Kaksi pisteytystapaa DNA-sekvenssien luokitteluun

Marjo Salinto
marjo.salinto@cs.helsinki.fi

Tiedon louhinta biomolekyyliaineistoista
Helsingin yliopisto, tietojenkäsittelytieteen laitos
Raportti C-2003-52, s. 31-39, marraskuu 2003

Tiivistelmä

DNA-sekvenssien luokittelu on tärkeä ongelma laskennallisessa biologiassa. Luokittelija ennustaa kuuluuko luokittelematon sekvenssi johonkin luokkaan C vai ei. Tässä seminaariraportissa tarkastellaan kahta pistemäärien laskutapaa DNA-sekvenssien luokittelua varten. Pisteytystavat (luokittelijat) eroavat toisistaan tavassa käsitellä perussekvenssejä ja tavassa laskea opetussekvensseille ja luokittelemattomille sekvensseille pistemääriä.

Hahmopohjainen luokittelija soveltaa hahmojen etsimisalgoritmia aktiivisten hahmojen löytämiseksi perusaineistosta. Luokiteltavan sekvenssin pistemäärä perustuu luokiteltavan sekvenssin ja perussekvenssien hahmojen pisimmän yhteisen osamerkkijonon pituuteen.

Sormenjälkipohjainen luokittelija soveltaa hajautukseen perustuvaa sormenjälkitekniikkaa. Sekvensseille generoidaan osasekvenssit eli segmentit ja niille sormenjäljet. Sen jälkeen hajautetaan kunkin segmentin sormenjälki tiedostoon. Luokiteltavalle sekvenssille ja perussekvensseille tehdään segmentointi ja sormenjäljet samalla tavalla. Luokiteltavan sekvenssin sormenjäljet hajautetaan käyttäen samaa hajautusfunktioita kuin perussekvenssien tapauksessa. Tämän jälkeen verrataan luokiteltavan sekvenssin ja perussekvenssien sormenjälkiä toisiinsa, ja kun sormenjäljet täsmäävät kasvatetaan vastaavan perussekvenssin pistemäärää. Luokiteltavan sekvenssin kohdistaminen perussekvenssin korkeimman pistemäärän tuottavan paikan kanssa tuottaa parhaimman kohdistamisen näiden kahden sekvenssin välillä. Luokiteltavan sekvenssin pistemäärä perustuu perussekvenssien pistemääriin.

Empiirisesti on osoitettu molempien pisteytysalgoritmien toimivan tehokkaasti. Niillä on myös hyvä suorituskyky.

1 Yleistä DNA-sekvenssien luokittelusta

Tämä seminaariraportti pohjautuu suurelta osin kirjan "Pattern Discovery in Biomolecular Data: Tools, Techniques and Applications" [WSS99] neljänteen lukuun, joten lähde mainitaan myöhemmin vain silloin, kun on käytetty jotain muuta lähdettä.

DNA-sekvenssillä tarkoitetaan nukleiinihapon nukleotidijärjestystä. DNA-sekvenssien luokittelu on tärkeä ongelma laskennallisessa biologiassa. Luokittelemattomalla sekvenssillä S luokittelija ennustaa kuuluuko sekvenssi tiettyyn luokkaan C. Avaintekijä tämän päivän suurten tietomäärien hallitsemiseksi ja analysoimiseksi on tehokkaiden, tarkkojen ja vaihtoehtoisten tekniikoiden saatavuus. Näillä tekniikoilla voidaan saada selville samankaltaisuuksia vastalöydettyjen ja jo erilaisiin kirjastoihin (tietokantoihin) talletettujen sekvenssien välillä [CR93]. Monia tekniikoita on esitetty erilaisten luokittelijoiden tekemisek-

si näistä luokiteltujen sekvenssien kirjastoista. Erilaisilla hajautustaulutekniikoilla ja ad-hoc-heuristiikoilla on pyritty lyhentämään analysointiohjelmistojen suoritusaikoja.

Yleisesti ottaen DNA-sekvenssien luokittelun tekniikat voidaan jakaa kolmeen luokkaan.

1. Konsensuksen etsintä: Otetaan joukko luokan C sekvenssejä ja generoidaan konsensussekvenssi, jota sitten käytetään ei-karakterisoitujen DNA-sekvenssien tunnistamiseen. Konsensussekvenssi kuvastaa nukleotidin yleisintä muotoa kussakin sekvenssin kohdassa [WRSSWY]. (DNA:n neljä nukleotidiä ovat : A (adeniini), C (sytosiini), G (guaniini) ja T (tymiini)).
2. Induktiivinen oppiminen/neuroverkot: Otetaan joukko luokan C sekvenssejä ja joukko sekvenssejä, jotka eivät kuulu luokkaan C ja sitten oppimistekniikoita käyttäen johdetaan sääntö, joka määrää kuuluuko luokittelematon sekvenssi S luokkaan C vai ei.
3. Sekvenssien kohdistus: Kohdistetaan luokittelematon sekvenssi S luokkaan C kuuluvien sekvenssien kanssa käyttäen olemassa olevia välineitä, kuten FASTA ja liitetään S luokkaan C, jos S:n paras kohdistamispistemäärä on riittävän korkea.

Tässä raportissa tarkastellaan kahta menetelmää pistemäärien laskemiseksi DNA-sekvenssien luokittelua varten: hahmojen etsimisalgoritmiin perustuva menetelmä ja hajautukseen perustuva sormenjälkitekniikka. Ensimmäinen menetelmä esitellään melko lyhyesti ja toinen tarkemmin.

Luvussa 2 esitettävien luokittelijoiden käyttö perustuu seuraavaan lähestymistapaan. Ensin valitaan satunnaisesti sekvenssijoukko $\mathcal{B}$ luokasta C siten, että jokaisella sekvenssillä on sama todennäköisyys tulla poimituksi otokseen [WMSSCL96]. Otanta tehdään palauttamatta, jolloin kukin sekvenssi tulee enintään kerran otokseen. Tätä valittua sekvenssijoukkoa kutsutaan perusaineistoksi. Sitten otetaan luokasta C toinen sekvenssijoukko, jota kutsutaan positiiviseksi opetusaineistoksi. Jokaiselle positiiviselle opetussekvenssille lasketaan pistemäärä perussekvenssien suhteen. Näin saatua minimipistemäärää kutsutaan positiiviseksi alarajaksi L_p. Seuraavaksi otetaan joukko sekvenssejä, jotka eivät ole luokassa C ja kutsutaan tätä joukkoa negatiiviseksi opetusaineistoksi. Jokaiselle negatiiviselle opetussekvenssille lasketaan pistemäärä perussekvenssien suhteen. Näin saatua maksimipistemäärää kutsutaan negatiiviseksi ylärajaksi U_n. Kullakin tekniikalla on oma tapansa laskea pistemääriä [WRSSWY]. Olkoon $B_{high} = \max \{L_p, U_n\}$ ja $B_{low} = \min\{ L_p, U_n\}$. Kun luokitellaan sekvenssiä S, lasketaan S:n pistemäärä suhteessa perussekvensseihin. Tätä pistemäärää merkitään c:llä. Jos $c \geq B_{high}$, niin S kuuluu luokkaan C. Jos $c \leq B_{low}$, niin S ei kuulu luokkaan C. Jos $B_{low} < c < B_{high}$, niin ei oteta kantaa, kuuluuko S luokkaan C vai ei.

Seuraavaksi esitettävät kaksi luokittelijaa eroavat toisistaan tavassa käsitellä perussekvenssejä ja tavassa laskea opetussekvensseille ja luokittelemattomille sekvensseille pistemääriä.

2 Kaksi luokittelijaa DNA-sekvenssien luokittelemiseksi

2.1 Hahmopohjainen luokittelija

Hahmot, joita sekvenssijoukoista etsitään ovat muotoa $*X_1*X_2*\ldots$, missä X_1, X_2,… ovat sekvenssien segmenttejä eli osasekvenssejä ja *:t mitä tahansa merkkejä, joilla ei tässä

yhteydessä ole merkitystä. Hahmon esiintyvyys eli aktiivisuus tarkoittaa niiden sekvenssien määrää, joissa hahmo esiintyy sallitulla etäisyydellä. Tässä esitettävää luokittelijaa kutsutaan hahmopohjaiseksi luokittelijaksi, koska se soveltaa hahmojen etsimisalgoritmia aktiivisten hahmojen löytämiseksi perusaineistosta. Algoritmi koostuu kahdesta vaiheesta: (1) etsitään kandidaattisegmenttejä perussekvenssijoukosta S poimitusta pienestä otoksesta ja (2) yhdistetään kandidaattisegmentit kandidaattihahmoiksi ja arvioidaan hahmon aktiivisuus koko sekvenssijoukossa S. Algoritmi on esitetty tarkemmin lähteenä olevan kirjan luvussa 4.1.2 [WSS99].

Esimerkki 1 [WRSSWY]. Olkoon hahmon pituus suurempi kuin kuusi eli hahmo sisältää vähintään seitsemän nukleotidiä, esiintyvyys on kolme eli hahmo löytyy kolmesta perussekvenssistä ja yksi mutaatio (täsmäämättömyys, poisto tai lisäys) sallitaan, kun täsmätään hahmoa perussekvensseihin. Tällöin seuraavista kolmesta perussekvenssistä löytyy kaksi aktiivista hahmoa GCCGGGC ja GCCAGGC, jotka on perussekvensseissä alleviivattu

GGAGAGGCCGGGCGTGTGCCGGTAC
GGCCAGGCGGCAGATCTTGACCAGG
TGTAATCAGAGCGCCAGGCAAACAT.

Olkoon $\mathcal{R}$ joukko aktiivisia hahmoja, jotka on löydetty perussekvenssijoukosta $\mathcal{B}$. Luokiteltavan sekvenssin S ja hahmon $P \in \mathcal{R}$ välinen pistemäärä score(S,P) = |L|, missä L on S:n ja P:n pisin yhteinen osamerkkijono. Score(S,P) on siis nukleotidien määrä L:ssä.

S:n pistemäärä suhteessa perussekvensseihin määritellään S:n ja P:n välisten pistemäärien maksimilla kerrottuna 100:lla seuraavasti

$$\text{score}(S) = \max\{\text{score}(S,P) \mid P \in \mathcal{R}\} \times 100.$$

Esimerkki 2. Esimerkin 1 perussekvensseillä ja luokiteltavalla sekvenssillä S=GCCGTTTTTTTTTTTTTTTTTTT saadaan hahmon GCCGGGC tapauksessa pistemääräksi 4 ja hahmon GCCAGGC tapauksessa pistemääräksi 3. Tällöin S:n pistemäärä on 400.

2.2 Sormenjälkipohjainen luokittelija

Toinen luokittelija, jota kutsutaan sormenjälkipohjaiseksi luokittelijaksi, ottaa pistemäärien laskemiseksi käyttöön hajautukseen perustuvan sormenjälkitekniikan [CR93, WMSSCL96]. Olkoon S sekvenssi ja seg segmentti eli S:n osasekvenssi. Segmentin seg sormenjälki f on mahdollisesti epäjatkuva segmentin seg osasekvenssi, joka alkaa segmentin ensimmäisestä merkistä (nukleotidistä). Väli paikassa (positiossa) p tarkoittaa sitä, että nukleotidiä segmentin paikasta p ei poimita. Sallittujen välien määrä sormenjäljissä määritellään parametrilla gap.

Esimerkki 3. Olkoon S=ACGTTGCA ja S:n segmentti seg=ACGTTG. ATT on segmentin seg kolmen nukleotidin sormenjälki kahdella välillä (yksi väli paikassa 2 ja toinen paikassa 3).

2.2.1 Sormenjälkitiedostojen muodostaminen

Annetusta perussekvenssijoukosta $\mathcal{B}$ poimitaan kunkin perussekvenssin segmentit ja hajautetaan kunkin segmentin sormenjäljet tiedostoon. Olkoon S sekvenssi perusaineistosta $\mathcal{B}$. Otetaan kaikki S:n segmentit seg, jotka ovat n:n pituisia ja generoidaan sormenjäljet segmentistä seg. Sormenjälkien pituus vaihtelee 2:sta n-1:een. Sormenjälkiä ei käytetä suoraan tiedostossa indeksinä, vaan tavallisesti indeksinä käytetään kokonaislukuarvoja [CR93]. Hajautusfunktion valinta on kriittinen tekijä algoritmin tarkkuuden ja tehokkuuden kannalta. Hajautusfunktioita h_k, $2 \leq k \leq n-1$ käytetään hajauttamaan kaikki k:n pituiset sormenjäljet sormenjälkitiedostoon $\mathcal{F}_k$. Tiedostossa kuhunkin sormenjälkeen f liittyy kokonaislukupari (x,y). Tämä pari toimii f:n paikan osoittimena. X tarkoittaa sitä, että f on generoitu $\mathcal{B}$:n x:nnen sekvenssin segmentistä ja y tarkoittaa, että ensimmäinen f:n nukleotidi esiintyy sekvenssin y:nnessä paikassa.

Esimerkki 4. Olkoon kolme perussekvenssiä

S_1 = ACGTTGCA
S_2 = ACCAGTG
S_3 = CGGACTA.

Oletetaan, että segmentin pituus on kuusi. Silloin saadaan seuraavat segmentit S_1:stä

ACGTTG
CGTTGC
GTTGCA.

Otetaan segmentti seg=ACGTTG. Oletetaan, että väli on kaksi (gap=2). Silloin voidaan generoida seuraavat kolmen nukleotidin sormenjäljet

ACG (ei väliä)
AGT (yksi väli paikassa 2)
ACT (yksi väli paikassa 3)
ATT (yksi väli paikassa 2 ja toinen väli paikassa 3)
AGT (yksi väli paikassa 2 ja toinen väli paikassa 4)
ACT (yksi väli paikassa 3 ja toinen väli paikassa 4).

Esimerkiksi sormenjäljet AGG ja ATG eivät täytä väliehtoa, koska niissä on kolme väliä.

Seuraavassa taulukossa on esitetty segmentin kaikki (2-5 nukleotidin pituiset) sormenjäljet [WSS99].

	2:n nuk-leotidin sormen-jäljet	3:n nuk-leotidin sormen-jäljet	4:n nuk-leotidin sormen-jäljet	5:n nuk-leotidin sormenjäl-jet
Väli 0	AC	ACG	ACGT	ACGTT
Väli 1	AG	ACT, AGT	ACGT, AGTT, ACTT	ACGTG, AGTTG, ACTTG
Väli 2	AT	ACT, AGT, ATT	ACGG, ATTG, ACTG, AGTG	

Taulukko 1. Segmentin ACGTTG 2-5 nukleotidin sormenjäljet (segmentin pituus kuusi ja väli 2).

Olkoon f=XYZ kolmen nukleotidin sormenjälki ja hajautusfunktio

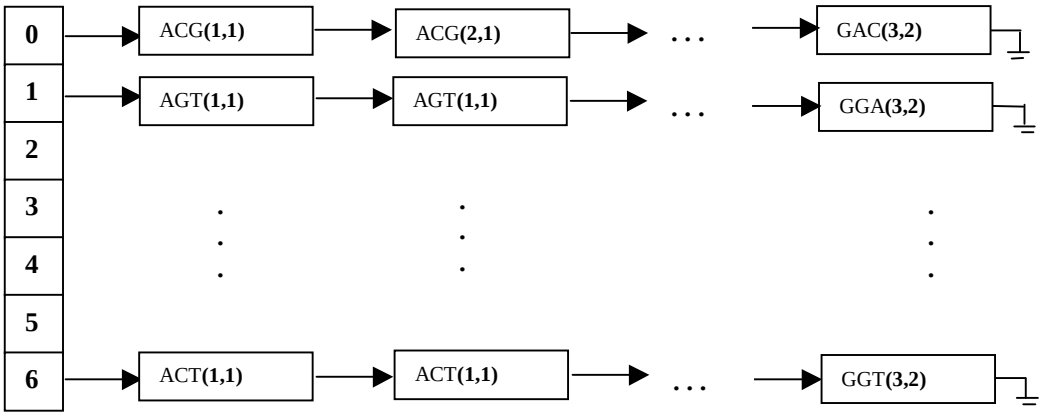
$$h_3(f) = [num(X) \times 4^2 + num(Y) \times 4^1 + num(Z)] \bmod 7,$$

missä num(X) on X:n ASCII-arvo - 64. Oletetaan, että generoidut hajautusfunktion arvot (indeksit) noudattavat tasajakaumaa eli jokainen indeksi on yhtä todennäköinen [CR93].

Esimerkki 5. A:n ASCII-arvo on 65, C:n 67, G:n 71 ja T:n 84. Hajautusfunktion arvot esimerkiksi sormenjäljille ACG ja ACT saadaan seuraavasti

$$h_3(ACG) = (1 \times 4^2 + 3 \times 4^1 + 7) \bmod 7 = 35 \bmod 7 = 0$$
$$h_3(ACT) = (1 \times 4^2 + 3 \times 4^1 + 20) \bmod 7 = 48 \bmod 7 = 6.$$

Sormenjälkitiedosto $\mathcal{F}_3$ voidaan esittää kuvan 1 mukaisena rakenteena [WSS99].



Kuva 1. Kolmen perussekvenssin kolmen nukleotidin sormenjälkitiedosto $\mathcal{F}_3$.

Esimerkiksi merkintä ACG(1,1) tarkoittaa, että sormenjälki ACG on generoitu S_1:stä ja se alkaa S_1:n ensimmäisestä paikasta. Merkintä GGA(3,2) tarkoittaa, että sormenjälki GGA on generoitu S_3:sta ja alkaa sen toisesta paikasta. Muut merkinnät vastaavasti. Se, että esimerkiksi AGT(1,1) esiintyy useamman kerran tarkoittaa sitä, että sama sormenjälki voidaan muodostaa samasta paikasta alkaen siten, että siinä on eri määrä välejä (sallituissa rajoissa).

Indeksi (γ) voitaisiin myös muodostaa esimerkiksi seuraavasti, jos jokaiseen merkkiin k:n mittaisessa sormenjäljessä liittyisi yksikäsitteinen arvo ν_i, $0 \leq \nu_i \leq \tau$, missä τ on mahdollisten merkkien määrä (DNA-sekvensseissä mahdolliset merkit ovat A, C, G ja T, joten τ on neljä) [CR93]

$$\gamma = \sum_{i=1}^{k} \upsilon_i \tau^{(i-1)}$$

DNA-sekvensseillä {A,C,G,T} -> {0,1,2,3}. Esimerkiksi sormenjälki AATCGT, voitaisiin muuttaa muotoon 003123 ja silloin indeksiksi saataisiin $0\times4^0+0\times4^1+3\times4^2+1\times4^3+2\times4^4+3\times4^5 = 3696$.

2. 2. 2 Algoritmi pistemäärien laskemiseksi

Kun lasketaan sekvenssin S pistemäärää, S segmentoidaan samalla tavalla kuin perussekvenssit ja sormenjäljet generoidaan tulossegmenteistä. Sen jälkeen hajautetaan sormenjäljet käyttäen samaa hajautusfunktioita kuin perussekvensseillä.

Kun S:n sormenjälki paikassa p ja perussekvenssin sormenjälki paikassa q täsmää, lisätään yksi perussekvenssin sopivan paikan pistemäärään kuten seuraavassa esitetään. Olkoon sekvenssi S=CGATGCAT ja perussekvenssi S_1=ACGTTGCA. Kolmen nukleotidin sormenjälki on TGC alkaen S_1:n paikasta 5 ja S:n paikasta 4 (q=5 ja p=4). Lisätään yksi paikan q-p+1=5-4+1=2 pistemäärään.

Intuitiivisesti: Jos kohdistetaan ensimmäinen S:n nukleotidi S_1:n toisen nukleotidin kanssa, voidaan nähdä näiden kahden sormenjäljen yhteneväisyys

S_1: ACGTTGCA
S : CGATGCAT.

Yleisesti ottaen, jos kohdistetaan S:n ensimmäinen nukleotidi S_1:n k:nnen paikan kanssa, ja paikan k pistemäärä on n, voidaan nähdä n yhteneväisyyttä sormenjälkiä kohdistettaessa. Siten S:n ensimmäisen nukleotidin ja S_1:n korkeimman pistemäärän tuottavan paikan kohdistaminen tuottaa parhaimman kohdistamisen näiden kahden sekvenssin välillä [CR93].

Algoritmi pistemäärien laskemiseksi pseudokoodina on esitetty kuvassa 2 [WSS99]. Algoritmin syötteenä on sekvenssi S, joukko $\mathcal{B}$ perussekvenssejä ja $\mathcal{B}$:n sormenjälkitiedostot. Algoritmin tuloksena on histogrammi pistemääristä.

```
/* F sisältää S:stä generoidut sormenjäljet. */
for jokainen F:n sormenjälki f do
  begin
    /* Olkoon f:n pituus k. */
    hajauta f hajautusfunktioita h_k käyttäen sormenjälkitiedostoon F_k;
    for jokainen f:n ja perussekvenssin sormenjäljen täsmääminen F_k:ssa do
      begin
        /* (i,q) osoittaa perussekvenssin i sormenjäljen ensimmäisen nukleotidin. */
        /* Oletetaan, että f:n ensimmäinen nukleotidi sijaitsee S:n paikassa p. */
        lisätään yksi i:nnen perussekvenssin paikan q-p+1 pistemäärään;
      end;
  end;
```

Kuva 2. Algoritmi pistemäärien laskemiseksi pseudokoodina.

B on $\mathcal{B}$:n perussekvenssi ja p on B:n paikka $1 \leq p \leq |B|$. Score(B[p]) on pistemäärien kokonaismäärä, mikä on lisätty paikkaan p. Perussekvenssin B pistemäärä score(B) on B:n eri paikkojen pistemäärien maksimi. S:n pistemäärä perussekvenssien suhteen on kaikkien perussekvenssien pistemäärien maksimi kerrottuna skaalaustekijällä 100/|S|. Skaalaustekijällä 'vakioidaan' S:n pituuden vaikutusta [WRSSWY]. Mitä pidempi merkkijono sitä todennäköisemmin enemmän pistemääriä. Pistemäärien määräytyminen voidaan esittää seuraavasti

score(B) = max {score(B[p]) | 1<=p<=|B|} ja
score(S) = max {score(B) | B $\in$ $\mathcal{B}$ } / |S|×100.

Esimerkki 6. Perussekvenssin S_1 pistemäärä esimerkiksi paikassa -1 (1-3+1) tarkoittaa sitä, että kohdistetaan sekvenssit seuraavasti

```
S1:      ACGTTGCA
S :  CGATGCAT,
```

jolloin sormenjäljet AG, AT ja AC täsmäävät (väli=2). Näin ollen pistemääräksi saadaan kolme. Paikan 2 (2-1+1 ja 3-2+1) pistemäärä tarkoittaa sitä, että kohdistetaan sekvenssit seuraavasti

```
S1:   ACGTTGCA
S :     CGATGCAT,
```

jolloin seuraavat sormenjäljet täsmäävät

CGT	2 krt	CGG	1 krt
CTG	2 krt	CGTG	2 krt
CGTC	2 krt	CGGC	1 krt
CTGC	2 krt	CG	1 krt

CT	2 krt	GTG	2 krt
GTC	2 krt	GGC	1 krt
GTGC	2 krt	GTGA	2 krt
GGCA	1 krt	GTCA	2 krt
GT	2 krt	GG	1 krt
CGTGC	2 krt	GTGCA	2 krt,

joten pistemääräksi tulee 34.

Algoritmi tuottaa tulokseksi S_1:lle pistemäärän 34 (max {3,0,0,34,0}), S_2:lle pistemäärän 7 ja S_3:lle pistemäärän 19. Score(B) on max {34,7,19}=34 ja score(S) on 34/8×100=425.

2. 2. 3 Algoritmien ominaisuuksista

Algoritmin tehokkuutta voidaan mitata seuraavilla mittareilla

$$PR = NumCorrect/NumTest \times 100\% \text{ ja}$$

$$NR = NumNoOpinion/NumTest \times 100\%.$$

PR (precision rates) kuvaa algoritmin tarkkuutta ja NR (no-opinion rates) konservatiivisuutta eli sitä kuinka suuri osa sekvensseistä jätetään luokittelematta. NumCorrect on oikein luokiteltujen testisekvenssien määrä, NumTest on testisekvenssien kokonaismäärä ja NumNoOpinion on niiden testisekvenssien määrä, joiden luokitteluun ei ole otettu kantaa eli ne on jätetty luokittelematta.

Väärillä positiivisilla tarkoitetaan niitä, jotka luokitellaan väärin kuuluviksi tarkasteltavaan luokkaan ja väärillä negatiivisilla niitä, jotka luokittelijan mukaan eivät kuulu tarkasteltavaan luokkaan, mutta oikeasti ne kuuluisivat.

Esimerkki 7 [WRSSWY]. Raportissa esitettyjä kahta algoritmia on verrattu FASTA:aan ja tulokseksi on saatu, että FASTA on konservatiivisempi menetelmä kuin edellä esitetyt, koska se antaa helpommin luokittelusta tulokseksi 'ei kantaa', mutta se tuottaa erittäin vähän (esimerkissä 0) vääriä positiivisia ja vääriä negatiivisia tuloksia. FASTA:lla saatiin esimerkkitapauksessa pienempi osa luokiteltua oikein (alhaisempi PR) kuin tässä esitellyillä kahdella menetelmällä. Hahmopohjainen luokittelija osoittautui hieman sormenjälkipohjaista luokittelijaa paremmaksi, koska se antoi yhden väärän positiivisen tuloksen vähemmän eikä se jättänyt yhtään luokittelematta, kun sormenjälkipohjainen luokittelija jätti yhden.

Testeissä on osoitettu, että perussekvenssien määrän sekä positiiviseen ja negatiiviseen opetusaineistoon kuuluvien sekvenssien määrän suhteen esitetyt algoritmit ovat aika stabiileja eikä näiden tekijöiden muutoksilla ole suurta vaikutusta tuloksiin, kunhan vain huolehditaan siitä, että määrät eivät ole liian pieniä (100:a on käytetty suositeltavana minimiarvona).

Korkeat esiintyvyysluvut (aktiivisuus), lyhyet aktiiviset hahmot ja suuret etäisyysarvot (sallitaan paljon mutaatioita) heikentävät hahmopohjaisen luokittelijan suorituskykyä. Kun esiintyvyys kasvaa, niin niiden osuus kasvaa, joita ei pystytä luokittelemaan ja oikein luokiteltavien osuus pienenee. Aktiivisten hahmojen pituus 11 ja enintään yhden etäisyydet ovat testeissä osoittautuneet hyviksi arvoiksi algoritmin toimivuuden kannalta.

Sormenjälkipohjaisen luokittelijan tapauksessa segmentin pituudella ja sormenjäljen välillä ei ole olennaista vaikutusta algoritmin tulokseen, joskin pienillä väleillä algoritmi on

paljon nopeampi kuin suurilla väleillä. Testeissä vähintään kuuden pituiset segmentit ilman väliä ovat osoittautuneet hyviksi.

3 Pohdintaa

Empiirisesti on osoitettu molempien tässä esitettyjen algoritmien toimivan tehokkaasti. Niillä on hyvä suorituskyky esimerkiksi FASTA:an verrattaessa.

Tässä esitettyjen algoritmien ja konsensuslähestymistavan yhdistämisellä on myös saatu hyviä tuloksia.

Artikkelin kirjoittajat mainitsivat itsekin puutteena sen, että luokittelijoiden tuloksissa ei ole luottamusvälejä kuvaamassa saatujen estimaattien luotettavuutta ja tarkkuutta.

Hahmopohjaisessa luokittelijassa pistemäärän laskemisessa keskeisintä on luokiteltavan sekvenssin ja perussekvensseistä löydettyjen hahmojen pisimmän yhteisen merkkijonon pituus, kun taas sormenjälkipohjaisessa luokittelijassa keskeisintä ei ole yhteisten sormenjälkien pituus vaan lukumäärä. Esimerkiksi score(S) voidaan näillä luokittelijoilla laskea eri perussekvenssistä, jos yhdessä on yksi pitkä yhteinen hahmo ja toisessa useampia ja lyhyempiä yhteisiä sormenjälkiä. Itse luokittelun tuloksen kannalta tämä ei kuitenkaan ole ratkaisevaa, vaan nämä kaksi luokittelijaa vaikuttavat aika toistensa kaltaisilta.

Hajautusfunktion valinta on mainittu kriittiseksi tekijäksi, mutta löydetyissä lähteissä ei ole kuitenkaan käsitelty sitä, miten se pitäisi valita, vaan on esimerkeillä esitetty, minkälainen hajautusfunktio voisi olla.

Viitteet

[CR93] A. Califano and I. Rigoutsos. FLASH: A Fast Look-Up Algorithm for String Homology. *In Proceedings of the First International Conference on Intelligent Systems for Molecular Biology*, pages 56-64, Menlo Park, Calif., 1993.

[WMSSCL96] J. T. L. Wang, T. G. Marr, D. Shasha, B. A. Shapiro, G. W. Chirn, and T. Y. Lee. Complementary classification approaches for protein sequences. *Protein Engineering*, 9(5):381-386, 1996.

[WSS99] Edited by J. T. L. Wang, B. A. Shapiro, and D. Shasha. *Pattern Discovery in Biomolecular Data: Tools, Techniques and Applications*. Oxford University Press, New York, Oxford, 1999.

Internet-viite

[WRSSWY] J. T. L. Wang, S. Rozen, B. A. Shapiro, D. Shasha, Z. Wang, and M. Yin. New Techniques for DNA Sequence Classification.