Geenien etsintä

Jukka Kohonen
Jukka.Kohonen@cs.helsinki.fi

Tiedon louhinta biomolekyyliaineistoista
Helsingin yliopisto, tietojenkäsittelytieteen laitos
Raportti C–2003–52, s. 89–99, marraskuu 2003

Tiivistelmä

Geenien etsintä on tehtävä, jossa DNA-sekvenssiin eli merkeistä C, A, G ja T koostuvaan merkkijonoon liitetään tieto siitä, missä kohdissa sijaitsee geenejä ja missä ei. Se on erikoistapaus yleisemmästä merkkijonon annotointiongelmasta.

Laskennallinen geenien etsintä perustuu olemassaolevaan, biologisesti perusteltuun informaatioon: erityisiin aloitus- ja lopetussignaaleihin, geenisekvenssien sisältöä koskevaan tilastotietoon sekä ennestään tunnettuihin geenisekvensseihin.

Tyypilliset menetelmät perustuvat piilo-Markov-malleihin. Mallin näkyvät tilat (tuotetut merkit) vastaavat havaittua DNA-sekvenssiä, piilotilat puolestaan haluttua tietoa geenien sijainneista. Mallin parametrit estimoidaan opetusdatasta esimerkiksi EM-algoritmilla. Opetettua mallia voidaan sitten käyttää geenien etsintään uusista DNA-sekvensseistä esimerkiksi Viterbin algoritmilla.

1 Johdanto

Eliöiden DNA-sekvenssit ovat aakkostosta {C,A,G,T} rakentuneita merkkijonoja. Ne eivät kuitenkaan ole homogeenisia, vaan koostuvat *jaksoista*, joilla on erilaisia tehtäviä. Tärkeimpiä jaksoja ovat *geenit*. Karkeasti voidaan sanoa, että geeni on DNA:n tarkkarajainen kappale, joka tuottaa (*koodaa*) tiettyä proteiinia. Esimerkiksi ihmisen genomi on pituudeltaan noin 3200 Mb.[1] Tästä noin 2000 Mb on täysin geenien ulkopuolista materiaalia. Lopuista 1200 Mb:stäkin vain noin 48 Mb on varsinaisia geenejä, muu on erilaista geeneihin liittyvää materiaalia kuten introneja [Son03].

Geenien etsintä (gene finding) on tehtävä, jossa annetuista DNA-sekvensseistä on löydettävä, missä paikoissa on geeni ja missä ei. Kyseessä on erikoistapaus *annotaatio-ongelmasta*: merkkijono on jaettava osiin ja kukin osa luokiteltava johonkin annetuista luokista. Kuvassa 1 on esimerkki annotaatiosta, jossa luokkia on kolme:

- *eksonit* eli varsinaiset koodaavat jaksot,
- geenin eksonien väliset *intronit* sekä
- geenienväliset (intergenic) jaksot.

Yleisemmin annotoinnissa voi olla käytettävissä enemmänkin luokkia, esimerkiksi pseudogeeni, transposoni, mikrosatelliitti jne. [Son03]

[1] Mb = megabase, miljoona emästä (merkkijonon merkkiä).

```
AGCGGATCCCCCGGTGGCCTCATGTCGCGCAGTGGAACCGATCCTCAGCAACGCCAGCAGGCGTCAGAGG
000000000000000000000CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

CGGACGCCGCAGCAGCAACCTTCCGGGCAAACGGTAACTGCACCGCGGCAGGGACTCGCTGGGGCGCGGA
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII

GCCGAGCCCTCCCCTTCCTTAGGAAGCTTTCGTCCCCCTCCGAAGGTTGGAACGCTCATCCCGAGCCAGA
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII

CCGACAAGGCGTACAGTCTGCAGGCCTCTACGAGCAGCAGGCCAATTGGCGCTGGGAAAGTCCAATCCTG
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII

GGCCTCTAGCTCCTGAGCGGGACAGGGCCGAGAGGGCGCTCCCGAGCTTGGGCCTGCTGGTGGGTGAGAC
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII

CCAGGAGAGAGGGAGCTAGAGGGGGGAGCTCTGAGGACTGATCTTGACTGTCTGCCCCCAGACCATCAGC
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIICCCCCCCCC

ATATCCGCTACAACCCGCTGCAGGATGAGTGGGTGCTGGTGTCAGCTCACCGCATGAAGCGGCCCTGGTA
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

GGGTCAAGTGGAGCCCCAGCTTCTGAAGACAGTGCCCCGCCATGACCCTCTCAACCCTCTGTGTCCTGGG
C000000000000000000000000000000000000000000000000000000000000000000000

GCCATCCGAGCCAACGGAGAGGTAAGCCTGTAGAGCCCTGCATCTGCAGGCTGGGCCACGG
0000000000000000000000000000000000000000000000000000000000000
```

Kuva 1: Annotointi kolmeen luokkaan. Ylärivillä ihmisen DNA-sekvenssiä. Alarivillä kunkin emäksen annotaatio: 'C' merkitsee koodaavaa jaksoa (eksonia), 'I' intronia ja '0' geenienvälistä jaksoa. Kuvan sekvenssi sisältää yhden geenin, joka koostuu kahdesta eksonista (harmaat alueet), joiden välissä on introni. Huomaa ensimmäisen eksonin aloituskodoni ATG ja viimeisen eksonin lopetuskodoni TAG. [Kro97]

Geenien etsintää tehdään monenlaisin laskennallisin menetelmin. Niitä on vertailtu GASP-kokeilussa, jossa sama banaanikärpäsen (*Drosophila melanogaster*) *Adh*-alue annotoitiin eri menetelmillä; löydöksiä verrattiin laboratoriotutkimuksiin ja asiantuntijoiden käsityöhön perustuvaan "oikeaan" annotaatioon. GASP-kokeilua selostetaan laajasti Genome Researchin erikoisnumerossa [Gen00]. Yleiskuvaus kokeilun asetelmasta ja tuloksista on artikkelissa [RHH$^+$00], katsaus geenien etsinnän perusmenetelmiin artikkelissa [Sto00] ja yksityiskohtaisemmat kuvaukset kokeiluun osallistuneista menetelmistä omina artikkeleinaan.

Tämän raportin rakenne on seuraava. Luvussa 2 esitellään muutamia informaation lähteitä, joihin geenien etsintä voi perustua. Luvussa 3 esitellään *piilo-Markov-mallit* ja luvussa 4 kerrotaan, miten niitä käytetään DNA-sekvenssien annotointiongelmissa. Tässä tarvittava *Viterbin algoritmi* esitellään luvussa 5. Luvussa 6 mainitaan lyhyesti muutamia uudempia piilo-Markov-malleihin perustuvia menetelmiä ja luvussa 7 on yhteenveto.

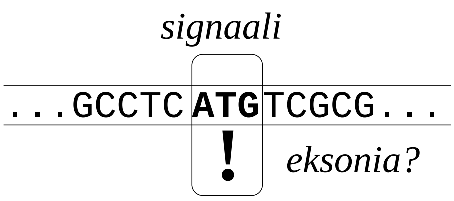
2 Informaation lähteet

Seuraavaksi tarkastelemme muutamia perusperiaatteita, joihin erilaiset geenien etsintämenetelmät perustuvat. Jaottelu on Stormon [Sto00] esittämä.

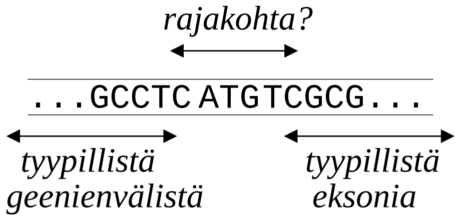
2.1 Signaalit sekvenssissä

Idealtaan hyvin yksinkertainen geenien etsintätapa on etsiä sekvenssistä erityisiä *signaaleja*, jotka sijaitsevat jaksojen rajakohdissa. Signaaleja ovat esimerkiksi geenien alussa olevat aloituskodonit (ATG tai GTG) ja lopussa olevat lopetuskodonit (TAG, TAA tai TGA). Geenin sisällä voidaan puolestaan etsiä ns. luovuttaja- ja vastaanottajakohtia (donor, acceptor sites), joissa eksoni vaihtuu introniksi tai päinvastoin. Muita signaaleja ovat esim. geeniä toisinaan edeltävät promoottorit (kuten TATATA...).

Valitettavasti signaalien löytäminen ja tulkinta ei ole mitenkään yksiselitteistä. Signaalit ovat epävarmoja: esimerkiksi kaikki DNA:sta löytyvät ATG-jonot eivät suinkaan ole geenin aloitussignaaleja. Signaalin tunnistamiseen käytetään kohtalaisen mutkikkaitakin mal-



(a) Signaali (aloituskodoni).



(b) Tilastollinen informaatio.

Kuva 2: Annotointia eri informaationlähteiden perusteella.

leja (mm. hermoverkkoja), ja silti saadaan vain *todennäköisyys* sille, onko tietyssä paikassa signaali vai ei. Emme siis voi olla varmoja, että paikasta todella alkaa vaikkapa eksoni. Voimme kuitenkin päätellä, että *jos* paikan lähettyviltä pitäisi alkaa eksoni, *niin* aloituskohta on todennäköisesti juuri tässä eikä muutamaa merkkiä aiemmin tai myöhemmin (kuva 2(a)).

2.2 Sekvenssin tilastollinen sisältö

Toinen geenien löytämistä auttava informaationlähde on se, että erityyppisten jaksojen sisältämät sekvenssit ovat tilastollisilta ominaisuuksiltaan erilaisia.

Yksinkertaisin tilastollinen ominaisuus on yksittäisten emästen jakauma. Ajatellaanpa esimerkiksi, että tiedämme ihmisen eksoneissa esiintyvän emäksiä osuuksin (C 34%, A 24%, G 32%, T 10%), mutta eksonien ulkopuolella tasaosuuksin[2]. Mielivaltainen DNA-sekvenssin kohta voidaan nyt luokitella katsomalla sen sopivankokoisessa ympäristössä emästen jakaumaa: jos havaittu jakauma muistuttaa tunnettua eksonien emäsjakaumaa, arvaamme kyseisen kohdan kuuluvan eksoniin.

Pitkän jakson olemassaolon voi näinkin yksinkertaisella mallilla löytää melko hyvin, mutta sen rajakohdat jäävät epämääräisiksi, sillä frekvenssitilastoa ei voi mielekkäästi kerätä mielivaltaisen lyhyeltä alueelta. Rajakohta ilmenee vain summittaisesti siitä, että sen toisella puolella on paremmin yhdenlaiseen jaksoon sopivaa merkkijakaumaa, ja toisella puolella toisenlaiseen jaksoon sopivaa (kuva 2(b)). Lyhyitä jaksoja ei ehkä löydetä lainkaan.

Kehittyneempiä tilastollisia ominaisuuksia ovat mm. emäskolmikoiden (kodonien) jakaumat. Esimerkiksi kodonit GCU, GCC, GCA ja GCG ovat *synonyymejä* eli koodaavat samaa proteiinia, alaniinia. Synonyymejä ei yleensä esiinny eksoneissa tasaosuuksin, vaan jotkin niistä ovat muita yleisempiä. Tällaista (eliölajista toiseen vaihtelevaa) kodonien valintapreferenssiä voidaan käyttää eksonien tunnistamisessa: jos sekvenssissä on kodoneja kyseiselle eliölle tyypillisissä suhteissa, sekvenssi on todennäköisesti eksonia.

2.3 Sekvenssien samankaltaisuus

Kolmas mahdollinen informaationlähde ovat ennestään tunnetut sekvenssit. Jos nyt tutkittavasta sekvenssistä löytyy jakso, joka merkkijonona suunnilleen täsmää jonkin ennestään tunnetun sekvenssin kanssa, voidaan arvata, että jaksojen tehtävätkin ovat samankaltaiset. Näin voidaan arvata tutkittavan sekvenssin olevan eksoni, jos se muistuttaa tunnettua eksonia; introni, jos se muistuttaa tunnettua intronia jne.

Tunnetut sekvenssit ja niiden tunnetut tehtävät voivat muodostaa hyvinkin laajan tietokannan, jolloin algoritmisena haasteena on, miten etsiä tietokannasta ne sekvenssit, jotka

[2]Esimerkin jakaumat on valittu mielivaltaisesti eivätkä ne välttämättä ole realistisia.

parhaiten täsmäävät tutkittavana olevan sekvenssin kanssa.

3 Piilo-Markov-mallit

Edellä nähtiin, että DNA-sekvenssin annotointiin on tarjolla monenkinlaista epävarmaa, osittaista informaatiota. Olennaiseksi ongelmaksi muodostuukin, miten rakentaa yhtenäinen malli, joka kykenee hyödyntämään näitä erityyppisiä informaationjyväsiä.

Esimerkiksi tilastollisen informaation ja signaalien voi toivoa tukevan toisiaan seuraavasti. Tilastollinen informaatio tarjoaa luotettavan summittaisen tiedon jakson luokituksesta, vaikkei tarkkoja rajakohtia; signaalit auttavat määrittämään jaksolle tarkat rajakohdat. Toisaalta jos löytyy signaali, jonka mukaan jostain paikasta tulisi alkaa esimerkiksi eksoni, mutta tilastollinen informaatio ei tue tätä, signaali voidaan hylätä virheellisenä.

3.1 Samankaltaisia ongelmia muualla

Algoritmisesti samankaltaisia merkkijono-ongelmia esiintyy monilla muillakin tieteenaloilla, mm. koodausteoriassa (virheenkorjaavat koodit, error-correcting codes) ja puheentunnistuksessa. Yhteistä näissä ongelmissa on, että havaittu merkkijono on jonkin mutkikkaan, osin satunnaisen prosessin tulos. Itse prosessin kulku on osittain tuntematon ja tavoitteena onkin päätellä havaitusta merkkijonosta "takaperin", millaisen prosessin tuloksena se on todennäköisesti voinut syntyä.

Virheenkorjaavassa koodissa on alkuperäiseen merkkijonoon lisätty (deterministisesti) redundanssia. Näin saatuun koodattuun merkkijonoon on sitten tullut satunnaisia virheitä jonkin todennäköisyysjakauman mukaisesti. Vastaanottaja havaitsee koodatun, virheitä sisältävän merkkijonon; tehtävä on päätellä, mikä oli alkuperäinen merkkijono ennen koodausta.

Puheentunnistuksessa tyypillinen ongelmanasettelu on, että foneemijono realisoituu monimutkaisen ääntämisprosessin tuloksena äänisignaaliksi, johon edelleen sekoittuu taustahälyä ja mittausvirheitä jonkin satunnaisprosessin mukaisesti. Tehtävä on havaitusta äänisignaalista päätellä alkuperäinen foneemijono.

Luontevan ratkaisun tällaisiin merkkijono-ongelmiin tarjoavat *piilo-Markov-mallit* (HMM, hidden Markov model).

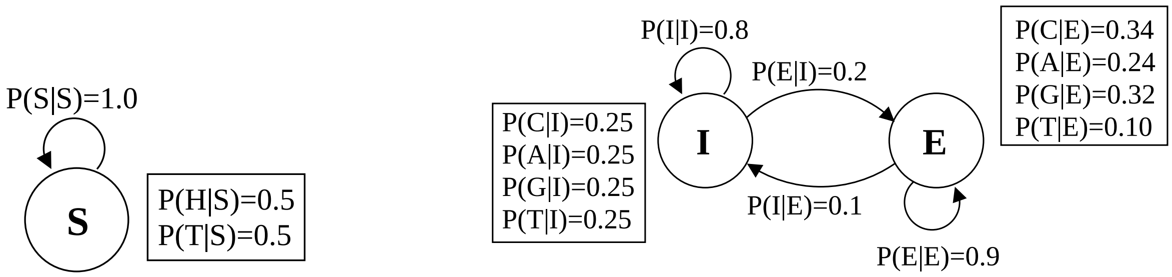
3.2 Lyhyt johdatus Markov-malleihin

Piilo-Markov-mallia voi ajatella probabilistisena tilakoneena, joka mallintaa jotakin askelittain etenevää satunnaisprosessia. Prosessilla on pieni joukko mahdollisia *tiloja*. Kullakin askelella prosessi satunnaisesti joko siirtyy tilasta toiseen (*siirtymä*, transition) tai pysyy samassa tilassa (*itsesiirtymä*, self-transition). Kullakin askelella se myös *tuottaa* satunnaisesti jonkin aakkossymbolin jakaumasta, joka määräytyy senhetkisen tilan mukaan[3]. Näin syntyy kaksi satunnaismuuttujajonoa: tilojen jono X ja tuotettujen merkkien jono Y. Tavallisesti ajatellaan, että voimme suoraan havaita vain tuotetun merkkijonon, mutta prosessin sisällä tapahtuneet tilasiirtymät ovat tuntemattomia ja niitä voidaan vain estimoida; siksi prosessin tiloja kutsutaan myös *piilotiloiksi*.

Kuvassa 3(a) on yksinkertainen, harhatonta kolikonheittoa kuvaava Markov-malli, jossa on vain yksi tila ja siten vain yksi (itse)siirtymä. Prosessin tilat muodostavat deterministisen jonon $X = SSS\ldots$, mutta tuotettu merkkijono on satunnainen, esim. $Y = HHTTHTT\ldots$

[3]Tuotetun symbolin jakauma voisi määräytyä myös tapahtuneen *tilasiirtymän* mukaan kuten lähteessä [HSF97b].

Kuvassa 3(b) on mielenkiintoisempi Markov-malli. Ajatellaan, että meillä on kaksi nelisivuista, painotettua noppaa E ja I. Kummankin nopan sivuilla on kirjaimet C, A, G ja T, mutta painotukset ovat erilaiset (kuvan laatikot). Kullakin ajanhetkellä heitämme yhtä noppaa ja kirjoitamme tuloksen paperille; sitten satunnaisesti joko vaihdamme noppaa (siirtymä) tai pidämme käytössä saman nopan (itsesiirtymä). Prosessin tilat — so. kumpaa noppaa kulloinkin heitimme — muodostavat satunnaisen tilajonon, esim. $X = EEEEEIIIIEEE\ldots$, ja tuotetut kirjaimet satunnaisen merkkijonon, esim. $Y = CGCACATTGAGCG\ldots$



(a) Harhaton kolikko.

(b) Kaksi nelisivuista noppaa.

Kuva 3: Yksinkertaisia piilo-Markov-malleja.

Siirtymien ja merkkien todennäköisyyksille on tässä käytetty lyhennysmerkintöjä. Esimerkiksi siirtymätodennäköisyys I:stä E:hen merkitään $P(X_{i+1} = E \mid X_i = I)$, tai lyhyemmin $P(E|I)$. Vastaavasti merkin C tuottamisen todennäköisyys tilassa E merkitään $P(Y_i{=}C \mid X_i{=}E)$, tai lyhyemmin $P(C|E)$.

Piilo-Markov-malli kuvaa tapahtumat ajassa etenevänä satunnaisprosessina. Koska prosessin tapahtumiin on liitetty ehdolliset todennäköisyydet ja prosessin eteneminen riippuu vain kulloisestakin tilasta eikä lainkaan aiemmasta historiasta, voidaan helposti asettaa ja ratkaista seuraavanlaisia kysymyksiä.

Tilajonon todennäköisyys. Jos lähtötila on tasatodennäköisyyksin I tai E, miten todennäköisesti toteutuu tilajono $X = IIEEE$? Ratkaisu on lähtötilan I todennäköisyyden ja tarvittavien siirtymien todennäköisyyksien tulo,

$$\begin{aligned} P(X{=}IIEEE) &= P(X_1{=}I) \cdot P(I|I) \cdot P(E|I) \cdot P(E|E) \cdot P(E|E) \\ &= 0.5 \cdot 0.8 \cdot 0.2 \cdot 0.9 \cdot 0.9. \end{aligned}$$

Merkkijonon ehdollinen todennäköisyys. Jos prosessissa toteutuu tilajono $X = IIEEE$, miten todennäköisesti se tuottaa merkkijonon $Y = TTCGA$? Ratkaisu on Y:n merkkien ehdollisten todennäköisyyksien tulo,

$$\begin{aligned} P(Y{=}TTCGA \mid X{=}IIEEE) &= P(T|I) \cdot P(T|I) \cdot P(C|E) \cdot P(G|E) \cdot P(A|E) \\ &= 0.25 \cdot 0.25 \cdot 0.34 \cdot 0.32 \cdot 0.24. \end{aligned}$$

Piilo-Markov-mallien varsinainen vahvuus ilmenee siinä, että päättelyn suunta voidaan kääntää:

Tilajonon ehdollinen todennäköisyys, kun merkkijono tunnetaan. Jos prosessi tuottaa merkkijonon $Y = TTCGA$, miten todennäköistä on, että tilajono oli $X = IIEEE$? Ratkaisu perustuu ehdollisen todennäköisyyden kääntämiseen Bayesin kaavalla:

$$P(X|Y) = \frac{P(Y|X) \cdot P(X)}{P(Y)},$$

joten

$$P(X{=}IIEEE \mid Y{=}TTCGA) = \frac{P(Y{=}TTCGA \mid X{=}IIEEE) \cdot P(X{=}IIEEE)}{P(Y{=}TTCGA)}.$$

Edelleen voidaan kysyä: jos merkkijono oli $Y = TTCGA$, *mikä* kaikista mahdollisista tilajonoista oli todennäköisin? Tähän sopiva algoritmi esitellään luvussa 5.

4 Annotointiongelmiin sopivia piilo-Markov-malleja

4.1 Tilastolliseen informaatioon perustuva malli

Edellä esitetty kahden nelisivuisen nopan malli (kuva 3(b)) sopii itse asiassa kuvaamaan tilastolliseen informaatioon perustuvaa genomin karkeata annotointia kahdenlaisiin jaksoihin: eksoneihin (E) ja muuhun (I). Mallin tuottamat tila- ja merkkijonot ovat todennäköisimmin sellaisia, että

- tilajono pysyy pitkään samassa tilassa ja vain silloin tällöin siirtyy tilasta toiseen, ja
- I-tilassa tuotettujen merkkien frekvenssi on tasan jakautunut, mutta E-tilassa tuotettujen frekvenssi luvussa 2.2 esimerkkinä annetun eksonien emäsjakauman mukainen.

Kääntäen: kun merkkijonolle (DNA-sekvenssille) etsitään mallin mukainen todennäköisin tilajono, löytynyt tilajono on sellainen, että se mahdollisimman hyvin vastaa oletettuja eksonien ja niiden ulkopuolisten alueiden emäsjakaumia. E-tilassa tuotetut merkit voidaan nyt annotoida eksoneihin kuuluviksi ja I-tilassa tuotetut eksonien ulkopuolisiksi.

Kun eksonien ulkopuolinen alue halutaan edelleen jakaa introneihin ja geenienväliseen alueeseen, erotetaan tilasta I vielä kolmas tila 0. Tilojen E, I ja 0 väliset siirtymätodennäköisyydet asetetaan sellaisiksi, että syntyvien jaksojen pituudet osapuilleen vastaavat tunnettuja eksonien, intronien ja geenienvälisten jaksojen tyypillisiä pituuksia. Kirjainten C, A, G ja T todennäköisyydet kussakin tilassa asetetaan vastaavien jaksojen tunnettuja emäsfrekvenssejä vastaaviksi.

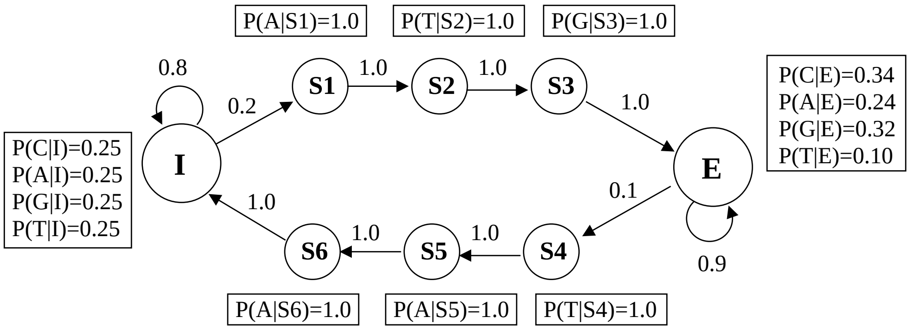
Tässä mallissa huomioitiin vasta yksittäisten emästen jakauma eksonin sisällä. Mallia voidaan laajentaa niin, että yhtä annotointiluokkaa (kuten eksonia) ei vastaa vain yksi *tila*, vaan kokonainen useasta tilasta ja niiden välisistä siirtymistä koostuva *osamalli*, joka esimerkiksi kuvaa erilaisten kodonien tai kodonijonojen todennäköisyyksiä eksonin sisällä.

4.2 Signaalien huomiointi Markov-mallissa

Mallia on nyt helppo laajentaa huomioimaan myös jaksosta toiseen siirtymistä osoittavat signaalit. Jaksoja kuvaavien osamallien väliin lisätään erityiset *siirtymämallit*. Siirtymämalli on suunniteltu siten, että se tuottaa sellaisia lyhyitä merkkijonoja, jotka ovat tällaisissa siirtymissä tyypillisiä.

Kuvassa 4 on esimerkki hyvin yksinkertaisesta signaaleja huomioivasta mallista. Se on saatu kuvan 3(b) mallista lisäämällä siirtymämallit yhdelle aloituskodonille ATG (siirtymä I:stä E:hen) ja yhdelle lopetuskodonille TAA (siirtymä E:stä I:hin).

Siirtymämalli sopii dataan hyvin nimenomaan silloin, kun datassa on havaittavissa siirtymälle tyypillinen signaali. Siirtymiä sisältävä piilo-Markov-malli siis suosii sellaisia tilajonoja, joissa tila vaihtuu esimerkiksi E:stä I:ksi täsmälleen siihen sopivan signaalin kohdalla *ja* siirtymän molemmin puolin emästen frekvenssit sopivat mallin mukaisiin jakaumiin. Näin malli hyödyntää sekä tilastollisen että signaaleista saadun informaation.



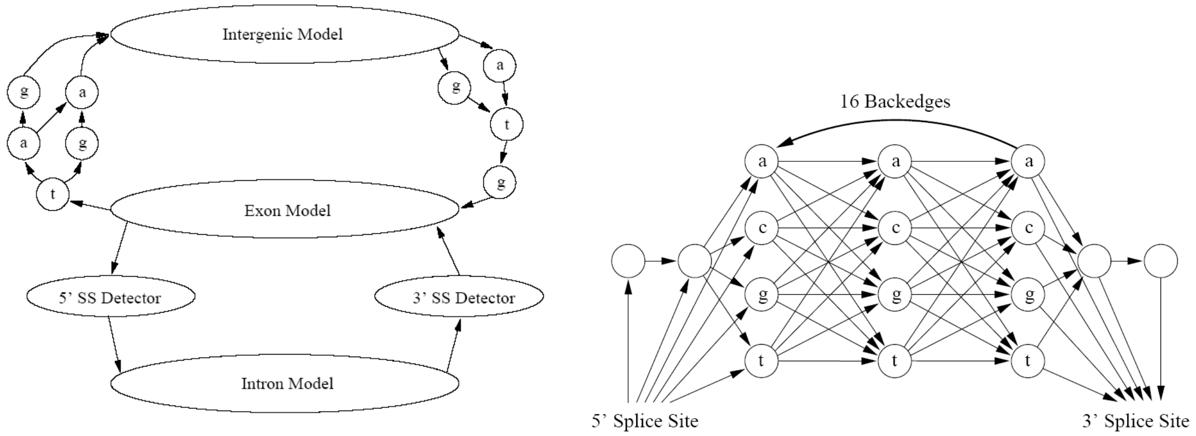
Kuva 4: Yksinkertainen siirtymäsignaaleja huomioiva malli.

On huomattava, että siirtymämalli *ei pakota* mallia siirtymään esimerkiksi eksonitilaan jokaisen merkkijonossa esiintyvän ATG-sekvenssin kohdalla. Kun malli on tilassa I, se voi seuraavaksi tuottaa jonon ATG kahdella tavalla: siirtymällä tilojen $S1, S2, S3$ kautta tilaan E *tai* pysymällä seuraavat kolme askelta edelleen tilassa I. Näiden tapahtumien todennäköisyydet vain ovat erilaiset ja siksi malli *pyrkii* noudattamaan siirtymäsignaaleja.

4.3 Realistinen kokonaismalli

Geenien etsinnässä käytettävät piilo-Markov-mallit sisältävät lukuisia osamalleja ja tiloja. Mallin rakenteen suunnittelu tehdään yleensä käsityönä, tunnetun DNA-sekvenssien rakennetta koskevan tiedon pohjalta. Mallin parametrit (siirtymätodennäköisyydet ja merkkien todennäköisyydet) voidaan asettaa käsityönä tai *opettamalla* mallia tunnetulla, jo annotoidulla aineistolla. Näin saadaan esimerkiksi mallin käyttämät merkkien frekvenssitilastot vastaamaan tunnettuja emäsjakaumia kussakin tilassa. Opetuksessa voidaan käyttää ns. *EM-algoritmia* (expectation maximization), jota ei kuitenkaan tässä raportissa käsitellä.

Hyvä esimerkki tällaisesta kehittyneestä mallista on Hendersonin *et al.* kehittämä ihmisen genomin malli [HSF97b]. Mallin karkea rakenne on esitetty kuvassa 5. Huomaa siirtymämallit erilaisille signaaleille: aloituskodoneille ATG ja GTG, lopetuskodoneille TAG, TAA ja TGA, luovuttajakohdille (5' splice site) sekä vastaanottajakohdille (3' splice site).



Kuva 5: *Vasemmalla:* Ihmisen DNA:ta kuvaava piilo-Markov-malli. Ympyrät ovat yksittäisiä tiloja, mutta ellipsit kuvaavat osamalleja, jotka todellisuudessa koostuvat useasta tilasta ja niiden välisistä siirtymistä [HSF97b]. *Oikealla:* Tarkempi kuva intronimallista [HSF97a].

Tilojen määrä voi tällaisessa mallissa kasvaa melko suureksi. Samojen kirjoittajien aiemmassa kirjoituksessa [HSF97a] kuvataan mallia tarkemmin ja mainitaan siinä olleen peräti 241 tilaa ja 1003 niiden välistä siirtymää. Mallin rakenne ei ole kuitenkaan niin monimutkainen kuin tästä voisi päätellä, sillä osamallit ovat systemaattisesti rakennettuja, puumaisia ja hilamaisia verkkoja.

Mallia testattiin annotoimalla sata sekvenssiä ihmisen DNA:ta, yhteensä noin 656 000 emästä. Eksonijaksoja sekvensseissä oli 633 kpl, yhteensä noin 135 000 emästä. Malli löysi 778 eksonijaksoa, joista 426 oli oikeita. Näistä 292:ssa ainakin toisen päätepisteen ja 111:ssä molempien päätepisteiden tarkka paikka oli oikein. Yksittäisistä emäksistä 81 % annotoitiin oikein. [HSF97b]

4.4 Piilo-oletus jaksojen pituusjakaumista

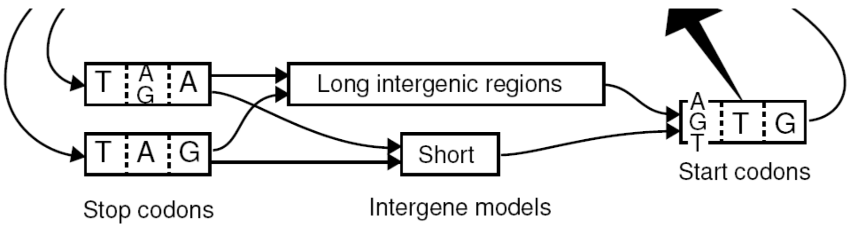
Tarkastellaan vielä kuvan 3(b) mallia. Mallin itsesiirtymien todennäköisyydet ovat silmiinpistävän korkeat, lähellä ykköstä. Tämä merkitsee, että jouduttuaan tiettyyn tilaan prosessilla on taipumus pysyä samassa tilassa melko pitkään. Tämä on hyvin tyypillistä DNA-sekvenssiä kuvaaville Markov-malleille. Taustalla on se empiirinen tieto, että intronit, eksonit ja geenienväliset jaksot todella ovat melko pitkiä, tyypillisesti kymmenien tai satojen merkkien pituisia.

Lähempi tarkastelu paljastaa, että asettamalla tilalle korkean itsesiirtymätodennäköisyys tulemme implisiittisesti olettaneeksi, että kyseistä tilaa vastaavien jaksojen pituudet noudattavat *geometrista jakaumaa*[4]. Olkoon nimittäin tilan S itsesiirtymän todennäköisyys $P(S|S) = \alpha$. Tämä merkitsee, että kullakin askeleella, aiemmista riippumatta, jakso päättyy todennäköisyydellä $1 - \alpha$. Siten jakson pituus L noudattaa jakaumaa

$$P(L{=}1) = 1 - \alpha, \quad P(L{=}2) = \alpha \cdot (1 - \alpha), \quad \ldots, \quad P(L{=}l) = \alpha^{l-1} \cdot (1 - \alpha), \ldots$$

Pituuden odotusarvo on $\mathbb{E}(L) = 1/(1 - \alpha)$. Jos esim. $\alpha = 0.99$, on $\mathbb{E}(L) = 1/0.01 = 100$.

Todellisuudessa DNA-sekvenssin jaksojen pituudet eivät kuitenkaan noudata tällaista geometrista jakaumaa. Niinpä esimerkiksi eräässä *E. coli* -bakteerin DNA:ta kuvaavassa mallissa [KMH94] geenienvälisten jaksojen pituusjakaumaa on paranneltu siten, että käytössä on kaksi vaihtoehtoista osamallia: "lyhyt" ja "pitkä" (kuva 6). Edellinen kuvaa 1–14 merkin pituisia jaksoja; jälkimmäinen kuvaa huomattavastikin pidempiä jaksoja ja on rakenteeltaan melko mutkikas. Periaatteessa mitä tahansa pituusjakaumaa voidaan approksimoida tällä tavoin osamalleja yhdistelemällä.



Kuva 6: Ei-geometrisen pituusjakauman mallintaminen kahdella osamallilla. [KMH94]

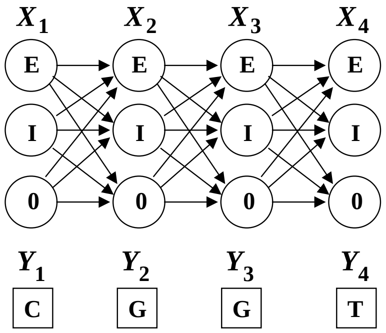
Suoremman ratkaisun tarjoavat *yleistetyt* piilo-Markov-mallit (GHMM, generalized hidden Markov model) [PAC01, Rab89]. Niissä kuhunkin tilaan on erikseen koodattu tilan *keston* jakauma. Tällaisessa mallissa ei käytetä itsesiirtymiä, vaan kun malli joutuu tiettyyn tilaan, se tuottaa *useita* merkkejä — kyseisen tilan kestojakauman mukaisen satunnaisen määrän — ja siirtyy välittömästi sen jälkeen johonkin muuhun tilaan.

[4]Puhutaan usein myös eksponenttijakaumasta. Tarkkaan ottaen näillä on se ero, että geometrinen jakauma on diskreetti, mutta eksponenttijakauma on jatkuva jakauma.

5 Piilotilojen jonon estimointi Viterbin algoritmilla

Kun piilo-Markov-mallin rakenne ja parametrit on kiinnitetty, voidaan annettu DNA-sekvenssi Y annotoida etsimällä se piilotilojen jono X, joka parhaiten sopii selittämään (tuottamaan) annetun sekvenssin, eli jolla $P(X|Y)$ on suurin. Kunkin merkin Y_i annotaatio on nyt yksinkertaisesti sen tilan X_i tai useasta tilasta koostuvan osamallin nimi, jossa prosessi oli merkin tuottaessaan.

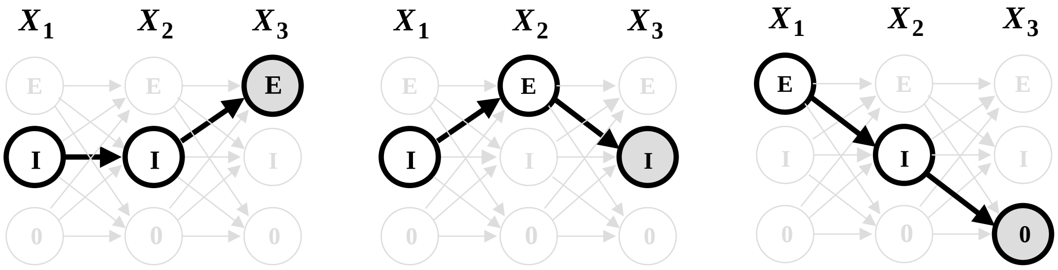
Ongelma onkin, miten löytää tämä paras tilajono. Ilmeinen "raa'an voiman" menetelmä olisi kokeilla läpi kaikki mahdolliset tilasiirtymien jonot (kuva 7). Näistä sitten valittaisiin se tilajono, jolle $P(X|Y)$ on suurin. Tämä menetelmä ei ole käytännössä käyttökelpoinen, sillä kokeiltavien tilajonojen määrä olisi eksponentiaalinen DNA-sekvenssin pituuteen nähden.



Kuva 7: Mahdollisia tilajonoja kuvaava ns. trellis-kaavio, kun mallissa on $k = 3$ tilaa $\{E, I, 0\}$ ja merkkijonon pituus on $n = 4$. Erilaisia tilajonoja on k^n kpl.

Todennäköisimmän tilajonon löytämiseen on onneksi olemassa tehokas, polynomiaikainen *Viterbin algoritmi*, joka perustuu dynaamiseen ohjelmointiin [Boy]. Menetelmässä on kaksi vaihetta. Olkoot mallin tilat $S_1, S_2, \ldots, S_k$.

Ensimmäisessä vaiheessa edetään havaittua merkkijonoa alusta loppuun. Kun ollaan i:nnen merkin kohdalla, selvitetään *kullekin* mahdolliselle tilalle S_j $(j = 1, \ldots, k)$, mikä olisi todennäköisin *kyseiseen* tilaan päätyvä tilajono $X_1,\ X_2,\ \ldots,\ X_{i-1},\ X_i = S_j$ (kuva 8) ja mikä on sen todennäköisyys $\delta_{ji} := P(X_1, \ldots, X_i|Y)$.



Kuva 8: Viterbin algoritmi vaiheessa $i = 3$. Todennäköisimmät polut tiloihin $X_3 = E$, $X_3 = I$ ja $X_3 = 0$ korostettu.

Tämän parhaan osajonon löytämiseen riittää, kun tiedetään vastaava tieto *edellisestä* paikasta $i - 1$. Tällöin voidaan nimittäin kokeilla läpi kaikki edellisen paikan mahdolliset tilat $X_{i-1} = S_{j'}(j' = 1, \ldots, k)$ ja tutkia, mikä tulisi uuden osajonon todennäköisyydeksi, kun perään liitetään haluttu tila S_j. Tämä todennäköisyys on helppo laskea: kerrotaan edellinen osatodennäköisyys tarvittavalla siirtymätodennäköisyydellä ja tarvittavan merkin Y_i tuottamisen todennäköisyydellä. Siten

$$\delta_{ji} = \max_{j'}\{\delta_{j',i-1} \cdot P(X_i{=}S_j \mid X_{i-1}{=}S_{j'}) \cdot P(Y_i \mid X_i{=}S_j)\}.$$

Käytännössä osajonoja ei tarvitse tallettaa kokonaan, vaan riittää, kun jokaisesta askeleesta i muistetaan kullekin tilalle j kaksi asiaa:

- todennäköisimmän tilaan j päätyvän i-pituisen osajonon todennäköisyys δ_{ji} ja
- indeksi, joka kertoo mistä tilasta kyseinen osajono edellisellä askeleella tuli.

Ensimmäisen vaiheen loputtua tunnetaan kullekin lopputilalle $X_n = S_j$ todennäköisimmän polun todennäköisyys δ_{jn}. Valitaan suurin ja seurataan vain indeksiketjua taaksepäin. Tämän toisen vaiheen päätyttyä on saatu selville todennäköisin tilajono X.

Viterbin algoritmissa täytyy pitää yllä kahta $k \times n$ -taulukkoa (δ-matriisi ja indeksitaulukko). Aikavaatimus on luokkaa $O(en)$, missä e on mallin siirtymien (kaarten) lukumäärä.

6 Muunnelmia teemasta

6.1 Parittaiset Markov-mallit

Geenien etsintää voidaan tehdä lajienvälisesti esimerkiksi niin, että tutkitaan yhtäaikaa hiiren ja ihmisen genomia. Jos ne pystytään *rinnastamaan* (align) eli löytämään, mitkä kohdat ihmisen genomissa vastaavat mitäkin kohtia hiiren genomissa, ja tunnetaan hiiren genomin eksonikohdat, voidaan luonnollisesti arvata, että vastaavat kohdat ihmisen genomissa ovat eksoneja.

Askel pidemmälle on suorittaa kahden sekvenssin rinnastus ja annotointi *yhtäaikaa* yhdellä ns. *parittaisella* piilo-Markov-mallilla [PAC01].

6.2 Toiseksi paras, kolmanneksi paras jne.

Viterbin algoritmi löytää sen yksittäisen tilajonon, joka on havaitulle merkkijonolle kaikkein todennäköisin "selitys".

On havaittu, että mikäli samalle datalle on olemassa useita edes suunnilleen yhtä uskottavia, toisistaan poikkeavia selitysmalleja, nämä muutkin selitykset voivat sisältää biologisesti relevanttia informaatiota. Olisi siis hyödyllistä löytää nämä ns. suboptimaalisetkin tilajonot. Viterbin algoritmista on olemassa muunnoksia, joilla tämäkin onnistuu deterministisesti: voidaan siis etsiä esimerkiksi k parasta (uskottavinta) tilajonoa.

7 Yhteenveto

Laskennallinen geenien etsintä on perusasetelmaltaan yksinkertainen hahmontunnistustehtävä: annetusta merkkijonosta on löydettävä halutunlaisia hahmoja (osamerkkijonoja). Tehtävää kuitenkin hankaloittaa se, että hahmoja määrittelevää informaatiota on monentyyppistä — esimerkiksi jaksojen *rajakohtia* koskevia signaaleja ja jaksojen *sisältöä* koskevaa tilastollista tietoa.

Piilo-Markov-mallit tarjoavat tehokkaan mahdollisuuden yhdistellä tällaisia erityyppisiä osittaisen informaation palasia. Käsityöksi jää vain mallin *rakenteen* suunnittelu, mutta mallin parametrien estimoinnissa sekä varsinaisessa geenien etsintätyössä päästään hyödyntämään hyvin tunnettuja, yleiskäyttöisiä EM- ja Viterbi-algoritmeja.

Viitteet

[Boy] Roger Boyle. Hidden Markov Models. Web tutorial at `http://www.comp.leeds.ac.uk/roger/HiddenMarkovModels/html_dev/main.html`.

[Gen00] *Genome Research*, 10(4), April 2000. Available at `http://www.genome.org/content/vol10/issue4/`.

[HSF97a] J. Henderson, Steven Salzberg, and Kenneth H. Fasman. Finding genes in DNA with a hidden Markov model, 1997. Draft of paper submitted to Journal of Computational Biology. Available at `http://www.cs.jhu.edu/~salzberg/veil.ps`.

[HSF97b] J. Henderson, Steven Salzberg, and Kenneth H. Fasman. Finding genes in human DNA with a hidden Markov model. *Journal of Computational Biology*, 4(2):127 – 141, 1997.

[KMH94] Anders Krogh, I. Saira Mian, and David Haussler. A hidden Markov model that finds genes in E. coli DNA. Technical Report UCSC-CRL-93-33, University of California Santa Cruz, 1994.

[Kro97] Anders Krogh. Two methods for improving performance of a HMM and their application for gene finding. In *Proceedings of the 5th International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pages 179 – 186, 1997.

[PAC01] Lior Pachter, Marina Alexandersson, and Simon Cawley. Applications of generalized pair hidden Markov models to alignment and gene finding problems. In *Proceedings of the fifth annual international conference on Computational biology*, pages 241–248. ACM Press, 2001.

[Rab89] Lawrence B. Rabiner. A tutorial on hidden Markov models. *Proceedings of the IEEE*, 77(2):257 – 286, February 1989.

[RHH+00] M. G. Reese, G. Hartzell, N. L. Harris, U. Ohler, J. F. Abril, and S. E. Lewis. Genome annotation assessment in Drosophila melanogaster. *Genome Research*, 10(4):483 – 501, April 2000.

[Son03] Mingzhou Song. Introduction to fundamentals of molecular biology. Lecture notes at `http://www.cs.gc.cuny.edu/~msong/87100/slides/topic1.pdf`, 2003.

[Sto00] Gary D. Stormo. Gene-finding approaches for eukaryotes. *Genome Research*, 10(4):394 – 397, April 2000.