hyväksymispäivä arvosana

arvostelija

# Markovin piilomallit geenien tunnistamisessa

Julius Laitala

Helsinki 7. toukokuuta 2018
HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

| Tiedekunta — Fakultet — Faculty | Laitos — Institution — Department |
| --- | --- |
| Matemaattis-luonnontieteellinen tiedekunta | Tietojenkäsittelytieteen laitos |

Tekijä — Författare — Author
Julius Laitala

Työn nimi — Arbetets titel — Title

Markovin piilomallit geenien tunnistamisessa

Oppiaine — Läroämne — Subject
Tietojenkäsittelytiede

| Työn laji — Arbetets art — Level | Aika — Datum — Month and year | Sivumäärä — Sidoantal — Number of pages |
| --- | --- | --- |
| Kandidaatintutkielma | 7. toukokuuta 2018 | 23 sivua |

Tiivistelmä — Referat — Abstract

Tämä tutkielma tarkastelee Markovin piilomallien käytön geenien tunnistamiseen nykytilaa. Tutkielmassa keskitytään geenien ennustamiseen, eli niiden tunnistamiseen nojaamatta puhtaaseen samanlaisuuteen.

Tutkielmassa esitellään aluksi lyhyesti genetiikan perustermit sekä geenien tunnistamisen ongelman taustat. Tämän jälkeen tutustutaan Markovin piilomallien peruskäsitteisiin ja tarkastellaan Markovin piilomallien perusongelmia ja algoritmeja.

Kun genetiikan ja Markovin piilomallien taustat ovat selvillä, tutkielma syventyy geenien tunnistukseen Markovin piilomalleilla. Aluksi tarkastellaan geenien tunnistamiseen käytettyjen Markovin piilomallien rakennetta sekä muutamia geenien tunnistamisessa hyväksi todettuja piilomallivariantteja. Tämän jälkeen tutustutaan erilaisiin datatyyppeihin, joita käytetään apuna geenien tunnistamisessa.

Lopuksi tarkastellaan geenientunnistusohjelmistoja, jotka käyttävät hyödykseen Markovin piilomalleja, sekä sitä, kuinka hyviä tuloksia voidaan geenien tunnistamisessa saavuttaa kyseisellä työkalulla. Huomataan, että tehokkaimpia geenien tunnistuksessa näyttäisivät olevan yleistetyt Markovin piilomallit sekä mallit, jotka osaavat hyödyntää mahdollisimman paljon ”lisädataa”.

ACM Computing Classification System (CCS):
Applied computing → Life and medical sciences → Bioinformatics

Avainsanat — Nyckelord — Keywords
Markovin piilomalli, HMM, Geenien tunnistaminen

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — övriga uppgifter — Additional information

# Sisältö

# 1 Johdanto

Sekvensointimetodien kehittyessä saatavilla oleva DNA-raakadatan määrä kasvaa koko ajan kiihtyvällä tahdilla. Tätä kirjoitettaessa 7. toukokuuta 2018 Yhdysvaltain Kansallisen Bioteknologiatietokeskuksen tietokantaan oli talletettu 36616 eliön, soluelimen tai viruksen perimä [1]. 12. maaliskuuta 2018 vastaava luku oli 34896 – vajaassa kuukaudessa oli kantaan lisätty 1720 tietuetta.

Jotta tästä datasta saataisiin kaikki hyöty irti tulee selvittää, miten mikäkin perimän osa vaikuttaa eliön toimintaan. Tämä tapahtuu etsimällä sekä tunnistamalla siitä geenit, perimän toiminnalliset yksiköt. Koko tämän datamäärän analysointi ilman automaattista tietojenkäsittelyä on käytännössä mahdotonta [29]. Geenien löytäminen ei ole kuitenkaan triviaalia, eivätkä koneelliset keinot geenien löytämiseen ole pysyneet kasvavan datamäärän tahdissa [19]. Perimään, geeneihin ja geenien tunnistamisen ongelmaan tutustutaan tarkemmin kappaleessa 2 – samalla rajoitetaan tarkempi tarkastelu geenien tunnistamiseen ennustamalla.

Markovin piilomallit ovat osoittautuneet työkaluksi, jolla geenien löytämisen haasteeseen voidaan vastata. Näiden kaksinkertaisesti satunnaisten tilastollisten mallien yleiseen rakenteeseen perehdytään kappaleessa 3. Markovin piilomallien laskennalliset ongelmat ja niihin tehokkaan ratkaisun takaavat algoritmit esitellään kappaleessa 4. Kappaleessa 5 puolestaan tarkastellaan, minkälaisia ovat rakenteeltaan geenien tunnistuksessa käytetyt Markovin piilomallit. Esimerkkinä käytetään suosittua ja pitkälle kehitettyä AUGUSTUS-ohjelmistoa.

Pelkkä perimän emäsdata riittää geenien tunnistamiseen. Vihjeiden saaminen siihen liittyvästä datasta, kuten RNA:sta tai proteiineista, kasvattaa kuitenkin tunnistustarkkuutta. Erilaisia datatyyppejä, joita käytetään geenien tunnistuksessa apuna, esitellään kappaleessa 6. Kappaleessa 7 tutustutaan suosituimpiin geenientunnistusohjelmistoihin sekä siihen, kuinka hyvin ne ennustavat geenejä.

Kun yleiskuva geenien tunnistuksesta, Markovin piilomalleista sekä näitä yhdistävistä ohjelmistoista on luotu, tiivistetään kappaleessa 8, minkälainen on nykytiedon valossa toimiva geenejä tunnistava Markovin piilomalli – miten se on rakennettu, mitä variantteja se käyttää ja mitä dataa sille syötetään.

# 2 Geenien tunnistaminen

## 2.1 DNA ja geeni

Jokaisen eliön perimä sijaitsee *DNA*:na tunnetussa rakenteessa. DNA on *nukleotideista* eli nukleiinihapoista koostuva kaksoisketju, joka sisältää ohjeet selviytymiselle välttämättömien molekyylien muodostamiseen sekä suuren osan tiedosta, jonka avulla näiden molekyylien muodostumista säädellään. Eliöt voidaan jakaa *aitotumaisiin* ja *esitumaisiin* sen perusteella, missä eliön (tai sen solujen) DNA sijaitsee; aitotumaisilla solun tumassa, esitumaisilla yhdessä rengasmaisessa kromosomissa

sekä plasmideissa.

Jokainen nukleotidi koostuu *emäksestä, sokeriosasta ja fosfaatista*. DNA:n emäsosia ovat guaniini `G`, adeniini `A`, sytosiini `C` sekä tymiini `T`. Kaksoisketju muodostuu siten, että kohdassa, jossa toisessa ketjussa sijaitsee adeniini `A`, sijaitsee toisessa ketjussa tymiini `T`. Kohdassa jossa toisessa ketjussa on puolestaan guaniini `G` löytyy toisesta ketjusta sytosiini `C`. Molemmilla ketjuilla on suunta, joka määräytyy sokeriosan mukaan – *5'-3'-suunta*, missä järjestyksessä DNA-ketju yleensä lähes aina kirjoitetaan, tai *3'-5'-suunta*. Kaksoisketjussa vastakkaiset ketjut ovat aina erisuuntaisia.

*Geeni* on DNA-jakso, jonka emästen järjestys sisältää tiedon tietyn molekyylin rakenteesta. Geenin voidaan määritellä koostuvan *RNA*:ksi kääntyvästä eli *transkriptoituvasta* alueesta sekä sitä edeltävistä *promoottoreista*, jotka edesauttavat *transkription* alkamista [19]. RNA on myös nukleotidiketju – se eroaa DNA:sta sokeriosansa osalta sekä siten, että siinä tymiinin tilalla on urasiili, `U`.

Transkriptiossa muodostetaan geeniä vastaava RNA-molekyyli. Tämä RNA-molekyyli voi itsessään olla toiminnallinen (*ei-koodaava RNA* eli *ncRNA*) [20], tai edelleen toimia ohjeena proteiinin muodostukseen *translaationa* tunnetussa prosessissa; tällöin puhutaan *lähetti-RNA*:sta eli *mRNA*:sta. Transkriptio tapahtuu aina 5'-3'-suuntaan.

Proteiineja koodaavien eli lähetti-RNA:ksi kääntyvien geenien voidaan ajatella koostuvan *kodoneista*, kolmen emäsparin jonoista, joista jokainen merkitsee translaatiossa jotakin tiettyä aminohappoa. Esimerkiksi emäsjono `ACC` (adeniini, sytosiini, sytosiini) tulkitaan ihmisen translaatiossa treoniini-aminohapoksi. Erityisiä kodoneita ovat *aloituskodonit*, esimerkiksi ihmisen metioniinia koodaava `AUG`, jotka kertovat mistä translaatio alkaa, sekä *lopetuskodonit*, jotka merkitsevät translaation loppukohtaa. Aloituskodonista alkavaa ja lopetuskodoniin loppuvaa, lopetuskodonitonta DNA-ketjun pätkää kutsutaan *avoimeksi lukukehykseksi*.

Aitotumaisilla eliöillä lähetti-RNA:n transkriptoituva alue sisältää kuitenkin muutakin tietoa, kuin itse proteiinin rakennusohjeen: ennen translaatiota lähetti-RNA:sta poistetaan *intronit* prosessissa, jota kutsutaan *silmukoinniksi*. Intronien poiston jälkeen jäljelle jääviä, varsinaisesti tietoa proteiinista sisältäviä osia sanotaan *eksoneiksi*.

## 2.2 Geenien tunnistaminen koneellisesti

Geenien koneellinen tunnistaminen ei ole triviaali ongelma [19]. Esitumaisilla yksinkertainen avoimen lukukehyksen haku, eli aloituskodonin etsiminen ja seuraavaan lopetuskodoniin pysähtyminen, toimii jotenkuten, mutta aitotumaisilla tämä lähestymistapa on suurilta osin poissa laskuista. Avoimen lukukehyksen tunnistamisen tapaisilla yksinkertaisilla menetelmillä erittäin lyhyet geenit hukkuvat taustakohinaan, ncRNA:ta koodaavia geenejä ei kodonirakenteen puutteen takia huomata, ja koodaavan emäsjonon sisällä olevat intronit aiheuttavat ongelmia [8].

Yleisesti geenien tunnistamisen ongelmaa voi lähestyä kahdella tavalla. Ensinnä-

kin voidaan tukeutua *samanlaisuuteen*, eli pyrkiä löytämään analysoitavasta DNA-jonosta rinnastuksia jo tunnettuihin geeneihin. Sellaisten lajien, joiden läheisiltä lajeilta tunnetaan jo geenejä, geeneistä noin puolet voidaan löytää tällä menetelmällä [19]. Samanlaisuuteen nojautuminen ei kuitenkaan toimi hyvin esimerkiksi epätavallisten geenien tapauksessa tai kun geneettinen etäisyys, eli ajallinen etäisyys yhteisestä esi-isästä, on suuri [19].

Toisaalta voidaan pyrkiä tunnistamaan geenejä yleisten geenin ominaisuuksien perusteella, eli *ennustamaan* geenejä. Esimerkiksi proteiineja koodaavien eksonien kodoneista johtuvaa kolmijaksoisuutta sekä sitä, että intronit sisältävät pääasiassa enemmän adeniinia ja tymiiniä kuin eksonit, voidaan käyttää apuna sen ennustamiseen, onko tietyssä kohtaa DNA-ketjua geeni [19]. Nämä menetelmät ovat kuitenkin samanlaisuuteen perustuvia menetelmiä herkempiä virheellisten tunnistusten synnyttämiseen. Myöskään ennustavat menetelmät eivät välttämättä löydä kaikkein eksoottisimpia geenejä, sillä nekin on luotu olemassa olevan datan oletusten pohjalta [19].

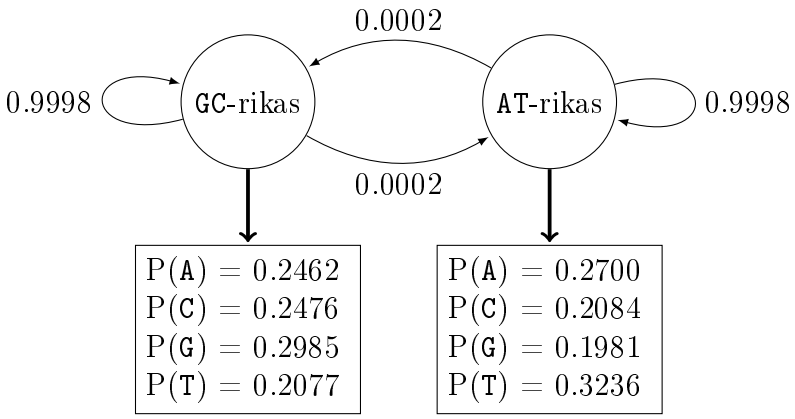
Markovin piilomalleja on sovellettu sekä samanlaisuuteen että ennustamiseen perustuvaan geenien tunnistukseen [19]. Tässä tutkielmassa keskitytään nimenomaan geenejä ennustaviin Markovin piilomalleihin.

# 3 Markovin piilomallit

Markovin piilomallit (hidden Markov model, HMM) ovat ”kaksinkertaisesti satunnaisia” tilastollisia malleja. Ne koostuvat yhdestä piilotetusta *Markov-ketjusta*, jonka tilojen muutoksia määrää *siirtymätodennäköisyys* (ensimmäinen satunnaisuuden taso), sekä kuhunkin tilaan liittyvästä *emissiotodennäköisyydestä*, joka määrää millä todennäköisyydellä tila tuottaa mitäkin merkkiä mallin tuottamaan merkkijonoon (toinen satunnaisuuden taso). Kuvaan 1 on luonnosteltu yksinkertainen, DNA:ta `AT`- ja `GC`-rikkaisiin alueisiin jaotteleva Markovin piilomalli.

Markovin piilomallit ovat omiaan mallintamaan tilannetta, jossa ei-havaittu prosessi (Markov-ketju) tuottaa mitattavissa olevia havaintoja (tuotettu merkkijono). Ne ovat tunnettuja lähekkäisten symbolien, alueiden tai tapahtumien korrelaatioiden mallintajina, ja niitä on käytetty runsaasti esimerkiksi puheentunnistuksessa [29].

Piilomallit toimivat hyvin myös monissa bioinformatiikan tehtävissä, ja niitä on käytetty geenien tunnistamisen lisäksi muun muassa geenien ja proteiinien ositteluun kemiallisesti erilaisiin alueisiin ja DNA:n toiminnallisuuden ennustamiseen [8], sekä emästen tunnistamiseen kromatogrammipiikeistä, sekvensoinnin virheiden mallinnukseen, proteiinin toissijaisen rakenteen tunnistamiseen, koodaamattoman RNA:n tunnistamiseen ja RNA:n rakenteelliseen rinnastukseen [29].




Kuva 1: Yksinkertainen DNA:ta ositteleva Markovin piilomalli. Todennäköisyydet on laskettu lambda-bakteriofagin perimästä EM-algoritmilla [8].

## 3.1 Rakenne

Markovin piilomalli voidaan määritellä matemaattisesti seuraavasti: Olkoon piilotettu tilajono *Markovin ketju*, eli prosessi, jossa jokaisella ajanhetkellä tehdään tilasiirtymä seuraavaan tilaan siten, että seuraavan tilan todennäköisyys riippuu vain yhdestä tai useammasta edellisestä tilasta [9]. Sitä, kuinka monesta tilasta todennäköisyys riippuu, kutsutaan Markovin ketjun *asteeksi*. Oletetaan tarkastelun helpottamiseksi, että piilotetun Markovin ketjun aste on 1, eli että sen seuraava tila riippuu vain ja ainoastaan nykyisestä tilasta.

Olkoon $H = \{H_1, H_2, \ldots, H_N\}$ Markovin ketjun tilojen joukko, ja $O = \{O_1, O_2, \ldots, O_M\}$ havaittavien symbolien joukko. Kuvassa 1 havainnollistetussa mallissa piilotilojen joukko $H$ on $\{\texttt{GC}\text{-rikas}, \texttt{AT}\text{-rikas}\}$ ja havaittavien symbolien joukko $O$ on emästen joukko $\{\texttt{A}, \texttt{C}, \texttt{G}, \texttt{T}\}$.

Olkoon $h = h_1 h_2 \cdots h_L$ piilotilojen jono, ja $s = s_1 s_2 \cdots s_L$ havaittujen merkkien jono. Merkitään *tilasiirtotodennäköisyyttä* eli todennäköisyyttä, että Markov-ketjussa siirrytään tilasta $i$ tilaan $j$, seuraavasti:

$$t(i, j) = P(h_n = j | h_{n-1} = i)\,. \tag{1}$$

Kuvassa 1 tilasiirtotodennäköisyys on esitetty numeroina siirtymää kuvaavien nuolten vieressä; esimerkiksi $t(\texttt{GC}\text{-rikas}, \texttt{AT}\text{-rikas}) = 0.0002$.

Merkitään *alkutilatodennäköisyyttä*, eli todennäköisyyttä, että ensimmäinen Markovin ketjun tila on $i$, seuraavasti:

$$\pi(i) = P(h_1 = i)\,. \tag{2}$$

Kuvan 1 Markovin piilomalliin alkutilatodennäköisyyttä ei ole merkitty. Voimme

määrittää sen esimerkiksi siten, että molemmilla tiloilla on yhtä suuri todennäköisyys olla ketjun tilajonon ensimmäinen tila: $\pi(\texttt{GC}\text{-rikas}) = 0.5, \pi(\texttt{AT}\text{-rikas}) = 0.5$.

Merkitään *havaintotodennäköisyyttä*, eli todennäköisyyttä, että tilassa $i$ havaitaan symboli $x$, seuraavasti:

$$e(x|i) = P(s_n = x|h_n = i)\,. \tag{3}$$

Havaintotodennäköisyydet on merkitty kuvaan 1 tilojen alla oleviin laatikoihin. Esimerkiksi $e(\text{T}|\texttt{AT}\text{-rikas}) = 0.3236$.

Yhtälöt 1, 2 ja 3 kuvaavat täysin Markovin piilomallin; merkitään Markovin piilomallia $\Theta = \{t(i,j), \pi(i), e(x|i)\}$. Nyt voidaan helposti laskea todennäköisyys havaita Markovin piilomallissa $\Theta$ merkkijono $s$, kun tilajono on $h$:

$$\begin{aligned} &P(s,h|\Theta) = P(s|h,\Theta)P(h|\Theta)\,, \\ &\text{missä} \\ &P(s|h,\Theta) = e(s_1|h_1)e(s_2|h_2)\cdots e(s_L|h_L)\,, \\ &P(h|\Theta) = \pi(h_1)t(h_1,h_2)t(h_2,h_3)\cdots t(h_{L-1},h_L)\,. \end{aligned} \tag{4}$$

Markovin piilomalliesimerkistämme (kuva 1) voimme näin laskea, että emäsjono `ATTGC` on huomattavasti todennäköisempi tilajonolla `AT`-rikas, `AT`-rikas, `AT`-rikas, `GC`-rikas, `GC`-rikas (todennäköisyys noin $2.088 \times 10^{-7}$) kuin tilajonolla `GC`-rikas, `GC`-rikas, `GC`-rikas, `AT`-rikas, `AT`-rikas (todennäköisyys noin $4.382 \times 10^{-8}$).

# 4 Perusongelmat ja -algoritmit

Markovin piilomalleihin liittyvät perusongelmat ovat *pisteytysongelma, optimaalisen rinnastuksen ongelma* sekä *koulutusongelma*. Tehokkaat ratkaisut Markovin piilomallien perusongelmiin saadaan *Viterbi-algoritmilla, Eteenpäin-algoritmilla* ja *Baum-Welch-algoritmilla* [29].

Myös tässä kappaleessa oletetaan tarkastelun helpottamiseksi Markovin piilomallin asteen olevan 1. Esitellyistä algoritmeista on kuitenkin olemassa myös muunnelmia sekä korkeamman asteen piilomalleille että kappaleessa 5.2 käsiteltäville piilomallivarianteille.

## 4.1 Pisteytysongelma ja Eteenpäin-algoritmi

Pisteytysongelmassa annettuna on merkkijono $s$ sekä Markovin piilomalli $\Theta$, ja tavoitteena on laskea merkkijonon havaintotodennäköisyys $P(s|\Theta)$. Naiivisti tämän voi tehdä summaamalla kaikkien mahdollisten tilajonojen todennäköisyydet tuottaa merkkijono $s$:

$$P(s|\Theta) = \sum_{h \in H^L} P(s, h|\Theta)\,. \tag{5}$$

Tämä on kuitenkin erittäin raskasta; mahdollisia tilajonoja $L$-pituiselle merkkijonolle on $|H|^L$, minkä seurauksena naiivin ratkaisun aikavaativuus on eksponentiaalinen merkkijonon pituuteen nähden.

Eteenpäin-algoritmi (*Forward-algorithm*) on tehokas ratkaisu pisteytysongelmaan [29]. Kaikkien mahdollisten tilajonojen läpikäynnin sijaan se käyttää dynaamista ohjelmointia. Algoritmi määrittelee Eteenpäin-muuttujan (*Forward-variable*)

$$\alpha(n, i) = P(s_1 \cdots s_n, h_n = i|\Theta)\,, \tag{6}$$

joka kertoo todennäköisyyden että tilajonon $n$ tila $h_n$ on $i$, kun merkkijonon merkit $1, \ldots, n$ ovat $s_1, \ldots, s_n$. Dynaamisen ohjelmoinnin periaatteiden mukaisesti voi tämän muuttujan laskea rekursiivisesti:

$$\begin{aligned} &\alpha(n, i) = \sum_{k \in H} [\alpha(n-1, k) t(k, i) e(s_n|i)]\,, \\ &\text{jossa} \\ &\alpha(1, i) = \pi(i) e(s_1|i)\,. \end{aligned} \tag{7}$$

Merkkijonon havaintotodennäköisyyden voi laskea tämän avulla seuraavasti:

$$P(s|\Theta) = \sum_{k \in H} \alpha(L, k)\,. \tag{8}$$

Eteenpäin-algoritmin avulla todennäköisyys havaita merkkijono $s$ Markovin piilomallissa $\Theta$ ratkeaa ajassa $\mathcal{O}(LN^2)$, missä $L$ on merkkijonon pituus ja $N$ piilotilojen määrä.

## 4.2 Optimaalisen rinnastuksen ongelma ja Viterbi-algoritmi

Optimaalisen rinnastuksen ongelmassa annetaan merkkijono $s$ sekä Markovin piilomalli $\Theta$. Tavoitteena on löytää parhaiten merkkijonon selittävä tilajono

$$h^* = \arg\max_h P(h|s, \Theta)\,. \tag{9}$$

Tässäkin ongelmassa jokaisen tilajonon naiivin vertailun aikavaativuus on merkkijonon pituuden suhteen eksponentiaalinen, ja tämäkin ongelma ratkeaa dynaamisella ohjelmoinnilla, *Viterbi-algoritmilla.*

Viterbi-algoritmissa määritellään muuttuja

$$\gamma(n,i) = \max_{h_1,\ldots,h_{n-1}} P(s_1 \cdots s_n, h_1 \cdots h_{n-1}, h_n = i|\Theta)\,, \tag{10}$$

eli suurin todennäköisyys sille, että $h_n = i$ jollain jonolla $h_1 \cdots h_{n-1}$ kun on havaittu merkit $s_1 \cdots s_n$. Muuttujan voi laskea tehokkaasti rekursiivisesti:

$$\begin{aligned} &\gamma(n,i) = \max_{k\in H}[\gamma(n-1,k)t(k,i)e(s_n|i)]\,, \\ &\text{jossa} \\ &\gamma(1,i) = \pi(i)e(s_1|i)\,. \end{aligned} \tag{11}$$

Lopuksi saadaan suurin havaintotodennäköisyys $P^*$ siten, että

$$P^* = \max_{k\in H} \gamma(L,k)\,. \tag{12}$$

Kun suurin havaintotodennäköisyys on laskettu, on itse todennäköisyyden aiheuttanut tilajono helppo selvittää seuraamalla laskentaa taaksepäin: Tila $i$, joka tuotti suurimman havaintotodennäköisyyden, tuotti todennäköisesti merkkijonon viimeisen merkin. Tila $j$, jolla saavutettiin viimeistä merkkiä vastaavaa tilaa laskiessa suurin $\gamma(L-1,j)t(j,i)e(s_L|i)$, vastaa toisiksi viimeistä merkkiä. Seuraaminen jatkuu tähän tapaan kunnes saadaan ensimmäisen merkin todennäköisesti tuottanut tila $a$, joka toisen merkin tuottanutta tilaa $b$ laskiessa maksimoi arvon

$$\begin{aligned} &\gamma(2-1,a)t(a,b)e(s_2|b) \\ =&\pi(a)e(s_1|a)t(a,b)e(s_2|b)\,. \end{aligned} \tag{13}$$

Viterbi-algoritmin avulla paras mahdollinen tilajono $h^*$ syötteelle $s$ löytyy ajassa $\mathcal{O}(LN^2)$ [29].

## 4.3 Koulutusongelma ja Baum-Welch-algoritmi

Koulutusongelmassa tavoitteena on laskea optimaaliset Markovin piilomallin $\Theta^* = \{t^*(i,j), \pi^*(i), e^*(x|i)\}$ parametrit.

Yksinkertaisimmillaan Markovin piilomallin voi kouluttaa *ohjatusti*. Tällöin on annettuna joukko toisiinsa liittyviä havaittuja merkkijonoja sekä joukko piilotilajonoja, joiden tiedetään synnyttäneen annetut merkkijonot. Näistä voidaan triviaalisti laskea Markovin piilomallin parametrit:

$$
\begin{aligned}
t^*(i,j) &= \frac{\text{Siirtymien määrä tilasta } i \text{ tilaan } j \text{ aineistossa}}{\text{Siirtymien määrä tilasta } i \text{ mihin tahansa tilaan aineistossa}}\,, \\
\pi^*(i) &= \frac{\text{Tilalla } i \text{ alkavien piilotilajonojen määrä aineistossa}}{\text{Piilotilajonojen kokonaismäärä aineistossa}}\,, \\
e^*(x|i) &= \frac{\text{Kuinka usein tila } i \text{ tuotti merkin } x \text{ aineistossa}}{\text{Tilan } i \text{ tuottamien merkkien kokonaismäärä aineistossa}}\,.
\end{aligned}
\tag{14}
$$

Usein ohjatussa koulutuksessa joitakin siirtymiä tai emissioita ei löydy koulutusaineistosta. Näille määritellään todennäköisyydeksi yleensä pieni *pseudotodennäköisyys*, jotta ne eivät olisi estimoidussa mallissa täysin mahdottomia.

Ohjattua kouluttamista mielenkiintoisempi tapaus on *ohjaamaton* koulutus. Siinä on annettu vain joukko havaittuja merkkijonoja $S$. Tavoitteena on laskea pelkistä merkkijonoista optimaaliset Markovin piilomallin $\Theta^*$ parametrit – piilotiloja ei tunneta:

$$
\Theta^* = \arg\max_{\Theta} P(S|\Theta)\,. \tag{15}
$$

Baum-Welch-algoritmi on tehokas heuristinen menetelmä saada riittävän hyvä ratkaisu ohjaamattomaan koulutusongelmaan [29]. Se on odotusarvon maksimointialgoritmi (*EM-algorithm*), joka käyttää hyväkseen *Eteenpäin-taaksepäin-algoritmia* (*Forward-backward-algorithm*).

Eteenpäin-taaksepäin-algoritmissa on yhdistetty *Taaksepäin-algoritmi* (*Backward-algorithm*), jolla voidaan laskea todennäköisyys tuottaa merkkijono $s_{n+1} \cdots s_L$ kun $h_n$ tiedetään, sekä Eteenpäin-algoritmi. Sitä käyttämällä saavutetaan pelkkää Eteenpäin-algoritmia tarkemmat tilakohtaiset todennäköisyydet, vaikka koko havainnolle saatava todennäköisyys ei välttämättä ole paras tai välttämättä edes mahdollinen [29].

Baum-Welch-algoritmi toimii karkealla tasolla seuraavasti:

1. Aloitetaan joistakin ”arvaamalla” valituista parametreista,
2. Lasketaan...
   (a) ...Eteenpäin-taaksepäin-algoritmin avulla todennäköisyydet olla tietyssä tilassa tiettyyn aikaan
   (b) ...Kohdan 2a tulosten ja Taaksepäin-algoritmin avulla todennäköisyydet olla tietyssä tilassa aikaan $t$ ja tietyssä tilassa aikaan $t+1$
3. Arvioidaan uudet parametrit kohdassa 2 laskettujen tulosten avulla,
4. Palataan kohtaan 2.

Tätä jatketaan, kunnes parametrit eivät enää muutu – tällöin on saavutettu *paikallinen maksimi*. Paikallinen maksimi on parametrien ”huippukohta”, jota lähellä olevat parametriyhdistelmät ovat löydettyjä parametreja huonompia. Kyseessä eivät välttämättä ole annettua merkkijonoa parhaiten selittävät parametrit, mutta usein saavutetaan riittävän hyvä Markovin piilomalli [2].

Baum-Welch on huomattavasti Eteenpäin- ja Viterbi-algoritmeja monimutkaisempi, ja täten sen tarkempi tarkastelu on tämän tekstin laajuuden ulkopuolella. Algoritmi on johdettu erittäin ymmärrettävästi Jeff Bilmesin artikkelissa ”A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models” [2].

Markovin piilomallin voi kouluttaa myös *puoliohjatusti*. Tällä tarkoitetaan koulutusta, jossa kouluttaminen aloitetaan ohjatusti, mutta se jatketaan loppuun ohjaamattomasti. Esimerkiksi geenejä tunnistavan Markovin piilomallin opettamisen voi aloittaa sukulaislajin tunnetuilla geeneillä, ja näin alustetun piilomallin koulutusta voi sitten jatkaa varsinaisesti tunnistettavana olevan lajin genomilla ohjaamattomasti [16].

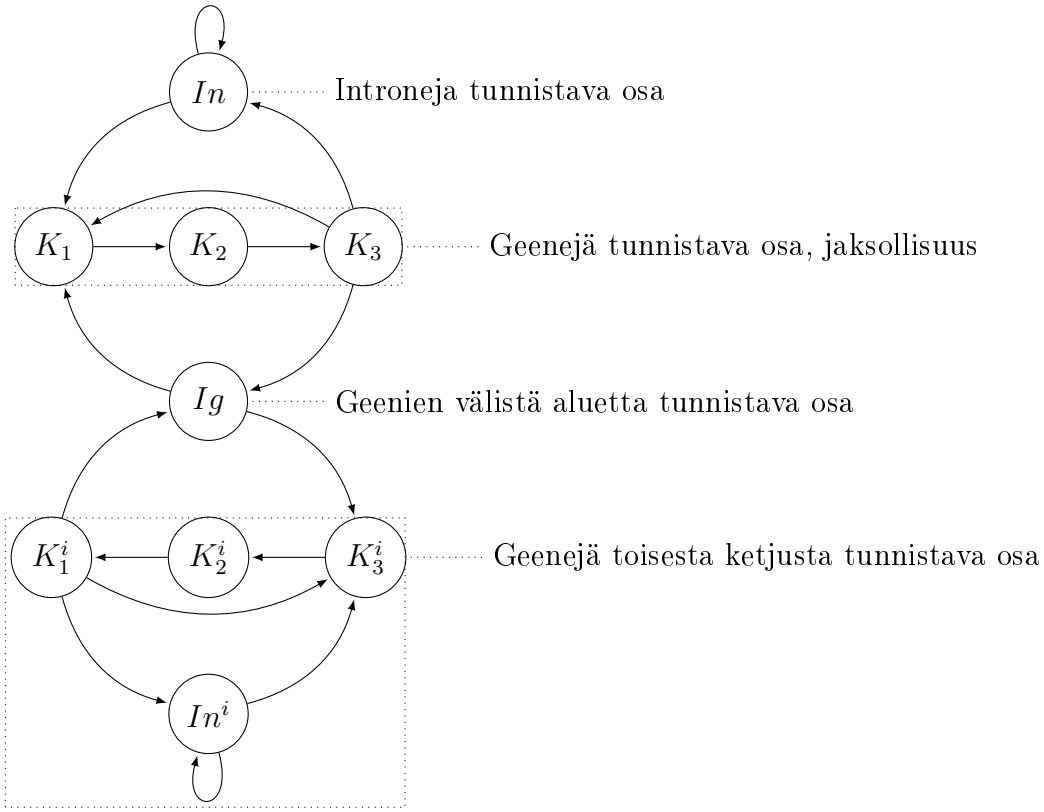
# 5 Geenejä tunnistava Markovin piilomalli

Markovin piilomallit ovat erinomainen työkalu geenien tunnistamiseen. Markov-ketjun voidaan tässä tapauksessa ajatella kuvaavan DNA:ssa piilossa olevaa toiminnallisuutta, ja sen tuottamien merkkien itse DNA:ta. Yhdistämällä tällä tavoin useita erilaisia todennäköisyyksiin nojaavia signaaleja Markov-ketjut suoriutuvat geenien tunnistamisesta huomattavasti paremmin kuin yksinkertaiset avoimen lukukehyksen haut. [8].

## 5.1 Yleisiä rakenteellisia ominaisuuksia

Geenejä tunnistavan Markovin piilomallin voi usein jakaa osiin sen perusteella, minkä tyyppistä rakennetta (ja sitä kautta toiminnallisuutta) mikäkin Markov-ketjun osa tunnistaa. Eksoneille, introneille ja geenien väliselle alueelle on usein kaikille oma joukkonsa tiloja [16, 18, 26].

Toinen yleisesti esiintyvä mahdollisuus jakaa malli osiin on DNA-ketjun suunnat; useissa geenejä tunnistavissa Markovin piilomallissa on yhdistetty kaksi peilikuvamaista Markovin piilomallia, joista toinen tunnistaa geenejä annetussa, 5’-3’-suuntaisessa ketjussa, ja toinen tunnistaa niitä takaperin annettua ketjua vastapäätä olevasta ketjusta [18, 26]. Vastakkaisesta ketjusta tunnistaminen onnistuisi myös ajamalla data kahteen kertaan Markovin piilomallin läpi – ensimmäisellä kerralla niin kuin se on annettu ja toisella kerralla takaperin siten, että jokainen emäs on muutettu vastakkaiseksi. Tämä aiheuttaa kuitenkin paljon vääriä geenitunnistuksia kohdissa, jossa vastakkaisessa ketjussa on geeni [3]. Toisaalta peilikuvatiloja käyttämällä jäävät jotkin päällekkäiset geenit, esimerkiksi ihmisen tyypin 1 neuro-




Kuva 2: Runsaasti pelkistetty esimerkki geenejä tunnistavasta Markovin piilomallista, joka sisältää kappaleessa 5.1 kuvatut ominaisuudet. Tilat $K_1 \dots K_3$ tunnistavat kodoneita lukusuuntaan annetusta DNA-ketjusta, kun taas tilat $K_1^i \dots K_3^i$ tunnistavat niitä takaperin vastakkaisesta ketjusta. Tilat $In$ ja $In^i$ tunnistavat introneita, ja tila $Ig$ tunnistaa geenien välistä aluetta. Emissio- ja siirtymätodennäköisyydet on jätetty selkeyden vuoksi kirjoittamatta.

fibromatoosigeeni kromosomissa 17, tunnistamatta – tämän takia joissain geeniejä tunnistavissa Markovin piilomalliohjelmistoissa voi peilikuvatilat ottaa halutessaan pois käytöstä [26].

Geenejä tunnistavat Markovin piilomallit sisältävät yleensä myös *jaksollisuutta*, eli tilaketjuja, joissa siirrytään aina tiettyyn seuraavaan tilaan. Yleinen jakson pituus on kolme – tämän taustalla on kodonien pituus [18, 19, 26].

Kuvassa 2 on esitelty pelkistetysti näitä ominaisuuksia.

## 5.2 Geenien tunnistuksessa käytettyjä Markovin piilomallin variantteja

Geenejä tunnistava Markovin piilomalli on harvoin niin yksinkertainen, kuin kappaleissa 3.1 ja 5.1 hahmoteltiin. Usein mallin aste on suurempi kuin 1 [24,26], piilotettu ketju ei ole puhdas Markovin ketju [16,18,24,26], sekä jotkin tilat tuottavat merkkejä emissiotodennäköisyyttä monimutkaisemmilla malleilla [26].

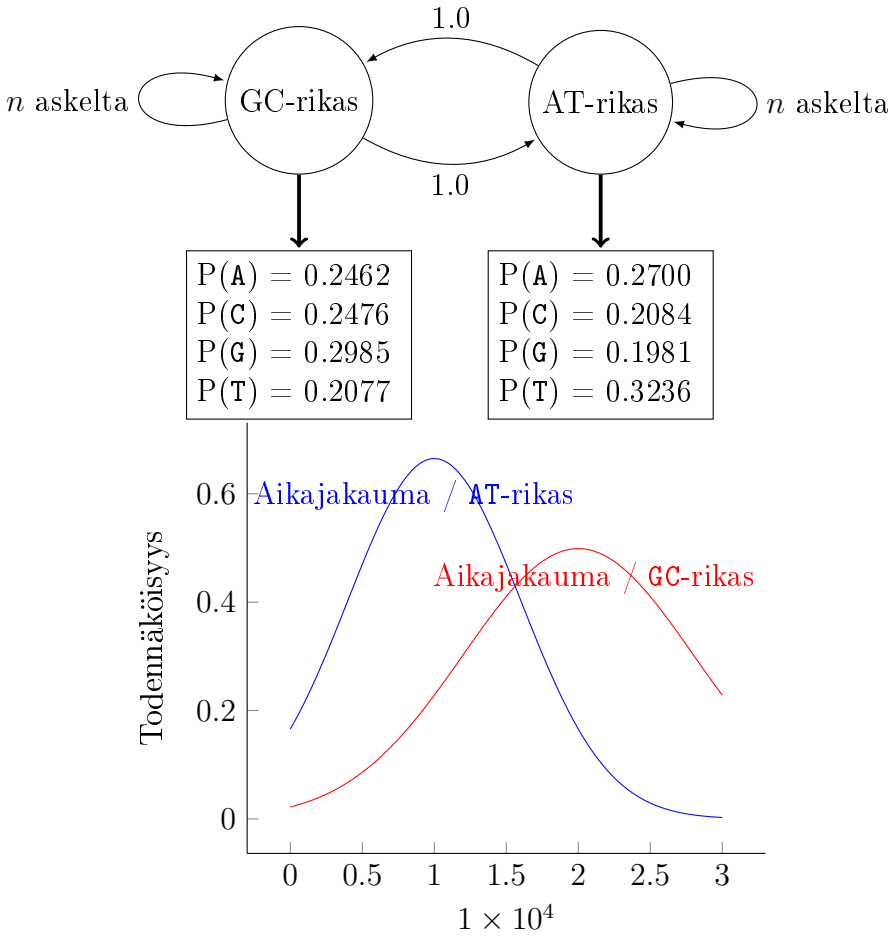
Näistä muokatuista Markovin piilomalleista on geenien löytämisessä ylivoimaisesti suosituin *yleistetty Markovin piilomalli* [16,18,26]. Monista malleista löytyy kuitenkin elementtejä myös esimerkiksi *pari-Markovin piilomalleista* [25, 29] ja *interpoloidusta Markovin piilomallista* [24, 26]. Yleisiä ovat myös useampien mallien yhdistelmät [22, 25].

Tarkastelemme seuraavaksi Markovin piilomallin variantteja. Käytämme tarkastelussa esimerkkinä suosittua AUGUSTUS-geenientunnistusohjelmistoa, sillä sen eri versioiden pohjina toimivat Markovin piilomallit sisältävät elementtejä sekä yleistetystä Markovin piilomallista, interpoloidusta Markovin piilomallista että pari-Markovin piilomallista [25, 26].

### 5.2.1 Yleistetty Markovin piilomalli

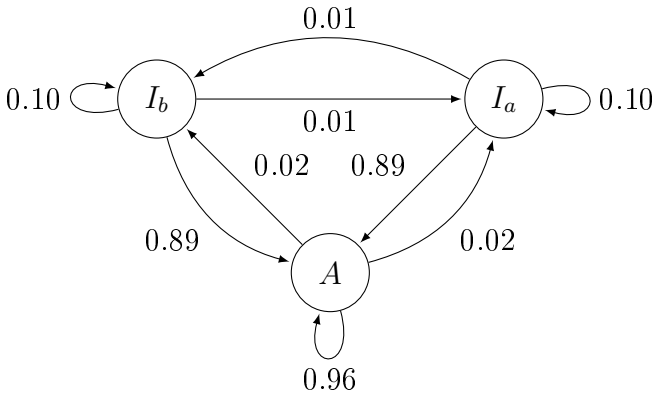
Geenejä tunnistavissa Markovin piilomalleissa on tavallista, että eräs mahdollinen ja usein todennäköisin siirtymä on tilasta takaisin itseensä [22]. Tämän seurauksena Markovin ketju pysyy usein samassa tilassa useamman kuin yhden askeleen ajan ennen seuraavaan tilaan siirtymistä; esimerkiksi kuvan 1 Markovin piilomallissa on tilan vaihtumisen todennäköisyys vain 0.0002. Ajan, jonka ketju pysyy samassa tilassa, huomataan olevan todennäköisyysjakaumaltaan geometrisesti vähenevä – tätä aikaa kutsutaan *kestoajaksi* [22].

Yleistetyssä Markovin piilomallissa (*generalized HMM* tai *hidden semi-Markov model*) jokaisella mallin tilalla on *aikajakauma*. Tilasiirtymän tapahtuessa uusi tila laskee aikajakaumasta satunnaisen kestoajan ja pysyy tilassa tämän ajan verran askeleita ennen siirtymistä seuraavaan, toiseen tilaan. Tämän avulla voidaan tarkemmin kontrolloida kestoajan jakaumaa ja täten kasvattaa tehokkuutta tietyissä tilanteissa [22].




Kuva 3: Kuvan 1 Markov-malli muutettuna yleistetyksi Markov-malliksi. Kestoaikojen jakaumat ovat karkeita arvauksia, ja niiden yksikkönä on $1 \times 10^4$ askelta.

Havaintotodennäköisyydet:
P(A) = 0.25
P(C) = 0.25
P(G) = 0.25
P(T) = 0.25




Kuva 4: Yksinkertainen pari-Markovin piilomalli, joka luo kaksi pääosin rinnakkaista DNA-ketjua. Jokaisen tilan havaintotodennäköisyydet ovat poikkeuksellisesti samat. Tila $I_a$ lisää merkkejä merkkijonoon $a$, ja tila $I_b$ merkkijonoon $b$. Tila $A$ on rinnastettu lisäystila, joka lisää saman merkin sekä merkkijonoon $a$ että jonoon $b$.

AUGUSTUS-geenientunnistusohjelmiston pohjana toimiva Markovin piilomalli sisältää esimerkiksi introneita tunnistavan tilan, jossa pysyttävä aika koulutetaan tiettyyn pituuteen asti todellista intronien pituusjakaumaa läheisesti vastaavaksi. Tämän ”rajapituuden” jälkeen jakauma jatkuu geometrisesti vähenevänä; näin pidetään muistinkulutus järkevän pienenä ja mahdollistetaan ennaltanäkemättömän pitkien intronien havaitseminen [26].

Esimerkki yleistetystä Markovin piilomallista löytyy kuvasta 3 – tilassa GC-rikas saatettaisiin saada kestoajaksi $2.1 \times 10^4$, jolloin malli tulostaisi $2.1 \times 10^4$ merkkiä tilan GC-rikas todennäköisyysjakauman mukaisesti ennen siirtymistä tilaan AT-rikas.

### 5.2.2 Pari-Markovin piilomalli

Markovin piilomallien biologisissa sovelluksissa on usein tärkeää vertailla kahta merkkijonoa, joiden epäillään liittyvän toisiinsa [22, 29]. Pari-Markovin piilomalli on Markovin piilomallivariantti, joka tuottaa yhden havaitun merkkijonon sijaan kaksi toisiinsa liittyvää merkkijonoa.

Mallin määritelmän tasolla tämä tapahtuu lisäämällä uusia tilatyyppejä. Merkitään pari-Markovin piilomallin $\Theta_p$ tuottamia kahta merkkijonoa kirjaimilla $a$ ja $b$. Mallin $\Theta_p$ jokaisen tilan määritellään olevan joko *lisäystila* merkkijonoon $a$, lisäystila merkkijonoon $b$ tai *rinnastutettu lisäystila*, joka lisää saman merkin molempiin merkkijonoihin.

AUGUSTUS-ohjelmiston AUGUSTUS+-laajennuksessa tuotiin ohjelmiston pohjana olevaan Markov-malliin pari-Markovin piilomallimaisia elementtejä. Siinä mahdollistettiin itse emäsjonon tuottamisen lisäksi useiden tiettyyn emäskohtaan liittyvien *vihjeiden* tuottaminen. Näihin vihjeisiin kuuluvat muun muassa *start* eli translaation aloitus ja *exon* eli tieto eksonissa olemisesta. Jokaisen AUGUSTUS+-laajenuuksen vihjetyypin voi mallintaa omana merkkijononaan – jos tietyssä kohtaa emäsjonoa ei

ole kyseistä vihjettä, on vihjemerkkijonossa tässä kohtaa tyhjä vihje; jos vihje havaitaan, kirjoitetaan vihjemerkkijonoon vihjeen arvosana eli todennäköisyys että vihje pitää paikkansa [25].

Esimerkkinä pari-Markovin piilomallista on kuva 4; siinä esitetyssä mallissa tilajono $I_a AAI_b A$ voisi tuottaa esimerkiksi merkkijonot

$$\begin{aligned} a &= \texttt{ACC-G}\,, \\ b &= \texttt{-CCTG}\,. \end{aligned} \tag{16}$$

(Yhtälössä 16 tyhjän merkin kohdalle lisätty viiva selvyyden vuoksi)

### 5.2.3 Interpoloitu Markovin piilomalli

Geenejä tunnistavan asteen $k$ Markovin piilomallin tarvitsee oppia $4^{k+1}$ todennäköisyyttä [24]. Ongelma korkeamman asteen Markovin piilomallien käyttämisessä geenien tunnistamiseen on, että dataa ei aina ole edes koko genomissa riittävästi luotettavien todennäköisyyksien laskemiseksi kaikille mahdollisille yhdistelmille, vaan jotkut yhdistelmät hukkuvat yleisempien alle sekä koulutuksessa lisättävien pseudomäärien aiheuttamaan kohinaan. Tietyt jopa kahdeksan emäksen yhdistelmät ovat kuitenkin niin yleisiä, että niiden havaitsemisesta olisi hyötyä geenien tunnistamisessa [24].

Eräs ratkaisu tähän ongelmaan on interpoloitu Markovin piilomalli. Siinä lasketaan koulutusvaiheessa jokaiselle tilalle siirtymätodennäköisyydet asteesta 0 asteeseen $d$ – aste 0 tarkoittaa yksinkertaisia tilasta riippumattomia prioritodennäköisyyksiä tuottaa merkki, ja aste $d$ on geenejä tunnistavassa Markovin piilomallissa usein 8 [19]. Jokaiselle lasketulle siirtymätodennäköisyydelle $t(i, j_1 \dots j_k)$, jossa $j_n$ on tila jonka arvioitiin vastaavan $n$ merkkiä sitten havaittua merkkiä ja $k$ on tällä hetkellä tarkasteltava aste, arvioidaan myös painoarvo tarkastelemalla, havaittiinko koulutusdatassa tilajonoa $j_1 \dots j_k$ tilastollisesti merkittävä määrä. Todellinen siirtymätodennäköisyys saadaan painotetusti yhdistelemällä eri asteiden siirtymätodennäköisyydet. Näin saadaan käyttöön suuremman asteen Markovin piilomallin hyödyt kun dataa tietylle pidemmälle tilajonolle on runsaasti, mutta toisaalta voidaan toimia lyhyemmän, tarkemmat siirtymätodennäköisyydet omaavan tilajonon perusteella datan ollessa puutteellista [24].

AUGUSTUS-ohjelmistossa interpoloitua Markovin piilomallia käytetään esimerkiksi eksonisisältöä mallintavassa Markovin piilomallin osassa. Siinä interpoloidun mallin aste on neljä, ja se on koulutettu kaikilla koulutusjoukon eksoniosuuksilla [26].

# 6 Geenien tunnistuksessa käytetty data

Markovin piilomallin voi kouluttaa tunnistamaan geenejä ohjaamattomasti, pelkän DNA-datan avulla [18]. Tämä voi olla ainoa vaihtoehto jos lajilta, jonka geenejä

yritetään tunnistaa, ei ole tunnistettu vielä riittävästi geenejä ja jos sillä ei ole riittävän läheistä, paremmin tunnettua sukulaislajia. Geenintunnistuksen tulos kuitenkin useimmiten paranee, jos kouluttaminen tehdään ohjatusti, käyttämällä genomia, josta osa geeneistä on jo tunnistettu, tai läheisen sukulaislajin genomilla, joka on hyvin kartoitettu.

Molempiin koulutusmenetelmiin voidaan kuitenkin tuoda mukaan dataa, joka ei varsinaisesti sisällä valmiiksi tunnistettuja geenejä mutta joka auttaa silti geenien löytämisessä. Tällaista dataa ovat esimerkiksi muut, läheisten lajien genomit, *proteiinirinnastukset*, sekä viime vuosikymmenen aikana suuresti hyödynnetty RNA:sta muodostettu *kompelementaarinen DNA* eli *cDNA* ja siitä johdettu *RNA-sekvenssointidata* (*RNA-seq*).

## 6.1 Useiden genomien samanaikainen käyttö

Lähekkäisten lajien genomit ovat usein hyvinkin samankaltaisia. Esimerkiksi kädellisten välillä koodaavat alueet eroavat toisistaan vain muutaman prosentin verran [7]. Tämän seurauksena usean, läheisen lajin genomin käytöllä samanaikaisesti voidaan parantaa geenien löytämisen tunnistustarkkuutta.

Usean genomin käyttö toteutetaan muodostamalla genomeiden rinnastus eli selvittämällä, missä kohtaa genomit ovat samanlaisia ja missä kohdissa niissä on eroja. Tätä rinnastusta voidaan sitten käyttää apuna geenien tunnistamiseen joko yhdestä tai useammasta syötegenomista. Muunmuassa N-SCAN sekä AUGUSTUS$_{gcp}$ ovat useita genomeja samanaikaisesti hyödyntäviä ohjelmistoja; edellinen tunnistaa geenejä yhdestä genomista ja käyttää muita syötegenomeja vain tietolähteinä, kun taas jälkimmäinen tunnistaa geenejä kaikista syötegenomeista samanaikaisesti [10, 15].

## 6.2 Proteiinirinnastukset

Useimmiten geenejä tunnistaessa ollaan kiinnostuneita nimenomaan proteiineja koodavista geeneistä. Koska tietyllä eliöllä tiettyjä aminohappoja koodaavat aina tietyt kodonit, voidaan proteiinia, jonka aminohappokoostumus tunnetaan, käyttää hyödyksi näiden kodonien paikan löytämisessä DNA-ketjusta – proteiini voidaan ”muuttaa takaisin” sen muodostaneiksi kodoneiksi ja näin saada vihjeitä siitä, minkälainen geeni sen on tuottanut.

Tällainen ratkaisu on toteutettu esimerkiksi AUGUSTUS-ohjelmiston AUGUSTUS-PPX-laajennuksessa [13].

## 6.3 Komplementaarinen DNA ja RNA-sekvensointidata

Komplementaarinen DNA (cDNA)on lähetti-RNA:sta käänteisellä transkriptiolla muodostettu DNA-ketju. Se sisältää siis tietoa alkuperäisen eksonin emäksistä. RNA-sekvensointi viittaa viime vuosikymmenen lopulla kehiteltyihin menetelmiin, jonka

avulla saadaan cDNA:sta sekvensoitua kokonaisia lähetti-RNA-ketjuja ja siten kokonaisia geenejä [28].

Tällainen tieto on erittäin hyödyllistä geenien tunnistamisessa - RNA:sta kerätty tieto on käytännössä emäsjonoja, jotka varmasti ovat osa jotakin geeniä, tai kokonaisia geenejä. Jäljelle jää vain geenien paikan löytäminen geenienvälisten alueiden ja introneiden keskeltä, sekä mahdollisesti niiden hyödyntäminen vielä sekvensoimattomien geenien löytämiseksi. Mahdollisuus RNA-sekvensointidatan käyttöön onkin luotu moniin suosituimpiin Markovin piilomalleja hyödyntäviin geenientunnistusohjelmistoihin, ja teknologian pohjalta on luotu uusia ohjelmistoja [17, 21, 27].

# 7 Ohjelmistoja

Kuten jo johdannossa todettiin, ovat Markovin piilomallit sopiva työkalu geenien löytämisen ongelman ratkaisemiseen. Ei siten ole kovinkaan ihmeellistä, että monet geenientunnistusohjelmistot toimivat juurikin Markovin piilomallejen avulla.

”Perinteisiä” sekä nykyäänkin paljon käytettyjä ohjelmistoja, jotka pohjautuvat Markovin piilomalleihin, ovat jo mainitun AUGUSTUS-ohjelmiston lisäksi Glimmer, GeneMark sekä SNAP. Taulukkoon 1 on koottu perustietoja näistä ohjelmistoista.

Yksittäisten sovellusten lisäksi suosittuja ovat myös useista ohjelmistoista tietoja yhdistelevät *ohjelmistoputkistot*. Monen tietolähteen vertailun ja hyödyntämisen avulla saavutetaan usein hyviä tuloksia pienemmällä vaivalla kuin vain yhtä ohjelmistoa käyttämällä. Putkistoja, joissa hyödynnetään Markovin piilomalleja käyttäviä geenientunnistusohjelmistoja, ovat suositun MAKER2-ohjelmiston lisäksi uudempi BRAKER1 sekä erikoitusneemmat Seqping ja SnowyOwl. Näistä on koottu tietoja taulukkoon 2.

## 7.1 Vertailua

Eräs tapa vertailla geenientunnistusohjelmistoja on tarkastella niiden *herkkyyttä* $Sn$ (*sensitivity*) sekä *tarkkuutta* $Sp$ (*specificity*). Nämä kaksi mittaria määritellään vertailemalla emäksistä saatua ennustetta ”oikeaan” geenidataan, seuraavasti:

$$\begin{aligned} Sn &= \frac{\text{Oikein ennustettujen piirteiden määrä}}{\text{Piirteiden todellinen määrä}}, \\ Sp &= \frac{\text{Oikein ennustettujen piirteiden määrä}}{\text{Ennustettujen piirteiden kokonaismäärä}}. \end{aligned} \tag{17}$$

Yhtälössä 17 *piirteet* voivat tarkoittaa vaikka koodaavia emäksiä, eksoneita tai geenejä. Koodaava emäs on ennustettu oikein, jos sekä ennustuksessa että todellisuudessa se sijaitsee eksonissa, kun taas eksoni on ennustettu oikein jos sen alku- sekä loppukohta ovat ennusteessa ja todellisuudessa samat. Geeni puolestaan on oikein

| Ohjelmisto | Ensimmäinen julkaisuvuosi | Uusin julkaisu | Käytetty Markovin piilomalli | Erikoistuminen | Verkkosivu | Lähteet |
|---|---|---|---|---|---|---|
| AUGUSTUS | 2003 | 2016 | Yleistetty Markovin piilomalli (myös piirteitä muista varianteista) | Aitotumaiset | `http://bioinf.uni-greifswald.de/augustus/` | [15,26] |
| Glimmer | 1998 | 2012 | Interpoloitu Markovin piilomalli | Mikrobit | `https://ccb.jhu.edu/software/glimmer/` | [14,24] |
| GeneMark | 1993 | 2014[a] | Useita versioita; esim. GeneMark-ET 3.0:ssa yleistetty Markovin piilomalli | Omat versionsa mikrobeille, aitotumaisille sekä metagenomeille[b] | `http://opal.biology.gatech.edu/GeneMark/` | [3,17,18] |
| SNAP | 2004 | 2004[c] | Yleistetty Markovin piilomalli | Yleinen | `http://korflab.ucdavis.edu/software.html` | [16] |
| CodingQuarry | 2015 | 2015 | Yleistetty Markovin piilomalli | Sienet | `https://sourceforge.net/projects/codingquarry/` | [27] |

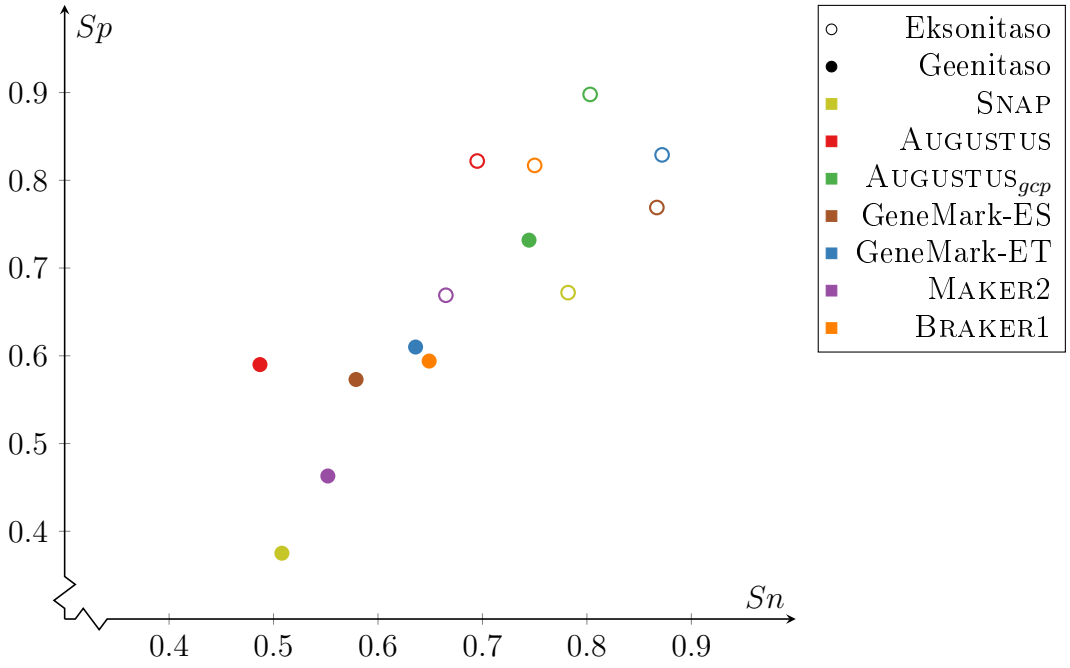
[a] Kirjoitushetkellä yksi artikkeli vertaisarviointiprosessissa
[b] *Metagenomilla* tarkoitetaan genomia, joka sisältää useiden lajien, tavallisesti mikrobien, perimää
[c] Ohjelman uusin versio kuitenkin vuodelta 2013

Taulukko 1: Tietoja kappaleessa 5.2 käsitellystä AUGUSTUS-ohelmistosta sekä kappaleessa 7 mainituista muista ”perinteisistä” ohjelmistoista on koottu tähän taulukkoon. Mukana on myös yksi lupaava uusi tulokas, CodingQuarry.

| Ohjelmisto-putkisto | Julkaisu-vuosi | Käytettyjä Markovin piilo-mallia hyödyntäviä geenientunnistus-ohjelmistoja | Erikoistuminen | Verkkosivu | Lähteet |
|---|---|---|---|---|---|
| MAKER2 | 2011[a] | AUGUSTUS, GeneMark-ES, SNAP | Toisen sukupolven genomiprojektit | http://www.yandell-lab.org/software/maker.html | [12] |
| SnowyOwl | 2014 | AUGUSTUS, GeneMark-ES | Sienet | https://sourceforge.net/projects/snowyowl/ | [23] |
| BRAKER1 | 2015 | AUGUSTUS, GeneMark-ET | RNA-sekvensointi-datan käyttö | http://exon.gatech.edu/braker1.html | [11] |
| Seqping | 2017 | AUGUSTUS, GlimmerHMM, MAKER2, SNAP | Kasvit | https://sourceforge.net/projects/seqping/ | [6] |

[a] Alkuperäinen MAKER julkaistiin vuonna 2007 [5].

Taulukko 2: Tietoja kappaleessa 7 esitellyistä geenientunnistusohjelmistoputkistoista.




Kuva 5: Taulukossa 3 esitellyt herkkyydet ja tarkkuudet visualisoituna karteesiseen koordinaatistoon. $x$-koordinaatti kuvaa herkkyyttä, kun taas $y$ tarkkuutta.

| Ohjelmisto | Eksonitaso | | Geenitaso | | Käytetty datakokoelma | Huomioita | Lähteet |
|---|---|---|---|---|---|---|---|
| | *Sn* | *Sp* | *Sn* | *Sp* | | | |
| SNAP[a] | 78,2 | 67,2 | 50,8 | 37,5 | Ensembl 14.3.1 | Koulutettu ohjatusti pelkällä DNA-datalla | [16] |
| AUGUSTUS | 69,5 | 82,2 | 48,7 | 59,0 | FlyBase r5.55 | Koulutettu ohjatusti pelkällä DNA-datalla | [11] |
| AUGUSTUS$_{gcp}$ | 80,3 | **89,8** | **74,5** | **73,2** | FlyBase r6.04 | Koulutettu puoliohjatusti; ennustettu samanaikaisesti neljän eri *Drosophila*-lajikkeen geenejä, käytetty apuna lajien evoluutiopuuta sekä RNA-sekvensointidataa | [15] |
| GeneMark-ES | 86,7 | 76,9 | 57,9 | 57,3 | FlyBase r5 [sic] | Koulutettu ohjaamattomasti pelkällä DNA-datalla | [17] |
| GeneMark-ET | **87,2** | 82,9 | 63,6 | 61,0 | FlyBase r5 [sic] | Koulutettu puoliohjatusti RNA-sekvensointidatan avulla | [17] |
| MAKER2 | 66,5 | 66,9 | 55,2 | 46,3 | FlyBase r5.55 | Koulutettu puoliohjatusti RNA-sekvensointidatan avulla | [11] |
| BRAKER1 | 75,0 | 81,7 | 64,9 | 59,4 | FlyBase r5.55 | Koulutettu ohjaamattomasti, käytetty apuna RNA-sekvensointidataa | [11] |

[a] Luvut vuodelta 2004; nykyinen SNAP-versio suoriutunee paremmin.

Taulukko 3: Tähän taulukkoon on kerätty herkkyyksiä ($Sn$) sekä tarkkuuksia ($Sp$) suosituista, Markovin piilomallia käyttävistä ja aitotumaisille hyvin soveltuvista geenientunnistusohjelmistoista. Arvot ovat prosenttiosuuksia. Parhaimmat arvot on lihavoitu. Vertailudatana useimmissa tapauksessa on käytetty FlyBaseen (`http://flybase.org/`) talletettuja *Drosophila melanogasterin* eli banaanikärpäsen geenejä. SNAP-ohjelmiston tapauksessa banaanikärpäsen vertailugenomi on haettu Ensembl-tietokannasta (`https://www.ensembl.org/index.html`).

ennustettu, jos sen kaikki eksonit on ennustettu oikein. Herkkyys siis mittaa, kuinka suuri osa ennustettavista piirteistä löydetään, kun taas tarkkuutta silmällä pitämällä varmistetaan, ettei virheellisesti tunnistettujen piirteiden määrä ole liian suuri.

Taulukossa 3 on vertailtu muutamien geenientunnistusohjelmistojen herkkyyksiä sekä tarkkuuksia. Taulukon 3 luvut on visualisoitu kuvan 5 karteesiseen koordinaatistoon.

# 8 Geenien tunnistamisen Markovin piilomalleilla nykytila

Minkälainen on siis nykytiedon valossa toimivin mahdollinen geenejä tunnistava Markovin piilomalli?

Toimivimman mallin yleistä runkoa voi lähteä rakentamaan kappaleen 5.1 mukaisella rakenteella; useat geenejä tunnistavat Markovin piilomallit noudattavat enemmän tai vähemmän sitä.

Taulukosta 1 huomataan, että lähes poikkeuksetta geenejä tunnistettaessa päädytään käyttämään yleistettyä Markovin piilomallia. Tämän syynä on epäilemättä se, että geometrisesti vähenevät todennäköisyysjakaumat harvoin kuvaavat biologisia prosesseja – ensimmäinen askel hyvään prosessin kuvaukseen on päästä niistä eroon, ja tämä onnistuu erinomaisesti yleistetyllä Markovin piilomallilla. Mukaan kannattaa sekoittaa myös vähän interpoloitua Markovin piilomallia – pidemmälle menneisyyteen katsomisesta on usein lähinnä hyötyä.

Taulukosta 3 puolestaan huomataan, että mitä useampia erilaisia datalähteitä voidaan hyödyntää, sitä parempia lopputuloksia voidaan saavuttaa. Nähdään, että puoliohjattu ja ohjaamaton koulutus on jopa ohjattua koulutusta tehokkaampaa, kun erilaista dataa on paljon saatavilla. Toimivimman Markovin piilomallin tulisi siis pystyä hyödyntämään mahdollisimman montaa datatyyppiä sekä koulutuksen että tunnistamisen aikana – tässä kenties pari-Markovin piilomallimaiset ominaisuudet voivat auttaa.

# 9 Yhteenveto

Tämä tutkielma alkoi tutustumalla geeneihin ja niiden löytämiseen. Seuraavaksi tutustuttiin Markovin piilomalleihin sekä niiden perusongelmiin ja -algoritmeihin ja huomattiin, että ne ovat yksinkertainen mutta – hieman jalostettuna – tehokas ratkaisu geenien löytämisen ongelmaan.

Tutkielmassa käytiin läpi geenejä tunnistavan Markovin piilomallin perusrakenne sekä muutama geenien tunnistuksessa usein käytetty Markovin piilomallin variantti. Tämän jälkeen tutustuttiin DNA:n lisäksi geenien tunnistuksessa käytettyyn dataan, sekä kaikkea aikaisemmin läpikäytyä hyödyntäviin ohjelmistoihin. Lopuksi tiivistet-

tiin geenejä tunnistavien Markovin piilomallien nykytila tarkastelemalla, minkälainen on toimivimmaksi todettu geenintunnistuspiilomalli.

Toimivinta Markovin piilomallia etsiessä huomattiin, että monien erilaisten syötedatojen käyttäminen on hyödyllistä. Tästä voidaan päätellä, että Markovin piilomalleilla tapahtuvan geenien tunnistamisen kehittäminen ei ole vain tietojenkäsittelytieteen tutkimuksen harteilla. Suurta apua saadaan kehittyvistä biologisista menetelmistä. Näin saatua dataa ei voi kuitenkaan suoraan syöttää vanhoihin ohjelmistoihin, vaan tarvitaan menetelmiä, kenties eteenpäin kehitettyjä Markovin piilomalleja, joiden avulla uutta tietoa voidaan hyödyntää geenien tunnistamiseen.

Moisès Burset ja Roderic Guigo totesivat vuonna 1996 vertaillessaan senaikaisia geenientunnistusohjelmistoja, että matkaa ohjelmistoon, joka saa kokonaan ratkaistua genomin, on vielä paljon [4]. Geenien tunnistus, varsinkin Markovin piilomalleja käyttäen, on edennyt pitkän matkan sitten heidän tutkimuksensa. Mikään geenientunnistusohjelmisto ei kuitenkaan onnistu vieläkään kokonaisessa genominratkaisussa – kenties Markovin piilomalleista löytyy tulevaisuudessa ratkaisu joka saavuttaa, Bursetin ja Guicon sanoin, tämän ”geenien tunnistuksen lopullisen päämäärän”.

# Lähteet


1 Bethesda (MD): National Library of Medicine (US), National Center for Biotechnology Information. Genome [Internet]. Saatavilla: https://www.ncbi.nlm.nih.gov/genome/, 2012 -. [viitattu 2018 05 07].

2 Bilmes, J. A., et al. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. *International Computer Science Institute 4*, 510 (1998), 126.

3 Borodovsky, M., ja McIninch, J. GENMARK: parallel gene recognition for both DNA strands. *Computers & chemistry 17*, 2 (1993), 123–133.

4 Burset, M., ja Guigó, R. Evaluation of gene structure prediction programs. *Genomics 34*, 3 (1996), 353 – 367.

5 Cantarel, B. L., Korf, I., Robb, S. M., Parra, G., Ross, E., Moore, B., Holt, C., Alvarado, A. S., ja Yandell, M. MAKER: an easy-to-use annotation pipeline designed for emerging model organism genomes. *Genome research 18*, 1 (2008), 188–196.

6 Chan, K.-L., Rosli, R., Tatarinova, T. V., Hogan, M., Firdaus-Raih, M., ja Low, E.-T. L. Seqping: gene prediction pipeline for plant genomes using self-training gene models and transcriptomic data. *BMC Bioinformatics 18*, 1 (Tammikuu 2017), 1–7.

7 Chen, F.-C., ja Li, W.-H. Genomic divergences between humans and other hominoids and the effective population size of the common ancestor of humans

and chimpanzees. *The American Journal of Human Genetics 68*, 2 (2001), 444 – 456.

8 Cristianini, N., ja Hahn, M. W. *Introduction to Computational Genomics, a Case Studies Approach*. Cambridge University Press, 2007.

9 Gilks, W., Richardson, S., ja Spiegelhalter, D. *Markov Chain Monte Carlo in Practice*. Chapman & Hall/CRC, 1996.

10 Gross, S. S., ja Brent, M. R. Using multiple alignments to improve gene prediction. *Journal of Computational Biology 13*, 2 (2006), 379–393. PMID: 16597247.

11 Hoff, K. J., Lange, S., Lomsadze, A., Borodovsky, M., ja Stanke, M. BRAKER1: Unsupervised RNA-Seq-based genome annotation with GeneMark-ET and AUGUSTUS. *Bioinformatics 32*, 5 (2016), 767–769.

12 Holt, C., ja Yandell, M. MAKER2: an annotation pipeline and genome-database management tool for second-generation genome projects. *BMC Bioinformatics 12*, 1 (Joulukuu 2011), 491.

13 Keller, O., Kollmar, M., Stanke, M., ja Waack, S. A novel hybrid gene prediction method employing protein multiple sequence alignments. *Bioinformatics 27*, 6 (2011), 757–763.

14 Kelley, D. R., Liu, B., Delcher, A. L., Pop, M., ja Salzberg, S. L. Gene prediction with glimmer for metagenomic sequences augmented by classification and clustering. *Nucleic Acids Research 40*, 1 (2012), e9.

15 König, S., Romoth, L. W., Gerischer, L., ja Stanke, M. Simultaneous gene finding in multiple genomes. *Bioinformatics 32*, 22 (2016), 3388–3395.

16 Korf, I. Gene finding in novel genomes. *BMC Bioinformatics 5*, 1 (Toukokuu 2004), 59.

17 Lomsadze, A., Burns, P. D., ja Borodovsky, M. Integration of mapped RNA-Seq reads into automatic training of eukaryotic gene finding algorithm. *Nucleic Acids Research 42*, 15 (2014), e119.

18 Lomsadze, A., Ter-Hovhannisyan, V., Chernoff, Y. O., ja Borodovsky, M. Gene identification in novel eukaryotic genomes by self-training algorithm. *Nucleic Acids Research 33*, 20 (2005), 6494–6506.

19 Mathé, C., Sagot, M., Schiex, T., ja Rouzé, P. Current methods of gene prediction, their strengths and weaknesses. *Nucleic Acids Research 30*, 19 (2002), 4103–4117.

20 Mattick, J. S., ja Makunin, I. V. Non-coding RNA. *Human Molecular Genetics 15*, täydennysosa 1 (2006), R17–R29.

21 Minoche, A. E., Dohm, J. C., Schneider, J., Holtgräwe, D., Viehöver, P., Montfort, M., Rosleff Sörensen, T., Weisshaar, B., ja Himmelbauer, H. Exploiting single-molecule transcript sequencing for eukaryotic gene prediction. *Genome Biology 16*, 1 (Syyskuu 2015), 184.

22 Pachter, L., Alexandersson, M., ja Cawley, S. Applications of generalized pair hidden markov models to alignment and gene finding problems. *Journal of Computational Biology 9*, 2 (2002), 389–399. PMID: 12015888.

23 Reid, I., O'Toole, N., Zabaneh, O., Nourzadeh, R., Dahdouli, M., Abdellateef, M., Gordon, P. M., Soh, J., Butler, G., Sensen, C. W., ja Tsang, A. SnowyOwl: accurate prediction of fungal genes by using RNA-Seq and homology information to select among ab initio models. *BMC Bioinformatics 15*, 1 (Heinäkuu 2014), 229.

24 Salzberg, S. L., Delcher, A. L., Kasif, S., ja White, O. Microbial gene identification using interpolated markov models. *Nucleic Acids Research 26*, 2 (1998), 544–548.

25 Stanke, M., Schöffmann, O., Morgenstern, B., ja Waack, S. Gene prediction in eukaryotes with a generalized hidden markov model that uses hints from external sources. *BMC Bioinformatics 7*, 1 (Helmikuu 2006), 62.

26 Stanke, M., ja Waack, S. Gene prediction with a hidden markov model and a new intron submodel. *Bioinformatics 19*, täydennysosa 2 (2003), ii215–ii225.

27 Testa, A. C., Hane, J. K., Ellwood, S. R., ja Oliver, R. P. Codingquarry: highly accurate hidden markov model gene prediction in fungal genomes using RNA-seq transcripts. *BMC Genomics 16*, 1 (Maaliskuu 2015), 170.

28 Wang, Z., Gerstein, M., ja Snyder, M. RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics 10* (2009), 57.

29 Yoon, B.-J. Hidden markov models and their applications in biological sequence analysis. *Current Genomics 10*, 6 (09 2009), 402–415.