Geenisäätelyverkkojen mallintaminen keskeytysmutanttien ja aikasarjojen ekspressiomittauksista

Kimmo Palin

Helsinki 27. tammikuuta 2003

Pro gradu –tutkielma

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Sisältö

Luku 1

Johdanto

1.1 Ongelman tausta ja motivaatio

Biologian tutkimuksessa on käytetty tietokoneita käytännössä niiden keksimisestä asti [Tur52]. Näiden vuosikymmenten aikana erilaisiin biologisiin sovelluksiin on kehitetty paljon erilaisia tietojenkäsittelymenetelmiä. Tyypillisiä biologiassa tarvittuja apuneuvoja ovat merkkijonomenetelmät, joita käytetään erilaisissa DNA–sekvensointiprojekteissa, kuten sekvenssoitaessa ihmisen genomia [MMH+01, VAM+01]. Merkkijonomenetelmät ovat keskeisiä myös bioinformatiikassa, joka tutkii biologisia tietokantoja [Bio03]. Eräs uusimpia tietojenkäsittelyn sovellusalueita biokemian alalla on geenien toiminnan, eli geenieksрression, analyysi [ESBB98]. Suurin geenieksрression ja sen säätelyn esittämä laskennallinen haaste on löytää geenisäätelyn verkkomaiset suhteet saatavilla olevasta vähäisestä ja kohinaisesta datasta [DWFS99].

Geenien toiminnan säätelyllä on merkittävä rooli tuotettaessa organismin *fenotyyppiä* eli ilmiasua. Geenieksрression säätely mahdollistaa organismin monimutkaisen toiminnan jo pienellä määrällä geenejä. Esimerkiksi ihmisellä, hyvin monimutkaisella monisoluisella organismilla, on vain noin 30 000 geeniä kun yksisoluisella leivontahiivalla niitä on jo noin 6 000 kappaletta.

Suhteellisen pieni määrä geenejä saa aikaan valtavan määrän erilaisia eläviä olentoja pienten bakteerien (Mycoplasma Genitalium 470 geeniä [FGW+95]) ja ihmisen (noin 30 000 geeniä [MMH+01, VAM+01]) väliltä. Jo yhden yksilönkin sisällä on suuria eroja solujen fenotyyppien suhteen. Esimerkiksi hermosolut ovat erilaisia kuin lihassolut tai verisolut. Koska kaikissa soluissa on sama DNA–sekvenssin koodaama perimä, fenotyyppien erot johtuvat geenien erilaisesta toiminnasta eri soluissa.

Tämän tutkielman tavoitteena on kehittää menetelmiä eräässä mielessä *kausaalisen geenisäätelymallin* oppimiseen kokeellisesta datasta. Kausaalisella mallilla [Pea01] tarkoitetaan mallia, joka ei ainoastaan kuvaa ekspressiomittaustulosten rakennetta, vaan antaa myös viitteitä nuo tulokset aikaan saaneista syy–seuraus–suhteista.

Kausaalisuudella tarkoitetaan tässä yhteydessä suhdetta, joka toimii myös ulkopuolisella väliintulolla. Tarkasteltaessa esimerkiksi jäätelön syönnin ja hukkumiskuolemien suhdetta tilastot osoittavat, että ihmisiä hukkuu useimmiten jäätelön menekin ollessa huipussaan Tämä suhde on tilastollinen korrelaatio. Kausaalisuus voitaisiin todeta vain jos järjestelmän ulkopuolelta pakotetaan jäätelön syöntiä ylöspäin ja tämän jälkeen havaittaisiin hukkumiskuolemien lisääntyvän. Hukkumiskuolemat eivät selvästikään ole seurausta jäätelön syönnistä mutta molemmat voivat olla seurausta vaikka vuorokauden keskilämpötilasta. Tämäkin kausaalinen hypoteesi on tosin vaikeasti testattavissa.

Tutkielman rakenne on seuraava. Aluksi luvuissa 1.2 ja 1.3 kuvataan tarvittavissa määrin ongelman biologista taustaa ja käytössä olevia koemenetelmiä. Luvussa 2.1 tarkastellaan menetelmää, jonka avulla voidaan teoriassa, olettaen säätelymallit yksinkertaisiksi, vertailla eri tavoilla saatujen geenisäätelyverkkojen sopivuutta ekspressiomittauksiin.

Tutkielman keskiosa lähestyy kausaalisesti perustellun geenisäätelyverkon oppimista kahdelta eri suunnalta, toisaalta verkkoa kasvattavasti ja toisaalta karsivasti. Luvussa 2.2 tarkastellaan tilannetta, jossa lähtökohtana on kaikki geenit ja kaikki kokeellisesti havaitut kausaaliset kaaret sisältävä verkko. Tätä verkkoa karsimalla pyritään

löytämään suppea, fyysisiä geenisäätelymekanismeja alkuperäistä paremmin kuvaava verkko.

Luvussa 2.3 geenisäätelyverkon oppimista lähestytään vastakkaiselta suunnalta lähtemällä tyhjästä verkosta ja lisäämällä siihen tarpeellinen määrä kaaria. Kaaret säilyttävät tietyt syy–seuraus–suhteet ja selittävät säädeltävän geenin toimintaa. Nämä kaksi lähestymistapaa, karsiva ja rakentava, tuottavat käsitteellisestikin erilaiset verkot.

Tutkielman lopussa, luvussa 3, kerrotaan edellä kuvattujen menetelmien toteutuksen ja testauksen yksityiskohdista sekä vertaillaan eri menetelmillä löydettyjä verkkoja toisiinsa. Viimeisenä luvussa 4 tarkastellaan tutkielmassa saatuja tuloksia ja tehtyjä päätelmiä.

Suurimmalta osaltaan tutkielma perustuu julkaistuihin menetelmiin ja niiden pieniin muunnoksiin. Omaa kontribuutiota ovat luvun 2.1.4 kaksoiskoodaava bayesverkko sekä kappaleen 2.2 min–max –verkko ja siihen liittyvät menetelmät. Myös algoritmin 4 muutokset ja luvussa 3 esitelty toteutus ovat uutta kontribuutiota.

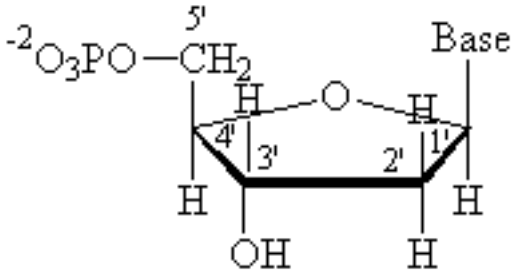
1.2 DNA, geeni ja ekspressio

Jokaisen elollisen olennon jokainen solu sisältää tiedon kyseisen yksilön perimästä. Tämä tieto on *aitotumallisilla* (eukaryote) organismeilla tallennettuna solun ytimessä eli *tumassa* sijaitseviin pitkiin *DNA* polymeereihin (DeoxyriboNucleicAcid), joita kutsutaan *kromosomeiksi*. Tumattomilla organismeilla (prokaryote) kromosomit kelluvat vapaina *solulimassa*, eli *sytoplasmassa* (cytoplasm). Tällaisia organismeja ovat "tavalliset" eubakteerit (eubacterium) ja arkeobakteerit (archaea) [The03]. Muut yksisoluiset ja kaikki monisoluiset organismit ovat aitotumallisia.

Perimän sisältävä DNA–polymeeri rakentuu toisiinsa liittyneistä *nukleotideistä* (nucleotide). Jokainen nukleotidi rakentuu kolmesta osasta, deoksiriboosi–sokeriryhmästä (deoxyribose), fosfaattiryhmästä sekä emäksestä (base). Nukleotidissä oleva emäs on

yksi neljästä perimässä esiintyvästä emäksestä: adenosiini, sytosiini, guaniini tai tymiini (katso esim. [Far, BPSS]).

Yksilön perimä määräytyy sen DNA:ssa olevien, eliölajista riippuen miljoonien tai miljardien nukleotidien järjestyksestä. Tämä järjestys voidaan lukea laboratoriomenetelmillä ja kirjoittaa jonona nukleotideja kuvaavia A–, C–, G– ja T–kirjaimia. Nukleotidien sokeriryhmän hiiliatomit numeroidaan vakiintuneeseen tapaan kuten kuvassa 1.1. Numeroinnin mukaan emäs sitoutuu nukleotidin $1'$–hiileen ja nukleotidi sitoutuu $5'$ hiilellä fosfodiesterisidoksella sitä edeltävän nukleotidin $3'$ hiileen. Näin ollen DNA–sekvenssin suunta on hyvin määritelty ja vakiintunut käytäntö on ilmoittaa DNA–sekvenssit alkaen siitä päästä, jossa on vapaa $5'$–atomi ja päättyen siihen päähän jossa on vapaa $3'$–atomi. DNA–sekvenssien yhteydessä käytetään usein yksikköä emäspari (bp, base pair) kuvaamaan nukleotideista muodostuneen merkkijonon pituutta.



Kuva 1.1: Deoksyribonukleotidin rakenne.

Normaaleissa olosuhteissa DNA–polymeerit ovat kolmiulotteiselta rakenteeltaan kaksoiskierteitä [WC53]. Rakenne muodostuu kun nukleotidien emäkset muodostavat keskenään vetysidoksilla niin sanottuja Watson–Crick pareja. Watson–Crick pareja ovat A ja T sekä C ja G, eli DNA–sekvenssi $5'$-ACGT-$3'$ pariutuu sekvenssin $3'$-TGCA-$5'$ kanssa. Tällä tavalla pariutunut rakenne puolestaan taipuu kierteelle kuin korkkiruuvi.

Olion DNA–sekvenssissä on joitakin kohtia, jotka ajoittain *transkriptoituvat* lähetti–RNA molekyyliksi (messenger RiboNucleicAcid, mRNA). Näitä transkriptoituvia DNA:n osia sanotaan geeneiksi. Transkriptiota sanotaan myös geenin *ekspressioksi*

ja transkription voimakkuutta geenin *ekspressiotasoksi*. mRNA on DNA:n kaltainen polymeeri, jossa deoksyriboosin sijasta sokeriryhmänä on riboosi. Näiden sokerien ero on riboosin $2'$ hiileen liittynyt OH–ryhmä, jota deoksyriboosissa ei ole. Lisäksi RNA eroaa DNA:sta siten että DNA:n tymiiniemäksen tilalla RNA:ssa esiintyy urasiili, jota merkitään kirjaimella U. Transkriptiossa muodostunut mRNA–molekyyli siirtyy tuman ulkopuolelle solulimaan ja sieltä ribosomiin, jossa siitä muodostuu (*translate*) proteiini.

Proteiinit ovat solun kemiallisia työkaluja. Ne ovat DNA:n ja RNA:n tavoin polymeerejä mutta rakentuvat nukleotidien sijaan aminohapoista. Erilaisia aminohappoja on 20 erilaista. Yhdessä proteiinissa niitä on muutamasta kymmenestä muutamaan tuhanteen.

Proteiinit hoitavat lähes kaikki solun tehtävät. Ne muun muassa vastaavat solun rakenteesta ja katalysoivat solussa tapahtuvia kemiallisia reaktioita. Osa proteiineista on niin sanottuja *transkriptiofaktoreita* (transcripton factor), jotka säätelevät geenien transkriptiota. Fyysisesti transkription säätely tapahtuu siten, että tietyt transkriptiofaktorit sitoutuvat säädeltävän geenin *säätelyalueella* (promoter region) oleviin *sitoumapisteisiin* (binding site) ja katalysoivat yhteistoiminnassa transkription alkua.

Säätelyalue on geenin lähistöllä DNA–sekvenssissä oleva alue, joka on merkittävä geenin toiminnan säätelylle. Säätelyalueen sijainti ei ole erityisen hyvin määritelty eikä sen löytäminen ole käytännössä kovin helppoa. Korkeammilla organismeilla, kuten nisäkkäillä, geenien säätelyalueet voivat sijaita useiden tuhansien emäsparien päässä itse säädeltävästä geenistä. Yksinkertaisilla eliöillä, kuten leivontahiivalla, säätelyalueiden uskotaan löytyvän varsin läheltä geeniä ja vielä tiettyyn suuntaan geenistä. Nykykäsityksen mukaan tutkittaessa hiivan geenien säätelyalueita riittää keskittyä vain noin 200–700 emäsparin alueelle geenistä DNA–sekvenssin alkuun päin [ZZ99].

Säätelyalueilla sijaitsevat sitoumapisteet ovat tyypillisesti lyhyitä, alle 20 emäsparin pituisia DNA–merkkijonohahmoja. Kullekin transkriptiofaktorille tai usean trans-

kriptiofaktorin muodostamalle säätelykompleksille on olemassa oma erityinen DNA–hahmo, johon se pyrkii sitoutumaan. Muut proteiinit eivät yleensä sitoudu samaan hahmoon. Samankaltaiseen sitoumapisteeseen sitoutuvat transkriptiofaktorit ovat ainakin osittain samankaltaisia, jolloin niiden DNA:han tarttuvat osat eivät eroa toisistaan merkittävästi.

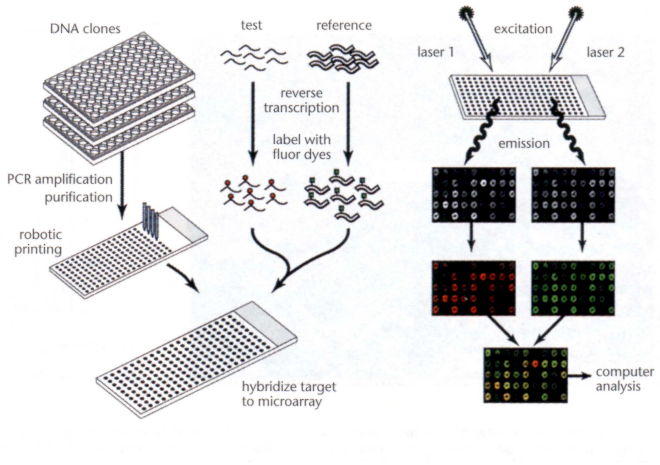
Näitä kahta transkription säätelyyn liittyvää osaa, DNA:ssa olevaa nukleotidihahmoa ja siihen sitoutuvaa proteiinia, kutsutaan joskus *cis-* ja *trans-* säätelytekijöiksi. Sanat cis ja trans ovat latinan prepositioita ja tarkoittavat "tällä puolella" ja "toisella puolella". Sanojen merkitys selkenee kun transkription säätelyä katsoo geenin näkökulmasta: Geeni on DNA:ta ja cis–aktiiviset säätelytekijät ovat kiinteästi sen lähellä mutta trans-aktiiviset proteiinit voivat olla hyvinkin kaukana geenistä ja ne ovat erilaisia kuin DNA.

Cis– ja trans– säätelykomponenttien merkittävin ero on niiden pysyvyydessä. Cis–aktiiviset säätelytekijät ovat pysyvä osa eliön DNA:ta ja ne säilyvät samanlaisina koko organismin eliniän ympäristöstään riippumatta. Trans–aktiiviset säätelytekijät puolestaan tulevat ja menevät riippuen niitä tuottavien geenien toiminnasta ja soluun ulkopuolelta tulevista signaaleista. Genomin cis–aktiiviset elementit määrittelevät ikään kuin pohjimmaisen säätelyohjelman ja trans–aktiiviset säätelytekijät ovat tuon ohjelman syötteitä ja tulosteita. Kumpikaan komponentti ei kuitenkaan pysty yksin aiheuttamaan säätelyä.

1.2.1 Ekspression mittaus

Tuhansien geenien mRNA–ekspressiotasot voidaan mitata yhtäaikaisesti niin sanotulla mikrosirulla (microarray). Mikrosirun toiminta on esitetty kuvassa 1.2 [Exp02]. Siru valmistetaan kiinnittämällä siihen osia tarkastelun kohteena olevien geenien DNA:sta (vas. ylhäällä). Jokaiselle geenille yksilöllinen DNA–jakso kiinnitetään tiettyyn paikkaan lasilevyä. Kuvan 1.2 keskikohdalla on kaksi eri oloissa kasvatettua näytettä, joiden geenien suhteellisista ekspressiotasoista ollaan kiinnostuneita. Kummastakin näytteestä kerätään lähetti–RNA:ta, josta tehdään komplementaarinen

DNA (complementary DNA, cDNA) käänteistranskriptiolla (reverse transcription). Saatuihin DNA–näytteisiin kiinnitetään väriaineet; toiseen näytteeseen punainen, toiseen vihreä. Tämän jälkeen näytteet sekoitetaan keskenään ja niiden annetaan pariutua mikrosirulla olevien vastaavien DNA–jaksojen kanssa.



Kuva 1.2: Mikrosirulla tapahtuvan ekspressiomittauksen prosessi.

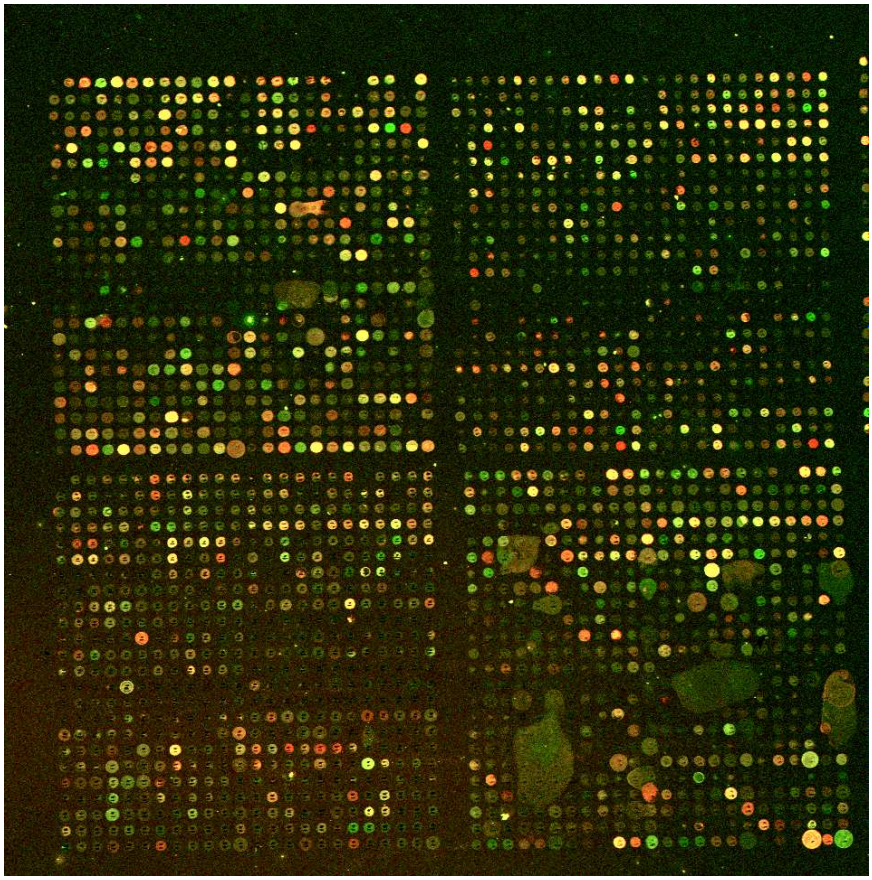
Kun näytteet on sitoutettu mikrosirulle, siru pestään ja siitä otetaan kaksi valokuvaa laser–valossa (kuvassa oikealla). Kuvat otetaan käyttämällä toisessa kuvassa punaista ja toisessa vihreää laseria. Nämä yksiväriset kuvat voidaan yhdistää, jolloin saadaan kuvan 1.3 [AED+00] kaltainen tulos.

Kuvassa 1.3 näkyvät pisteet vastaavat sirulle kiinnitettyjä DNA–jaksoja. Pisteiden väri– ja kirkkausvaihtelut johtuvat käytetyissä näytteissä olleista ekspressiotasojen eroista: tietyn geenin piste näyttää punaiselta, mikäli geeni oli aktiivisempi punaisella värjätyssä näytteessä, ja vastaavasti vihreältä, mikäli se oli aktiivisempi vihreällä värjätyssä näytteessä.

Mikrosirusta otetut kuvat ovat yleensä varsin epäselviä, eikä ekspressiosuhteiden eroja ole helppoa laskea kaikille geeneille kuvan kirkkausarvoista. Mikrosirujen kuvankäsittely on itsessäänkin hyvin aktiivisen tutkimuksen alainen tieteenala [YBS01]. Tässä tutkielmassa oletetaan ekspressiotasot annetuiksi yksikäsitteisinä reaalilukuina.

Mikrosirun nerokkuus on ekspressiomittausten miniatyrisoinnissa ja automatisoin-

nissa. Yhdellä neliön muotoisella ja noin tuuman levyisellä mikrosirulasilla voidaan tehdä tuhansia ekspressiomittauksia kerrallaan. Yksi levyllä oleva mRNA–näyte on kooltaan vain luokkaa 0.1 mm oleva piste. Biologien käyttämää mikrosirua voikin hyvin verrata tietokoneen mikrosiruun, jonka pieni koko mahdollisti tietokoneiden nopeutumisen ja yleistymisen.



Kuva 1.3: Osa mikrosirulta luettua kuvaa.

Eräs mikrosirutekniikan sisäinen ominaisuus on sen antamien tulosten suhteellisuus. Mitattujen intensiteettien absoluuttiset arvot ovat liian kohinaisia, jotta niistä voitaisiin tehdä johtopäätöksiä geenin ekspressiosta. Tästä johtuen mikrosirumittausten tulokset ilmoitetaan aina eri aallonpituuksilla havaittujen intensiteettien suhteena. Suhteen tarkastelu vähentää mittausarvojen kohinaa mutta samalla aiheuttaa sen, että kaksi mikrosirukoetta ovat vertailukelpoisia vain, jos kontrollinäyte on molemmissa kokeissa sama.

Vielä intensiteettien suhteuttamisen jälkeenkin mikrosirumittaukset ovat hyvin kohinaisia. Edes niin yksinkertainen kysymys kuin "Onko tämän geenin ekspressiotaso poikkeava näissä koeolosuhteissa suhteessa kontrollinäytteeseen?" ei ole yksinkertaisesti ja yleisesti hyväksytyin menetelmin vastattavissa. Tyypillisesti poikkeavasti ekspressoituvien geenien valinta tehdään jollain mielivaltaisella kynnysarvolla, esimerkiksi yli kaksinkertaisen muutoksen voidaan sanoa olevan merkitsevä.

Yksinkertainen kynnysarvotus jättää huomiotta geenin yksilöllisen ja luonnollisen ekspressiovaihtelun. Tämän luonnollisen vaihtelun johdosta geenin ekspressiota pitäisi tarkastella pikemminkin satunnaismuuttujana kuin deterministisenä funktiona säätelijöiltä ekspressiotasoille.

Mikäli geenin ekspressiotasoa normaaliolosuhteissa kuvaavan satunnaismuuttujan X jakauma tunnetaan, voidaan tilastollisesti tunnistaa kokeessa ekspressiotaan muuttanut geeni. Muuttujan X jakauman tunnuslukuja, kuten odotusarvoa μ ja keskihajontaa σ voidaan arvioida kokeellisesti tekemällä sopivia mikrosirumittauksia. Normaaliolosuhteiden vaihtelua voidaan mitata kokeilla, joissa molemmat näytteet on kasvatettu toisistaan riippumattomasti mahdollisimman samoissa olosuhteissa. Kun tällaisia saman suhde samaan –mittauksia on tarpeeksi jokaiselle geenille, geenikohtaiset μ ja σ voidaan arvioida datasta.

Kokeeseen todennäköisesti reagoinut geeni voidaan tunnistaa mittaustuloksesta, joka poikkeaa odotusarvostaan vähintään k keskihajontaa. Tälle raja–arvotukselle saadaan pätevä virhetodennäköisyys Chebyshevin epäyhtälöstä,

$$\mathrm{P}(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}. \tag{1.1}$$

Tämän epäyhtälön perusteella voidaan esimerkiksi sanoa, että yli kahden keskihajonnan muutos tapahtuu vain 25% todennäköisyydellä mikäli koetilanteella ei ollut vaikutusta kyseisen geenin ekspressioon. Mikäli muuttuneita geenejä on sirulla enemmän kuin neljännes, voidaan päätellä kokeella olleen jotain vaikutusta genomin transkriptiotoimintaan. Valitettavasti Chebyshevin raja on usein hyvin karkea ja yksittäisen geenin kannalta hyödylliset raja-arvot vaativat joitain vahvoja oletuksia tarkasteltavan muuttujan jakaumasta.

1.3 Geenisäätelyverkot

Jokaisen proteiinin koodaa jokin geeni. Tästä seuraa että jokaista transkriptiofaktoria koodaa jokin geeni. Koska transkriptiofaktorit puolestaan säätelevät geenien toimintaa, voidaan geenien toisiinsa kohdistamat vaikutukset kuvata verkkona, *geenisäätelyverkkona* tai *geeniverkkona* (Gene regulatory network, Genetic Network [CD01]).

Määritelmä 1 *Suunnattu verkko $\mathcal{G}$ on pari (V, E), jossa V on solmujen joukko ja $E = \{(a, b)|a, b \in V\}$ on verkon kaarien joukko. Graafisessa esityksessä verkon solmut kuvataan nimettyinä ympyröinä ja kaari $(a, b) \in E$ kuvataan nuolena $a \rightarrow b$.*

Määritelmä 2 *Geenisäätelyverkko on verkko $\mathcal{G}$, jossa solmut vastaavat tarkasteltavan organismin geenejä ja kaaret geenien välisiä toiminnallisia suhteita.*

Geenisäätelyverkossa on solmu jokaiselle geenille ja geenistä A on kaari geeniin B, mikäli A:n toiminta vaikuttaa suoraan geenin B:n toimintaan. Kaarten määräytymisen yksityiskohdat riippuvat paljolti tarjolla olevasta datasta ja halutusta sovelluskohteesta. Geenisäätelyn toiminnalliseen kuvaamiseen usein käytettyjä malleja ovat muiden muassa Boolen funktiot [SL98], lineaariyhtälöt [DWFS99], differentiaaliyhtälöt [CHC99], bayesverkot [FLNP00, SBS+02], dynaamiset bayesverkot [MM99, FMR98] ja sigmoidifunktiot [WWS99].

Kaikkien näiden mallien, ja samalla koko geenisäätelyilmiön, ongelma on vaadittavien ekspressiomittausten suuri määrä. Nykyään tarjolla olevat tietokannat eivät sisällä läheskään riittävää määrää informaatiota, jotta yllä mainituilla verkkomalleilla saataisiin tyydyttäviä tuloksia. Boolen funktioille (verkoille) on todistettu [AKMM98] että verkon päättelyyn tarvittavien kokeiden määrä on välillä $\Omega(n^D)$ ja $O(n^{2D})$, missä n on geenien määrä ja D on yläraja yksittäisen geenin säätelijöille. Esimerkiksi leivontahiivalle nämä luvut on arvioitu olevan noin $n = 6000$ ja $D = 5$,

joten täydellisten geenisäätelyverkkojen havaitseminen datasta on laskennallisesti hyvin raskasta.

Käytännössä Boolen funktioita rajoitetummilla malleilla on saavutettu jonkin asteista menestystä. Esimerkiksi D'Haeseleer [DWFS99] ilmoitti löytäneensä merkittäviä säätelysuhteita hyvin pienellä datamäärällä käyttämällä lineaarista regressiota.

Lineaarinen regressio ja differentiaaliyhtälöt ovat luonnollinen tapa mallintaa monia nykyisin saatavilla olevia ekspressiodatajoukkoja, sillä monet niistä ovat aikasarjoja. Aikasarjana toteutettu mikrosirukoe käyttää useita mikrosiruja, joilla mitattavat näytteet on otettu samasta näytekasvustosta eri ajanhetkinä. Esimerkiksi solun jakautumiseen vaikuttavia geenejä voidaan etsiä tekemällä useita ekspressiomittauksia usean jakautumissyklin ajalta ja etsiä geenit joiden ekspressiotasot käyttäytyvät samankaltaisesti joka syklissä [SSZ+98].

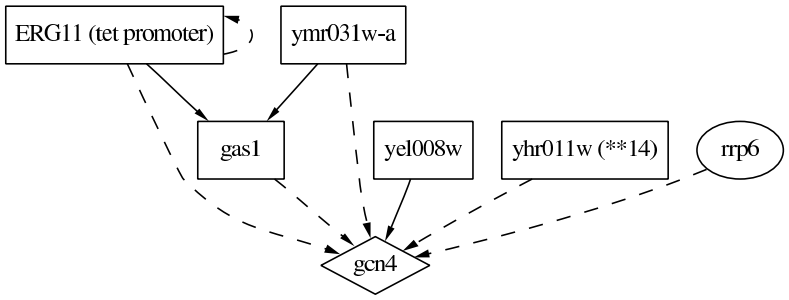
1.3.1 Geenin keskeytys ja keskeytysverkot

Biologit pystyvät laboratoriossa keskeyttämään (disrupt) organismin tietyn geenin toiminnan [WSA+99] ja, mikäli organismi jää eloon, mittaamaan muiden geenien ekspressiotasot keskeytyksen jälkeen [HMJ+00]. Kun tiedetään geenien normaalit ekspressiotasot olosuhteissa, joissa yhtään geeniä ei ole keskeytetty, geenin keskeytystä seuraavalla ekspressiomittauksella voidaan löytää ne geenit, joihin kyseisen geenin keskeytys vaikuttaa. Normaalitilanteen ekspressiotasojen tietäminen on tärkeää satunnaiskohinan vaientamiseksi, sillä pelkkä mikrosirulta saatavien ekspressioiden kynnysarvottaminen on monissa tapauksissa liian virhealtista.

Tällaisista *keskeytyskokeista* voidaan rakentaa hieman geenisäätelyverkkoa muistuttava *keskeytysverkko*.

Määritelmä 3 *Keskeytysverkko on verkko $\mathcal{K}$, jossa solmut vastaavat tarkasteltavan organismin geenejä ja verkossa on kaari $a \rightarrow b$ mikäli geenin a keskeyttäminen muuttaa geenin b ekspressiota.*

Keskeytysverkko ei vastaa määritelmän 2 mukaista geenisäätelyverkkoa, sillä kaari A:sta B:hen syntyy myös, mikäli A säätelee C:tä ja C säätelee B:tä [Wag02]. Näiden *transitiivisesti sulkevien* kaarien poisto keskeytysverkosta on yksi tämän tutkielman pääteemoista.



Kuva 1.4: Leivontahiivan aminohapposynteesiä säätelevän geenin gcn4 säätelijät keskeytysverkossa. Katkoviivalla piirretyt kaaret kuvaavat ekspression nousua keskeytyskokeessa ja kiinteät kaaret kuvaavat laskua.

Esimerkiksi kuvan 1.4 keskeytysverkko (kokeet: [HMJ+00]) näyttää siltä, että geenit ERG11 ja ymr031w-a eivät säätele suoraan gcn4–geeniä vaan säätelyvaikutus kulkee geenin gas1 kautta. Tämä aidosta keskeytysdatasta piirretty keskeytysverkko kertoo myös datan huonosta laadusta: ilmeisesti mittavirheen ansiosta ERG11–geenin keskeytys näyttää lisäävän sen itsensä ekspressiota.

1.3.2 Säätelyn cis– ja trans–komponentit

Tässä tutkielmassa on tarkoitus yhdistää mitattu ekspressiodata DNA–sekvenssistä saatuun informaatioon. Tavoitteena on luoda malli geenisäätelyverkosta, joka kuvaa mahdollisimman hyvin solun sisällä tapahtuvia fyysisiä säätelymekanismeja.

Useimmat aiemmin ehdotetut geenisäätelymallit pyrkivät vain mallintamaan ekspressiomittauksista saatua informaatiota eivätkä ne ota kantaa solun sisällä tapahtuviin fyysisiin mekanismeihin. Fyysisiä cis–trans mekanismeja mallintamalla voidaan tavoitella jollain tasolla “todellisen” geenisäätelyverkon kuvausta.

Cis–trans säätelyn mallinnuksessa pyritään mallintamaan sekä solussa toimivat trans-

kriptiofaktoriproteiinit (trans) että geenien säätelyalueilla DNA:ssa olevat sitoumapisteet (cis). Geenisäätelyn mallintamisessa tärkeä osa on ekspressiomittauksilla, joissa normaalia (wild type) näytettä on verrattu keskeytysmutanttiin (null mutant) eli näytteeseen, jossa yhden geenin toiminta on keskeytetty. Tällä tavalla saadaan tietoa mitä yhden trans–säätelijän puuttuminen vaikuttaa muihin geeneihin.

Säätelyn cis–komponentin päättelyssä tarkastellaan säädeltävien geenien säätelyalueita. Jos samalla tavalla ekspressoituvien geenien säätelyalueilta löytyy jollain tavalla tyypillinen nukleotidihahmo, tätä hahmoa voidaan pitää sitoumapisteenä ja näin lisätä uskottavuutta niille ekspressiokokeiden perusteella päätellyille säätelysuhteille, joiden kohteilla on tuo sitoumapiste. Vastaavasti, mikäli mikrosiru kertoo geenin ekspression muuttuneen mutta geenillä ei ole sitoumapistettä, voidaan kyseisen ekspressiomuutoksen uskoa johtuvan jostain toissijaisesta säätelysuhteesta. Toissijaisella säätelysuhteella tarkoitetaan tässä esimerkiksi jonkin kolmannen geenin välittämää vaikutusta.

Geenisäätelyverkkojen mallintamista yleisemmin cis–trans käsitteitä on laskennallisessa biologiassa käytetty uusien cis–sitoumapisteiden löytämiseen. Mahdollisia sitoumapisteitä haetaan niiden geenien yläalueilta, joiden ekspressiotasot on mitattu samankaltaisiksi [BJVU98, ESBB98, VBJ$^+$00, BLS01, JCCS01, GLCV01, HB00]. Joskus ekspressiodataa on tutkittu myös toisinpäin etsimällä ensiksi mahdollisia sitoumapistehahmoja ja laskemalla jokin samankaltaisuusmitta niiden geenien ekspressioille, joilla on kyseinen hahmo säätelyalueellaan [BBS01, CBE01]. Jälkimmäistä menetelmää on käytetty myös tutkittaessa sitoumapisteiden yhteistoimintaa [PSC01].

Viimeaikoina muutamat tutkimukset ovat kohdistuneet myös säätelyn cis– ja trans–komponenttien yhtäaikaiseen käsittelyyn. Näissä tutkimuksissa geenit klusteroidaan sekvenssin ja ekspressiotasojen mukaan klustereihin [HB00] tai datasta opetellaan bayesverkko [SBS$^+$02], josta voidaan myös päätellä geenisäätelyverkko. Ehdotettu geenisäätelyverkon tuottava menetelmä [SBS$^+$02] opettelee kerralla erittäin suuren määrän parametrejä ja on laskennallisesti varsin raskas.

Luku 2

Menetelmät

2.1 Löydettyjen geenisäätelyverkkojen yhteensopivuus havaintoihin

Luvuissa 2.2 ja 2.3 esitetään kaksi menetelmää geenisäätelyverkkokandidaattien löytämiseen ekspressiomittauksista ja säätelyalueiden sekvensseistä saadulla informaatiolla. Menetelmät tuottavat helposti kandidaatteja geenisäätelyverkoiksi mutta kandidaattien hyvyyden arviointi on hankalaa. Luotuja verkkoja pitäisi jollain tapaa verrata tuntemattomaan "oikeaan" geenisäätelyverkkoon.

Menetelmiä voi luonnollisesti vertailla käyttämällä keinotekoisesti luotuja testitapauksia, mutta tällaiset vertailut riippuvat välttämättä datan tuottamisessa käytetyn mallin oletuksista eivätkä niinkään luonnossa esiintyvien geenisäätelyverkkojen ominaisuuksista. Jotta generoidun datan tuomat ongelmat voidaan välttää, täytyy suunnitella jokin muu keino antaa hyvyysluku löydetyille geenisäätelyverkkokandidaateille. Tähän tarkoitukseen voidaan käyttää informaatioteoriassa käytettävää *kuvauspituuden* (description length) käsitettä.

Ennen syventymistä kuvauspituuteen ja geenisäätelyverkkojen hyvyysmittaan, luvussa 2.1.1 esitellään hieman satunnaismuuttujan epävarmuutta mittaavan entro-

pian käsitettä. Entropian lisäksi luvussa 2.1.1 esitellään myös yhteinen informaatio, jota käytetään kappaleessa 2.2.2 rakennettaessa geenisäätelyverkkoehdokkaita.

2.1.1 Satunnaismuuttujan entropia

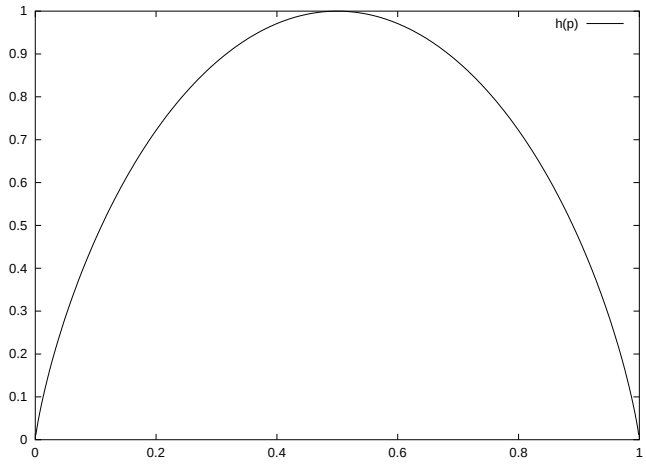
Satunnaismuuttujan X epävarmuutta mitataan *entropialla* $\mathrm{H}(X)$ [Sha48]. Entropia on sitä suurempi mitä suurempi epävarmuus kyseisen satunnaismuuttujan arvoon liittyy. Samankaltainen tulkinta voidaan antaa myös fysiikan entropiakäsitteelle: järjestelmän entropia on sitä suurempi mitä tasaisemmin järjestelmän energia on jakautunut. Informaatioteoreettinen entropia diskreetille satunnaismuutujalle X, joka voi saada arvot x, lasketaan kaavalla

$$\mathrm{H}(X) = -\sum_{x} \mathrm{P}(X = x) \log \mathrm{P}(X = x). \tag{2.1}$$

Mikäli kaavassa (2.1) esiintyvä todennäköisyys on 0, kaavassa esiintyvää logaritmia ei voida laskea. Tässä tapauksessa summan termi määritellään jatkuvasti saamaan arvon 0. Kaksiarvoisen satunnaismuuttujan X tapauksessa entropian kaava yksinkertaistuu binäärientropiaksi $\mathrm{h}(p) = -p \log p - (1 - p) \log(1 - p)$, jossa p on todennäköisyys $p = \mathrm{P}(X = 1)$. Binäärientropian käyttäytyminen näkyy kuvassa 2.1. Kuvasta huomataan entropian ominaisuus, jonka mukaan ennalta määrätyllä satunnaismuuttujalla on pieni entropia ja tasaisesti jakautuneella suuri. Kuvasta 2.1 nähdään, että binäärisen satunnaismuuttujan entropia on 0, mikäli se saa aina arvon 1 (tai arvon 0). Binäärientropia saavuttaa maksiminsa 1 kun molemmat lopputulokset ovat yhtä todennäköisiä.

Kahdelle satunnaismuuttujalle X ja Y käytetään merkintää $\mathrm{H}(X|Y)$ kuvaamaan muuttujan X *ehdollista entropiaa* annettuna Y. Ehdollinen entropia on X:n entropian odotusarvo yli Y:n jakauman. Toisin sanoen muuttujan X entropia annettuna Y on

$$\mathrm{H}(X|Y) = -\sum_{y} \mathrm{P}(y) \sum_{x} \mathrm{P}(x|y) \log \mathrm{P}(x|y) = -\sum_{x,y} \mathrm{P}(x, y) \log \mathrm{P}(x|y). \tag{2.2}$$



Kuva 2.1: Binäärientropia $h(p)$

Kahden muuttujan *yhteinen entropia* määritellään satunnaismuuttujan (X, Y) entropiaksi. Tällöin voidaan laskea

$$\mathrm{H}(X,Y) = -\sum_{x,y} \mathrm{P}(x,y) \log \mathrm{P}(x,y). \tag{2.3}$$

Soveltamalla Bayesin kaavaa saadaan yhtälö (2.3) jaettua yksinkertaisempiin tekijöihin

$$\begin{aligned} \mathrm{H}(X,Y) &= -\textstyle\sum_{x,y} \mathrm{P}(x)\mathrm{P}(y|x) \log \mathrm{P}(x)\mathrm{P}(y|x) \\ &= -\textstyle\sum_{x} \Big(\mathrm{P}(x) \log \mathrm{P}(x) - \mathrm{P}(x) \sum_{y} \mathrm{P}(y|x) \log \mathrm{P}(y|x) \Big) \\ &= \mathrm{H}(X) + \mathrm{H}(Y|X) \end{aligned} \tag{2.4}$$

Yhtälöistä (2.2) ja (2.4) huomataan, että riippumattomien satunnaismuuttujien X ja Y yhteinen entropia on niiden entropioiden summa $\mathrm{H}(X,Y) = \mathrm{H}(X) + \mathrm{H}(Y)$. Muussa tapauksessa yhteinen entropia on muuttujien yhteisen informaation verran summaa pienempi.

Kahden satunnaismuuttujan X ja Y *yhteinen informaatio* $\mathrm{I}(X,Y)$ (mutual information) on sen epävarmuuden määrä, jonka toisen muuttujan tunteminen vähentää toisen entropiasta. Yhteinen informaatio mittaa nimensä mukaisesti sitä epävarmuutta, joka kahdelle satunnaismuuttujalle on yhteistä. Yhteinen informaatio voidaan laskea kaavalla

$$\mathrm{I}(X,Y) = \mathrm{H}(X) - \mathrm{H}(X|Y) = \mathrm{H}(X) + \mathrm{H}(Y) - \mathrm{H}(X,Y). \tag{2.5}$$

Yhteistä informaatiota tullaan käyttämään hyväksi luvussa 2.2.2 määriteltäessä hinnoittelua keskeytysverkosta karsittaville kaarille. Entropia puolestaan on alaraja satunnaismuuttujan kuvauspituuden odotusarvolle.

2.1.2 Kuvauspituus ja MDL

Kuvauspituus [Sha48] (description length, code length) on informaatioteorian (katso [CT91]) peruskäsitteitä. Informaatioteoriassa satunnaismuuttujan *koodauksella* tarkoitetaan sen arvon esittämistä bittijonona. Koodaus voidaan ymmärtää ykkösten ja nollien muodostamana jonona, jota vastaa yksikäsitteinen X:n arvo x. Kohinaton koodauslause [Sha48] (Noiseless coding theorem) sanoo, että satunnaismuuttujan X koodaus on odotusarvoiselta pituudeltaan vähintään $\mathrm{H}(X)$ bittiä, jossa entropia lasketaan 2 kantaisella logaritmilla. Logaritmin kantaluvulla ei ole suurta merkitystä, sillä muuttujan X odotusarvoinen kuvauspituus binäärikoodilla on aina $\mathrm{H}(X)/\log 2$. Merkitään satunnaismuuttujan X arvon x koodisanan pituutta $\mathrm{DL}(x)$. Shannon todisti myös, että *optimaalisessa* koodauksessa x:n kuvauspituus on sen *logaritmisen uskottavuuden* (log likelihood) vastaluku $\mathrm{DL}(x) = -\log \mathrm{P}(x)$.

Esimerkkinä satunnaismuuttujan koodauksesta olkoon satunnaismuuttuja A, jolla $\mathrm{P}(A = a) = \frac{1}{256}$ kaikilla $a = 0, \ldots, 255$. Tämän muuttujan entropia

$$\mathrm{H}(A) = -\sum_{a=0}^{255} \frac{1}{256} \log \frac{1}{256} = 8, \tag{2.6}$$

joten keskimäärin tarvitaan vähintään 8 bittiä koodaamaan välille $[0, 255]$ tasaisesti jakautuneet kokonaisluvut. Lisäksi jokaiselle luvulle voidaan laskea optimaalinen koodinpituus $\mathrm{DL}(a) = -\log \frac{1}{256} = 8$. Tällaiset koodinpituudet saavuttava koodaus on esimerkiksi binääriluku,

$$b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0 = \sum_{i=0}^{7} b_i 2^i, \tag{2.7}$$

jossa b_i ovat bittejä 1 tai 0. Kohinattoman koodauslauseen mukaan on mahdotonta löytää näille luvuille ja tälle jakaumalle koodia, joka keskimäärin tuottaisi lyhyemmän bittiesityksen.

Optimaalisen koodauksen suunnittelu onnistuu tehokkaasti mekaanisilla algoritmeilla *universaalisti*, eli mille tahansa satunnaismuuttujalle. Ehkä tunnetuimpia universaaleja optimaalisia koodeja ovat Huffman koodaus [Huf52], Lempel-Ziv koodaus [ZL77] ja aritmeettinen koodaus [Ris76]. Alkujaan Shannon kehitti ideansa entropiasta, informaatiosta ja koodeista toisen maailmansodan aikana suunnitellessaan viestiyhteyksiä Yhdysvaltain ja Britannian välille. Nykyään koodauslause antaa esimerkiksi alarajan tiedon pakkaukselle tietokoneissa.

Syy miksi kuvauspituus on kiinnostava geenisäätelyverkkojen yhteydessä on niin sanottu *minimikuvauspituus periaate* [Ris86, Ris78] (Minimum description length, MDL). Minimikuvauspituusperiaate (minimikuvausperiaate) on eräässä mielessä matemaattinen muotoilu Occamin partaveitsestä [Occ24]. MDL–periaatteen mukaan tiettyä ilmiötä kuvaamaan ehdotetuista malleista $\mathcal{M}$ paras on se, jonka kuvauspituus yhdessä mallinnettavan datan D kanssa $\mathrm{DL}(D, \mathcal{M})$ on lyhin

$$\mathrm{DL}(D, \mathcal{M}) = -\log \mathrm{P}(D|\mathcal{M}) - \log \mathrm{P}(\mathcal{M}) = \mathrm{DL}(D|\mathcal{M}) + \mathrm{DL}(\mathcal{M}). \tag{2.8}$$

Tässä tapauksessa kuvauspituuteen otetaan huomioon sekä malli että mallinnettava data. Hyvä ominaisuus mallin ja datan yhdistävässä menetelmässä on sen jakautuminen osiin. Yhtälön (2.8) ja MDL–periaatteen mukaan hyvä malli geenisäätelyverkolle on sellainen joka antaa saaduille mittausarvoille suuren todennäköisyyden ja on itsekin yksinkertainen. Siis tavoite on tulkita geenisäätelyverkko siten, että ekspressiomittauksille saadaan jokin todennäköisyysjakauma ja lisäksi pitäisi pystyä laskemaan mallin kuvauspituus tavalla tai toisella. Seuraavaksi hahmotellaan muutamia lähestymistapoja geenisäätelyverkon kuvaamiseen probabilistisena mallina.

2.1.3 Bayesverkot geenisäätelyverkon mallina

Ehkä yleisin probabilistinen geenieksprression malli on bayesverkot [Pea88]. Bayesverkot ovat graafinen esitystapa tietyt ehdolliset riippumattomuusoletukset täyttäville todennäköisyysjakaumille.

Määritelmä 4 *Verkko $\mathcal{G}$ on syklinen mikäli sen kaaria seuraamalla voidaan kulkea*

sykli $x_1 \rightarrow x_2 \rightarrow \cdots \rightarrow x_1$. *Verkko on syklitön mikäli se ei sisällä yhtään sykliä, eli ei ole syklinen.*

Määritelmä 5 *Bayesverkko* $\mathcal{B}$ *on pari* $(\mathcal{G}, \mathrm{P})$*, jossa* $\mathcal{G}$ *on syklitön suunnattu verkko (Directed Acyclic Graph, DAG) ja* P *on todennäköisyysjakauma satunnaismuuttujille* $\mathbf{X} = (X_1, \ldots, X_n)$*. Verkon* $\mathcal{G}$ *solmut vastaavat satunnaismuuttujia* $\mathbf{X}$ *ja kaaret kuvaavat satunnaismuuttujien välisiä riippuvuuksia. Solmun* X *isäsolmuja verkossa merkitään* $\mathrm{PA}(X) = \{Y | Y \rightarrow X \in G\}$*. Todennäköisyysjakauman* P *pitää toteuttaa yhtälö*

$$\mathrm{P}(\mathbf{X}) = \prod_{i=1}^{n} \mathrm{P}(X_i | \mathrm{Pa}(X_i)), \tag{2.9}$$

jossa $\mathrm{Pa}(X)$ *on isämuuttujien* $\mathrm{PA}(X)$ *saamat arvot. Merkitään lisäksi mallin kaikkien satunnaismuuttujien joukkoa* $\mathbf{X}$*, yksittäistä satunnaismuuttujaa* X *ja sen saamaa arvoa* x*.*

Bayesverkon rakenteen oppiminen havaitusta jakaumasta on todistettavasti vaikeaa [Chi96] mutta siihen on kehitetty useita heuristisia algoritmeja [Hec95, FNP99, Fri98, HGC94]. Bayesverkkojen oppimisalgoritmeja on kahta eri tyyppiä. Toinen tyyppi perustuu satunnaismuuttujien riippumattomuutta tutkiviin testeihin [PV91] ja toinen hakee jonkin hyvyysfunktion optimoivaa verkkoa syklittömien suunnattujen verkkojen joukosta [FMR98, Fri98].

Bayesverkkojen luonnollinen graafinen esitystapa vaikuttaa sopivalta geenisäätelyverkkojen oppimiseen. Bayesverkkoja onkin käytetty monasti geenieksprression analyysiin [FLNP00, BF01, MM99, SBS$^+$02] mutta ne eivät suoraan sovi geenisäätelyverkon malliksi.

Eräs bayesverkkojen ongelma on, että pelkästään havaittuun dataan perustuen ei voida päätellä verkon kaarien suuntaa. Esimerkiksi verkko $X_1 \rightarrow X_2$ sisältää samat riippumattomuudet kuin verkko $X_2 \rightarrow X_1$. Vaikka niinsanotuissa kausaaliverkoissa [Pea01] voidaan tehdä jos–niin päättelyä pelkästä havainnoidusta datasta niin mikään algoritmi ei mene toiseen suuntaan ja löydä tuollaista kausaaliverkkoa pel-

kistä havainnoista. Usein tämä heikkous on kierretty hakemalla pelkästään tilastollisia riippuvuuksia kausaalisten yhteyksien sijaan.

Toinen, jopa suurempi, ongelma bayesverkoissa on niiden syklittömyys. Geenisäätelyssä (esimerkiksi keskeytysverkoissa) on takaisinkytkentää, jota bayesverkoilla ei pysty mallintamaan. Eräs tapa välttyä sykleiltä on yksinkertaisesti kieltää ne mahdollisista geenisäätelyverkoista. Tämä olisi kuitenkin hyvin raskas ja luonnoton rajoitus geenisäätelyverkolle, joka selvästikin sisältää takaisinkytkentää (esimerkiksi solun jakautumissykli).

Vaikka bayesverkot eivät sovikaan mallintamaan geenisäätelyverkkoa, voidaan niitä käyttää hyväksi muulla tavoin. Erityisesti syklisetkin geenisäätelyverkot voidaan kuvata *kaksoiskoodaavana* bayesverkkona, jolloin on mahdollista arvioida verkon soveltuvuutta suhteessa havaittuun dataan.

2.1.4 Kaksoiskoodaava bayesverkko

MDL–periaatteen käyttö geenisäätelyverkon valinnassa vaatii verkon esittämistä probabilistisena mallina. Malliksi ei sovellu suoraan geenisäätelyverkkoa vastaava bayesverkko, sillä geenisäätelyverkot ovat yleisesti syklisiä. Syklisyys voidaan kuitenkin kiertää kuvaamalla geenisäätelyverkko niin sanottuna kaksoiskoodaavana bayesverkkona.

Kaksoiskoodaava bayesverkko on rakenteeltaan kaksijakoinen ja se esittää kaikki havainnot kahtena arvoltaan identtisenä mutta tilastollisesti riippumattomana muuttujana. Kaksoiskoodaava bayesverkko ei pysty ennustamaan tulevia havaintoja mutta se pystyy laskemaan uskottavuuden jo ennestään tunnetulle datalle. Tulevien havaintojen ennustamiseen soveltuvia kaksijakoisia *dynaamisia bayesverkkoja* käsitellään luvussa 2.3.1.

Määritelmä 6 *Mikäli verkossa* $\mathcal{G} = (S, N)$ *pätee* $a \neq b$ *kaikilla* $a \rightarrow b$*, niin sitä vastaava kaksoiskoodaava bayesverkko* $\mathcal{B} = (\mathcal{K}, P)$*, on bayesverkko jossa* $\mathcal{K} = (V, E)$

ja

$$V = \{\Delta x, x | x \in S\} \quad ja \tag{2.10}$$

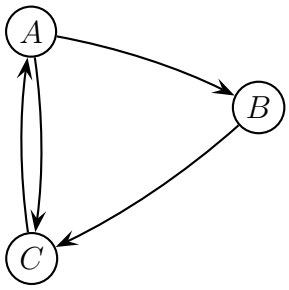
$$E = \{\Delta x \rightarrow y | x \rightarrow y \in N\} \tag{2.11}$$

Kaksoiskoodaavan bayesverkon solmuille voi antaa intuitiivisen merkityksen siten, että solmu x vastaa yhden geenin ekspressiotasoa ja solmu Δx on kyseisen geenin keskeytetty versio. Verkon kaaret kulkevat keskeytetystä solmusta siihen, johin kyseisen geenin keskeytys on vaikuttanut. Verkon kaaria voidaan pitää myös säätelysuhteina, jolloin Δx on säätelijä ja x on säädelty.

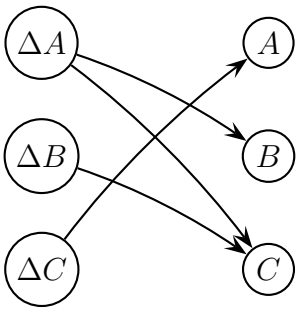
On huomattava että geenin itsesäätely on tässä mallissa kiellettyä, eli kaksoiskoodaavassa bayesverkossa ei saa olla kaarta $\Delta x \rightarrow x$. Itsesäätelyn havaitseminen keskeytyskokeessa on mahdotonta, joten sen kieltäminen ei ole todellinen rajoitus, vaikka geenin itsesäätely onkin biologisesti mahdollista. Lisäksi itsesäätelyn kieltämisellä vältetään verkkoon tuleva keinotekoinen deterministinen yhteys samat arvot saavien muuttujien Δx ja x välillä.

Kaksoiskoodaavassa bayesverkossa joukon $\{\Delta x\}$ kuvauspituus on kiinteällä datajoukolla vakio, kun data koodataan määritelmän 6 antamalla jakauman tekijöinnillä. Todennäköisyydeksi $\mathrm{P}(\Delta x = 1)$ voidaan valita mikä tahansa kiinteä todennäköisyys paitsi 0 tai 1, esimerkiksi 0.5. Tarjottu geenisäätelyverkko sopii dataan sitä paremmin mitä lyhyempi on joukon $\{x\}$ kuvauspituus. Esimerkiksi kuvassa 2.2(a) esitetty syklinen geenisäätelyverkko esitetään kuvan 2.2(b) mukaisena kaksoiskoodaavana bayesverkkona. Kuvasta kannattaa huomata, että jokaista geenisäätelyverkon solmusta A lähtevää kaarta vastaa kaksoiskoodaavassa verkossa solmusta ΔA lähtevä kaari ja saapuvaa kaarta solmuun A saapuva kaari. Samoin muille geeneille B ja C.

Lasketaan diskretoidun ekspressiodatan $D = \{\mathbf{x_i} \in \{0,1\}^m | i = 1, \ldots, n\}$ kuvauspituus mallintaen sitä kaksoiskoodaavalla bayesverkolla $\mathcal{B}$. Diskretoidussa datajoukossa on $x_{ij} = 1$ mikäli geeni j ekspressoitui kokeessa i ja $x_{ij} = 0$ mikäli geeni j ei ekspressoitunut. Yhtälöstä (2.8) tiedetään että $\mathrm{DL}(\mathcal{B}, D) = \mathrm{DL}(\mathcal{B}) + \mathrm{DL}(D|\mathcal{B})$,



(a) Geenisäätelyverkko



(b) Kaksoiskoodaava bayesverkko

Kuva 2.2: Geenisäätelyverkon kuvaaminen bayesverkkona.

josta summan termit voidaan laskea erikseen. Jälkimmäinen termi voidaan laskea

$$\mathrm{DL}(D|\mathcal{B}) = \sum_{i=1}^{n} \left(\sum_{j=1}^{m} \mathrm{DL}(\Delta X_j = x_{ij}) + \sum_{j=1}^{m} \mathrm{DL}(X_j = x_{ij}|\mathrm{Pa}(X_j)) \right), \qquad (2.12)$$

joka vastaa koodia, missä ensin tallennetaan muuttujien ΔX_i arvot koodattuna kiinteäpituisella koodilla ja myöhemmin näitä arvoja käyttäen tallennetaan samat arvot muuttujissa X_i käyttämällä arvoista $\mathrm{Pa}(x_i)$ riippuvaa koodia. X_i:n arvoa ei voida lukea suoraan muuttujasta ΔX_i, sillä käytetyssä todennäköisyysmallissa muuttujat X_i ja ΔX_i ovat riippumattomia (eli niiden välillä ei ole verkossa kaarta).

Koodinpituuden toinen termi, joka kuvaa mallin monimutkaisuutta, voidaan laskea antamalla mallin koodaus. Δ–muuttujien jakaumat voidaan koodata kertomalla muuttujien määrä ja jakaumat tilassa $\log m$. Tilaa saadaan säästettyä, kun jokaisella Δx –muuttujalla on sama jakauma. Jokaiselle X_i muuttujalle puolestaan kerrotaan sen vanhempien joukon alkiot $\mathrm{PA}(X_i)$ ja jokaista vanhempien arvoasetusta vastaava todennäköisyys. Yhteensä tällaisen koodin pituus on (unohtaen yksittäisten parametrien pituudet)

$$\mathrm{DL}(\mathcal{B}) = \log m + \sum_{i}^{m} \left(|\mathrm{PA}(X_i)| \log m + 2^{|\mathrm{PA}(X_i)|} \right). \qquad (2.13)$$

Yllä esitettyyn koodinpituuteen perustuvaa mallinvalintakriteeriä voidaan kritisoida

monella tapaa. Tärkeä kritisoitava oletus on mitan yhteys biologiaan. Ei ole mitenkään selvää, että informaatioteoriaan perustuva mitta on missään suhteessa yhteensopiva biologisen todellisuuden kanssa. Voidaan kysyä onko havainnot tilastollisessa mielessä parhaiten selittävä malli järkevä biologiselta kannalta. Tämä kritiikki on hyvinkin aiheellista eikä tätä hyvyysmittaa voi pitää millään muotoa biologisena totuutena. Tätä hyvyysmittaa käytettäessä on oltava erityisen tarkka, jotta löydetty verkko olisi biologisesti mielekäs. Erityisesti kuvauspituuden minimoivaa kaksoiskoodaavaa bayesverkkoa vastaava geenisäätelyverkko ei yleisesti kerro mitään geenien välisistä syy–seuraus suhteista.

Suurin vika MDL–kriteerissä on, että se tuottaa pienillä datamäärillä alisovitettuja malleja [AG00]. Tämä tulee selvästi esille tarkasteltaessa geenisäätelyverkkoja, joissa arvioidaan olevan keskimäärin vain 4–8 kaarta solmua kohden. Ongelmaa pahentaa geenisäätelyverkon *epäsäännöllisyys* (scale free, small world [Dit01]) [RSB+02, CD99, Bar02], jonka johdosta verkossa on solmuja, joilla on hyvin korkea tuloaste. Tällaiset solmut kasvattavat mallin parametrien määrää hyvin paljon ja edellä esitellyllä MDL–kriteerillä valittu verkko jää kohtuuttoman harvaksi nykyään käytössä olevilla datamäärillä. Myöhemmin Rissanen on kehittänyt MDL–menetelmää [Ris99] siten, että alisovitusta ei pitäisi enää tapahtua. Uusi menetelmä on kuitenkin huomattavasti aikasempaa monimutkaisempi eikä siihen enää tässä tutkielmassa palata. Käytännön tuloksia ehdotetuille menetelmille ja MDL–kriteerille esitellään jäljempänä luvussa 3.

2.2 Keskeytysverkosta geenisäätelyverkkoon

Keskeytysverkkoja on suhteellisen helppo rakentaa ekspressiomittausten perusteella niille geeneille, jotka eivät ole välttämättömiä organismin selviytymiselle. Esimerkiksi tavallisen leivontahiivan (Saccharomyces cerevisiae) noin 6200 geenistä on onnistuttu keskeyttämään 95% ja alle 20% keskeytyksistä on letaaleja (eli tappavia) [WSA+99, Sac02]. Tämä suhteellisen pieni elintärkeiden geenien määrä on

yhtäpitävä sen havainnon kanssa, että geenisäätelyverkot ovat epäsäännöllisiä eli sen solmujen astejakaumalla on vahva häntä [Wag02, FB02, RSB+02]. Vahvahäntäinen solmujen astejakauma tarkoittaa, että osalla geeneistä on hyvin paljon yhteyksiä mutta suurimmalla osalla yhteyksiä on vain vähän. Tällainen verkkorakenne antaa organismille suojaa satunnaisen geenin vikaantumista (mutaatiota) vastaan [CD99]. Korkean yhteysasteen omaavien geenien on havaittu olevan normaalia vanhempia [FB02], eli evoluutiossa paremmin säilyneitä. Tämä saattaa olla seurausta niiden suuresta merkityksestä organismin hyvinvoinnille.

Onneksi onnistuneista ei-tappavista keskeytysmutanteista on julkisesti saatavilla jonkin verran ekspressiomittausdataa, joka kelpaa keskeytysverkon rakentamiseen. Erityisen kiinnostava ekspressiodatajoukko on Hughes et.al. [HMJ+00] julkaisema 300 ekspressiomittausta hiivalle. Näistä mittauksista 270 oli tehty yhden ja 6 kahden keskeytetyn geenin hiivamutanteille. Mutatoimatonta hiivaviljelmää oli käsitelty kemikaaleilla 13 kokeessa. Mittauksista 49 oli tehty hiivan ollessa haploidisessa tilassa (yksinkertainen kromosomisto) ja loput 251 hiivan ollessa diploidinen (diploid state, steady state, kaksinkertainen kromosomisto). Yhteensä 11 kokeessa yksi geeni oli keskeytetty tetrasykliinisäätelyllä. Kussakin kokeessa testi– ja kontrollinäytettä oli kasvatettu useita sukupolvia ennen RNA–näytteen eristämistä.

Näiden 300 kokeen lisäksi samoilla menetelmillä tehtiin 63 koetta, joissa kaksi samalla tavalla kasvatettua, mutatoimatonta, hiivanäytettä merkittiin eri väriaineilla ja näytteiden suhteelliset ekspressiotasot mitattiin normaalilla tavalla mikrosirulla. Nämä ylimääräiset mittaukset mahdollistavat valistuneen arvauksen jokaisen geenin yksilöllisestä ja satunnaisesta vaihtelusta. Tätä tietoa voidaan käyttää geenien ekspressiotason muutoksen merkitsevyyden arvioinnissa.

Mutatoimisen jälkeen jakautuneilla soluilla tehty ekspressiomittaus paljastaa sekä keskeytyksen välittömät että välilliset vaikutukset. Tulosten tulkitsemisen ongelma on, ettei välillisiä ja välittömiä vaikutuksia pystytä erottamaan toisistaan. Jotta keskeytyskokeista rakennettu keskeytysverkko vastaisi paremmin luvussa 1.3 esiteltyä geenisäätelyverkkoa, on siitä poistettava välillisiä vaikutuksia kuvaavat kaaret.

2.2.1 Keskeytysverkon transitiivinen reduktio

Suoraviivainen tapa poistaa epäsuoraa säätelyä kuvaavia kaaria on laskea keskeytysverkon transitiivinen reduktio [GMSU89, Wag01]. Tämä menetelmä on herkkä ekspressiodatan kohinalle ja kaarien valinnassa käytetylle menetelmälle. Puhtaasti verkkoteoriaan perustuva menetelmä ei myöskään ota huomioon mittaustulosten uskottavuutta kunkin kaaren yhteydessä.

Floyd–Warshall algoritmin [Flo62, CLR89] tyyppisellä lähestymistavalla voidaan hinnoitella verkon kaaret ja ottaa verkon karsimisessa huomioon myös säätelyn uskottavuus. Uskottavuusmitan tulisi olla sellainen, että se antaa suoralle säätelylle suuren uskottavuuden eli pienen hinnan, ja korottaa hintaa, kun säätelysuhteella on enemmän välittäjiä. Kun käytössä on tällainen hinnoittelufunktio c, keskeytysverkon kaari A:sta B:hen voidaan korvata sellaisella A:n ja B:n välisellä polulla, jonka kallein kaari on halvempi kuin A:n ja B:n välinen suora kaari. Toisin sanoen karsitussa verkossa halutaan säilyttää geenien A ja B välinen min–max –polku, joka on halvin niiden välinen polku, jossa polun hinta on sen kalleimman kaaren hinta

$$c(A, B) = \min_{p=A \rightsquigarrow B} \max_{x \to y \in p} c(x \to y). \tag{2.14}$$

Verkon kaikkien min–max –polkujen hinnat saadaan laskettua algoritmilla 1 kohtalaisen nopeasti, ajassa $O(n^3)$, jossa n on tarkasteltavien geenien määrä. Algoritmin toiminnan kuvaamiseksi määritellään ensin yksi uusi merkintätapa. Merkitään $a \rightsquigarrow_k b$ sellaista polkua $a \to x_{i_1} \to \cdots \to b$ solmujen a ja b välillä, minkä välisolmuille x_i pätee $i < k$. Välisolmulla tarkoitetaan polulla olevia solmuja lukuun ottamatta alku– ja loppusolmua.

Nyt algoritmin 1 toimintaa selventää sen uloimman, muuttujaa k iteroivan, silmukan invariantti $K[i, j] = c(i \rightsquigarrow_k j)$. Intuitiivinen tulkinta algoritmin toiminnalle on se, että min–max –polkuja haetaan kasvavan polunpituuden suhteen ja polkua kasvatetaan kasvattamalla sallittujen välisolmujen joukkoa. Tästä joukosta voidaan polulle valita uusi solmu tai jättää se valitsematta, jos se ei paranna aiemmin löydettyä polkua. Algoritmin kaksi sisintä silmukkaa tarkistaa kaikille solmuille i ja j,

onko polku $i \rightsquigarrow_k k \rightsquigarrow_k j$ halvempi kuin $i \rightsquigarrow_k j$. Näistä vaihtoehdoista halvempi tulee poluksi $i \rightsquigarrow_{k+1} j$.

Taulukko Π puolestaan säilyttää kirjanpitotietoa varsinaisista min–max –poluista. Taulukkoa Π voidaan pitää eräänlaisena tienviittana, jonka mukaan solmusta i pitää lähteä suuntaan $\Pi[i, j]$ kun halutaan min–max –polkua pitkin solmuun j. Seuraavassa solmussa katsotaan taas uutta tienviittaa solmuun j.

Algoritmi 1 Min–max polkujen laskeminen.

Syöte: Kaarien $i \rightarrow j$ hinnat $c(i \rightarrow j)$.

Tuloste: Min–max polku $i \rightsquigarrow j$ hinnaltaan $K[i, j]$ lähtee solmun $\Pi[i, j]$ kautta.

```
1:  for i ← 1, ..., n do
2:      for j ← 1, ..., n do
3:          K[i, j] = c(i → j)
4:          Π[i, j] = j
5:      end for
6:  end for
7:  for k ← 1, ..., n do
8:      for i ← 1, ..., n do
9:          for j ← 1, ..., n do
10:             p = max(K[i, k], K[k, j])
11:             if p < K[i, j] then
12:                 K[i, j] ← p
13:                 Π[i, j] ← Π[i, k]
14:             end if
15:         end for
16:     end for
17: end for
```

Min–max –polut sisältävä aliverkko on hieman samankaltainen kuin verkon pienin virittävä puu. Normaalissa suunnatussa tapauksessa min–max –verkko ja pienin virittävä puu eroavat tosin jo käsitteiden tasolla, sillä suunnatun verkon virittävässä

puussa on aina juuri, jota min–max –verkossa ei välttämättä ole. Sen sijaan suuntaamattoman verkon min–max –verkko on hyvin samankaltainen kuin sen pienin virittävä puu.

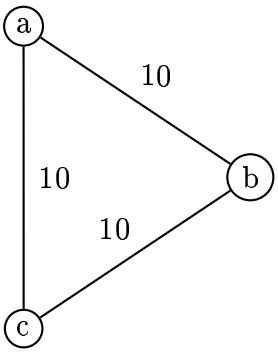
Täsmällisesti sanottuna jokaisesta suuntaamattomasta min–max –verkosta voidaan konstruoida pienin virittävä puu ja jokainen suuntaamaton pienin virittävä puu on min–max –verkko. Suuntaamaton min–max –verkko on virittävä puu, mikäli se on syklitön. Syklittömyys on helppo saattaa voimaan jättämällä min–max –verkon sykleistä pois kallein kaari. Tällöin minkään polun max–hinta ei muutu. Esimerkiksi kuvan 2.3(a) verkosta voidaan jättää mikä tahansa 10 hintainen kaari pois muuttamatta verkon min–max –ominaisuutta.

Todistetaan että suuntaamattoman verkon syklitön min–max –verkko on alkuperäisen verkon virittävistä puista pienin. Tarkastellaan tilannetta kuvassa 2.3(b), jossa min–max –verkon kaari $i - j$ korvataan halvemmalla kaarella $k - l$ ja solmut i ja k ovat keskenään samassa verkon komponentissa ja vastaavasti j ja l ovat omassa komponentissaan. Tällaista kaarta $k - l$ ei kuitenkaan voi olla olemassa, sillä se on vähintään yhtä kallis kuin kallein polun $k \sim i - j \sim l$ kaari. Erityisesti se on kalliimpi kuin kaari $i - j$. Näin ollen syklitöntä min–max –verkkoa ei voi keventää, joten se on pienin virittävä puu.

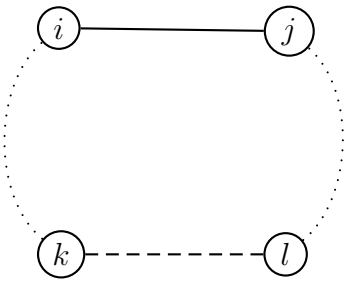
Se että pienin virittävä puu on min–max –verkko voidaan nähdä Kruskalin algoritmia mukailevalla induktiolla [CLR89, Kru56]. Todetaan aluksi että yksisolmuinen verkko on itsensä min–max –verkko ja pienin virittävä puu. Oletetaan että väite pätee kaikille verkoille, joissa on $k < n$ solmua.

Valitaan n solmuisen verkon pienimmästä virittävästä puusta yksi kaari, joka jakaa puun kahteen osaan. Molemmissa osapuissa on vähemmän kuin n solmua, joten osat ovat induktio oletuksen mukaan osapuiden virittämien verkkojen pienimpiä virittäviä puita.

Valittu kaari on edullisin osaverkot yhdistävistä kaarista, sillä muutoin saataisiin aikaan pienempi virittävä puu valitsemalla jokin muu osat yhdistävä kaari. Tämä kaari kuuluu myös verkon min–max –verkkoon, mikä nähdään yllä läpikäydyllä ku-



(a) Syklinen, suuntaamaton min–max –verkko



(b) Kiinteä ja pisteviiva kuuluu min–max –verkkoon, katkoviiva ehdotetaan virittävän puun alennukseksi

van 2.3(b) argumentilla. Näin ollen n solmuinen pienin virittävä puu on verkon min–max –verkko.

Algoritmin 1 kriittinen osa on hinnoittelufunktion c valinta. Funktion pitää antaa korkea arvo verkosta puuttuville kaarille ja mahdollisesti huomioida geenisäätelyn suunta geenistä toiseen. Keskeytyskokeita tarkasteltaessa voidaan mikä tahansa hinnoittelufunktio c muuttaa siten, että kaikki ne kaaret, joita ei havaittu keskeytyskokeessa, saavat äärettömän korkean hinnan.

Algoritmin 1 biologisesti kyseenalainen ominaisuus on hyväksyttävän polun valinta. Ei ole ilmeistä, että mielivaltaisen pitkä min–max –polku on biologisesti järkevä säätelyreitti. Polun hinta saattaa tarvita jonkin polun pituudesta riippuvan korjaustermin. Tällaisen hinnoittelufunktion optimointia ei tarkastella tässä tutkielmassa.

2.2.2 Karsittavien kaarien hinnoittelu

Keskeytysverkkojen karsimisessa oleellinen osa on kaarien hinnoittelumetriikalla. Jos metriikka antaa halvan hinnan epäsuoraa säätelyä kuvaavalle kaarelle, keskeytysver-

kosta karsitaan väärät kaaret ja tulos on virheellinen. Hinnoittelumetriikan pitää ottaa huomioon välittäjägeenien aiheuttama kohina säätelyssä.

Yksinkertainen mahdollisuus hinnoittelumetriikan valinnaksi on käyttää jotain tunnettua etäisyysmittaa. Ekspressioprofiilien klusteroinnissa on käytetty useita erilaisia etäisyysmittoja. Tavallisin mitta on euklidinen etäisyys, joka mitatuille geenien x ja y ekspressioprofiileille $(x_1, \ldots, x_n)$ ja $(y_1, \ldots, n)$ on $\sqrt{\sum_{x=1}^{n}(x_i - y_i)^2}$. Tämän metriikan heikkous on sen herkkyys absoluuttisille arvoille. Jos toinen geeneistä varioi luonnollisesti paljon ja toinen vähän, etäisyys kasvaa suureksi vaikka profiili olisikin samankaltainen. Toinen usein käytetty mitta on kosinikorrelaatio. Se ei ole herkkä absoluuttisille arvoille. Kosinikorrelaatio määritellään kahden vektorin x ja y välisen kulman kosinina $\cos \angle(x, y)$, jossa kulma taas lasketaan sisätulon laskusäännöillä. Kaavoina siis

$$\cos \angle(x, y) = \frac{x \cdot y}{|x||y|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{(\sum_{i=1}^{n} x_i^2)(\sum_{i=1}^{n} y_i^2)}}. \tag{2.15}$$

Kosinikorrelaatio varioi arvojen -1 ja 1 välissä olemattoman korrelaation saadessa arvon 0. Geenisäätelyverkon kaaren $A \to B$ hinnaksi määritellään siten $c(A \to B) = 1 - |\cos \angle(A, B)|$. Huomattavaa tässä hinnoittelutavassa, ja kaikilla metriikoihin perustuvissa tavoissa, on hinnan symmetrisyys: $c(A \to B) = c(B \to A)$.

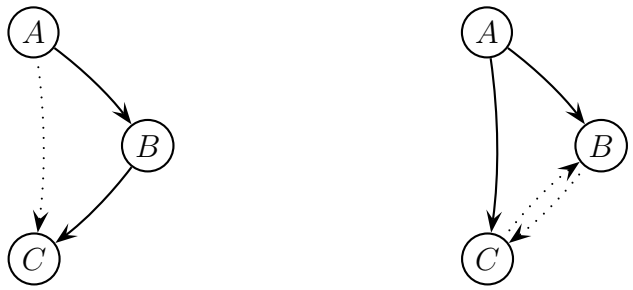
Eräs toinen moneen yhteyteen sopiva samankaltaisuusmitta on yhteinen informaatio. Geenien ekspressiotasoja kuvaavista satunnaismuuttujista saadaan otos toistetuilla ekspressiomittauksilla. Mittausten pitää olla toisistaan riippumattomia mikäli otokseen halutaan toistoja. Aikasarja ei periaatteessa siis kelpaa, koska peräkkäiset kokeet ovat hyvin paljon riippuvaisia toisistaan. Yksinkertaisuuden vuoksi geenin ekspressiosta käytetään usein vain kahta arvoa, geeni joko ekspressoituu tai sitten ei. Arvioidaan geenin X ekspressoitumistodennäköisyyttä

$$\mathrm{P}(X) = \frac{\#\text{ kokeet, joissa geeni } X \text{ ekspressoituu}}{\text{Kokeiden määrä}} \tag{2.16}$$

Käyttäen yhteistä informaatiota, keskeytysverkon kaarille voidaan antaa hinnat. Kaarelle $A \to B$ määrätään hinta

$$c(A \to B) = -\mathrm{I}(A, B), \tag{2.17}$$

joka on sitä pienempi, mitä paremmin geenien A ja B ekspressiot vastaavat toisiaan.



Kuva 2.3: Esimerkki bayesverkkona kuvatusta geeniverkosta

Tarkastellaan kuvan 2.3 bayesverkkoa, jossa geeni A vaikuttaa geeniin B, joka puolestaan vaikuttaa geeniin C. Jotta yhteiseen informaatioon perustuva hinnoittelumetriikkamme olisi järkevä, geenien A ja C yhteinen informaatio pitää olla pienempi kuin kuin geenien A ja B tai geenien B ja C.

Puhtaasti ekspressiomittauksiin perustuvien hinnoittelumetriikkojen lisäksi voidaan kuvitella kaarten hinnoitteluun käytettävän muitakin datalähteitä. Eräs vaihtoehtoinen tapa hinnoitella kaaria on käyttää hyväksi geenien säätelyalueiden sisältämää informaatiota. Geenien cis–säätelyalueilla sijaitsevat sitoumapisteet antavat mikrosirumittauksia täydentävää tietoa geeniekspression säätelystä. Tätä yhteyttä on aiemmin käytetty hyväksi ryhmiteltäessä geenejä ekspressiotasojen ja säätelyalueiden mukaan klustereihin [HB00, BF01].

Sitoumapisteinformaation lisääminen säätelyverkon oppimiseen on haastavaa, sillä “sitoumapiste” ei ole hyvin määritelty käsite. Käyttämällä yksinkertaista määritelmää “lyhyt nukleotidihahmo” saadaan runsaasti vääriä positiivisia ja negatiivisia sitoumapisteitä, jotka vaikeuttavat cis–säätelyverkon rakentamista. Yhdistämällä ekspressiomittaustulokset ja sekvenssitiedot yhdeksi todennäköisyysmalliksi, kahdesta huonosta informaatiolähteestä voidaan toivottavasti rakentaa yksi kohtuullinen malli, sillä oletettavasti eri lähteistä tuleva informaatio on ainakin osittain toisistaan riippumatonta.

Merkitään $A \dashrightarrow B$, mikäli geenin A ekspressiotaso vaikuttaa geenin B ekspressioon ja $A \multimap B$, mikäli geenin A tuottama proteiini sitoutuu geenin B säätelyalueelle. Olkoon $\mathbb{E}$ tehtyjen ekspressiomittausten tulokset ja $\mathbb{S}$ organismin genomi (erityisesti geenien säätelyalueet). Kaaren hinnoittelussa voidaan käyttää sitoutumisen ja ekspressiovaikutuksen yhteistodennäköisyyttä, jota voidaan laskea kaavoilla

$$\begin{aligned} \mathrm{P}(A \dashrightarrow B, A \multimap B|\mathbb{E},\mathbb{S}) &= \mathrm{P}(A \dashrightarrow B|A \multimap B,\mathbb{E},\mathbb{S})\mathrm{P}(A \multimap B|\mathbb{E},\mathbb{S}) \quad (2.18)\\ &= \mathrm{P}(A \multimap B|A \dashrightarrow B,\mathbb{E},\mathbb{S})\mathrm{P}(A \dashrightarrow B|\mathbb{E},\mathbb{S}) \quad (2.19)\\ &\overset{\perp\!\!\!\perp}{=} \mathrm{P}(A \multimap B|\mathbb{E},\mathbb{S})\mathrm{P}(A \dashrightarrow B|\mathbb{E},\mathbb{S}) \quad (2.20) \end{aligned}$$

Yhtälön (2.19) tekijä $\mathrm{P}(A \dashrightarrow B)$ on kohtalaisen helppo arvioida toistetuista ekspressiokokeista, kun tulkitaan että $\mathrm{P}(A \dashrightarrow B) = \mathrm{P}(B = 1|A = 1)$. Termin $\mathrm{P}(A \multimap B|A \dashrightarrow B,\mathbb{S},\mathbb{E})$ arvioiminen on kuitenkin hankalampaa.

Sitoumapisteen ja säätelyn yhteistodennäköisyyttä $\mathrm{P}(A \dashrightarrow B, A \multimap B|\mathbb{E},\mathbb{S})$ voidaan approksimoida iteratiivisesti. Ensiksi valitaan A:n keskeytyksessä vahvasti säädellyt geenit, joiden säätelyalueilta haetaan yleistä hahmoa. Hahmon jollain sopivalla tavalla määriteltyä sopivuutta geenin säätelyalueeseen käytetään parametrina sitoumapisteen todennäköisyydestä ensimmäisessä vaiheessa. Tästä jatketaan valitsemalla seuraavaan iteraatioon geenit, joilla yhteistodennäköisyys on tarpeeksi korkea.

Laskennallisesti halvempi vaihtoehto on tehdä selvästikin väärä oletus, että sitoumapisteen ja vaikutuksen olemassaolo ovat toisistaan riippumattomia kuten yhtälössä (2.20). Tällä oletuksella iterointi voidaan pysäyttää heti ensimmäiseen kertaan.

Yleisten hahmojen hakuun geenien säätelyalueilta on olemassa valmiita ohjelmia kuten MEME [BE94], AlignACE [HETC00] ja Spexs [Vil98]. Näitä käyttämällä on mahdollista opetella ainakin rajoitettu määrä säätelysuhteita. Tällainen cis–trans säätelyn hinnoittelumenetelmä on hahmoteltu algoritmina 2. Algoritmi laskee kaaren hinnaksi arvon $-\log \frac{\mathrm{P}(B=1|A=1)\mathrm{P}(A\multimap B)}{\mathrm{P}(B=1)\sum_X \mathrm{P}(A\multimap X)}$, joka on sitä pienempi mitä enemmän geenipari A, B eroaa satunnaisesta geeniparista.

Algoritmin 2 kyseenalaisin osa on arvon $\mathrm{P}(site$ geenin B yläalueella) laskenta. Jos

Algoritmi 2 Cis–trans säätelyn $A \rightarrow B$ hinnoittelu.

Syöte: Geenien säätelyalueet ja ekspressiotasot keskeytyskokeissa.

Tuloste: Kaarien hinta $c(A \rightarrow B)$.

$P_e = \mathrm{P}(B = 1|A = 1)$.

$P_{e0} = \mathrm{P}(B = 1)$.

$site$ = Parhaiten geenin A keskeytyksessä ekspressiotaan muuttaneiden geenien säätelyalueet tunnistava hahmo.

$P_b = \mathrm{P}(site$ geenin B yläalueella)

$P_{b0} = \mathrm{P}(site$ satunnaisella yläalueella)

return $-\log(P_e/P_{e0}) - \log(P_b/P_{b0})$

mahdollinen sitoumapistehahmo $site$ on annettu jossain mielessä probabilistisesti, on mahdollista laskea todennäköisyydeksi tulkittava hyvyys hahmon esiintymiselle tietyssä sekvenssissä. Erityisesti niin sanotut painomatriisimenetelmät [SSGE82, Sto00] tuottavat tiettyjen oletusten vallitessa todennäköisyyden sille, että tietty DNA–sekvenssi toimii cis–säätelytekijänä.

Painomatriisit

Painomatriisit ovat yleisesti käytetty menetelmä cis–säätelytekijöiden kuvaukseen ja hakuun [Sto00]. Menetelmä kuvaa sitoumapisteet matriisina $W(b, i)$, jossa b on jokin nukleotideista A, C, G tai T, $1 \leq i \leq l$ ja l on kyseisen sitoumapisteen pituus. Sekvenssin $S = s_1 s_2 \ldots s_l$ pisteluku saadaan summaamalla sen nukleotidien painot yhteen

$$Score(S) = \sum_{i=1}^{l} W(s_i, i). \tag{2.21}$$

Mitä suuremman pisteluvun sekvenssi saa, sitä todennäköisemmin se on sitoumapiste. Yhtälössä (2.21) esiintyvää additiivisuusoletusta on viimeaikoina kritisoitu siitä ettei se täysin vastaa kokeellisesti saatuja tuloksia. Tästä huolimatta additiivisuutta pidetään kohtalaisen hyvänä approksimaationa todellisuudelle varsinkin huomioitaessa että additiivinen malli käyttää useimpiin vaihtoehtoihinsa verrattuna varsin vähän parametreja [BBS02].

Jos oletetaan, että hahmon jokainen nukleotidi vaikuttaa toisista riippumattomasti sitoutumisenergiaan ja siten sitoutumisen kokonaistodennäköisyyteen, voidaan termodynaamisin argumentein osoittaa [HLS94] että

$$\mathrm{P}(S \text{ on sitoumapiste}) = \frac{e^{Score(S)}}{Z}, \tag{2.22}$$

jossa Z on summa $\sum e^{Score(T)}$ yli genomin kaikkien l pituisten sekvenssien T. Mikäli lisäksi oletetaan, että genomin nukleotidit jakautuvat toisistaan riippumattomasti todennäköisyyksillä $\mathrm{P}(b)$, voidaan Z laskea analyyttisesti

$$Z = \prod_{i=1}^{l} \sum_{b \in \{A,C,G,T\}} \mathrm{P}(b) e^{W(b,i)} \tag{2.23}$$

ja käyttämällä tätä tulosta, saadaan optimaaliseksi W, joka parhaiten erottelee sitoumapisteet taustagenomista

$$W(b,i) = \ln \frac{f(b,i)}{\mathrm{P}(b)}, \tag{2.24}$$

jossa $f(b,i)$ on nukleotidin b frekvenssi kyseisen transkriptiofaktorin sitoumapisteiden kohdassa i. Oletus riippumattomasti jakautuneista nukleotideista ei luonnollisesti ole täysin oikea, mutta se on usein tyydyttävän hyvä arvio todellisuudesta.

Käyttämällä kaavan (2.24) määrittelemiä painoja ja oletusta nukleotidien riippumattomuudesta, saadaan

$$e^{Score(S)} = e^{\sum_{i=1}^{l} \ln \frac{f(b,i)}{\mathrm{P}(b)}} = \prod_{i=1}^{l} \frac{f(b,i)}{\mathrm{P}(b)} = \frac{\mathrm{P}(S \text{ sitoumapiste})}{\mathrm{P}(S \text{ satunnainen})}, \tag{2.25}$$

joka on toinen algoritmin 2 käyttämistä todennäköisyyksien suhteista.

2.3 Geenisäätelyverkon rakentaminen keskeytysmittauksista

Luvussa 2.2.1 esitetylle karsinnalle vaihtoehtoinen menetelmä geenisäätelyverkkojen löytämiseen on rakentaa verkko lähtien pelkistä solmugeeneistä, joihin lisätään kaaria jäljestä päin. Tämä konstruktiivinen tapa on ollut erityisen suosittu varhaisissa yrityksissä muodostaa mikrosirumittauksista geenisäätelyverkkoja.

Konstruktiivinen lähestymistapa on erityisen herkkä mittausdatassa olevalle kohinalle, sillä tyypillisesti mittauspisteitä on huomattavasti vähemmän suhteessa haettaviin parametreihin kuin tarkasteltaessa keskeytysverkkoja. Keskeytysverkko, jossa yhdellä geenillä on lähteviä kaaria, voidaan rakentaa teoreettisesti perusteltavilla menetelmillä jo kolmella mikrosirulla, joista saadaan riittävästi informaatiota ekspression muutoksen keskiarvon ja hajonnan arvioimiseen. Ilman ulkopuolista väliintuloa (geenin keskeytys) kolmella mikrosirulla on mahdotonta löytää luotettavia kandidaatteja yhdenkään geenin säätelijöiksi (tai säädeltäviksi). Luottamuksen puute johtuu yksinkertaisesti liian suuresta vapausasteesta: Säätelevät geenit valitaan liian suuresta joukosta, jolloin on todennäköistä löytää sopivat geenit aivan satunnaisellakin datalla.

Yksi rakentavan lähestymistavan ongelma on verkkoon lisättävien kaarien tulkinta. Ei ole selvää miten kahden geenin välille voidaan lisätä säätelysuhdetta kuvaava suunnattu kaari siten että kaarella olisi kausaalinen tulkinta kuten keskeytyskokeilla löydetyissä kaarissa. Seuraavassa tavoitellaan geenisäätelyverkon rakentamista eräässä mielessä pseudokausaalisen tulkinnan pohjalta. Kaaria lisätään ajasta riippuvan kriteerin perusteella, mutta lisäyksissä rajoitutaan vain niihin kaariin, jotka ovat perusteltuja myös keskeytysmittausten perusteella.

2.3.1 Dynaamiset bayesverkot

Dynaamiset bayesverkot [DK89, Mur02] kuvaavat sellaisten satunnaismuuttujien jonon jakaumaa, jossa seuraavan satunnaismuuttujan jakauma riippuu edellisen muuttujan arvosta. Dynaamisissa bayesverkoissa satunnaismuuttujien jono on oikeastaan jono satunnaismuuttujien joukkoja ja seuraavan satunnaismuuttujajoukon jakauma riippuu edellisen joukon arvoista bayesverkon määrittelemällä tavalla.

Tärkeä dynaamisen bayesverkon ominaisuus on sen rajoitettu muisti. Dynaamisella bayesverkolla on niin sanottu Markov–ominaisuus eli sen määräämä jakauma riippuu ainoastaan annetusta verkosta ja edellisen aikapisteen arvoista; varhaisemmilla arvoilla ei ole merkitystä tulevaan jakaumaan kunhan nykyisen aikapisteen arvot

tunnetaan. Tekniseltä kannalta dynaaminen bayesverkko siis määrittelee Markov–prosessin.

Määritelmä 7 *Markov–prosessi on jono satunnaismuuttujia $X_0, X_1, \ldots$, joilla on Markov–ominaisuus*

$$\mathrm{P}(X_{m+1} = j | X_m = i_m, \ldots, X_0 = i_0) = \mathrm{P}(X_{m+1} = j | X_m = i_m), \tag{2.26}$$

eli tuleva tila on riippumaton menneestä annettuna nykytila.

Markov–ominaisuus on sangen käytännöllinen mallinnettaessa aikasarjoja, sillä mallintaminen voidaan tehdä hyvin pienissä paloissa, mikä puolestaan helpottaa mallin valintaa. Mallinnuksen yhteydessä on huomioitava, onko Markov–ominaisuus todella voimassa.

Eräs esimerkki Markov–prosessin käytännöllisyydestä on aikasarjan $(x_0, \ldots, x_m)$ logaritmisen uskottavuuden laskeminen, joka voidaan tehdä termeittäin kaavalla

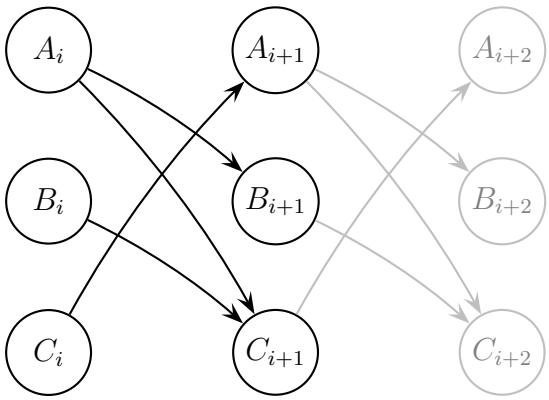
$$\log \mathrm{P}(x_0, \ldots, x_m) = \log \mathrm{P}(x_0) + \sum_{i=0}^{m-1} \log \mathrm{P}(x_{i+1} | x_i). \tag{2.27}$$

Sen jokainen tekijä on helppo laskea tarkastelemalla vain nykyisen ja edellisen muuttujan arvoa. Tämä ominaisuus nousee hyvin tärkeäksi luvussa 2.3.3, jossa esitellään algoritmi dynaamisten bayesverkkojen oppimiseen datasta.

Itse dynaaminen bayesverkko kuvaa yhtälön (2.26) mukaista tilannetta siinä tapauksessa, että muuttujat X_i ovat satunnaismuuttujien joukkoja, eivätkä yksittäisiä satunnaismuuttujia. Kahden muuttujajoukon X_i ja X_{i+1} välille on määritelty bayesverkko.

Määritelmä 8 *Dynaaminen bayesverkko on $\mathcal{DB}$ on Markov–prosessi, jossa muuttujajoukon X_{i+1} jakauma ehdolla X_i on määritelty bayesverkkona $\mathcal{B}$, jonka kaaret lähtevät joukosta X_i ja päätyvät joukkoon X_{i+1}.*

Esimerkkinä dynaamisesta bayesverkosta olkoon kuvan 2.4 verkko, joka mallintaa kuvassa 2.2(a) olevan geenisäätelyverkon toimintaa. Kuvan 2.4 tumma osuus kuvaa



Kuva 2.4: Geenisäätelyverkon kuvaaminen dynaamisena bayesverkkona.

varsinaista bayesverkkoa ja harmalla hahmoteltu jakso kuvaa kuinka sama bayesverkko toistuu prosessin edetessä.

Huomattavaa on kuvan 2.4 dynaamisen bayesverkon ja kuvan 2.2(b) kaksoiskoodaavan bayesverkon samankaltaisuus. Vaikka dynaamisen ja kaksoiskoodavan bayesverkon topologiat ovatkin lähestulkoon samat, niiden tulkinta ja käyttökelpoisuus eroavat toisistaan hyvinkin paljon.

Luvun 2.1.4 kaksoiskoodaava bayesverkko ei ole ennustava malli toisin kuin dynaaminen bayesverkko. Kaksoiskoodaava verkko sopii vain mittaamaan tehdyn ennustuksen sopivuutta annettuun dataan ja siinä data pitää tietää ennakolta. Dynaaminen bayesverkko puolestaan tekee ennusteita tulevasta nykyisen tilanteen perusteella.

Erityisen merkittävä ero on opetellun rakenteen tulkinnassa. Dynaamisessa bayesverkossa kaari $A \rightarrow B$ kuvaa tilannetta, jossa ensin toimii A ja tämän jälkeen B. Kaksoiskoodaavassa verkossa sama kaari tarkoittaa, että A ja B toimivat samaan aikaan. Vaikka dynaaminen bayesverkko ei välttämättä kuvaakaan kausaalista "A aiheuttaa B:n"–suhdetta, se kuitenkin kertoo ajan suhteen etenevistä ilmiöistä, joita kausaalinen suhde aina tuottaa.

Ei–kausaalinen kaari voi sopia dynaamiseen bayesverkkoon esimerkiksi sisarsolmujen eri reaktionopeuksien perusteella. Jos esimerkiksi A vaikuttaa nopeasti B:hen ja hitaasti C:hen, voi havaintojen perusteella vaikuttaa siltä kuin A vaikuttaisi vain

B:hen, joka puolestaan vaikuttaisi vain C:hen. Tällaisessa tilanteessa oikean topologian löytäminen vaatisi ulkoisella väliintulolla suoritettuja kokeita.

2.3.2 Geenieksprression mallinnus dynaamisilla bayesverkoilla

Dynaamiset bayesverkot sopivat sangen suoraviivaisesti geenieksprression mallinnukseen [MM99]. Kun tarjolla on vakiomittaisin aikavälein tehtyjä mikrosirukokeita, ekspression muutoksia on helppo mallintaa käyttämällä dynaamista bayesverkkoa, jonka solmuja ovat edellisen ja seuraavan aikapisteen ekspressiotasot kaikille geeneille.

Erityisen kiinnostavia tuloksia saadaan opittaessa verkon rakenne suoraan havainnoista [FMR98]. Tuloksena saatava verkko ei välttämättä kuvaa syy–seuraus–suhteita, kuten luvussa 2.3.1 havaittiin, mutta aikariippuvat suhteet riittävät ennustamaan genomin toimintaa tulevaisuudessa ja helpottavat testattavien hypoteesien muodostamista kausaalisista suhteista.

Dynaamisten bayesverkkojen rakenteen oppiminen havaitusta datasta ei ole helppoa, kuten ei tavallistenkaan bayesverkkojen. Itse asiassa bayesverkkojen rakenteen oppimisongelma on todistettu NP-kovaksi [Chi96]. Käytännön algoritmit [Hec95] etsivät verkon rakennetta heuristisesti, joko testaamalla solmujen riippumattomuutta tai optimoimalla jotain hyvyysfunktiota.

Useimmin käytettävät hyvyysfunktiot perustuvat mallin monimutkaisuuden ja dataan sopivuuden yhdistelmälle. Luvussa 2.1.2 esitelty MDL–periaate antaa erään, varsin yleisesti käytetyn, arvotusmenetelmän bayesverkkojen rakenteelle. Muita arvotuskriteerejä ovat muiden muassa BIC (Bayesian Information Criterion)[Sch78], AIC (Akaike Information Criterion) [Aka78] ja ristiinvalidointi [Sto74]. Vertailtaessa näitä arvotusmenetelmiä keskenään [AG00], on todettu että BIC ja, yleisessä muodossaan, MDL tuottavat alimääriteltyjä malleja mikäli dataa on käytössä vain vähän. Tämä on varsin huono ominaisuus tilanteessa, jossa ehdotettavan geenisäätelyverkon on tarkoitus tarjota laboratoriossa testattavia hypoteeseja.

Yllä mainituista mallinvalintamenetelmistä yleisimmin bayesverkkojen oppimisessa käytetään BIC ja AIC kriteereitä. Niiden arvot on varsin helppo laskea opitusta mallista ilman sivuun jätettyä testidataa. BIC arvo lasketaan mallille $\mathcal{M}$ opetusdatalla D funktiolla

$$BIC(D) = \mathrm{DL}(D|\mathcal{M}) + \frac{1}{2}k \log |D|, \tag{2.28}$$

jossa k on mallin parametrien määrä. Mallin parametrien määrä kasvaa tyypillisesti hyvin nopeasti mallin muuttuessa monimutkaisemmaksi. Esimerkiksi binääriarvoisten muuttujien bayesverkkossa on $2^{|\mathrm{PA}(i)|}$ parametriä, jokaiselle solmulle i.

BIC kriteerin rangaistustermi on suhteellisen suuri kun käytössä on vain vähän dataa. Tästä johtuen BIC tuntuu suosivan yksinkertaisia malleja ellei tarjolla ole todella paljon dataa. Toinen mallinvalintakriteeri AIC rankaisee pienillä datamäärillä mallin monimutkaisuudesta huomattavasti BIC–kriteeriä lempeämmin. AIC lasketaan kaavalla

$$AIC(D) = \mathrm{DL}(D|\mathcal{M}) + k \log \mathrm{e}, \tag{2.29}$$

Empiirisissä kokeissa [AG00] on tultu tulokseen, että ristiinvalidointi olisi käytännössä paras tapa vältää mallin ylisovittamista. Geenisäätelyverkkoja mallinnettaessa aina läsnä oleva ongelma, empiirisen datan vähyys, tulee kuitenkin myös ristiinvalidoinnin ongelmaksi. Koska ristiinvalidoinnissa jätetään aina sivuun pieni osa esimerkkidatasta, voi helposti sattua ettei jäljelle jää juuri yhtään opetusdataa, jolloin oppiminenkin vaikeutuu. Kokeissa todettiin ristiinvalidoinnin tai AIC–kriteerin olevan parhaita valittaessa mallin kompleksisuutta.

Dynaamisten bayesverkkojen soveltaminen geenieksprression aikasarjojen mallintamiseen ei ole uusi idea [Fri98, MM99]. Aiemmin verkon oppimiseen on suunniteltu käytettäväksi *rakenteellista EM –algoritmia* [Fri98] (Structural EM), jonka etuna on mahdollisuus tehdä mallinnusta myös tilanteessa, jossa kaikkia muuttujia ei tunneta etukäteen. Tällaisia piilomuuttujia geenieksprression yhteydessä voivat olla esimerkiksi ympäristöolot tai, mallinnettaessa vain osaa geeneistä, solun sisäinen tila.

Käytännössä kaikkien bayesverkkojen rakenteen oppimiseen kehitettyjen algoritmien ongelma on niiden aikavaatimus käsiteltäessä isoja verkkoja. Seuraavassa luvussa

esitellään hyvin yksinkertainen algoritmi REVEAL, joka oppii dynaamisen bayesverkon rakenteen kohinattomassa tapauksessa. Algoritmi on myös helppo muuttaa siten, että se valitsee oikean verkon jollain edellä esitetyllä mallinvalintakriteerillä.

2.3.3 REVEAL–algoritmi

Ajan mukaan etenevän geeniekspression mallintamiseen kohinattomassa ja deterministisessä tapauksessa on kehitetty algoritmi REVEAL [LFS98]. Vaikka alkuperäinen esitys perustuukin geeniverkon rakentamiseen, yleiseltä kannalta REVEAL–algoritmi pyrkii oppimaan dynaamisen bayesverkon rakennetta [MM99].

Määritellään aluksi hieman algoritmissa 3 käytettäviä merkintöjä. Merkitään geenejä kokonaisluvuilla $1, \ldots, n$. Olkoon $\mathrm{E}_0[i]$ satunnaismuuttuja, joka kuvaa geenin i ekspressiotasoa hetkellä 0 ja $\mathrm{E}_1[i]$ satunnaismuuttuja, joka kuvaa geenin ekspressiotasoa ajanhetkellä 1. Lisäksi, kun R on joukko kokonaislukuja $1 \leq i \leq n$, merkitään $\mathrm{E}_0[R]$:llä kaikkien joukon R geenien ekspressiotasoja hetkellä 0.

REVEAL algoritmi on esitetty algoritmina 3. Yksinkertaisesti kuvattuna algoritmi hakee jokaiselle geenille i pienimmän joukon geenejä $PA(i)$, joiden ekspressio sisältää saman informaation kuin geenin i ekspressio. Koska algoritmi kokeilee kaikki mahdolliset säätelijöiden joukot jokaiselle geenille, algoritmin aikavaatimus on $O(n2^n)$.

Algoritmi 3 ei ole käytännöllinen käytännössä eikä teoriassa. Eksponentiaalinen aikavaatimus tekee algoritmin tehottomaksi, vaikka se sen antamat tulokset olisivat muuten hyviä. Eksponentiaalista aikavaatimuksta voidaan pienentää rajoittamalla säätelijöiden määrää jollain luvulla $k \leq n$. Tällöin aikavaatimus on noin luokkaa $O(n \sum_{i=1}^{k} \binom{n}{i})$, joka on sekin hyvin suuri jo pienillä k. Huomattavaa on että k ei voi olla kovin pieni, sillä geenisäätelyverkon epäsäännöllisyydestä seuraa, että joillain geeneillä on erittäin suuri joukko säätelijöitä [RSB+02], vaikka keskimäärin säätelijöitä on vain vähän.

Teoreettista aikavaatimusta pahempi ongelma on algoritmin tekemä oletus, että kaikki muutokset ovat biologisesti merkittäviä. Algoritmi valitsee geenin i sääte-

Algoritmi 3 Algoritmi REVEAL

Syöte: Kaikkien geenien edelliset E_0 ja nykyiset E_1 ekspressiotasot.

Tuloste: Säätelijät $PA(i)$ kaikilla geeneillä i.

1: **for** $i \leftarrow 1, \ldots, n$ **do**
2: **for** $k \leftarrow 1, \ldots, n$ **do**
3: **for all** geenien osajoukko R, jolla $|R| = k$. **do**
4: **if** $\mathrm{I}(\mathrm{E}_1[i], \mathrm{E}_0[R]) = \mathrm{H}(\mathrm{E}_1[i])$ **then**
5: $PA(i) \leftarrow R$
6: next i;
7: **end if**
8: **end for**
9: **end for**
10: **end for**

lijöiksi sen joukon geenejä, joiden yhteinen informaatio geenin i informaation kanssa on maksimaalinen. Koska ekspression mittaukset ovat aina epätarkkoja, ottamalla lisää geenejä säätelijöiksi saadaan aina parempi säätelijöiden joukko. Ainut mikä estää valitsemasta kaikkia geenejä jokaisen geenin säätelijöiksi on mittausten rajallisuus, jolloin mittausten pohjalta saadut arviot entropiasta ja yhteisestä informaatiosta käyvät yhteen, kunhan tarpeeksi monta säätelijää päästään valitsemaan tarpeeksi suuresta mahdollisten säätelijöiden joukosta. Koska esimerkiksi hiivalla on noin 6000 geeniä ja ekspressioaikasarjoissa on korkeintaan muutamia kymmeniä mittauksia, jokaiselle geenille on hyvin suurella todennäköisyydellä mahdollista löytää parikymmentä geeniä, jotka täydellisesti selittävät kyseisen geenin toiminnan koko aikasarjassa.

Täydellinen selitys ei yleisesti ottaen ole paras malli geenin säätelylle. Johtuen jo mittausten sisältämästä kohinasta, täydellisesti mittauksiin sopiva malli mallintaa lähinnä mittauskohinaa ja varsinainen säätelymekanismi hukkuu taustalle. Tämän vuoksi algoritmissa 3 rivillä 4 esiintyvä ehto pitää muuttaa muotoon, jossa pyritään kasvattamaan jotain mallinvalintakriteeriä, esimerkiksi luvussa 2.3.2 käsiteltyjä.

REVEAL–algoritmin alkuperäisessä esityksessä, algoritmina 3, solmun vanhempien ekspressiotasot haetaan aikasarjan edelliseltä hetkeltä ja yhteisen informaation maksimointi tapahtui raa'alla läpikäynnillä kaikkien geenijoukkojen suhteen. Kuten jo yllä viitattiin, tarpeeksi suuresta genomista tämä menetelmä löytää aina geenille sopivat säätelijäkandidaatit, mikäli mittausdataa on tarjolla vain vähän. Tällaiset "säätelijät" eivät välttämättä ole missään todellisessa suhteessa tarkasteltavaan geeniin, eikä niillä voi ennustaa tulevia havaintoja.

Jotta algoritmin tuottama verkko kuvaisi paremmin geenien todellisia säätelysuhteita, geenin säätelijöiden valintaa voidaan rajoittaa vain niihin geeneihin, jotka on havaittu vaikuttavan tarkasteltavaan geeniin keskeytyskokeessa. Keskeytysmittauksilla saadaan vähennettyä mahdollisten säätelijöiden määrää radikaalisti ja rakentuvan verkon laadun voidaan uskoa paranevan, kun verkon kaarilla tiedetään olevan jokin toiminnallinen yhteys.

Keskeytysmittausten käyttämisen riski ovat puuttuvat kokeet. Jos kaikista geeneistä ei ole keskeytysmutanttien ekspressiomittauksia, voi jokin väligeeni jäädä huomaamatta, koska sitä ei ole keskeytetty eikä sitä näin ollen oteta mukaan mahdolliseksi säätelijäksi. Tätä ongelmaa voi yrittää kiertää huomioimalla mahdollisina säätelijöinä kaikki geenit, joihin on vaikuttanut vähintään yksi niistä keskeytyksistä, jotka ovat vaikuttaneet tutkittavaan geeniin. Tämä kuitenkin tuhoaa kaarien keskeytyksellä havaitun kausaalisuuden eikä tätä menetelmää tarkastella tässä tutkielmassa tämän enempää.

Kun käytössä on sekä keskeytys– että aikasarjadataa, REVEAL voidaan muokata algoritmiksi 4. Siinä geenin i säätelijät valitaan joukosta $\Delta(i)$, johon kuuluu geenit, joiden keskeytys vaikuttaa geenin i ekspressioon. Tästä mahdollisten säätelijöiden joukosta valitaan pieni joukko geenejä, jotka kausaalisesti [Pea01] säätelevät geeniä j. Muutetun menetelmän ensimmäisessä vaiheessa käytetään keskeytysmittauksia ja toisessa vaiheessa, arvioitaessa säätelijäjoukon hyvyyttä, käytetään aikasarjoja ja luvussa 2.3.2 esiteltyä mallinvalintafunktiota. Tällä järjestelyllä saadaan hakuavaruutta rajoitettua niin kooltaan kuin myös ominaisuuksiltaan.

Algoritmi 4 Muokattu algoritmi REVEAL

Syöte: Kaikille geeneille i, geenit $\Delta(i)$, joiden keskeytys vaikuttaa i:n ekspressioon.

Funktio $Score(R, i)$, joka arvottaa säätelijät R geenille i.

Tuloste: Säätelijät $PA(i)$ kaikilla geeneillä i.

```
1: for i ← 1, ..., n do
2:    S ← 0
3:    for k ← 1, ..., |Δ(i)| do
4:       for all osajoukko R ⊂ Δ(i), jolla |R| = k do
5:          if Score(R, i) > S then
6:             PA(i) ← R
7:             S ← Score(R, i)
8:          end if
9:       end for
10:   end for
11: end for
```

Vaikka algoritmi 4 löytääkin ekspressiodatasta kausaalisen mallin, joka kertoo syy–seuraus–suhteita, se ei välttämättä paljasta säätelyn fyysistä reittiä. Koska keskeytysmittauksissa geenisäätelyn suhteet ovat romahtaneet kasaan, joukko $\Delta(i)$ sisältää geenin suoranaisten säätelijöiden lisäksi säätelijöiden säätelijät ja niiden säätelijät ja niin edespäin. Koska jokainen ylimääräinen säätelyn taso lisää säätelyn epävarmuutta, voidaan toivoa, että näistä mahdollisista säätelijöistä juuri geenin suorat säätelijät ennustavat ekspressiotason parhaiten.

Luku 3

Toteutus ja testaus

3.1 Yleistä

Algoritmit 1, 2 ja 4 toteutettiin testaustarkoituksiin käyttämällä Python–ohjelmointikieltä [Ros97]. Python on tulkittava korkean tason kieli, joka helpottaa nopeaa ohjelmankehitystä säilyttäen kuitenkin mahdollisuuden helposti optimoida tiettyjen rutiinien tilan ja ajan kulutusta.

Testauksessa käytettiin esimerkkidatana julkisia mikrosirumittauksia, joiden tulokset haettiin Stanford Microarray Database –tietokannasta [SHBK+01]. Käytetyt mittaukset sisälsivät sekä aikasarjoja [SSZ+98, DIB97, ZSV+00, YSG+02] että keskeytyskokeiden tuloksia [HMJ+00].

Luvussa 3.2 käsitellään keskeytysverkon karsimiseen perustuvan menetelmän toteutusta ja sen tuloksia eri hinnoittelumetriikoilla. Luvussa 3.3, käsitellään luvun 2.3 rakentavaan lähestymistapaan liittyviä toteutusseikkoja ja lopulta luvussa 3.4 vertaillaan molempien menetelmien tuottamia verkkoja toisiinsa.

3.2 Karsiva menetelmä

Kaikki hinnoittelumetriikat käyttävät ekspressiodatanaan niin sanottua Rosetta–datajoukkoa [HMJ+00]. Yhteistä informaatiota käyttävässä hinnoittelussa ekspressiotasot diskretoidaan positiivisiin ja negatiivisiin tapauksiin. Positiivisissa tapauksissa ekspressiosuhde on yli yhden ja negatiivisissa vastaavasti alle yhden. Tällä diskretoinnilla pyritään jakamaan luonnollinen kohina tasaisesti positiivisille ja negatiivisille tuloksille. Kosinikorrelaatio puolestaan lasketaan suoraan diskretoimattomista ekspressiosuhteen logaritmeista kaavalla (2.15).

Algoritmin 2 käyttämät sitoumapistehahmot etsittiin niiden geenien säätelyalueilta, joiden ekspressio muuttui kyseisessä keskeytyskokeeessa korkeintaan p-arvolla 0.05 Käytetty p-arvo saatiin alkuperäisjulkaisusta [HMJ+00] ja se ottaa huomioon myös yksittäiselle geenille tyypillisen satunnaisvaihtelun. Ekspressiotaan muuttaneiden geenien säätelyalueet valittiin geenin yläalueen (eli 5' pään) DNA–sekvensistä edellisen geenin loppuun. Geenien säätelyalueet rajoitettiin kuitenkin pisimmillään 1000 bp pituisiksi.

Valituista sekvensseistä haettiin yleisiä hahmoja AlignACE [HETC00] ohjelmalla. Jokaista löydettyä hahmoa etsittiin kaikilta geenien yläalueilta ja haku tuotti jokaiselle geenille i suurimman pisteluvun S_i kuvaamaan löydetyn hahmon sopivuutta kyseiselle yläalueelle. Ohjelman löytämistä hahmoista valittiin keskeytettyä geeniä edustamaan se, joka sai suurimman z arvon. Kun tarkastellaan n geeniä, joista $1, \ldots, k$ on muuttunut, saadaan z arvo laskettua kaavalla

$$z = \frac{\frac{1}{n}(\sum_{i=1}^{k} S_i - \sum_{i=k+1}^{n} S_i)}{\sum_{i=1}^{n} \frac{(S_i - \overline{S})^2}{n-1}}, \tag{3.1}$$

jossa $\overline{S}$ on kaikkien S_i keskiarvo. Rajalla, kun n ja k kasvavat suuriksi, z jakautuu keskeisen raja–arvolauseen perusteella standardinormaalisti. Yhteensä 175 keskeytetylle geenille löytyi hahmo, joka löydettiin vähintään 5 geenin yläalueelta. Muissa tapauksissa hahmo oli liian heikosti säilynyt erottuakseen huomattavasti normaalista genomista. Jokainen 175 hahmoa haettiin kaikkien geenien yläalueilta, jolloin jokaiselle hahmolle jokaisella yläalueella saatiin arvo $S_i = \log \frac{\mathrm{P(Sitoutuu)}}{\mathrm{P(Taustaa)}}$.

Tässä käytetty tapa hakea sitoumapistehahmot samalla datalla, johon kyseisiä hahmoja sovelletaan, aiheuttaa ongelmia tulosten uskottavuuden suhteen. Ilman riippumattomasta lähteestä saatua testidataa on mahdotonta sanoa, ovatko löydetyt hahmot tulosta todellisesta fyysisestä säätelystä, vai vain huonolla onnella saatu tulos mittauksissa esiintyvästä satunnaisesta kohinasta.

Algoritmia 1 testattiin käyttämällä julkaistua keskeytysmutanteista mitattua ekspressiodataa [HMJ+00]. Karsimisen lähtökohtana on keskeytysverkko, johon on otettu mukaan kaari $A \rightarrow B$ mikäli geenin B ekspression muutos on saanut p-arvon alle 0.1 geenin A keskeytysmutantissa. Verkkoon otettiin mukaan vain sellaiset geenit, joiden yksittäismutanteista oli mittauksia. Tällaisia geenejä oli käytössä 277 kappaletta. Sekvenssitietoa käyttävää hinnoittelua testatessa rajoituttiin vielä geeneihin, joille oli yllä kuvatulla tavalla löydetty säätelyhahmot. Näin geenejä oli käytössä 171.

Karsinnan lähtökohtana olleessa keskeytysverkossa oli yhteensä 2904 kaarta (912 kun geenejä oli 171). Näistä sekä kosinikorrelaatio että yhteinen informaatiometriikka säilyttivät vajaa puolet (1165 ja 1237 kaarta). Yli 50 prosenttia kaarista (684 kappaletta) oli samoja molemmilla tavoilla karsituissa verkoissa.

Metriikka	Kosinikorrelaatio	Yhteinen informaatio	Sekvenssidata
Kosinikorrelaatio	1165	684	318
Yhteinen informaatio	—	1237	345
Sekvenssidata	—	—	595

Taulukko 3.1: Eri kaarihinnoittelumenetelmien tuottamien kaarien määrät ja kuinka paljon verkoilla oli yhteisiä kaaria. Kaikissa kolmessa verkossa on 235 yhteistä kaarta.

Sekvenssidataa hyödyntävä karsinta, joka lähti muita menetelmiä pienemmästä verkosta, säilytti verkossa yhteensä 595 kaarta. Näistä kaarista 345 oli samoja kuin yhteisen informaation verkossa (jossa oli 1084 kaarta pienen lähtöverkon alueella) ja 318 kaarta oli yhteisiä kosinikorrelaatiolla lasketun verkon kanssa (maksimi 975). Kaikkien kolmen verkon kanssa yhteisiä kaaria oli yhteensä 235 kappaletta.

Solmujen tuloasteet vaihtelevat runsaasti kullekin geenille. Taulukossa 3.2 on tun-

nuslukuja eri verkkojen tuloasteiden jakaumista. Taulukon luvut ovat 10–prosenttipiste, mediaani (50–prosenttipiste), aritmeettinen keskiarvo, 90–prosenttipiste ja maksimi (100–prosenttipiste). Tilastojen laskennassa on käytetty pelkästään niitä solmuja, joihin on liittynyt vähintään yksi kaari.

Taulukossa 3.2 kaikkien verkkojen tuloasteet antavat viitteitä geenisäätelyverkon epäsäännöllisestä rakenteesta, jossa suurimmalla osalla geenejä on vähän säätelijöitä (puolella on 3 tai alle) ja osalla on hyvin paljon säätelijöitä (esimerkiksi kosinikorrelaatiolla karsitussa verkossa 10 prosentilla geeneistä on 15–68 säätelijää).

Tuloasteet	10%	50%	Keskiarvo	90%	100%
Kosinikorrelaatio	1	3	5.9	15	68
Yhteinen informaatio	1	3	6.3	14	50
Sekvenssidata	1	3	4.1	9	21

Taulukko 3.2: Geenien säätelijöiden määrä.

Taulukossa 3.3 on vastaavia lukuja geenien lähtöasteille. Lähtöasteilla numerot ovat pienempiä ja asteiden jakauma on tasaisempi, mutta tässäkin tapauksessa näkyy selvänä jakauman vinoutuma, joka tuottaa hyvin suuria asteita.

Lähtöasteet	10%	50%	Keskiarvo	90%	100%
Kosinikorrelaatio	1	3	4.4	9	36
Yhteinen informaatio	2	4	4.7	9	54
Sekvenssidata	1	2	4.0	9	28

Taulukko 3.3: Säädeltävien geenien määrä.

3.3 Rakentava menetelmä

Luvussa 2.3 käsitelty algoritmi 4 toteutettiin käyttämällä yhtälön (2.29) AIC mallinvalintakriteeriä ja useita eri aikasarjoja [SSZ+98, DIB97, ZSV+00, YSG+02]. Yhteensä nämä aikasarjat sisälsivät 126 REVEAL–algoritmille käypää aikaviipaletta.

Algoritmissa 4 käytettävät tunnetut välilliset säätelijät $\Delta(i)$ saatiin keskeytyskokeista [HMJ+00] valitsemalla geenit, joden keskeytyksessä geenin i p–arvo oli pienempi kuin 0.05. REVEAL algoritmilla tarkasteltiin ainoastaan geenejä, jotka oli keskeytetty ja joille löytyi sekä aikasarja että keskeytysmittausdataa. Tällaisia geenejä oli yhteensä 195 kappaletta. Periaatteessa ei olisi ollut mahdotonta hakea tällä menetelmällä säätelijöitä hiivan kaikille geeneille, mutta ajan säästämiseksi säätelyn kohteet rajoitettiin samaan joukkoon kuin mahdolliset säätelijät.

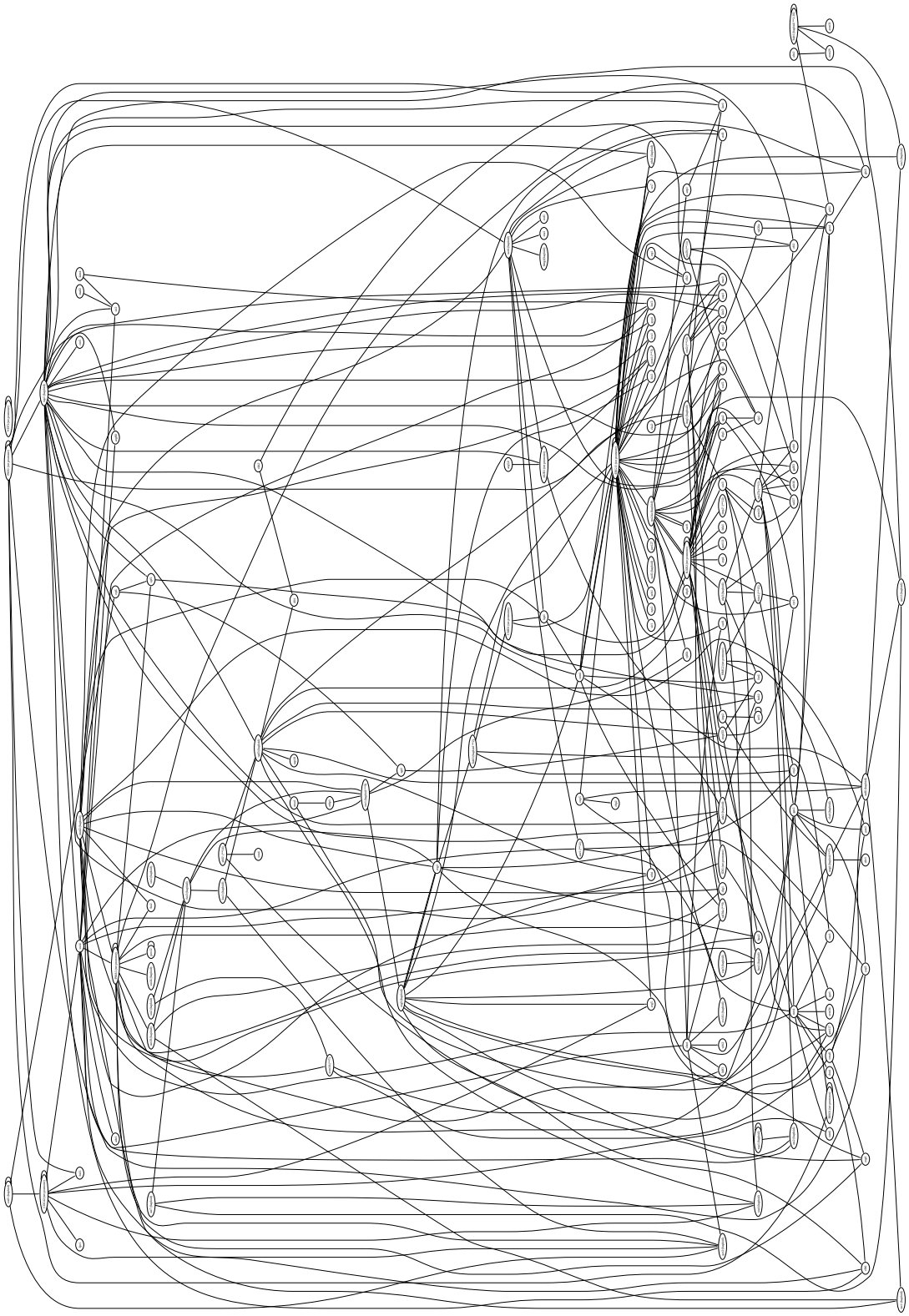
Tarkasteltavien geenien rajoittamisen lisäksi laskentaa rajoitettiin haarautumisasteella. Ne geenit, joihin keskeytyskokeissa vaikutti enemmän kuin 25 muuta geeniä, jätettiin pois laskennasta. Lopuille geeneille haettiin korkeintaan 10 säätelijää. Tämä raja ei ole todellinen rajoitus tällä datamäärällä, sillä kymmenen säätelijää omaavalle solmulle pitäisi arvioida $2^{10} = 1024$ parametriä, mikä on mahdotonta vain 126 havainnosta.

Itse asiassa yhtälöstä (2.29) huomataan, että binäärimuuttujan paras AIC–arvo kiinteällä m havainnolla on korkeintaan m, koska data voidaan kuvata ilman mallia m bitin jonona. Koska datan AIC–arvo on rajoitettu, on myös mallin parametrien määrä rajoitettu. Käyttämällä 2 kantaista logaritmia ja arvioimalla solmun isien p määrää yhtälössä (2.29), saadaan

$$\begin{aligned} AIC(D) &\leq 2^p \log e \leq m \\ \Leftrightarrow \quad p + \log\log e &\leq \log m \\ \Leftrightarrow \quad p &\leq \log m - \log\log e, \end{aligned} \tag{3.2}$$

jossa m on käytössä olevan opetusdatajoukon koko. Epäyhtälöistä (3.2) nähdään, että 126 havainnolla AIC kriteerin mukaan optimaalisen binääriarvoisen bayesverkon solmujen tuloaste on korkeintaan 6. Testeissä algoritmin 4 tuottaman verkon solmujen tuloasteet olivat korkeintaan 4.

Algoritmin 4 tuottama geenisäätelyverkko on esitetty kuvassa 3.1. Verkko sisältää 159 kaariin liittynyttä solmua, joista 79:llä on lähteviä ja 143:llä tulevia kaaria. Yhteensä verkossa on 340 kaarta. Verkon lähtö- ja tuloasteiden tilastoja on näkyvillä taulukossa 3.4. Taulukosta selviää, että lähtöasteiden jakauma vastaa kohtalaisen hy-



Kuva 3.1: Algoritmin 4 tuottama säätelyverkko.

Kaarellisista solmuista	10%	50%	Keskiarvo	90%	100%
Tuloaste	1	2	2.4	4	4
Lähtöaste	1	2	4.3	10	34

Taulukko 3.4: Säädeltävien geenien määrä.

vin nykyistä käsitystä geenisäätelyverkon rakenteesta eli verkon asteiden jakaumalla on vahva häntä suurille arvoille. Rakennetun verkon solmujen tuloasteet jakautuvat kuitenkin varsin tasaisesti, vaikka odotusarvoisesti niidenkin pitäisi noudattaa samankaltaista jakaumaa lähtöasteiden kanssa.

Tuloasteen rajoitteisuus johtuu verkon rakennusvaiheessa valitusta AIC–mallinvalintakriteeristä, joka rankaisee monimutkaisista malleista. Erityisesti algoritmin 4 tapaisessa bayesverkon rakennuksessa AIC–kriteeri rankaisee paikallisesta kompleksisuudesta, jota ilmenee juuri silloin kun geenillä on monimutkainen säätelymekanismi. Monimutkaisen mekanismin selvittämiseen tarvitaan luonnollisesti paljon havaintoja, jotta mekanismin oikea rakenne saadaan selvitettyä. Voidaankin olla melko varmoja siitä, että käytössä ollut 126 mittauksen datajoukko ei ollut riittävä tuloasteen jakauman tarkkaan arviointiin.

3.4 Rakentavan ja karsivan lähestymistavan erot

Luvun 2.2 karsiva ja luvun 2.3 rakentava lähestymistapa tuottaa hyvin paljon toisistaan poikkeavia geenisäätelyverkon malleja. Suurin ero lienee verkkojen kaarten tulkinnassa. Karsitussa verkossa jokaista kaarta kohdellaan yksilönä, jonka lisäys ja poisto koskettaa vain tätä yhtä kaarta. Rakennetussa verkossa kaaret puolestaan otetaan mukaan ryhmänä, jonka ennustuskyky voi olla parempi kuin sen jäsenten ennustuskykyjen summa.

Kaarten yksilöllisen ja ryhmitellyn tarkastelun ero tulee esiin välittömästi käytettäessä luvun 2.1.4 kuvauspituusmittaa. Luvun 2.2 karsiva menetelmä tuottaa hyvin suuria verkkoja, joiden kuvauspituudet ovat väliltä 10^5–10^{20} kun taas luvun 2.3 ra-

kentava menetelmä tuottaman verkon kuvauspituus on noin 35500. Tämä ero johtuu yhtälön (2.13) eksponentiaalisesta termistä, joka kasvattaa suuren sisääntuloasteen omaavien geenien hinnan valtavan suureksi. Samankaltainen eksponentiaalinen termi yhtälön (2.29) AIC–kriteerissä rajoittaa rakennettavan verkon tuloasteita, joten on luonnollista että rakennetun verkon kuvauspituus on lyhyt.

Rakennetun ja karsittujen verkkojen rakenteen merkittävä ero on verkon kaarien määrä. Kosinikorrelaatiolla, yhteisellä informaatiolla ja sekvenssidatalla karsituissa verkoissa on vastaavasti 1165, 1237 ja 595 kaarta kun algoritmin 4 rakentamassa verkossa oli ainoastaan 340 kaarta. Kaikille neljälle verkolle yhteisiä kaaria oli vain kaksi (ECM18 $\rightarrow$ RAD6 ja ERG4 $\rightarrow$ UBR1), joista kumpikaan ei vaikuttanut biologisesti mielekkäiltä. Kokeet vahvistavat yhtälöstä (2.13) nähtävän ominaisuuden, että informaatioteoriaan pohjautuva verkkojen vertailumenetelmä rankaisee hyvin paljon verkon paikallisesta monimutkaisuudesta.

Eräs huomattava ero eri tavalla löydettyjen säätelyverkkojen ominaisuuksissa on solmujen astejakaumat. Karsituissa verkoissa solmujen tulo– ja lähtöasteiden jakaumat ovat hyvin vinoutuneet, kuten biologiselta verkolta odottaakin, mutta rakennetun verkon tuloasteet ovat jakautuneet hyvin pienelle alueelle. Tämä ero johtuu kaarten tulkinnan eroista, jonka perusteella karsitussa verkossa jokainen säätelijä voi selittää saman osan geenin ekspressiokäyttäytymisestä kun taas rakennetussa verkossa kaikki kaaret selittävät geenin toimintaa yhdessä. Koska käytössä on vain rajoitetusti dataa, monimutkaisesti säädeltyjen geenien kaikkia säätelijöitä on hyvin vaikea löytää. Tämä tilanne paranee ajan myötä kun käyttöön saadaan enemmän mittausdataa, mutta tuskin koskaan pystytään tekemään tarpeeksi mittauksia, jotta joidenkin geenien 2^{20} parametriä pystyttäisiin tyydyttävästi approksimoimaan ilman lisäoletuksia.

Luku 4

Yhteenveto

Biologisten organismien geeniekspression säätely on hyvin monimutkainen prosessi ja sen mekanismien löytäminen suuressa mittakaavassa on hyvin vaikeaa. Nykyaikaisilla mikrosiruilla voidaan mitata tuhansien geenien ekspressiotasot yhdellä kertaa, mutta yksi mikrosiru tuottaa vain yhden pisteen monituhatulotteisessa avaruudessa kun perinteisen tilastotieteen kannalta olisi parempi saada tuhansia pisteitä yhdessä ulottuvuudessa. Tämä mikrosirutekniikan hyvin syvällinen ominaisuus lisää merkittävästi haastetta geenisäätelyn analyysiin.

Pyrittäessä etsimään geenisäätelyn suhteita mikrosirumittauksilla voidaan tehdä kahdenlaisia koesarjoja. Geeniekspressiota voidaan mitata joko aikasarjana, jolloin synkronoidusta solupopulaatiosta otetaan näytteitä tiettyinä ajanhetkinä, tai keskeytyskokeina, joissa yksi tai useampi organismin geeneistä on keinotekoisesti keskeytetty. Jälkimmäinen, keskeytyskokeisiin perustuva menetelmä antaa mahdollisuuden geenisäätelyn kausaaliseen mallinnukseen. Kausaalisessa mallissa voidaan sanoa "Koska geeni A on keskeytetty, geenin B ekspressio poikkeaa normaalista."

Kausaalinen malli geenisäätelyverkosta olisi hyvin hyödyllinen lähtökohta jatkettaessa tutkimusta geenisäätelyn fyysisiin mekanismeihin. Kausaliteetin mallintaminen on kuitenkin hyvin vaikeaa ja kallista ympäristössä, jossa jokaista mahdollista väliintuloa kohti pitää tehdä hidas ja kallis laboratoriokoe. Vaikka aikaa ja rahaa

olisikin käytössä rajoittamattomasti, kaikkia geenejä ei edes voida keskeyttää ilman että tutkittava organismi kuolee.

Hieman kausaliteettia muistuttava ja paljon helpompi ja halvempi menetelmä geenisäätelyn mallintamiseen ovat dynaamiset mallit. Dynaamisella mallilla voidaan antaa ennusteita kuten "Geenin A ekspressio on koholla, joten geenin B ekspressio tulee kohoamaan". Tällaiset dynaamiset mallit ovat hyödyllisiä pyrittäessä ennustamaan tulevia tapahtumia, mutta ne eivät anna kausaliteetin kaltaista lähtökohtaa säätelyn fyysisen mekanismin löytymiseen. Kahden samalla tavalla ekspressoituvan geenin säätelymekanismit voivat olla hyvinkin erilaisia, mitä ei pelkällä dynaamisella mallinnuksella pysty havaitsemaan.

Myös keskeytyskokeilla on ongelmansa. Eräs suurimmista ongelmista on keskeytyskokeissa osittain ilmenevä transitiivinen sulkeuma. Jos keskeytyskokeissa havaitaan että geeni A vaikuttaa geeniin B ja geeni B vaikuttaa geeniin C niin niissä usein havaitaan myös että geeni A vaikuttaa geeniin C. Tällaiset transitiiviset kaaret tekevät yksinkertaisista keskeytyskokeista rakennetusta verkosta hyvin monimutkaisen. Kuitenkin olisi hyödyllistä mahdollisuuksien mukaan välttää turhaan raportoituja säätelysuhteita, joten tällaiset transitiiviset geenisäätelyverkon kaaret tulisi jollakin tapaa poistaa.

Eräs tapa poistaa transitiiviset kaaret on laskea verkon transitiivinen reduktio. Tämä menetelmä kuitenkin olettaa, että keskeytysmittaukset ovat olleet virheettömiä ja transitiivisuus on levinnyt niin pitkälle kuin se tulee leviämään. Vaihtoehtoinen menetelmä transitiiviselle reduktiolle on säilyttää geenikeskeytyksistä saadussa verkossa vain niin sanotut min–max –polut, jonkin sopivan hinnoittelun suhteen. Min–max –polku kahden solmun välillä on niiden välinen halvin polku, jossa polun hinta on sen kalleimman kaaren hinta. Tällä tavalla kahdesta vaihtoehtoisesta reitistä valitaan se, jonka heikoin lenkki on vahvin.

Min–max –polun käsittelyssä ratkaisevaa on kaarten hinnoittelufunktion valinta. Funktion pitäisi antaa luotettavalle kaarelle halpa hinta ja epävarmalle suuri. Hyvän hinnoittelufunktion suunnitteleminen on hyvin vaikeaa. Ilman vahvaa biologista

tietämystä voidaan helposti ehdottaa hinnoittelussa käytettäväksi normaaleja etäisyysmittoja geenien ekspressioprofiilien välillä. Tällaisia mittoja ovat esimerkiksi kosinikorrelaatio ja yhteinen informaatio. Lisäksi apuna voidaan käyttää genomisekvenssistä saatuja ominaisuuksia kuten proteiinien sitoumapisteiden löytymistä geenien yläalueilta.

Transitiivisten kaarien ongelma voidaan kiertää valitsemalla geenin säätelijöiksi vain osajoukko kaikista mahdollisista kausaalisista säätelijöistä. Valitettavasti tässäkin lähestymistavassa törmätään ongelmaan oikean valintakriteerin valinnassa.

Ylipäänsä geenisäätelyverkkojen oppimisen ongelmana on tavoitteen epämääräisyys. Perinteiset informaatioteoriaan ja tilastollisen mallin valintaan kehitetyt menetelmät eivät ole kelvollisia arvioitaessa geenisäätelyverkkojen hyvyyttä. Usein käytetyt mallinvalintakriteerit kuten AIC, BIC ja MDL toimivat ainoastaan kun käytössä on runsaasti dataa. Geenisäätelyä mallinnettaessa dataa on kuitenkin käytössä hyvin vähän ja malliavaruus on hyvin monimutkainen. Kaiken lisäksi todellisen mallin uskotaan olevan monimutkainen eikä yksinkertainen kuten kaikki tilastolliset mallinvalintakriteerit olettavat. Yksinkertainen malli on hyvä pyrittäessä ennustamaan tulevia havaintoja mutta sen rakenne ei välttämättä vastaa mittaustulokset todellisuudessa tuottanutta järjestelmää. Hyvä geenisäätelymallin valintakriteeri sisältää välttämättä runsaasti tietoa geenien toiminnasta ja, mikäli tietoa ei tule tarpeeksi datasta, biologinen tietämys pitää koodata mallinvalintakriteeriin ulkopuolelta kokeneen biologin ja biologisten tietokantojen avulla.

Eri lähestymistavat geenisäätelyverkon oppimiseen tuottavat luonnollisesti hyvin erilaisia ratkaisuja. Verkkojen karsinnassa käytettävien hinnoittelufunktioiden järkevyyttä on lähes mahdotonta arvioida ilman syvällistä tuntemusta käsillä olevan organismin geenisäätelystä. Toisaalta voidaan epäillä, onko verkkoa rakentavasti kokoavan algoritmin käyttämä menetelmä tyydyttävä, kun se tuottaa verkkoja, joiden solmujen asteluvut poikkeavat merkittävästi oletettavasti oikeista arvoista.

Vaikka geenisäätelyä pyrittäisiinkin mallintamaan jollain boolen funktioita yksinkertaisemmalla mallilla, on kyseenalaista kuinka pitkälle suurisuuntaisilla ja kohdis-

tamattomilla kokeilla päästään. Mitä ilmeisemmin tietojenkäsittelyn osuus geenisäätelyverkkojen päättelyssä tulee rajoittumaan koesuunnitteluun, verkon laajamittaisten ominaisuuksien analysointiin sekä kokeita tekevän biologin avustamiseen.

Lähteet

AED+00 Alizadeh, A. A., et al., Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature* 403, 6769 (2000), 503–511.

AG00 Van Allen, T., Greiner, R., Model Selection Criteria for Learning Belief Nets: An Empirical Comparison. *Proc. 17th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 2000, 1047–1054.

Aka78 Akaike, H., A Bayesian analysis of the minimum AIC procedure. *Annals of the Institute of Statistical Mathematics* 30, A (1978), 9–14.

AKMM98 Akutsu, et al., Identification of Gene Regulatory Networks by Strategic Gene Disruptions and Gene Overexpressions. *SODA: ACM-SIAM Symposium on Discrete Algorithms (A Conference on Theoretical and Experimental Analysis of Discrete Algorithms)*. 1998.

Bar02 Barabási, A.-L., *Linkit: Verkostojen uusi teoria*. Terra Cognita, 2002. Alkuperäisteksti: LINKED: The New Science of Networks. How Everything is Connected to Everything Else and What it Means for Science, Business and Everyday Life.

BBS01 Birnbaum, K., Benfey, P. N., Shasha, D. E., cis element/transcription factor analysis (cis/TF): a method for discovering transcription factor/cis element relationships. *Genome research* 11, 9 (2001), 1567–1573.

BBS02 Benos, P. V., Bulyk, M. L., Stormo, G. D., Additivity in protein-DNA interactions: how good an approximation is it? *Nucleic Acids Res* 30, 20 (2002), 4442–4451.

BE94 Bailey, T., Elkan, C., Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proceedings of the International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, Menlo Park, California, 1994, 28–36.

BF01 Barash, Y., Friedman, N., Context-specific Bayesian clustering for gene expression data. *RECOMB*. 2001, 12–21.

Bio03 Bioinformatiikan sanasto. WWW–sivu, 22.1.2003. `http://www.csc.fi/molbio/sanasto/html/`.

BJVU98 Brazma, A., et al., Predicting gene regulatory elements in silico on a genomic scale. *Genome research* 8, 11 (1998), 1202–1215.

BLS01 Bussemaker, H. J., Li, H., Siggia, E. D., Regulatory element detection using correlation with expression. *Nature Genetics* 27, 2 (2001), 167–171.

BPSS Brazma, A., et al., A quick introduction to elements of biology — cells, molecules, genes, functional genomics, microarrays. WWW–sivu 13.2.2002. `http://industry.ebi.ac.uk/~brazma/Biointro/biology.html`.

CBE01 Chiang, D. Y., Brown, P. O., Eisen, M. B., Visualizing associations between genome sequences and gene expression data using genome-mean expression profiles. *Bioinformatics* 17 Suppl 1 (2001), S49–S55.

CD99 Carlson, J. M., Doyle, J., Highly optimized tolerance: A mechanism for power laws in designed systems. *Physical Review E* 60, 2 (1999), 1412–1427.

CD01 Coffman, J. A., Davidson, E. H., Genetic Networks. Web-sivu 14.2.2002. Ensyclopedia of Life Sciences. Nature Publishing Group., 2001. `http://www.els.net`.

CHC99 Chen, T., He, H. L., Church, G. M., Modeling gene expression with differential equations. *Pac Symp Biocomput* (1999), 29–40.

Chi96 Chickering, D. M., Learning Bayesian Networks is NP-Complete. D. Fisher, H. Lenz (toim.), *Learning from Data: Artificial Intelligence and Statistics V*, Springer-Verlag. 1996, 121–130.

CLR89 Cormen, T. H., Leiserson, C. E., Rivest, R. L., *Introduction to Algorithms*. The MIT Press and McGraw-Hill Book Company, 1989.

CT91 Cover, T. M., Thomas, J. A., *Elements of Information Theory*. Wiley Series in Telecommunications, John Wiley & Sons, New York, NY, USA, 1991.

DIB97 DeRisi, J. L., Iyer, V. R., Brown, P. O., Exploring the Metabolic and Genetic Control of Gene Expression on a Genomic Scale. *Science* 278, 5338 (1997), 680–686.

Dit01 Ditlevsen, P., Internetin herkkä ydin. *Tieteen Kuvalehti* , 6 (2001).

DK89 Dean, T., Kanazawa, K., A model for reasoning about persistence and causation. *Computational Intelligence* 5, 3 (1989), 142–150.

DWFS99 D'Haeseleer, P., et al., Linear modeling of mRNA expression levels during CNS development and injury. *Pac Symp Biocomput* (1999), 41–52.

ESBB98 Eisen, M. B., et al., Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of the United States of America* 95, 25 (1998), 14 863–14 868.

Exp02 Experimental Procedures. WWW–sivu, 22.10.2002. `http://www.ym.edu.tw/excellence/HBP/HBP_CP4/procedure.htm`.

Far Farabee, M., On-Line Biology Book. WWW–sivu 13.2.2002. `http://gened.emc.maricopa.edu/bio/bio181/BIOBK/BioBookTOC.html`.

FB02 Featherstone, D. E., Broadie, K., Wrestling with pleiotropy: genomic and topological analysis of the yeast gene expression network. *Bioessays* 24, 3 (2002), 267–274.

FGW+95 Fraser, C. M., et al., The minimal gene complement of Mycoplasma genitalium. *Science* 270, 5235 (1995), 397–403.

FLNP00 Friedman, N., et al., Using Bayesian networks to analyze expression data. *J Comput Biol* 7, 3-4 (2000), 601–620.

Flo62 Floyd, R. W., Algorithm 97: Shortest path. *Communications of the Association for Computing Machinery* 5 (1962), 345.

FMR98 Friedman, N., Murphy, K., Russell, S., Learning the Structure of Dynamic Probabilistic Networks. G. F. Cooper, S. Moral (toim.), *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers, Inc., San Francisco, 1998.

FNP99 Friedman, N., Nachman, I., Peér, D., Learning Bayesian Network Structure from Massive Datasets: The "Sparse Candidate" Algorithm. *Proceedings Fifteenth Conference on Uncertainty in Artificial Intelligence*. 1999, 206–215.

Fri98 Friedman, N., The Bayesian Structural EM Algorithm. G. F. Cooper, S. Moral (toim.), *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI '98)*. Morgan Kaufmann, San Francisco, CA, 1998, 129–138.

GLCV01 Ge, H., et al., Correlation between transcriptome and interactome mapping data from Saccharomyces cerevisiae. *Nature Genetics* 29, 4 (2001), 482–486.

GMSU89 Gries, D., et al., An Algorithm for Transitive Reduction of an Acyclic Graph. *Science of Computer Programming* 12, 2 (1989), 151–155.

HB00 Holmes, I., Bruno, W. J., Finding regulatory elements using joint likelihoods for sequence and expression profile data. *Proceedings of the International Conference on Intelligent Systems for Molecular Biology (ISMB)* 8 (2000), 202–210.

Hec95 Heckerman, D., A tutorial on learning with bayesian networks. Teknikinen Raportti MSR-TR-95-06, Microsoft Research, Redmond, Washington, 1995.

HETC00 Hughes, J. D., et al., Computational identification of cis-regulatory elements associated with groups of functionally related genes in Saccharomyces cerevisiae. *J Mol Biol* 296, 5 (2000), 1205–1214.

HGC94 Heckerman, D., Geiger, D., Chickering, D. M., Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *KDD Workshop*. 1994, 85–96.

HLS94 Heumann, J. M., Lapedes, A. S., Stormo, G. D., Neural Networks for Determining Protein Specificity and Multiple Alignment of Binding Sites. *Proceedings of the International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, Menlo Park, California, 1994, 188–194.

HMJ+00 Hughes, T. R., et al., Functional discovery via a compendium of expression profiles. *Cell* 102, 1 (2000), 109–126.

Huf52 Huffman, D. A., A Method for the Construction of Minimum-Redudancy Codes. *Proceedings of the Institute of Radio Engineers* (1952), 1098–1101.

JCCS01 Jakt, L. M., et al., Assessing clusters and motifs from gene expression data. *Genome research* 11, 1 (2001), 112–123.

Kru56 Kruskal, J. B., On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society* 7 (1956), 48–50.

LFS98 Liang, S., Fuhrman, S., Somogyi, R., Reveal, a general reverse engineering algorithm for inference of genetic network architectures. *Pac Symp Biocomput* (1998), 18–29.

MM99 Murphy, K., Mian, S., Modelling gene expression data using dynamic Bayesian networks, 1999.

MMH+01 McPherson, J. D., et al., A physical map of the human genome. *Nature* 409, 6822 (2001), 934–941.

Mur02 Murphy, K., *Dynamic Bayesian Networks: Representation, Inference and Learning*. väitöskirja, University of California at Berkeley, 2002.

Occ24 of Occam, W., Quodlibeta, 1324. Katso myös: The Oxford Dictionary of Quotations, 4.ed, toim Angela Partington, 1996.

Pea88 Pearl, J., *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA., 1988.

Pea01 Pearl, J., *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2001.

PSC01 Pilpel, Y., Sudarsanam, P., Church, G. M., Identifying regulatory networks by combinatorial analysis of promoter elements. *Nature Genetics* 29, 2 (2001), 153–159.

PV91 Pearl, J., Verma, T. S., A Theory of Inferred Causation. J. F. Allen, R. Fikes, E. Sandewall (toim.), *KR'91: Principles of Knowledge Representation and Reasoning*. Morgan Kaufmann, San Mateo, California, 1991, 441–452.

Ris76 Rissanen, J., Generalized Kraft's inequality and arithmetic coding. *IBM Journal of Research and Development* 20 (1976), 198–203.

Ris78 Rissanen, J., Modeling by shortest data description. *Automatica* 14 (1978), 465–471.

Ris86 Rissanen, J., Stochastic complexity and modeling. *Annals of Statistics* 14, 3 (1986), 1080– 1100.

Ris99 Rissanen, J., Hypothesis Selection and Testing by the MDL Principle. *The Computer Journal* 42, 4 (1999), 260–269.

Ros97 van Rossum, G., Scripting the Web with Python. *World Wide Web Journal* 2, 2 (1997).

RSB+02 Rung, J., et al., Building and analysing genome-wide gene disruption networks. *Bioinformatics* 18, 90002 (2002), 202S–210.

Sac02 Saccharomyces Genome Deletion Project, 1.3.2002. `http://www-sequence.stanford.edu/group/yeast_deletion_project/`.

SBS+02 Segal, E., et al., From Promoter Sequence to Expression: A Probabilistic Framework. G. Myers, S. Hannenhalli, S. Istrail, P. Pevzner, M. Waterman (toim.), *Proceedings of the Sixth Annual International Conference on Computational Biology*. ACM Press, 2002, 263–272.

Sch78 Schwartz, G., Estimating the dimension of a model. *Annals of Statistics* 6 (1978), 461–464.

Sha48 Shannon, C., A mathematical theory of communication. *Bell System Technical Journal* 27 (1948), 379–423, 623–656.

SHBK+01 Sherlock, G., et al., The Stanford Microarray Database. *Nucleic Acids Res* 29, 1 (2001), 152–155.

SL98 Szallasi, Z., Liang, S., Modeling the normal and neoplastic cell cycle with "realistic Boolean genetic networks": their application for understanding carcinogenesis and assessing therapeutic strategies. *Pac Symp Biocomput* (1998), 66–76.

SSGE82 Stormo, G. D., et al., Use of the 'Perceptron' algorithm to distinguish translational initiation sites in E. coli. *Nucleic Acids Res* 10, 9 (1982), 2997–3011.

SSZ+98 Spellman, P. T., et al., Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Molecular biology of the cell* 9, 12 (1998), 3273–3297.

Sto74 Stone, M., Cross-validatory choice and assesment of statistical predictions. *Journal of the Royal Statistical Society Series B* 36 (1974), 111–147.

Sto00 Stormo, G. D., DNA binding sites: representation and discovery. *Bioinformatics* 16, 1 (2000), 16–23.

The03 The Biology Project: Prokaryotes, Eukaryotes, & Viruses Tutorial. WWW–sivu, 22.1.2003. `http://www.biology.arizona.edu/cell_bio/tutorials/pev/main.html`.

Tur52 Turing, A. M., The Chemical Basis of Morphogenesis. *Phil. Trans. Roy. Soc. London* B, 237 (1952), 37–72.

VAM+01 Venter, J. C., et al., The sequence of the human genome. *Science* 291, 5507 (2001), 1304–1351.

VBJ+00 Vilo, J., et al., Mining for putative regulatory elements in the yeast genome using gene expression data. *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB–2000)*. AAAI Press, European Bioinformatics Institute, EMBL Outsta-

tion, Hinxton, Cambridge, United Kingdom. vilo@ebi.ac.uk, 2000, 384–394.

Vil98 Vilo, J., Discovering frequent patterns from strings. Tekninen Raportti C-1998-9, University of Helsinki, Department of Computer Science, Helsinki, 1998.

Wag01 Wagner, A., How to reconstruct a large genetic network from n gene perturbations in fewer than n^2 easy steps. *Bioinformatics* 17, 12 (2001), 1183–1197.

Wag02 Wagner, A., Estimating coarse gene network structure from large-scale gene perturbation data. *Genome research* 12, 2 (2002), 309–315.

WC53 Watson, J., Crick, F., A Structure for DNA. *Nature* 171 (1953), 737–738.

WSA+99 Winzeler, E. A., et al., Functional characterization of the S. cerevisiae genome by gene deletion and parallel analysis. *Science* 285, 5429 (1999), 901–906.

WWS99 Weaver, D. C., Workman, C. T., Stormo, G. D., Modeling regulatory networks with weight matrices. *Pac Symp Biocomput* (1999), 112–123.

YBS01 Yang, Y. H., Buckley, M. J., Speed, T. P., Analysis of cDNA microarray images. *Brief Bioinform* 2, 4 (2001), 341–349.

YSG+02 Yoshimoto, H., et al., Genome-wide analysis of gene expression regulated by the calcineurin/Crz1p signaling pathway in Saccharomyces cerevisiae. *J Biol Chem* 277, 34 (2002), 31 079–31 088.

ZL77 Ziv, J., Lempel, A., A universal algorithm for sequencial data compression. *IEEE Transactions on Information Theory* IT-23 (1977), 337–343.

ZSV+00 Zhu, G., et al., Two yeast forkhead genes regulate the cell cycle and pseudohyphal growth. *Nature* 406, 6791 (2000), 90–94.

ZZ99 Zhu, J., Zhang, M. Q., SCPD: a promoter database of the yeast Saccharomyces cerevisiae. *Bioinformatics* 15, 7-8 (1999), 607–611.

Algoritmit

Kuvat

Taulukot