestyksessä, kunnes kaikki sekvenssit on rinnastettu.

Parittain etenevä monen sekvenssin rinnastus on heuristinen menetelmä. Se ei välttämättä löydä parasta rinnastusta. Esimerkiksi sekvenssien rinnastusjärjestys algoritmin viimeisessä vaiheessa voi vaikuttaa lopputulokseen. Au-

Monen sekvenssin rinnastus pareittain etenevällä monen sekvenssin rinnastusalgoritmillä.

Rinnastettavat sekvenssit:

A CGTTGTATTA
B CGATGATTA
C CGTGTATTA
D CGATGTTA

1. Tehdään kaikki parittaiset rinnastukset ja lasketaan kaikille sekvenssipareille samankalisuutta kuvaava arvo. (Esimerkissä on käytetty clustalw:n oletusarvoja)

A & B 44 pistettä
```
A  CGTTGTATTA
B  CGATG-ATTA
   **  **  ****
```

A & C 77 pistettä
```
A  CGTTGTATTA
C  CGT-GTATTA
   ***  ******
```

A & D 62 pistettä
```
A  CGTTGTATTA
D  --CGATGTTA
     *  ***
```

B & C 44 pistettä
```
B  CGATGATTA
C  CGTGTATTA
   **   ****
```

B & D 75 pistettä
```
B  CGATGATTA
D  CGATG-TTA
   *****  ***
```

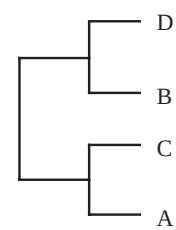
C & D 50 pistettä
```
C  CGTGTATTA
D  CG-ATGTTA
   **  *  ***
```

2. Tehdään rinnastuksia vastaava etäisyysmatriisi, josta lasketaan Neighbor Joining -algoritmilla samankaltaisuuksia kuvaava puukaavio.

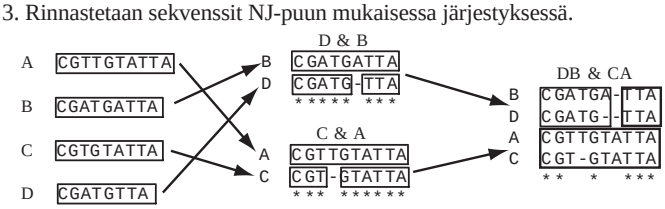
Etäisyysmatriisi

	A	B	C	D
A		44	77	62
B	44		44	75
C	77	44		50
D	62	75	50	

NJ-puu



3. Rinnastetaan sekvenssit NJ-puun mukaisessa järjestyksessä.



Kuva 8.1: *Parittain etenevän sekvenssirinnastuksen toimintaperiaate*

tomaattisten rinnastusohjelmien antamat tulokset ovat usein vain lähtökohta varsinaiselle rinnastukselle. Algoritmin antama tulos on todennäköisesti vain yksi monista yhtä hyvistä rinnastuksista. Rinnastuksia kannattaa tehdä käyttäen hieman erilaisia parametrejä ja muokata lopputuloksia käsin rinnastuseditorilla. Rinnastusta tehdessä kannattaa apuna käyttää myös muuta sekvenssiin liittyvää tietoa kuten sekundäärirakenteiden tai sekvenssimotiivien sijaintia.

8.1 Usean sekvenssin rinnastus Clustal-ohjelmilla

ClustalW on eräs yleisimmin käytetyistä rinnastusohjelmista. CSC:llä ClustalW on käytettävissä Corona-palvelimella komentoriviversiona (`clustalw`) ja X-yhteyttä käyttävänä versiona (`clustalx`) Lisäksi ClustalW-ohjelmasta on selainkäyttöinen versio Tutkijan käyttöliittymässä. (`https://hotpage.csc.fi/appl/molbio/`). Ohjelma voidaan asentaa myös PC- ja Mac-koneisiin.

ClustalW käyttää parittain etenevää monen sekvenssin rinnastusalgoritmia. ClustalW:ssä algoritmiä on kehitetty niin, että se huomioi aminohapposekvenssien koostumuksen ja vanhojen aukkojen sijainnin sekvensseissä uusien aukkojen sijoittelussa. ClustalW:llä on myös mahdollista rinnastaa erillisiä rinnastuksia, lisätä sekvenssejä olemassa olevaan rinnastukseen ja käyttää rinnastuksessa sekundäärirakennetietoa. ClustalX on ClustalW:n päälle rakennettu graafinen käyttöliittymä ja se toimii laskennallisesti aivan samalla tavoin kuin ClustalW:kin.

Lisätietoja Clustalin ominaisuuksista löytyy muun muassa seuraavista lähteistä:

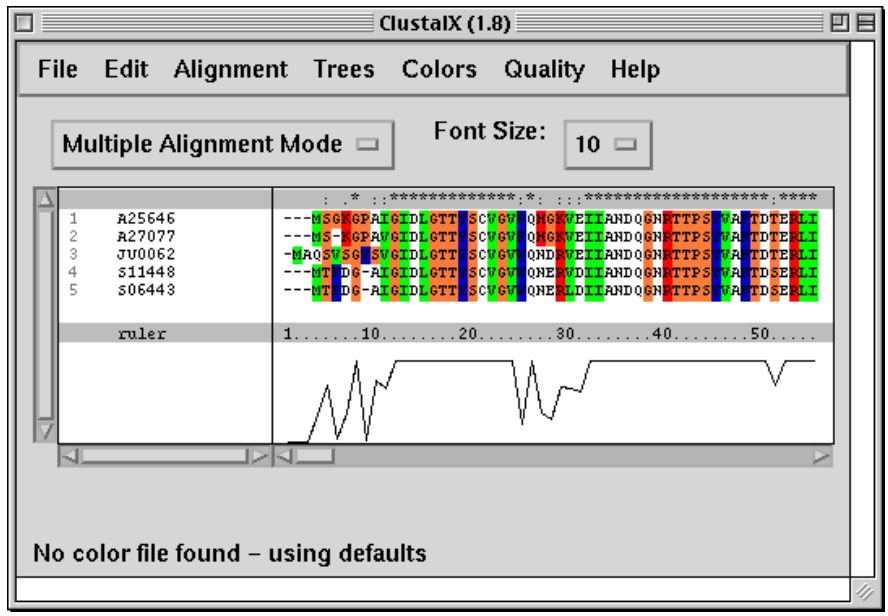
`http://www.csc.fi/molbio/progs/clustalw/`
`http://www.csc.fi/molbio/progs/clustalw/clustalx.pdf`.
`http://www.csc.fi/molbio/progs/clustalw/ms.html`

8.1.1 ClustalX Coronalla

ClustalX aukaistaan kappaleessa 3.1.2 annettujen ohjeiden mukaisesti aukaistussa X-ikkunassa komennolla `clustalx &`. Ohjelma saadaan käynnistymään tausta-ajona lisäämällä komennon perään &-merkki. Ohjelman ollessa tausta-ajossa voi komentorivillä antaa uusia komentoja.

Ohjelma aukeaa näytölle uuteen ikkunaan.

Ohjelmassa on kaksi työskentelytilaa, usean sekvenssin rinnastaminen ja profiilien rinnastaminen. Seuraavassa kuvataan usean sekvenssin rinnastuksen muodostaminen. Analysoitavat sekvenssit kootaan FastA-muodossa yhdeksi monen sekvenssin tiedostoksi, sekvenssin toisensa jälkeen. Erillisten FastA-muotoisten sekvenssitiedostojen yhdistäminen voidaan tehdä esimerkiksi tie-



Kuva 8.2: *ClustalX-käyttöliittymä.*

dostojen nimet sisältävän listatiedoston ja EMBOSS:in seqret-ohjelman avulla.

Sekvenssit luetaan ohjelmaan valinnalla *File/Load Sequences*. Avautuvan valikon avulla valitaan haluttu tiedosto ja painetaan *OK*. Tällöin sekvenssit ilmestyvät näkyviin ClustalX:n ikkunassa. Ikkunassa voi tarvittaessa muuttaa sekvenssien järjestystä nimiä kopioimalla ja liittämällä *Edit*-valikon kautta. Sekvenssin alla näkyvä käyrä kertoo kunkin rinnastuskohdan konservoitumisasteen. Mitä korkeammalla käyrä on, sitä samankaltaisempia kaikki sekvenssit ovat kyseisen kohdan suhteen.

Valikon *Alignment* valinnalla *Do complete alignment* voidaan tehdä usean sekvenssin rinnastus. ClustalX kirjoittaa rinnastuksen automaattisesti Clustal-muodossa .aln-päätteiseen tiedostoon. Jos rinnastus on tarpeen tallentaa myös jossakin muussa muodossa voidaan valinnalla *Alignment/Output Format Options* valita tarvittavat sekvenssirinnastusmuodot ennen rinnastuksen tekemistä. *Alignment*-valikon valinnalla *Alignment Parameters/Multiple Alignment Parameters* voidaan vaihtaa usean sekvenssin rinnastukseen vaikuttavia parametrejä, erityisesti aukkosakkoparametreja. Aukkosakkoparametrien arvot vaikuttavat useimmiten rinnastukseen huomattavasti, eikä oletusarvoilla saatu rinnastus välttämättä ole paras mahdollinen. Niinpä aukkosakkoparametrien arvoja kannattaa vaihdella. *Alignment*-valikon valinnalla *Alignment Parameters/Pairwise Alignment Parameters* voidaan valita parittaisen rinnastuksen parametrit. Valittavista optioista *Slow-accurate* ja *Fast-approximate* vaikuttavat ajoaikaan ja rinnastuksen tarkkuuteen. Slow-optiolla rinnastuksen muodostuminen kestää pidempään, eikä sitä kannata käyttää, jos sekvens-

sejä on kovin paljon (>100) tai ne ovat kovin pitkiä (>10 000 bp). Fast-optiolla rinnastus etenee nopeammin kuin slow-optiolla, mutta tarkkuus kärsii hieman.

Sekvenssien eroja ja yhtäläisyyksiä voi korostaa usealla tavalla valikon *Colors* alla olevilla vaihtoehdoilla. Valikossa *Help* ohjelman ominaisuuksia esitellään lyhyesti ja yleistajuisesti.

ClustalX tallentaa rinnastuksen yhteydessä puukaavion, joka on kuitenkin vain rinnastuksen tarpeisiin muodostettu kaavio. Rinnastettujen sekvenssien perusteella voidaan muodostaa dendrogrammi neighbor-joining-menetelmällä. Tämä tapahtuu valikon *Trees* valinnalla *Draw N-J trees*. Muodostettava puu kuvaa sekvenssien samankaltaisuuksia. ClustalX ei kuitenkaan kuvanna näin saatua puukaaviota. Ne antavat vain puunpiirto-ohjeet, joiden avulla puun voi piirtää esimerkiksi PHYLIP-paketin ohjelmalla Drawgram tai Treeview-ohjelmalla.

8.1.2 ClustalW Tutkijan käyttöliittymässä

ClustalW-ohjelmaa voidaan käyttää Tutkijan käyttöliittymässä, ja se sisältää samat asetusmahdollisuudet kuin ClustalX-ohjelmakin, jonka käyttö on kuvattu yllä. Tutkijan käyttöliittymässä luetaan ensin rinnastettavat sekvenssit ClustalW:hen FastA-muodossa. Jos tarkoituksena on käyttää sekvenssirinnastusta PHYLIP-paketin ohjelmissa, on tarpeen merkitä optio Phylip output alignment format, joka löytyy heti sekvenssien syöttöön tarkoitetun kentän alapuolelta. Tällöin Clustal tallentaa rinnastuksen PHYLIP-muodossa. Muut rinnastusmuodot voidaan valita kohdasta Output parameters. Rinnastuksen nopeutta ja tarkkuutta säätelevä optio löytyy Multiple sequence alignment-kohdan alta, toisin kuin ClustalX:ssä. Slow- ja Fast-optioiden asetukset on kuitenkin esitetty omina kohtinaan. Kun kaikki tarvittavat asetukset on tehty, valitaan sivun yläreunasta Run clustalw. Ajon päätyttyä rinnastetut sekvenssit on mahdollista lähettää analysoitavaksi jollakin PHYLIP-paketin ohjelmalla. Jatkoanalyysi voidaan valita tulostiedostot sisältävän Tutkijan käyttöliittymän välisivun pudotusvalikoista. PHYLIP-paketin käyttö on kuvattu tarkemmin tämän kirjan luvussa Molekyylisystematiikka.

8.2 Usean sekvenssin rinnastus T-Coffeeta käyttäen

T-Coffee yhdistää parittaisen ja usean sekvenssin rinnastuksen parhaat puolet. Sekvenssit rinnastetaan progressiivisella menetelmällä siten, että usean sekvenssin rinnastus sopii parhaiten samojen sekvenssien parittaisiin rinnastuksiin. T-Coffee antaa Clustalia tarkempia tuloksia, mutta rinnastuksen muodostaminen vie vastaavasti pitempään. Tarvittaessa tarkkoja tuloksia suhteellisen pienille (<100) sekvenssijoukoille, kannattaa T-Coffeeta käyttää Clustal-ohjelmien sijaan.

Erittäin nopeisiin ja tarkkoihin usean sekvenssin rinnastuksiin voidaan käyttää myös Sepeli-palvelimelta löytyvää Muscle-ohjelmaa.

8.2.1 T-Coffee Coronalla

T-Coffee löytyy Coronalta, ja sitä käytetään rivikäyttöliittymän kautta. T-Coffee lukee rinnastettavat sekvenssit samaan tapaan kuin ClustalX, FastA-muodossa tiedostosta, jossa sekvenssejä voi olla useita peräkkäin. T-Coffee ajo käynnistyy yksinkertaisimmillaan komennolla:

```
corona > t_coffee sekvenssit.fasta
```

Ajoa voidaan muokata parametreillä, jotka sijoitetaan komentoon. Jos esimerkiksi halutaan tallettaa rinnastukset Clustal-muodossa ja käyttää aukkosakkoparametrejä 10 avaamiselle ja 1 jatkamiselle, voidaan T-Coffee ajaa komennolla

```
corona > t_coffee sekvenssit.fasta -output=clustalw \
-gapopen=10 -gapext=-1
```

Lisätietoja T-Coffeen käytöstä löytyy sivulta: `http://www.csc.fi/molbio/progs/t_coffee/` sekä @CSC-lehden artikkelista: `http://www.csc.fi/lehdet/atcsc/atcsc1-2004/atcsc_2004-1.pdf`, sivut 10-11.

8.3 Rinnastusten editointi- ja visualisointiohjelmia

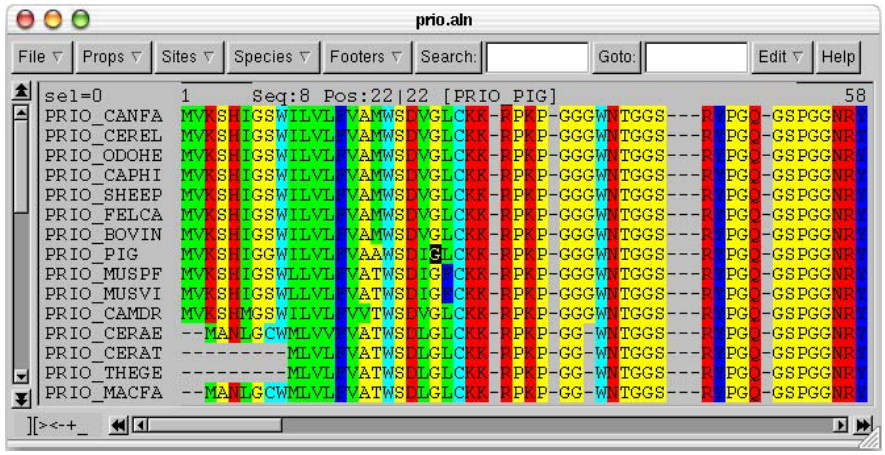
ClustalW:n ja muiden rinnastusohjelmien tekemiä sekvenssirinnastuksia voi tarkastella ja jatkokäsitellä Coronalla muun muassa Seaview ja Jalview ohjelmilla. Molemmat ohjelmat on mahdollista asentaa myös paikallisesti Windows-, Linux- tai Mac OS X-työasemaan.

8.3.1 Rinnastus- ja sekvenssieditori Seaview

Seaview on valikkokäyttöinen sekvenssien ja sekvenssirinnastusten analysointi- ja muokkausohjelma. Valikkokäyttöisen ohjelman käyttö tapahtuu X-ikkunoinnin kautta. X-yhteyksien käyttöönotto on esitetty kappaleessa 3.1.2. Ohjelma Seaview käynnistyy Coronalla komennolla `seaview`.

Ohjelman käynnistyttyä näytölle aukeaa kuvassa 8.3 esitetty ikkuna.

Valikon *File* komentojen avulla voidaan lukea ja tallettaa sekvenssitiedostoja ja sekvenssirinnastuksia. Ohjelma pystyy käyttämään seuraavan tyyppisiä sekvenssitiedostoja: Fasta, Phylip, Clustal, MSF, MASE ja NEXUS. Seaview ei lue oikein Fasta tai msf-tyyppisiä tiedostoja, mikäli jonkin sekvenssin kommenttirivin pituus on yli 78 merkkiä. Ongelman voi kiertää muuttamalla tiedostot komennolla `seqret` Clustal-muotoon (`seqret sekvenssit.fasta aln::sekvenssit.aln`).



Kuva 8.3: *Sekvenssieditori Seaview.*

Valikolla *Props* säädellään sekvenssien esitystapaan. Tästä valikosta löytyy myös valitsin *Allow seq. edition*, joka pitää olla valittuna sekvenssejä muokatessa. Valikkojen *Sites* ja *Species* avulla voidaan määritellä rinnastuksesta sekvenssialueita ja linkittää halutut sekvenssit yhdessä muokattavaksi kokonaisuudeksi. Napin *Search:* avulla voidaan etsiä valitusta sekvenssistä tietty osasekvenssi kirjoittamalla etsittävä alisekvenssi napin viereiseen ruutuun, siirtämällä kursori oikean sekvenssin kohdalle ja paina nappia. Napin *Goto:* avulla kursori siirretään haluttuun sekvenssikohtaan.

Valikon *Edit* komentojen avulla rinnastuksessa olevia sekvenssejä voidaan nimetä, monistaa, poistaa, lukea sekä luoda rinnastukseen uusia sekvenssejä. Lisäksi valikko sisältää komennot konsensussekvenssien ja pistekarttojen (dotplot) laskemiseen.

Lisätietoa Seaview-ohjelmasta löytyy osoitteesta:

```
http://www.csc.fi/molbio/progs/seaview/
```

Ohjelman voi myös asentaa omalle koneelle. Asennustiedosto ja ohje löytyvät Seaview-ohjelman kotisivulta osoitteesta:

```
http://pbil.univ-lyon1.fr/software/seaview.html
```

8.3.2 Jalview rinnastuseditori

Jalview on yksinkertainen valikkokäyttöinen rinnastusten analysointi- ja muokkausohjelma. Seaview-ohjelmasta poiketen sillä voi muokata vain sekvenssien rinnastusta mutta ei itse sekvenssejä. Jalview-ohjelman käyttö vaatii X-pääteyhteyden.

EMBOSS-alustuksen (`use emboss`) jälkeen Jalview:ta voidaan käyttää Jemboss-ohjelmassa (katso kappaletta 5.6) tai se voidaan käynnistää sinällään komennolla:

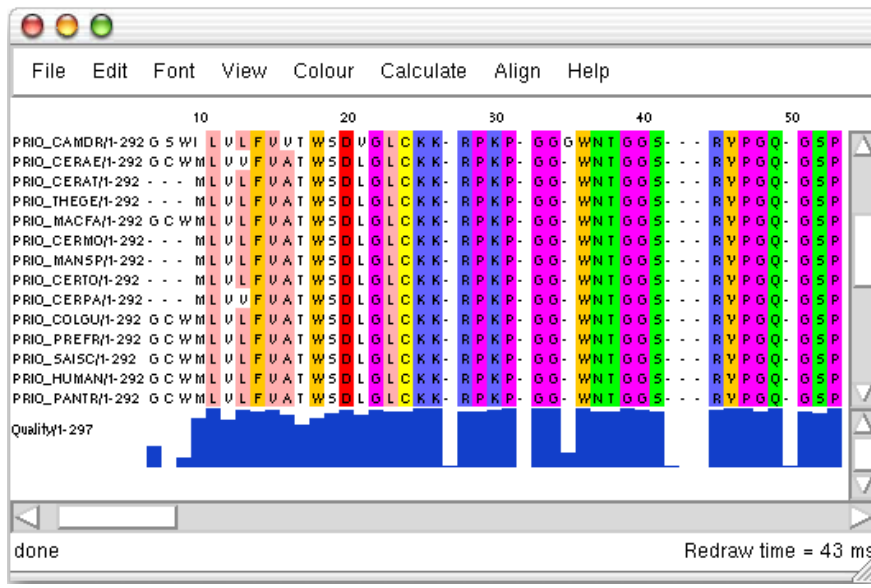
```
Jalview tiedoston_nimi formaatti
```

Huomaa iso alkukirjain komennossa Jalview. Esimerkiksi ohjelman clustalx laskema sekvenssirinnastus tiedostossa sekvensst.aln luetaan ohjelmaan komennolla:

```
corona > Jalview sekvenssit.aln CLUSTAL
```

Muita Jalview-ohjelman ymmärtämiä formaatteja ovat mm. FASTA ja MSF.

Kuvassa 8.4 on esimerkki rinnastuksen tarkastelusta Jalview:lla. Osoittamalla jotakin sekvenssikohtaa hiirellä voidaan sekvenssejä liikutella toistensa suhteen ja muokata rinnastuksessa olevia aukkoja.



Kuva 8.4: *Rinnastuseditori Jalview.*

Valikon *File* komentojen avulla voidaan lukea ja tallettaa sekvenssirinnastuksia sekä tehdä rinnastuksista PostScript-kuvia (PostScript-tulostus toimii vain Jemboss- ohjelmassa). Valikossa *Edit* voidaan muodostaa sekvensseisstä yhtenäisesti muokattavia sekvenssiryhmiä, valita sekvenssejä ja poistaa niitä. Valikosta *Font* voidaan säätää rinnastuksen esityksessä käytettävää kirjasintyyppiä ja kokoa. Valikoilla *View* ja *Colour* vaikutetaan rinnastuksen esitystapaan. Valikkojen *Calculate* ja *Align* avulla rinnastukselle voidaan tehdä erilaisia analyysejä, kuten piirtää sekvenssien samankaltaisuussuhteita kuvaavia puita tai yhdistää rinnastuksessa esiintyviä identtisiä tai samankaltaisia sekvenssejä.

Lisätietoa Jalview:sta löytyy osoitteesta

```
http://www.csc.fi/molbio/progs/emboss/jalview.html
```

8.4 Kätketyt Markovin mallit

Kätkettyihin Markovin malleihin (hidden Markov models, HMM) perustuvia sekvenssihaku- ja rinnastusmenetelmiä pidetään herkimpinä tällä hetkellä yleisesti käytetyistä menetelmistä. HMM-malleissa sekvenssirinnastusten antama tieto sekvenssiryhmälle ominaisista piirteistä esitetään todennäköisyysmallina. Yksittäistä sekvenssiä voidaan verrata tähän malliin ja laskea todennäköisyys, jolla verrattava sekvenssi kuuluu mallin kuvaamaan sekvenssiryhmään.

Yksinkertaisiin, pisteytysmatriiseja käyttäviin menetelmiin (esim. BLAST ja Smith-Waterman) verrattuna HMM-menetelmien etuna on sekvenssirinnastuksen sisällä olevien vaihteluiden huomioiminen samankaltaisuutta arvioitaessa. Kutakin rinnastuksen saraketta vastaavalle Markovin mallin kohdalle määritetään omat parametrinsä, jolloin tuloksena on eri aminohappojen tai nukleotidien yleisyyttä kuvaava todennäköisyysprofiili (HMM-profiili). Näin esimerkiksi rinnastetuista membraaniproteiinien sekvensseistä tehty HMM-profiili pystyy huomioimaan sen, että tietyt aminohapot esiintyvät useammin solun ulkopuolisissa proteiinin osissa, kun taas toiset esiintyvät yleisemmin solun sisäpuolella tai solukalvossa olevissa osissa. Vastaavassa tilanteessa BLAST olettaa aminohappojakauman olevan sama koko sekvenssissä.

HMM-profiileja käyttävät menetelmät muistuttavat huomattavasti sekvenssiprofiileihin perustuvia menetelmiä (esim. Prophet EMBOSS:issa). HMM-menetelmien ominaisuudet ja niiden matemaattiset perusteet ovat kuitenkin paremmin tunnettuja ja käyttökelpoisempia kuin profiilimenetelmissä.

HMM-menetelmien käyttöä rajoittaa niiden vaatima laskentateho sekä menetelmän monimutkaisuus. HMMER-ohjelmalla tehtävät sekvenssihaut vaativat kymmeniä tai satoja kertoja enemmän laskentatehoa kuin BLAST:in kaltaiset yksinkertaisemmat menetelmät. Lisäksi syötteenä käytettävän sekvenssirinnastuksen luominen ja parametrien valinta vaatii esityöskentelyä.

8.4.1 HMMER-ajon vaiheet

HMMER-ohjelmistopaketti ja sen EMBOSS-versio sisältävät joukon ohjelmia joiden avulla rinnastetuista sekvensseistä voidaan luoda Markovin malleja eli HMM-profiileja, tehdä HMM-profiileihin perustuvia sekvenssihakuja tietokannoista, luoda uusia sekvenssirinnastuksia tai etsiä proteiiniperheitä Pfam-tietokannasta. Kullekin HMMER-analyysin vaiheelle on oma ohjelmansa. Ohjelmien nimet eri käyttöympäristöissä näet taulukosta 8.1.

HMM-profiilin luominen

Mikäli Pfam-tietokannasta löytyy sopiva HMM-profiili, kannattaa käyttää sitä. Yleensä analyysin aluksi on kuitenkin luotava uusi HMM-profiili. Profii-

lin luominen on analyysin tärkein vaihe. Mikäli käytetty profiili ei ole hyvä, eivät haku- tai rinnastustuloksetkaan ole parhaita mahdollisia.

Profiilin luomiseksi tarvitaan luotettava, tutkittavaa sekvenssijoukkoa hyvin kuvaava sekvenssirinnastus. Rinnastuksessa tulisi olla useita kymmeniä sekvenssejä. Mitä laajempi linjaus sitä parempi. Sekvenssien samankaltaisuuden tulisi myös olla sopivan tasoista. Hyvin erilaisten sekvenssien luotettava rinnastus esim. clustalw-ohjelmalla saattaa olla vaikeaa. Toisaalta, jos sekvenssit ovat miltei identtisiä, ne antavat vain vähän tietoa siitä, millaista vaihtelua samaan ryhmään kuuluvissa sekvensseissä esiintyy.

Rinnastuksen lisäksi Markovin mallin luomisessa käytettävälle ohjelmalle *ehmmbuild* on kerrottava, millaisiin hakuihin tai rinnastuksiin HMM-profiilia aiotaan käyttää: Tarkastellaanko samankaltaisuutta koko HMM-profiilin matkalta (kattava rinnastus), vai halutaanko etsiä kohtia, joissa verrattava sekvenssi on samankaltainen HMM-profiilin osan kanssa (paikallinen rinnastus). Lisäksi valitaan, etsitäänkö vain paras vai useita rinnastuksia. Vaihtoehtoja on siis neljä:

- Etsitään sekvenssistä kaikki koko HMM-mallin mittaiset rinnastukset
- Etsitään sekvenssistä vain paras HMM-mallin mittainen rinnastus
- Etsitään sekvenssistä kaikki HMM-malliin jonkin osan kanssa rinnastuvat kohdat
- Etsitään sekvenssistä vain paras HMM-mallin jonkin osan kanssa rinnastuva kohta

Käytettävän sekvenssirinnastuksen ja tuotettavan HMM-profiilin tyypin lisäksi käyttäjä voi halutessaan vaikuttaa ehmmbuildin käyttämiin tilastollisiin virheenkorjausmenetelmiin. Ehmmbuild pyrkii kompensoimaan aineiston (pienestä) koosta aiheutuvia puutteita käyttämällä annetun rinnastuksen lisäksi itse luomaansa ylimääräistä ”pseudo-dataa”. Tämä data luodaan satunnaisesti, joko käyttäen joukkoa Dirichlet-jakaumia tai tavallisen pisteytysmatriisin (esim. PAM120) antamaa jakaumaa.

Rinnastettujen sekvenssien liiallista samankaltaisuutta kompensoidaan sekvenssien erilaisilla painotuksilla. Hyvin samankaltaisten sekvenssien painotukset ovat vähäisempiä kuin toisistaan enemmän poikkeavien sekvenssien. Näin joukko hyvin samankaltaisia sekvenssejä ei vääristä HMM-profiilia. Käyttäjä voi joko itse määrittää rinnastettujen sekvenssien painotukset tai antaa ehmmbuild-ohjelman laskea ne. Oletusarvona ehmmbuild käyttää etäisyysmenetelmillä lasketun sukupuun kaltaista luokittelua, josta painotukset johdetaan, mutta myös muita menetelmiä on valittavissa.

Kalibrointi

Kalibroinnilla muutetaan HMM-profiilia paremmin hakuihin sopivaksi. Sekvenssirinnastuksissa (ehmmalign) käytettyjä profiileja ei tarvitse kalibroida.

Kalibrointi vaatii jonkin verran laskentatehoa, ja jos HMM-profiili on pitkä, kannattaa kalibroinnissa käyttää Coronassa eräajojärjestelmää ja useampaa prosessoria.

Haku

Sekvenssihaun perusperiaate on HMMER-paketissa sama kuin tavallisissa pisteytysmatriisimenetelmissä: sekvenssiä ja HMM-profiilia verrataan toisiinsa dynaamista optimointialgoritmiä käyttäen. Rinnastukselle lasketaan sen hyvyyttä kuvaava pistemäärä (hit score) ja edelleen E-arvo, jolla kuvataan löytyneen samankaltaisuuden merkittävyyttä. HMMERin tapauksessa algoritmi poimii ensin verrattavista sekvensseistä HMM-profiilin kanssa samankaltaiset alueet (ohjelma käyttää näistä nimitystä domain). Tämän jälkeen sekvenssit laitetaan ”paremmuusjärjestykseen” sen mukaan, kuinka monta ja kuinka samankaltaista domainia kustakin sekvenssistä on löytynyt.

Sekvenssihaun tekemiseksi ohjelmalle annetaan tiedot sekvenssijoukosta, johon haku tehdään. Yleensä tämä on sekvenssitietokanta, mutta se voi olla myös itsetehty valikoima sekvenssejä. Tämän lisäksi on tietysti määritettävä, mitä HMM-profiilia käytetään ja mihin tulokset talletetaan. Käyttäjä voi halutessaan vaikuttaa myös itse algoritmin parametreihin, mm. siihen, kuinka korkeaa pistetystä tai E-arvoa pidetään riittävän hyvänä.

Haun tuloksena saadussa tiedostossa on ensin esitetty löytyneet sekvenssit E-arvojen mukaisessa järjestyksessä. Tämän jälkeen tulevat tiedot näistä sekvensseistä löytyneistä domaineista, sekvenssien ja HMM-profiilien väliset rinnastukset sekä pistejakauman histogrammi.

8.4.2 Haut Pfam-tietokannoista

HMM-mallien yleisin käyttötapa on verrata proteiini- tai nukleotidisenkvenssiä tunnettuja proteiiniperheitä vastaavia HMM-malleja sisältävään Pfam-tietokantaan (katso Liite 1), ja sillä tavoin tunnistaa sekvenssin funktio tai sukulaisproteiiniperhe.

Coronassa tämä vertailu voidaan proteiinisekvenssille tehdä EMBOSS-paketin ohjelmalla `ehmmpfam`. Nukleotidisekvensseille on käytettävissä Wise2-ohjelmistopaketin ohjelmat `genewisedb` (genomisille sekvensseille) ja `estwisedb` (EST-sekvensseille). Haku voidaan tehdä käyttäen joko *Pfam_ls*- tai *Pfam_fs*-tietokantaa. Pfam_ls-tietokantaa käytettäessä haussa etsitään koko HMM-mallin kattavia osumia kun taas Pfam_fs-tietokannan tapauksessa etsitään osittaisia osumia. Yleensä ensisijaisena hakutietokantana käytetään Pfam_ls-tietokantaa. EMBOSS:in alustuksen yhteydessä määrittyvä `$HMMERDB`-ympäristömuuttuja kertoo Pfam_ls- ja Pfam_fs-tietokantojen sijainnin Coronassa.

Proteiinisekvensseille käytettävän ehmmpfam-ohjelman yhteydessä on valitun Pfam-tietokannan ja hakusekvenssin lisäksi käyttäjän on määritettävä

myös tulostettavien rinnastusten määrä (`-a`) ja poimittavien osumien laatua kuvaava E-arvo (`-e`).

```
corona1 > ehmmpfam $HMMERDB/Pfam_ls proteiini.fasta -a 10 \
-e 1.0 tulos.txt
```

Tulostiedosto sisältää tiedot löytyneistä Pfam-osumista sekä rinnastukset parhaimmille osumille.

Nukleotidisekvensseille käytettävät Wise2-ohjelmat eivät kuulu EMBOSS:iin joten ne eivät ymmärrä USA-merkintätapaa. Haussa käytettävien nukleotidisekvenssien on oltava fasta-muodossa. Wise2- ja natiivi HMMER-ohjelmistot otetaan Coronassa käyttöön komennolla `use hmmer`. Esimerkiksi yhdelle genomiselle sekvenssille tehtävä haku tehdään komennolla.

```
corona1 > use hmmer
corona1 > genewisedb -pfam Pfam_ls -dnas nukleotidi.fasta \
  > Pfam_ls_osumat.txt
```

Mikäli haku halutaan tehdä usealle nukleotidisekvenssille, ei valitsinta `-dnas` käytetä. Esimerkiksi joukolle EST-sekvenssejä tehtävä haku Pfam_fs tietokantaa vastaan käynnistyy komennolla.

```
corona1 > use hmmer
corona1 > estwisedb -pfam Pfam_fs -dnas estit.fasta >  \
Pfam_ls_osumat.txt
```

Genewisedb- ja estwisedb-haut saattavat kestää hyvin kauan, etenkin jos vertailtavia nukleotidisekvenssejä on useita. Käytännössä nämä haut kannattaakin yleensä tehdä eräajoina (katso kappale 3.6). Wise2-komentojen muut valitsimet saat näkyviin käyttämällä komennon yhteydessä valitsinta `-help`.

Ehmmpfan- ja Wise2-ohjelmien tulostiedostot sisältävät vain vähän tietoa löytyneiden Pfam-mallien biologisesta merkityksestä. Tarkemmat tiedot löydetyistä Pfam-malleista voi tarkistaa hakemalla kyseisen mallin tiedot esim. SRS-palvelimen avulla.

Tietty HMM-malli voidaan Coronassa hakea Pfam-tietokannasta komennolla `ehmmfetch`". Esimerkiksi alla oleva komento hakee COX2-nimisen HMM-mallin Pfam_ls-tietokannasta. Malli tallennetaan tiedostoksi COX2.hmm.

```
corona1 > ehmmfetch $HMMERDB/Pfam_ls COX2 COX2.hmm
```

Taulukko 8.1: *Markovin malleihin liittyviä ohjelmia Coronassa.*

Komento	*Selitys*
`ehmmpfam`	Etsi Pfam-tietokannasta sekvenssiä tai profiilia vastaava HMM-profiili
`ehmmbuild`	Luo uusi HMM-profiili annetun rinnastuksen perusteella
`ehmmcalibrate`	Kalibroi HMM-profiili tietokantahakuja varten
`ehmmsearch`	Etsi sekvenssitietokannasta samankaltaisia sekvenssejä käyttäen HMM-profiilia
`ehmmalign`	Rinnasta sekvenssejä olemassa olevaan HMM-profiiliin
`ehmmemit`	Tuota annetun HMM-profiilin mukaisia uusia sekvenssejä
`ehmmfetch`	Hae tietty HMM-profiili HMM-profiilitietokannasta (Pfam, PfamFrag tai oma)
`ehmmconvert`	Muuta HMM-profiili binääritiedostoksi tai tavalliseksi profiiliksi
	esim. Prophet- tai ProfileSearch-ohjelmaa varten
`genewise`	Vertaa genomisia nukleotidisekvenssiä proteiinisekvenssiin tai proteiini-HMM-profiiliin
`genewisedb`	Vertaa genomisia nukleotidisekvenssejä proteiinisekvenssijoukkoon (tietokantaan) tai proteiini-HMM-profiilijoukkoon (tietokantaan)
`estwise`	Vertaa EST-sekvenssiä proteiinisekvenssiin tai proteiini-HMM-profiiliin
`estwisedb`	Vertaa EST-sekvenssejä proteiinisekvenssijoukkoon (tietokantaan) tai proteiini-HMM-profiilijoukkoon (tietokantaan)

8.4.3 Omien HMM-profiilien käyttö Coronassa

Coronassa voidaan proteiinirinnastuksen perusteella luoda myös uusia Markovin malleja ja tehdä niillä hakuja proteiinisekvenssijoukoista tai tietokannoista. Alla esimerkki Markovin mallin luomisesta, kalibroinnista ja mallilla tehtävästä hausta EMBOSS-ohjelmistolla.

Alustetaan EMBOSS.

```
corona > use emboss
```

Haetaan SwissProt-tietokannasta kaikki prionisekvenssit (nimen alkuosa PRIO). Sekvenssejä on 69.

```
corona > seqret "swiss:prio*" prionit.fasta
```

prionit.fasta-tiedoston sisältö voidaan tarkistaa infoseq-komennolla

```
corona > infoseq prionit.fasta
```

Rinnastetaan sekvenssit ClustalW-ohjelmalla (käytetään oletusarvoja)

```
corona > clustalw prionit.fasta
```

Tuloksena saadusta rinnastuksesta (prionit.aln) määritetään haussa käytettävä paikallisiin rinnastuksiin tarkoitettu HMM-profiili.

```
corona1 /wrk/kkmattil> ehmmbuild
Build a profile HMM from an alignment.
Input (aligned) protein sequence set: prionit.aln
Name for this HMM.: prionit
Alignment preference
         D : global-multidomain
         F : local-multidomain
         G : global-singledomain
         S : local-singledomain
Select preference [D]: D
HMMER hidden markov model output file [stdout]: prionit.hmm

(tämän jälkeen ohjelma käynistyy ja näytölle tulostuu
 tietoa ajon etenemisetä)
```

Tuloksena saatu HMM-profiili tulostui tiedostoon prionit.hmm. Tiedoston alussa on tietoa HMM-profiilista. Mm. siitä millä komennoilla se on luotu. Otsikkorivien jälkeen alkaa varsinainen HMM-malli. Alla in tulosetettu näytölle prionit.hmm-mallin 25 ensimmäistä riviä.

```
corona1 > head -25 prionit.hmm
HMMER2.0  [2.3.2]
NAME  prionit
LENG  265
ALPH  Amino
RF    no
CS    no
MAP   yes
COM   hmmbuild -n prionit.hmm --pbswitch 1000 --archpri 0.850000 --idlevel 0.620000 ...
NSEQ  68
DATE  Fri Dec 13 13:13:06 2006
CKSUM 433
XT      -8455     -4  -1000  -1000  -8455     -4  -8455     -4
NULT      -4  -8455
NULE     595  -1558     85    338   -294    453  -1158    197    249    902  -1085  ...
HMM        A      C      D      E      F      G      H      I      K      L      M  ...
         m->m   m->i   m->d   i->m   i->i   d->m   d->d   b->m   m->e
         -415      *  -2000
     1  -1747  -1367  -3986  -3410   -268  -3597  -2452   2141  -3119   2013   2530 ...
     -   -149   -500    233     43   -381    399    106   -626    210   -466   -720 ...
     -    -21  -6672  -7714   -894  -1115   -701  -1378   -415      *
     2  -2228  -1908  -3555  -3373   -604  -3255  -2515    -62  -2963   2976    358 ...
     -   -149   -500    233     43   -381    399    106   -626    210   -466   -720 ...
     -    -21  -6672  -7714   -894  -1115   -701  -1378      *      *
     3   1351   -621  -2503  -1989   -507  -2230  -1401    894  -1747     89    153 ...
     -   -149   -500    233     43   -381    399    106   -626    210   -466   -720 ...
```

Seuraavaksi prionit.hmm kalibroidaan hakuja varten. Huomaa, että kalibroitu profiili kirjoitetaan alkuperäisen profiilin päälle. Tulostiedostoon (kalibrointi.out) kirjautuu vain tietoja kalibroinnin suorittamisesta.

```
corona1 > ehmmcalibrate
Calibrate HMM search statistics
HMMER hidden markov model file: prionit.hmm
HMMER hmmcalibrate program output file [prionit.ehmmcalibrate]: kalibrointi.out

hmmcalibrate --cpu 4  --mean 350.000000  --num 5000  --sd 350.000000 ...

hmmcalibrate -- calibrate HMM search statistics
HMMER 2.3.2 (Oct 2003)
Copyright (C) 1992-2003 HHMI/Washington University School of Medicine
Freely distributed under the GNU General Public License (GPL)
```

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
HMM file:                 prionit.hmm
Length distribution mean: 350
Length distribution s.d.: 350
Number of samples:        5000
random seed:              0
histogram(s) saved to:    [not saved]
POSIX threads:            4
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

HMM    : prionit
mu     :  -181.928482
lambda :     0.162973
max    :  -129.479996
//
```

Kalibroidun profiilin avulla voidaan etsiä samankaltaisia sekvenssejä useita proteiinisekvenssejä sisältävästä fasta-muotoisesta tiedostosta komennolla `ehmmsearch`.

```
corona1 > ehmmsearch -A 20 -e 0.1 prionit.hmm \
 proteiinit.fasta > prionithaku.out
```

Yllä oleva komento vertaa prionit.hmm-mallia tiedostossa proteinit.fasta oleviin sekvensseihin Tulostiedostoon `prionithaku.out` tulee tiedot osumista joiden saama osuman laatua kuvaava E-arvo on parempi (eli pienempi kuin) 0.1. Rinnastukset esitetään enintään 20 parhaalle osumalle.

Mikäli HMM-mallia halutaan verrata yksittäiseen nukleotidisekvenssiin (esim. yhteen kromosomiin) voidaan haussa käyttää `genewise`- tai `estwise`-ohjelmia.

```
corona1 > genewise -hmmer prionit.hmm sekvenssi1.fasta  \
 > tulos.txt
```

Mikäli HMM-mallia halutaan verrata nukleotidisekvenssijoukkoon se voidaan tehdä genewisedb- ja estwisedb-ohjelmilla. Ohjelmia käytetään samaan tapaan kuin Pfam-hakujen yhteydessä sillä erotuksella että yksittäistä HMM-mallia käytettäessä valitsin `-pfam` korvataan valitsimella `-hmmer`(katso sivu 8.4.2).

8.4.4 Lisätietoa HMMERista

BioBoxin HMMER-sivun www-osoite on:

`http://www.csc.fi/english/research/sciences/bioscience/programs/hmmer/`

8.5 Muita monen sekvenssin analyysiohjelmia

EMBOSS-paketin ohjelma *MEME* etsii sekvenssijoukosta motiiveja eli kaikissa sekvensseissä esiintyviä lyhyitä samanlaisia sekvenssijaksoja. Sekvenssien ei tarvitse olla rinnastettuja. EMBOSS-paketista löytyy myös ohjelmia

(esimerkiksi *prophecy* ja *prophet*), jotka käsittelevät rinnastettujen sekvenssien profiileja. Ne sisältävät tiivistettynä rinnastettujen sekvenssien sisältämän tiedon ja niitä voidaan käyttää esimerkiksi tietokantahauissa Markovin mallien tapaan. Myös MEME:n vastauksena antama motiivi käy profiilihaun syöttötiedostoksi.

9 Molekyylisystematiikka

9.1 Molekyylisystematiikan taustaa

Suurin osa eliöiden sukulaisuussuhteiden määrittämiseen pyrkivistä tutkimuksista käyttänee nykyisin sekvenssiaineistoja. Sukulaisuussuhteita voidaan määrittää rinnastettujen sekvenssien perusteella esimerkiksi seuraavassa esiteltävien ohjelmien avulla. Ohjelmat esitellään tässä lyhyesti, sillä menetelmien teoreettinen tarkastelu on esitelty oppaassa Bioinformatiikan perusteet:

`http://www.csc.fi/english/research/sciences/bioscience/books/bioinfo`,

ja ohjelmien tarkemmat käyttöohjeet oppaissa Molecular systematics ja Primer to phylogenetic analysis using the PHYLIP package:

`http://www.csc.fi/english/research/sciences/bioscience/books/molsyst`
`http://koti.mbnet.fi/tuimala/oppaat/phylip2.pdf`.

Eliöiden sukulaisuussuhteiden määrittäminen sekvenssien perusteella pohjautuu ajatukseen, että kaikilla tutkittavilla lajeilla (sekvensseillä) on aiemmin ollut yhteinen kantamuoto (sekvenssi), josta nykyiset sekvenssit ovat kehittyneet. Kaikkia sekvensseissä tapahtuneita muutoksia on luultavasti mahdotonta jäljittää, sillä samaan sekvenssikohtaan sattuneet useammat muutokset eivät välttämättä näy kuin yhtenä ainoana havaittuna muutoksena. Evoluutiomallien avulla voidaan yrittää korjata tällaisista muutoksista johtuvia ongelmia.

Sukulaisuussuhteiden määrittämiseen voidaan käyttää yhtälailla DNA- kuin aminohapposekvenssejäkin, mutta DNA-sekvenssien etuna on, että ne sisältävät aminohapposekvenssejä enemmän informaatiota. Toisaalta, kovin kaukaisten sukulaisten DNA-sekvenssejä voi olla vaikeampi rinnastaa kuin aminohapposekvenssejä.

9.2 Menetelmät

Molekyylisystematiikan käytetyimpiä menetelmia ovat parsimoniamenetelmä ja suurimman uskottavuuden (ML) menetelmä. Etäisyysmenetelmät, ku-

ten Neighbor-joining ovat huomattavasti vähemmän käytettyjä. Perinteiset ML-menetelmät ovat laskennallisesti suhteellisen raskaita, ja Bayesilaiset menetelmät ovatkin kasvattaneet suosiotaan niiden kustannuksella. Seuraavassa esitellään eri menetelmät ja niihin oleellisesti liittyvät käsitteet.

9.2.1 Etäisyysmenetelmät

Etäisyysmenetelmissä lasketaan rinnastettujen sekvenssien väliset erot, jotka muutetaan evoluutiomallia käyttäen etäisyydeksi. Etäisyyksien perusteella voidaan muodostaa puu käyttäen jotakin algoritmia, kuten UPGMA tai neighbor-joining. Etäisyysmenetelmiä pidetään turhan yksinkertaisina ja osittain vanhentuneina. Siksi ensisijaiseksi analyysimenetelmäksi tulisi valita jokin muu menetelmä. Mahdollisen poikkeuksen tekee LogDet-evoluutiomalli etäisyysmenetelmiin yhdistettynä. Tällainen menetelmä toimii hyvin, jos sekvenssit eroavat nukleotidifrekvensseiltään.

9.2.2 Parsimonia

Parsimoniamenetelmissä etsitään puuta, joka minimoi sekvensseihin johtaneiden evolutiivisten muutosten määrän. Parhaan parsimoniapuun pituus (muutosten määrä) on siis pienin mahdollinen. Parhaan puun löytämiseksi käytetään yleensä jonkinlaista heuristista hakua.

9.2.3 Suurimman uskottavuuden menetelmät

Suurimman uskottavuuden menetelmissä käytetään evoluutiomallia, jonka perusteella puun uskottavuus lasketaan. Paras puu saa kaikkein suurimman uskottavuusarvon. Parhaan puun löytämiseksi käytetään yleensä jonkinlaista heuristista hakua.

9.2.4 Ulkoryhmä ja juuri

Usein puut esitetään juurellisina, esimerkiksi vasemmalta oikealle suuntautuvina kaavioina. Puun juurtaminen voidaan tehdä esimerkiksi käyttämällä sopivaa ulkoryhmää. Puun juurtaminen antaa puussa tapahtuneelle evoluutiolle suunnan. Ulkoryhmänä käytetään usein tutkittavan ryhmän (sisäryhmän) sisarryhmää. Esimerkiksi tukittaessa ihmisen, simpanssin ja gorillan sukulaisuussuhteita, voitaisiin puun juurtamiseen käyttää orankia, joka on mainittujen lajien sisarryhmä.

9.2.5 Evoluutiomalli

Evoluutiomallilla tarkoitetaan mallia, jota noudattaen sekvenssien uskotaan evolvoituneen. Etäisyysmenetelmien yhteydessä evoluutiomallia käytetään sekvenssien välisten etäisyyksien laskemiseen. ML- ja Bayesilaisissa menetelmissä mallia tarvitaan puun uskottavuuden määrittämiseen. Parsimoniamenetelmässä malli on hankalampi käsite, mutta se määritellään tässä tavaksi, jolla esimerkiksi nukleotidimuutoksia painotetaan. Esimerkiksi yksinkertainen malli parsimoniamenetelmän tapauksessa on tilanne, jossa kukin nukleotidimuutos lisää puun pituutta yhtä paljon (Fitch-parsimonia).

Yksinkertaisin DNA-evoluutiomalli on Jukes-Cantorin vuonna 1969 esittämä malli, jossa kaikkien nukleotidien runsaussuhteet ovat yhtä suuret, ja kaikki muutokset nukleotidista toiseksi ovat yhtä todennäköisiä. Kimuran kahden parametrin malli laajentaa Jukes-Cantorin mallia sallimalla nukleotidimuutoksille erilaiset todennäköisyydet: transitioille ja transversioille voidaan määrätä toisistaan eroavat todennäköisyydet. Hasegawa-Kishino-Yano-malli (HKY) antaa sekä nukleotidifrekvenssien että muutostodennäköisyyksien vaihdella. Esimerkiksi PHYLIP-ohjelmissa Dnaml ja Dnamlk toteutettu F84-malli muistuttaa HKY-mallia hyvin likeisesti, vaikka se on määritelty hiukan eri tavalla. Aminohapposekvensseille tarkoitetut evoluutiomallit ovat useinmiten todellisiin havaintoihin perustuvia malleja, kuten PAM- tai JTT-matriisit.

9.2.6 Heuristinen haku

Mahdollisten puiden määrä kasvaa hyvin voimakkaasti analyysiin valittujen sekvenssien lukumäärän suhteen. Luotettavin tapa parhaan puun löytämiseksi olisi käydä läpi kaikki mahdolliset puut (exhaustive search tai branch-and-bound). Nykyisillä menetelmillä ei kuitenkaan ole järkevää käydä läpi kaikkia mahdollisia puita noin 15-20 sekvenssiä suuremmille aineistoille. Jos aineisto on tätä suurempi, pitää turvautua heuristisiin menetelmiin, joiden ei voida taata löytävän parasta mahdollista puuta. Yleensä heuristisella haulla löytynyt puu on kuitenkin varsin lähellä parasta mahdollista puuta.

Heuristisia hakumenetelmiä ovat muun muassa nearest-neighbor interchange (NNI), subtree pruning and regrafting (SPR) ja tree bisection and reconnection (TBR). Näistä hakumenetelmistä tehokkain on TBR, ja vähiten tehokas on NNI. SPR:n teho jää näiden ääripäiden väliin. Teholla tarkoitetaan tässä kykyä löytää kaikkein paras mahdollinen puu. PHYLIP-ohjelmissa oletusarvoinen hakumenetelmä on yleensä NNI, mutta valitsemalla global search, ohjelma voidaan vaihtaa käyttämään SPR-menetelmää.

9.2.7 Tulosten arviointi

Puun luotettavuutta voidaan testata eri tavoin. Jo perinteiseksi muodostunut bootstrapping-menetelmä lienee yleisimmin käytetty menetelmä, mutta

myös jackknifing-menetelmä ja Bremerin tukiarvo ovat yleisesti käytössä. Bootstrapping-menetelmässä muodostetaan uudelleenotannalla uusia aineistoa n kappaletta, jossa n on tyypillisesti 100-1000. Uudelleenotannassa kukin alkuperäisen rinnastuksen kohta voidaan valita uuteen rinnastukseen satunnaisen määrän kertoja, jopa jättää kyseinen kohta kokonaan huomiotta. Kukin uudelleenotannalla muodostettu rinnastus on saman pituinen kuin alkuperäinen aineisto. Uudelleenotannalla muodostetut aineistot analysoidaan kuten alkuperäinen aineisto, ja saatujen puiden perusteella muodostetaan enemmistökompromissipuu (majority rule). Enemmistökompromissipuussa esitetään sellaiset lajiryhmät, jotka esiintyivät vähintään 50% uudelleenotannalla muodostettujen aineistojen perusteella saaduista puista. Jos enemmistökompromissipuussa esitetty numeroarvo on korkea (esimerkiksi yli 70%), voidaan olettaa, että puussa esitetty lajiryhmä on oikeasti olemassa tai että aineisto tukee sellaisen lajiryhmän olemassaoloa voimakkaasti.

9.3 Ohjelmat

9.3.1 PHYLIP Coronalla

Phylip-ohjelmapaketti sisältää ohjelmia, joilla voidaan analysoida sekvenssirinnastuksia etäisyys-, parsimonia- ja suurimman uskottavuuden menetelmillä. Ennen PHYLIP-ohjelmien käyttöä on muodostettava sekvenssirinnastus esimerkiksi Clustal-ohjelmalla. Sekvenssirinnastus tulee tallettaa .phy (PHYLIP) -muodossa. Muutoin analysointi ei onnistu. Jos rinnastus on jossakin muussa muodossa, sen voi muuttaa PHYLIP-muotoon esimerkiksi Clustalia käyttäen.

Phylip-rinnastuksen tuntee siitä, että ensimmäisellä rivillä on kaksi numeroa. Ensimmäinen ilmaisee rinnastuksessa olevien sekvenssien lukumäärän, ja jälkimmäinen rinnastuksen pituuden.

```
    13    404
kiivi3      GCTTAGCCCT AAATCCTGGT GCTTGCATTA CCCAAGTACC CGCCCGAGAA
kiivi1      GCTTAGCCCT AAATCCTGGT ACTTACGTTA CCTAAGTACC CGCCCGAGAA
kiivi2      GCTTAGCCCT AAATCCTGGT GCTTACATTA CCTAAGTACC CGCCCGAGAA
moa1        GCTTAGCCCT AAATCCAGAT ACTTACCCTA CACAAGTATC CGCCCGAGAA
kasuaari    GCTTAGCCCT AAATCTTGAT ACTCGCTATA CCTGAGTATC CGCCCGAGAA
rhea2       GCTTAGCCCT AAATCCCGAT ACTTACCCCA CCCAAGTATC CGCCCGAGAA
moa3        GCTTGGCCCT AAATCTAGAT ACTTACCCTA CACAAGTATC CGCCTGAGAA
emu         GCTTAGCCCT AAATCTTGAT ACTCACCTTA CCAGAGCATC CGCCTGAGAA
moa4        GCTTAGCCCT AAATCCAGAT ACTTACCCCA CACAAGTATC CGCCCGAGAA
strutsi     GCTTAGCCTT AAATCATGAT ACTTACCCCA CCTAAGTATC CGCCCGAGAA
moa2        GCTTAGCCCT AAATCCAGAT ACTTACCCTA CACAAGTATC CGCCCGAGAA
rhea1       GCTTAGCCCT AAATCCTGAT ACTTACCCCA CCTAAGTATC CGCCCGAGAA
tinami      GCTTAGCCCT AAATCTAGAT ACTTAAACTA CCCAAGTATC CGCCTGAGAA
(jatkuu)
```

PHYLIP-ohjelmapaketti alustetaan Coronalla seuraavasti:

```
corona1 ~> use phylip
Phylip 3.65 in use
```

```
[phylip is now in use]
corona1 ~>
```

Nyt yksittäiset PHYLIP-ohjelmat käynnistyvät omilla nimillään. Seuraavassa kuvataan yksinkertainen analyysi parsimoniamenetelmällä. Tutkittavia lajeja on 13, joten aineistolle voidaan käyttää branch-and-bound hakualgoritmia. Tämä löytyy PHYLIP-ohjelmasta Dnapenny. Ajetaan Dnapenny:

```
corona1 ~> dnapenny
```

Seuraavaksi ohjelma kysyy syötetiedoston nimeä. Esimerkkirinnastuksemme löytyy tiedostosta ratite.phy.

```
dnapenny: can't find input file "infile"
Please enter a new file name> ratite.phy
```

Ohjelman asetuksia muutetaan valikon kautta. Dnapenny-ohjelman valikko näyttää seuraavalta:

```
Settings for this run:
  H        How many groups of  100 trees:  1000
  F        How often to report, in trees:   100
  S           Branch and bound is simple?  Yes
  O                      Outgroup root?  No, use as outgroup species  1
  T              Use Threshold parsimony?  No, use ordinary parsimony
  W                      Sites weighted?  No
  M           Analyze multiple data sets?  No
  I          Input sequences interleaved?  Yes
  0   Terminal type (IBM PC, ANSI, none)?  ANSI
  1    Print out the data at start of run  No
  2  Print indications of progress of run  Yes
  3                       Print out tree  Yes
  4         Print out steps in each site  No
  5  Print sequences at all nodes of tree  No
  6       Write out trees onto tree file?  Yes

Are these settings correct? (type Y or the letter for one to change)
```

Kukin rivi vastaa yhtä asetusta. Rivin edessä oleva kirjain merkitsee valintaa, jolla asetusta voidaan muuttaa. Analyysissä on mukana ulkoryhmä tinami, joka on rinnastuksen 13. laji (ylhäältä alaspäin laskettuna). Asetetetaan tinami ulkoryhmäksi käyttämällä asetusta O. Kirjoitetaan riville O, ja painetaan Return. Tällöin ohjelma kysyy ulkoryhmän numeroa, joka sille annetaan:

```
Type number of the outgroup:
13
```

Ohjelma valikko näyttää nyt tehdyn valinnan:

```
Settings for this run:
  H        How many groups of  100 trees:  1000
  F        How often to report, in trees:   100
  S           Branch and bound is simple?  Yes
  O                      Outgroup root?  Yes, at sequence number 13
  T              Use Threshold parsimony?  No, use ordinary parsimony
  W                      Sites weighted?  No
  M           Analyze multiple data sets?  No
  I          Input sequences interleaved?  Yes
  0   Terminal type (IBM PC, ANSI, none)?  ANSI
```

```
  1    Print out the data at start of run  No
  2  Print indications of progress of run  Yes
  3                        Print out tree  Yes
  4         Print out steps in each site  No
  5  Print sequences at all nodes of tree  No
  6       Write out trees onto tree file?  Yes

Are these settings correct? (type Y or the letter for one to change)
```

Asetukset hyväksytään ja ohjelma saatetaan ajoon valinnalla Y ja Return. Kaikki Phylip-ohjelmat asetetaan ajoon tällä samalla tavalla.

Ajon aikana Dnapenny kertoo analyysin etenemisestä:

```
How many
trees looked                                                Approximate
at so far      Length of        How many                    percentage
(multiples     shortest tree    trees this short            searched
of  100):      found so far     found so far                so far
----------     ------------     -------------               ------------
      1            -                 0                        0.00
      2            -                 0                        0.00
      3            -                 0                        0.00
      4            -                 0                        0.00
      5            -                 0                        0.00
      6          228.0               1                        0.00
      7          228.0               1                        0.00
(jatkuu)
    626          225.0               1                       97.78
    627          225.0               1                       98.14
    628          225.0               1                       98.52
    629          225.0               1                       99.05
    630          225.0               1                       99.15

Output written to file "outfile"

Trees also written onto file "outtree"

28.38u 0.06s 4:18.26 11.0%
```

Viimeisen sarakkeen ilmoittama prosenttiluku saavutti 100%, ja ajo päättyi, sillä kaikki mahdolliset puut ehdittiin jo käydä läpi. Tämä takaa sen, että paras mahdollinen puu varmasti löytyi. Analyysi kesti 4 minuuttia 18 sekuntia. Heuristisen menetelmien nopeudesta kertonee jotakin, että sama analyysi ohjelmalla Dnapars, joka käyttää heuristista hakua, kesti alle sekunnin.

Dnapenny tuotti kaksi tulostiedosto, joista outfile on ihmissilmille tarkoitettu ja outtree tietokoneelle tarkoitettu. Tarkastellaan outfile-tiedoston sisältöä:

```
Penny algorithm for DNA, version 3.65
 branch-and-bound to find all most parsimonious trees


requires a total of            225.000

One most parsimonious tree found:


  +-----------------------------------tinami
  !
  !                                  +--rhea1
  1  +-------------------------------8
  !  !                               +--rhea2
  !  !
  !  !                               +--moa2
  +--4                          +----12
     !                          !     +--moa1
     !     +------------------7
     !     !                    !     +--moa4
     !     !                    +----10
     +-----2                          +--moa3
           !
           !            +--------------strutsi
           !            !
           +------------3              +--kasuaari
                        !  +---------9
                        !  !         +--emu
                        +--5
                           !         +--kiivi1
                           !      +-11
                           +------6  +--kiivi2
                                  !
                                  +-----kiivi3

  remember: (although rooted by outgroup) this is an unrooted tree!
```

Tiedostossa on merkkigrafiikalla piirretty puu, joka yhdistää kaikki sekvenssirinnastuksessa mainitut lajit. Puu pituus on 225 muutosta eli sekvensseissä on tapahtunut nukleotidimuutoksia yhteensä 225 kappaletta. Esimerkiksi artikkelia kirjoitettaessa tämä puun pituus on tärkeä tieto.

Jos haluaa piirtää artikkeliin kauniin kuvan saamastaan puusta, kannattanee käyttää vaikkapa Rod Pagen kirjoittamaa Treeview-ohjelmaa, jossa on varsin monipuoliset ominaisuudet puiden kuvantamiseen. Treeview osaa lukea outtree-tiedoston ja piirtää sen perusteella kuvan.

Saadun puun luotettavuutta voidaan vielä arvioida bootstrapping-menetelmällä. Aluksi ajetaan ohjelma seqboot, jolla muodostetaan uudelleenotannalla uusia rinnastuksia. Seqboot-ohjelmalle annetaan sama sekvenssirinnastustiedosto kuin yllä Dnapenny-ohjelmallekin. Oletusasetuksilla ohjelma muodostaa 100 uutta rinnastusta, mutta asetuksen muuttaminen onnistuu optiolla R. Lopuksi ohjelma vielä kysyy paritonta satunnaisnumeroa.

```
corona1 ~> seqboot
seqboot: can't find input file "infile"
Please enter a new file name> rat.phy

Bootstrapping algorithm, version 3.65

Settings for this run:
  D      Sequence, Morph, Rest., Gene Freqs?  Molecular sequences
  J  Bootstrap, Jackknife, Permute, Rewrite?  Bootstrap
  %    Regular or altered sampling fraction?  regular
  B      Block size for block-bootstrapping?  1 (regular bootstrap)
  R                    How many replicates?  100
  W              Read weights of characters?  No
  C                Read categories of sites?  No
  S     Write out data sets or just weights?  Data sets
  I             Input sequences interleaved?  Yes
  0      Terminal type (IBM PC, ANSI, none)?  ANSI
  1       Print out the data at start of run  No
  2     Print indications of progress of run  Yes

  Y to accept these or type the letter for one to change
Y
Random number seed (must be odd)?
13
```

Seqboot yrittää kirjoittaa automaattisesti tiedostoon outfile, mutta se on jo olemassa edellisen analyysin jäjiltä, joten ohjelma kysyy apua käyttäjältä:

```
seqboot: the file "outfile" that you wanted to
     use as output data file already exists.
     Do you want to Replace it, Append to it,
     write to a new File, or Quit?
     (please type R, A, F, or Q)
```

Tähän voi vastata vaikkapa R, mikä tarkoittaa, että tiedosto kirjoitetaan yli. Jos ei halua tehdä tätä, voi ohjelman suorituksen keskeyttää (Q) ja nimetä tuon edellisen analyysi tulostiedoston uudelleen. Vastaavanlainen kysymys voi tulla vastaan myös outtree:n osalta.

AJon jälkeen ohjelma kirjoittaa uudet sekvenssirinnastukset outfile-tiedostoon:

```
completed replicate number   10
completed replicate number   20
completed replicate number   30
completed replicate number   40
completed replicate number   50
completed replicate number   60
completed replicate number   70
completed replicate number   80
completed replicate number   90
completed replicate number  100

Output written to file "outfile"

Done.
```

Ennen seuraavaa analyysiä nimetään outfile infile:ksi (`mv outfile infile`). PHYLIP-ohjelmat nimittäin oletusarvoisesti lukevat infile-tiedostosta ja kirjoittavat outfile-tiedostoon. Analyysi menee kauhealla tavalla pieleen, jos yrittää lukea samaa tiedostoa kuin johon kirjoittaa. Siis, outfile-nimistä tiedostoa ei voi käyttää missään PHYLIP-ohjelmassa syötetiedostona.

Käynnistetään jälleen Dnapenny. Ohjelma ei nyt kysy syötetiedoston nimeä, sillä se lukee automaattisesti infile:n. Muutetaan ohjelman asetusta M (Analyze multiple datasets?). Tällöin ohjelma kysyy käytetäänkö aineistoja (D) vai painoarvoja (W), ja käytämme aineistoja. Lisäksi ohjelma kysyy aineistojen lukumäärää. Tämä luku on yhtäsuuri kuin yllä Seqboot-ohjelmassa arvottujen aineistojen lukumäärä (tässä 100). Asetuksen muuttamisen jälkeen hyväksytään muutos ja ajetaan ohjelma.

Ohjelma analysoi kunkin aineiston, ja kirjoittaa kutakin kohden vähintään yhden puu tiedostoihin outfile ja outtree. Ajon loputtua tiedostossa outtree on siis vähintäänkin 100 puuta. Näiden perusteella muodostetaan enemmistökompromissipuu ohjelmalla Consense. Ennen consense-ohjelman ajamista nimetään tiedosto outtree uudelleen nimellä intree. Consense-ohjelman valikko näyttää seuraavalta:

```
Consensus tree program, version 3.65

Settings for this run:
 C         Consensus type (MRe, strict, MR, Ml):  Majority rule (extended)
 O                              Outgroup root:  No, use species  1
 R                 Trees to be treated as Rooted:  No
 T           Terminal type (IBM PC, ANSI, none):  ANSI
 1                Print out the sets of species:  Yes
 2         Print indications of progress of run:  Yes
 3                              Print out tree:  Yes
 4                Write out trees onto tree file:  Yes

Are these settings correct? (type Y or the letter for one to change)
```

Muutetaan ulkoryhmän asetusta. Nyt käytettävä ulkoryhmä pitää lukea puutiedostosta. Tarkastellaan infile-tiedostoa, ja sen ensimmäistä puuta, joka päättyy puolipisteeseen. Ensimmäinen puu näyttää seuraavalta:

```
(((emu:0.03589,kasuaari:0.01856):0.02351,(strutsi:0.03507,
((tinami:0.10922,(rhea1:0.02475,rhea2:0.02104):0.02135):0.02351,
(moa2:0.00000,((moa4:0.00866,moa3:0.01856):0.00743,moa1:0.01856)
:0.00619):0.03496):0.02547):0.02021):0.02599,(kiivi2:0.00124,
kiivi1:0.00124):0.02475,kiivi3:0.02847);
```

Ulkoryhmäksi halutaan jälleen tinami, joka on vasemmalta oikealle, ylhäältä alas laskettuna neljäs laji puussa. Asetetaan siis ulkoryhmäksi numero 4, ja ajetaan ohjelma. Tuloksena on kaksi tiedostoa, outtree ja outfile. Tarkastellaan tiedostoa outfile:

```
CONSENSUS TREE:
the numbers on the branches indicate the number
of times the partition of the species into the two sets
which are separated by that branch occurred
among the trees, out of 100.00 trees

                                                +------moa2
                                        +-76.7-|
                                        |       +------moa1
                +---------------96.6-|
                |                       |       +------moa4
                |                       +-68.5-|
                |                               +------moa3
                |
         +-57.5-|                               +------emu
         |      |              +--------82.0-|
         |      |              |               +------kasuaari
         |      |       +-69.7-|
         |      |       |      |               +------kiivi1
         |      |       |      |       +-93.5-|
  +------|      +-83.0-|       +-95.5-|       +------kiivi2
  |      |              |              |
  |      |              |              +-------------kiivi3
  |      |              |
  |      |              +---------------------------strutsi
  |      |
  |      |                                       +------rhea2
  |      +-----------------------------97.7-|
  |                                              +------rhea1
  |
  +-----------------------------------------------tinami
```

Kussakin puun haarassa on esitetty lukuarvo, joka on nyt niiden puiden lukumäärä, joissa kyseinen ryhmä esiintyi. Esimerkiksi kiivi1 ja kiivi2 kulkivat yhdessä 93.5%:ssa puista. Kyseinen haara voitaneen siten tulkita varsin luotettavaksi.

On tavanomaista esittää bootstrapping-arvot alkuperäisen puun haaroissa. Parhaan tuloksen saa, kun muokkaa alkuperäistä puuta jossakin piirto-ohjelmassa. Piirto-ohjelmat eivät osaa lukea outtree:tä, mutta puun voi tallentaa jossakin kuvaformaatissa Treeview-ohjelmasta.

9.3.2 PHYLIP Tutkijan käyttöliittymässä

Osa PHYLIP-ohjelmista on saatavilla myös Tutkijan käyttöliittymässä. Dnapenny-ohjelmaa ei Tutkijan käyttöliittymästä löydy, mutta sitä vastaava heuristista hakua käyttävä Dnapars-ohjelma kylläkin. Tutkijan käyttöliittymässä ohjelmalle annetaan ensin sekvenssirinnastus, kuten UNIX:ssakin. Asetukset löytyvät rinnastuksen syötelaatikon alapuolelta. Bootstrapping-analyysi on yksi asetus muiden joukossa kunkin ohjelman kohdalla. Siten analyysin alkuperäisellä aineistolla ja bootstrapping:iä käyttäen voi suorittää täyttämällä asetukset vain yhden kerran.

Tutkijan käyttöliittymässä ohjelmia voi helposti putkittaa. Kun yhden PHYLIP-ohjelman tulos on valmistunut, voi tulossivulta valita seuraavan ohjelman, jolle tulostiedoston haluaa syöttää. Näin välttyy tiedostojen uudelleennimeämisen vaivalta.

Huonona puolena Tutkijan käyttöliittymässä on, että se mahdollistaa korkeintaan kahden tunnin mittaisten töiden ajamisen. Jos työ vaatii enemmän ajoaikaa, se on pakko suorittaa UNIX-puolella.

9.3.3 Muut ohjelmat

PHYLIP-ohjelmapaketti on moniin muihin ohjelmiin verrattuna suhteellisen helppokäyttöinen, mutta sen haittapuolena on, ettei sillä ole mahdollista analysoida tehokkaasti kovin suuria aineistoja. Heuristinen haku SPR-menetelmää käyttäen ei ole kovinkaan tehokasta, eikä sen avulla ole yleensä mahdollista löytää kaikkein parasta puuta vähänkään suuremmilla aineistoilla. Laskenta suurimman uskottavuuden menetelmää käyttäen on myös suhteellisen hidasta.

Jos aineisto on kovin suuri, jolloin koko aiheuttaa vaatimuksia hakualgoritmeille, on parasta käyttää jotakin muuta ohjelmaa. Suurille (> 50 lajia) parsimonia-menetelmällä analysoitaville aineistoille soveltuu parhaiten TNT, joka on erittäin nopea. Suurille suurimman uskottavuuden-menetelmällä analysoitaville aineistoille soveltuu erittäin hyvin RAxML, jossa on myös enemmän evoluutiomallivaihtoehtoja kuin PHYLIP:in Dnaml:ssä.

Näiden tehokkaampien ohjelmien käyttö on kuvattu tarkemmin oppaassa `http://www.csc.fi/english/research/sciences/bioscience/books/molsyst`

Liite

A Tärkeimmät CSC:llä ylläpidetyt tietokannat

Biotieteisiin ja sekvenssianalytiikkaan liittyviä tietokantoja löytyy internetistä valtavasti. Useimmat tietokannat ovat vapaasti käytettävissä ja niihin on tehty käyttöä helpottamaan erilaisia käyttöliittymiä. Usein tietokannan voi myös ladata omalle koneelle.

CSC ylläpitää palvelimillaan noin kahtakymmentä biotietokantaa, joista keskeisimmät on esitelty lyhyesti alla. Tietokantoja voi käyttää kuvauksessa mainittujen ohjelmien avulla.

A.1 Sekvenssitietokannat

A.1.1 EMBL

EBI:n (European Bioinformatics Insitute) ylläpitämä EMBL ja NCBI:n (National Center for Biotechnology Information) ylläpitämä GenBank ovat keskeisimpiä nukleotidisekvenssitietokantoja. EMBL ja GenBank vaihtavat tietoja paitsi toistensa, myös DDBJ:n (DNA DataBank of Japan) kanssa päivittäin, joten niiden sisältö on käytännöllisesti katsoen sama. Sekvenssit ovat peräisin suoraan tutkijoilta, sekvenointikeskuksista, julkaisuista sekä patenttivirastoista. Genomi- ja EST (expressed sequence tag) -sekvenointiprojektien myötä EMBL:n koko kaksinkertaistuu melkein vuosittain, ja vuoden 2006 lopussa sekvenssejä oli jo yli 83 miljoonaa. Näistä noin puolet oli EST-sekvenssejä. EMBL:stä julkaistaan uusi versio (engl. release) kolmen kuukauden välein, ja versioiden välillä kertyvät uudet sekvenssit tallennetaan päivittäin julkaistavaan update-osaan.

EMBL on kattava ja hyvin ajan tasalla oleva tietokanta, mutta se sisältää myös paljon päällekkäisyyttä ja virheitä. Havaittujen virheiden poistamista vaikeuttaa se, että tietojen korjaus- ja päivitysoikeus on vain sekvenssin tallettaneella tutkijalla. EMBL:n sekvenssit on jaettu 15 taksonomiseen alaosastoon (engl. division) ja 12 metodiseen luokkaan (eng. dataclass) alkuperänsä perusteella. Esim. EST- ja HTG (high-throughput genome) -sekvensseillä on omat luokkansa. EMBL on CSC:llä käytettävissä SRS:n kautta, ja se on myös linkitetty EMBOSS- ja BLAST-työkaluihin.

```
http://www.ebi.ac.uk/embl/
```

A.1.2 RefSeq

NCBI:n ylläpitämä referenssisekvenssikokoelma RefSeq perustettiin vaihtoehdoksi paljon päällekkäisyyttä ja virheitä sisältäville EMBL- ja GenBank-sekvenssivarastoille. RefSeq pyrkii tarjoamaan kattavan ja mahdollisimman vähän päällekkäisyyttä sisältävän kokoelman hyvälaatuisia ja asiantuntijoiden tarkistamia sekvenssejä keskeisimmistä organismeista. RefSeq sisältää mRNA- ja ncRNA-sekvenssien lisäksi myös genomi- ja proteiinisekvenssejä. Splice-variantit esitetään omina tietueinaan. Korkealaatuisten sekvenssien lisäksi RefSeq sisältää myös automaatisesti ennustettuja sekvenssejä, jotka on merkitty X-alkuisilla accession-numeroilla. Jokaisella sekvenssillä on comment-kenttään merkitty tarkistustaso (engl. status key), joista reviewed on paras. CSC:llä voi tehdä BLAST-hakuja RefSeq-tietokannan nukleotidisekvenssejä vastaan.

```
http://www.ncbi.nlm.nih.gov/RefSeq/
```

A.1.3 UniProt

UniProt Knowledgebase on EBI:n ja SIB:n (Swiss Institute of Bioinformatics) yhteistyössä ylläpitämä proteiinitietokanta, joka koostuu kahdesta osasta. SwissProt-osan tekee arvokkaaksi sen korkealuokkainen asiantuntija-annotaatio, vähäinen päällekkäisyys ja hyvä tietokantaintegraatio. Annotaatio sisältää tietoja proteiinin funktiosta, translaation jälkeisistä modifikaatioista, domeeneista, aktiivisista kohdista, rakenteesta, variaatioista sekä mahdollisista proteiiniin liittyvistä sairauksista. Seuraamalla linkkejä tietokantoihin on mahdollista hakea esim. proteiinia vastaava DNA-sekvenssi (EMBL), proteiinin 3D-rakenne (PDB), kuvaus proteiiniin liittyvistä geneettisistä sairauksista (OMIM) tai proteiiniperheelle ominainen sekvenssimotiivi (PROSITE). SwissProt on integroitu kaikkiaan 60 muun tietokannan kanssa. Osa SwissProtin sekvensseistä on suoraan tutkijoiden tallettamia, osa on peräisin julkaisuista ja osa on EMBL:n koodaavien sekvenssien translaatioita. Laadun varmistamiseksi kaikki sekvenssit tarkastetaan ja annotoidaan manuaalisesti ennen niiden tallettamista tietokantaan. Tietokannan ylläpitäjät myös korjaavat tietoja tarvittaessa.

UniProtin TrEMBL-osa sisältää kaikki EMBL:n koodaavien sekvenssien translaatiot sekä muut sekvenssit, jotka aiotaan sisällyttää SwissProt-osaan. Toisin kuin SwissProt, TrEMBL on annotoitu vain koneellisesti. Esim. arvio proteiinin mahdollisesta funktiosta, aktiivisista kohdista ja domeeneista perustuu InterPro-hakuihin. CSC:llä UniProt on käytettävissä EMBOSSin, BLASTin ja SRS:n kautta.

```
http://www.ebi.ac.uk/swissprot/
```

A.2 Tunnistetietokannat

Proteiinin rakenteelle ja/tai toiminnalle tärkeät sekvenssialueet muuttuvat hitaasti evoluution aikana. Proteiiniperheen konservoituneista alueista voidaan muodostaa eri menetelmin tunnisteita, joita voidaan hyödyntää uuden proteiinisekvenssin karakterisoinnissa. Tunnistetietokantahaut ovat sekvenssitietokantahakuja tehokkaampia erityisesti kaukaisten sukulaisten löytämisessä.

A.2.1 PROSITE

SIB:n (Swiss Institute of Bioinformatics) ylläpitämä PROSITE-tietokanta sisältää eri proteiiniperheille ja -domeeneille tunnusomaisia sekvenssimotiiveita. Niiden avulla voidaan määrittää, kuuluuko uusi proteiini johonkin tunnettuun proteiiniperheeseen tai onko siinä jotain tunnettuja domeeneja, mikä puolestaan voi edesauttaa proteiinin funktion selvittämisessä.

PROSITEn motiivit perustuvat monen sekvenssin rinnastuksista valittuihin n. 10-20 aminohapon pituisiin konservoituneisiin alueisiin. Niistä muodostetaan tunnistehahmo (eng. regular expression tai pattern), jonka diagnostinen teho testataan hauilla UniProt-tietokantaan. Hahmoilla on oma merkintätapansa, jossa hakasuluissa [] ilmoitetaan vaihtoehtoiset aminohapot, aaltosuluissa {} kielletyt aminohapot ja tavallisissa suluissa () aminohappojen lukumäärä.

Esimerkiksi merkintä [IL]-X(2,4)-P-C tarkoittaa sekvenssiä, jossa on ensin isoleusiini tai leusiini, kahdesta neljään mitä tahansa aminohappoa, proliini ja lopuksi joku muu aminhappo kuin kysteiini. PROSITE-tietokannan yhteen motiiviin perustuva tunnistus toimii vain riittävän konservoituneille proteiiniperheille ja -domeeneille. Tunnistustehon laajentamiseksi tietokantaan on lisätty myös profiileita, jotka ovat taulukkomuotoisia esityksiä aminohappojen ja aukkojen pisteistä sekvenssin eri kohdissa. Toisin kuin hahmot, profiilit yleensä kattavat proteiiniperheen tai domeenin koko sekvenssin pituudelta. PROSITE jakautuu kahteen alitietokantaan: PROSITE sisältää tunnisteet ja PROSITEDOC kuvauksen ko. proteiiniperheen tai domeenin rakenteesta ja funktiosta. EMBOSS-ohjelma patmatmotifs etsii proteiinisekvenssille osumia PROSITE-tietokannan hahmoista. PROSITE on CSC:llä käytettävissä myös SRS:n kautta.

```
http://www.expasy.org/prosite/
```

A.2.2 PRINTS

Toisin kun PROSITE, PRINTS-tietokanta ei käytä proteiiniperheiden karakterisointiin yksittäisiä sekvenssimotiiveja vaan kokonaisia motiivisarjoja (eng. fingerprints). Naapurimotiivien luoma konteksi lisää tunnisteen diagnostista luotettavuutta, mikä puolestaan helpottaa kaukaisempien sukulaisproteiinien tunnistamista.

PRINTS-tietokantaa ylläpitää Manchesterin yliopisto. Monen sekvenssin rinnastuksesta löytyneistä konservoituneista alueista muodostetaan frekvenssimatriisi, jolla tehdään haku UniProt-tietokantaan. Jos haussa löytyy uusia, kaikki sarjan motiivit sisältäviä sekvenssejä, nämä lisätään alkuperäiseen linjaukseen ja haku toistetaan. Iterointia jatketaan kunnes uusia sekvenssejä ei enää löydy. Sekä iteroinnissa käytetyt alkuperäiset motiivit että tuloksena saadut lopulliset motiivit viedään PRINTS-tietokantaan aukottomina paikallisina linjauksina. Tietokannassa kerrotaan lopullisen motiivisarjan ja sen yksiköiden löytämien sekvenssien lukumäärä tunnisteen diagnostisen tehon arvioimiseksi. EMBOSS-ohjelma pscan etsii proteiinisekvenssille osumia PRINTS-tietokannasta. Lisäksi tietokanta on CSC:llä käytettävissä SRS:n kautta.

`http://bioinf.man.ac.uk/dbbrowser/PRINTS/`

A.2.3 Pfam

Pfam sisältää proteiiniperheiden ja -domeenien sekvenssirinnastuksia ja niistä tehtyjä profiili-HMM:ja (hidden Markov model). Pfam tehdään UniProt-tietokannasta, ja yksi proteiini voi kuulua useampaan perheeseen. Pfam jakautuu kahteen osaan. Pienemmän PfamA-osan perheet perustuvat tarkistettuihin monen sekvenssin rinnastuksiin, kun taas kooltaan suuremman PfamB-osan perheet on luotu automaattisesti. PfamA-tietokannassa on alkuperäisen rinnastuksen ja siitä tehdyn HMM:n lisäksi automaattinen rinnastus kaikista HMM:n löytämistä sekvensseistä, sekä lyhyt kuvaus proteiiniperheestä. Pfam-tietokantaa voi CSC:llä käyttää SRS:n sekä EMBOSS-, HMMER- ja Wise2-ohjelmilla.

`http://www.sanger.ac.uk/Software/Pfam/`

A.2.4 InterPro

Vaikka eri menetelmiin perustuvat tunnistetietokannat ovat kooltaan melko samanlaisia, ne eivät ole päällekkäisiä. Jokaisella on oma optimaalinen käyttöalueensa, esim. funktionaaliset kohdat (PROSITE), domeenit (Pfam), perheiden hierarkkinen luokittelu (PRINTS) tai signaalivälitysdomeenit (SMART, Simple Modular Architecture Research Tool). Näiden tietokantojen vahvuudet on pyritty yhdistämään InterPro-tietokannan avulla. InterProssa samaa domeenia tai perhettä kuvaavat tunnisteet on yhdistetty yhdeksi tietueeksi. Jokainen tietue sisältää kuvauksen domeenin tai perheen funktiosta, annotaation, kirjallisuusviitteet, linkin alkuperäiseen tietokantaan sekä listan tunnisteen tunnistamista proteiineista. InterPro sisältää edellä mainittujen tietokantojen lisäksi myös ProDom-, TIGRFAM-, PIRSF, Superfamily, Gene3D- ja Panther-tietokannat, ja siihen integroidaan jatkuvasti lisää uusia jäseniä. CSC:llä InterPro on käytettävissä SRS:n kautta.

`http://www.ebi.ac.uk/interpro/`

A.3 PDB-rakennetietokanta

PDB (Protein Data Bank) on 1971 perustettu biologiten makromolekyylien rakennetietokanta, jota ylläpitää nykyisin RCSB (Research Collaboratory for Structural Bioinformatics) -konsortio. Siihen kuuluvat Rutgers University, San Diego Supercomputer Center ja National Institute of Standards and Technology. PDB sisältää proteiinien lisäksi nukleiinihappojen, proteiini-nukleiinihappokompleksien ja hiilihydraattien rakenteita. Suurin osa rakenteista on määritelty röntgenkristallografialla tai NMR:llä ja pieni osa muilla menetalmillä tai teoreettisesti. PDB:n rakenteet ovat tarkistettuja ja asiantuntijoiden annotoimia. PDB päivitetään kerran viikossa. PDB:tä voi käyttää CSC:ssä mm. SRS, BLAST, EMBOSS- ja InsightII-ohjelmien kautta. Rakenteita voi visualisoida mm. Discovery Studio, InsightII, Bodil, ja gOpenMol -ohjelmilla.

`http://www.rcsb.org/pdb/`

B Yleisimmistä sekvenssitiedostomuodoista

Oheisissa kappaleissa käsitellään yleisimmät ja tässä oppaassa mainitut sekvenssitiedostomuodot.

B.1 FASTA

FastA on yleinen tiedostomuoto, jota käyttävät mm. CSC:n BLAST-palvelin ja EMBOSS-ohjelmisto. FastA-tiedosto alkaa suurempi kuin -merkillä (>), jonka jälkeen samalla rivillä on kommentit. Ensimmäisen rivivaihdon jälkeen alkaa sekvenssi. Tiedoston lopussa ei ole lopetusmerkkiä.

```
> Kommentteja saa olla vain tällä rivillä
GAACATCCTTTTATCTGAGAACAATGTTGTGAAGATTTGCGACTTTGGCCTGGCCCGGGA
TATTTATAAGAACCCTGATTACGTGAGGAGAGGAGATACTCGACTTCCCCTAAAATGGAT
GGCTCCCGAATCCATCTTTGACAAGGTCTACAGCACCAAGAGT
```

FastA-muotoinen sekvenssitiedosto voi sisältää useita sekvenssejä. Tällöin sekvenssit ovat tiedostossa päällekkäin ja kahden sekvenssin erottimena toimii kommenttirivi

```
> Kommentteja saa olla vain tällä rivillä
GAACATCCTTTTATCTGAGAACAATGTTGTGAAGATTTGCGACTTTGGCCTGGCCCGGGA
TATTTATAAGAACCCTGATTACGTGAGGAGAGGAGATACTCGACTTCCCCTAAAATGGAT
GGCTCCCGAATCCATCTTTGACAAGGTCTACAGCACCAAGAGT
> seuraava sekvenssi
CACTAGCTTCAGCATGGGCTAGCGGATCCCGAGCTCGATCGGAGCTGGATGGCCCCAAAG
TGCGTACCGTACGAGGCGGATCGATTCGATCGATGCTAACCCGATCGGAGGCTGAGCTGA
CGATGAAGCTAGGAGCTAGGCGTTTATAAACCCATCTAGGATC
```

B.2 RAW/PLAIN

Raw- ja plain-tiedostomuodot sisältävät pelkän sekvenssin ilman aloitus- tai lopetusmerkkejä. Raw ei sisällä rivinvaihtoja eikä muotoiluja.

```
GAACATCCTTTTATCTGAGAACAATGTTGTGAAGATTTGCGACTTTGGCCTGGCCCGGGA
TATTTATAAGAACCCTGATTACGTGAGGAGAGGAGATACTCGACTTCCCCTAAAATGGAT
GGCTCCCGAATCCATCTTTGACAAGGTCTACAGCACCAAGAGT
```

B.3 SWISS-PROT, EMBL ja NCBI

SWISS-PROT-, EMBL- ja NCBI (esimerkiksi GenBank-)-tiedostomuodot sisältävät edellisiä tiedostomuotoja enemmän tietoa sekvenssistä. Näitä tiedostomuotoja käsitellään kunkin vuoden ensimmäisessä Nucleic Acid Research -lehden numerossa. Nämä tiedostomuodot koostuvat kentistä, joissa voi olla useita rivejä. Rivien alussa oleva tunnus kertoo rivin sisällön. EMBL:n ja SWISS-PROTin kenttien tunnus on kaksikirjaiminen, kun taas NCBI:n kenttätunnus on kokonainen sana. EMBL- ja GenBank-tiedostomuotojen ero esitetään alla lituruohon PR1-geeni -tietueen avulla. Ensimmäinen esimerkki on EMBL-tietokannasta. Tyhjät XX-rivit helpottavat tietueen selaamista. Ensimmäisellä rivillä on sekvenssin tunnus (ID = identification) ja tyyppi (DNA). Toisella rivillä on AC-hakukoodi, jonka pitäisi pysyä muuttumattomana sekvenssipäivityksissä, minkä takia sitä voi käyttää esimerkiksi SRS- tai GCG-haussa (`fetch em:m59196`). Päiväystietojen (DT) jälkeen kerrotaan eliö (OS), josta kyseinen sekvenssi on määritetty. OC-rivillä on taksonomista tietoa. Julkaisutietojen (RN, RP, RA, RT, RL) jälkeen viitataan muihin tietokantoihin, joista löytyy tietoja samasta geenistä. FT-kentissä karakterisoidaan sekvenssi, joka esitetään viimeisenä. Sekvenssin jälkeen tulee tiedoston lopetusmerkkirivi, jossa on vain kaksi vinoviivaa (//).

```
ID   ATPR1      standard; DNA; PLN; 1107 BP.
XX
AC   M59196;
XX
SV   M59196.1
XX
DT   06-JUL-1991 (Rel. 28, Created)
DT   04-MAR-2000 (Rel. 63, Last updated, Version 2)
XX
DE   Arabidopsis thaliana pathogenesis-related protein 1 (PR-1) gene,
DE   complete cds.
XX
KW   pathogenesis-related protein 1.
XX
OS   Arabidopsis thaliana (thale cress)
OC   Eukaryota; Viridiplantae; Streptophyta; Embryophyta; Tracheophyta;
OC   euphyllophytes; Spermatophyta; Magnoliophyta; eudicotyledons;
OC   core eudicots; Rosidae; eurosids II; Brassicales; Brassicaceae;
OC   Arabidopsis.
XX
RN   [1]
RP   1-1107
RA   Metzler M.C., Cutt J.R., Klessig D.F.;
RT   "Isolation and characterization of a gene encoding a PR-1-like protein from
RT   Arabidopsis thaliana";
RL   Plant Physiol. 96:346-348(1991).
XX
DR   Demeter; M59196; M59196.
DR   MENDEL; 6344; Arath;Ypr1;6344.
DR   SPTREMBL; Q39186; Q39186.
XX
FH   Key             Location/Qualifiers
FH
FT   source          1..1107
FT                   /db_xref="taxon:3702"
FT                   /organism="Arabidopsis thaliana"
FT                   /strain="Columbia"
FT                   /tissue_lib="genomic lambda EMBL3"
FT   CAAT_signal     242..255
FT                   /gene="PR-1"
```

```
FT   TATA_signal     301..310
FT                   /gene="PR-1"
FT   sig_peptide     380..502
FT                   /gene="PR-1"
FT   CDS             380..910
FT                   /codon_start=1
FT                   /db_xref="SPTREMBL:Q39186"
FT                   /gene="PR-1"
FT                   /product="pathogenesis-related protein 1"
FT                   /protein_id="AAA32841.1"
FT                   /translation="MNEMSFFGYSFIVVALFFDLTQAYRHTPAQPPKANANGDVKPQET
FT                   LVVHNKARAMVGVGPMVWNETLATYAQSYAHERARDCAMKHSLGPFGENLAAGWGTMSG
FT                   PVATEYWMTEKENYDYDSNTCGGDGVCGHYTQIVWRDSVRLGCASVRCKNDEYIWVICS
FT                   YDPPGNYIGQRPY"
FT   mat_peptide     503..907
FT                   /gene="PR-1"
FT                   /product="pathogenesis-related protein 1"
XX
SQ   Sequence 1107 BP; 350 A; 190 C; 233 G; 334 T; 0 other;
     TACGGTATCG ATAAAGCTTG GAGATTGAAA GTTGACGCTA AAACTGTAAT TTTGTATTGA        60
     GTTATGGTCT TATTGAATGA GATGTATTAT TGATAGTGAT ATTCTGTTTA TATACAAGAG       120
     TAATCGGAGG ACTAATTAGA TCAATTAGAT GAGTTTTCTT GATCTGTAGA ATATCAAGTT       180
     TCCTTATTCT ATCAATTTCT TGTCCTGGAT TTACACTATG TAAAGAGTTT TTCAACATTA       240
     GGAAAGTGAA AAGAATAAGA TATAAAACGT TCCAAACTCC GTAACCCCAC CAAAGCAACT       300
     TCTATAAATA CGTTAGCCAA GAAAATGCTT TCTTTTACGA CACAATAAGT TCCAAAACAA       360
     TTAACCACAC AAATACAAAA TGAATGAAAT GTCGTTCTTT GGTTATAGTT TCATCGTAGT       420
     AGCATTATTC TTCGATTTAA CTCAAGCCTA TCGTCACACT CCCGCTCAAC CGCCAAAAGC       480
     AAACGCAAAC GGTGATGTCA AACCGCAAGA AACGCTCGTG GTTCACAACA AGGCCCGAGC       540
     CATGGTCGGA GTCGGACCAA TGGTGTGGAA CGAAACTCTT GCGACCTATG CACAGAGCTA       600
     CGCACATGAA CGAGCCAGAG ACTGTGCCAT GAAGCATTCC TTGGGACCAT TCGGCGAGAA       660
     TCTAGCCGCG GGTTGGGGAA CGATGAGCGG TCCGGTAGCA ACTGAGTATT GGATGACGGA       720
     GAAGGAAAAT TACGATTATG ATAGTAACAC GTGTGGTGGT GATGGTGTGT GTGGACACTA       780
     CACTCAGATC GTGTGGCGTG ACTCGGTTCG ACTTGGTTGT GCCTCCGTGA GATGTAAGAA       840
     TGATGAGTAT ATTTGGGTGA TTTGTAGCTA TGATCCTCCG GGGAATTACA TCGGTCAACG       900
     TCCATATTAG TGATTGGATT TTATTATACT GAATCAACAA GATTACTATA TATGGTAATT       960
     AATGTGATTT AAGAAAAATT CATACTATAA CATTTTGTGT CCTTAAATGA TAAAAGATAT      1020
     ATAAATGCAC ACTAGTTTTT TTACCATCAA TGTGAGATAT GTTTCCATGA TATATCAAAG      1080
     TTGTTGATTT GATTATAATC GACTTCA                                          1107
//
```

Seuraavassa on sama sekvenssi GenBank-tiedostomuodossa. Siinä viitataan kenttiin kokonaisilla sanoilla. Sekvenssikuvauksen (`definition`) jälkeen ilmoitetaan hakukoodi (`accession`), avainsanat (`keywords`), eliö (`source`) ja taksonomista tietoa (`organism`). Lisäksi annetaan kirjallisuusviitteet. Kohdassa `features` selitetään sekvenssin rakennetta ja toimintaa. Kun tietue sisältää mRNA:ta, on tässä yhteydessä määritelty mahdollisesti tiedossa oleva proteiinia koodaava alue, sekä NCBI:n tietue, joka sisältää kyseisen proteiinin. Proteiinitietueet sisältävät viitteen proteiinia koodaavaa DNA:ta vastaavaan tietueeseen.

```
LOCUS       ATHPR1       1107 bp    DNA             PLN       27-APR-1993
DEFINITION  Arabidopsis thaliana pathogenesis-related protein 1 (PR-1) gene,
            complete cds.
ACCESSION   M59196
VERSION     M59196.1  GI:166804
KEYWORDS    pathogenesis-related protein 1.
SOURCE      Arabidopsis thaliana (strain Columbia) (library: genomic lambda
            EMBL3) DNA.
  ORGANISM    Arabidopsis thaliana
            Eukaryota; Viridiplantae; Streptophyta; Embryophyta; Tracheophyta;
            euphyllophytes; Spermatophyta; Magnoliophyta; eudicotyledons; core
            eudicots; Rosidae; eurosids II; Brassicales; Brassicaceae;
            Arabidopsis.
REFERENCE   1  (bases 1 to 1107)
  AUTHORS   Metzler,M.C., Cutt,J.R. and Klessig,D.F.
  TITLE     Isolation and characterization of a gene encoding a PR-1-like
```

```
            protein from Arabidopsis thaliana
  JOURNAL   Plant Physiol. 96, 346-348 (1991)
FEATURES             Location/Qualifiers
     source          1. .1107
                     /organism="Arabidopsis thaliana"
                     /strain="Columbia"
                     /db_xref="taxon:3702"
                     /tissue_lib="genomic lambda EMBL3"
     CAAT_signal     242. .255
                     /gene="PR-1"
     gene            242. .910
                     /gene="PR-1"
     TATA_signal     301. .310
                     /gene="PR-1"
     sig_peptide     380. .502
                     /gene="PR-1"
     CDS             380. .910
                     /gene="PR-1"
                     /codon_start=1
                     /product="pathogenesis-related protein 1"
                     /protein_id="AAA32841.1"
                     /db_xref="GI:166805"
                     /translation="MNEMSFFGYSFIVVALFFDLTQAYRHTPAQPPKANANGDVKPQE
                     TLVVHNKARAMVGVGPMVWNETLATYAQSYAHERARDCAMKHSLGPFGENLAAGWGTM
                     SGPVATEYWMTEKENYDYDSNTCGGDGVCGHYTQIVWRDSVRLGCASVRCKNDEYIWV
                     ICSYDPPGNYIGQRPY"
     mat_peptide     503. .907
                     /gene="PR-1"
                     /product="pathogenesis-related protein 1"
BASE COUNT      350 a    190 c    233 g    334 t
ORIGIN
 1 tacggtatcg ataaagcttg gagattgaaa gttgacgcta aaactgtaat tttgtattga
              61 gttatggtct tattgaatga gatgtattat tgatagtgat attctgttta tatacaagag
             121 taatcggagg actaattaga tcaattagat gagttttctt gatctgtaga atatcaagtt
             181 tccttattct atcaatttct tgtcctggat ttacactatg taaagagttt ttcaacatta
             241 ggaaagtgaa aagaataaga tataaaacgt tccaaactcc gtaaccccac caaagcaact
             301 tctataaata cgttagccaa gaaaatgctt tcttttacga cacaataagt tccaaaacaa
             361 ttaaccacac aaatacaaaa tgaatgaaat gtcgttcttt ggttatagtt tcatcgtagt
             421 agcattattc ttcgatttaa ctcaagccta tcgtcacact cccgctcaac cgccaaaagc
             481 aaacgcaaac ggtgatgtca aaccgcaaga aacgctcgtg gttcacaaca aggcccgagc
             541 catggtcgga gtcggaccaa tggtgtggaa cgaaactctt gcgacctatg cacagagcta
             601 cgcacatgaa cgagccagag actgtgccat gaagcattcc ttgggaccat tcggcgagaa
             661 tctagccgcg ggttggggaa cgatgagcgg tccggtagca actgagtatt ggatgacgga
             721 gaaggaaaat tacgattatg atagtaacac gtgtggtggt gatggtgtgt gtggacacta
             781 cactcagatc gtgtggcgtg actcggttcg acttggttgt gcctccgtga gatgtaagaa
             841 tgatgagtat atttgggtga tttgtagcta tgatcctccg gggaattaca tcggtcaacg
             901 tccatattag tgattggatt ttattatact gaatcaacaa gattactata tatggtaatt
             961 aatgtgattt aagaaaaatt catactataa cattttgtgt ccttaaatga taaaagatat
            1021 ataaatgcac actagttttt ttaccatcaa tgtgagatat gtttccatga tatatcaaag
            1081 ttgttgattt gattataatc gacttca
//
```

B.4 Clustal ohjelmien aln-muoto

ClustalW ja ClustalX-ohjelmien tekemät monen sekvenssin rinnastukset tallentuvat .aln-muotoisiksi monen sekvenssin tiedostoiksi. Myös EMBOSS-ohjelmat voivat käyttää aln-muotoa. Aln-muodossa sekvenssit on esitetty rinnakkain. Jokaisen sekvenssirivin alussa on sekvenssin nimi ja sen jälkeen sekvenssi. Myös aukkojen sijainti näkyy tiedostossa. Sekvenssien alapuolella voi lisäksi olla sarakkeessa olevien nukleotidien tai aminohappojen samankaltaisuutta kuvaava merkki.

```
CLUSTAL W (1.8) multiple sequence alignment


PRIO_GORGO      --MANLGCWMLVLFVATWSDLGLCKKRPKPGG-WNTGGSRYPGQGSPGGNRYPPQGG---
PRIO_HUMAN      --MANLGCWMLVLFVATWSDLGLCKKRPKPGG-WNTGGSRYPGQGSPGGNRYPPQGG---
PRIO_SAISC      --MANLGCWMLVLFVATWSDLGLCKKRPKPGG-WNTGGSRYPGQGSPGGNRYPPQGGGWG
PRIO_SHEEP      MVKSHIGSWILVLFVAMWSDVGLCKKRPKPGGGWNTGGSRYPGQGSPGGNRYPPQGG---
                   :::*.*:****** ***:*********** ************************

PRIO_GORGO      ----GGWGQPHGGGWGQPHGGGWGQPHGGGWGQPHGGG-WGQGGGTHSQWNKPSKPKTNM
PRIO_HUMAN      ----GGWGQPHGGGWGQPHGGGWGQPHGGGWGQPHGGG-WGQGGGTHSQWNKPSKPKTNM
PRIO_SAISC      QPHGGGWGQPHGGGWGQPHGGGWGQPHGGGWGQPHGGG-WGQGGGTHNQWNKPSKPKTNM
PRIO_SHEEP      ----GGWGQPHGGGWGQPHGGGWGQPHGGGWGQPHGGGGWGQGG-SHSQWNKPSKPKTNM
                    ********************************** ***** :*.************
```

B.5 MSF eli Multiple Sequence File format

MSF on GCG-ohjelmien monisekvenssinen tiedostomuoto. Tiedosto alkaa kommenteilla ja tiedolla siitä, millä ohjelmalla ja millä parametreilla (esimerkiksi aukkosakkoparametrit) tulos on saatu. Näiden jälkeen listataan sekvenssit, joista kerrotaan nimi (`Name`), pituus (`Len`), tarkistusluku (`Check`) ja paino (`weight`). Paino-parametri selitetään usean sekvenssin vertailun yhteydessä. Seuraavalla rivillä on vain kaksi vinoviivaa (`//`) , joiden jälkeen tulevat sekvenssit kymmenen merkin jaksoissa, 50 merkkiä rivillä. Sekvenssit on esitetty MSF-tiedostoissa samankaltaiset osat päällekkäin. Rivi alkaa sekvenssin nimellä. ~-merkki osoittaa sekvenssin olevan lyhyempi kuin muut sekvenssit. Pisteillä (.) kuvataan sekvensseihin linjauksen vuoksi sijoitettuja aukkoja. Rivin alussa on numero, joka ilmoittaa kyseisen rivin ensimmäisen nukleotidin tai aminohapon järjestysnumeron alkuperäisessä sekvenssissä. Esimerkki on phylip-kappaleessa esitelty ensimmäinen välivaihe betaglobiinisekvenssien linjauksessa.

```
!!NA_MULTIPLE_ALIGNMENT 1.0
PileUp of: @beta.list

 Symbol comparison table: GenRunData:pileupdna.cmp  CompCheck: 6876

                 GapWeight: 5
            GapLengthWeight: 1

 beta.msf  MSF: 444  Type: N  November 24, 1999 13:05  Check: 4782 ..

 Name: rabbit          Len:   444  Check:  316  Weight:  1.00
 Name: human           Len:   444  Check: 8021  Weight:  1.00
 Name: chicken         Len:   444  Check:  484  Weight:  1.00
 Name: opossum         Len:   444  Check: 1775  Weight:  1.00
 Name: frog            Len:   444  Check: 4186  Weight:  1.00

//

        1                                                   50
 rabbit  ATGGTGCATC TGTCCAGTGA GGAGAAGTCT GCGGTCACTG CCCTGTGGGG
  human  ATGGTGCACC TGACTCCTGA GGAGAAGTCT GCCGTTACTG CCCTGTGGGG
chicken  ATGGTGCACT GGACTGCTGA GGAGAAGCAG CTCATCACCG GCCTCTGGGG
opossum  ATGGTGCACT TGACTTCTGA GGAGAAGAAC TGCATCACTA CCATCTGGTC
   frog  ~~~ATGGGTT TGACAGCACA TGATCGTCAG CTGATCAACA GCACCTGGGG
```

B.6 Phylip-tiedostomuodot

Phylipissä on kaksi omaa tiedostomuotoa, joiden erona on sekvenssien esitysjärjestys. Esitysjärjestyksen voi muuttaa Phylipin sisällä. Jos sekvenssi on pidempi kuin yksi rivi, voidaan sekvenssit esittää rivi kerrallaan kustakin sekvenssistä. Tämä tiedostotyyppi on englanniksi *interleaved file format*. Rivien kymmenen ensimmäistä merkkiä on varattu nimelle: jos nimi on lyhyempi, jätetään ylijäävät kohdat tyhjiksi.

```
     5    444
rabbit     ATGGTGCATC TGTCCAGTGA GGAGAAGTCT GCGGTCACTG CCCTGTGGGG
human      ATGGTGCACC TGACTCCTGA GGAGAAGTCT GCCGTTACTG CCCTGTGGGG
chicken    ATGGTGCACT GGACTGCTGA GGAGAAGCAG CTCATCACCG GCCTCTGGGG
opossum    ATGGTGCACT TGACTTCTGA GGAGAAGAAC TGCATCACTA CCATCTGGTC
frog       ---ATGGGTT TGACAGCACA TGATCGTCAG CTGATCAACA GCACCTGGGG

           CAAGGTGAAT GTGGAAGAAG TTGGTGGTGA GGCCCTGGGC A-GCTGCTGG
           CAAGGTGAAC GTGGATGAAG TTGGTGGTGA GGCCCTGGGC A-GCTGCTGG
           CAAGGTCAAT GTGGCCGAAT GTGGGGCCGA AGCCCTGGCC AGGCTGCTGA
           TAAGGTGCAG GTTGACCAGA CTGGTGGTGA GGCCCTTGGC AGGATGCTCG
           CAAACTATGT GCCAAGACTA TTGGACAAGA GGCCCTTGGA CGTCTGCTGT
```

Sekvenssit voidaan esittää myös peräkkäin, jolloin sekvenssi esitetään kokonaan ennen seuraavaa sekvenssiä (*sequential file format*).

```
     5    444
rabbit     ATGGTGCATC TGTCCAGTGA GGAGAAGTCT GCGGTCACTG CCCTGTGGGG
           CAAGGTGAAT GTGGAAGAAG TTGGTGGTGA GGCCCTGGGC AGCTGCTGGT
human      ATGGTGCACC TGACTCCTGA GGAGAAGTCT GCCGTTACTG CCCTGTGGGG
           CAAGGTGAAC GTGGATGAAG TTGGTGGTGA GGCCCTGGGC AGCTGCTGGT
chicken    ATGGTGCACT GGACTGCTGA GGAGAAGCAG CTCATCACCG GCCTCTGGGG
           CAAGGTCAAT GTGGCCGAAT GTGGGGCCGA AGCCCTGGCC AGGCTGCTGA
opossum    ATGGTGCACT TGACTTCTGA GGAGAAGAAC TGCATCACTA CCATCTGGTC
           TAAGGTGCAG GTTGACCAGA CTGGTGGTGA GGCCCTTGGC AGGATGCTCG
frog       ATGGGTTTGA CAGCACATGA TCGTCAGCTG ATCAACAGCA CCTGGGGCAA
           ACTATGTGCC AAGACTATTG GACAAGAGGC CCTTGGACGT CTGCTGTGGA
```

Ennen sekvenssinimiä ja sekvenssejä tiedoston ensimmäisellä rivillä kerrotaan sekvenssien määrä sekä merkkien määrä sekvensseissä. Näiden jälkeen samalla rivillä kerrotaan mahdolliset ohjelmakohtaiset optiot (lisätietoa ohjelman ohjeistuksessa).

Hakemisto

(, 44
*, 26, 53, 54, 56
-aformat, 78
-auto, 59
-help, 54
-opt, 56
-ossingle, 58
-outseq, 56
-rformat, 59
-sask, 58
-sbegin, 58
-send, 58
-sequence, 56
-verbose, 58
.cma, 93
>, 26
@, 60
$ARCHIVE, 19, 33
$HOME, 19
$METAWRK, 20
, 46
, 27
|, 26
~, 27

aaltoviiva, 27
accession code, 88
affine gap penalty, 71
aln-tiedostomuoto, 133
argumentti, 21
arkistohakemisto, 19, 33
ASCII, 23
asis, 47
asteriski, 26
aukkosakot, 71
aukonlaajennussakko, 71
aukonmuodostussakko, 71

bg, 27
binääritiedostot, 23
BioView Viewer, 89
bl2seq, 92
BLAST, 83
    Bit score, 88
    hakutuyypit, 84
    kometorivikäyttö, 91
    odotusarvo, 88
    parametrit, 87, 91
    syöte, 85
    tietokannat, 87, 94
    word, 83
    www-käyttöliittymä, 85
BLASTn, 84
BLASTp, 84
blastpgp, 92
BLASTx, 84
BLOSUM, 70
BLOSUM-matriisit, 70
BLOSUM62, 69
bmp, 30

cd, 22
ClustalW, 98
ClustalX, 98
cp, 24
CSC:n palvelinympäristö, 18

datan arkistointi, 33
DDBJ, 125
division, 125
dotmatcher, 66
dotplot, 65
dottup, 65
dust, 87
dynaaminen optimointi, 72
dynamic programming, 72

E-arvo, 87
EBI, 37
EDNAMAT, 71
ehmmalign, 105
ehmmbuild, 105, 109
ehmmcalibrate, 109
emacs, 29
EMBASSY, 44, 63
EMBL, 125
EMBL-tiedostomuoto, 131
EMBOSS, 44
    alustus, 51
    HMMER, 108

komentorivikäyttö, 51
kuvat, 54
käyttöoppaat, 63
listatiedosto, 59
oletusarvot, 52
paramerit, 61
sekvenssirinastus, 72
tietokannat, 51
tulostiedostot, 50, 53, 78
valitsimet, 54, 58
WWW-käyttöliittymä, 47
X-käyttöliittymä, 61
embossdata, 61
EOF-rakenne, 32
eog, 30
erikoismerkit, 26
estwisedb, 106
Excel, 59
exit, 18

FASTA, 83
Fasta-tiedostomuoto, 52, 130
fasta-tiedostomuoto, 45
fg, 27

gap, 71
genewisedb, 106
getz, 42
ggv, 30
global alignment, 76
grep, 24
gzip, 34

hahmo, 84, 127
head, 24
hidden Markov models, 104
HMM, 104
HMM-haut, 106
HMM-profiili, 104

infoseq, 60
InterPro, 128

Jalview, 102
Jemboss, 61
jpg, 30
juurihakemisto, 18

Kätketyt Markovin mallit, 104
kattava rinnastus, 76
kenoviiva, 27
kill, 27
komennon täydennys, 26
komentotulkki, 25
konservoituneet alueet, 127
kotihakemisto, 19

less, 23
logoff, 18
ls, 21, 22
lukukehys, 54, 56

man, 25
matcher, 72
MegaBLAST, 92
MEME, 110
metatyöhakemisto, 19
mkdir, 22
molekyylisystematiikka, 112
monen sekvenssin rinnastus, 96
more, 23
motiivi, 110, 127
msf-tiedostomuoto, 134
multiple sequence alignment, 96
mv, 25

nano, 28
needle, 76
Needlemanin ja Wunchin algoritmi, 76

Ohjelmakohtaiset valitsimet, 54
oletusarvot, 52
Optional qualifiers, 54

paikallinen rinnastus, 72
pairwise progressive sequence alignment, 96
PAM-matriisit, 70
Parallel BLAST, 93
pattern, 127
pb, 91, 93
PDB, 129
PDF- tiedostot, 30
Pfam, 104, 128
PFP, 87
PHI-BLAST, 84, 92
Phylip-tiedostomuoto, 135
PID, 27
pisteytysmatriisi, 68
plotorf, 54
PNG, 54
png, 30
PostScript, 30
PostScript-kuvat, 54
PRINTS, 127
profiili, 111
ProfileSearch, 111
projektihakemisto, 20
prophecy, 111
PROSITE, 127
ps, 27
ps2pdf, 30
PSI-BLAST, 84, 92
putki, 26
PuTTY, 17

pwd, 22
pääteyhteys, 10, 16

qsub, 31
`qsub`, 33

rakennetietokanta, 129
RefSeq, 126
regular expression, 127
release, 125
rinnastus, 68
rivikomennot, 16
rm, 25
rmdir, 22

salasana, 15
Scientist's interface, 11
scoring matrix, 68
seaview, 101
SEG, 87
sekvenssialueen määrittely, 58
sekvenssien luku, 48
sekvenssihaut, 82
sekvenssimuodon muuttaminen, 47, 52
sekvenssiprofiili, 111
sekvenssirinnastus, 68
seqret, 52
showdb, 51
SMART, 128
Smithin ja Watermanin algoritmi, 72
SRS, 36
ssh, 16
stretcher, 76
suodatus, 87
syötetiedosto, 48

tail, 24
tar, 34
tausta-ajo, 27
TBLASTn, 84
TBLASTx, 84
tcsh-komentotulkki, 25
tekstitiedostot, 23
tfm, 52
tiedostomuodot, 45, 46
tiedostonimet, 23
tietokannat, 46
tiff, 30
tilde, 27
transeq, 55
translointi, 55
tunnistetietokannat, 127
Tutkijan käyttöliittymä, 11, 47

UniProt, 126
Unix, 16
Unix-hiiri, 18
USA-määrittely, 52
USA-merkintätapa, 45
use emboss, 51

valitsin, 21
verkkolevy, 20
Vienna, 44

water, 72
wget, 94
Wise2, 107
wossname, 52

X-emulaattori, 17
X-pääte, 17
X- pääteyhteys, 17

2006
ISBN 952–5520–21–8 (nid.)
ISBN 952–5520–22–6 (PDF)
http://www.csc.fi/