Tietojenkäsittelytiede **32** Heinäkuu 2011 sivut 10–15
Toimittaja: Jorma Tarhio

Algoritmitutkimuksen rooli bioinformatiikassa*

Veli Mäkinen
Helsingin yliopisto
Tietojenkäsittelytieteen laitos &
Tietotekniikan tutkimuslaitos HIIT
vmakinen@cs.helsinki.fi

Bioinformatiikka on monitieteinen tutkimusala, joka pyrkii ratkomaan biologian ja lääketieteen ongelmia laskennallisesta näkökulmasta. Monitieteisyys tulee ilmi eri lähestymistavoissa ongelmanasetteluun: Tilastotieteilijä on kiinnostunut, onko tutkittavalla geenin muunnoksella tilastollisesti merkittävä vaikutus syöpäalttiuteen. Biofyysikko on kiinnostunut geenimuunnoksen aiheuttamasta proteiinin rakenteen atomitason sidosten energiatasojen muutoksesta ja sen vaikutuksesta säätelymekanismeihin. Tietojenkäsittelijä on kiinnostunut, onko olemassa tehokas laskennallinen menetelmä laskea vaikkapa tilastollinen merkittävyys annetun, geenien ilmentymistä mittaavan datan perusteella tai oppia jokin geenien säätelymekanismia kuvaava malli. Sanomattakin lienee selvää, että bioinformatiikan kentällä pärjäisi parhaiten aikansa supertutkija, joka kykenee relevanttiin tutkimukseen monella eri tieteenalalla. Supertutkijoiden suhteellinen osuus kaikista tutkijoista on kuitenkin sen verran pieni, että toimiva tieteiden välinen yhteistyö lienee bioinformatiikan tutkimuksen kantava voima.

Tieteiden välisessä yhteistyössä ongelmana on tieteellisten paradigmojen kirjo. Kokeelliset tieteet vannovat hypoteesin testauksen nimeen: Monelleko datapisteelle hypoteesi toimii/ei toimi suhteessa koko dataan? Paras menetelmä on se, joka tukee parhaiten hypoteesia jonkin valitun mittarin mukaisesti. Eksaktit tieteet määrittelevät tarkasti ratkaistavan ongelman ja pyrkivät löytämään optimaalisen ratkaisun kyseiseen ongelmaan. Tarkasti määritelty ongelma on usein karkea yksinkertaistus biologisesta ilmiöstä, jolloin sen optimaalinen ratkaisu saattaa helposti hävitä hypoteesin testauksessa jollekin sitä huomattavasti yksinkertaisemmalle menetelmälle. Entä jos lisämittauksilla menetelmien järjestys vaihtuukin? Entä jos hypoteesintestauksen voittaa hybridimenetelmä, joka yhdistelee eri painokertoimilla aiemmin kehitettyjä menetelmiä? Näiden kysymyksien äärellä on hyvä muistaa, että lopullisena tavoitteena on ymmärtää jotain biologian osa-aluetta aiempaa paremmin. Hyvin toimiva yksinkertainen malli voi olla tässä suhteessa informatiivisempi kuin hieman paremmin toimiva monimutkainen malli — sanoo tähän yhteyteen sovitettu minimaalisuuden periaate (Occam's razor). Bioinformatii-

*Kirjoitus perustuu professorin juhlaluentoon 15.12.2010.

OHJE KIRJAPAINOLLE: B5-arkin vasen- ja yläreuna kohdistetaan A4-arkin vasempaan ja yläreunaan. Näin pitäisi marginaaliksi tulla taitteen puolella noin 33 mm ja muualla noin 22 mm.

kan julkaisufoorumit eivät mitenkään eksplisiittisesti mainosta, mitä ominaisuutta painotetaan artikkelien arviointiperusteina. Tämä johtaa jonkin verran rönsyilevään tutkimuskenttään, jossa eri tutkimusten keskinäistä paremmuutta on hankala evaluoida.

Oma taustani on algoritmitutkimuksessa, missä tavoitteena on kehittää mahdollisimman hyviä laskentamenetelmiä eli algoritmeja annettuihin ongelmiin. Tavallisimpia hyvän algoritmin kriteerejä ovat aika- ja tilatehokkuus. Julkaisukulttuuri alalla on selkeä: Jos kehität tunnettua paremman algoritmin tunnettuun ongelmaan tai tehokkaan algoritmin hyvin mielenkiintoiseen uuteen ongelmaan, julkaisukynnys yleensä ylittyy. Ongelman merkittävyys, ratkaisun eleganssi, tai ratkaisun tekninen vaikeus sitten määräävät, kuinka hyvällä foorumilla tulosta pääsee esittelemään. Nämä yhtenevät kriteerit luovat suhteellisen helposti hahmotettavan tutkimuskentän, kun on aina periaatteessa mahdollista löytää paras ratkaisu tiettyyn ongelmaan. Toinen asia on sitten rönsyily ongelma-avaruudessa; aina on olemassa pieni variantti tunnetusta ongelmasta, jota ei ole vielä ratkaistu. Näiden ongelmarönsyjen katkaisu ei ole aivan mutkatonta, koska silloin tällöin jonkun rönsyn päässä voi kasvaakin aivan uudenlainen kukka. Perustutkimukseksikin näitä rönsyjä voidaan kutsua, kun intuitiolla suunnataan tutkimusta sinne, mikä vain tuntuu tärkeältä.

Sovelluslähtöinen algoritmitutkimus on olennainen osa bioinformatiikkaa. Tällä alueella annetut ongelmat ovat oikeasti tarkkaan mietittyjä kuvauksia vaikkapa biologisesta ilmiöstä. Mahdollisimman aika- ja tilatehokkaan algoritmin kehittäminen ongelmaan on usein vasta lähtökohta tämäntyyppisessä tutkimuksessa. Mallia tarkennetaan kokeellisten tulosten perusteella ja uutta algoritmia kehitetään aiempaa tarkempaan ongelmanasetteluun. Koska menetelmien käytännön tehokkuus on oleellista sovellettavuuden vuoksi, teoreettisten aika- ja tilavaativuusanalyysien rinnalla mitataan myös kokeellista suorituskykyä — tässä yhteydessä puhutaan kokeellisesta algoritmitutkimuksesta (engl. algorithm engineering).

Klassinen sekvenssilinjaus on hyvä esimerkki sovelluslähtöisestä algoritmitutkimuksesta: Kahden DNA-jonon X ja Y (esitettynä ketjuina nukleotidiemäksiä `A`,`C`,`G`,`T`) sukulaisuutta voidaan arvioida sekvenssien samankaltaisuuden kautta; jos jono X voidaan muuttaa jonoksi Y pienellä määrällä evoluutiossa mahdollisesti tapahtuvia mutaatioita (esimerkiksi yhden nukleotidin vaihtuminen toiseksi, poistuminen jonosta, tai uuden nukleotidin lisäys keskelle jonoa), niin X ja Y ovat samankaltaisia. Toisin sanoen, niillä voi olla evoluutiossa yhteinen kantamuoto tai sitten samankaltaisuus on vain sattumaa. Esimerkiksi jonoilla $X =$ ACAGT ja $Y =$ CACTA pienintä määrää kuvatun laisia mutaatioita voidaan esittää sekvenssilinjauksena

```
ACAGT-
|  | |
-CACTA,
```

missä sarakkeet kuvaavat tarvittavia operaatioita; `A` vastaan `-` kuvaa nukleotidin `A` poistoa jonosta X, `G` vastaan `C` kuvaa nukleotidin `G` korvautumista nukleotidilla `C` jonossa X ja `-` vastaan `A` kuvaa nukleotidin `A` lisäystä jonoon X. Samankaltaisuuden selvittämiseen on olemassa tunnettu ratkaisu [16] (ks. kuva 1), jonka aikavaatimus on neliöllinen, eli luokkaa mn, missä m ja n ovat jonojen X ja Y pituudet. Käytännössä tämä yksinkertainen mallinnus-

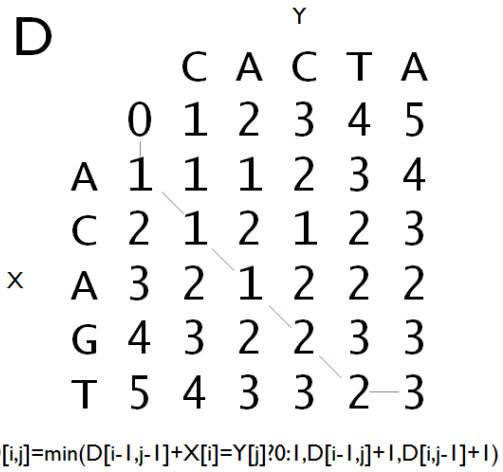
tapa ei ole riittävän tarkka kuvaamaan oikeita sukulaisuussuhteita. Mallista on kehitetty tarkennuksia, jotka kuvaavat paremmin biologista realiteettia, ja niille on kehitetty räätälöityjä algoritmeja [13, 15].

Koska samankaltaisuuden mittaaminen on niin keskeinen asia bioinformatiikassa, kokeellinen algoritmitutkimus on tuottanut toinen toistaan tehokkaampia käytännön ratkaisuja tähän ongelmaan. Tässä kohtaa paradigmojen sekoittuminen on kuitenkin vienyt voiton; parhaimmatkaan tarkat ratkaisut eivät ole riittävän tehokkaita etsittäessä esimerkiksi uuden sekvenssin samankaltaisuuksia yli kaiken nykyisin saatavilla olevan sekvenssidatan (jota on teratavuittain). On luovuttu tarkasta samankaltaisuuden laskennasta ja tyydytty heuristisiten mutta nopeiden menetelmien, kuten BLAST [1], implisiittisesti määrittämiin samankaltaisuusmittoihin, jotka perustuvat yhteisiin lähes tarkasti säilyneisiin lyhyisiin osajonoihin.

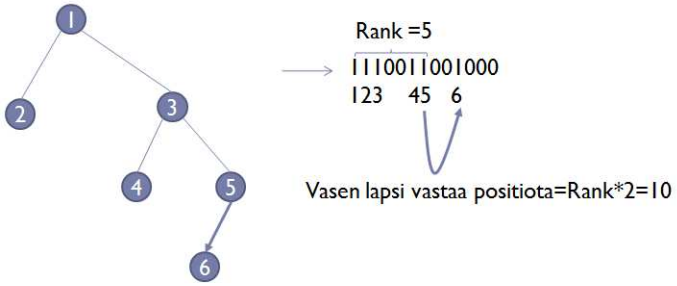
Eräs algoritmiperustutkimuksen rönsy on palauttamassa kokeellisen algoritmitutkimuksen keskeiseen rooliin samankaltaisuusvertailuissa. Täysin teoreettisesta mielenkiinnosta alkanut tiiviiden tietorakenteiden tutkimus (ks. esim. [2, 3, 4, 12] ja kuva 2) on viime vuosina löytänyt tärkeitä sovelluksia uusien sekvensointimenetelmien kehittyessä. Uudet sekvensointimenetelmät mahdollistavat yksilöiden genomien lukemisen pienissä pätkissä kustannustehokkaasti. Näitä pieniä pätkiä voidaan verrata lajin konsensusgenomiin ja paikantaa yksilön eroavaisuudet muusta populaatioista. Kyseessä oleva vertailu on helppo erikoistapaus yleisestä samankaltaisuusvertailusta, koska saman lajin sisällä syntyvä vaihtelu on kohtuullisen rajoitettua. Tähän erikoistapaukseen ja sen lukuisiin muunnelmiin tiiviiden tietorakenteiden päälle kehitetyt algoritmit [5, 6, 7, 8, 9] ovat osoittautuneet sekä tila- että aikavaativuuksiltaan varteenotettaviksi vaihtoehdoiksi heuristisimmille sekvenssilinjausmenetelmille.

Yksi lähitulevaisuuden keskeisiä haasteita bioinformatiikassa tulee olemaan alati kehittyvien sekvensointiteknologioiden täysimittainen hyödyntäminen (ks. esim. [11] ja kuva 3). Monet nykyisin hieman epäsuorasti tehtävät mittaukset voidaan uusilla sekvensointiteknologioilla tehdä tarkemmin ja tietokantojen sekvenssikuvaustiedot (engl. annotation) päivittää uuden tietämyksen valossa. Kuten tähänkin asti, algoritmitutkimuksen haasteena bioinformatiikassa on pysyä bioteknologian kehityksen tahdissa, sillä eilisen teknologialle kehitetty algoritmi voi jäädä nopeasti unhoon. Tässä auttaa kovasti yleistettävyyden periaate, eli kannattaa tutkia hieman laajempaa kokonaisuutta, kuin mitä käsillä olevan ongelman ratkaisu vaatii. Varmaan pahitteeksi ei ole myös miettiä tilannetta muutamaa vuotta eteenpäin; kuten mikä esitystapa olisi sopiva vaikka tuhannen ihmisen genomin tallentamiseksi niin, että sekvenssidatan sisältämä informaatio olisi mahdollisimman tehokkaasti käytettävissä? Yksilöiden genomien tapauksessa niiden sisältämä informaatio on erittäin redundanttia, koska samoja sekvenssipätkiä esiintyy useissa. Tiivis informaatiota hukkaamaton esitys mahdollistaa kyseisen kaltaisen datan kokonaisvaltaisen analyysin kuviteltavissa olevilla laskentaresursseilla; on muun muassa mahdollista mallintaa tuhannen yksilön genomidataa otteena populaatiosta ja tehdä sekvenssilinjausta vasten mitä tahansa mahdollista populaation rekombinaatiota [10, 14].

Algoritmitutkimuksen rooli ei ole mitenkään yksikäsitteinen bioinformatiikan kentässä. Joitain yksittäisiä esimerkkejä löytyy, joissa laajalti käytössä olevat bioinformatiikan käytännön menetel-



Kuva 1: Editointietäisyyden laskenta.

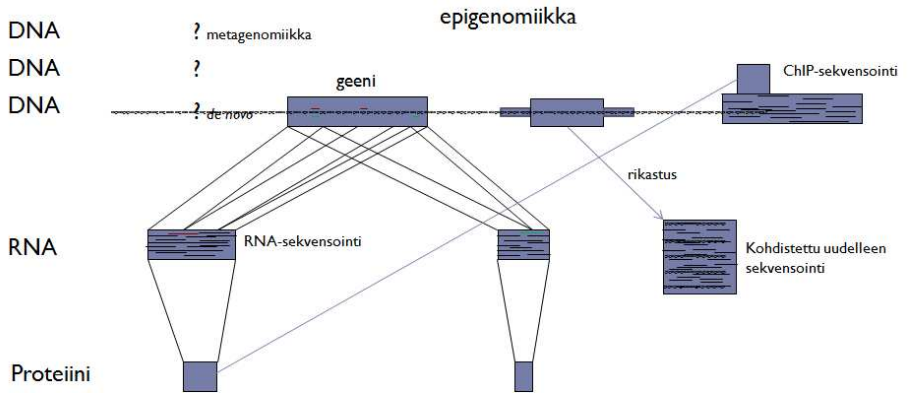


Kuva 2: Binääripuun esittäminen bittivektorilla.

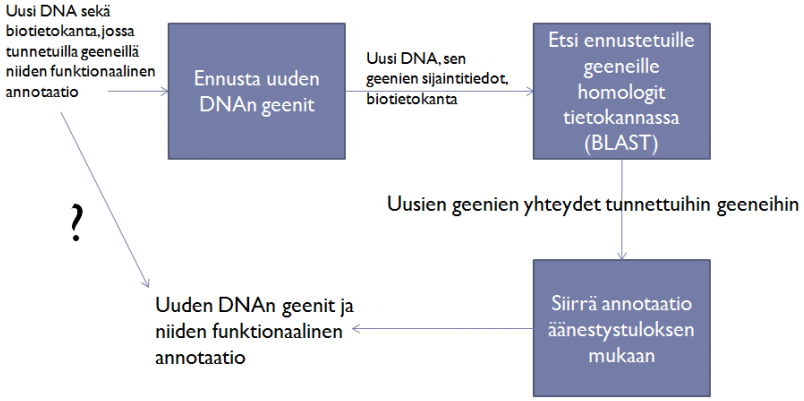
mät perustuvat täysin algoritmitutkimuksen periaattein kehitettyihin menetelmiin. Useimmiten näitä periaatteita noudattaen kehitettyjä menetelmiä käytetään kuitenkin vain osina ketjutettuja kokonaisuuksia (engl. pipeline), joissa eri osat toimivat riippumattomasti, lukien vain edellisen osan tulosteen omana syötteenään, kuten kuvassa 4 on havainnollistettu. Tämä on yksi tapa jäsentää vaikeita ongelmia helpommin ratkaistaviksi osiksi. Monet ketjutukset ovat saaneet standardin aseman bioinformatiikan menetelmäkehityksessä, jolloin niiden osien kehitykseen panostetaan valtavasti, mutta harvemmat huomaavat kyseenalaistaa koko ketjutuksen järkevyyttä. Tässä kohtaa koen, että algoritmitutkijoilla on vielä hyvin paljon annettavaa bioinformatiikan kentällä.

Viitteet

1. S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J. Mol. Biol.*, 215(3):403–410, 1990.
2. P. Ferragina and G. Manzini. Indexing compressed texts. *Journal of the ACM*, 52(4):552–581, 2005.
3. R. Grossi and J. Vitter. Compressed suffix arrays and suffix trees with applica-



Kuva 3: Sekvensointisovellusten atlas.



Kuva 4: Tyypillinen ketjutus bioinformatiikassa.

tions to text indexing and string matching. *SIAM Journal on Computing*, 35(2):378–407, 2006.

4. G. Jacobson. Space-efficient static trees and graphs. In *Proc. 30th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 549–554, 1989.
5. T. W. Lam, W. K. Sung, S. L. Tam, C. K. Wong, and S. M. Yiu. Compressed indexing and local alignment of dna. *Bioinformatics*, 24(6):791–797, 2008.
6. Ben Langmead, Cole Trapnell, Maihai Pop, and Steven L. Salzberg. Ultrafast and memory-efficient alignment of short dna sequences to the human genome. *Genome Biology*, 10(3):R25, 2009.
7. Heng Li and Richard Durbin. Fast and accurate short read alignment with burrows-wheeler transform. *Bioinformatics*, 27(14):1754–60, 2009.
8. Ruiqiang Li, Chang Yu, Yingrui Li, Tak-Wah Lam, Siu-Ming Yiu, Karsten Kristiansen, and Jun Wang. Soap2. *Bioinformatics*, 25(15):1966–1967, 2009.
9. V. Mäkinen et al. Unified view of backward backtracking in short read mapping. In *Algorithms and Applications*, volume 6060 of *LNCS*, pages 182–195. Springer, 2010.
10. V. Mäkinen, G. Navarro, J. Sirén, and Vä-

limäki N. Storage and retrieval of highly repetitive sequence collections. *Journal of Computational Biology*, 17(3):281–308, 2010.

11. M. L. Metzker. Sequencing technologies – the next generation. *Nature Reviews Genetics*, 11:31–46, 2010.
12. G. Navarro and V. Mäkinen. Compressed full-text indexes. *ACM Computing Surveys*, 39(1):article 2, 2007.
13. Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970.
14. Jouni Sirén, Niko Välimäki, and Veli Mäkinen. Indexing finite language representation of population genotypes. *CoRR*, abs/1010.2656, 2010.
15. Temple F. Smith and Michael S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.
16. Robert A. Wagner and Michael J. Fischer. The string-to-string correction problem. *Journal of the ACM*, 21(1):168–173, 1974.