TARTU ÜLIKOOL
MATEMAATIKA-INFORMAATIKATEADUSKOND
Arvutiteaduse instituut
Tarkvarasüsteemide õppetool
Rakendusinformaatika eriala

Marek Zäuram

Kaalumaatriksite otsimise meetodid

Diplomitöö

Juhendaja: Jaak Vilo, PhD

Autor: ..... “.....“ 2004
Juhendaja: ..... “.....“ 2004
Õppetooli juhataja: ..... “.....“ 2004

TARTU 2004

Sisukord

Sissejuhatus

Viimaste aastate jooksul on toimunud tormiline areng väga erinevate organismide genoomide sekveneerimisel, mis omakorda on tugevalt mõjutanud bioinformaatika edasiminekut ja arengut. Üha laialdasemalt on tegevusvaldkonnaks DNA-st signaalide, mis omavad mingisugust bioloogilist tähendust, otsimine. Selleks tarbeks on loodud mitmeid erinevaid tööriistu ja programme. Samuti on toimunud hüppeline areng erinevate andmebaaside koostamisel. Need erinevad tööriistad ja andmebaasid on teinud kättesaadavaks võimaluse geeniregulatsiooni mehhanismide modelleerimiseks ja arusaamiseks, mis on tänapäeva molekulaarbioloogia suurimaid väljakutseid.

Antud töö koosneb kahest osast: esimeses pooles antakse ülevaade erinevatest signaalide (mustrite) esitamiste ja analüüsimise meetoditest. Lähema vaatluse alla on võetud kaalumaatriks ja selle erinevad variatsioonid. Teine pool on praktilisema kallakuga: vaadeldakse mustritest koostatud kaalumaatriksite abil genoomi analüüsi jaoks loodud programmi (realisatsioon C keeles), kaalumaatriksi sekventsile sobitamise jõualgoritmi ja selle edasiarendusi ning genoomi analüüsi tulemuste visualiseerimist.

1 Bioloogiline taust

Transkriptsiooni faktorid seonduvad DNA-le geeni promootori regioonis, et reguleerida ekspressiooni. Transkriptsiooni faktorid on tavaliselt seotud rohkem kui ühe geeni reguleerimisega, kusjuures enamus geene on reguleeritud teatava arvu transkriptsiooni faktorite poolt.

Transkriptsiooni faktoreid siduvad seondumiskohad on küllalt lühikesed ja tavaliselt hõlmavad umbes tosin nukleotiidi. Sellest olenemata võib üks ja sama transkriptsiooni faktor siduda palju erinevaid DNA järjestusi. Paljude järjestuste joondamine peaks teostama õigesti transkriptsiooni faktoreid seondumiskohtade võrdlusi üksteisega. Osad järjestuste positsioonid kalduvad olema tunduvalt konserveeritumad võrreldes teistega – selliseid regioone nimetatakse motiivideks.

Antud töös vaadeldakse just eelnevalt joondatud või mõnel muul viisil saadud transkriptsiooni faktorite esitamise viise ning nendest koostatud kaalumaatriksiga sekventsi analüüsi, et leida uusi ja nende seni avastamata omadusi.

2 Mustrite esitamise võimalused

DNA seondumiskohtade analüüsi ja ennustamise põhiprobleemid võib jagada kaheks:

1. juba avastatud ja tuntud seondumiskohtade põhjal sellise esituse leidmine, mille põhjal oleks võimalik otsida uusi sekventse ja usaldusväärselt ennustada neid kohti, kus asuvad või võivad esineda uued seondumiskohad;
2. ühise faktoriga seondumiskohtade sekventside olemasolul (teadmata on ka ainult nende seondumiskohtade asukohad sekventsides) otsida seondumiskohti sekventsides ning leida esitus lähtudes antud valgu spetsiifilisusest.

2.1 Alamsõned

Lihtsam viis transkriptsioonifaktorite seondumiskohtade kirjeldamiseks on sõned ehk jada tähtedest tähestikust ∑. DNA sekventsid baseeruvad 4 tähelisel tähestikul ∑ = {A, C, G ja T} ning antud kontekstis võib DNA-d vaadelda kui pikka sõnet. Kahjuks ei ole transkriptsioonifaktorid tavaliselt kitsendatud (ingl. k. „*restricted*") üheks täielikult konserveerunud sõneks. Seega selline esitamine on ebapiisav, sest näiteks mingi sõne otsimisel leiab ainult vaste 100%-ilse sobivuse korral.

Toome ära viis alamsõnet (motiivi), mida hakkame edaspidi seondumiskohtade esitusviisidel kasutama.

motiiv 1	ACAATG
motiiv 2	TCAATC
motiiv 3	ACAAGC
motiiv 4	AGAATC
motiiv 5	ACCATC

Tabel 2.1: Näitemotiivid

2.2 Konsensusjärjestus

Konsensusjärjestuse mõistet kasutatakse laialdaselt esitamaks seondumiskohtasid. Motiivi konsensusjärjestus on motiivi esitus, kus kõige sagedamini esineva motiiv isendit kasutatakse kui üldist esitust. Selline esitlusviis on lihtne, kuid tulemused on märkimisväärse informatsiooni kaoga võrreldes esitatud algses joonduses. Selline lihtne esitus ei luba motiivis mittevastavuste olemasolu. Konsensusjärjestusega on lihtne esitada teatud hulka motiive, kuid on keeruline leida konsensusjärjestust, mis oleks optimaalne ennustamaks uute seondumiskohtade esinemisi.

Tabelis 2.1 toodud näitemotiividel oleks konsensusjärjestus järgmine: **ACAATC**

2.3 Regulaaravaldis

Võrreldes konsensusjärjestustega lubavad regulaaravaldised suuremat painduvust motiivide esitamisel. Mittevastavuste olemasolu teataval positsioonil motiivis on esitatud tavakohaste sümbolitega. Regulaaravaldis esitab konsensusjärjestuste perekonda, mis esinevad sarnase tõenäosusega. Hoolimata regulaaravaldiste palju täpsemale esitusele võrreldes

konsensusjärjestustega, esineb sellise esituse kasutamisega osalise informatsiooni kadu, näiteks varjub tõenäosus millega teatav nukleotiid võib esineda mingis kindlas positsioonis.

Sümbol	**Tähendus**	**Aminohape / Tähendus**
G	G	Guaniin
A	A	Adeniin
T	T	Tümiin
C	C	Tsütosiin
R	G või A	
Y	T või C	
M	A või C	
K	G või T	
S	G või C	
W	A või T	
H	A või C või T	Mitte G, järgneb G -le
B	G või T või C	Mitte A, järgneb A-le
V	G või C või A	
D	G või A või T	Mitte C, järgneb C-le
N	G või A või T või C	

Tabel 2.2: Ühetäheliste koodide IUPAC soovitatud esitus [8]

Tabelis 2.1 toodud näitemotiividel oleks regulaaravaldis avaldatav järgmisel kujul: [AT][CG][AC]A[TG][GC]. Siit me näeme, et antud regulaaravaldisega on leitav ka järgmine muster: TGCAGG, kuigi antud muster meie näitemotiivides ei leidu.

Mõned kasulikud näited regulaaravaldistest:

1. Kandilised sulud [] – saab kasutada näitamaks ära hulka erinevate alternatiivsete sümbolitega. “GAT[TA]AG” tähendab kas "GATTAG” või “GATAAG” (see on samane ka “GATWAG”-iga IUPAC-i kodeeringus)
2. Loogad {N arv kordusi} – number, mitu eelnevat sümbolit
   - A{8} tähendab "AAAAAAAA"

- [AG]{8} tähendab "8 korda valikut, kas A või G"
- CGG[ACGT]{11}CCG tähendab "CGG, millele järgneb täpselt 11 korda A, C, G või T, ning lõpus veel CCG"

3. Loogad {N, M} ehk muutuv arv kordusi – vahemik, mitu minimaalsete ja maksimaalsete korduste vahel.
   GATAAG[ACGT]{0,30}GATAAG tähendab "kaks GATAAG, mille vahel võib olla 0 kuni 30-nd A, C, G või T-d"
4. Alternatiivsed võimalused – Kaks alternatiivset võimalust eraldatakse "|" sümboliga.
5. CACGTTTT | CACGTGGG tähendab "kas CACGTGGG või CACGTTTT"

2.4 Kaalumaatriks

Isegi kõige deterministlikumad mustrid ei suuda näidata kogu mustrisse peidetud olulist informatsiooni. Näiteks eeldame, et meil on muster, mis sisaldab esimesel positsioonil 40% juhtudel C ja 60% juhtudel G. Mitmetimõistetav regulaaravaldise sümbol [CG] annab sama tähtsuse mõlemale nukleotiidile. See on tähtis tugevate mustrite juures, kuid samas võib osutuda väga tähtsaks ka nõrkade mustrite korral, kus võib vaja minna kogu olemasolevat informatsiooni, et eristada mustrit suvalisest sekventsist [20].

Motiive on võimalik kirjeldada ka tõenäosuslikult tabelina. Selliselt kirjeldatud tabeleid nimetatakse kaalumaatriksiteks. Kaalumaatriksis esitavad veerud motiivi positsiooni ning read kirjeldavad vastava nukleotiidi esinemise tõenäosust või siis kaalu, millega nad võivad esineda antud positsioonis (või panust, mille antud positsioon vastavalt annab kogu tulemusse).

3 Kaalumaatriksite meetodid

3.1 Joonduse maatriks ehk absoluutsete sageduste maatriks

Erinevad kirjanduslikud allikad nimetavad seda tüüpi maatriksit veel ka „*counts matrix*" ja „*position weight matrix*" (*PWM*). Seda tüüpi maatriks on kõige lihtsam maatriksi tüüp, kus maatriksi välja kaal saadakse valemiga

$$n_{b,\,i} = \sum_{k=1}^{n} C_b(s_{k,i}) \qquad \begin{matrix} b = A,\ C,\ G,\ T \\ i = 1,\ldots,l \end{matrix}$$

$$kus\ C_b(q) = \begin{cases} 1 & kui\ \ b = q \\ 0 & vastupidisel\ juhul, \end{cases} \qquad \textbf{(3.1)}$$

kus i on motiiv pikkusega l, N on motiivide arv; motiivid $s_1,..,s_N$, kus sekvents $s_k = s_{k1},..,s_{kl}$, mille liige s_{ki} on hulgast {A, C, G, T} (seda DNA motiivi puhul). Iga tegur selles maatriksis näitab mitu korda antud nukleotiid on loetletud selles teatavas positsioonis.

Näiteks tabel 3.1 kirjeldab, et tabelis 2.1 toodud näitemotiividel on nukleotiidi „A" loendatud esimeses positsioonis 4 korda. Samuti märkame, et antud sekventsidel on ainult üks positsioon täielikult konserveerunud – vastavalt neljas (nukleotiid „A").

	1	**2**	**3**	**4**	**5**	**6**
A	4	0	4	5	0	0
C	0	4	1	0	0	4
G	0	1	0	0	1	1
T	1	0	0	0	4	0

Tabel 3.1: Absoluutsete sageduste maatriks. Saadud tabelis 2.1 olevatest näitemotiividest.

3.2 Sagedusmaatriks

Sageli nimetakse sagedusmaatriksit ka suhtelise sageduse maatriksiks. Iga motiiv maatriksi positsioonis *i* kirjeldab suhtelist sagedust f_{bi}, millega seda kindlat sümbolit *b* on mõõdetud. See sagedus on võrdne kindlate sümbolite esinemiste arvuga antud joonduse veerus jagatud joondatud sekventside arvuga. Iga veerg sellises maatriksis on diskreetne tõenäosuslik panus, s.t. liites kokku veerus olevad sagedused, saame 1 ning kõik esinemised veerus on suuremad või võrdsed nulliga. Suhteliste sageduste arvutamiseks tuleb kõikides positsioonides iga sümboli esinemiste arv jagada kogu motiivide esinemiste arvuga.

Et arvutada suhtelise sagedustega maatriksist mingisuguse sekventsi tõenäosust, siis peame defineerima sõltumatuse eelduse.
Sõltumatuse eeldus: suvaline sümbol positsioonil *i* on sõltumatu sümbolist, mis esineb suvalises teises positsioonis. Teisisõnu, sümbolid, mis paiknevad kahes erinevas positsioonis ei ole korrelatsioonis. Kuigi see sõltumatuse eeldus ei ole alati realistlik, annab seda õigustada. Esimeseks õigustuseks oleks see, et ta aitab hoida mudelit ja analüüsi tulemus on lihtne. Teiseks on see, et ta omab ennustavat jõudu mõningates (aga eeldavalt mitte kõigis) situatsioonides [9].

Sõltumatuse eelduse saab teha ka täpsetes tõenäosuslikes terminites:
Ütleme, et kaks tõenäosuslikku sündmust E ja F on sõltumatud, kui tõenäosus, et nad mõlemad esinevad, on nende mõlema tõenäosuse korrutis ehk siis $P(E \& F) = P(E) \cdot P(F)$.

Kasutades antud sõltumatuse eeldust saame, et tõenäosus suvaliselt valitud saidil (saidiks nimetame siin seondumiskohta) r_1, ... , r_n on määratud eelnevalt sõnastatud definitsiooni järgi järgmiselt [9]:

$$P(t = r_1, r_2, \ldots, r_n \mid t \text{ on sait}) = P(t_1 = r_1 \,\&\, t_2 = r_2 \,\&\, \ldots \,\&\, t_n = r_n \mid t \text{ on sait})$$
$$= \prod_{j=1}^{n} P(t_j = r_j \mid t \text{ on sait})$$
$$= \prod_{j=1}^{n} A_{r_j, j} \qquad \textbf{(3.2)}$$

Näiteks, kui tahame teada tõenäosust suvaliselt valitud seondumiskohal „ACAATC". Kasutades eelnevat definitsiooni ning tabelit 3.1. saame:

T (t = ACAATC | *t on sait*) = (0.8*0.8*0.8*1.0*0.8*0.8) = 0,32768

Kuigi antud tõenäosus on suhteliselt väikene, on see suurim võimalik tõenäosus suvalise seondumiskoha jaoks, kuna igas positsioonis on esitatud kõige tõenäosuslikum nukleotiid.

	1	**2**	**3**	**4**	**5**	**6**
A	4/5=0.8	0/5=0	4/5=0.8	5/5=1	0/5=0	0/5=0
C	0/5=0	4/5=0.8	1/5=0.2	0/5=0	0/5=0	4/5=0.8
G	0/5=0	1/5=0.2	0/5=0	0/5=0	1/5=0.2	1/5=0.2
T	1/5=0.2	0/5=0	0/5=0	0/5=0	4/5=0.8	0/5=0

Tabel 3.2: Suhteliste ehk relatiivsete sageduste maatriks. Saadud tabelis 2.1 olevatest näitemotiividest ning arvutamiseks on kasutatud tabeli 3.1 absoluutseid sagedusi. Relatiivne sageduse arvutamine: igas positsioonis iga sümboli esinemine jagatud koguarvuga. Iga veerg sellises maatriksis on diskreetne tõenäosuslik panus.

3.3 Tõenäosussuhte meetod

Inglise keelsetes materjalides kutsutakse seda tüüpi maatriksit „*odds score matrix*". Seda maatriksit võib teisiti nimetada ka suhteliste sageduste ja eeldatava sageduse suhte maatriksiks. Absoluutsete sageduste maatriksit saab teisendada oodatud tõenäosussuhteks kasutades vaadeldud tõenäosusi. Olgu f_{bi} tõenäosus, millega see sümbol b esineb maatriksi veerus i ja olgu p_b oodatud sümboli b (tausta)sagedus suvalistes sekventsides, mida saab hinnata üle kõigi esinemiste selles suures DNA sekventside andmebaasis (sageli võetakse ka p_b väärtuseks 1 jagatud tähestiku suurusega, mis näiteks DNA puhul on 4.). Olgu n_{bi} summa, mis saadakse, kui loendatakse kokku sümboli b esinemised antud maatriksi veerus i ja olgu N kõigi (motiivide) esinemiste arv. Siis n_{bi}/N on f_{bi} hinnang ja sümboli b tõenäosussuhe esinemiseks veerus i on hinnatav järgmiselt:

$$x_{bi} = \frac{n_{b,i} \diagup N}{p_b} \qquad \textbf{(3.3)}$$

Tõenäosussuhe on lihtne ja see maksimeerib jälgitavate sümbolite valitavust, kuid sellel on ka tõsised puudused. Tõenäosussuhe ei luba teha säilitavaid asendusi. Seega bioloogilises kontekstis olev kaalumaatriks võib tulla tundetu distantsiliselt seotud perekonna liikmete suhtes [1]. Lisaks sisaldavad tavaliselt absoluutsete sageduste maatriksid palju 0 väärtusi, mis jäävad tõenäosussuhte maatriksiks teisendades nulliks. Nullväärtused võivad näidata seda, et sümbolid ei saa esineda selles positsioonis, mis on esitatud selle veeruga. Samuti on võimalus, et vaadeldud sümboli leidmiseks ei ole olnud piisavalt suur arv mustreid (motiive) selles hulgas. Antud juhul takistab see konverteerimast tõenäosussuhte maatriksit logaritmilisele

kujule, sest logaritm nullist on negatiivne lõpmatus. Selle vältimiseks kasutatavat pseudoloendi meetodit kirjeldame allpool.

	1	2	3	4	5	6
A	0.8/0.25=3.2	0/0.25=0	0.8/0.25=3.2	1/0.25=4	0/0.25=0	0/0.25=0
C	0/0.25=0	0.8/0.25=3.2	0.2/0.25=0.8	0/0.25=0	0/0.25=0	0.8/0.25=3.2
G	0/0.25=0	0.2/0.25=0.8	0/0.25=0	0/0.25=0	0.2/0.25=0.8	0.2/0.25=0.8
T	0.2/0.25=0.8	0/0.25=0	0/0.25=0	0/0.25=0	0.8/0.25=3.2	0/0.25=0

Tabel 3.3: *Odds score* ehk tõenäosussuhte maatriks

3.4 Logaritmiline tõenäosussuhte maatriks

Kui tõenäosussuhe on defineeritud nukleotiidi esinemissagedusega antud veerus jagatud selle nukleotiidi tausta sagedusega, siis saame nende korrutist transformeerida summaks kasutades logaritmi tõenäosussuhtest. Logaritmi kasutakse sageli sellisteks korrutise muundamiseks summaks. Sageli kasutatakse seda ka selleks, et leevendada liigvähestest motiividest saadud maatriksit, kus mõned väärtused on väga väikesed st tõenäosussuhe võib tulla sellel juhul väga suur.

Näiteks olgu meil antud sekvents $s = s_1, s_2, \ldots, s_n$, suhtelise sageduse maatriks f_{bi} ning p_b aluse b taustasagedus, siis logaritmilist tõenäosussuhet saab defineerida järgnevalt [9]:

$$\log_2 LikelihoodRatio(s) = \log_2 \prod_{j=1}^{n} \frac{f_{s_j,i}}{p_b} = \sum_{j=1}^{n} \log_2 \frac{f_{s_j,i}}{p_b} \qquad \textbf{(3.4)}$$

Selline meetrika on mõnikord tuntud kui logaritmiline tõenäosussuhe või *lod* number. Tuleb täheldada, et kui $f_{bi} = 0$, siis suhe on lõpmatu, mis on halb väljaarvatud juhul kui me oleme absoluutselt kindlad, et maatriks ei sisalda iial seda sümbolit b sellel positsioonil i. Et sellist juhtu vältida,

lisatakse tavaliselt kindlustuseks kaalumaatriksi mudelile nullist erineva väärtusega tõenäosus. Kaalumaatriks on enamasti konstrueeritud reaalsetest andmetest ja seetõttu nullväärtusega kaalud vahetevahel juhuslikult ilmnevad limiteeritud andmehulga pärast. Sageli saab lahendada selle probleemi lisades antud mudelile nn pseudoarve.

	1	2	3	4	5	6
A	$\log_2 3.2 = 1.68$	0	1.68	2	0	0
C	0	1.68	-0.32	0	0	1.68
G	0	-0.32	0	0	-0.32	-0.32
T	$\log_2 0.8 = -0.32$	0	0	0	1.68	0

Tabel 3.4: Logaritmiline tõenäosussuhte maatriks ilma pseudoarvude lisamiseta.

3.5 Pseudoloenduste (ingl. k. „*pseudocounts*") meetodid

Peamine probleem, millega eelnevalt vaadeldud numbrilised meetodi silmitsi seisavad, on see, kuidas konverteerida vaadeldud sümbolite loendustevektorit sagedus- või skooride vektoriks. Vaadeldud loenduste hulk on lõplik ja tavaliselt alati sisaldavad null loendusi ühe või rohkemate sümbolite jaoks. Nagu eespool tähendatud, null sagedused on ebasoovitud sekventside analüüsis, sest nad võivad välistada tõese, kuigi mitte tavalise antud perekonna liikme. Seetõttu peab tutvustama metodoloogiat vaadeldud loendite hulga suurendamiseks ja hindamaks õige populatsiooni sagedusi. Põhiliselt on probleem sageduste hindamisega ja domineeriv meetod on pseudoloendite või tehisloendite lisamine vaadeldud loenditele.

Kuigi Bayesi teoreem on aluseks pseudoloendite lähenemise, jätab ta lahti küsimuse, kui palju lisada neid pseudoloendeid, ehk siis kui paljul kaalu lisada tähtsamale informatsioonile. See küsimus on tekitanud palju erinevaid meetodeid kombineerimaks pseudoloendeid vaadeldud loenditega.

Olemasolevad meetodid baseeruvalt laialdaselt intuitiivsetel või empiirilistele alustele, mille tõttu võib väita, et optimaalset meetodit ei eksisteeri.

Alternatiivne viis konstrueerida logaritmilist tõenäosussuhte kaalumaatriksit on lisada hüpoteetilisi mustreid valimisse. Iga veeru jaoks hõlmab see loendustele lisamist „pseudoloendit", mis baseerub mingisuguselt usul, näiteks tegelikul mittelõplikult vaadeldul sümbolite jaotusel selles veerus. Olgu t_{bi} pseudoloendite arv sümboli b jaoks veerus i ja T_i olgu kogu pseudoloendite arv veerus i, n_{bi} summa (esinemiste arv), mis saadakse kui loendatakse kokku sümboli b esinemised antud maatriksi veerus i ja olgu N kõigi motiivide arv. Nii $\frac{n_{b,i}}{N}$ ja $\frac{t_{b,i}}{T_i}$ on f_{bi} hinnang ja kaalutud keskmise hinnang on:

$$f_{bi} = \frac{N}{N+T_i} * \frac{n_{bi}}{N} + \frac{T_i}{N+T_i} * \frac{t_{bi}}{T_i} = \frac{n_{bi}+t_{bi}}{N+T_i} \quad \textbf{(3.5)}$$

N ja T_i suhteline suurus peegeldab seda, kui tugevasti iga hinnang panustab. Kui N on suurem millegi poolest T_i, siis vaadeldud loendid domineerivad, kuna pseudoloendid domineerivad kui vastupidine on tõsi. Pseudoloendid peaksid olema konstrueeritud kindlustamaks, et valem 3.5 koonduks $\frac{n_{b,i}}{N}$ nii, et leiaks rohkem perekonna liikmeid [2].

Mitmed autorid on valinud T_i selliselt, et see oleks mingisugune funktsioon mustrite arvust N kaalumaatriksis, näiteks $T_i = \sqrt{N}$ [7]. See valik ei ole küll ideaalne, kuid N-i suuremate väärtuste puhul on see rahuldav valik [4].

Lihtsaim viis valimaks pseudoloendite koguarvu veerus on see, kui T_i on konstante kogu kaalumaatriksis. Konstant peab olema piisavalt suur domineerimaks loendusi väheste sekventside korral ja peaks olema kindlaks määratud empiiriliselt. Kui me lubame T_i saamaks väga suureks võrreldes N, siis see kahandab puhta t_{bi} pseudoloendite hinnangut.

Üks kõige lihtsamaid ja seega ka kõige levinumaid pseudoloendusi on kui pseudoloenduste koguarvuks T_i võtame N + 1, ning igale maatriksi kaalule lisatakse 1/ N või antud sümboli taustasagedus p_b. Valem tuleb siis järgmine:

$$w_{bi} = \log(\frac{(f_{bi} + p_b)/(N+1)}{p_b}) \approx \log(\frac{f_{ib}}{p_b}) \qquad \textbf{(3.6)}$$

Seda lahendust kasutan ka enda programmis.

3.6 Informatsiooni sisalduse maatriks

Olulisel kohal kaalumaatriksite kirjeldamisel on ka informatsioonisisalduse järgi loodud maatriksid. Erinevate regulatoorsete süsteemide seondumiskohtade võrdlemisel, on töötatud välja informatsiooni sisalduse ning selle sõltuvus seondumiskohtade sagedusest genoomis [11]. Informatsiooni sisaldust saidi igal positsioonil võib esitada nii:

$$I_i = \log_2 J + \sum_{b=A}^{T} f_{b,i} \log_2 f_{b,i} \qquad \textbf{(3.7)}$$

kus J on tähestiku suurus (DNA puhul 4), i on positsioon saidis, b viitab võimalikele alustele ning $f_{b,j}$ on iga sümboli leitud sagedus positsioonil i. I_i väärtus on 0, kui kõikide aluste esinemise tõenäosus on 25% ja 2 bitti juhul,

kui positsioon on täielikult konserveerunud ehk antud positsioonis esineb vaid üks sümbol neljast.

Statistilise mehhaanika teooriat kasutades on näidatud, et aluste sageduste logaritmid peaksid olema proportsionaalsed nende aluste seostumisenergia panusega [1]. See teooria toetab informatsiooni sisalduse analüüsi ja soovitab, et informatsiooni sisaldus on seotud seondumiskohtade hulga keskmise seostumisenergiaga. Näiteks pärmi puhul valem 3.7 valem viitab positiivsele informatsiooni sisaldusele ja seega spetsiifilisele seostumisenergiale igal juhuslikul seondumiskohtade hulgal. Parandatud valem, mis võtab arvesse ka pärmis valitsevat nukleotiidide suhet, on järgmine [13]:

$$I_{seq(i)} = \sum_{b=A}^{T} f_{b,i} \log_2 \frac{f_{b,i}}{p_b} \qquad \textbf{(3.8)}$$

kus p_b on aluse *b* sagedus kogu genoomis. Valem 3.7 on valemi 3.8 erijuht, kus p_b on kõikide *b* jaoks 0.25. Sellel normaliseeritud logaritmilisel tõenäosussuhtel on palju teisi nimesid. Informatsiooni teooriast motiveerituna on teda kutsutud *Kullback-Leiberi* informatsiooniks või relatiivseks entroopiaks. Kui tuletada see suure hälbe printsiibist, siis on teda kutsutud „*large-derivation rate function*".

Ka on I_{seq} seotud termodünaamikaga. Eriti tavalise valgu seotud DNA sekventside informatsiooni sisaldusega, mis on ühenduses valgu-DNA vastastikmõju termodünaamikaga. I_{seq} mõõdab proteiini siduva funktsionaalse DNA saidi keskmise ΔG ja proteiini siduva suvalise DNA sekventsi ΔG vahelist suhet [1]. Seega I_{seq} on DNA funktsionaalse sekventsi ja DNA suvalise sekventsi vahelise siduvuse diskriminatsiooni mõõt. *'seq'* alamlaiend näitab seda, et valem 3.8 on informatsiooni sisaldus tuletatud

sekventsi joonduse statistilistest omadustest. Fieldsi kirjutises [3] arutatakse informatsiooni sisaldusele väga lähedasest I_{spec} , mis on tuletatud termodünaamika kaudu [6].

Valemil 3.8 on palju erinevaid omadusi, mis rahuldavad intuitiivseid joonduse informatsiooni sisalduse ideid. Valem 3.8 on kauguse mõõt jaotuse keskpunktist, kus iga $f_{i,j} = p_{i,j}$. Kui $f_{i,j} = p_{i,j}$, siis kaugus on minimaalne ja võrdub nulliga. Kaugus on maksimaalne kui esineb eksklusiivselt kõige vähem oodatud täht, st $f_{m,j} = 1$ ja $p_m \leq p_i$ kõigi i-de korral. Schneider täheldas [11], et e^{-Iseq} on ligilähedaselt võrdne sagedusega, millega esinevad DNA seondumis-saidid pärmi genoomis. Hertz ja Stormo [5] tulid välja veel täpsema antud suhte kirjeldusega: e^{-Iseq} on ülemine limiit statistilisele ootusele mis sagedusega esinevad joonduse sõnad juhuslikus sekventsis.

Valem 3.9 näitab, kuidas leida parimat maatriksit, kui on teada kõrge afiinsusega saidid, kuid pole teada täpset seostumisafiinsust. Eeldame, et teatakse ka organismi täielikku genoomijärjestust, st kust valk ja seondumiskohad on ise pärit. Lisamiseeldusest lähtudes, et iga positsioon panustab sõltumatult kogu seostumisenergiasse, on meil maatriks *H(b, i)*, mis sisaldab seostumisenergia panuseid oma elementidena. Iga üksiku järjestuse S_α (alamsõne) kogu seostumisenergia on antud $H(b, i) \bullet S_\alpha$ poolt. Tõenäosus, et valk seonduks saidile järjestusega S_α, arvestades kõiki võimalikke seondumiskohtade kogu genoomis, on kirjeldatud valemiga:

$$P(\, S_\alpha \; on \; seostunud \,) = \frac{e^{-H(b,i)\bullet S_\alpha}}{Z} \qquad \textbf{(3.9)}$$

kus Z on alamfunktsioon, üle kõigi genoomi seondumiskohtade seostumisafiinsuste summa. Teades, et meie seondumiskohad on kõrge seostumistõenäosusega, on järgmine loogiline samm maatriksi leidmine, mis maksimeerib kõikidele seondumiskohtadele seondumise tõenäosust. Kuna

eeldame, et genoom on põhiolemuselt juhuslik, siis me saame arvutada *Z*-i analüütiliselt [5]. Genoomid ei ole juhuslikud järjestused, kuid eeldus on kehtiv kui lühikesed alamjärjestused, seondumiskohtade pikkustega, esinevad genoomi aluste eeldatava sagedusega. Sellisel juhul juhuslikkuse eeldus on kehtiv. Antud eeldusele toetudes võib näidata, et maatriksi *H(b, i)* elemendid, mis maksimeerivad seostumise tõenäosust hulgale funktsionaalsetele seondumiskohtadele, on lihtsalt

$$H(b,i) = -\ln \frac{f_{b,i}}{p_b}. \qquad \textbf{(3.10)}$$

Seega, kui on hulk kindla faktori tuntud seondumiskohti, siis $-\ln \frac{f_{b,i}}{p_b}$ on maksimaalne tõenäosuse hinnang seostumisenergia panusele iga aluse kohta igas positsioonis ja I_{seq} on kõigi tuntud seondumiskohtade keskmine seostumisenergia [14].

	1	2	3	4	5	6
A	1.21	-2.58	1.21	1.51	-2.58	-2.58
C	-2.58	1.95	0.13	-2.58	-2.58	1.95
G	-2.58	0.13	-2.58	-2.58	0.13	0.13
T	-0.54	-2.58	-2.58	-2.58	1.17	-2.58

Tabel 3.5: Informatsiooni sisalduse maatriks

4 Maatriksi sobitamine

Peatükk vaadeldakse lühidalt kuidas sobitada olemasolevat kaalumaatriksit tekstile. Samuti tuuakse välja sobitamise jõumeetod ning võrreldakse seda nelja edasiarendusega.

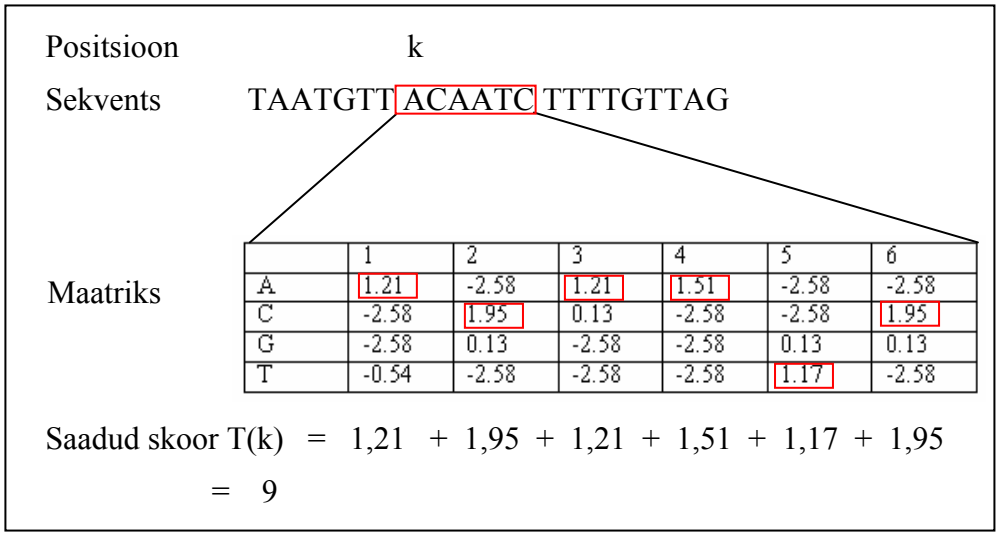
4.1 Maatriksiga otsimine

Skoorimaatriks esitab nukleiinhappe või valgu segmendi seotust sekventside perekonnas. Teisiti öeldes, sellist tüüpi konstruktsioon sisaldab erinevaid tüüpe eelnevalt vaadeldud kaalumaatrikseid. Skoorimaatriks *S* esindab tühikuteta sekventsi perekonna lokaalset joondust. Joondus koosneb mitmetest külgnevatest positsioonidest, iga positsioon on esindatud veeruga skoori maatriksis. Vaheldumisi iga veerg *j* koosneb skooride S_j *(b)* vektoritest, üks skoor iga võimaliku nukleotiidi *b* jaoks [18].

Skoori maatriksit saab kasutada sekventsi analüüsis liigutades maatriksit üle sekventsi ja arvutades (alam)segmentide skoori. Iga segmendi skoor on lihtsalt vastava maatriksi tabelielementide summa (sagedusmaatriksi ja tõenäosussuhte maatriksi puhul korrutis), millega iga sümboli skoor vastab maatriksi veerus. Kui sekvents koosneb sümbolitest $a_1, \ldots, a_L$ ja segment pikkusega *J* algab positsioonis *k*, $(1 \le k \le L - J + 1)$, siis segmendi skoor on järgmine (vaata ka Joonis 4.1):

$$T = \sum_{j=1}^{J} S_j(a_{k+j-1}) \qquad \textbf{(4.1)}$$

Intuitiivselt, kõrgem segmendi skoor näitab kõrgemat tõenäosust, et sekvents ühtib antud skoori maatriksiga.



Joonis 4.1: Sekventsil positsiooni k skooride arvutamine kasutades kaalumaatriksit

4.1.1 Jõumeetod

Jõumeetodi algoritm on kirjeldatud Joonis 4.2. Kuna sisuliselt on tegemist täpse otsimise meetodiga, siis tema ajaline keerukus on *O(mn)*, sõltudes otseselt teksti ja mustri pikkusest.

Algoritm: Kaalumaatriksiga otsimine - jõumeetod
Sisend: Kaalumaatriks ($W_{i,j}$), $i = 1..|\Sigma|$, $j = 1..m$,
string S, künnisväärtus K
Väljund: Kõik $W_{i,j}$ esinemised *S*-is mis ületavad künnist *K*

```
1. for p = 1 .. |S|-m+1
2.     sum = 0 ;
3.     for j = 1 .. m
4.         sum  +=  W[ S[p+j-1] ][ j ]
5.     if sum ≥ K then raporteeri, et positsioonis p oli skoor sum
```

Joonis 4.2: Kaalumaatriksi tekstile sobitamise jõumeetod

Suured on tekstid, millest otsitakse - näiteks inimese genoom on umbes 3,2 Gb, pärmil (mida siiani kõige rohkem uuritud) umbes 12 Mb. Järjest enam kasvavad ka transkriptsiooni faktorite andmebaasid (mustriandmebaasid). Seega on ka andmemahud äärmiselt mahukad. Kuna sageli tehakse ühtede andmete peal mitmeid erinevaid teste, et katsetada mitmesuguseid hüpoteese, siis on äärmiselt oluline arvutuskiirus. Sageli ei ole vaja hoida ja sorteerida suurt hulka skoore, mis suure tõenäosusega ei oma tegelikku tähendust. Sellepärast sai arendatud 4 jõumeetodi edasiarendust. Järgmises paragrahvis vaatlemegi nende algoritme, eripärasusi ning võrdleme nende ja jõumeetodi vahelisi kiiruse erinevusi erinevates situatsioonides.

4.1.2 Eeltöötlus

Ennem kui asuda antud algoritmide kallale, tuleks vaadata eeltöötlust, mida nad vajavad. Antud algoritmide kiirused sõltuvad otseselt kasutaja

etteantud lävest. Lävi defineerib kasutaja poolt ära, mida ta peab tähenduslikuks skooriks ja mida mitte (läve valik sõltub veel ka meetodist, millega on kaalumaatriks leitud ning ka mustri pikkusest). Seega võime oletada, et leides mustritest saadud kaalumaatriksi veergude maksimaalsed väärtused (st igas veerust üks sümbol, mis panustaks kogu skoori kõige rohkem) ning arvutades sellest veergude maksimaalsete väärtuste vektorist uue vektori, mille elemendi väärtuseks vastavas veeru positsioonis oleks temast järgnevate elementide summa, saame efektiivsemalt antud probleemi lahendada. Seega saame iga maatriksi veeru positsiooni jaoks teada temast järgnevate positsioonide maksimaalse võimaliku summa.

Meetod: Veergude maksimumid
Sisend: Kaalumaatriks ($W_{i,j}$), $i = 1..|\Sigma|$, $j = 1..m$,
maatriksi veergude arv j ja ridade arv i
Väljund: Tagastab kaalumaatriksi $W_{i,j}$ veergude maksimaalsest
väärtustest koostatud vektori, mille pikkuseks on j

```
1.  M [j] -- vektor, kuhu salvestaks igale positsioonile kaalumaatriksi
          -- vastava veeru elementide maksimaalne väärtus
2.  for a = 0 .. j
3.      for b = 0 .. i
4.          if M[b] < W[b][a]
5.              M[b] = W[b][a];
6.  return M
```

Joonis 4.3: Veergude maksimaalsete väärtuste leidmise meetod

```
Meetod: Järgnevate elementide summa
Sisend: Massiiv M, mis on saadud kaalumaatriksi veergude
        maksimaalsetest väärtustest, ja massiivi pikkus i
Väljund: Tagastab massiivi M elementide summad
         positsioonist p kuni massiivi lõpuni
1. sum_from_pos[ i ] -- vektor, kuhu salvestaks
                     -- igale positsioonile sisendmassiivi temast
                     -- järgnevate elementide summa
2. for a = 1 .. i
3.     for j = a + 1 .. i
4.         sum_from_pos [a] + = M [ j ];
5. return sum_from_pos
```

Joonis 4.4: Järgnevate elementide summa leidmise meetod

Veergude maksimumi leidmise ajaline keerukus sõltub kaalumaatriksi veergude ja ridade arvust (ehk siis vastavalt mustri pikkusest ja tähestiku suurusest) ning on vastavalt *O (mn)*. Teiseks kirjeldatud eeltöötluseks vajaliku järgnevate elementide summa leidmise meetodi ajaline keerukus on *O(n·log n)*. Seda saaks veelgi parandada, kui alustada liitmist viimasest elemendist ettepoole, sel juhul oleks vaja meil teha ainult *(n-1)* tehet. Kuid kuna mustri pikkus (tavaliselt 5-30) ja tähestiku suurus (DNA puhul 4, aminohapete puhul 20) on võrreldes teksti suurusega imeväikesed ning eeltöötlust tehakse ainult üks kord, siis jõumeetodi edasiarenduste ajalisse keerukusse ei lisa need eriti märkimisväärset panust.

4.1.3 Ettevaatav skoorimine (1a)

Vaatleme nüüd esimest jõualgoritmi edasiarendust. Ettevaatava skoorimise (vt Joonis 4.5) erinevuseks võrreldes jõualgoritmiga on see, et

kontrollitakse vastavalt etteantud veerust (st antakse ette veeru number, mitmendast veerust) alates iga skoori liitmise järel, kas antud alamskoor pluss järgmiste veergude maksimaalsete skooride kogusumma ületab läve. Kui see jääb allapoole läve, siis antud segmendi skoori arvutamine katkestatakse. Kontrollimist võib alustada juba ka esimesest veerust alates. Sel juhul võib esineda olukord kui lävi on valitud liiga kõrge ja kaalumaatriksi skoorid igas veerus on ühtlaselt jaotunud, et alamsegmendi skoori arvutamise tsükkel katkestatakse juba peale esimest veergu.

Algoritm: Kaalumaatriksi otsimine – ettevaatav skroorimine (1a)

Sisend: Kaalumaatriks ($W_{i,j}$), $i = 1..|\Sigma|$, $j = 1..m$,
string S, künnisväärtus K, positsioon t, millal hakata testima kaalust ülemineku võimalust

Väljund: Kõik $W_{i,j}$ esinemised S-is mis ületavad künnist K

Eeltöötlus: Arvutame kaalumaatriksi veergude maksimaalsed igas veeru positsioonis ning sellest tulemustest uue vektori, mille elementi väärtuseks on maksimaalsete vektori elementide summa sellest positsioonist kuni viimase positsioonini.

```
 1. sum_from_pos [ j ] -- eeltöötlusel saadav vektor
 2. for p = 1 .. |S|-m+1
 3.     sum = 0 ;
 4.     for j = 1 .. t
 5.         sum  +=  W[ S[p+j-1] ][ j ]
 6.     for t .. m
 7.         if sum + sum_from_pos [ t ] < K then
 8.             break
 9.         sum  +=  W[ S[p+t-1] ][ j ]
11.     if sum ≥ K then raporteeri, et positsioonis p oli skoor sum
```

Joonis 4.5: Ettevaatav skoorimine (1a)

4.1.4 Ettevaatav skoorimine ühe lävetestiga (1b)

Teine vaadeldav meetod on väga sarnane esimesele. Ainuke erinevus esimesega on see, et läveületamise kontroll tehakse ainult üks kord ning selle hindamise järel tehakse otsus, kas on üldse võimalik läve ületada või mitte, st. kas on mõtet edasi arvutada või katkestada arvutamine. Selline lähenemine võib anda võib anda ajavõitu eelmise vaadeldud algoritmiga,

kuna tehakse vähem läve ületamist kontrollivaid võrdlusi. Probleemseks kohaks on siin läve ületamise testi õigesse veerupositsiooni paigutamisel.

Algoritm: Ettevaatav skoorimine ühe lävetestiga (1b)
Sisend: Kaalumaatriks ($W_{i,j}$), $i = 1..|\Sigma|$, $j = 1..m$,
string S, künnisväärtus K, positsioon t, millal testitakse kaalust ülemineku võimalust
Väljund: Kõik $W_{i,j}$ esinemised S-is mis ületavad künnist K
Eeltöötlus: Arvutame kaalumaatriksi veergude maksimaalsed igas veeru positsioonis ning sellest tulemustest uue vektori, mille elementi väärtuseks on maksimaalsete vektori elementide summa sellest positsioonist kuni viimase positsioonini.

```
 1. sum_from_pos [ j ] -- eeltöötlusel saadav vektor
 2. for p = 1 .. |S|-m+1
 3.     sum = 0 ;
 4.     for j = 1 .. t
 5.         sum  +=  W[ S[p+j-1] ][ j ]
 6.     if sum + sum_from_pos [ t ] < K then
 7.         break
 8.     for t .. m
 8.         sum  +=  W[ S[p+t-1] ][ j ]
10.     if sum > K then raporteeri, et positsioonis p oli skoor sum
```

Joonis 4.6: Ettevaatav skoorimine ühe testiga (1b)

4.1.5 Sorteeritud järjekorraga ettevaatav skoorimine (2a)

Järgnevalt vaatleme meetodit, mis on väga sarnane ettevaatavale skoorimisele. Selles üritatakse parandada ettevaatava skoorimise meetodit. Lisaks eelnevatele eeltöötlustele tehakse siin veel üks. Selleks sorteeritakse veergude maksimaalsete vektor kahanevalt säilitades elementide eelneva

järjekorra. Tänu sellele operatsioonile saame arvutamiseks uue järjekorra, mis peaks andma seda, et need veerud, mis on rohkem konserveerunud, lisatakse segmendi skoori esimesena, järelikult peaks tõusma ka tõenäosus, et tuleb teha vähem arvutusi. Kõik muu on võrreldes ettevaatava skoorimise meetodiga sama. Selle algoritmi kirjeldus on joonisel 4.7.

Algoritm: Sorteeritud järjekorraga ettevaatav skoorimine (2a)

Sisend: Kaalumaatriks ($W_{i,j}$), $i = 1..|\Sigma|$, $j = 1..m$,
string S, künnisväärtus K, positsioon t, millal hakata testima kaalust ülemineku võimalust

Väljund: Kõik $W_{i,j}$ esinemised S-is mis ületavad künnist K

Eeltöötlus: Arvutame kaalumaatriksi veergude maksimaalsed igas veeru positsioonis, sorteerime need kahanevalt (jättes meelde esialgse järjekorra) ning sellest tulemusest arvutame uue vektori, mille elementi väärtuseks on maksimaalne järelejäänud summa järgmises positsioonist kuni veergude lõpuni.

```
 1. sum_from_pos [ j ] -- eeltöötlusel saadav vektor
 2. jrk [ j ] -- sorteeritud veergude maksimaalsete vektori järjekord enne
              -- sorteerimist
 3. for p = 1 .. |S|-m+1
 4.     sum = 0 ;
 5.     for j = 1 .. t
 6.         sum  +=  W[ S[p + jrk [ j ] - 1] ][  jrk [ j ] ]
 7.         for t .. m
 8.             if sum + sum_from_pos [ t ] < K then
 9.                 break
10.             sum  +=  W[ S[p + jrk [ t ] - 1] ][  jrk [ t ] ]
11.         if sum ≥ K then raporteeri, et positsioonis p oli skoor sum
```

Joonis 4.7: Jõumeetodid kolmanda edasiarenduse algoritm (2a), kus kasutatakse eeltöötlusena arvutatud uut järjekorda ning läve ületamist kontrollitakse alates ettemääratud veerust iga veeru järelt.

4.1.6 Sorteeritud järjekorraga ettevaatav skoorimine ühe lävetestiga (2b)

Selle algoritmiga üritame eelmises paragrahvis vaadeldud lisa eeltöötlust rakendada eelnevalt vaadeldud ettevaatava skoorimise algoritmile, mis kasutab ainult ühte läve ületamise testi.

Algoritm: Sorteeritud järjekorraga ettevaatav skoorimine ühe lävetestiga
Sisend: Kaalumaatriks ($W_{i,j}$), $i = 1..|\Sigma|, j = 1..m$, string S, künnisväärtus K, positsioon t, millal testitakse kaalust ülemineku võimalust
Väljund: Kõik $W_{i,j}$ esinemised S-is mis ületavad künnist K
Eeltöötlus: Arvutame kaalumaatriksi veergude maksimaalsed igas veeru positsioonis, sorteerime need kahanevalt (jättes meelde esialgse järjekorra) ning sellest tulemusest arvutame uue vektori, mille elementi väärtuseks on maksimaalne järelejäänud summa järgmises positsioonist kuni veergude lõpuni.

```
 1. sum_from_pos [ j ] -- eeltöötlusel saadav vektor
 2. jrk [ j ] -- sorteeritud veergude maksimaalsete vektori järjekord enne
              -- sorteerimist
 3. for p = 1 .. |S|-m+1
 4.     sum = 0 ;
 5.     for j = 1 .. t
 6.         sum  +=  W[ S[p+j-1] ][ j ]
 7.     if sum + sum_from_pos [ t ] < K then
 8.         break
 9.     for t .. m
10.         sum  +=  W[ S[p+t-1] ][ j ]
```

Joonis 4.8: Sorteeritud järjekorraga ettevaatav skoorimine ühe läve testiga (2b)

4.2 Eksperiment

Et teada saada, milline algoritm siis ikkagi on erinevatest situatsioonides kõige parem, tegin mitmeid teste erinevatest mustritest saadud kaalumaatriksiga. Testsekventsid sain *Expression Profiler* (EP) [15] andmebaasist ning selleks oli *Saccharomyces cerevisiae* (pärm) kõik ülesvoolu olevad sekventsid (6423 sekventsi) pikkusega 600bp (lisaks on algus koodoni tähis _ATG_, mille eemaldasin). See fail on saadaval [10], samuti ka selle tööga kaasasoleval CD-l.

Mustrid, millest saada kaalumaatrikseid, genereerisin ise, kasutades *EP* koosseisus olevat tööriista *PATMATCH* [17]. Selleks valisin juba eelnevalt testsekventsidest leitud 2 kõige enam esinenud mustrit G.GATGAG.T ja TG.AAA.TTT [16] ning otsisin nende kõiki ligikaudseid esinemisi, et saada kaalumaatriksile sisendiks olevat mustrite hulka. Kuna antud mustrid on suhteliselt lühikesed ja konserveerunud, siis otsustasin lubada sisse ühe mittetabamuse suvalisel positsioonil ning lisasin algusse ja/või lõppu kõiki sümboleid antud positsioonil lubava sümboli ‚**.**'. Seda tegin selleks, et leida sellest hulgast erineva pikkusega ning erineva konserveerituse asukohaga mustreid.

Antud töös vaatlen kahe erineva pikkusega kaalumaatriksit: 10 ja 25 ja pikema mustri puhul genereerisin mitu profiili, kus konserveerunud (tegelikult otsitav muster) muster asetses, kas kaalumaatriksi esimestes, keskmistes või viimastes veergudes (vaata jooniste 4.9 – 4.12 sekventsilogosid).

Oma testimist teostasin arvutil, millel oli Intel Pentium 4, 2,6 GHz protsessor ja 512 Mb DDR mälu ning operatsioonisüsteemiks Linux Fedora Core 1 (gcc versiooniga 3.3.2).

4.2.1 Test lühikese ja konserveerunud motiiviga

Nüüd natuke ka tulemustest. Mis selgelt kõikide testimiste puhul välja tuli, oli see, et edasiarendused hakkavad ennast ära tasuma vaid pikkade mustrite korral, mis ei ole täielikult konserveerunud, vaid on osaliselt mingis osas konserveerunud. Samuti sõltub nende kiirus ka sellest millise lävega otsitakse.

Kõigepealt vaatleksin eelnevalt mainitud kahest motiivist (lubades ühte mittesobivust suvalises positsioonis) saadud profiilidest koostatud kaalumaatriksiga teostatud teste erinevate lävedega. Kuna kaalumaatriksid olid koostatud kasutades informatsiooni sisaldust arvutaval meetodil, siis kaalumaatriks sisaldab ka negatiivseid väärtusi ja seega võib segmendi skoor tulla ka negatiivne. Kuid tavaliselt ei paku negatiivsed skoorid huvi ning selle võtsingi erinevateks lävedeks läved alates 1-st kuni kaalumaatriksi 100%-list sobivust näitava skoorini. Kuna edasiarendused sõltuvad sellest, kuna hakatakse või kuna tehakse test skoori üle läve mineku võimalikkusest, siis vaatlesingi seda aspekti, lastes igal meetodil erineva lävega teha läbi testi, kus alustati kontrollimist või kontrolliti (vastavalt algoritmile) läve võimalikku ületamist antud alamsekventsi poolt. Ühe joonise peale ei oleks kõik need testi tulemused mahtunud, seega tegin joonised (vt jooniste 4.9 – 4.12), kus näidatakse kõiki viit meetodit ühe lävega ning kus alustatakse testimise tegemist esimesest ja lõpetatakse viimase positsiooniga.

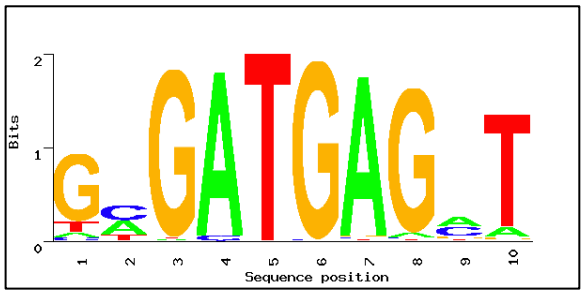
Lühikese ja peaaegu täielikult konserveerunud motiivi ja madala läve korral tuli välja see, et need algoritmid, mis kasutavad ainult ühte testi läve ületamise kontrolliks, ületavad jõumeetodi tehtavat aega u 1/5 võrra, kui testi tehakse esimestes positsioonides. Samamoodi ületavad kõik

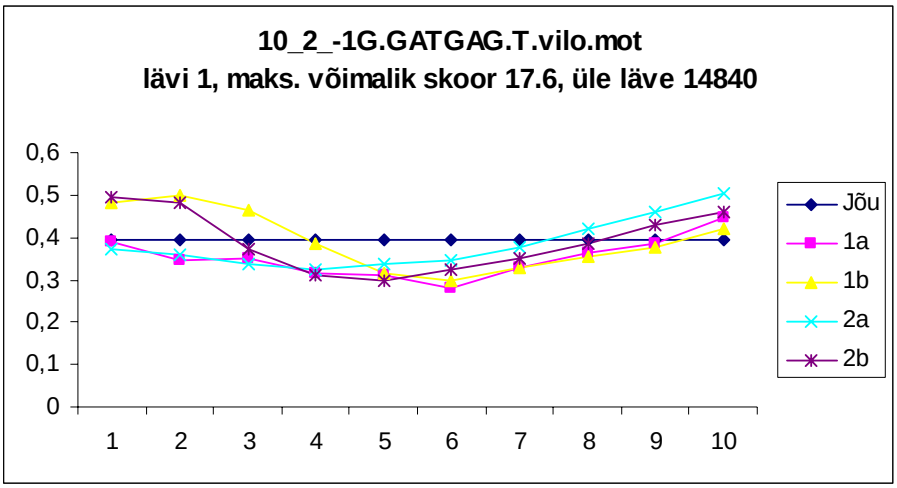
edasiarendused ajaliselt jõumeetodit, kui test jäetakse viimastele positsioonidele.

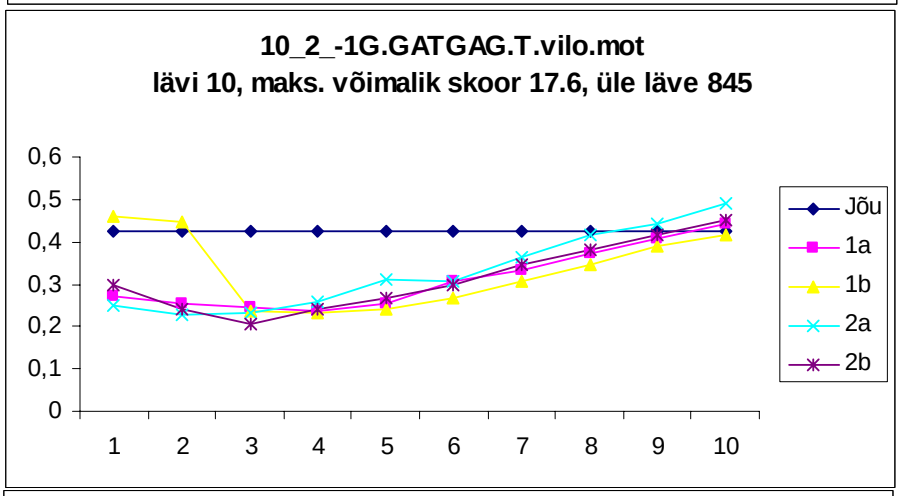
Mida kõrgemaks lävi kasvab, seda selgemalt tuleb esile see, et testimist läve ületamiseks (või ühte testi) tuleb teha juba esimestel veergudel. See annab võrreldes jõumeetodiga kahe kuni kolme kordse paremuse. Joonistelt tuleb selgelt välja ka see, et kõrge läve korral (90-100% maksimumist) käituvad kõik edasiarendused väga sarnaselt ning otseseid eelistusi ei saagi välja tuua. Kõrge läve korral on näha ka see, et testi viimastele veergudele jätmine kasvatab testi tegemise aega lineaarselt ning isegi viimastel positsioonidel tehtavat testi kasutavad meetodid ületavad jõumeetodi tehtavat aega.

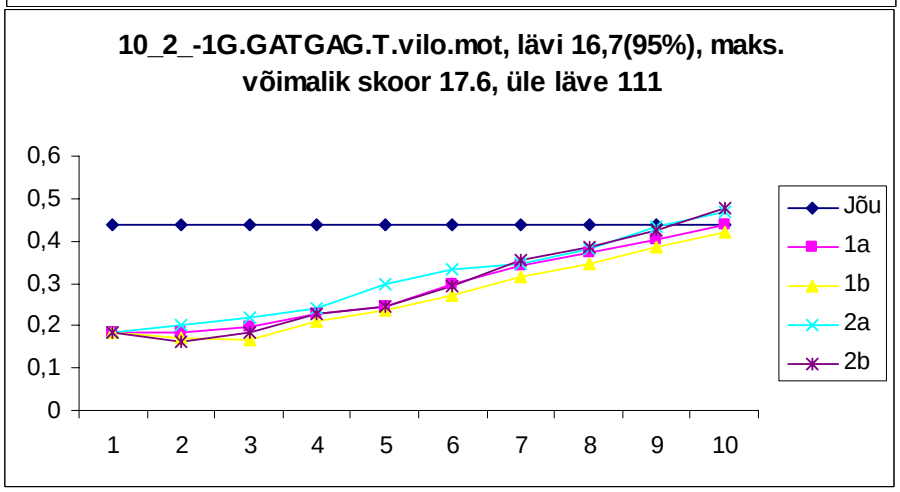
Meie vaadeldud kahel kaalumaatriksil on ühel esimestel veeru positsioonidel konserveeruvus väike. Seal on näha kohe teise ja neljanda meetodi erinevus: kuna neljas kasutab eelnevat sorteerimist, siis saab ta ka teada juba esimestel positsioonidel, kas antud alamskoor ületab või mitte läve.

Kokkuvõtlikult võib ütelda, et joonistelt 4.9 - 4.10 selgub, et madalate lävede korral ei tasu peaaegu täielikult konserveerunud maatriksi testi alustamine esimestel ja viimastel veergudel ära, sest meetodid ületavad või on ligilähedased jõumeetodiga. samuti on näha, et mida kõrgemaks muutub lävi, seda kasulikum on teha testi esimestel positsioonidel ning ei ole erilist vahet millise edasiarendust kasutada väljaarvatud ettevaatav ühe läve testiga skoorimine esimesel paaril positsioonil).

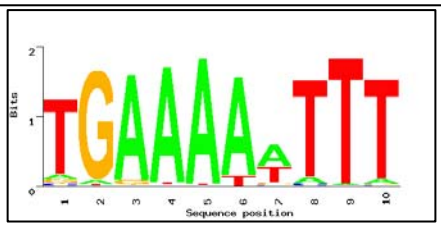


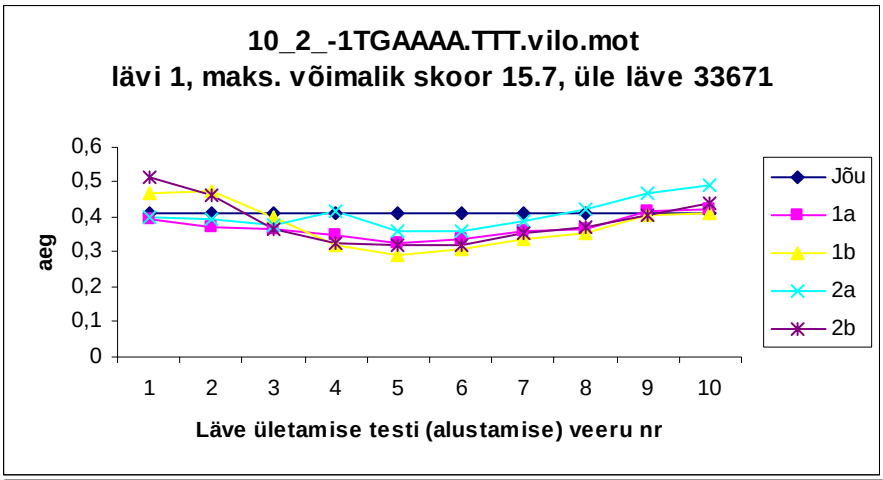


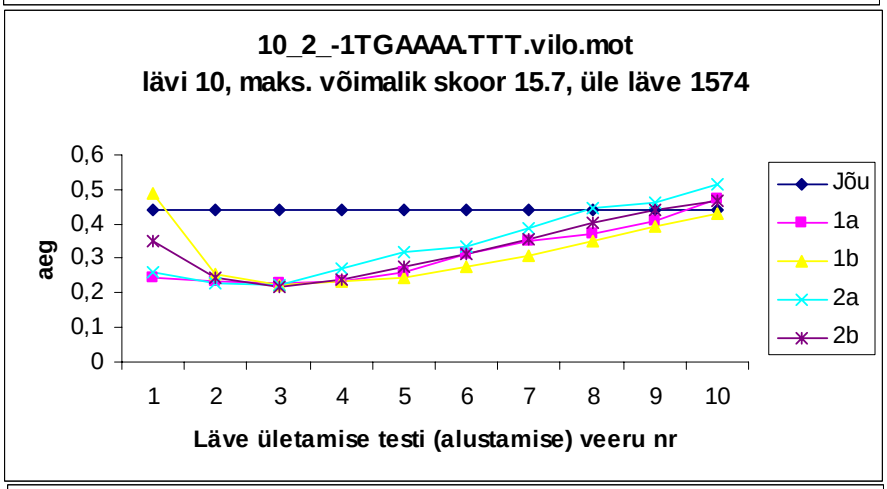


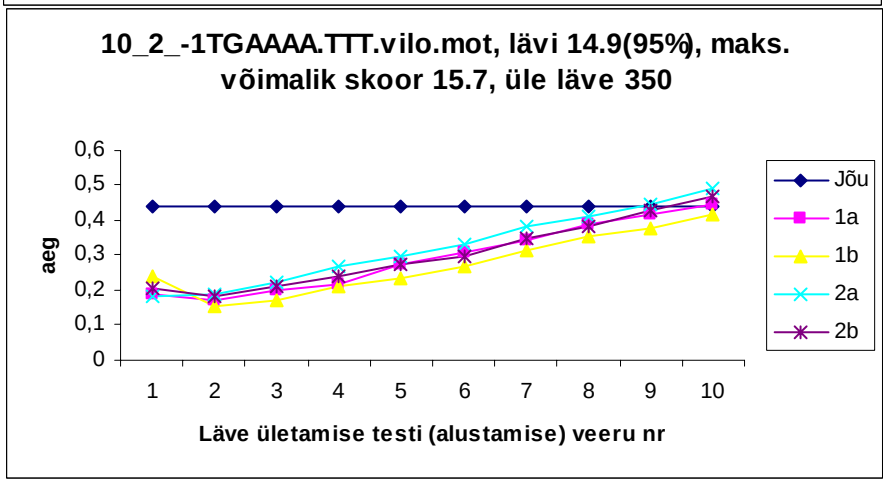


Joonis 4.9: Joonisel on tehtud kolm testi erinevate lävedega, et vaadata milline ülevalpool kirjeldatud meetoditest oleks kõige sobivam suhteliselt lühikese ja konserveerunud (10-st veerus 7) kaalumaatriksi korral.









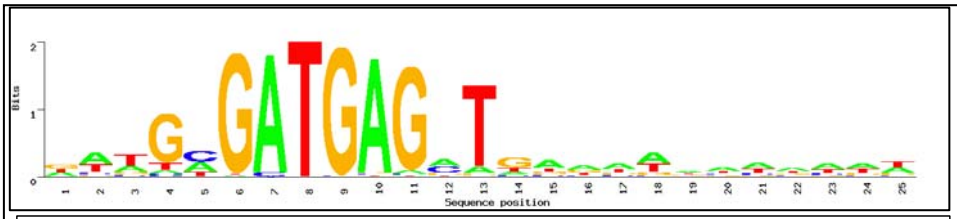
Joonis 4.10: Kui eelmisel joonisel oli kaalumaatriksiks valitud seitsme peaaegu täielikult konserveerunud veeruga maatriksi, siis siin joonisel on valitud maatriksiks ühtlasemalt konserveerunud maatriks (9 positsiooni 10st on ühtlastelt konserveerunud ning üks natukene vähem). Nende kaalumaatriksite maksimaalne võimalik kaal on väga sarnane, vastavalt 26,2 ja 26,6. Antud joonise toetab täielikult eelnevalt kirjeldatud väiteid.

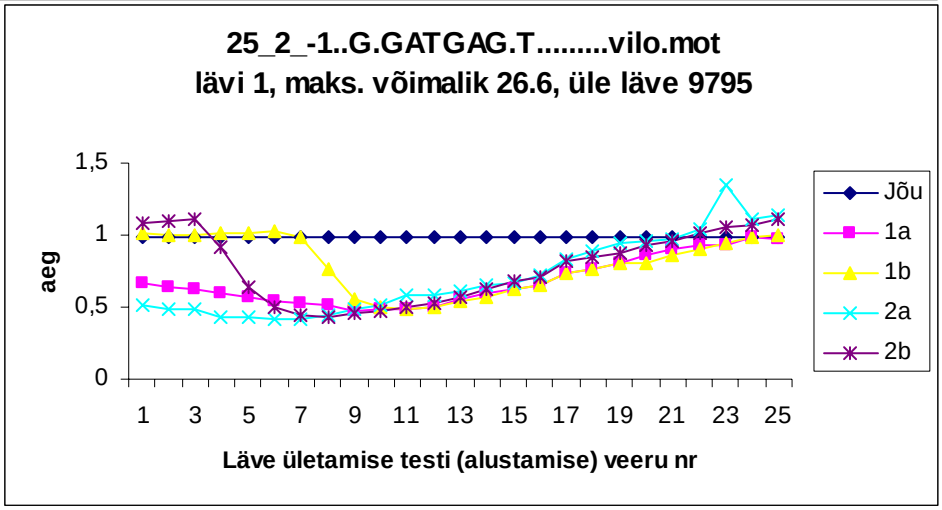
4.2.2 Test pika ja osaliselt konserveerunud motiiviga

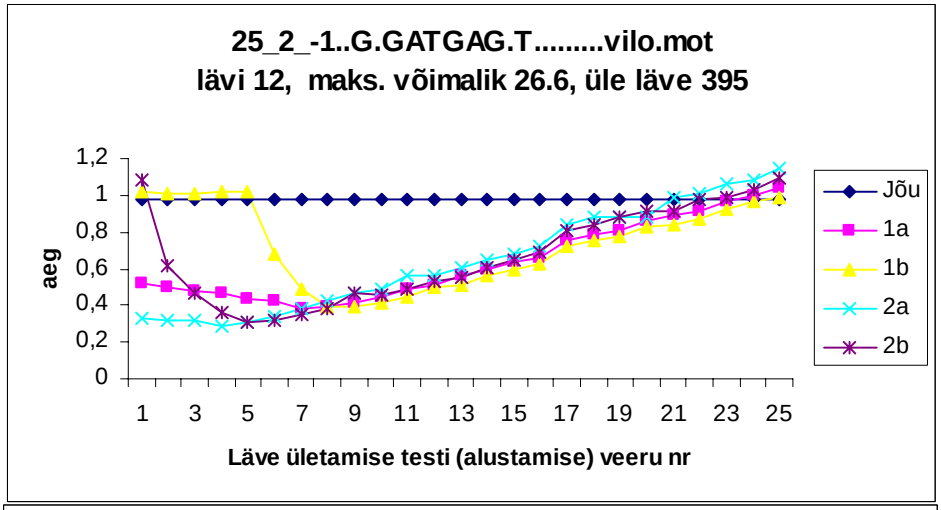
Järgnevalt vaataks olukorda, kus pikendasin motiivi G.GATGAG.T 25 sümboliliseks (antud testimisel kasutasin kaalumaatriksi genereerimiseks motiivide otsimisel „-1:.....G.GATGAG.T...” ja „-1:...G.GATGAG.T.....”), nii et see konserveerunud osa oli kas alguses, keskel või lõpus. Nüüd vaadates jooniseid 4.11 – 4.12 tuleb selgelt välja eelsorteerimist tegevate ja mitte tegevate meetodite vahed. Eelsorteerimisest saadud uus liitmise järjekorra eelis tuleb välja selle kaalumaatriksi puhul, millel konserveerunud osa asetseb viimastes positsioonides. Siit tuleb välja ka eelnevalt mainitud ainult ühte läve testi kasutavate meetodite omavaheline erinevus. Kui esimesed positsioonid on vähemkonserveerunud, siis eelsorteerimist tegev meetod avastab varem lävest mitteülemineku võimalusi.

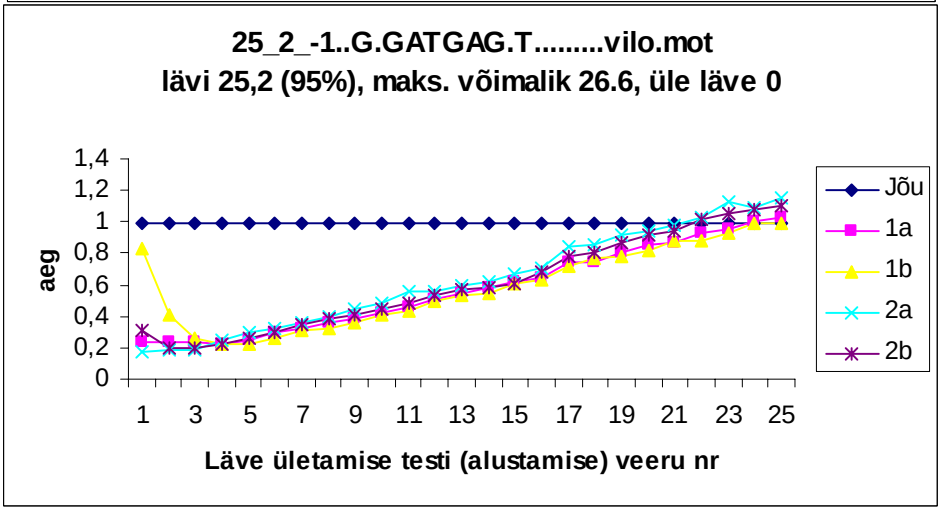
Ilmnes ka see, et sorteeritud järjekorraga ettevaatav skoorimine (joonistel *2a* nime all) on kõige stabiilsem ja parem meetod pikkade motiivide korral. Tema ajavõit, kui läve ületamise testi hakata tegema esimestel positsioonidel, on olenevalt lävest 2-5 kordne (madalama läve puhul väiksem).

Samuti sai lõplikult selgeks ka see, et kui otsida väga kõrge lävega, siis tuleks teha testi kohe esimestel positsioonidel ning ei ole erilist erinevust, millise meetodiga seda teha.

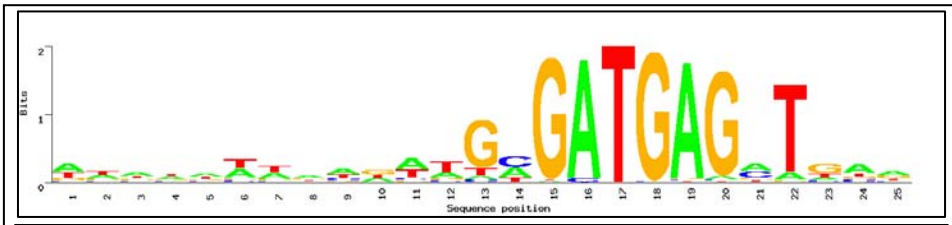


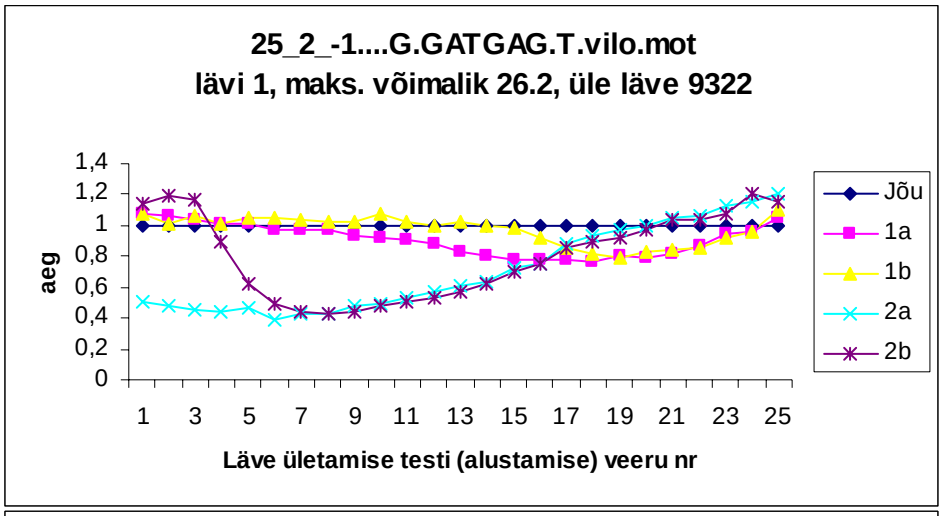


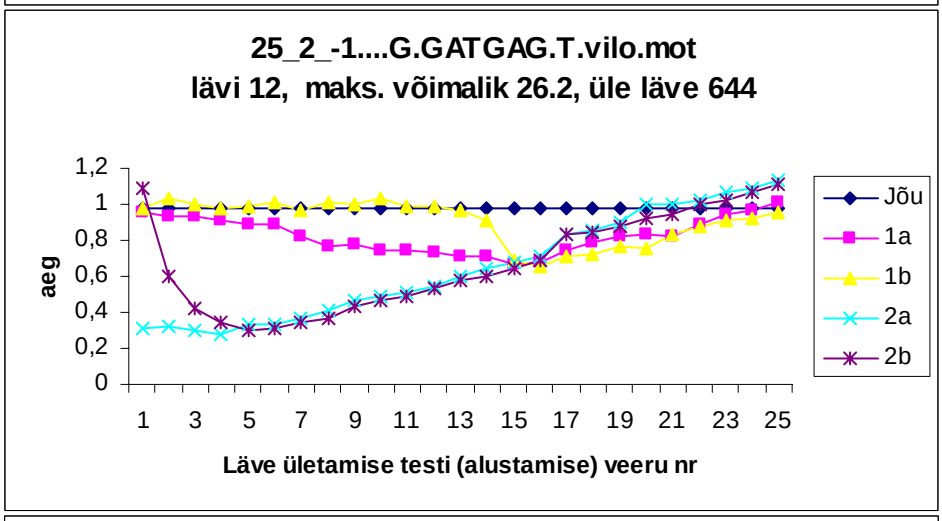


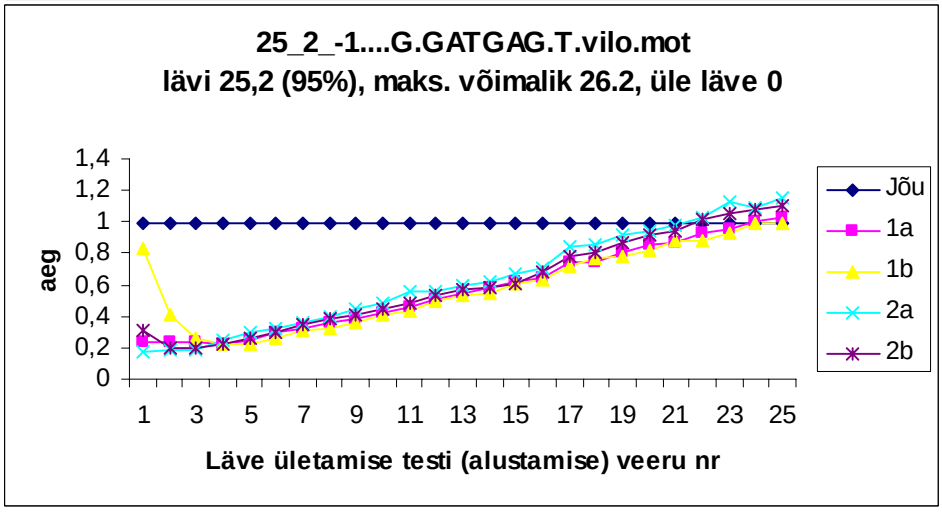


Joonis 4.11: Antud joonisel pikendasime eelneval joonisel vaadeldut mustrit 25 sümboliliseks, lisades viimastele positsioonidele suvalisi sümboleid nii, et konserveerunud osa asetseks esimeses pooles. Antud juhul tuleb selgelt välja kõikide edasiarenduste meetodite eelised jõumeetodi ees. Jooniselt järeldub ka, et a ja b meetodid käituvad sarnaselt madalate ja keskmiste lävede puhul. Kõrgete lävede puhul ei ole samas erilist vahet, millist edasiarendust kasutada - kõik käituvad sarnaselt. Samas on selgelt näha ka 1b ja 2b ning 1a ja 2a väike erinevus: kuna a 2 meetodid kasutavad eelnevalt arvutatud uut summeerimise järjekorda, siis nad avastavad läve ülemineku võimalused varem.









Joonis 4.12: Kui jooniselt 4.11 pikendasin mustrit nii, et konserveerunud osa asetses maatriksi alguses, siis nüüd vaatleme maatriksit, mille konserveerunud osa asetseb maatriksi viimastel veergude. Antud näide toetab täielikult eelmise joonise juures väljatoodud väiteid.

4.2.3 Testide kokkuvõte

Kokkuvõtteks võib öelda, et lühikeste ja peaaegu täielikult konserveerunute motiivide korral ei ole väga suurt ajavõitu võrreldes jõumeetodiga, kui teha otsimist suvalise edasiarendusega (enamvähem kõik on võrdsed). Kui seda teha, siis on kasulik madalate lävede puhul alustada läve ületamise testi tegemist keskmistest positsioonidest ning nihutada seda läve tõstmise korral järjest esimestele positsioonidele. Pikkade ja osaliselt konserveerunud motiivide korral on kõige kasulikum meetod sorteeritud järjekorraga ettevaatav skoorimine (joonistel *2a* nimega) ning teha teste juba esimesest positsioonidest alates.

Allpool olevast tabelist on näha, et kõikide eelpool vaadeldud meetodite jaoks leidub situatsioon, kus ta on teistest parem.
Seletades lahti antud tabeli, siis:

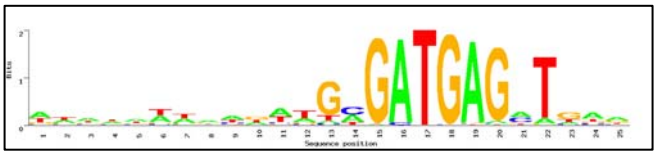
1. Jõumeetod on teistest parem, kui otsitakse tugevalt konserveerunud maatriksiga madala lävega ning kui edasiarendustel läveületamise testi rakendada viimastel veergudel
2. Ettevaatav skoorimine (1a) on parim, kui otsitakse tugevalt konserveerunud maatriksiga kõrge lävega ning läve ületamise testi tehakse esimestel postisioonidel
3. Ettevaatav skoorimine ühe läve ületamise testiga (1b) on teistest kiirem, kui otsitakse tugevalt konserveerunud maatriksiga üle keskmise lävega ning keskmistel positsioonidel
4. Sorteeritud järjekorraga ettevaatav skoorimine (2a) on parim, kui otsitakse pika kaalumaatriksiga, milles on üks konserveerunud positsioon viimastes positsioonides, ning otsimist teostakse üle keskmise lävega ja testi alustatakse esimestes positsioonides.

5. Sorteeritud järjekorraga ühe läve ületamise testiga ettevaatav skoorimine (2b) on parim, kui otsitakse pika kaalumaatriksiga, milles on konserveerunud positsioonid viimastes positsioonides, ning otsimist teostakse madala lävega ja testi tehakse keskmistes positsioonides.

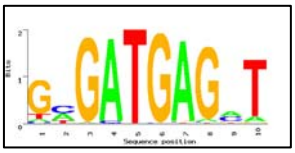
Eelnevast järeldub, et kõik vaadeldud meetodid on mingite kindlate parameetritega teistest parem, seega ka teatavates situatsioonides kasulikum kasutada.

	Jõu	1a	1b	2a	2b
10_2_-1G.GATGAG.T.vilo.mot, veerg 1, 17.59 maks. võimalik 17.59, üle läve 49	0,424	**0,158**	0,180	0,174	0,206
10_2_-1G.GATGAG.T.vilo.mot, veerg 6, lävi 12, maks. võimalik 17.59, üle läve 458	0,430	0,316	**0,268**	0,314	0,312
25_2_-1....G.GATGAG.T.vilo.mot, veerg 1, lävi 12, maks. võimalik 26.2, üle läve 399	0,982	0,956	0,982	**0,312**	1,09
25_2_-1....G.GATGAG.T.vilo.mot, veerg 5, lävi 19.66, maks. võimalik 26.2, üle läve 59	0,992	0,688	1,022	0,314	**0,248**
10_2_-1G.GATGAG.T.vilo.mot, veerg 10, lävi 1, maks. võimalik 17.59, üle läve 14840	**0,394**	0,448	0,420	0,504	0,458

Tabel 4.1: Tabeliga tuuakse välja iga meetodi jaoks situatsioon, kus ta on teistest meetoditest parem



Joonis 4.13: 25_2_-1....G.GATGAG.T.vilo.mot



Joonis 4.14: 10_2_-1G.GATGAG.T.vilo.mot

5 Kaalumaatriksi tekstile sobitamise realisatsioon

Eesmärgiks oli kirjutada programm, mis saaks sisendiks sekventside faili, mustrite faili ning kui kasutaja soovib analüüsi teha teistsuguse sümbolite jaotusega kui on sekventsi fail, siis ka tähestiku fail. Sekventsi ja motiivi fail võib olla nii FASTA [19] kui ka tavalises formaadis (üks rida on üks sekvents-motiiv). FASTA formaadi eripäraks on see, et enne sekvents on üherealine kirjeldus, mis algab ‚>' sümboliga. Mustrifailid võib anda sisendiks ka juba olemas-oleva kaalumaatriksi kujul csv-failina. Programm suudab mustritest koostada nelja erinevat kaalumaatriksi tüüpi: tavaline loendusmaatriks, sagedusmaatriks, tõenäosussuhte ja logaritmilise informatsiooni maatriksit. Nendest kõigist oli juttu eespool.

Programmi tööpõhimõte on väga lihtne ning selle protsess on näitlikult kirjeldatud Lisa 2 oleval joonisel. Esmalt kontrollitakse kõiki sisendparameetreid ning loetakse mällu sisendfailid. Seejärel arvutakse kõikide sisendsekventside pikkused ja kontrollitakse, et ükski ei oleks lühem kui on muster ise. Mustrid peavad olema kõik ühepikkused. Järgnevalt konverteeritakse kaalumaatriks kasutaja soovitud kujule ning alustatakse selle sobitamist sekventsidele (antud versioonis ei ole veel rakendatud jõumeetodi edasiarendusi).

Programmi väljundeid on kaks. Esiteks on väljundiks iga sekventsi segmentaalsed skoorid. Juhul kui otsitakse eeldefineeritud lävega, siis kirjutatakse faili ainult lävega võrdsed või üle läve olnud segmendi skoorid. Fail formaat on äärmiselt selge ja arusaadav. Kõigepealt on sekventsi järjekorra number sisendsekventsides ning sellele järgnevad selle sekventsi segmentide skoorid kujul „positsioon_sekventsis skoor segmendi_sõne". Juhul kui otsitakse ilma läveta, siis on võimalik saada ka antud sekventside

kohta visuaalselt infot. Selleks jagatakse kõik skoorid kasutaja etteantud vahemikesse ning väljastatakse vahemikesse mahtunud skooride arvud. Antud infot töötlen Perlis realiseeritud skriptiga ning kasutan Perli GD teeki, et teha illustreerivat histogrammi (joonist) skooride jaotuse kohta sisendtekstil (vaata näidet Lisa 3 olevalt joonistelt).

Kirjeldatud programmide töötavad ainult UNIX-i laadses keskkonnas ning on testitud RedHat 9.0 ja Fedora Core 1 operatsioonisüsteemis (gcc versioon mõlemal 3.3.2).

Hetkel on programmid ilma graafilise kasutajaliideseta, see-eest on nende kasutamine väga lihtne. Graafilise (veebi põhise) kasutajaliidese tegemine on planeeritud järgmiseks etapiks.

Kokkuvõte

Käesolevas diplomitöös on käsitletud bioloogilises kontekstis tähendust omavate mustrite esitamise erinevaid meetodeid, eriti on peatutud kaalumaatriksi ideel ning vaadeldud erinevaid võimalusi antud kaalumaatriksisse kaalude saamiseks, kaasa arvatud pseudoloendite meetod.

Töö tulemusena on konstrueeritud mitu erinevat algoritmi kaalumaatriksi sobitamiseks tekstidele ning vaadeldud lühidalt nende omavahelisi erinevusi ja testitud neid mitmes erinevas situatsioonis.

Töö praktilise osana valmis programm, millega saab etteantud tekstist ja mustritest valmistatud kaalumaatriksi abil arvutada teksti osalõikudele (mustri pikkusega) skoore, mis näitab kui hästi antud osa tekstist sobib kaalumaatriksiga.

Searching with Position-specific Weight Matrices

Diploma work
Marek Zäuram

Abstract

The recent advances in genome sequencing projects are bringing the importance of computer supported DNA sequence analysis to the attention of large number of scientists. Computationally identifying regulatory motifs in genomes is undoubtedly leading the list of important tasks in this context. The aim of our work is to review different representations of transcription factor binding sites and to create a program for sequence analysis.

In this work we review several methods to represent biologically meaningful motifs. Especially we examine methods of the position weight matrices and study how to obtain values to the different types of matrices. In particular we concentrate in a log-likelihood scoring scheme called information content and pseudocounts method.

In the experimental part of this work we present four algorithms for increasing the speed of matching scoring matrices against long DNA sequences. We describe them in the different tests cases and situations.

Practical part represents new program for sequence analysis using position-specified weight matrices and visualization tool for illustrating scores over the genome.

Viited

[1] Berg, O.G.; von Hippel, P.H. *Selection of DNA binding sites by regulatory proteins. Statistical-mechanical theory and application to operators and promoters*. J. Mol. Biol., 1987

[2] Claverie, J.-M. *Some useful statistical properties of position-weight matrices*, Comput. Chem, 1994, 18, 287-293

[3] Fields, D.S.; He Y, Al-Uzri, A.Y.; Stormo, G.D. *Quantitative specificity of the Mnt repressor*, Journal of Molecular Biology, 1997, 271(2):178-94

[4] Heinikoff, Jorja G; Heinikoff, Steven. *Using substitution probabilities to improve position-specific scoring matrices*, 1996.

[5] Hertz, Gerald Z.; Stromo, Gary D. *Identifying DNA and protein patterns with statistical significant alignments of multiple sequences*, Bioinformatics, 1999, 15, 563-577

[6] Heumann, J.; Lapedes, A.; Stormo G. *Neural networks for determing protein specificity and multible alignment of binding sites.* In Conferece on Intelligent Systems for Molecular Biology, 1994, volume 2, 188-194.

[7] Lawrence, C.E.; Altschul, S.F.; Boguski, M.S.; Liu, J.S, *Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment*, Science, 1993, 268, 208-214.

[8] Moss, G.P. *Nomenclature for Incompletely Specified Bases in Nucleic Acid Sequences*, http://www.chem.qmw.ac.uk/iupac/misc/naseq.html (väisatud viimati 24.05.2003)

[9] Rosenthal, Elisabeth. *Finding Instances of Known Sites (Lecture notes)*, 2000.

[10] Sequence data (yeast), http://ep.ebi.ac.uk/EP/PATMATCH/SEQUENCES/Yeast_-600_+2_W_all.fa (viimati väisatud 17.03.2004)

[11] Schneider, T.D.; Stormo, G.D.; Gold, L.; Ehrenfeucht, A. *Information content of binding sites on nucleotide sequences*, J. Mol. Biol.,1986, 188, 415-431.

[12] Schneider, Thomas D. *Information Theory Primer*, 2003 http://www.lecb.ncifcrf.gov/~toms/paper/primer/ (viimati väisatud 17.03.2004)

[13] Stormo, Gary D. *DNA binding sites: represation and discovery*, Bioinformatics, 2000, 14, 16-23.

[14] Stormo, G.D.; Fields, D.S. *Specificity, free energy and information content in protein-DNA interactions*, Trends Biochem. Sci., 1998

[15] Vilo, J.; Kapushesky, M.; Kemmeren, P. *Expression Profiler: tools and documentation http://ep.ebi.ac.uk/* (viimati väisatud 17.03.2004)

[16] Vilo, Jaak. *Pattern Discovery from Biosequences*, University of Helsinki, 2002, PhD thesis

[17] Vilo, Jaak. *PATMATCH,* 2001, http://ep.ebi.ac.uk/EP/PATMATCH/ (viimati väisatud 17.03.2004)

[18] Wu, Thomas D.; Nevill-Manning, Graig G. Brutlag; Douglas L. *Fast probablistic analysis of sequence function using scoring matrices*, Bioinformatics, 2000, 16, 233-244

[19] Genomatix Software GmbH, *DNA Sequence formats*, 2004, http://www.genomatix.de/online_help/help/sequence_formats.html

[20] Brejova,, Bronna; DiMarco, Chrysanne; Vina, Tomaš. *Finding Patterns in Biological Sequences*, 2000.

LISAD

Lisa 1. Programmide sisend-väljund andmete skeem

Lisa 2. Üldine kaalumaatriksi sobitamise programmi protsessi skeem

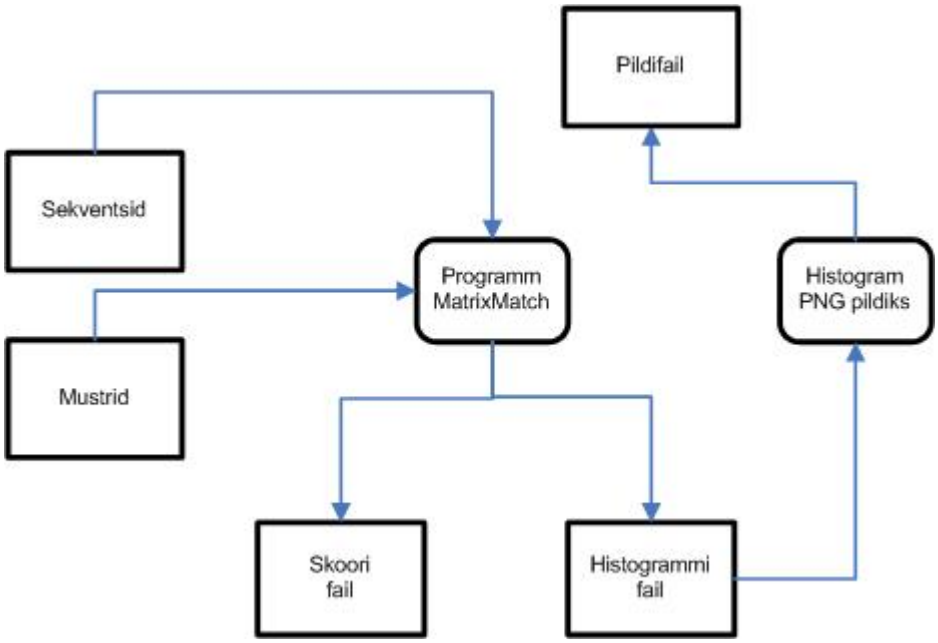
Lisa 3. Visualiseerimise skriptist saadud näide.

Lisa 4. Kaalumaatriksi sobitamise programmi kasutusjuhend

Lisa 5. Visualiseerimise (histogrammi) joonistamise skripti kasutusjuhend

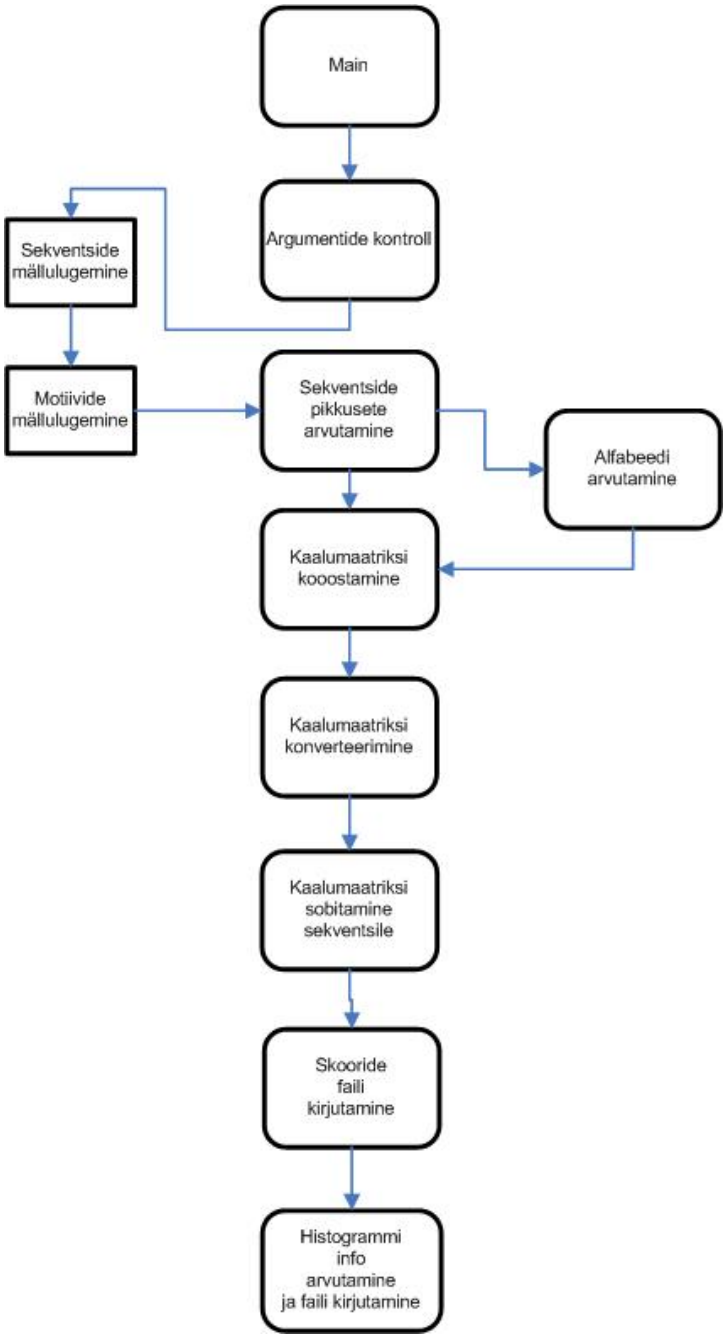
Lisa 1.

Programmide sisend-väljund andmete skeem



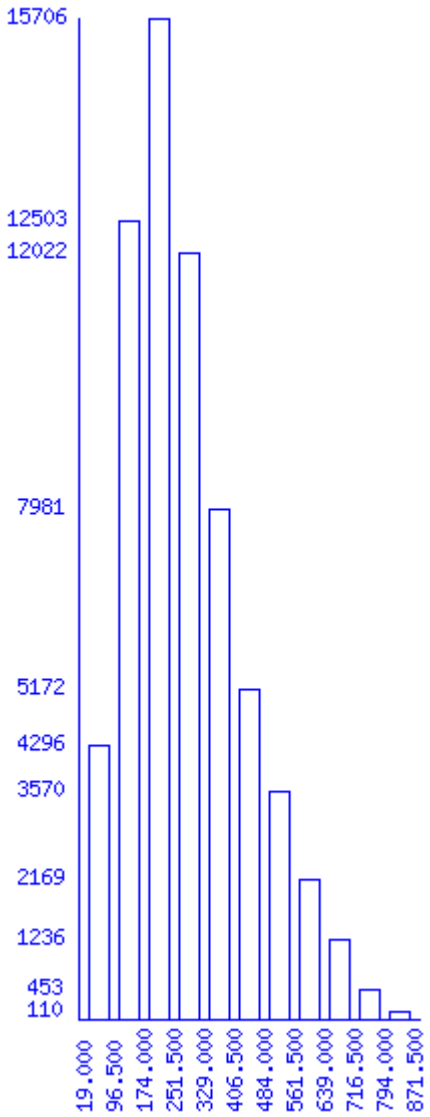
Lisa 2.

Üldine kaalumaatriksi sobitamise programmi protsessi skeem



Lisa 3.

Visualiseerimise skriptist saadud näide.



Lisa 4.

Kaalumaatriksi sobitamise programmi kasutusjuhend

Programmile etteantavad parameetrid

Programmi tööks vajalikke parameetreid saab, kui käivitada programm ühegi parameetrita.

```
#./matrixmatch
[-p pattern file]
[-pt pattern file type: 1-fasta, 2-plain sequence(default) 3-csv]
[-ptc csv type (only need if the type of pattern file is cvs) 0-verical (default) 1- horizontal]
[-ptdelim csv delimiter (only need if the type of pattern file is csv). Default ";"]
[-ptcno flag is set, then no need to convert matrix (only if file type is csv)]
[-f sequence file]
[-ft sequence file type: 1-fasta(default), 2-plain sequence]
[-pm matrix type: 1-count(default), 2-frequency, 3-logratio, 4-oddscore]
[-mthres threshold]
[-mthres% - threshold is percentage form maximum matrix score]
[-verbose output all to screen]
[-pverbose output matrix to screen]
[-sf name of the result score file (default "sequence_file_name.score")]
[-hstf name of the result histgram file (default "sequence_file_name.hst")]
[-slots number of histogram's slots]
[-alpha "char 'percent from total'" for exampe "A 10 C 20 G 30 T 40"]
[-alphaf "file name"]
```

Parameetrite täielikum kirjeldus:

-p patter_file

Antud parameeter on vajalik programmi tööks. Sellega antakse programmile ette mustrifaili asukoht. Võimalik on anda ette see suhtelise või absoluutse teena. Näiteks

-p /home/marek/data/pattern_file.mot

või

-p ../../data/pattern_file.mot

-pt pattern file type: 1-fasta, 2-plain sequence(default) 3-csv

Parameeteriga näidatakse mustrifaili tüüp. Mustrifail tüübid on järgmised:

1-fasta tüüpi

2-tavaline sekventside järjestus

3-csv tüüpi maatriks

-ptc cvs type (only need if the type of pattern file is cvs) 0-vertical (default) 1-horizontal

Parameetriga näidatakse ära programmi imporditava csv failide tüübid. Antud parameeter on ainult vajalik siis, kui mustrifaili tüübiks on csv.

Tüübid:

0-Vaikimisi parameeter. Tähed asetsevad vertikaalis (rea alguses) ning andmed vasakult paremale horisontaalis.

1-Tähed asetsevad esimesel real ning andmed vertikaalis ülalt alla.

-ptdelim csv delimiter (only need if the type of pattern file is cvs). Default ";"

Csv faili väljade eraldussümbol. Vaikimisi kasutatakse ";".

-ptcno flag is set, then no need to convert matrix (only if file type is csv)
Antud lippu kasutakse siis, kui csv-na sisestavad andmed on juba õigel kujul ja neid ei ole vaja teisendada.

-f sequence file
Sekventsifaili asukoht. Võimalik on anda ette see suhtelise või absoluutse teena. Näiteks
-f /home/marek/data/sequence1.seq
või
-f ../../data/sequence1.seq

-ft sequence file type: 1-fasta(default), 2-plain sequence
Sekventsifaili tüüp. Võimalikud tüübid:
1-fasta formaadis fail (vaikimis)
2-tavaline sekventside järjestus

-pm matrix type: 1-count(default), 2-frequency, 3-logratio, 4-oddscore
Antud parameetriga määratakse ära maatriksi tüüp.
1- absoluutsete sageduste maatriks
2- relatiivse sageduse maatriks
3-informatsiooni sisalduse maatriks.
4-oddscore ehk relatiivese sageduse ja taustasageduse jagatis.

-mthres threshold
Antud parameetriga antakse ette lävi, millest kõrgemaid tulemusi raporteeritakse.
Näiteks:
-mscore 0.8
Raporteerikse kõik tulemused, mis on suuremad 0.8-t.

-mthres%: threshold is percentage form maximum matrix score
Kui lipp on olemas, siis võetakse **–mscore**'ile antud parameetrit protsendina maksimaalsest võimalikust maatriksi summast.

-verbose output all to screen
Kui antud parameeter on antud, siis kuvatakse programmi vahetulemused jooksvalt ekraanile.

-pverbose output matrix to screen
Kui antud parameeter on antud, siis kuvatakse maatriks ekraanile.

-sf name of the result file (default "pattern_file_name.score")
Parameetriga antakse ette tulemusfaili asukoht. Võimalik on anda ette see suhtelise või absoluutse teena. Näiteks
-sf /home/marek/data/result.score
või
-sf ../../data/result.score
Vaikimisi väärtuseks on sekventsifail.score.

-hstf name of the result histgram file (default "sequence_file_name.hst") Antud parameetriga antakse programmile ette fail, kuhu kirjutatakse histogrammi andmed. Võimalik on anda ette see suhtelise või absoluutse teena. Näiteks
-sf /home/marek/data/result.hst
või
-sf ../../data/result.hst
Vaikimisi väärtuseks on sekvetsifail.hst.
Saab kasutada ainult siis, kui läve (-mthres) ei ole defineeritud.

-slots number of histogram's slots
Parameeteriga antakse ette histogrammis olevate vahemike arv. Saab kasutada ainult siis, kui läve (-mthres) ei ole defineeritud.

-alpha "char 'percent from total'" for exampe "A 10 C 20 G 30 T 40"
Selle argumendiga saab käsurealt ette anda informatsiooni sisalduse maatriksi jaoks vajalikku taustasagedust ja alfabeeti.
Vaikimisi võetakse alfabeediks ja sümbolite sagedusteks sekventsifaili sümbolite sagedus

[-alphaf "file name"]

Parameetriga antakse ette alfabeeti sisaldav fail.

Faili formaat peab olema järgmine:

SÜMBOL_1 SAGEDUS

...

SÜMBOL_N SAGEDUS

Sisend failide formaadid:

Erinevate DNA sekventside esitusformaatide kohta saab lugeda: http://www.genomatix.de/online_help/help/sequence_formats.html

FASTA

Sekventsid, mis on fasta tüüpi, algavad üherealise kirjeldusega, millele järgneb ühe või mitmerealine sekventsi andmed. Kirjelduse rida algab "suurem kui" sümboliga (">") Sõna, mis järgneb "suurem kui" sümbolile (">"), on sekventsi "ID" (nimi). Ülejäänud rida on antud sekventsi kirjeldus. "ID" ja kirjeldus ei ole kohustuslikud Kõik read võiksid olla lühemad kui 80 sümbolit (http://ngfnblast.gbf.de/docs/fasta.html). Sekvents lõppeb, kui algab uus rida "suurem kui" sümboliga (">").

Näide fasta tüüpi kirjest:

>YAL036C

TGTTCTTTCTTCTTCTGCTTCTCCTTTTCCTTTTTTTCCTTCTCCTTTTCCTTCTT

GGACTTTAGTATAGGCTTACCATCCTTCTTCTCTTCAATAACCTTCTTTTCTTG

CTTCTTCTTCGATTGCTTCAAAGTAGACATGAAGTCGCCTTCAATGGCCTCAG

CACCTTCAGCACTTGCACTTGCTTCTCTGGAAGTGTCATCTGCACCTGCGCTG

CTTTCTGGATTTGGAGTTGGCGTGGCACTGATTTCTTCGTTCTGGGCGGCGTCT

>YAL025C

CTTAGAAGATAAAGTAGTGAATTACAATAAATTCGATACGAACGTTCAAATAGTCAAGAATTTCAT

PLAIN SEQUENCE

Seda tüüpi sekventsi fail on tavaline tekstifail, kus ühel real paiknev tekst on sekvents st iga rida on omaette sekvents.

Näide kuue sekventsilisest failist (ehk kuue realisest failist):

Rea nr., sekvents nr.	Rida
1	TATAAT
2	TAATAT
3	ATAATT
4	TGATGT
5	AAGATT
6	TATAAT

Antud näide asuv testandmete kataloogis ./data failis nimega pat1.mot.

CSV andmetena lähteandmed.

Csv-na saab programmile ette anda ainult eelnevalt kaalumaatriksiks arvutatud andmeid. Programm tunneb kahte tüüpi csv-d.

Tüüp 0:

A;2.00;4.00;3.00;4.00;3.00;0.00

C;0.00;0.00;0.00;0.00;0.00;0.00

G;0.00;1.00;1.00;0.00;1.00;0.00

T;4.00;1.00;2.00;2.00;2.00;6.00

Esimeses veerus asub sümbol ning järgnevates veergudes selle sümboli kaal vastavas positsioonis.

Tüüp 1:

A;C;G;T

2.00;0.00;0.00;4.00

4.00;0.00;1.00;1.00

3.00;0.00;1.00;2.00

4.00;0.00;0.00;2.00

3.00;0.00;1.00;2.00

0.00;0.00;0.00;6.00

Sümbolid asuvad esimeses reas ning nende all veerus on vastava sümboli kaal vastavas positsioonis.

CSV failis peab väljade eraldajate vahel olema vähemalt tühik või siis '0' st ei tohi esineda situatsiooni <eraldaja><eraldaja> vaid peaks

olema <eraldaja>0.00<eraldaja>. Vastasel juhul lõpetab programm veateatega.
Mõlemad antud näite puhul on programmi seisukohalt tegu identsete andmetega ning nad on saadud PLAIN SEQUENCE näitest ja on absoluutse sageduse skaalas.

Väljund failide formaadid:
Hetkel väljastaks programmist kahte tüüpi faile. Ühes on kaalumaatriks sekventsile sobitamise tulemused (.score) ning teises vahemikesse jagatud ning kokkuloendatud sobitamise tulemused (.hst) ehk fail, millest tehakse histogramm.
Mõlemas failis kirjutatakse kommentaare '#' sümboliga.
score tüüpi fail
Faili algusse kirjutakse kommentaaridena programmi käivitanud käsurea parameetrid. Sammuti kirjutatakse kommentaarina kaalumaatriks ning tähestiku suurus ja muud arvutatud tulemused (aritmeetiline keskmine, standardhälve jne.). Andmeid kirjutakse järgnevas vormis: kõigepeal kommentaarina sekventsi number ning järgnevalt siis kolmes veerus andmed: esimese veerus on sobitamise alustamise alguspunkt sekventsis, teises tulemus ning kolmandas lõik sekventsist, mille peal sobitamist tehti.
Antud näide on saadud kasutades testandmete kataloogis ./data olevat faili 34x22.mot kui sekventsifaili ning pat1.mot kui mustrifaili ning tulemuse saamiseks on kasutatud absoluutse sageduse maatriksit ja kaalude liitmist.
Näites on mustri pikkuseks 6 ja sekventsi pikkuseks 34. Vaatame kuidas esimese sekventsi alguspunktiga 0 tulemus on saadud: lõiguks on võetud sümbolid, alguspositsiooniga 0 ning lõpp-positsiooniga 5 (k.a.). Antud sõneks on AGCTGT. Tulemus saadakse kui summeeritakse sõnes esineva sümboli kaalud esinemispositsioonides: A kaal esimeses positsioonis on 2.00, G kaal teises positsioonis on 0.00 ning C kaal kolmandas postitsoonis on 1.00 jne. Ehk AGCTGT tulemus on saadud järgmiselt: 2.00+0.00+1.00+2.00+1.00+6.00 = 15.00

Näide:

```
#Result file.
#Parameters of command line:
#./marek -f ../data/34x22.mot -ft 2 -p ../data/pat1.mot -pt 2
#The size of charset: 4.
#          A       C       G       T
#1.     2.00    0.00    0.00    4.00
#2.     4.00    0.00    1.00    1.00
#3.     3.00    0.00    1.00    2.00
#4.     4.00    0.00    0.00    2.00
#5.     3.00    0.00    1.00    2.00
#6.     0.00    0.00    0.00    6.00
#Mean is 12.136364, deviation is 4.218154 threshold is not set (all
occurences)
#There were 57918 sums over threshold
#The maximum possible sum is 6.000000

#sequence no. 1
    0     12.00    AGCTGT
    1       4.00    GCTGTA
    ...
    27    11.00    CAGATG
    28    9.00    AGATGA

#sequence no. 22
    0    19.00    TCAATT
    1    11.00    CAATTA
     ...
    28    13.00    TCAATA
```

hst tüüpi fail

Esimeses veerus kuvatakse vahemiku numbrit (nummerdama hakatakse 0-st), teises veerus on vahemiku algus ning kolmandas veerus vahemiku

lõpp. Neljandas veerus kuvatakse nende tulemuste arv, mis sellesse vahemikku jäid.

Näide:

```
# HISTOGRAM of distances in ../data/98.seq
#buck    from            to                nr of elements inbucket
0            0.000000    2.400000    650
1            2.400000    4.800000    2597
...
9            24.000001  26.400001  1
```

Lisa 5.

Visualiseerimise (histogrammi) joonistamise skripti kasutusjuhend

```
[zauram@kiwi perl]$ perl hst_to_png.pl --help
HST_TO_PNG.MAN(1)     User Contributed Perl Documentation    HST_TO_PNG.MAN(1)

TITLE
SYNOPSIS
DESCRIPTION
ARGUMENTS
        --help
               print complete man page
        -i, --input_file FILE
               input histogram file
        -o, --output_file FILE
               path to output SWOG commands file
        -p, --output_image FILE
               path to output imamge file (type is png)

OPTIONS
        --version
               print program info
        --debug 0
               don't print debugging information (default)
        --debug 1
               print debugging information
        -s, --swog FILE
               the path to the SWOG perl script (default is "../swog_0.1/swog.pl")

REPORTING BUGS
       Report bugs to <mz@math.ut.ee>.

LICENSE
AUTHOR
       Written  by Marek Zauram Copyright (C)  2004 All rights reserved

perl v5.8.3                       2004-05-20                HST_TO_PNG.MAN(1)
```