TARTU ÜLIKOOL
MATEMAATIKA-INFORMAATIKATEADUSKOND
MATEMAATILISE STATISTIKA INSTITUUT

Kaspar Märtens

MDL-meetod diferentsiaalselt metüleeritud regioonide tuvastamiseks

Bakalaureusetöö (6 EAP)

Juhendaja: Raivo Kolde

Tartu 2013

Sisukord

Sissejuhatus

Bioloogilist huvi pakub küsimus, millised tegurid reguleerivad geenide avaldumist. DNA metülatsioon on üks mitmetest mehhanismidest, mida rakkudes kasutatakse geenide vaigistamiseks. Metülatsioon omab funktsionaalset rolli ainult DNA järjestuse kindlatel positsioonidel, mida nimetatakse CpG saitideks. Tihti on järjestikuste CpG saitide metüleeritus sarnane, seega on mõttekas otsida ühesuguse metülatsioonimustriga pikemaid regioone. Diferentsiaalselt metüleeritud regioonideks (DMR) nimetatakse selliseid järjestikusi CpG saite, kus erinevate gruppide (näiteks vähihaigete ja tervete, noorte ja vanade indiviidide või erinevat tüüpi kudede) vahel on metüleerituses erinevusi.

Käesoleva bakalaureusetöö eesmärgiks on välja töötada meetod diferentsiaalselt metüleeritud regioonide tuvastamiseks, mida saaks kasutada eelkõige metülatsioonikiibi andmetel. Selleks soovime jagada DNA järjestuse optimaalsel viisil segmentideks ning seejärel teha iga segmendi kohta otsuse, kas seal esineb diferentsiaalne metülatsioon või mitte.

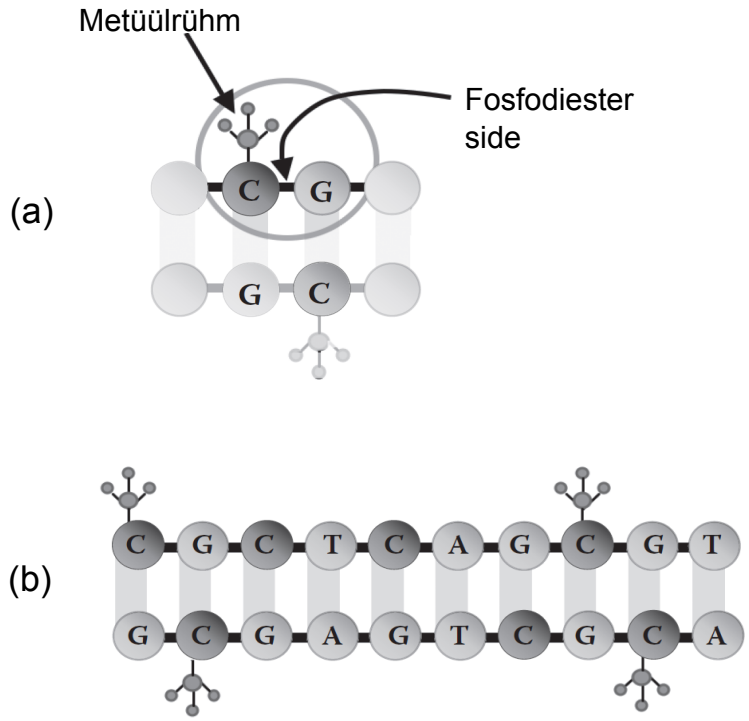
Töö algul antakse ülevaade DNA metülatsioonist ning formuleeritakse matemaatiline probleem, mis seisneb andmetest järjestikuste segmentide leidmises. Järgneb ülevaade tõenäosuslike mudelite kodeerimisest ning MDL-printsiibist, sest sellele toetume optimaalse segmentatsiooni leidmisel. Peatükis 3 on toodud üldine raamistik, mille kohaselt jagada andmed parimal viisil segmentideks, kasutades segmendiviisi defineeritud mudeleid ning valides neist MDLi mõttes parima. Selline raamistik võimaldab kasutada segmentidel andmete kirjeldamiseks suvalisi mudeleid, mille alusel on võimalik arvutada andmete tõepära. Seejärel on seda raamistikku kasutatud kahe konkreetse meetodi jaoks. Neist on lähemalt uuritud meetodit, mis põhineb segmentidele lineaarsete mudelite sobitamisel: testime seda nii simuleeritud kui ka bioloogilistel andmetel, lisaks võrdleme saadud tulemusi ühe võimaliku alternatiiviga.

Need mõlemad meetodid implementeeriti programmeerimiskeeles R.

Tänan oma juhendajat Raivo Koldet põneva probleemipüstituse, arvukate selgituste ning igasuguse abi eest. Samuti tänan Sven Lauri väärtuslike nõuannete eest.

1 DNA metülatsioonist

DNA järjestus koosneb nukleotiididest A, T, G, C. Kui DNA järjestuses esinevad kõrvuti C ja G, nimetatakse seda CpG dinukleotiidiks ehk CpG saidiks. CpG dinukleotiide esineb inimese genoomis neli korda harvem võrreldes olukorraga, kui nad paikneksid juhuslikult (arvestades C ning G sisaldust DNAs) [1]. CpG saarteks nimetatakse suhteliselt lühikesi (mõnisada kuni mõni tuhat baasipaari) DNA piirkondi, kus CpG dinukleotiidide osakaal on suhteliselt suur. On teada, et rohkem kui pooled geenidest sisaldavad CpG saari ning et CpG saarte metülatsioon reguleerib geenide avaldumist [2].



Joonis 1. DNA metülatsiooni illustratsioon. Joonise (a) osas on näide ühest CpG dinukleotiidist, mis on metüleeritud. Joonise (b) osas on näide DNA kaksikahelast, mille kummalgi ahelal on kaks metüleeritud CpG saiti. [3]

DNA metüleerumine (vt joonis 1) toimub selgroogsetel organismidel tavaliselt CpG dinukleotiidides asuva tsütosiini (C) muundumisena 5-metüül-tsütosiiniks. Seega räägime edaspidi metülatsioonist ainult CpG saitide kontekstis. DNA metülatsioon mõjutab otseselt geenide avaldumist. On teada, et kõik CpG saidid ei ole metüleerunud ning et nende metülatsioon on koespetsiifiline. Lisaks teame, et lähestikku asuvad CpG saidid on metüleeritud sarnaselt (seda kinnitab näiteks joonis 2 lk 19). See asjaolu ongi probleemipüstituse aluseks: kui järjestikuste CpG saitide metüleeritus on sarnane, on mõttekas grupeerida sellised saidid pikemateks regioonideks ning uurida üksikute saitide

asemel saadud regioone. Pealegi on intuitiivselt selge, et kui näeme mingisugust huvitavat mustrit mitmetel järjestikustel CpG-del, saame olla märksa kindlamad, et tegu pole juhusliku mustriga ning arvatavasti reguleerib nende saitide metülatsiooni üks ja sama mehhanism.

Iga CpG metülatsiooni kirjeldatakse beeta-väärtusega, mida võib interpreteerida selliste rakkude osakaaluna, milles oli see CpG metüleeritud. Seega jääb beeta-väärtus alati 0 ja 1 vahele, kusjuures 0 näitab madalat metüleerituse taset ning 1 kõrget. Diferentsiaalseks metülatsiooniks nimetatakse olukorda, kus CpG saidi metüleeritus on mõnedel indiviididel (näiteks vähihaigetel) või mõnedes kudedes (näiteks luuüdi korral) erinev kui teistes.

Illumina HumanMethylation450 metülatsioonikiibid võimaldavad saada beeta-väärtusi rohkem kui 450 000 erineva CpG kohta, nii on ühe geeni kohta teada keskmiselt 17 CpG metülatsioon. See annab hea võimaluse uurida DNA metülatsioonimustreid pikemate lõikude ehk regioonide kaupa.

TÜ Eesti Geenivaramul on erinevaid andmestikke, mis on saadud sellise tehnoloogia abil. Näiteks üks andmestik on 4 doonori 17 koe metülatsiooni kohta, mis võimaldab uurida koespetsiifilist diferentsiaalset metülatsiooni. Teine andmestik on 100 patsiendi erinevate vererakkude metülatsiooni kohta. Need indiviidid jagunevad vanuse järgi selgelt kahte gruppi: vanad ja noored. Seega saab muu hulgas uurida, kuidas on vananemine ning metülatsioon seotud.

Käesoleva töö eesmärgiks on grupeerida beeta-väärtuste alusel CpG saidid pikemateks regioonideks, et leida diferentsiaalselt metüleeritud regioone. Töö edasises osas jätame suuresti kõrvale probleemipüstituse bioloogilise tausta. See tähendab, et edaspidi mõistame püstitatud ülesande all järjestatud andmetest teatavate segmentide leidmist. Mõisteid *regioon* ja *segment* kasutame sünonüümidena, kusjuures esimest eelistame bioloogilistest andmetest rääkides ning teist abstraktsete järjestatud andmete kontekstis. Jagunegu andmestikus olevad objektid (näiteks patsiendid või koed) mingi tunnuse alusel gruppidesse. Siis mõistame püstitatud ülesande all andmetest selliste segmentide leidmist, kus kõigi gruppide keskväärtused ei ole võrdsed.

2 MDL-printsiip

Selles peatükis antakse ülevaade MDL-printsiibist, mis põhineb allikatel [4], [5] ja [6]. Seda printsiipi läheb meil vaja töö järgnevates peatükkides, et valida teatud mudelite seast välja parim. Kuna terves bakalaureusetöös eeldame, et kodeerime andmeid bittide abil, tähistame kõikjal kahendlogaritmi $\log := \log_2$.

Lühima esituse printsiip (*Minimum Description Length principle*, edasises: MDL-printsiip) on informatsiooniteoreetiline lähenemine, mida kasutatakse mudeli valiku meetodina.

See printsiip põhineb tähelepanekul, et mida rohkem reeglipärasust andmetes leidub, seda enam on võimalik neid andmeid kokku pakkida (ehk esitada neid andmeid mingi mudeli abil, kasutades vähemat arvu bitte).

Näiteks on intuitiivselt selge, et bitistring "0001000100010001...0001", mille pikkuseks on miljon bitti, on “väga reeglipärane” ning seda on võimalik esitada mõnel lühemal viisil.

Andmete *esitamise* all peame silmas nende kodeerimist. Seega MDL-printsiibi kohaselt vaadeldakse andmeid kodeeritava informatsioonina, ning selle kodeerimiseks kasutatakse potentsiaalseid mudeleid, mille seast otsime parimat. Eesmärgiks on valida ette antud mudelite seast selline, mis võimaldab andmeid enim kokku pakkida. Seejuures me ei eelda, et nende seas leiduks mingi jaotus, millest vaatlused tegelikult pärit on, vaid lihtsalt otsime mudelit, mis võimaldaks andmete lühimat esitust.

Olgu meil andmed D ning mingisugune mudelite hulk $\mathcal{M}$. Otsime mudelit $M \in \mathcal{M}$, mis andmeid D kõige paremini kirjeldaks. Eeldame, et iga mudel M määrab mingi tõenäosusjaotuse. MDL-printsiibi kohaselt on selliseks mudeliks $M \in \mathcal{M}$, mille korral minimeeritakse summa

$$L(M, D) := L(M) + L(D|M),$$

kus

- $L(M)$ tähistab mudeli M esituspikkust (*description length*)
- $L(D|M)$ andmete D esituspikkust tingimusel, et neid andmeid kirjeldatakse mudeli M põhjal.

Teisisõnu, parim mudel on selline, mille korral on mudeli M kirjeldamiseks ja mudeli M põhjal andmete D kirjeldamiseks kuluvate bittide arvu summa vähim võimalik. On

selge, et väga lihtsa mudeli korral on $L(M)$ väike, aga see võib andmeid suhteliselt halvasti kirjeldada ning seega on $L(D|M)$ suur. Ka vastupidi, väga keerulise mudeli korral on $L(M)$ suur, aga ta arvatavasti kirjeldab andmeid väga täpselt ning seega on $L(D|M)$ väike. Summa $L(M) + L(D|M)$ minimeerimisest võib mõelda kui kompromissi otsimisest mudeli keerukuse ning prognoosimistäpsuse vahel.

Järgnevas alapeatükis näitame, kuidas avaldub $L(D|M)$ mudeli M poolt defineeritud jaotusfunktsiooni või tihedusfunktsiooni kaudu.

2.1 Tõenäosusliku mudeli kodeerimine

Olgu $\mathcal{X}$ lõplik või loenduv hulk. Olgu $\mathcal{D} = \{0, 1\}$ ning $\mathcal{D}^* = \bigcup_{n=1}^{\infty} \mathcal{D}^n$. Iga hulga $\mathcal{X}$ elementi soovime esitada kodeerimistähtede $\mathcal{D} = \{0, 1\}$ lõpliku järjestkirjutisena ehk koodisõnana ehk bitistringina.

Koodiks nimetatakse kujutust $C : \mathcal{X} \to \mathcal{D}^*$. On loomulik eeldada, et kasutame kodeerimiseks prefikskoodi. Prefikskoodiks nimetatakse sellist koodi, mille korral ükski koodisõna ei ole mõne teise koodisõna prefiksiks ehk alguseks. Muuhulgas on prefikskoodid üheselt dekodeeritavad.

Elemendi $x \in \mathcal{X}$ kodeerimiseks (koodi C abil) kuluvate bittide arvu tähistame $L_C(x)$. Teisisõnu, $L_C(x)$ näitab elemendi x koodisõna pikkust. Nimetame seda ka elemendi x esituspikkuseks.

Prefikskoodi C korral kehtib Krafti võrratus:

$$\sum_{x_i \in \mathcal{X}} 2^{-L_C(x_i)} \leq 1.$$

Kehtib ka vastupidine. Kui on antud täisarvud n_i, mille korral $\sum 2^{-n_i} \leq 1$, siis leidub selline prefikskood C nii, et $n_i = L_C(x_i)$. (Tõestust vt [6].)

Meile pakub huvi aga tõenäosusliku mudeli kodeerimine. Seega vaatleme olukorda, kus kodeeritavad elemendid $x \in \mathcal{X}$ on juhuslikud. Olgu P tõenäosusjaotus hulgal $\mathcal{X}$. Siis on iga elemendi x esinemise tõenäosus $P(x)$. Krafti võrratusest järeldub, et leidub selline kood C, et iga $x \in \mathcal{X}$ korral $L_C(x) = \lceil -\log P(x) \rceil$. Siin $\lceil \cdot \rceil$ tähistab arvu ülemist täisosa. Sellist koodi nimetatakse Shannon-Fano koodiks.

Tegelikult huvitab meid MDL-printsiipi kasutades ainult andmete lühima esituspikkuse

leidmine, mitte aga kood ise, mille korral see saavutatakse. Soovime üldistada esituspikkuse funktsiooni L mõistet. Loobudes nõudest, et koodisõnade pikkused peavad olema täisarvud, defineerime iga $x \in \mathcal{X}$ korral

$$L(x) = -\log P(x)$$

ning nimetame seda funktsiooni L esituspikkuse funktsiooniks. On selge, et kui $-\log P(x)$ on täisarv, siis ühtib $L(x)$ Shannon-Fano koodi omaga. Vastasel juhul erineb ta sellest ülimalt ühe biti võrra.

Eeldame, et selliselt defineeritud koodipikkustega "kood" (sest mittetäisarvuliste koodipikkuste tõttu ei pruugi ta päriselt olla kood) rahuldab Krafti võrratust. Osutub, et siis on ta keskmiselt kõige lühema pikkusega kood.

Põhjendus. Tõepoolest. Jaotusega P juhusliku elemendi kodeerimiseks kuluvate bittide arvu keskväärtuseks on

$$\sum_{x \in \mathcal{X}} P(x)L(x),$$

mida võime ka nimetada koodi keskmiseks pikkuseks. Osutub, et koodi keskmine pikkus on alati vähemalt sama suur kui jaotuse P entroopia $H(P)$:

$$\sum_{x \in \mathcal{X}} P(x)L(x) \geq \underbrace{-\sum_{x \in \mathcal{X}} P(x) \log P(x)}_{H(P)}.$$

(Tõestust vt [6], sest seal toodud tõestus kehtib ka mittetäisarvuliste koodipikkuste korral.) On selge, et meie valitud $L(x) = -\log P(x)$ korral kehtib see võrratus võrdusena. See tähendab, et selliste koodipikkustega "kood" on lühima keskmise koodipikkusega ning seega on selliste koodipikkuste valik teatavas mõttes optimaalne. □

Kokkuvõttes, kui mudel M defineerib lõplikul või loenduval hulgal $\mathcal{X}$ jaotuse P, siis defineerime $L(x) = -\log P(x)$ iga $x \in \mathcal{X}$ korral.

Mida teha juhul, kui $\mathcal{X} = \mathbb{R}$?

Sellisel juhul defineerib mudel M sellel hulgal tihedusfunktsiooni f. Osutub (põhjendust vt lisa A), et sel juhul on mõistlik valida iga $x \in \mathcal{X}$ korral

$$L(x) = -\log f(x).$$

Analoogiliselt olukorraga $x \in \mathcal{X}$ defineerime iga $x^n = (x_1, ..., x_n) \in \mathcal{X}^n$ jaoks $L(x^n) = -\log P(x^n)$, kui $\mathcal{X}$ on lõplik või loenduv, ning $L(x^n) = -\log f(x^n)$, kui $\mathcal{X} = \mathbb{R}$.

Nüüd oleme defineerinud esituspikkuse funktsiooni L ka pideva juhusliku suuruse jaoks. Lõpuks soovime nii diskreetse kui ka pideva olukorra jaoks kasutada ühtset tähistust. Tähistades tõepära funktsiooni

$$\mathcal{L}((x_1, ..., x_n)|M) = \begin{cases} P(x_1) \cdot P(x_2) \cdot ... \cdot P(x_n), & \text{kui } M \text{ defineerib diskreetse jaotuse,} \\ f(x_1) \cdot f(x_2) \cdot ... \cdot f(x_n), & \text{kui } M \text{ defineerib pideva jaotuse,} \end{cases}$$

ning eeldades x_i sõltumatust, kehtib

$$L((x_1, ..., x_n)|M) = -\log \mathcal{L}((x_1, ..., x_n)|M).$$

3 Raamistik järjestikuste segmentide tuvastamiseks

Idee kasutada MDL-printsiipi parima segmentatsiooni valikul saime artiklist [7], kus seda kasutati binaarsete andmete ning *k*-keskmiste klasterdamise korral. Üldistame seda ideed, et saaksime segmentidele sobitada väga erinevaid mudeleid, mille alusel leida parim segmentatsioon ning iga segmendi korral parim mudel.

Olgu andmetabel $n \times p$ maatriks D, mida soovime veergude järgi segmentideks jagada.

Tähistame $D(a, b) := \{(d_{ij}) \in D : a \leq j \leq b\}$ andmetabeli veerud a-st kuni b-ni.

Tuues võrdluseks meie probleemipüstituse bioloogilise tausta, siis p tähistab CpG saitide arvu ja n uuritavate objektide (inimeste, kudede vms) arvu. Olgu D veerud järjestatud nii, nagu vastavad CpG saidid DNA järjestusel. Selliselt on andmeid lihtne järjestada, kasutades DNA koordinaate (sest igal kromosoomil on nukleotiidid nummerdatud kasvavalt). Näiteks võib selline andmetabel D koosneda ühe kromosoomi kõigist metülatsioonivääртustest.

3.1 Segmentatsioon. Segmendiviisiline mudel.

Soovime oma andmestiku jagada veergude alusel segmentideks. Defineerime kõigepealt segmentatsiooni. Üldiselt võib segmentatsioon koosneda 1 kuni p segmendist. Olgu $k \in \{1, ..., p\}$ segmentide arv ning s_i, e_i iga $i \in \{1, \ldots, k\}$ korral sellised, et

$$1 = s_1 \leq e_1 < \underbrace{e_1 + 1}_{s_2} \leq e_2 < \underbrace{e_2 + 1}_{s_3} \leq \ldots < \underbrace{e_{k-1} + 1}_{s_k} \leq e_k = p. \tag{1}$$

Nii saame segmendid $[s_1, e_1], [s_2, e_2], ..., [s_k, e_k]$, kus s_i ja e_i tähistavad vastavalt i-nda segmendi algust ja lõppu.

Sellist segmentide kogumit $\{[s_1, e_1], \ldots, [s_k, e_k]\}$ nimetame segmentatsiooniks. Tähistame kõigi segmentatsioonide hulga sümboliga $\mathcal{S}$. Teisisõnu,

$$\mathcal{S} = \bigcup_{k=1}^{p} \left\{ \bigcup_{i=1}^{k} [s_i, e_i] : \ s_1, e_1, ..., s_k, e_k \text{ korral kehtib (1)} \right\}.$$

Olgu $\mathcal{M}_1, ..., \mathcal{M}_r$ mudelite klassid, mille abil soovime andmeid igal segmendil kirjeldada.

Näide. Näiteks $r = 2$ korral võib $\mathcal{M}_1$ koosneda kõigist lineaarsetest mudelitest, mis sisaldavad ainult vabaliiget, ning $\mathcal{M}_2$ sellistest lineaarsetest mudelitest, kus mudelis on lisaks meile huvi pakkuv tunnus, kas patsient on vähihaige. Teisisõnu, sel juhul

$$\mathcal{M}_1 = \{y = \beta_0 + \varepsilon : \ \beta_0 \in \mathbb{R}\},$$
$$\mathcal{M}_2 = \{y = \beta_0 + \beta_1 \cdot I_{\text{on\_vähihaige}} + \varepsilon : \ \beta_0, \beta_1 \in \mathbb{R}\}.$$

Segmenti $[s_i, e_i]$ soovime kirjeldada mõne konkreetse mudeliga M_i mudeliklassist $\mathcal{M}_1$ või $\mathcal{M}_2$, kus mudeli parameetrid oleksid hinnatud näiteks suurima tõepära meetodil. Seega, olenevalt andmetest D segmendil $[s_i, e_i]$, võiksime mudeliks $M_i \in \mathcal{M}_1$ saada näiteks $y = 0.5 + \varepsilon$ ning mudeliks $M_i \in \mathcal{M}_2$ näiteks $y = 0.25 + 0.6 \cdot I_{\text{on\_vähihaige}} + \varepsilon$. □

Defineerime andmeid $D = (d_{ij})$ kirjeldava segmendiviisilise mudeli M mingi fikseeritud segmentatsiooni $S \in \mathcal{S}$ korral järgnevalt

$$M = \sum_{i=1}^{|S|} M_i I_{[s_i, e_i]}\,, \tag{2}$$

kus mudelid M_i kuuluvad mõnda mudeliklassidest $\mathcal{M}_1, ..., \mathcal{M}_r$. Siin I tähistab indikaatorfunktsiooni. Selline segmendiviisilise mudeli definitsioon tähendab, et iga segmenti $[s_i, e_i]$ kirjeldatakse täpselt ühe mudeliga M_i, mis sobitatakse sellele segmendile ette antud mudelite klasside $\mathcal{M}_1, ..., \mathcal{M}_r$ seast.

Ilmselt sisaldab selliselt defineeritud mudel M informatsiooni nii segmentatsiooni kui ka igale segmendile sobitatud mudeli M_i kohta.

Tähistame sümboliga $\mathcal{M}$ kõigi selliste mudelite hulga (üle kõigi segmentatsioonide ning mudelite M_i kõikvõimalike valikute mudeliklasside $\mathcal{M}_1, ..., \mathcal{M}_r$ seast). Teisisõnu,

$$\mathcal{M} = \left\{ \sum_{i=1}^{|S|} M_i I_{[s_i, e_i]} : \ S \in \mathcal{S}, \ M_i \in \bigcup_{j=1}^{r} \mathcal{M}_j \right\}. \tag{3}$$

3.2 MDLi mõttes parima segmentatsiooni leidmine

Kuna segmentatsioon on mudeli $M \in \mathcal{M}$ üks komponent, siis määrab mudeli valik üheselt segmentatsiooni. Seega oleme segmendiviisiliste mudelite defineerimise abil taandanud optimaalse segmentatsiooni leidmise ülesande parima mudeli valiku ülesandele: valida hulgast $\mathcal{M}$ andmeid D kõige paremini kirjeldav mudel M. Optimaalse mudeli

valikul toetume MDL-printsiibile.

MDLi mõttes optimaalse segmentatsiooni leidmiseks piisab iga $M \in \mathcal{M}$ korral arvutada $L(M) + L(D|M)$ ning valida M, mille korral see summa minimeeritakse. Selline mudel määrabki parima segmentatsiooni.

Leiame avaldise $L(M) + L(D|M)$ täpse kuju segmendiviisilise mudeli (2) jaoks.

Fikseeritud segmentatsiooni S korral avaldub eelnevalt defineeritud mudeli (2) esituspikkus kujul

$$L(M) = |S| \cdot \log p + |S| \cdot \log r + \sum_{i=1}^{|S|} L(\boldsymbol{\theta}_i), \text{ kus}$$

- $|S| \cdot \log p$ näitab segmentatsiooni esitamiseks kuluvate bittide arvu (sest selleks piisab teada kõigi segmentide alguspunkte ning neid on kokku $|S|$ tükki),
- $|S| \cdot \log r$ on bittide arv, mis kulub näitamaks, milline mudelite klass $\mathcal{M}_1, ..., \mathcal{M}_r$ osutus valituks igal segmendil,
- $L(\boldsymbol{\theta}_i)$ näitab i-ndal segmendil valituks osutunud mudeli M_i parameetrite $\boldsymbol{\theta}_i$ kodeerimiseks kuluvate bittide arvu. Tema täpne väärtus sõltub, mitut parameetrit $\boldsymbol{\theta}_i$ sisaldab ning kas need on täisarvulised või reaalarvulised. Reaalarvulisi parameetreid kodeeritakse tavaliselt (vt [5]) täpsusega $1/\sqrt{\tilde{n}}$, kus $\tilde{n}$ näitab vaatluste arvu, mille põhjal selle parameetri väärtus on hinnatud. Seega edasises eeldame, et reaalarvulise parameetri kodeerimiseks kulub $-\log \frac{1}{\sqrt{\tilde{n}}}$ ehk $\frac{1}{2} \log \tilde{n}$ bitti.

Paneme tähele, et kogu mudeli M esituspikkus avaldub mudelite $M_i I_{[s_i,e_i]}$ esituspikkuste summana, sest

$$L(M) = \sum_{i=1}^{|S|} (\log p + \log r + L(\boldsymbol{\theta}_i)) \tag{4}$$

$$= \sum_{i=1}^{|S|} L(M_i I_{[s_i,e_i]}). \tag{5}$$

Eeldame vaatluste D sõltumatust. Lisaks eeldame, et iga mudel M_i määrab mingi tõenäosusjaotuse (kas diskreetse või pideva). Tähistame andmete D tõepära fikseeritud mudeli M korral $\mathcal{L}(D|M)$. Arvestades tulemusi tõenäosusliku mudeli kodeerimise kohta (vt peatükk 2.1), avaldub andmete D esituspikkus, tingimusel et neid kodeeritakse mudeli

M järgi,

$$L(D|M) = -\log \mathcal{L}(D|M) = -\log \prod_{i=1}^{|S|} \mathcal{L}(D(e_i, s_i)|M) = -\sum_{i=1}^{|S|} \log \mathcal{L}(D(e_i, s_i)|M).$$

Arvestades, et andmete $D(e_i, s_i)$ kirjeldamiseks läheb mudelist $M = \sum_i M_i I_{[s_i,e_i]}$ vaja ainult liidetavat $M_i I_{[s_i,e_i]}$, siis

$$L(D|M) = -\sum_{i=1}^{|S|} \log \mathcal{L}(D(e_i, s_i)|M_i). \tag{6}$$

Näeme, et segmendi kaupa defineeritud mudelite esituspikkuse funktsioon $L(M)+L(D|M)$ on aditiivne, sest valemite (5) ja (6) põhjal kehtib

$$L(M) + L(D|M) = \sum_{i=1}^{|S|} \left(L(M_i I_{[s_i,e_i]}) + L(D(e_i, s_i)|M_i) \right).$$

Seega, fikseeritud segmentatsiooni korral saame kogu mudeli ning kõigi andmete esituspikkuse arvutada igal segmendil eraldi ning lõpuks tulemused kokku summeerida.

Arvutamegi vastavad esituspikkused igal segmendil kõikvõimalike segmentatsioonide jaoks ning esitame need arvud ülemise kolmnurkmaatriksi $A = (a_{ij})$ elementidena, kus element a_{ij} näitaks segmendile $[i, j]$ sobitatud mudelitest parima mudeli esituspikkust. Teisisõnu, olgu $p \times p$ maatriks A selline, et iga $j \geq i$ korral

$$a_{ij} := \min \left\{ L(M_{ij}) + L(D(i, j)|M_{ij}) : M_{ij} \in \mathcal{M}_h,\ h \in \{1, ..., r\} \right\},\ \text{kus}$$

- indeks ij näitab, et vaatleme segmenti $[i, j]$,
- $M_{ij} \in \mathcal{M}_h$ näitab andmetele $D(i, j)$ sobitatud mudelit h-ndast mudelite klassist.

Lugemaks maatriksist A välja andmete optimaalset segmentatsiooni, kasutame dünaamilise programmeerimise algoritmi [8]. Selle algoritmi idee põhineb tähelepanekul, et mingisuguse fikseeritud viimase segmendi $[s_{|S|}, e_{|S|}]$ korral on parim segmentatsioon selline, mille korral minimeeritakse andmete ja mudeli esituspikkus lõigul $[1, e_{|S|-1}]$. Järgneb selle algoritmi pseudokood.

Pseudokood 1 Dünaamilise programmeerimise algoritm optimaalse segmentatsiooni leidmiseks

Sisend: esituspikkuste maatriks $A = (a_{ij})$
$L_0 \leftarrow 0$
for all $j \in \{1, ..., p\}$ **do**
    $L_j \leftarrow +\infty$
    for all $i \in \{1, ..., j\}$ **do**
        if $L_j > L_{i-1} + a_{ij}$ **then**
            $I_j \leftarrow i - 1$
            $L_j \leftarrow L_{i-1} + a_{ij}$
        end if
    end for
end for
▷ Parima segmentatsiooni taastamine
$k \leftarrow p$
$\mathcal{S} \leftarrow \emptyset$
while $k > 0$ **do**
    $\mathcal{S} \leftarrow \{[I_k + 1, k], \mathcal{S}\}$
    $k \leftarrow I_k$
end while
return $\mathcal{S}$

Näeme, et pseudokoodis 1 arvutatakse iteratiivselt iga $j \in \{1, ..., p\}$ korral

$$L_j = \min \{L_{i-1} + a_{ij} : i \in \{1, ..., j\}\}$$

ning vastavate minimeerivate indeksite I_j abil leiamegi optimaalse segmentatsiooni $\mathcal{S}$.

3.3 Meetodi arvutusmahukus

Eelmises alapeatükis kirjeldasime, kuidas implementeerida optimaalse segmentatsiooni leidmist. Võib tekkida küsimus, kas kõikvõimalike segmentatsioonide läbivaatamine üldse on võimalik või on nende arv liiga suur.

Kui igale segmendile plaanime sobitada täpselt ühe mudeli (näiteks kui usume, et andmeid saab kirjeldada ainult üks mudelite klass $\mathcal{M}_1$), siis on kõigi mudelite hulga (3) elementide arv võrdne kõigi segmentatsioonide arvuga, milleks on $1+\cdots+p$ ehk $\frac{p(p+1)}{2}$. Kui soovime segmente kirjeldada mudelitega r erinevast mudelite klassist $\mathcal{M}_1, ..., \mathcal{M}_r$, siis on andmetele sobitatavate mudelite arv segmentatsioonide koguarvust r korda suurem.

On selge, et piisavalt suure andmetabeli veergude arvu p korral läheb kõigi segmentatsioonide arv nii suureks, et kõigi variantide läbivaatus on arvutuslikult liiga mahukas ülesanne.

Sellisel juhul teeme eelduse, et leidub mingi arv m, mida ühegi segmendi pikkus ei saa ületada. Selline eeldus on bioloogiliselt põhjendatud, kui usume, et kõik meie andmestikus olevad CpG saidid, mis asuvad teineteisest enam kui m kaugusel, asuvad teineteisest nii kaugel, et nende regulatsioon ei saa olla sarnane. Seega piisab vaadelda ainult selliseid segmente, mille pikkus ei ületa arvu m. Kõigi selliste segmentide arvu leidmiseks tuletasime arvutuseeskirja, vt lisa B.

Näiteks metülatsioonikiibi andmete korral osutus mõistlikuks kasutada $m = 50$. Kõigi mudelite hulgaks on $m = 50$ korral

$$\mathcal{M} = \left\{ \sum_{i=1}^{|S|} M_i I_{[s_i, e_i]} : \ e_i - s_i < 50, \ S \in \mathcal{S}, \ M_i \in \bigcup_{j=1}^{r} \mathcal{M}_j \right\}. \tag{7}$$

Sellisel juhul on segmentidele sobitatavate mudelite arv

$$\begin{cases} \frac{p(p+1)}{2} r & \text{kui } p \leq 50, \\ \left(50p - \frac{50(50-1)}{2}\right) r & \text{kui } p > 50. \end{cases}$$

Näeme, et selline eeldus vähendab suure n korral oluliselt segmentide arvu, millele sobitame mudeleid, ning teeb võimalikuks meie meetodi kasutamise. Näiteks $p = 1000$ ning $r = 2$ korral oleks kõikvõimalike mudelite arv 1001000, aga meie lisaeeldust rahuldavate mudelite (7) arv 98775, mis on ligikaudu 10 korda väiksem.

4 *K*-keskmiste klasterdamisel põhinev meetod

Soovime kasutada peatükis 3 välja töötatud raamistikku nüüd ühe konkreetse meetodi kirjeldamiseks.

Metülatsiooniandmete visuaalsel vaatlusel (vt näiteks joonis 2) selgub, et beeta-väärtused on teatavas mõttes plokkstruktuuriga. See tähendab, et näeme ühesugust metülatsioonimustrit mitmetel CpG-del järjest (seejuures on beeta-väärtused ligilähedaselt konstantsed) ning lisaks on sellistel segmenditel kas kõigi CpG-de metülatsioon sarnane või eristuvad selgelt mõned grupid, kus beeta-väärtused on ligilähedaselt võrdsed.

Soovimegi igal segmendil kasutada sellist mudelit, mis klasterdaks sarnaseid gruppe. Et iga klastri korral saaksime arvutada andmete nägemise tõepära, teeme olulise lihtsustava eelduse, et andmed on binaarsed. Seega on vaja selle meetodi kasutamiseks binariseerida beeta-väärtused. Nende jaotus on bimodaalne ning binariseerimisel tundub mõistlik võtta piiriks 0.5.

4.1 Meetodi kirjeldus

Vaatleme ainult i-ndale segmendile $[s_i, e_i]$ vastavat $n \times l_i$ andmetabelit $D(s_i, e_i) = (d_{jh})$, kus l_i tähistab segmendi pikkust. Lihtsuse huvides jätame siin peatükis edaspidi tähistuste juures indeksi i kirjutamata (peame lihtsalt meeles, et kõik hinnatud parameetrid käivad i-nda segmendi kohta). Eeldame, et $d_{jh} \in \{0, 1\}$.

Eeldame, et vaatlused j-ndas reas on Bernoulli jaotusest, mille parameetri hinnanguks võtame selle segmendi j-nda rea elementide keskmise. Teisisõnu, eeldame, et iga $j \in \{1, ..., n\}$ korral

$$d_{js_i}, ..., d_{je_i} \leftarrow Be(p_j), \text{ kusjuures } \hat{p_j} = \frac{1}{l_i} \sum_{h=s_i}^{e_i} d_{jh}.$$

Nüüd võtame võimalikeks mudelite klassideks $\mathcal{M}_1, ..., \mathcal{M}_{10}$, kus iga $k \in \{1, ..., 10\}$ korral on $\mathcal{M}_k$ selline mudel, kus oleme rakendanud arvudele $\{\hat{p_j} : j \in \{1, ..., n\}\}$ k-keskmiste (*k-means*) klasterdamist. K-keskmiste klasterdamise eesmärgiks on leida sellised arvud $\theta_1, ..., \theta_k$, mida nimetame klastrikeskmisteks, nii, et minimeerida kauguste

ruutude summa

$$\sum_{j=1}^{n}(\hat{p_j}-\theta_{z_j})^2=\sum_{j\in\mathcal{I}_1}(\hat{p_j}-\theta_1)^2+...+\sum_{j\in\mathcal{I}_k}(\hat{p_j}-\theta_k)^2,\text{ kus}$$

- z_j näitab arvule $\hat{p_j}$ vastava klastri järjekorranumbrit,
- indeksite hulgad $\mathcal{I}_1,...,\mathcal{I}_k$ koosnevad vastavatesse klastritesse kuuluvate arvude $\hat{p_j}$ indeksitest.

Selle tulemusena oleme iga arvu $\hat{p}_1,...,\hat{p}_n$ liigitanud ühte klastrisse ehk liigitanud iga rea $j\in\{1,...,n\}$ ühte klastrisse. Olgu klasterdamise tulemusena saadud klastrikeskmised $\theta_1,...,\theta_k$. Tähistades j reale vastava klastri z_j, oleme j-ndale reale seadnud vastavusse klastrikeskmise θ_{z_j}. Saadud mudeli kohaselt on j-nda rea vaatlused jaotusest $Be(\theta_{z_j})$.

Näide. Kirjeldatud klasterdamisest on näide tabelis 1. Seal on kõigepealt toodud binaarne näiteandmestik, seejärel arvutatud reakeskmised, millele on rakendatud k-keskmiste klasterdamist $k=2$ korral. On esitatud nii tulemuseks saadud klastrid kui ka klastrikeskmised. Seega selle näite tulemusena saaksime mudeli $\mathcal{M}_2$, mille kohaselt oleksid andmestiku read $1,2,4$ jaotusest $Be(0.1333)$ ning read $3,5,6,7$ jaotusest $Be(0.85)$. □

Tabel 1. Näide klasterdamisest $k=2$ korral ühel 7×5 andmestikul.

					reakeskmine	klaster	klastrikeskmine
0	1	0	0	0	0.2	1	0.1333
0	0	0	0	0	0	1	0.1333
1	1	1	1	0	0.8	2	0.85
0	0	0	1	0	0.2	1	0.1333
1	1	1	1	1	1	2	0.85
0	1	1	1	1	0.8	2	0.85
1	1	1	0	1	0.8	2	0.85

Nüüd, arvestades eeldust Bernoulli jaotuse kohta, saame arvutada andmete tõepära mudeli $M_i\in\mathcal{M}_k$ jaoks:

$$\begin{aligned}\mathcal{L}(D(s_i,e_i)|M_i)&=\prod_{j=1}^{n}\prod_{h=s_i}^{e_i}\theta_{z_j}^{d_{jh}}\left(1-\theta_{z_j}\right)^{1-d_{jh}}=\prod_{j=1}^{n}\theta_{z_j}^{\sum_h d_{jh}}\left(1-\theta_{z_j}\right)^{\sum_h(1-d_{jh})}\\ -\log\mathcal{L}(D(s_i,e_i)|M_i)&=-\sum_{j=1}^{n}\log\left(\theta_{z_j}^{\sum_{h=s_i}^{e_i}d_{jh}}\left(1-\theta_{z_j}\right)^{\sum_{h=s_i}^{e_i}(1-d_{jh})}\right)\\ &=-\sum_{j=1}^{n}\left(\log\theta_{z_j}\sum_{h=s_i}^{e_i}d_{jh}+\log(1-\theta_{z_j})\sum_{h=s_i}^{e_i}(1-d_{jh})\right).\end{aligned}$$

Kasutamaks peatüki 3.2 valemit (4), on vaja veel leida $L(\boldsymbol{\theta}_i)$ ehk tuleb leida, milliseid mudeli parameetreid on i-ndal segmendil vaja kodeerida, eeldusel, et seal osutus valituks mudel $\mathcal{M}_{k_i}$. Nendeks on:

- reaalarvud $\theta_1, ..., \theta_{k_i}$, mille kodeerimiseks kulub $k_i \cdot \gamma$ bitti, kus γ näitab ühe reaalarvu kodeerimiseks kuluvate bittide arvu,
- iga rea $j \in \{1, ..., n\}$ kohta arv $\{1, ..., k_i\}$ näitamaks, millisesse klastrisse see rida kuulub. Selleks kulub $n \cdot \log k_i$ bitti.

Tähistades andmepunktide koguarvu vaadeldaval segmendil $\tilde{n} := n \cdot (e_i - s_i + 1)$, valime reaalarvulise parameetri kodeerimise täpsuseks $\gamma := 0.5 \log \tilde{n}$.

Kokku saame:

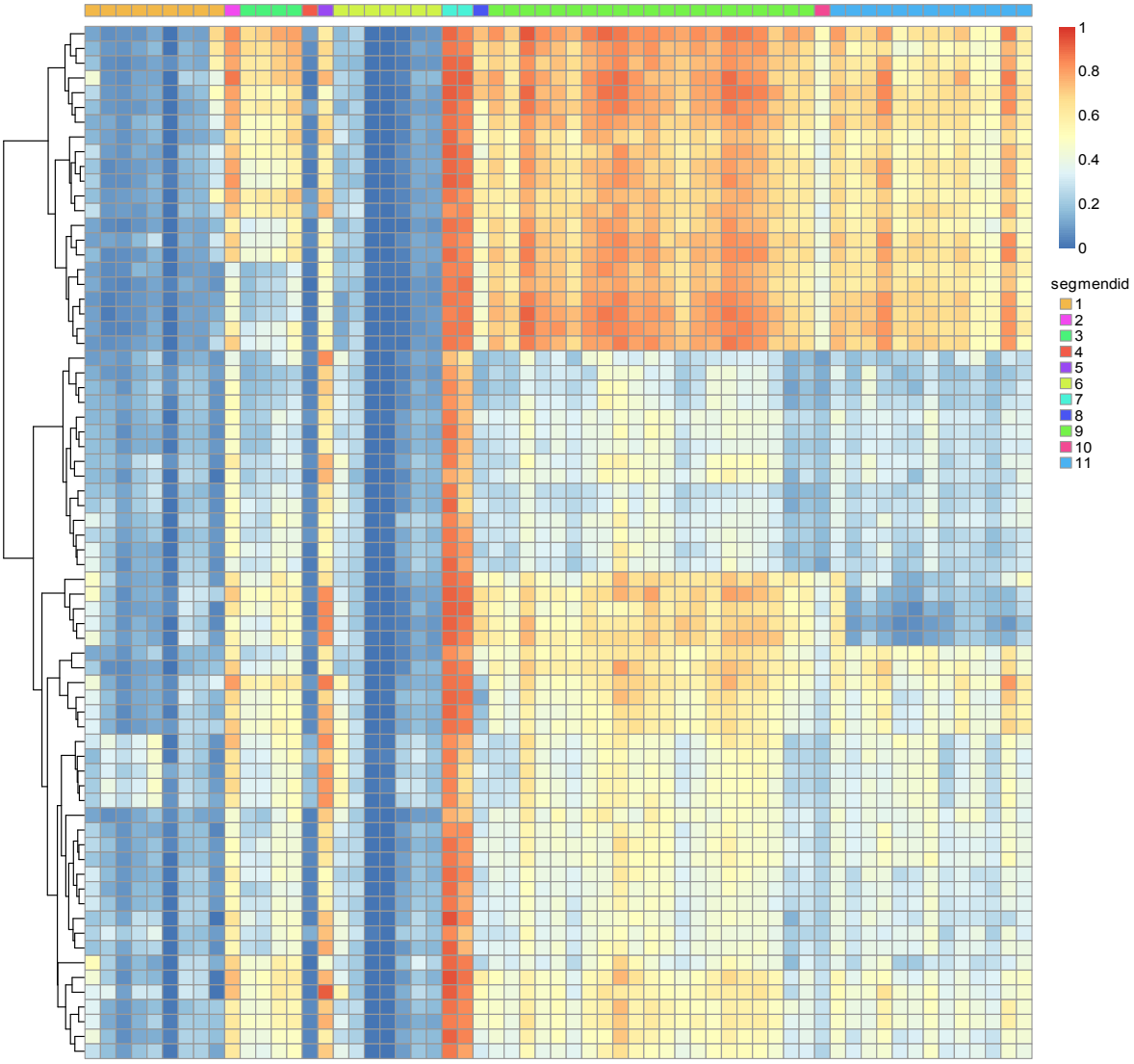
$$L(M) + L(D|M) = \sum_{i=1}^{|S|} (\log p + \log 10 + n \log k_i + 0.5 k_i \log \tilde{n}) + \\ - \sum_{i=1}^{|S|} \sum_{j=1}^{n} \left(\log \theta_{z_j} \sum_{h=s_i}^{e_i} d_{jh} + \log(1 - \theta_{z_j}) \sum_{h=s_i}^{e_i} (1 - d_{jh}) \right)$$

Nüüd saame kasutada kirjeldatud meetodit optimaalse segmentatsiooni leidmiseks. Lisaks saame iga segmendi kohta teada optimaalse klastrite arvu.

4.2 Rakendamine bioloogilistel andmetel

Saadud meetodit rakendasime peatükis 1 mainitud metülatsiooniandmetel 4 doonori 17 koe kohta. Tulemusena jagas meetod andmestiku 54995 regiooniks. Diferentsiaalselt metüleeritud regioonideks otsustasime nende seast nimetada selliseid, mille pikkus oli vähemalt 3 CpG-d ning kus klastrite arv oli vähemalt 2. Nende regioonide koguarv oli 1315.

Joonisel 2 on näide ühele DNA lõigule vastavatest andmetest ning ülaosas on näha k-keskmiste klasterdamise meetodiga leitud optimaalne segmentatsioon. Selle näite põhjal tundub, et meetodi tulemused on mõistlikud.



Joonis 2. Näide algoritmi tulemustest (kromosoom 2, genoomi koordinaadid 176708867 kuni 176725876). Iga veerg näitab ühte CpG saiti, iga rida ühe patsiendi ühte kude. Ülal on näha optimaalne segmentatsioon. Näiteks kõige parempoolsemal segmendil (nr 11) tuvastati 3 klastrit ning segmendil (nr 9) 5 klastrit.

5 Lineaarsel mudelil põhinev meetod

Eelnev meetod kasutas segmentidel juhendamata õppimist: klasterdamist. Tegelikkuses aga teame tihti, milliste gruppide vahel andmetes erinevusi otsime (näiteks vähihaigete ja tervete patsientide vahel). Tavapäraseks võtteks metülatsiooniandmete analüüsimisel on rakendada t-testi igal CpG saidil eraldi. Mitmetest CpG-dest koosneval segmendil oleks selle üldistuseks lineaarse mudeli kasutamine. Lisaks kahe (või enama) grupi keskmiste erinevuse arvesse võtmisele saaksime sel juhul mudelisse lisada ka teiste tunnuste mõjusid.

Käesolevas peatükis kasutame peatükis 3 kirjeldatud raamistikku meetodi jaoks, kus segmentidele sobitatavateks mudeliteks võtame lineaarsed mudelid.

5.1 Meetodi kirjeldus

Vaatleme i-ndale segmendile $[s_i, e_i]$ vastavat andmetabelit $D(s_i, e_i)$.

Soovime sellel segmendil kasutada mõnda lineaarset mudelit. Täpsemalt, kui andmestiku read jagunevad mingi faktortunnuse alusel k erinevasse gruppi, siis kõigepealt soovime teada, kas lineaarses mudelis üldse peaks see tunnus sees olema. Kui jah, siis võib meid hiljem huvitada mõne selle faktortunnuse kontrasti jaoks statistilise hüpoteesi testimine.

Seega on mõttekas kahe lineaarse mudeli kasutamine. Otsustame, et neis mudelites sisaldub kindlasti veergude mõju. Olgu esimese (nn lihtsama) mudeli mudelimaatriks X_1 selline, et seal sisaldub ainult veergude $s_i, ..., e_i$ mõju ning teise (nn keerulisema) mudeli mudelimaatriks X_2 selline, et seal sisaldub nii veergude mõju kui ka gruppide mõju. Saame kaks mudelite klassi:

$$\mathcal{M}_1 : \mathbf{y} = X_1\boldsymbol{\beta}_1 + \boldsymbol{\varepsilon} \text{ (lihtsam mudel)},$$
$$\mathcal{M}_2 : \mathbf{y} = X_2\boldsymbol{\beta}_2 + \boldsymbol{\varepsilon} \text{ (keerulisem mudel)},$$

kus kummagi mudeli puhul eeldame, et $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma^2 I)$.

See tähendab, et esimese mudeli prognoosiks on iga veeru korral üks arv, mis üldiselt sõltub veerust. Teise mudeli prognoos võtab lisaks veerule arvesse ka seda, millise grupiga parasjagu tegu on. Näiteks joonisel 3 on näide olukorrast, kus veergudel $1 - 10$ ja $21 - 30$ sooviksime kasutada lihtsamat, aga veergudel $11 - 20$ keerukamat mudelit gruppide A ja B jaoks.

Mõistagi võib mudel $\mathcal{M}_2$ sisaldada ka teisi tunnuseid, mida soovime mudelisse kaasata (näiteks pideva tunnusena patsientide vanust vms).

Saamaks teada, mitu reaalarvulist parameetrit on kummagi mudeli jaoks tarvis kodeerida, leiame vastavate mudelimaatriksite astakud: need näitavad vähimat vaja minevate parameetrite arvu. Mudeli $\mathcal{M}_1$ korral on mudelimaatriksi X_1 astak võrdne veergude arvuga ehk $rank(X_1) = e_i - s_i + 1$. Mudeli $\mathcal{M}_2$ jaoks $rank(X_2) = e_i - s_i + k$, kus k näitab gruppide arvu.

Lihtsuse mõttes tähistame edasises, sõltuvalt sellest, kumba mudelitest $\mathcal{M}_1$ või $\mathcal{M}_2$ parasjagu silmas peame, selle mudeli mudelimaatriksit X, parameetrite hinnangute vektorit $\hat{\boldsymbol{\beta}}$ ning vastavat prognooside vektorit $\hat{\mathbf{y}} = X\hat{\boldsymbol{\beta}}$. Tähistame vaatluste koguarvu vaadeldaval segmendil $\tilde{n} := n \cdot (e_i - s_i + 1)$ ning valime reaalarvulise parameetri kodeerimise täpsuseks $\gamma := 0.5 \log \tilde{n}$.

Nüüd, peatüki 3.2 valemite (4) ja (6) kohaselt

$$\begin{aligned} L(M) + L(D|M) &= \sum_{i=1}^{|S|} (\log p + \log r + L(\boldsymbol{\theta}_i)) - \sum_{i=1}^{|S|} \log \mathcal{L}(D(e_i, s_i)|M_i) \\ &= \sum_{i=1}^{|S|} (\log p + \log 2 + \gamma \cdot rank(X)) - \sum_{i=1}^{|S|} \log \mathcal{L}(D(e_i, s_i)|M_i), \end{aligned}$$

kus γ tähistab reaalarvu kodeerimiseks kuluvate bittide arvu ning, arvestades tehtud eeldust mudelite jääkide jaotuse kohta,

$$\mathcal{L}(D(e_i, s_i)|M_i) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^{\tilde{n}} \exp\left(-\frac{1}{2\sigma^2}(\mathbf{y} - X\boldsymbol{\beta})^T(\mathbf{y} - X\boldsymbol{\beta})\right).$$

Võttes $\boldsymbol{\beta}$ rolli tema hinnangu $\hat{\boldsymbol{\beta}}$ ning σ^2 rolli tema nihketa hinnangu

$$\hat{\sigma}^2 = \frac{(\mathbf{y} - \hat{\mathbf{y}})^T(\mathbf{y} - \hat{\mathbf{y}})}{\tilde{n} - rank(X)} =: \frac{RSS}{\tilde{n} - rank(X)},$$

saame

$$\begin{aligned} -\log \mathcal{L}(D(e_i, s_i)|M_i) &= \frac{\tilde{n}}{2}\log(2\pi) + \frac{\tilde{n}}{2}\log(\hat{\sigma}^2) + \frac{1}{2\hat{\sigma}^2 \ln 2} \cdot \underbrace{(\mathbf{y} - \hat{\mathbf{y}})^T(\mathbf{y} - \hat{\mathbf{y}})}_{RSS} \\ &= \frac{\tilde{n}}{2}\log(2\pi) + \frac{\tilde{n}}{2}\log(RSS) - \frac{\tilde{n}}{2}\log(\tilde{n} - rank(X)) + \frac{\tilde{n} - rank(X)}{2\ln 2}. \end{aligned}$$

Kokku saame

$$L(M) + L(D|M) = \sum_{i=1}^{|S|} (\log p + \log 2 + 0.5 \cdot \tilde{n_i} \cdot rank(X)) + \\ + \sum_{i=1}^{|S|} \left(\frac{\tilde{n_i}}{2} \log \left(\frac{2\pi \cdot RSS}{\tilde{n_i} - rank(X)} \right) + \frac{\tilde{n_i} - rank(X)}{2 \ln 2} \right).$$

Nüüd saame kasutada saadud meetodit optimaalse segmentatsiooni leidmiseks. Lisaks saame iga segmendi kohta teada, kumb lineaarne mudel sellel segmendil valituks osutus. Kui mudel $\mathcal{M}_2$, siis saame testida selle mudeli kohta statistilisi hüpoteese, näiteks, kas kahe grupi keskmised erinevad oluliselt. Kuna peatükis 3.3 leidsime andmetele sobitatavate mudelite koguarvu, kasutame mitmese testimise probleemi lahendamiseks siinkohal Bonferroni korrektsiooni.

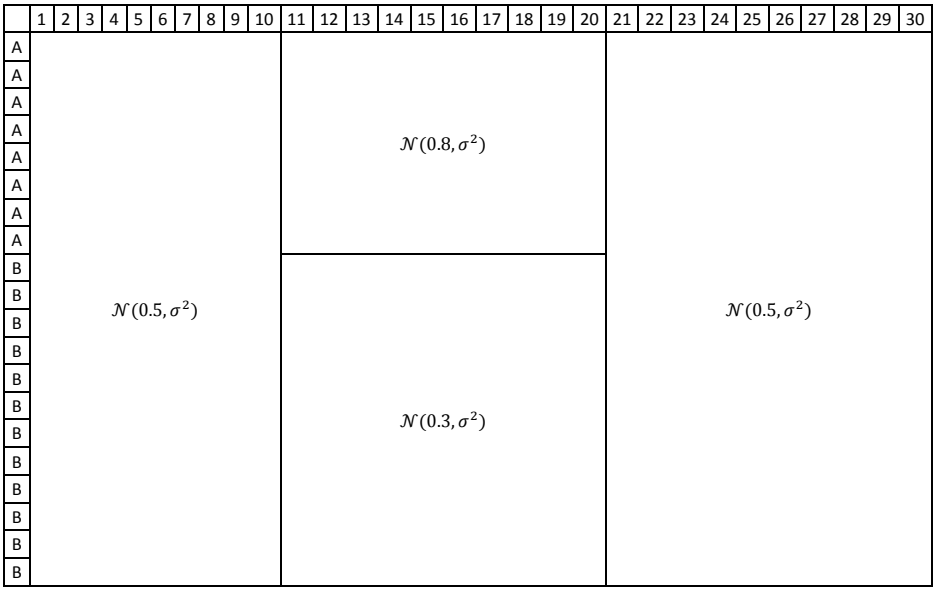
5.2 Algoritmi testimine

Järgnevas veendume, et meie algoritm töötab andmetel nii, nagu soovitud. Kui teaksime bioloogiliste metülatsiooniandmete kohta tõde (näiteks, et milliste CpG saitide korral tegelikult gruppide keskväärtused erinevad), saaksime mõistagi oma algoritmi headust hinnata neil andmetel. Paraku pole sellist tõde teada. Seega genereerime esmalt teatud eeskirja kohaselt andmeid ning uurime, millistel juhtudel saame oodatud tulemuse, millal mitte.

Samuti soovime oma algoritmi võrrelda mõne teadaoleva alternatiiviga. Üheks suhteliselt lihtsaks alternatiiviks oleks teha igal veerul t-test, kasutada mitmese testimise korrektsiooni ning hiljem teatava kriteeriumi alusel grupeerida järjestikused veerud, kus keskmiste erinevus osutus statistiliselt oluliseks, pikemateks segmentideks. Kuna kasutasime Bonferroni korrektsiooni oma meetodi jaoks, kasutame sedasama ka võrreldava t-testi jaoks.

5.2.1 Lihtsa skeemi järgi genereeritud andmed

Olgu meil vaatluse all olevaid objekte 20 ning jagunegu nad mingi tunnuse alusel gruppidesse A ja B, vastavalt 8 ja 12 objekti. Paigutame need objektid andmetabeli ridadesse. Lisaks olgu andmetabeli veergudes (järjestatult) erinevate tunnuste väärtused. Tähistame i-nda objekti j-nda tunnuse väärtust y_{ij}.



Joonis 3. Andmete genereerimise skeem.

Kõigepealt vaatleme joonisel 3 kujutatud olukorda, kus andmed esituvad mingi fikseeritud arvu σ korral järgneva mudeli kohaselt:

$$y_{ij} = 0.5 + \alpha_{ij} + \varepsilon_{ij}, \text{ kus } \varepsilon_{ij} \sim \mathcal{N}(0, \sigma^2) \text{ ja } \alpha_{ij} = \begin{cases} 0, & \text{kui } j \in \{1, .., 10, 21, ..., 30\}, \\ 0.3, & \text{kui } i \in \{1, ..., 8\}, j \in \{11, ..., 20\}, \\ -0.2, & \text{kui } i \in \{9, ..., 20\}, j \in \{11, ..., 20\}. \end{cases}$$

Nii oleme tekitanud olukorra, kus veergudes $11 - 20$ on gruppide A ja B keskväärtused erinevad, mujal võrdsed. Seejuures oleme α_{ij} valinud nii, et iga veeru elementide aritmeetilise keskmise keskväärtus oleks 0.5.

Ootaksime, et algoritm jagaks need 30 veergu kolmeks segmendiks $[1, 10]$, $[11, 20]$, $[21, 30]$, kusjuures segmendi $[11, 20]$ korral peaks osutuma parimaks selline mudel, kus on gruppide A ja B keskmiste erinevust näitav kordaja (nn keerulisem mudel). Teiste segmentide korral peaks osutuma valituks lihtsam mudel.

Genereerisime andmed kirjeldatud viisil erinevate σ väärtuste jaoks ning kordasime seda 50 korda. Simulatsiooni tulemused kahe σ väärtuse korral on näha joonisel 4.

Selgitus selle joonise interpreteerimiseks:

- Vasakpoolses tulbas on näha tulemused $\sigma = 0.4$ korral, parempoolses $\sigma = 0.7$

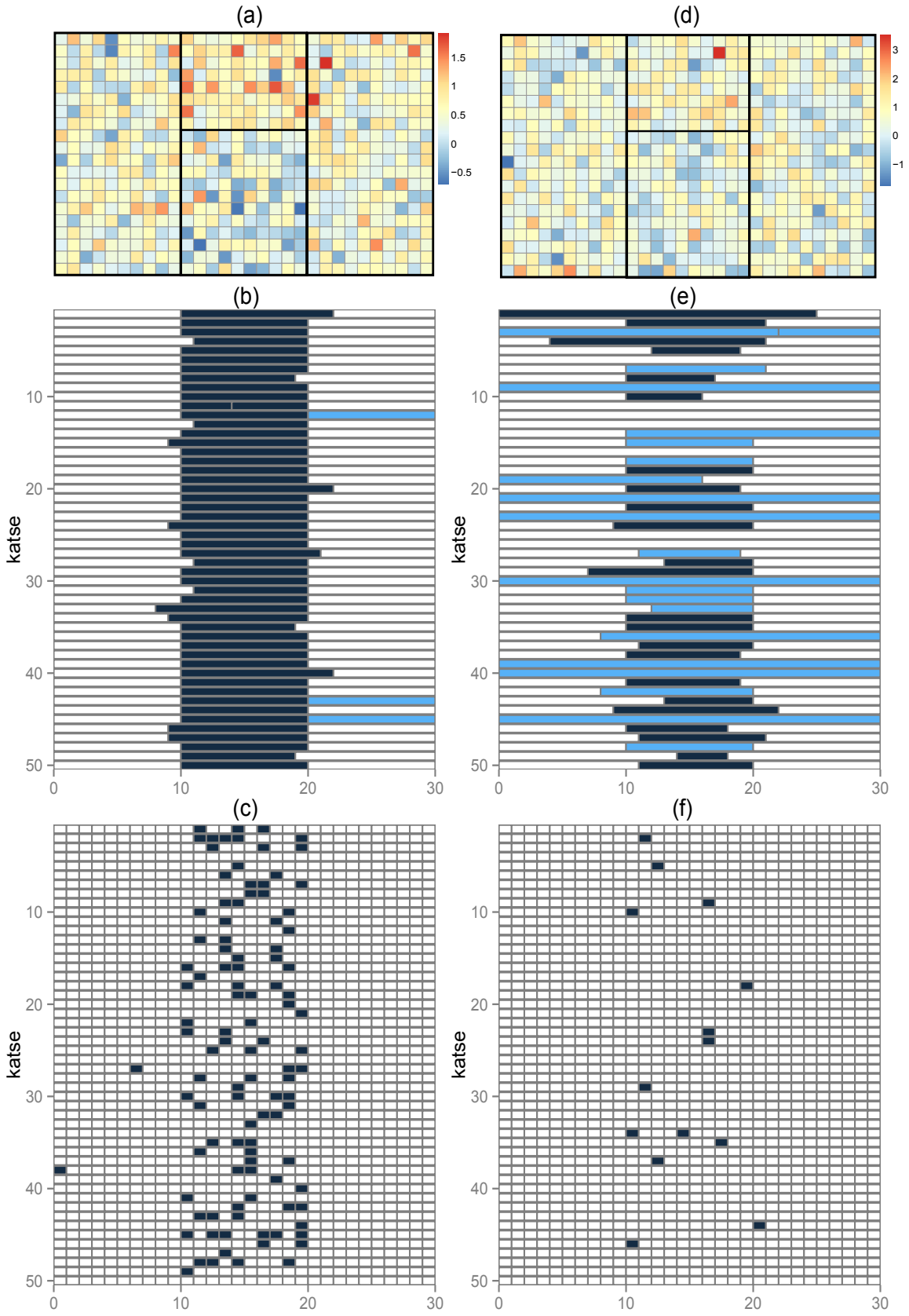
korral.

- Joonistel (a) ning (d) on kummagi σ väärtuse jaoks üks näide genereeritud andmestikust, andmaks lugejale intuitsiooni, kui "suur" või "väike" on kahe grupi keskmiste erinevus veergudes $11, ..., 20$.
- Joonistel (b) ning (e) on näha meie algoritmi leitud optimaalsed segmentatsioonid 50 erineval genereeritud andmestikul. Joonistel (c) ning (f) on näha nendesamade 50 andmestiku igal veerul tehtud t-testi tulemused, tuvastamaks kahe grupi keskväärtuste erinevust.
- Veergude numbrid x-teljel ühtivad joonise 3 tähistusega, samuti on jooniste (a), (b), (c) ning (d), (e), (f) vastavad veerud paigutatud kohakuti.
- Segmendi värvi interpretatsioon (jooniste (b) ning (e) jaoks):
  - Valget värvi segment tähendab, et sellel segmendil osutus valituks lihtsam mudel.
  - Helesinine värv näitab, et kuigi valituks osutus keerulisem mudel, siis gruppide A ja B erinevus sellel segmendil oli statistiliselt ebaoluline.
  - Tumesinine värv näitab, et valituks osutus keerulisem mudel ning gruppide A ja B erinevus sellel segmendil oli statistiliselt oluline.
- t-testi tulemuse värvi interpretatsioon (jooniste (c) ning (f) jaoks):
  - Tumesinine värv mingis veerus näitab, et selles veerus oli gruppide keskmiste erinevus statistiliselt oluline, kasutades Bonferroni korrektsiooni.

Jooniselt näeme, et kui $\sigma = 0.4$, siis t-test ei tuvasta paljudes veergudes keskmiste erinevust. Samas meie meetod töötab päris hästi, sest segmentatsioonid on enam-vähem õigesti leitud ning alati on valitud keskmisele segmendile õige mudel, kusjuures keskmiste erinevus neil segmentidel oli alati statistiliselt oluline. Mõnel üksikul juhul on meetod valinud keerulisema mudeli ka valele segmendile, aga nende mudelite puhul ei osutunud keskmiste erinevus oluliseks.

Olukorras $\sigma = 0.7$ leiab t-test keskmiste erinevuse üles ainult väga üksikutel juhtudel. Ka meie meetodi tulemused ei ole nii head kui väiksema σ korral, aga siiski silmanähtavalt paremad kui t-testi omad. Lisaks, näeme, et valepositiivseid segmente peaaegu ei leidu. Paljudel katsetel on kõik 30 veergu grupeeritud üheks segmendiks, aga isegi, kui sellistel segmentidel osutus valituks keerulisem mudel, ei olnud seal keskmiste erinevus ühelgi juhul statistiliselt oluline.

Järelikult võime väita, et meie meetod on tundlikum kui t-testil põhinev alternatiivne variant.



Joonis 4. Kaks näidet algoritmi tulemustest. Andmed on genereeritud joonise 3 skeemi põhjal. Vasakpoolses tulbas $\sigma = 0.4$, paremal $\sigma = 0.7$. Joonistel (a) ja (d) on näide ühest genereeritud andmestikust, (b) ja (e) on algoritmi leitud segmentatsioonid 50 katsel ning joonistel (c) ja (f) vastavad t-testi tulemused. Tumesinine värv näitab gruppide A ja B keskmiste statistiliselt olulist erinevust.

5.2.2 Realistlikuma skeemi järgi genereeritud andmed

Eelmise alapeatüki tulemused annavad lootust, et algoritm annab vähemalt teatud olukordades mõistlikke tulemusi. Samas oli eelmises näites mitmeid lihtsustavaid eeldusi, mis võivad näidata algoritmi tulemusi paremana kui nad päris andmetel oleksid. Näiteks eelnevas olid genereeritud andmete kõigis veergudes standardhälbed võrdsed, lisaks oli ühe segmendi kõigi veergude keskväärtus konstantne. Kuna meie eesmärk on kasutada seda metülatsioonikiibi andmetel, üritame genereerida võimalikult sarnase struktuuriga andmestiku. Alljärgnev pseudokood kirjeldab, millise põhimõtte järgi andmed genereerisime.

Pseudokood 2 Andmete genereerimise skeem (N erineva segmendi jaoks)

for all $i \in \{1, ..., N\}$ **do**
  $l \leftarrow Po(\lambda = 7) + 1$ ▷ Lõigu pikkus
  $\mu_0 \leftarrow \mathcal{N}(0.5, 0.2^2)$ ▷ Segmendi keskväärtus
  $\sigma \leftarrow \Gamma(3, 8)$ ▷ Segmendi standardhälve
  $on\_erinevus \leftarrow Be(0.5)$
  if $on\_erinevus == 1$ **then**
    $\Delta \leftarrow U(0.2, 0.6)$
    $märk \leftarrow 2 \cdot Be(0.5) - 1$
    $\mu_1 \leftarrow \mu_0 + märk \cdot 0.5 \cdot \Delta$
    $\mu_2 \leftarrow \mu_0 - märk \cdot 0.5 \cdot \Delta$
    $M_i \leftarrow$ GENEREERI_ANDMED$(l, \mu_1, \mu_2, \sigma)$
  else
    $M_i \leftarrow$ GENEREERI_ANDMED$(l, \mu_0, \mu_0, \sigma)$
  end if
end for
$M \leftarrow [M_1, ..., M_N]$

function GENEREERI_ANDMED$(l, \mu_1, \mu_2, \sigma)$
  for all $i \in \{1, ..., l\}$ **do**
    $c \leftarrow U(-1, 1)$ ▷ i-nda veeru mõju
    $a_1, ..., a_8 \leftarrow \mathcal{N}(\mu_1, \sigma^2) + c$
    $a_9, ..., a_{20} \leftarrow \mathcal{N}(\mu_1, \sigma^2) + c$
    $M_i = [a_1, ..., a_8, a_9, ..., a_{20}]^T$
  end for
  return $[M_1, ..., M_l]$ ▷ Tagastab 20 rea ning l veeruga maatriksi
end function

Genereerisime selle skeemi järgi andmed $N = 250$ jaoks.

Jättes hetkel kõrvale segmentide pikkused, võime püstitatud ülesannet käsitleda binaarse klassifitseerimisülesandena veergude jaoks, kus iga veeru kohta tuleks otsustada, kas

kahe grupi keskväärtused selles veerus erinevad või mitte. Nii võime oma algoritmi käsitleda binaarse klassifitseerijana, kus võimalikeks klassideks oleksid:

- "1" ehk positiivne klass: kahe grupi keskväärtused selles veerus erinevad,
- "0" ehk negatiivne klass: kahe grupi keskväärtused selles veerus ei erine.

Klassifitseerija headuse hindamiseks esitatakse tavaliselt tulemused segadustabelina (*confusion matrix*), kus ridades on tegelikud klassid ning veergudes prognoositud klassid. Meie algoritmi kui klassifitseerija kohta on genereeritud andmestikul saadud tulemused näha tabelis 2.

Tabel 2. Segadustabel algoritmi kui klassifitseerija jaoks.

		Prognoositud klass	
		1	0
Tegelik klass	1	TP = 796	FN = 286
	0	FP = 10	TN = 934

Võrdluseks on tabelis 3 näha samadel andmetel veeru kaupa tehtud t-testi kui klassifitseerija tulemused.

Tabel 3. Segadustabel t-testi kui klassifitseerija jaoks.

		Prognoositud klass	
		1	0
Tegelik klass	1	TP = 143	FN =939
	0	FP = 0	TN = 944

Näeme, et tegelikkuses negatiivsesse klassi kuuluvate veergude (nii korrektselt kui ka valesti klassifitseeritud) korral on tulemused kummagi klassifitseerija puhul ligilähedaselt võrdsed. Aga tegelikkuses positiivsesse klassi kuuluvate veergude seast tuvastas meie meetod 74% ning t-test ainult 13%. Kuna meie põhieesmärgiks on leida üles sellised veerud, kus gruppide keskväärtused erinevad, siis võime oma meetodi tulemust pidada suhteliselt heaks, pidades silmas, et esines ka üksikuid valepositiivseid.

5.2.3 Kuidas sõltuvad algoritmi tulemused mudeli parameetrite kodeerimise täpsusest

Kuigi tavaliselt valitakse mudeli reaalarvuliste parameetrite kodeerimise täpsus selliselt, nagu seda eelnevalt tegime, võib tekkida küsimus, kas äkki praktikas annaks meie meetod paremaid tulemusi, kui otsustaksime parameetreid kodeerida suurema või väiksema täpsusega. Teisisõnu, MDL-printsiibi kohaselt minimeeritava summa $L(M) + L(D|M)$ asemel võime hoopis soovida minimeerida summat $cL(M) + L(D|M)$ mingi reaalarvu c korral.

Selles alapeatükis uurimegi, kuidas muutuvad algoritmi tulemused, kui muuta kordaja c väärtust. Näiteks oleks mõistlik arvata, et suure c väärtuse korral saaksime keskmiselt pikemaid segmente, sest sel juhul on uue segmendi defineerimine suhteliselt kallis (kuna see toob kaasa mitmete uute parameetrite kodeerimise) ning eelistatud on pikemad segmendid, isegi kui seetõttu mudel ei kirjelda andmeid enam eriti täpselt.

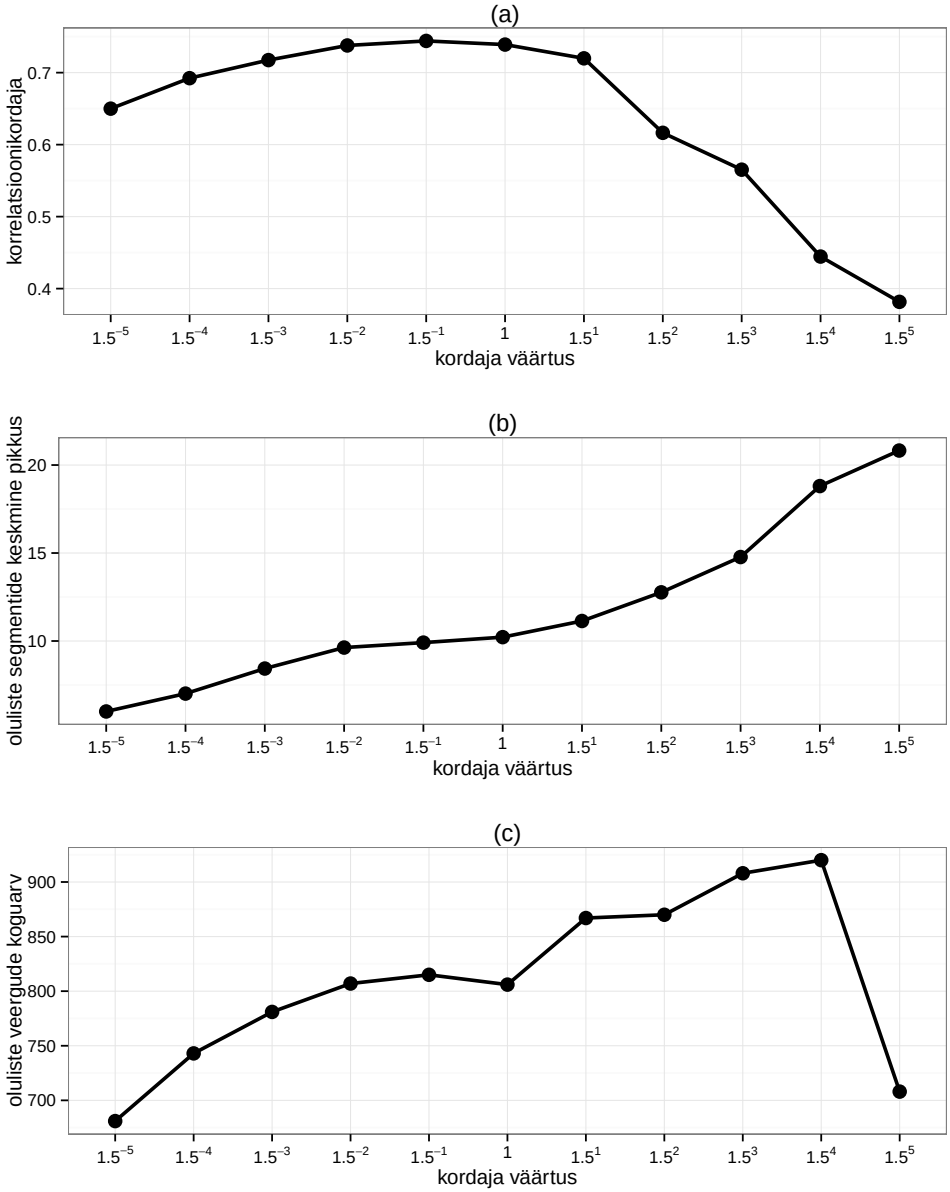
Proovisime erinevaid c väärtusi $c \in \{1.5^{-5}, 1.5^{-4}, ..., 1, ..., 1.5^{4}, 1.5^{5}\}$.

Joonisel 5 on näha tulemused, mida saime, kasutades eelmises alapeatükis kirjeldatud genereeritud andmestikul. Tulemuste headuse hindamiseks koostasime iga c jaoks segadustabeli ning arvutasime selle jaoks Matthew' korrelatsioonikordaja (vt lisa C), mille kohaselt oleks parimaks c väärtuseks ligikaudu 1.5^{-1}.

Lisaks uurisime oluliste segmentide keskmist pikkust, mis oodatavalt suureneb c kasvades. Seega, kui suurendada c väärtust, siis saaksime pikemaid segmente tulemuseks, aga nende kvaliteet (korrelatsioonikordaja mõttes) oleks viletsam. Selle põhjuseks võib olla asjaolu, et tulemuseks saadud segmendid on tegelikega võrreldes liiga pikad ning seetõttu klassifitseeritakse mitmeid veerge valesti.

Uurisime ka, milline oli veergude arv, mis tunnistati algoritmi poolt olulise erinevusega veeruks. Nende arv suurenes c kasvades teatud piirini, aga suurte c väärtuste korral langeb. Arvatavasti algul nende veergude arv kasvab seetõttu, et isegi kui pikem segment sisaldab teataval määral müra, liigitatakse see olulise erinevusega veeruks, kui piisavalt paljude veergude keskmised erinevad. Väga suure c väärtuse korral aga lähevad segmentide pikkused juba nii suureks, et enamikul segmentidel ei leita olulist keskmiste erinevust.

Saadud tulemusi arvesse võttes, on mõistlik jääda esialgse $c = 1$ juurde.



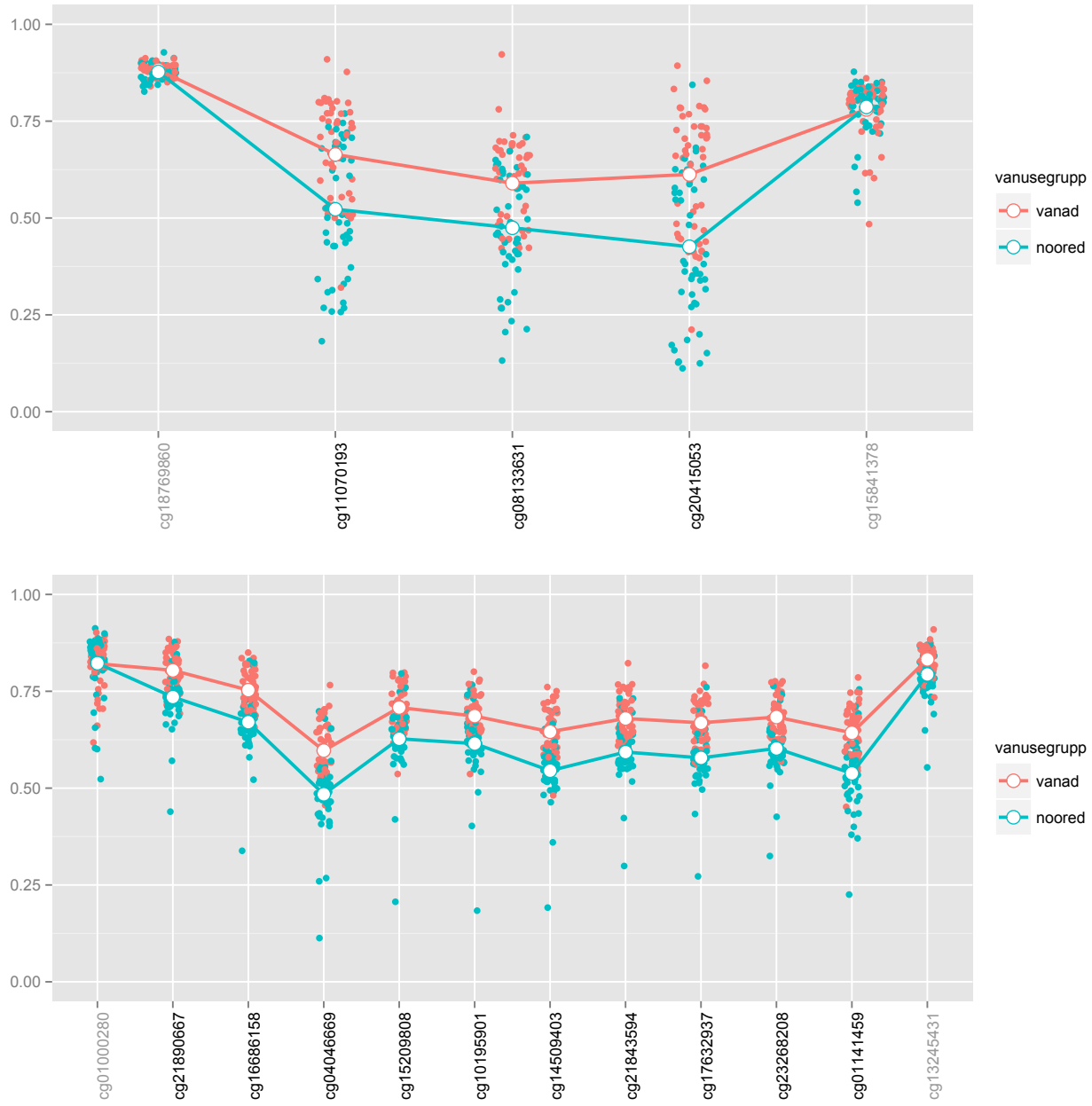
Joonis 5. Kuidas kordaja väärtuse muutmine mõjutab erinevaid näitajaid: (a) Matthew' korrelatsioonikordajale, (b) olulise erinevusega segmentide keskmisele pikkusele, (c) olulise erinevusega veergude koguarvule.

5.3 Rakendamine bioloogilistel andmetel

Rakendasime meetodit ka päris metülatsiooniandmetel. Selleks kasutasime mõlemat peatükis 1 mainitud andmestikku. Diferentsiaalselt metüleeritud regioonideks otsustasime nimetada selliseid, mille pikkus on vähemalt 3 CpG saiti ning kus gruppidevaheline erinevus on vähemalt 0.1.

Leidmaks diferentsiaalselt metüleeritud regioone andmestikust 100 patsiendi vererakkude metülatsiooni kohta, rakendasime igale vereraku tüübile kõigepealt peatükis 5.1 kirjeldatud meetodit, kusjuures keerukamasse mudelisse lisasime gruppi näitava tunnusena vanusegrupi: kas noor või vana. Saadud segmentidel testisime, kas vanusegrupi mõju on statistiliselt oluline. Näiteks CD4 immuunrakkude jaoks saime 8084 olulise erinevusega regiooni, nende keskmiseks pikkuseks oli 4.8. Kõige pikema regiooni pikkuseks oli 29 saiti.

Joonisel 6 on näha kaks näidet leitud regioonidest CD4 vererakkude jaoks. Esimese regiooni pikkus on 3 ning teise oma 10 CpG saiti. Esimese regiooni jaoks saime vanusegrupi mõju hinnanguks 0.148 ning teise regiooni jaoks 0.088. Kummagi regiooni jaoks on joonisele lisatud ka eelneva ning järgneva CpG kohta andmed, et oleks võimalik hinnata, kas regiooni piirid on mõistlikult paika pandud või jätkub samasugune gruppidevaheline erinevus ka väljaspool leitud regiooni. Vähemalt nende näidete põhjal on regiooni piir intuitsiooniga kooskõlas. Iga CpG saidi kummagi vanusegrupi keskmine on kantud joonisele, lisaks on need keskmised ühendatud joontega. Kuna metülatsioonikiibi andmed ei käi kõigi CpG saitide kohta, siis võib meie joonisele kantud CpG saitide vahel paikneda teisi saite, mille kohta meil andmestikus beeta-väärtusi pole. Seega, arvestades, et üldiselt on lähestikku paiknevate CpG saitide metüleeritus sarnane, leidub neil joontel teatav tähendus.



Joonis 6. Kaks näidet meetodi leitud diferentsiaalselt metüleeritud regioonidest. Horisontaalteljel on CpG saidid ning siltideks nende identifikaatorid. Tumedat värvi kirjaga CpG-d moodustavadki leitud regiooni. Kõige vasak- ning parempoolsemad CpG-d on joonisele kantud seetõttu, et lugeja saaks hinnata, kas segmendi piir on intuitiivselt õigesti defineeritud. Vertikaalteljel on metüleeritust näitavad beeta-väärtused, värv näitab patsientide vanusegruppe. Gruppide keskmised on ühendatud joontega.

Kokkuvõte

Käesolevas bakalaureusetöös on välja töötatud statistiline meetod DNA metülatsiooniandmete analüüsimiseks. On kirjeldatud üldine raamistik, mis võimaldab tuvastada diferentsiaalselt metüleeritud regioone, ning lisaks on implementeeritud kaks sellele raamistikule toetuvat konkreetset meetodit.

Raamistik, mille idee on pärit artiklist [7], seisneb andmete optimaalsel viisil segmentideks jagamises. Selleks sobitame erinevate pikkustega segmentidele statistilisi mudeleid ning valime välja andmete parima segmentatsiooni MDL-printsiibile toetudes.

Konkreetsete meetodite saamiseks kasutasime segmentidel mudelitena nii k-keskmiste klasterdamist kui ka lineaarseid mudeleid, lahendamaks kaht erinevat tüüpi ülesannet:

- Leidmaks segmente, kus andmed jagunevad gruppidesse, mis käituvad antud segmendil sarnaselt, aga meil ei ole teada, mille alusel gruppidesse jagunemine toimub, kasutasime klasterdamise mudelit.
- Leidmaks segmente, kus andmed jagunevad mõne meile teada oleva tunnuse (näiteks vanuse) järgi gruppidesse, sobitasime segmentidele lineaarseid mudeleid. Selline meetod võimaldab igal segmendil testida statistilisi hüpoteese gruppidevahelise erinevuse olulisuse kohta.

Testisime saadud meetodit genereeritud andmetel. Tulemused annavad alust arvata, et meie meetod annab paremaid tulemusi kui lihtne alternatiiv. Samuti rakendasime meetodeid mitmetel bioloogilistel andmestikel, kus leitud diferentsiaalselt metüleeritud regioone on kasutatud edasistes analüüsides ning ka laboris valideeritud.

Tööd on võimalik mitmes suunas jätkata:

- Kuna üldine raamistik on nüüd välja töötatud, saab kasutada seda erinevate mudelite jaoks, sõltuvalt andmete olemusest. Näiteks võiksime tavalise lineaarse mudeli asemel segmentidel andmeid kirjeldada Poissoni regressiooni abil.
- Kuna meetod on väga arvutusmahukas, tuleks otsida lahendusi, kuidas muuta algoritmi efektiivsemaks.
- Kuna töös kirjeldatud meetodid on osutunud praktikas kasulikuks, plaanime need avaldada R-i paketina.

An MDL Method for Identifying Differentially Methylated Genomic Regions

Bachelor's thesis (6 ECTS)

Kaspar Märtens

Summary

The goal of this thesis is to develop a statistical method for studying DNA methylation patterns in longer regions than just single CpG sites. We are especially interested in finding differentially methylated regions (DMRs).

Our idea is to find an optimal segmentation of the genome with the help of segmentwise defined models and the Minimum Description Length (MDL) principle. First, we describe the general framework for achieving this with an arbitrary set of models. We fit these models to each segment and choose the best segmentwise model in terms of the MDL principle. This model also defines the optimal segmentation of the data.

We have developed two specific methods based on this framework. The first one assumes binary data and uses k-means clustering on each segment. The other method is based on fitting linear models to the segments. In addition to finding an optimal segmentation of the data, this allows us to perform hypothesis testing on the segments.

Both these methods were implemented in R.

We studied the performance of these algorithms on generated data and also used them on real methylation data. In all of the observed cases, the results of our methods seem to be promising.

Lisa A Pideva tõenäosusliku mudeli kodeerimine

Olgu $\mathcal{X} = \mathbb{R}$ ning olgu f juhusliku suuruse X pidev tihedusfunktsioon. Järgneva arutelu idee on pärit allikast [9].

Kodeerimaks mingit elementi $x \in \mathbb{R}$, diskretiseerime hulga $\mathbb{R}$ piisava täpsusega δ. See tähendab, et iga $\delta > 0$ korral saame esitada hulga $\mathbb{R}$ loenduva ühendina

$$\mathbb{R} = \bigcup_{\substack{x=\delta t \\ t\in\mathbb{Z}}} \left[x - \frac{\delta}{2}, x + \frac{\delta}{2}\right] = \bigcup_{t\in\mathbb{Z}} \left[\left(t - \frac{1}{2}\right)\delta, \left(t + \frac{1}{2}\right)\delta\right].$$

Tähistame diskretiseeritud juhusliku suuruse X^δ, mille võimalike väärtuste hulk on $\{t\delta : t \in \mathbb{Z}\}$, kusjuures

$$p_t := \mathbf{P}(X^\delta = t\delta) = \int_{(t-\frac{1}{2})\delta}^{(t+\frac{1}{2})\delta} f(x)dx$$

Defineerime tihedusfunktsiooniga f pideva juhusliku suuruse X diferentsiaalse entroopia

$$H(X) = -\int_{-\infty}^{\infty} f(x)\log f(x)dx.$$

Siis piisavalt väikese $\delta > 0$ korral kehtib

$$H(X^\delta) \approx H(X) - \log\delta.$$

Põhjendus. Näitame, et $H(X) - H(X^\delta) \approx \log\delta$.

Integraali aditiivsuse tõttu

$$H(X) = -\int_{-\infty}^{\infty} f(x)\log f(x)dx = -\sum_{t\in\mathbb{Z}} \int_{(t-\frac{1}{2})\delta}^{(t+\frac{1}{2})\delta} f(x)\log f(x)dx.$$

Arvestades, et diskretiseeritud juhusliku suuruse X^δ entroopia avaldub

$$\begin{aligned} H(X^\delta) &= -\sum_{t\in\mathbb{Z}} p_t \log p_t = -\sum_{t\in\mathbb{Z}} \underbrace{\left(\int_{(t-\frac{1}{2})\delta}^{(t+\frac{1}{2})\delta} f(x)dx\right)}_{p_t} \log \underbrace{\left(\int_{(t-\frac{1}{2})\delta}^{(t+\frac{1}{2})\delta} f(z)dz\right)}_{p_t} \\ &= -\sum_{t\in\mathbb{Z}} \int_{(t-\frac{1}{2})\delta}^{(t+\frac{1}{2})\delta} f(x)\log\left(\int_{(t-\frac{1}{2})\delta}^{(t+\frac{1}{2})\delta} f(z)dz\right)dx, \end{aligned}$$

saame

$$H(X) - H(X^\delta) = \sum_{t\in\mathbb{Z}} \int_{(t-\frac{1}{2})\delta}^{(t+\frac{1}{2})\delta} \left[-f(x)\log f(x) + f(x)\log\left(\int_{(t-\frac{1}{2})\delta}^{(t+\frac{1}{2})\delta} f(z)dz \right) \right] dx.$$

Matemaatilisest analüüsist on teada omadus, et lõigul $[a,b]$ pideva funktsiooni g korral leidub punkt $\xi \in [a,b]$ nii, et $\int_a^b g(x) = g(\xi)\cdot(b-a)$. Seda kasutades saame, et iga $t \in \mathbb{Z}$ korral leidub punkt ξ_t nii, et

$$\int_{(t-\frac{1}{2})\delta}^{(t+\frac{1}{2})\delta} f(z)dz = \delta f(\xi_t).$$

Nüüd, seda rakendades

$$\begin{aligned}
H(X) - H(X^\delta) &= \sum_{t\in\mathbb{Z}} \int_{(t-\frac{1}{2})\delta}^{(t+\frac{1}{2})\delta} \left[-f(x)\log f(x) + f(x)\log\left(\delta f(\xi_t)\right) \right] dx \\
&= \sum_{t\in\mathbb{Z}} \int_{(t-\frac{1}{2})\delta}^{(t+\frac{1}{2})\delta} \left[-f(x)\log f(x) + f(x)\log\delta + f(x)\log f(\xi_t) \right] dx \\
&= \log\delta \int_{-\infty}^{\infty} f(x)dx + \sum_{t\in\mathbb{Z}} \int_{(t-\frac{1}{2})\delta}^{(t+\frac{1}{2})\delta} f(x)\left[\log f(\xi_t) - \log f(x)\right] dx \\
&= \log\delta + \sum_{t\in\mathbb{Z}} \int_{(t-\frac{1}{2})\delta}^{(t+\frac{1}{2})\delta} f(x)\left[\log\frac{f(\xi_t)}{f(x)}\right] dx.
\end{aligned}$$

Kui $\delta > 0$ on piisavalt väike, siis iga $x \in \left[\left(t-\frac{1}{2}\right)\delta, \left(t+\frac{1}{2}\right)\delta\right]$ korral $x \approx \xi_t$ ning funktsiooni f pidevuse tõttu $f(x) \approx f(\xi_t)$.

Siis $\log\frac{f(\xi_t)}{f(x)} \approx \log 1 = 0$ ning seega $H(X) - H(X^\delta) \approx \log\delta$. □

Peatükist 2.1 teame, et ühegi koodi keskmine pikkus ei saa olla väiksem kui jaotuse entroopia $H(X^\delta)$. Arvestades, et $H(X^\delta) \approx H(X) - \log\delta$, on selge, et sellise optimaalse keskmise pikkuse saavutame, kui defineerime iga $x \in \mathbb{R}$ korral

$$\tilde{L}(x) = -\log f(x) - \log\delta,$$

sest sel juhul on koodi keskmine pikkus

$$\int_{-\infty}^{\infty} f(x)\tilde{L}(x)dx = -\int_{-\infty}^{\infty} f(x)\log f(x) - \log\delta \int_{-\infty}^{\infty} f(x)dx = H(X) - \log\delta.$$

Esituspikkuste omavahelise võrdlemise juures pole konstandil $-\log\delta$ tähtsust, seega defineerime hoopis iga $x \in \mathbb{R}$ korral

$$L(x) = -\log f(x).$$

Lisa B Arvutuseeskiri segmentatsioonide koguarvu leidmiseks

Tähistame iga naturaalarvu a ja b korral lõigu

$$[a, b] := \{n \in \mathbb{N} : a \leq n \leq b\}$$

ning selle lõigu pikkuse $l_{[a,b]} := b - a + 1$.

Olgu $a \leq b$ ning $l_{[a,b]} = n$. Olgu $m \in \mathbb{N}$. Siis selliste lõikude $[i, j] \subset [a, b]$, mille korral $l_{[i,j]} \leq m$, koguarv on

$$\#\{[i, j] \subset [a, b] : l_{[i,j]} \leq m, i \leq j\} = \begin{cases} \frac{n(n+1)}{2} & \text{kui } n \leq m \\ mn - \frac{m(m-1)}{2} & \text{kui } n > m \end{cases}$$

Põhjendus. Kõigepealt vaatleme olukorda, kus $n \leq m$. Siis iga $i, j \in [a, b]$ korral kehtib $l_{[i,j]} \leq l_{[a,b]} = n \leq m$. Seega piisab leida lõikude $[i, j] \subset [a, b]$ arv. Et sellisteks lõikudeks $[i, j]$ on parajasti hulgad

$$\begin{aligned} &[a, a], [a, a + 1], \ldots, [a, b] \\ &[a + 1, a + 1], \ldots, [a + 1, b] \\ &\ldots \\ &[b, b], \end{aligned}$$

siis on nende arv $n + (n - 1) + \ldots + 1 = \frac{n(n+1)}{2}$

Olgu nüüd $n \geq m$. Paneme tähele, et $n = m$ korral on avaldised $mn - \frac{m(m-1)}{2}$ ning $\frac{n(n+1)}{2}$ võrdsed, seega kehtib $n = m$ korral eeskiri $mn - \frac{m(m-1)}{2}$. Näitamaks, et valem kehtib iga $n \geq m$ korral, kasutame matemaatilist induktsiooni.

Juhul $n = m$ on kehtivus põhjendatud. Kehtigu nüüd valem mingi $k \geq m$ korral ja olgu $[a, a + k - 1]$ lõik, mille korral see kehtib. Näitame nüüd, et valem kehtib $k + 1$ korral.

Selleks piisab näidata, et lõikude $[i,j] \subset [a, a+k]$ arv on m võrra suurem kui lõikude $[i,j] \subset [a, a+k-1]$ arv. Seega on vaja näidata, et leidub parajasti m sellist lõiku $[i,j]$, mille korral $[i,j] \subset [a, a+k]$ ning $[i,j] \not\subset [a, a+k-1]$. On selge, et kehtib samaväärsus

$$([i,j] \subset [a,a+k] \wedge [i,j] \not\subset [a,a+k-1]) \iff$$
$$\iff (a \leq i \leq a+k) \wedge (j = a+k).$$

Jääb veel leida, milliste i väärtuste korral $l_{[i,a+k]} \leq m$. Ilmselt on nendeks i väärtusteks $a+k, a+k-1, \ldots, a+k-(m-1)$. Neid on kokku m tükki, seega on lõike $[i,j] \subset [a, a+k]$ m võrra rohkem kui lõike $[i,j] \subset [a, a+k-1]$.

Seega kehtib väide ka iga $n \geq m$ korral. □

Lisa C Matthew' korrelatsioonikordaja

Olgu X, Y binaarsed tunnused, mis näitavad vastavalt meie klassifitseerija poolt prognoositud klassi ning tegelikku klassi. Olgu tabel 4 selle klassifitseerija segadustabel.

Tabel 4. Segadustabel klassifitseerija jaoks.

		Prognoositud klass (X)	
		1	0
Tegelik klass (Y)	1	TP	FN
	0	FP	TN

Masinõppes kasutatakse klassifitseerija ühe headuse näitajana Matthew' korrelatsioonikordajat, mis arvutatakse järgneva eeskirja kohaselt:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FN)(TP+FP)(TN+FP)(TN+FN)}}.$$

Selle kordaja eesmärgiks on iseloomustada segadustabelit ühe arvu abil, mis võtaks arvesse kõiki nelja tabelis olevat näitajat. On lihtne kontrollida, et kui eeldada, et $X \sim Be\left(\frac{TP+FP}{TP+FP+FN+TN}\right)$ ning $Y \sim Be\left(\frac{TP+TN}{TP+FP+FN+TN}\right)$ ning kasutada Bernoulli jaotuse dispersiooni valemit, siis on Matthew' korrelatsioonikordaja samaväärne Pearsoni korrelatsioonikordajaga binaarsete tunnuste X ja Y vahel. [10]

Viited

[1] S. Saxonov, P. Berg, *et al.*, "A genome-wide analysis of CpG dinucleotides in the human genome distinguishes two distinct classes of promoters," *Proceedings of the National Academy of Sciences of the United States of America*, pp. 1412–1417, 2006.

[2] P. A. Jones, "Functions of DNA methylation: islands, start sites, gene bodies and beyond," *Nature Reviews Genetics*, pp. 484–492, 2012.

[3] N. Thorne, J. Marioni, *et al.*, "DNA methylation arrays: Methods and analysis," in *Microarray Innovations: Technology and Experimentation in Drug Discovery and Biomedical Research*, 2009.

[4] P. D. Grünwald, *The Minimum Description Length Principle*. Cambridge, Massachusetts: The MIT Press, 2005.

[5] M. H. Hansen and B. Yu, "Model selection and the principle of minimum description length," *Journal of the American Statistical Association*, vol. 96, pp. 746–774, 2001.

[6] J. Lember, "Informatsiooniteooria. Loengukonspekt ja ülesanded," 2013.

[7] M. Koivisto, M. Perola, *et al.*, "An MDL method for finding haplotype blocks and for estimating the strength of haplotype block boundaries.," *Pacific Symposium on Biocomputing*, pp. 502–513, 2003.

[8] S. Laur, "Comprehensive overview of various haplotype block inference methods," 2009.

[9] T. Roos, "Information-Theoretic Modelling." University of Helsinki, 2009.

[10] P. Baldi, S. Brunak, *et al.*, "Assessing the accuracy of prediction algorithms for classification: an overview," *Bioinformatics*, pp. 412–424, 2000.

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, Kaspar Märtens (sünnikuupäev 23. november 1990),

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose **"MDL-meetod diferentsiaalselt metüleeritud regioonide tuvastamiseks"**, mille juhendaja on Raivo Kolde,

    1.1. reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;

    1.2. üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.

3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus 6. mail 2013