# Latenttiin muuttujamalliin perustuva ordinaatiomenetelmä

Jenni Niku

Tilastotieteen pro gradu

Jyväskylän yliopisto
Matematiikan ja tilastotieteen laitos
25. tammikuuta 2015

JYVÄSKYLÄN YLIOPISTO
Matematiikan ja tilastotieteen laitos
**Jenni Niku**: Latenttiin muuttujamalliin perustuva ordinaatiomenetelmä
Tilastotieteen pro gradu -tutkielma, 31 sivua + liitteitä 21 sivua, 25. tammikuuta 2015

## Tiivistelmä

Ordinaatiomenetelmissä useita vastemuuttujia sisältävän aineiston informaatio pyritään tiivistämään muutamaan muuttujaan, joista muodostetaan ordinaatiokuva. Klassiset ordinaatiomenetelmät ottavat huomioon aineiston ominaisuudet vain etäisyysmitan ja aineiston muunnosten valinnassa, jolloin ne eivät tarjoa mahdollisuutta oletusten ja menetelmän sopivuuden tarkasteluun. Tässä tutkielmassa tarkastellaan yleistettyä lineaarista latenttien muuttujien mallia ordinaatiomenetelmänä. Menetelmässä vastemuuttujien informaatio tiivistetään kahteen latenttiin muuttujaan, joista ordinaatiokuva muodostetaan. Mallin parametrit estimoidaan suurimman uskottavuuden menetelmällä, ja uskottavuusfunktiolle johdetaan Laplacen approksimaatio yleisen eksponentiaalisen perheen jakauman tapauksessa.

Latenttia muuttujamallia sovelletaan kahteen ekologian aineistoon, ja havainnollistetaan menetelmän mahdollistamia informaatiokriteerien käyttöä mallinvalinnassa sekä jäännöstarkasteluja oletusten tarkistamiseksi. Simulointikokeilla verrataan latenttia muuttujamallia klassisiin ordinaatiomenetelmiin. Tulosten perusteella voidaan todeta, että riittävän hyvilla alkuarvojen valinnalla latentti muuttujamalli toimii ordinaatiomenetelmänä vähintään yhtä hyvin tai paremmin kuin mikään vertailtavista klassisista menetelmistä, jotka toimivat vaihtelevalla menestyksellä aineistosta riippuen.

**Avainsanat:** Eksponentiaalinen jakaumaperhe, Laplacen approksimaatio, latentti muuttuja, moniulotteinen skaalaus, korrespondenssianalyysi, pääkoordinaattianalyysi.

# Sisältö

**1 Johdanto** **1**

**2 Klassiset ordinaatiomenetelmät** **2**
2.1 Moniulotteinen skaalaus . . . . . 2
2.2 Pääkoordinaattianalyysi . . . . . 3
2.3 Korrespondenssianalyysi . . . . . 3
2.4 Klassisten ordinaatiomenetelmien ongelmia . . . . . 4

**3 Latentti muuttujamalli** **7**
3.1 Dikotominen vaste . . . . . 8
3.2 Lukumäärävaste . . . . . 8

**4 Implementointi** **9**
4.1 Latentin muuttujamallin uskottavuusfunktio . . . . . 9
4.2 Laplacen approksimaatio . . . . . 10
4.3 Laplacen approksimaatio latenttien muuttujien mallin uskottavuusfunktiolle . . . . . 11
4.4 Bernoulli-jakautuneet vastemuuttujat . . . . . 12
4.5 Poisson-jakautuneet vastemuuttujat . . . . . 13
4.6 Negatiivibinomijakautuneet vastemuuttujat . . . . . 14
4.7 Dunn-Smyth residuaalit ja informaatiokriteerit . . . . . 15

**5 Sovelluksia** **17**
5.1 Muurahaiset . . . . . 17
5.2 Hämähäkit . . . . . 20

**6 Simulointikokeita** **25**

**7 Pohdintaa** **30**

**Viitteet** **31**

**Liite A Derivaatat** **32**
A.1 Bernoulli-jakautuneille vastemuuttujille . . . . . 32
A.2 Poisson-jakautuneille vastemuuttujille . . . . . 33
A.3 Negatiivibinomijakautuneille vastemuuttujille . . . . . 33
A.4 Derivaatat selittävät muuttujat sisältävälle mallille . . . . . 34

**Liite B Etäisyysmittoja** **36**

**Liite C R-koodi** **37**

# 1 Johdanto

Ordinaatiomenetelmät ovat yleisesti käytettyjä ekologiassa, kun halutaan visualisoida moniulotteista aineistoa mataladimensioisessa muodossa. Usein halutaan kuvata lajiston samankaltaisuutta eri paikoissa. Tällöin ordinaation tavoitteena on tiivistää useita vastemuuttujia sisältävän aineiston informaatio vain kahteen muuttujaan, joiden perusteella muodostetaan sirontakuvio. Sirontakuviossa toisiaan lähellä olevat paikat tulkitaan olevan tutkittavan asian kannalta samankaltaisia. Ordinaatiomenetelmää sanotaan rajoittamattomaksi, kun käytetään vain vastemuuttujista koostuvaa aineistoa.

Perinteiset ordinaatiomenetelmät, kuten moniulotteinen skaalaus (*multidimensional scaling*, MDS), pääkoordinaattianalyysi (*principal coordinate analysis*, PCoA) ja oikaistu korrespondenssianalyysi (*detrended correspondence analysis*, DCA), ovat algoritmipohjaisia tekniikoita. MDS-menetelmässä ordinaatiokuvan pisteiden paikkoja päivitetään iteratiivisesti, kunnes niiden parittaiset etäisyydet vastaavat mahdollisimman hyvin vastaavien paikkojen välisiä etäisyyksiä. Etäisyydellä tarkoitetaan jotakin aineiston mittauspaikkojen havainnoista laskettua erilaisuutta tai etäisyyttä kuvaavaa mittaa. PCoA-menetelmässä sovelletaan singulaariarvohajotelmaa parittaisia eroavaisuuksia kuvaavalle matriisille. Korrespondenssianalyysiä voidaan ajatella kyseisenä menetelmänä, jossa käytetään $\chi^2$ -etäisyysmittaa, ja DCA-menetelma taas on muunnelma tästä. Luvussa 2 käsitellään klassisia menetelmiä tarkemmin.

Perinteisissä ordinaatiomenetelmissä aineiston ominaisuudet, esimerkiksi keskiarvo-varianssi -suhde, pyritään ottamaan huomioon etäisyysmitan ja muunnosten valinnassa. Menetelmien ongelmana on se, että esimerkiksi jäännöstarkastelujen avulla ei voida tarkistaa valintojen sopivuutta tutkittavana olevalle aineistolle. Koska eri valinnat voivat tuottaa hyvin erilaisia tuloksia, saatetaan päätyä harhaanjohtaviin tuloksiin.

Malliperusteinen lähestymistapa korjaa perinteisten ordinaatiomentelmien ongelmia mahdollistaen jäännösten analysoinnin oletusten tarkistamiseksi, ja antaa työkaluja sopivimman mallin valitsemiseen. Artikkelissa (Hui et al., 2014) on esitelty mallipohjaisia lähestymistapoja ordinaatiomenetelmälle. Yksi näistä on malli, jossa vasteiden odotusarvoa selitetään latenteilla muuttujilla. Näiden muuttujien arvot jokaiselta mittauspaikalta voidaan ajatella olevan koordinaatit, jotka kuvaavat kunkin paikan sijaintia ordinaatiokuvassa.

Edellä mainitussa artikkelissa latenttia muuttujamallia on estimoitu suurimman uskottavuuden menetelmällä EM-algoritmin ja Monte Carlo -integroinnin avulla. Laskenta oli kuitenkin melko hidasta jo pienillä aineistoilla. Tämän työn tarkoituksena on laskea Laplacen approksimaatio kyseisen mallin uskottavuusfunktiolle, kun vastemuuttujat noudattavat eksponentiaalisen jakaumaperheen jakaumaa, ja soveltaa menetelmää kahteen ekologian aineistoon. Lisäksi mallia laajennetaan lisäämällä selittävät muuttujat malliin. Lopuksi mallipohjaista menetelmää verrataan klassisiin ordinaatiomenetelmiin simulointikokeiden avulla.

# 2 Klassiset ordinaatiomenetelmät

Oletetaan aineiston olevan $n \times p$ matriisi, jonka solun $(i, j)$ havainto on $y_{ij}$. Aineisto koostuu esimerkiksi $n$ havaintopaikasta, joista jokaiselta on mitattu $p$ muuttujaa. Tavoitteena on tiivistää aineiston informaatio paikkojen samankaltaisuudesta muutamaan muuttujaan, joiden perusteella tehdään päätelmät. Ennen laskentaa aineistolle voidaan tehdä tarvittaessa myös muunnoksia kuten standardointi. Tässä luvussa esitellään muutamia yleisimpiä klassisia ordinaatiomenetelmiä perustuen kirjan *Experimental Design and Data Analysis for Biologists* (Quinn ja Keough, 2002) lukuihin 15, 17, ja 18.

## 2.1 Moniulotteinen skaalaus

Moniulotteinen skaalaus, MDS, on käytetyin rajoittamaton ordinaatiomenetelmä ekologiassa. Muodostetaan paikkojen erilaisuutta kuvaava matriisi siten, että lasketaan jokaisen kahden paikan välinen etäisyys käyttäen aineistoon soveltuvaa etäisyysmittaa. Merkitään paikkojen $h$ ja $i$ välistä etäisyyttä $d_{hi}$, jolloin etäisyysmatriisi on $[d_{hi}]_{n\times n}$. Yksi usein käytetty mitta on Bray-Curtis -etäisyys:

$$d_{hi} = \frac{\sum_{k=1}^{p} |y_{hk} - y_{ik}|}{\sum_{k=1}^{p} (y_{hk} + y_{ik})}.$$

Muita etäisyysmittoja on esitelty liitteessä B. Seuraavaksi valitaan akselien lukumäärä $q$. Tämän jälkeen paikkojen $q$-ulotteisen avaruuden koordinaateille asetetaan lähtöarvot. Merkitään paikkaa $h$ vastaavia koordinaatteja $x_{h1}, \ldots, x_{hq}$ ja siirretään näitä koordinaatteja siten, että jokaisella askeleella koordinaattien välisten euklidisten etäisyyksien $\tilde{d}_{hi} = \|\mathbf{x}_h - \mathbf{x}_i\|$ ja aineistosta laskettujen etäisyyksien $d_{hi}$ yhteensopivuus paranee. Kun se ei enää parane, parhaat sijainnit ordinaatiokuvassa on saavutettu. Yhteensopivuutta voidaan mitata esimerkiksi Kruskalin stressiarvolla

$$S = \sqrt{\frac{\sum_{h,i} (\tilde{d}_{hi} - d_{hi})^2}{\sum_{h,i} (\tilde{d}_{hi})^2}},$$

joka saa arvon nolla, jos yhteensopivuus on täydellinen. Mitä pienempi stressiarvo on, sitä parempi on yhteensopivuus.

Epämetrisessä moniulotteisessa skaalauksessa (NMDS) aineistosta laskettujen etäisyyksien $d_{ij}$ sijaan käytetään niiden järjestyslukuja. MDS-menetelmä olettaa, että etäisyyksien $d_{ij}$ ja koordinaattien välisten etäisyyksien $\tilde{d}_{ij}$ suhde on lineaarinen, mutta näin ei aina välttämättä ole. NMDS menetelmässä oletetaan ainoastaan, että edellä mainittu suhde on monotoninen, eli

$$\tilde{d}_{hi} = f(d_{hi}),$$

missä $f : \mathbb{R}_+ \to \mathbb{R}_+$ on tuntematon monotoninen funktio. Koska tiedetään vain, että $\tilde{d}_{hi} \leq \tilde{d}_{h'i'}$, kun $d_{hi} < d_{h'i'}$, lasketaan järjestysluvut molemmille etäisyyksille. Tällöin stressiarvo perustuu näihin järjestyslukuihin (Kruskal, 1964).

## 2.2 Pääkoordinaattianalyysi

Esitellään pääkoordinaattianalyysi, (PCoA), vaiheittain. Muodostetaan aluksi etäisyysmatriisi $[d_{hi}]_{n\times n}$. Kuten aineistollekin, myös tälle voidaan tarvittaessa tehdä joitakin muunnoksia esimerkiksi negatiivisten ominaisarvojen välttämiseksi. Jälleen etäisyyttä kuvaavan mitan voi vapaasti valita. Lisäksi tehdään kaksinkertainen keskistys

$$d^*_{hi} = d_{hi} - \overline{d}_{h.} - \overline{d}_{.i} + \overline{d},$$

missä $\overline{d}_{h.}$ on rivin $h$, $\overline{d}_{.i}$ sarakkeen $i$ ja $\overline{d}$ koko matriisin keskiarvo. Merkitään muunnosten jälkeen saatua etäisyysmatriisia $D := [d^*_{hi}]_{n\times n}$. Matriisille $D$ sovellamme singulaariarvohajotelmaa, joka on nyt symmetriselle matriisille myös ominaisarvohajotelma

$$D = U\Lambda U^{-1}, \tag{1}$$

missä $\Lambda$ on $n \times n$ diagonaalimatriisi, jonka diagonaalilla on matriisin $D$ ominaisarvot $(\lambda_1, \ldots, \lambda_n)$. Matriisin $U$ sarake $i$, $\boldsymbol{u}_i$, on ominaisarvoa $\lambda_i$ vastaava ominaisvektori. Suurin osa etäisyysmatriisin $D$ informaatiosta sisältyy muutamaa suurinta ominaisarvoa vastaaviin ominaisvektoreihin. Ordinaatiokuva muodostetaan kahta suurinta ominaisarvoa vastaavista vektoreista ominaisarvojen neliöjuurilla skaalaamisen jälkeen.

## 2.3 Korrespondenssianalyysi

Korrespondenssianalyysi, (CA), on samankaltainen kuin pääkoordinaattianalyysi. Aluksi aineistolle tehdään standardointi

$$\tilde{y}_{ij} = \frac{y_{ij} - e_{ij}}{\sqrt{r_i}\sqrt{c_j}},$$

missä $y_{ij}$ on muuttujan $j$ havainto paikassa $i$, $e_{ij}$ on kyseisen havainnon odotettu arvo, $r_i$ on rivisumma ja $c_j$ on sarakesumma. Etäisyytenä käytetään $\chi^2$-mittaa

$$d_{hi} = \sqrt{\sum_{k=1}^{p} \frac{(\tilde{y}_{hk}/r_h - \tilde{y}_{ik}/r_i)^2}{c_k}}.$$

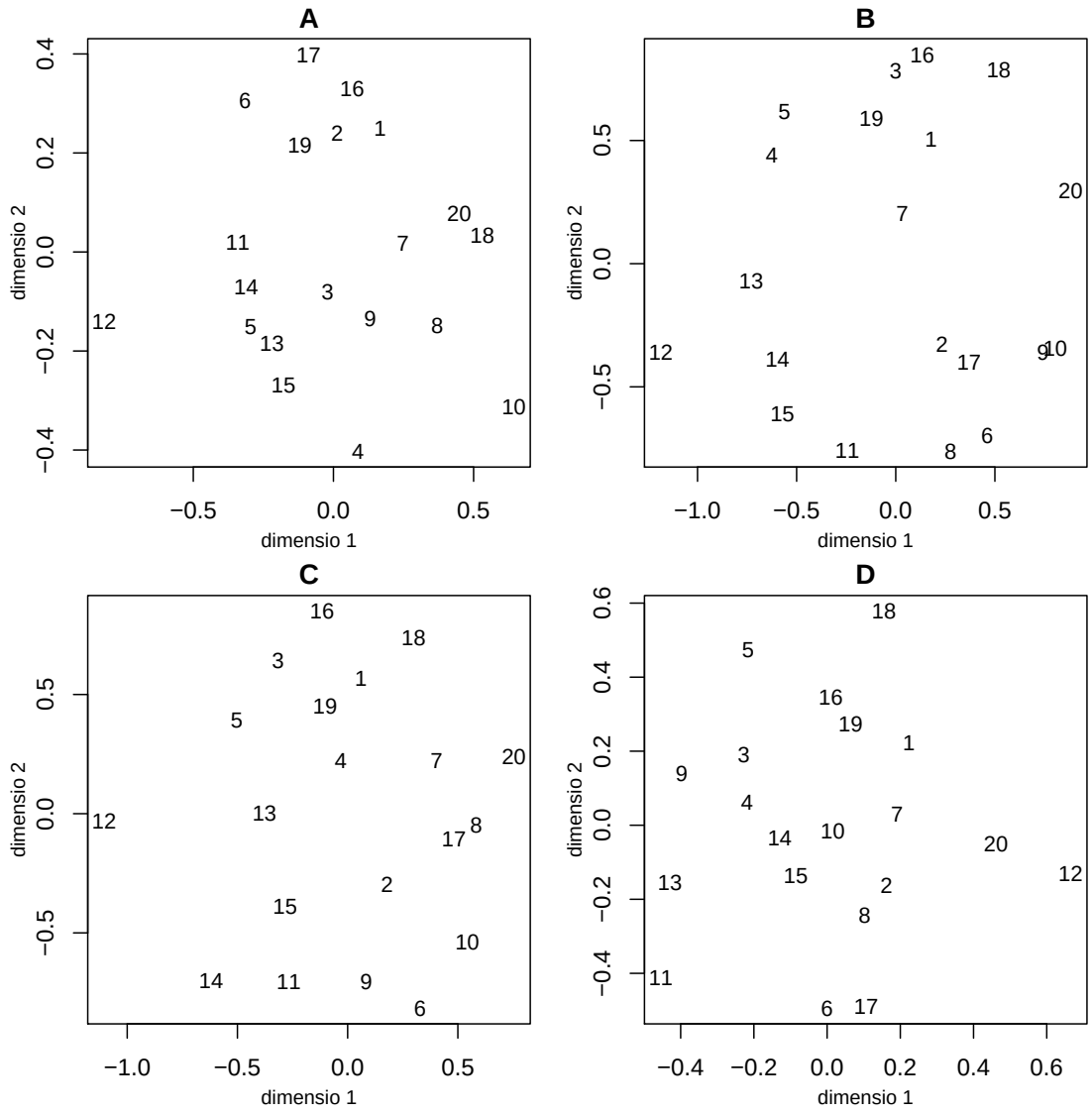
Merkitään etäisyysmatriisia $D := [d_{hi}]_{n\times n}$, ja tälle sovellamme ominaisarvohajotelmaa (1) ominaisarvojen ja ominaisvektoreiden löytämiseksi. Ordinaatiokuva muodostetaan ominaisvektoreista kuten PCoA-menetelmässä. Oikaistu korrespondenssianalyysi, (DCA), on nimensä mukaisesti kehittyneempi versio korrespondenssianalyysistä. CA-menetelmällä syntyneessä ordinaatiokuvassa on usein havaittavissa käyrämäinen

tai hevosenkenkämäinen muoto, joka ei johdu aineiston rakenteesta, vaan itse menetelmästä. DCA-menetelmällä pyritään korjaamaan tätä vääristymää. Siinä ensimmäinen akseli jaetaan osiin, joiden lukumäärän voi tutkija määrittää, ja osiot skaalataan siten, että kaikkien osioiden toisen akselin vastaavien arvojen keskiarvot ovat yhtäsuuret. Tätä toistetaan muutellen samalla osien välisiä rajoja jokaisella kierroksella. (Hill ja Gauch Jr, 1980).

## 2.4 Klassisten ordinaatiomenetelmien ongelmia

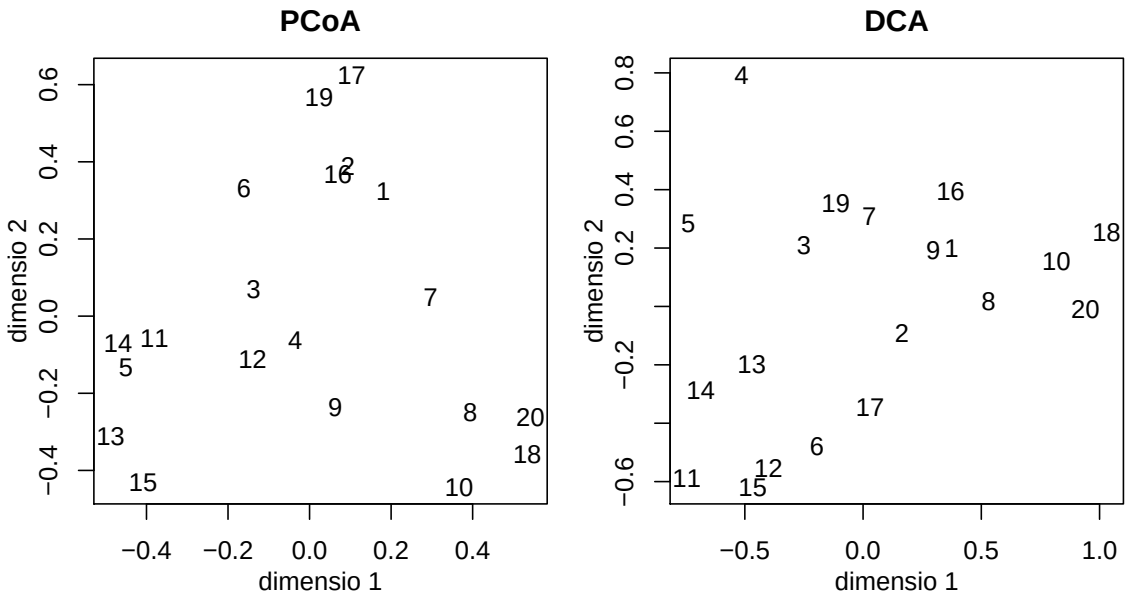
Klassisissa ordinaatiomenetelmissä ongelmia tuottavat etäisyysmitan valinta ja aineistolle tehtävät muunnokset, koska ei ole juurikaan olemassa diagnostisia keinoja tarkistaa, mitkä valinnat ovat sopivimpia käsiteltävänä olevalle aineistolle. Näin ollen päätökset on usein tehty perustuen uskomuksiin kuten aiempaan empiiriseen näyttöön sen sijaan, että ne perustuisivat itse aineistoon. Epäsopivista valinnoista saattaa seurata esimerkiksi aineistoon sopimaton keskiarvo-varianssi -suhde. Tämä taas voi aiheuttaa ordinaatiokuvan sijaintien trendin sekoittumisen hajonnan vaihteluun ja sen seurauksena harhaanjohtavia tuloksia.

Sovelletaan klassisia ordinaatiomenetelmiä muurahaisaineistoon (Stoklosa et al., 2014), jota tarkastellaan myöhemmin luvussa 5.1. Aineistossa on 18 muurahaislajin lukumääriä 20 paikalta. Kuvassa 1 on sovellettu tälle aineistolle NMDS-menetelmää eri etäisyysmitan ja muunnosten valinnoilla. Käytetyt etäisyysmitat ja Wisconsin-standardointi on määritelty liitteessä B. Ordinaatiokuvia vertailemalla voidaan havaita selviä poikkeamia: Kuvissa C ja D ei erotu klustereita juurikaan, kun taas kuvassa B erottuu selkeimmin kolme isompaa ryhmittymää. Myös ryhmien kokoonpanot vaihtelevat suuresti. Esimerkiksi kuvien A ja B yläreunan klusterit sisältävät osittain toisistaan poikkeavia paikkoja. Kolmanneksi, yksittäisten paikkojen sijainti suhteessa muihin vaihtelee huomattavasti. Esimerkiksi paikka 20 nähdään kuvassa B olevan selkeästi erillään muista, kun taas kuvassa A se on lähellä paikkoja 7 ja 18 ja kuvassa D melko lähellä paikkaa 12. Ongelmana menetelmässä on se, ettei tunneta keinoa tutkia, mikä kuva vastaa tai on lähimpänä todellisuutta.




Kuva 1: NMDS menetelmän tuottamia ordinaatiokuvia muurahaisaineistolle seuraavilla etäisyysmitoilla ja aineiston muunnoksilla: A. Bray-Curtis-etäisyys alkuperäiselle aineistolle, B. Bray-Curtis-etäisyys Wisconsin-standardoidulle aineistolle, C. Canberra-etäisyys Wisconsin-standardoidulle aineistolle, D. Euklidinen etäisyys logaritmiselle aineistolle.

PCoA- ja DCA-menetelmissä valintatilanteita ei ole yhtä monta kuin moniulotteisen skaalauksen tapauksessa, mutta menetelmät tuottavat jälleen kaksi hyvin erilaista ordinaatiokuvaa (Kuva 2). Nytkään ei voi mitenkään testata, kuinka hyvin tai huonosti menetelmät toimivat kyseessä olevalla aineistolla.




Kuva 2: Vasemmalla PCoA menetelmän tuottama ordinaatiokuva muurahaisaineistolle, ja oikealla vastaava DCA-menetelmällä.

# 3 Latentti muuttujamalli

Määritellään latentti muuttujamalli ordinaatiomenetelmälle kuten Hui et al. (2014) artikkelissaan sen esittelevät. Mallinnetaan vastemuuttujia $y_{ij}$, $i = 1, \ldots, n$, $j = 1, \ldots, p$, yleistetyllä lineaarisella latenttien muuttujien mallilla. Vaste voi olla esimerkiksi tietyn lajin $j$ lukumäärähavainto mittauspaikalla $i$ tai indikaattori, havaittiinko lajia kyseisellä paikalla. Vastemuuttujat ehdolla latentit muuttujat oletetaan noudattavan eksponentiaalisen perheen jakaumaa odotusarvolla $E(y_{ij}) = \mu_{ij}$. Odotusarvoa muuttujalle $j$ paikassa $i$ selitetään $q$:lla latentilla muuttujalla

$$g_j(\mu_{ij}) = \alpha_i + \beta_j + \boldsymbol{z}_i'\boldsymbol{\gamma}_j, \quad i = 1, \ldots, n, \; j = 1, \ldots, p, \tag{2}$$

missä $\boldsymbol{\gamma}_j' = [\gamma_{j1}, \ldots, \gamma_{jq}]$, $\boldsymbol{z}_i$ on $q$-dimensioinen latenttien muuttujien arvojen vektori paikassa $i$ ja $g_j(\cdot)$ on linkkifunktio. Oletetaan, että $\boldsymbol{z}_i \sim N_q(0, \boldsymbol{I})$ ja muuttujat $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$ ovat riippumattomia. Mallissa $\alpha_i$ kuvaa paikkakohtaista tasoa paikalla $i$ ja $\beta_j$ muuttujakohtaista tasoa muuttujalle $j$. Kerroin $\boldsymbol{\gamma}_j$ kuvaa lajin $j$ yhteyttä latentteihin muuttujiin $\boldsymbol{z}_i$. Ordinaatiomenetelmillä käytämme kahta latenttia muuttujaa ($q = 2$), jotka toisiaan vasten kuvattuna tuottavat halutun ordinaatiokuvan. Tällöin latenttien muuttujien arvot $\boldsymbol{z}_i' = (z_{i1}, z_{i2})$ vastaavat paikan $i$ koordinaatteja ordinaatiokuvassa. Voidaan olla kiinnostuneita myös siitä, mitkä lajit ovat paikkojen suhteen samankaltaisia. Tällöin piirretään lajeille ordinaatiokuva, jossa lajin $j$ koordinaatteja vastaa arvot $\boldsymbol{\gamma}_j$. Tätä tapausta ei kuitenkaan nyt käsitella tarkemmin.

Sammel et al. (1997) esittivät artikkelissaan latentin muuttujamallin diskreeteille ja jatkuville vasteille. Ehdotettu malli mahdollisti myös taustamuuttujien mukanaolon. Jos aineistossa on taustamuuttujia käytettävissä, ne voidaan myös tässä lisätä latenttiin muuttujamalliin (2), joka saa silloin muodon

$$g(\mu_{ij}) = \alpha_i + \beta_j + \boldsymbol{z}_i'\boldsymbol{\gamma}_j + \boldsymbol{x}_i'\boldsymbol{\delta}_j, \quad i = 1, \ldots, n, \; j = 1, \ldots, p, \tag{3}$$

missä $\boldsymbol{\delta}_j = [\delta_{j1}, \ldots, \delta_{jm}]'$ ja $\boldsymbol{x}_i = [x_{i1}, \ldots, x_{im}]'$ on paikan $i$ taustamuuttujien arvot. Kertoimet $\boldsymbol{\delta}_j$ kuvaavat taustamuuttujien vaikutusta lajin $j$ odotusarvoon.

Vasteen ehdollinen jakauma voidaan kirjoittaa yleisessä eksponentiaalisen perheen jakauman muodossa

$$f_j(y_{ij}|\boldsymbol{z}_i) = \exp\Big\{y_{ij}\, a_j(\mu_{ij}) + b_j(\mu_{ij}) + c_j(y_{ij})\Big\}, \tag{4}$$

missä funktiot $a_j(\cdot)$, $b_j(\cdot)$ ja $c_j(\cdot)$ riippuvat valitusta jakaumasta (Dobson, 2002) ja $\mu_{ij} = g_j^{-1}(\alpha_i + \beta_j + \boldsymbol{z}_i'\boldsymbol{\gamma}_j)$.

Mallipohjaisella lähestymistavalla on merkittäviä etuja verrattuna klassisiin ordinaatiomenetelmiin. Sovittamalla latentti muuttujamalli aineistoon voidaan tutkia mallin sopivuutta aineistoon sekä keskiarvo-varianssi -suhteen oikeellisuutta jäännösanalyysin avulla. Mallin oletusten voimassaoloa voidaan tutkia esimerkiksi kuvaamalla jäännökset ja ennusteet toisiaan vasten. Jäännöstarkastelujen perusteella voidaan selvittää, onko aineistossa ylihajontaa, linkkifunktion sopivuutta sekä mahdollisten poikkeavien havaintojen olemassaoloa. Jäännöstarkastelujen lisäksi latentti muuttujamalli antaa työkaluja mallinvalintaan sekä hypoteesien testaukseen (ks. luku 4.7).

## 3.1 Dikotominen vaste

Oletetaan dikotominen vaste $y_{ij} \in \{0, 1\}$, jolle $y_{ij} \sim \text{Bernoulli}(\mu_{ij})$, missä

$$\mu_{ij} = P(y_{ij} = 1) = E(y_{ij}).$$

Linkkifunktiona $g_j(\cdot)$ käytetään logit-funktiota. Eksponentiaalisen perheen esitys (4) Bernoullin jakaumalle parametrilla $\mu_{ij}$ saadaan, kun asetetaan

$$a_j(\mu_{ij}) = \log\left(\frac{\mu_{ij}}{1-\mu_{ij}}\right), \quad b_j(\mu_{ij}) = \log(1-\mu_{ij}), \quad c_j(y_{ij}) = 0.$$

## 3.2 Lukumäärävaste

Lukumäärävasteet oletetaan usein Poisson-jakautuneiksi, $y_{ij} \sim \text{Poisson}(\mu_{ij})$, ja linkkifunktioksi valitaan logaritmifunktio. Poisson-jakauma saadaan kaavasta (4), kun tehdään valinnat

$$a_j(\mu_{ij}) = \log \mu_{ij}, \quad b_j(\mu_{ij}) = -\mu_{ij}, \quad c_j(y_{ij}) = -\log(y_{ij}!).$$

Poisson-jakautuneen vastemuuttujan varianssille pätee $Var(y_{ij}) = \mu_{ij}$, mutta ekologian aineistojen tapauksessa tämä on usein epärealistinen oletus. Jos aineiston hajonta on suurta suhteessa odotusarvoon, voidaan käyttää negatiivista binomijakaumaa, jonka varianssi on $Var(y_{ij}) = \mu_{ij} + \phi_j \mu_{ij}^2$, missä $\mu_{ij}$ on vastemuuttujan odotusarvo ja $\phi_j$ on lajikohtainen dispersioparametri. Negatiivinen binomijakauma ei sellaisenaan ole eksponentiaalisen perheen jakauma, mutta jos ajatellaan dispersioparametri $\phi_j$ kiinteänä, sille saadaan eksponentiaalisen perheen esitys valitsemalla

$$\begin{aligned}
a_j(\mu_{ij}) &= \log\left(\frac{\mu_{ij}}{1/\phi_j + \mu_{ij}}\right), \\
b_j(\mu_{ij}) &= -\frac{1}{\phi_j}\log(1 + \phi_j \mu_{ij}), \\
c_j(y_{ij}) &= \log\left(\frac{\Gamma(y_{ij} + 1/\phi_j)}{y_{ij}!\,\Gamma(1/\phi_j)}\right).
\end{aligned}$$

# 4 Implementointi

## 4.1 Latentin muuttujamallin uskottavuusfunktio

Tässä työssä tarkastellaan mallin (2) parametrien $\boldsymbol{\theta} = \{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\phi}\}$ estimointia suurimman uskottavuuden menetelmällä. Jos käytetään mallia (3), $\boldsymbol{\theta} = \{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}, \boldsymbol{\phi}\}$, prosessi etenee vastaavasti. Oletetaan, että muuttujat $y_{ij}$ ovat ehdollisesti riippumattomia ehdolla latentit muuttujat, jolloin niiden yhteisjakauma on

$$\prod_{j=1}^{p} f_j(y_{ij}|\boldsymbol{z}_i; \boldsymbol{\theta})$$

missä $f_j(y_{ij}|\boldsymbol{z}_i; \boldsymbol{\theta})$ on muuttujan $y_{ij}$ ehdollinen tiheysfunktio eksponentiaalisesta jakaumaperheestä. Koska latenttien muuttujien oletettiin olevan standardinormaalijakautuneita ja riippumattomia, niiden yhteisjakauman tiheysfunktio $h(\boldsymbol{z}_i)$ on moniulotteinen normaalijakauma odotusarvolla $\boldsymbol{\mu} = 0$ ja kovarianssimatriisilla $\Sigma = \boldsymbol{I}_q$, eli

$$h(\boldsymbol{z}_i) = \frac{1}{(2\pi)^{\frac{q}{2}}} \exp\left(-\frac{1}{2}\boldsymbol{z}_i'\boldsymbol{z}_i\right).$$

Vastemuuttujien $\boldsymbol{y}_i$ ja latenttien muuttujien $\boldsymbol{z}_i$ yhteisjakauma on siis

$$f(\boldsymbol{y}_i, \boldsymbol{z}_i) = \prod_{j=1}^{p} f_j(y_{ij}|\boldsymbol{z}_i; \boldsymbol{\theta}) h(\boldsymbol{z}_i).$$

Latenttien muuttujien arvoja $\boldsymbol{z}_i$ ei havaita, mutta niiden jakauma tunnetaan. Vastemuuttujien marginaalijakauma saadaan integroimalla latenttien muuttujien yli, ts.

$$f_{\boldsymbol{\theta}}(\boldsymbol{y}_i) = \int \left[\prod_{j=1}^{p} f_j(y_{ij}|\boldsymbol{z}_i; \boldsymbol{\theta})\right] h(\boldsymbol{z}_i)\, d\boldsymbol{z}_i, \tag{5}$$

missä $\boldsymbol{y}_i = [y_{i1}, \ldots, y_{ip}]'$, $\boldsymbol{\theta} = \{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\phi}\}$, $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n]'$, $\boldsymbol{\beta} = [\beta_i, \ldots, \beta_p]'$, $\boldsymbol{\gamma} = [\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_p]$ ja $\boldsymbol{\phi} = [\phi_1, \ldots, \phi_p]'$.

Sekä paikkojen havainnot $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$ että latentit muuttujat $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$ ovat riippumattomia, joten uskottavuusfunktio on marginaalijakaumien (5) tulo. Logaritminen uskottavuusfunktio saa tällöin muodon

$$l(\boldsymbol{\theta}; \boldsymbol{y}) = \tag{6}$$
$$\sum_{i=1}^{n} \log \int \left[\prod_{j=1}^{p} \exp\left\{y_{ij}\, a_j(\mu_{ij}) + b_j(\mu_{ij}) + c_j(y_{ij})\right\}\right] \frac{1}{(2\pi)^{\frac{q}{2}}} \exp\left(-\frac{1}{2}\boldsymbol{z}_i'\boldsymbol{z}_i\right) d\boldsymbol{z}_i.$$

Tavoitteena on siis maksimoida logaritminen uskottavuusfunktio (6) parametrien $\boldsymbol{\theta}$ suhteen. Tämän ratkaiseminen analyyttisesti on mahdotonta, joten käytetään apuna Laplacen approksimaatiota integraalille (5).

## 4.2 Laplacen approksimaatio

Esitellään Laplacen approksimaation teoriaa yksi- ja moniulotteisissa tilanteissa mukaellen kirjaa *Statistical Models* (Davison, 2003). Tarkastellaan ensin yksiulotteista integraalia

$$I_n = \int e^{-nh(u)}\, du, \tag{7}$$

missä $h(u)$ on sileä konveksi funktio, jolla on minimi pisteessä $u = \tilde{u}$. Tällöin sen derivaatoille pätee $dh(\tilde{u})/du = 0$ ja $d^2h(\tilde{u})/du^2 > 0$. Merkitään $h_k = d^k h(\tilde{u})/du^k$, $k = 2, \ldots$. Taylorin sarja funktiolle $h$ määritellään

$$T(u) = h(\tilde{u}) + \sum_{k=1}^{\infty} \frac{h_k}{k!}(u - \tilde{u})^k. \tag{8}$$

Kun $u$ on lähellä minimipistettä $\tilde{u}$, Taylorin sarja antaa approksimaation

$$h(u) \doteq h(\tilde{u}) + \frac{1}{2}h_2(u - \tilde{u})^2. \tag{9}$$

Sijoitetaan tämä integraaliin (7), jolloin saadaan sille approksimaatio

$$\begin{aligned} I_n &\doteq \exp(-nh(\tilde{u})) \int_{-\infty}^{\infty} \exp(-nh_2(u - \tilde{u})^2/2)\, du \\ &= \exp(-nh(\tilde{u})) \int_{-\infty}^{\infty} \exp(-z^2/2) \left|\frac{du}{dz}\right| dz \\ &= \left(\frac{2\pi}{nh_2}\right)^{1/2} \exp(-nh(\tilde{u})). \end{aligned}$$

Ensimmäisen yhtäsuuruuden kohdalla tehdään muuttujanvaihto

$$z = (nh_2)^{1/2}(u - \tilde{u}),$$

ja toinen yhtäsuuruus toteutuu, koska normaalijakauman tiheysfunktio integroituu arvoon yksi. Tarkemmin

$$I_n = \left(\frac{2\pi}{nh_2}\right)^{1/2} \exp(-nh(\tilde{u})) \times \{1 + O(n^{-1})\},$$

missä $O(n^{-1})$ on termi, joka lähestyy nollaa samassa suhteessa termin $n^{-1}$ kanssa, kun $n$ kasvaa. Laplacen approksimaatio integraalille $I_n$ on siis

$$\tilde{I}_n = \left(\frac{2\pi}{nh_2}\right)^{1/2} \exp(-nh(\tilde{u})). \tag{10}$$

Moniulotteisen integraalin tapauksessa

$$I_n = \int e^{-nh(\boldsymbol{u})}\, d\boldsymbol{u}, \tag{11}$$

$\boldsymbol{u}$ on $q$-vektori, $h(\boldsymbol{u})$ on jälleen sileä konveksi funktio ja funktion $h(\boldsymbol{u})$ minimi saavutetaan kohdassa $\boldsymbol{u} = \tilde{\boldsymbol{u}}$. Tällöin $\tilde{\boldsymbol{u}}$ toteuttaa yhtälön

$$\frac{\partial h(\tilde{\boldsymbol{u}})}{\partial \boldsymbol{u}} = 0.$$

ja

$$h_2 := \left.\frac{\partial^2 h(\boldsymbol{u})}{\partial \boldsymbol{u} \partial \boldsymbol{u}'}\right|_{\boldsymbol{u}=\tilde{\boldsymbol{u}}}$$

on positiivisesti definiitti.

Merkitään $\boldsymbol{u}' = [u_1, \ldots, u_q]$. Kun $\boldsymbol{u}$ kuuluu arvon $\tilde{\boldsymbol{u}}$ ympäristöön, funktiolle $h$ saadaan Taylorin sarjasta approksimaatio

$$h(\boldsymbol{u}) \doteq h(\tilde{\boldsymbol{u}}) + \frac{1}{2!}(\boldsymbol{u} - \tilde{\boldsymbol{u}})' h_2 (\boldsymbol{u} - \tilde{\boldsymbol{u}}).$$

Kuten edellä, moniulotteisen normaalijakauman tiheyttä hyödyntäen saadaan integraalille (11) Laplacen approksimaatio

$$\tilde{I}_n = \left(\frac{2\pi}{n}\right)^{q/2} |h_2|^{-1/2} \exp(-nh(\tilde{\boldsymbol{u}})), \tag{12}$$

missä $|h_2|$ on Hessian-matriisin $h_2$ determinantti.

## 4.3 Laplacen approksimaatio latenttien muuttujien mallin uskottavuusfunktiolle

Esitetään nyt Laplacen approksimaation sovellus latenttien muuttujien mallille (Huber et al., 2004). Marginaalijakauman (5) voi kirjoittaa muodossa

$$f_{\boldsymbol{\theta}}(\boldsymbol{y}_i) = \int e^{p \cdot Q(\boldsymbol{\theta}, \boldsymbol{z}_i, \boldsymbol{y}_i)}\, d\boldsymbol{z}_i, \tag{13}$$

missä

$$Q(\boldsymbol{\theta}, \boldsymbol{z}_i, \boldsymbol{y}_i) = \frac{1}{p}\left[\sum_{j=1}^{p}\Big(y_{ij}\, a_j(\mu_{ij}) + b_j(\mu_{ij}) + c_j(y_{ij})\Big) - \frac{\boldsymbol{z}_i'\boldsymbol{z}_i}{2} - \frac{q}{2}\log(2\pi)\right]. \tag{14}$$

Soveltamalla Laplacen approksimaatiota saamme

$$f_{\boldsymbol{\theta}}(\boldsymbol{y}_i) = \left(\frac{2\pi}{p}\right)^{q/2} (\det(-\mathbf{U}(\hat{\boldsymbol{z}}_i)))^{-1/2} \exp(pQ(\boldsymbol{\theta}, \hat{\boldsymbol{z}}_i, \boldsymbol{y}_i))(1 + O(p^{-1})),$$

missä

$$\mathbf{U}(\hat{\boldsymbol{z}}_i) = \left.\frac{\partial^2 Q(\boldsymbol{\theta}, \boldsymbol{z}_i, \boldsymbol{y}_i)}{\partial \boldsymbol{z}_i' \partial \boldsymbol{z}_i}\right|_{\boldsymbol{z}_i = \hat{\boldsymbol{z}}_i} = -\frac{1}{p}\boldsymbol{\Gamma}(\boldsymbol{\theta}, \hat{\boldsymbol{z}}_i),$$

$$\boldsymbol{\Gamma}(\boldsymbol{\theta}, \boldsymbol{z}_i) = \sum_{j=1}^{p} \frac{\partial^2 \left(-y_{ij}\, a_j(\mu_{ij}) - b_j(\mu_{ij})\right)}{\partial \boldsymbol{z}_i' \partial \boldsymbol{z}_i} + \boldsymbol{I}_q \tag{15}$$

ja $\hat{\boldsymbol{z}}_i$ on $Q$-funktion maksimi latenttien muuttujien suhteen.

Laplacen approksimaatio logaritmiselle uskottavuusfunktiolle (6) on

$$\tilde{l}(\boldsymbol{\theta}; \boldsymbol{y}) = \sum_{i=1}^{n} \left( -\frac{1}{2} \log \det \left(\boldsymbol{\Gamma}(\boldsymbol{\theta}, \hat{\boldsymbol{z}}_i)\right) + \sum_{j=1}^{p} \Big( y_{ij}\, a_j(\mu_{ij}) + b_j(\mu_{ij}) + c_j(y_{ij}) \Big) - \frac{\hat{\boldsymbol{z}}_i' \hat{\boldsymbol{z}}_i}{2} \right). \tag{16}$$

Maksimoidaan logaritminen uskottavuusfunktio quasi-Newton -menetelmällä (Broyden, 1967). Tätä varten lasketut derivaatat muuttujien $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$ ja $\boldsymbol{\phi}$ suhteen löytyvät liitteestä B. Asetetaan rajoite $\gamma_{12} = 0$, jolloin $\boldsymbol{\gamma}$ määräytyy yksikäsitteisesti, lukuunottamatta rivien etumerkkejä. Laskentaprosessi etenee seuraavasti:

1. Valitaan alkuarvot parametreille $\boldsymbol{\theta}$ ja latenteille muuttujille $\boldsymbol{z}_i$.
2. Maksimoidaan uskottavuus (16) malliparametrien $\{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}\}$ suhteen.
3. Maksimoidaan uskottavuus (16) dispersioparametrien $\boldsymbol{\phi}$ suhteen.
4. Maksimoidaan $Q$-funktio latenttien muuttujien $\boldsymbol{z}_i$, $i = 1, \ldots, n$, suhteen.
5. Toistetaan kohtia 2-4 kunnes uskottavuusfunktion (16) arvo ei enää muutu.

Neljäs kohta voidaan vaihtoehtoisesti toteuttaa ennustamalla latenttien muuttujien $\boldsymbol{z_i}$ arvot MAP-menetelmällä (*modal/maximum a posteriori*), (Skrondal ja Rabe-Hesketh, 2004, luku 7). Mallin parametrien varianssit saadaan tarvittaessa numeerisesti lasketun Hessian-matriisin käänteismatriisista. Seuraavaksi lasketaan Laplacen approksimaatiot uskottavuusfunktiolle, kun vastemuuttujat ovat Bernoulli-, Poisson- tai negatiivibinomijakautuneita. Liitteessä C on toteutettu näiden mallien implementointi ja estimointi R-kielellä.

## 4.4 Bernoulli-jakautuneet vastemuuttujat

Vastemuuttujille oletetaan $y_{ij}|\boldsymbol{z}_i \sim \text{Bernoulli}(\mu_{ij})$. Linkkifunktioksi $g(\cdot)$ valitaan logit-funktio, jolloin latentti muuttujamalli (2) on

$$\text{logit}(\mu_{ij}) = \alpha_i + \beta_j + \boldsymbol{z}_i' \boldsymbol{\gamma}_j, \quad i = 1, \ldots, n, \; j = 1, \ldots, p.$$

Nyt vastemuuttujan odotusarvolle pätee

$$\mu_{ij} = \frac{\exp(\alpha_i + \beta_j + \boldsymbol{z}_i' \boldsymbol{\gamma}_j)}{1 + \exp(\alpha_i + \beta_j + \boldsymbol{z}_i' \boldsymbol{\gamma}_j)}.$$

Muuttujan $y_{ij}$ ehdollinen tiheysfunktio ehdolla $\boldsymbol{z}_i$ on

$$f(y_{ij}|\boldsymbol{z}_i;\boldsymbol{\theta}) = \exp\left(y_{ij}(\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j) - \log(1+\exp(\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j))\right).$$

Bernoullijakautuneiden vasteiden tapauksessa funktio (14) on

$$Q(\boldsymbol{\theta},\boldsymbol{z}_i,\boldsymbol{y}_i) = \\ \frac{1}{p}\left[\sum_{j=1}^{p}\left[y_{ij}(\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j) - \log\left(1+\exp(\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j)\right)\right] - \frac{\boldsymbol{z}_i'\boldsymbol{z}_i}{2} - \frac{q}{2}\log(2\pi)\right].$$

Lasketaan gammamatriisi:

$$\begin{aligned}
\boldsymbol{\Gamma}(\boldsymbol{\theta},\boldsymbol{z}_i) &= \sum_{j=1}^{p}\frac{\partial^2(-y_{ij}(\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j)+\log\left(1+e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j}\right))}{\partial\boldsymbol{z}_i'\partial\boldsymbol{z}_i} + \boldsymbol{I}_q \\
&= \sum_{j=1}^{p}\frac{\partial}{\partial\boldsymbol{z}_i'}\left(-y_{ij}\boldsymbol{\gamma}_j + \left(1+e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j}\right)^{-1}e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j}\boldsymbol{\gamma}_j\right) + \boldsymbol{I}_q \\
&= \sum_{j=1}^{p}\frac{e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j}\left(1+e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j}\right)\boldsymbol{\gamma}_j\boldsymbol{\gamma}_j' - \left(e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j}\right)^2\boldsymbol{\gamma}_j\boldsymbol{\gamma}_j'}{\left(1+e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j}\right)^2} + \boldsymbol{I}_q \\
&= \sum_{j=1}^{p}\frac{e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j}}{\left(1+e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j}\right)^2}\boldsymbol{\gamma}_j\boldsymbol{\gamma}_j' + \boldsymbol{I}_q
\end{aligned}$$

Laplacen approksimaatio logaritmiselle uskottavuusfunktiolle on nyt

$$\begin{aligned}
\tilde{l}(\boldsymbol{\theta};\boldsymbol{y}) = \sum_{i=1}^{n}\Bigg(&-\frac{1}{2}\log\det\left(\sum_{j=1}^{p}\frac{\exp(\alpha_i+\beta_j+\hat{\boldsymbol{z}}_i'\boldsymbol{\gamma}_j)}{(1+\exp(\alpha_i+\beta_j+\hat{\boldsymbol{z}}_i'\boldsymbol{\gamma}_j))^2}\boldsymbol{\gamma}_j\boldsymbol{\gamma}_j' + \boldsymbol{I}_q\right) \\
&+\sum_{j=1}^{p}\left[y_{ij}(\alpha_i+\beta_j+\hat{\boldsymbol{z}}_i'\boldsymbol{\gamma}_j) - \log\left(1+\exp(\alpha_i+\beta_j+\hat{\boldsymbol{z}}_i'\boldsymbol{\gamma}_j)\right)\right] - \frac{\hat{\boldsymbol{z}}_i'\hat{\boldsymbol{z}}_i}{2}\Bigg),
\end{aligned}$$

missä $\hat{\boldsymbol{z}}_i$ on $Q$-funktion maksimi.

## 4.5 Poisson-jakautuneet vastemuuttujat

Lasketaan Laplacen approksimaatio uskottavuusfunktiolle, kun oletetaan vastemuuttujille $y_{ij}|\boldsymbol{z}_i \sim \text{Poisson}(\mu_{ij})$. Linkkifunktioksi $g(\cdot)$ valitaan logaritmifunktio, jolloin latentti muuttujamalli (2) saa muodon

$$\log(\mu_{ij}) = \alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j, \;\; i=1,\ldots,n,\; j=1,\ldots,p. \tag{17}$$

Nyt

$$\mu_{ij} = \exp(\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j),$$

ja muuttujan $y_{ij}$ ehdollinen tiheysfunktio ehdolla $\boldsymbol{z}_i$ on

$$f(y_{ij}|\boldsymbol{z}_i) = \exp(y_{ij}(\alpha_i + \beta_j + \boldsymbol{z}_i'\boldsymbol{\gamma}_j) - \exp(\alpha_i + \beta_j + \boldsymbol{z}_i'\boldsymbol{\gamma}_j) - \log(y_{ij}!)). \tag{18}$$

Funktio (14) Poisson-jakautuneille vastemuuttujille on

$$Q(\boldsymbol{\theta}, \boldsymbol{z}_i, \boldsymbol{y}_i) = \tag{19}$$
$$\frac{1}{p}\left[\sum_{j=1}^{p}\left[y_{ij}(\alpha_i + \beta_j + \boldsymbol{z}_i'\boldsymbol{\gamma}_j) - \exp(\alpha_i + \beta_j + \boldsymbol{z}_i'\boldsymbol{\gamma}_j) - \log(y_{ij}!)\right] - \frac{\boldsymbol{z}_i'\boldsymbol{z}_i}{2} - \frac{q}{2}\log(2\pi)\right].$$

Lasketaan gammamatriisi (15):

$$\begin{aligned}\boldsymbol{\Gamma}(\boldsymbol{\theta}, \boldsymbol{z}_i) &= \sum_{j=1}^{p} \frac{\partial^2\left(-y_{ij}(\alpha_i + \beta_j + \boldsymbol{z}_i'\boldsymbol{\gamma}_j) + \exp(\alpha_i + \beta_j + \boldsymbol{z}_i'\boldsymbol{\gamma}_j)\right)}{\partial \boldsymbol{z}_i' \partial \boldsymbol{z}_i} + \boldsymbol{I}_q \\ &= \sum_{j=1}^{p} \exp(\alpha_i + \beta_j + \boldsymbol{z}_i'\boldsymbol{\gamma}_j)\boldsymbol{\gamma}_j\boldsymbol{\gamma}_j' + \boldsymbol{I}_q.\end{aligned}$$

Laplacen approksimaatio logaritmiselle uskottavuusfunktiolle on nyt

$$\begin{aligned}\tilde{l}(\boldsymbol{\theta}; \boldsymbol{y}) = \sum_{i=1}^{n}\Bigg( &-\frac{1}{2}\log\det\left(\sum_{j=1}^{p}\exp(\alpha_i + \beta_j + \hat{\boldsymbol{z}}_i'\boldsymbol{\gamma}_j)\boldsymbol{\gamma}_j\boldsymbol{\gamma}_j' + \boldsymbol{I}_q\right) \\ &+ \sum_{j=1}^{p}\left[y_{ij}(\alpha_i + \beta_j + \hat{\boldsymbol{z}}_i'\boldsymbol{\gamma}_j) - \exp(\alpha_i + \beta_j + \hat{\boldsymbol{z}}_i'\boldsymbol{\gamma}_j) - \log(y_{ij}!)\right] - \frac{\hat{\boldsymbol{z}}_i'\hat{\boldsymbol{z}}_i}{2}\Bigg),\end{aligned}$$

missä $\hat{\boldsymbol{z}}_i$ on funktion (19) maksimi.

## 4.6 Negatiivibinomijakautuneet vastemuuttujat

Vastemuuttujille oletetaan $y_{ij}|\boldsymbol{z}_i \sim \mathrm{NB}(\mu_{ij}, \phi_j)$, jolloin

$$E(y_{ij}) = \mu_{ij}$$

ja

$$Var(y_{ij}) = \mu_{ij} + \phi_j\mu_{ij}^2.$$

Linkkifunktioksi $g(\cdot)$ valitaan logaritmi-funktio, jolloin latentti muuttujamalli (2) on kuten Poisson-jakauman tilanteessa (17). Muuttujan $y_{ij}$ ehdollinen tiheysfunktio voidaan kirjoittaa muodossa

$$\begin{aligned}f(y_{ij}|\boldsymbol{z}_i) = \exp\Bigg(&y_{ij}(\alpha_i + \beta_j + \boldsymbol{z}_i'\boldsymbol{\gamma}_j) - \left(y_{ij} + \frac{1}{\phi_j}\right)\log\left(1 + \phi_j\exp(\alpha_i + \beta_j + \boldsymbol{z}_i'\boldsymbol{\gamma}_j)\right) \\ &+ y_{ij}\log\phi_j + \log\Gamma\left(y_{ij} + \frac{1}{\phi_j}\right) - \log y_{ij}! - \log\Gamma\left(\frac{1}{\phi_j}\right)\Bigg).\end{aligned}$$

Funktio (14) on negatiiviselle binomijakaumalle

$$Q(\boldsymbol{\theta}, \boldsymbol{z}_i, \boldsymbol{y}_i) = \frac{1}{p}\left[\sum_{j=1}^{p} \log f(y_{ij}|\boldsymbol{z}_i) - \frac{\boldsymbol{z}_i'\boldsymbol{z}_i}{2} - \frac{q}{2}\log(2\pi)\right].$$

Lasketaan gammamatriisi:

$$\begin{aligned}
&\boldsymbol{\Gamma}(\boldsymbol{\theta}, \boldsymbol{z}_i) \\
&= \sum_{j=1}^{p} \frac{\partial^2 \left(-y_{ij}(\alpha_i + \beta_j + \boldsymbol{z}_i'\boldsymbol{\gamma}_j) + \left(y_{ij} + \frac{1}{\phi_j}\right)\log\left(1 + \phi_j e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j}\right)\right)}{\partial \boldsymbol{z}_i' \partial \boldsymbol{z}_i} + \boldsymbol{I}_q \\
&= \sum_{j=1}^{p} \left(y_{ij} + \frac{1}{\phi_j}\right) \frac{\partial^2 \log\left(1 + \phi_j e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j}\right)}{\partial \boldsymbol{z}_i' \partial \boldsymbol{z}_i} + \boldsymbol{I}_q \\
&= \sum_{j=1}^{p} \left(y_{ij} + \frac{1}{\phi_j}\right) \frac{\partial}{\partial \boldsymbol{z}_i'} \frac{\phi_j e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j}}{1 + \phi_j e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j}} \boldsymbol{\gamma}_j + \boldsymbol{I}_q \\
&= \sum_{j=1}^{p} \left(y_{ij} + \frac{1}{\phi_j}\right) \frac{\phi_j e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j}(1 + \phi_j e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j} - \phi_j e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j})}{(1 + \phi_j e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j})^2} \boldsymbol{\gamma}_j\boldsymbol{\gamma}_j' + \boldsymbol{I}_q \\
&= \sum_{j=1}^{p} \left(y_{ij} + \frac{1}{\phi_j}\right) \frac{\phi_j e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j}}{(1 + \phi_j e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j})^2} \boldsymbol{\gamma}_j\boldsymbol{\gamma}_j' + \boldsymbol{I}_q.
\end{aligned}$$

Laplacen approksimaatio logaritmiselle uskottavuusfunktiolle on nyt

$$\begin{aligned}
\tilde{l}(\boldsymbol{\theta}; \boldsymbol{y}) = \sum_{i=1}^{n} \Bigg( &- \frac{1}{2}\log\det\left(\sum_{j=1}^{p}\left(y_{ij} + \frac{1}{\phi_j}\right)\frac{\phi_j \exp(\alpha_i + \beta_j + \hat{\boldsymbol{z}}_i'\boldsymbol{\gamma}_j)}{(1 + \phi_j \exp(\alpha_i + \beta_j + \hat{\boldsymbol{z}}_i'\boldsymbol{\gamma}_j))^2}\boldsymbol{\gamma}_j\boldsymbol{\gamma}_j' + \boldsymbol{I}_q\right) \\
&+ \sum_{j=1}^{p}\Bigg[y_{ij}(\alpha_i + \beta_j + \hat{\boldsymbol{z}}_i'\boldsymbol{\gamma}_j) - \left(y_{ij} + \frac{1}{\phi_j}\right)\log\left(1 + \phi_j \exp(\alpha_i + \beta_j + \hat{\boldsymbol{z}}_i'\boldsymbol{\gamma}_j)\right) \\
&+ y_{ij}\log\phi_j + \log\Gamma\left(y_{ij} + \frac{1}{\phi_j}\right) - \log y_{ij}! - \log\Gamma\left(\frac{1}{\phi_j}\right)\Bigg] - \frac{\hat{\boldsymbol{z}}_i'\hat{\boldsymbol{z}}_i}{2}\Bigg).
\end{aligned}$$

## 4.7 Dunn-Smyth residuaalit ja informaatiokriteerit

Mallinvalinnassa käytetään informaatiokriteereitä AIC ja BIC, jotka määritellään

$$\text{AIC} = 2 \cdot \#(\text{parametrit}) - 2 \cdot ll$$

ja

$$\text{BIC} = \log(n) \cdot \#(\text{parametrit}) - 2 \cdot ll,$$

missä $ll$ tarkoittaa logaritmisen uskottavuusfunktion arvoa.

Jäännöstarkasteluja varten määritellään residuaalit. Kun vastemuuttujat eivät ole

normaalijakautuneita, Pearsonin residuaalit ovat usein ei-normaalisia ja niiden varianssit ovat erisuuria. Erityisesti ongelmia ilmenee diskreeteillä vasteilla, kun yksittäisten arvojen määrät ovat pieniä. Esimerkkinä ovat Poisson-jakautuneet vasteet pienellä odotusarvolla. Käytetään siis Dunn-Smyth-residuaaleja riippumattomien vasteiden regressiomalleille (Dunn ja Smyth, 1996). Määritellään havainnolle $(i, j)$ Dunn-Smyth-jäännös

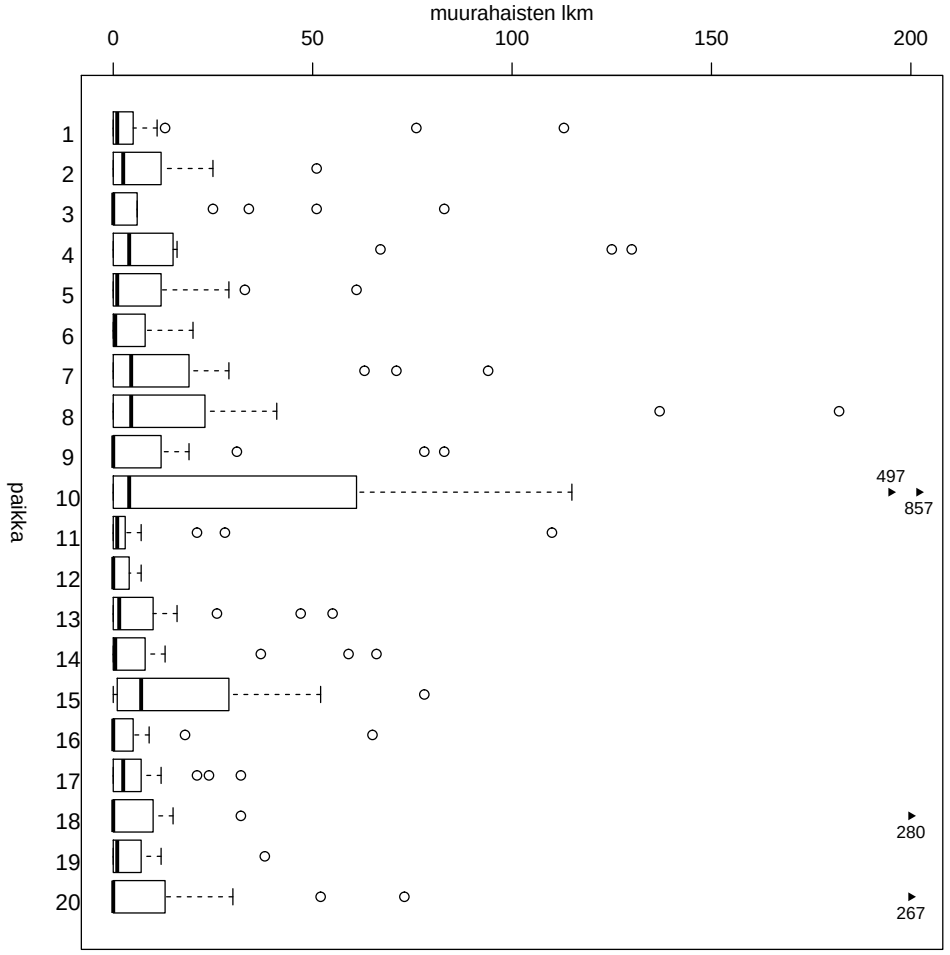
$$r_{ij} = \Phi^{-1}(u_{ij}F_{ij}(y_{ij}; \hat{\mu}_{ij}, \hat{\phi}) + (1 - u_{ij})F_{ij}^{-}(y_{ij}; \hat{\mu}_{ij}, \hat{\phi})), \tag{20}$$

missä $\Phi(\cdot)$ on standardi normaalijakauman ja $F_{ij}(y; \mu_{ij}, \phi)$ muuttujan $y_{ij}$ jakauman kertymäfunktio ja $F_{ij}^{-}(y)$ on raja-arvo, kun lähestytään arvoa $F_{ij}(y)$ negatiiviselta puolelta. Luku $u_{ij}$ on generoitu $(0, 1)$-tasajakaumasta. Dunn-Smyth-residuaalit ovat normaalijakautuneita, kun mallin yhteensopivuus aineiston kanssa on hyvä.

# 5 Sovelluksia

## 5.1 Muurahaiset

Tarkastellaan muurahaisaineistoa, joka sisältää usealta paikalta mitattujen muurahaislajien lukumääriä. Aineisto on kerätty eukalyptusmetsistä Kaakkois-Australiasta (Stoklosa et al., 2014). Alkuperäisessä aineistossa havaintopaikkoja oli 30 ja muurahaislajeja 35, mutta aineistoa pienennettiin siten, että todella suuria lukuja sekä todella paljon nollia sisältävät paikat ja lajit rajattiin pois. Käytettävään aineistoon jäi $n = 20$ havaintopaikkaa ja $p = 18$ muurahaislajia. Kuvassa 3 on piirretty laatikkokuviot jokaisen paikan muurahaisten lukumäärille. Aineistosta on kuvan perusteella vaikea havaita selkeitä ryhmittymiä, mutta paikkoja, joissa eri muurahaislajeja on havaittu pääosin hyvin vähän tai ei ollenkaan, näyttäisi olevan paikat 1, 3, 11, 12 ja



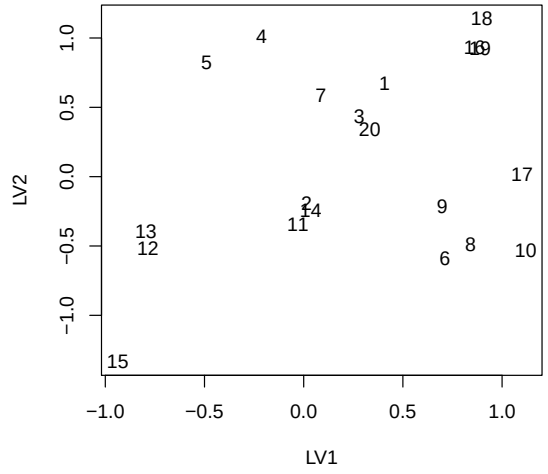

Kuva 3: Muurahaisaineisto. Havaintopaikat on numeroitu 1-20, ja jokaisen paikan havannoista on piirretty laatikkokuvio. Lajihavainnot, jotka ovat suuruudeltaan enemmän kuin 200, on merkitty kuvaan mustalla kolmiolla ja havainnon arvolla niiden vieressä.

16. Runsaammin muurahaisia näyttäisi olevan paikoissa 7, 8, 10 ja 15. Kuvasta ei kuitenkaan nähdä, ovatko lajit, joita on paljon, samoja eri paikoissa.

Sovitetaan latentti muuttujamalli (2) aineistoon sekä Poisson- että negatiivibinomijakaumaoletuksella NMDS-menetelmästä saadut arvot alkuarvoina. Mallinvalintakriteerien tarkastelemiseksi sovitetaan molemmat näistä malleista myös ilman paikkaparametria $\alpha_i$. Merkitään latenttia muuttujamallia Poisson-jakaumaoletuksella LVM-P ja negatiivibinomijakaumaoletuksella LVM-NB. Tutkitaan informaatiokriteerien avulla, mikä edellä mainituista malleista sopii aineistoon parhaiten. Taulukosta 1 nähdään, että sekä AIC- että BIC-arvon perusteella paras malli on sellainen, jossa vasteet ovat negatiivibinomijakautuneita. Lisäksi molemmilla jakaumilla malli on parempi, kun paikkaparametri on mukana. Kuvassa 4 on parhaan mallin ordinaatiokuva muurahaisaineistolle. Kuvasta nähdään, että esimerkiksi paikat 6, $8-10$ ja 17 ovat lähellä toisiaan, jolloin niiden voidaan tulkita olevan lajistoltaan keskenään samankaltaisia. Lisäksi on havaittavissa muutamia pienempiä ryhmiä. Tällaisia ovat esimerkiksi joukko $\{1, 3, 7, 20\}$ ja sen lähellä ryhmät $\{16, 18, 19\}$ ja $\{2, 11, 14\}$ sekä paikat 4 ja 5. Edellä mainituista ryhmistä selkeämmin erottuvat kuitenkin paikat 12 ja 13, jotka ovat lähellä toisiaan, ja paikka 15, joka näyttäisi olevan erillään muista.

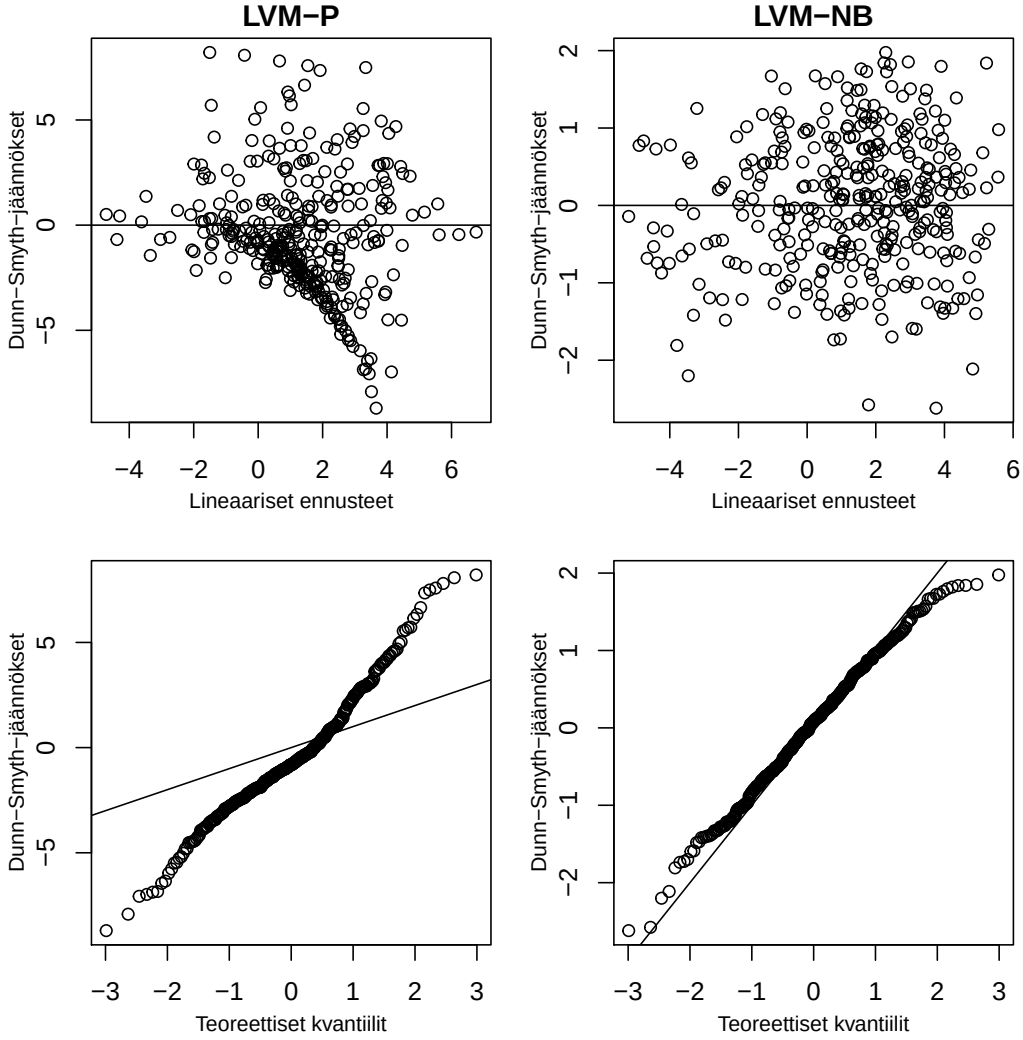
Taulukko 1: AIC- ja BIC-arvot eri jakaumaoletuksilla ja paikkaparametrilla tai ilman muurahaisaineistolle. Pienimmät informaatiokriteerien arvot on tummennettu.

| Menetelmä | Paikkaparametri | AIC | BIC |
|---|---|---|---|
| LVM-P | kyllä | 4679 | 4753 |
| | ei | 5194 | 5248 |
| LVM-NB | kyllä | **2015** | **2106** |
| | ei | 2067 | 2139 |




Kuva 4: Muurahaisaineiston ordinaatiokuva negatiivibinomijakaumaoletuksella.

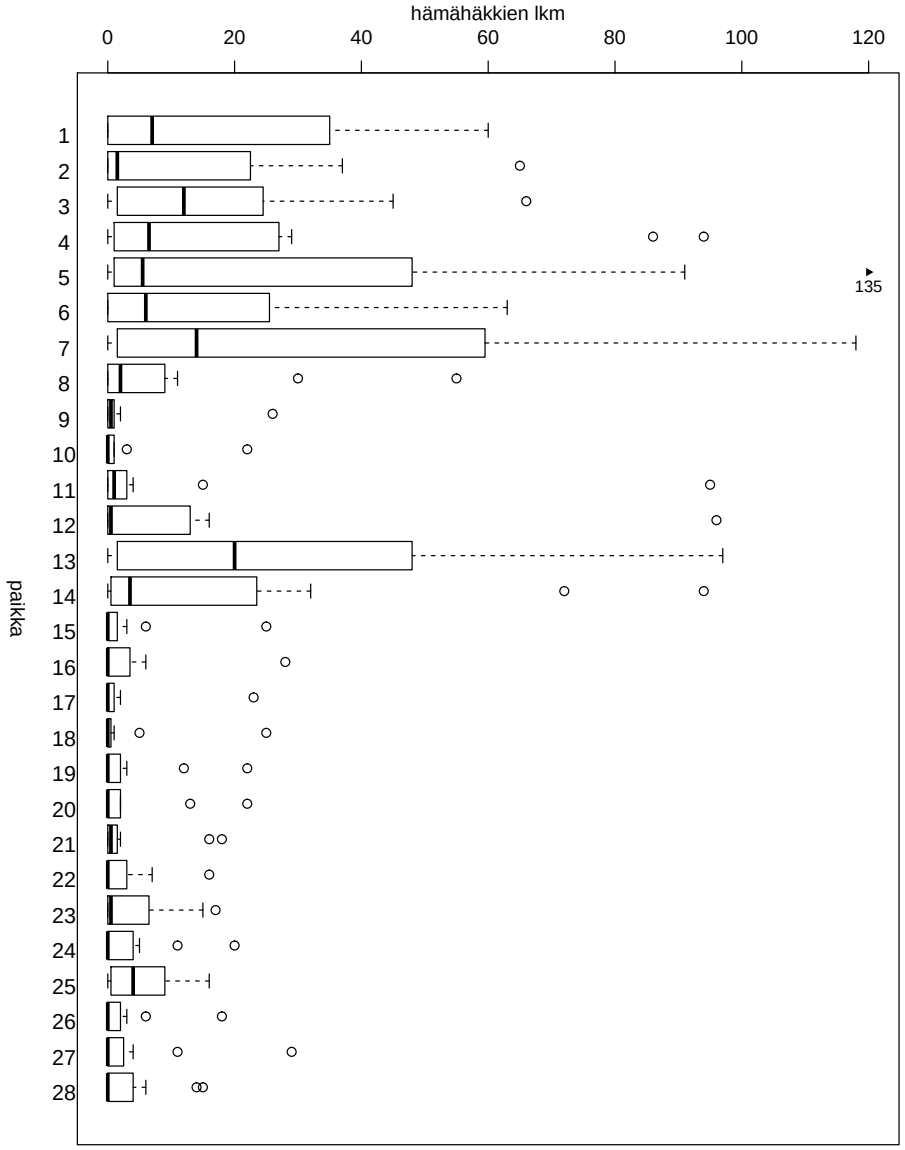
Tarkastellaan vielä eri jakaumaoletuksilla muodostettujen mallien diagnostiikkaa. Muodostetaan jäännöskuvat, joissa Dunn-Smyth-jäännökset on kuvattu lineaaristen ennusteiden suhteen, ja piirretään residuaalien kvantiilikuviot (Kuva 5). Poisson-jakauman jäännöskuvasta voidaan nähdä, että ennusteiden kasvaessa myös jäännökset kasvavat, mikä viittaa ylihajontaan aineistossa. Kvantiilikuvasta puolestaan voidaan sanoa, että jäännökset eivat ole normaalijakautuneita, koska ne eivät lainkaan osu suoralle niin kuin pitäisi. Negatiivibinomijakautuneille vasteille sen sijaan jäännökset näyttävät melko tasaisesti sijoittuvan nollan ympäristöön riippumatta ennusteiden suuruudesta. Myös kvantiilikuviossa jäännökset ovat hyvin suoralla. Nämä havainnot tukevat negatiivibinomijakaumaoletuksen paremmuutta, sillä Dunn-Smyth-jäännökset ovat normaalijakautuneita, kun malli on hyvä.




Kuva 5: Muurahaisaineiston Dunn-Smyth-residuaalit ja näiden kvantiilikuva latentille muuttujamallille Poisson-jakaumaoletuksella (vasen sarake) ja negatiivibinomijakaumaoletuksella (oikea sarake).

## 5.2 Hämähäkit

Tarkastellaan hämähäkkiaineistoa, joka on kerätty Haagissa, Alankomaissa (van der Aart ja Smeenk-Enserink, 1974). Aineistossa on $n = 28$ mittauspaikalta otettu näytteet, joista on laskettu $p = 12$ eri lajin hämähäkkejä. Sovitetaan latentti muuttujamalli hämähäkkiaineistolle, ja tehdään vastaavat tarkastelut kuin muurahaisaineistolle. Kuvassa 6 aineiston kunkin paikan havainnoille on piirretty laatikkokuvio, josta voidaan jo nähdä eroja paikkojen välillä. Esimerkiksi numeroita $1 - 7$ sekä $13 - 14$ vastaavilla paikoilla hämähäkkejä on havaittu huomattavasti enemmän kuin esimerkiksi paikoilla $9 - 11$ ja $15 - 21$, kun paikat on numeroitu $1 - 28$.

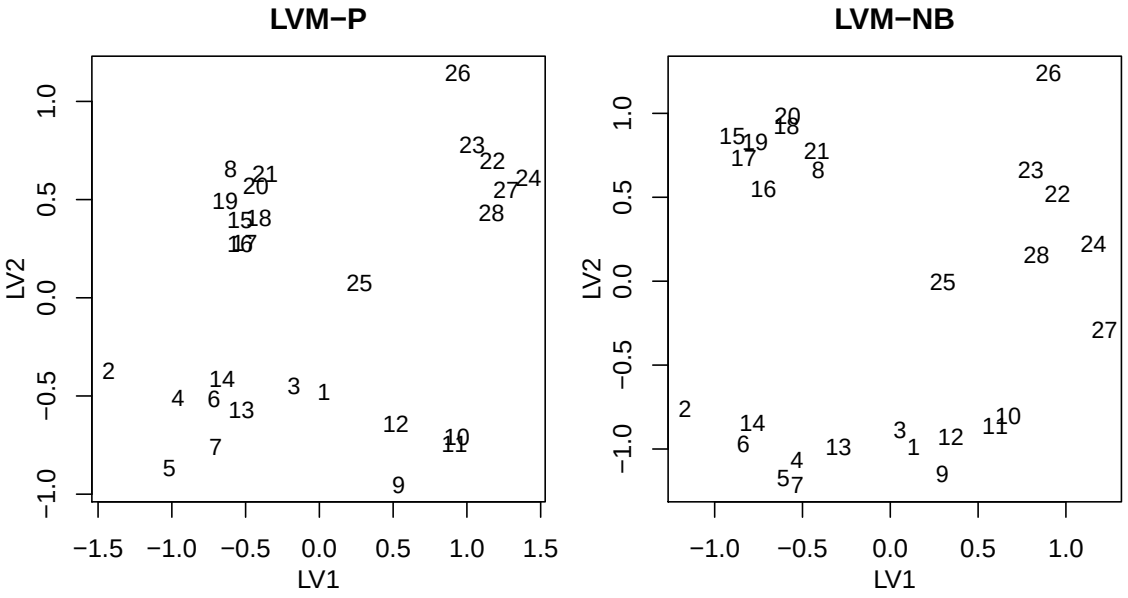



Kuva 6: Hämähäkkiaineisto. Havaintopaikat on numeroitu 1-28, ja jokaisen paikan havainnoista on piirretty laatikkokuvio.

Kun sovitetaan latentti muuttujamalli hämähäkeille, taulukosta 2 nähdään, että sekä AIC- että BIC-kriteerien perusteella negatiivinen binomijakauma on sopivampi jakaumaoletus kuin Poisson-jakauma, ja paikkaparametrin sisältävä malli on parempi. Informaatiokriteerien ero ei kuitenkaan ole yhtä suuri kuin muurahaisaineistolla, joten tarkastellaan sekä LVM-P- että LVM-NB-mallien tuottamia ordinatiokuvia (Kuva 7). Molempien mallien ordinaatiokuvat ovat melko samanlaiset. Poisson-mallilla oikean yläkulman klusteri on hieman tiiviimpi kuin negatiivisen binomijakauman tapauksessa, ja paikkojen $9-12$ muodostama ryhmä erottuu selvemmin. Molemmissa paikka 25 erottuu muista ja paikkojen $15-21$ sekä 8 muodostama ryhmä on selkeästi erillään muista. Kuten jo kuvasta 6 havaittiin, paikat $1-7$ ja $13-14$, joissa havaittiin paljon hämähäkkejä, ovat lähellä toisiaan myös ordinaatiokuvissa.

Taulukko 2: Hämähäkkiaineiston AIC- ja BIC-arvot eri malleille.

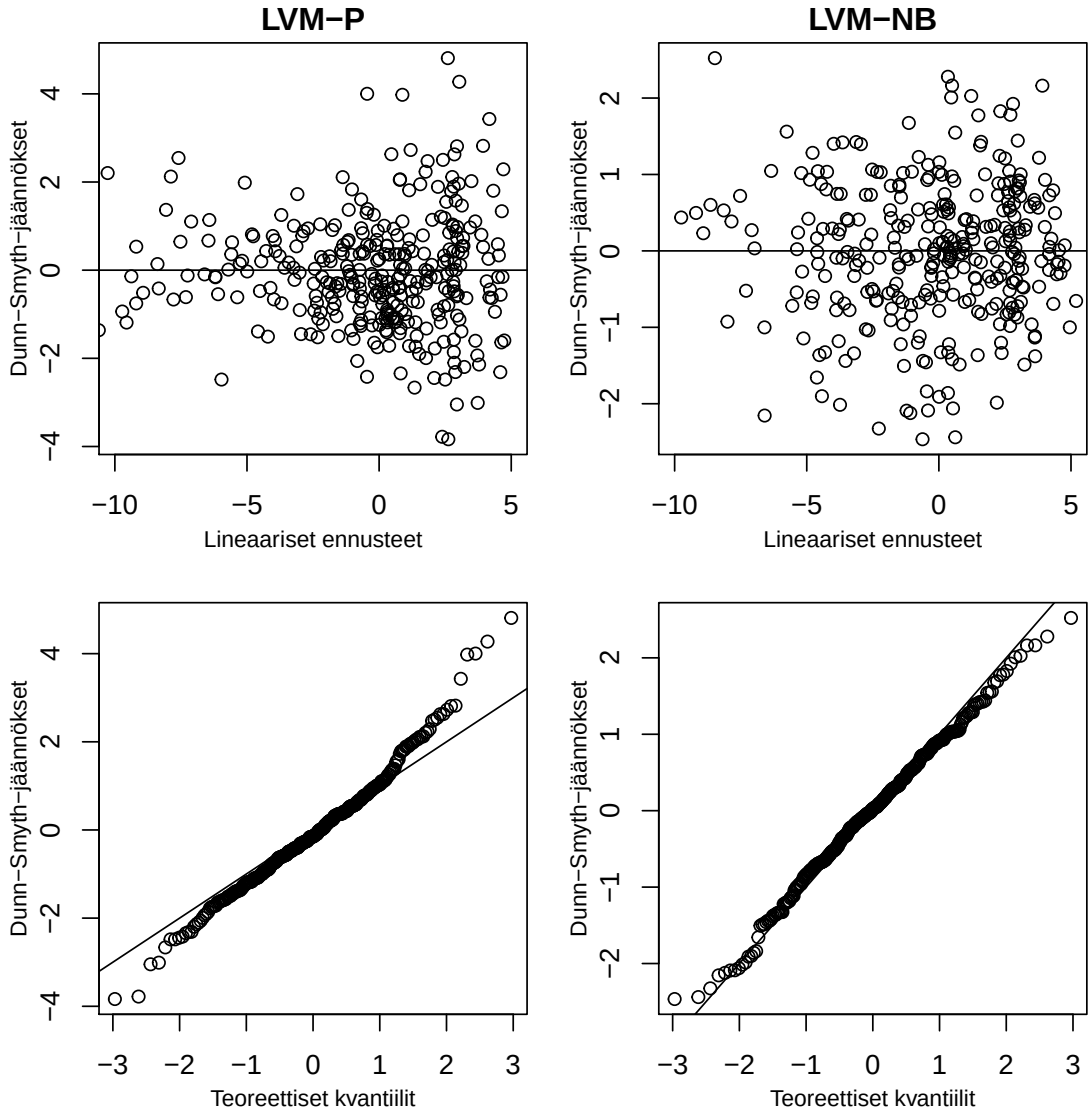
| Menetelmä | Paikkaparametri | AIC | BIC |
|---|---|---|---|
| LVM-P | kyllä | 1691 | 1776 |
| | ei | 1797 | 1845 |
| LVM-NB | kyllä | **1477** | **1578** |
| | ei | 1522 | 1586 |




Kuva 7: Vasemmalla latentin muuttujamallin Poisson-jakaumaoletuksella tuottama hämähäkkiaineiston ordinaatiokuva, ja oikealla vastaava negatiivibinomijakaumaoletuksella.

Tarkastelemalla jäännöskuvia ja kvantiilikuvia (Kuva 8) voidaan tehdä sama pää-

telmä jakaumaoletusten paremmuudesta, kuin informaatikriteerien perusteella. Poisson-jakaumaoletuksella jäännöskuvassa on selvästi havaittavissa viuhkamainen kuvio, eli kun ennusteet ovat suurempia myös jäännösten vaihtelu kasvaa. Kun havaintojen oletetaan olevan negatiivisesta binomijakaumasta, sekä jäännös- että kvantiilikuvio näyttävät huomattavasti paremmalta. Poisson-jakautunut malli sopii huonommin aineistoon, mutta sen tuottamassa ordinaatiokuvassa eri ryhmät erottuvat selkeämmin kuin negatiivibinomijakautuneilla vasteilla. Poisson-jakaumaoletuksella syntyvän ordinaatiokuvan ryhmät saattavat olla virheellisesti liian selkeitä, koska mallissa hajonta on liian pientä verrattuna todelliseen tilanteeseen.




Kuva 8: Dunn-Smyth-jäännökset ja kvantiilikuvio hämähäkkiaineistolle Poisson- (vasen sarake) ja negatiivibinomijakaumaoletuksella (oikea sarake).

Laajennetaan seuraavaksi latenttia muuttujamallia, ja lisätään malliin selittävät muuttujat kuten kaavassa (3). Ordinaatiomenetelmänä käytön lisäksi malli soveltuu nyt selittäjien lajikohtaisten vaikutusten tutkimiseen. Käytetään kahta selittävää muuttujaa, jotka ovat numeerisia muuttujia havaintopaikkojen olosuhteista: muuttuja *soil.dry* kuvaa maaperän kuivamassan määrää ja *reflection* kuvaa maaperän pinnan heijastusta taivaaseen. Sovitetaan malli (3) negatiivibinomijakaumaoletuksella ilman paikkaparametria, ja verrataan sitä tavalliseen yleistettyyn lineaariseen malliin

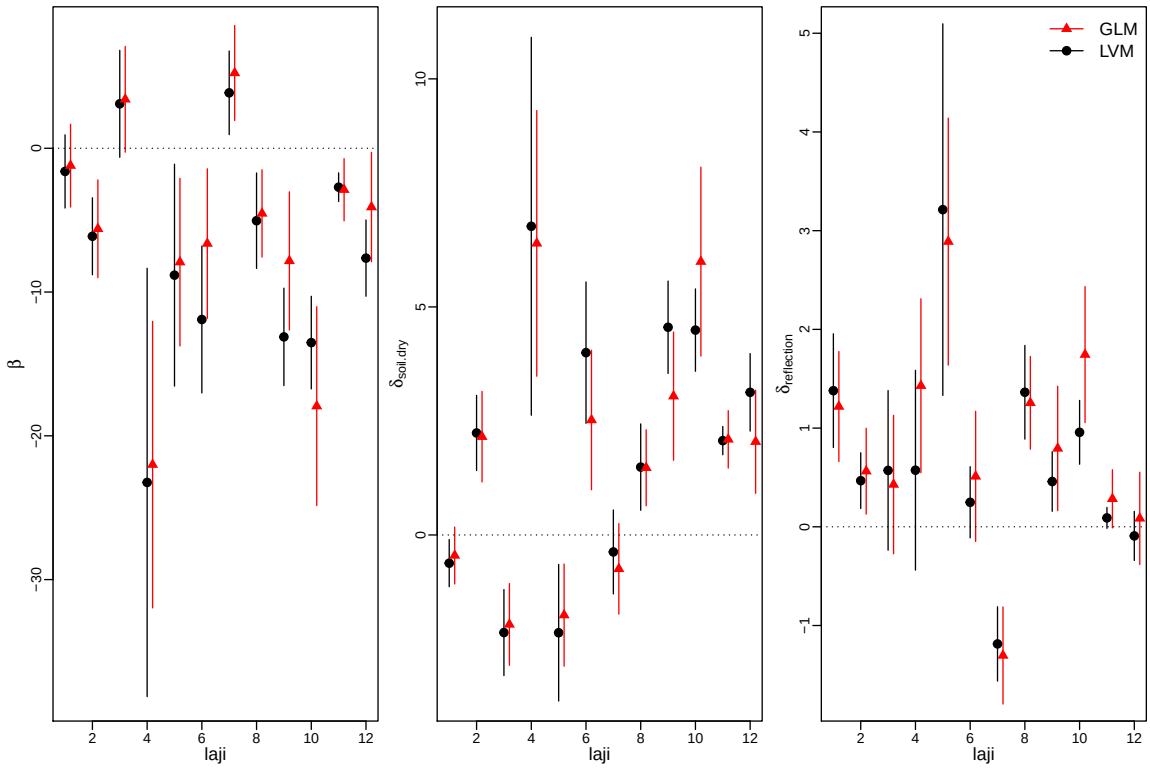
$$g(\mu_{ij}) = \beta_j + \boldsymbol{x}_i'\boldsymbol{\delta}_j, \quad i = 1, \ldots, n, \; j = 1, \ldots, p. \tag{21}$$

Edellinen malli eroaa mallista (3) latenttien muuttujien osalta. Mallissa (21) oletetaan lajien olevan riippumattomia, kun taas latentti muuttujamalli huomioi mahdollisen korrelaation.

Edellä esitetty yleistetty lineaarinen malli, (GLM), negatiivibinomijakaumaoletuksella sovitetaan R-ohjelmiston `mvabund`-paketin `manyglm`-funktiolla. AIC- ja BIC-kriteerien perusteella latentti muuttujamalli on parempi kuin malli (21) (Taulukko 3). Kuvasta 9 nähdään, että lajikohtaisten vakioden $\beta_j$ ja kertoimien $\boldsymbol{\delta}_j$ estimaatit tarkastelluilla malleilla ovat lähes samoja muutamaa lajia lukuunottamatta. Parametrien keskihajonnoissa sen sijaan löytyy eroja. Latentin muuttujamallin parametrien luottamusvälit ovat suurimmalla osalla lajeista selkeästi pienemmät tai yhtä pienet kuin tavallisessa yleistetyssä lineaarisessa mallissa. Ainoastaan lajien neljä ja viisi osalla parametreista luottamusvälit ovat selkeästi suuremmat latentilla muuttujamallilla. Näistä kahdesta lajista myös havaintoja oli hyvin vähän, mikä saattaa olla osasyynä eroille parametrien hajonnassa. Selittäjien lisäämisen johdosta latentteihin muuttujiin sisältyvä havaitsematon informaatio muuttuu, kun osa siitä havaitaan nyt eri paikoilta mitatuissa selittävissä muuttujissa.

Taulukko 3: Hämähäkkiaineiston AIC- ja BIC-arvot selittäjät sisältävälle latentille muuttujamallille ja tavalliselle yleistetylle lineaariselle mallille.

| Menetelmä | AIC | BIC |
|---|---|---|
| GLM-NB | 1542 | 1606 |
| LVM-NB | **1427** | **1523** |




Kuva 9: Latentin muuttujamallin (ympyrät) ja yleistetyn lineaarisen mallin (kolmiot) selittäjien kertoimet lajeittain, kun vasteet oletetaan negatiivibinomijakautuneiksi. Janat ovat parametrien luottamusvälit.

# 6 Simulointikokeita

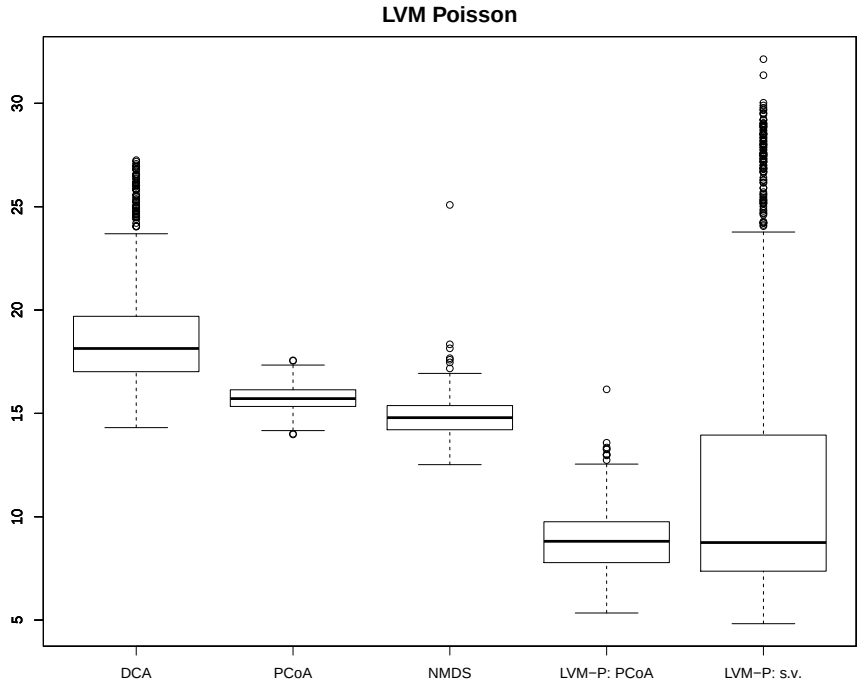
Verrataan seuraavaksi latenttia muuttujamallia klassisiin ordinaatiomenetelmiin simulointikokeiden avulla. Simuloidaan havaintoja hämähäkkiaineistoa vastaavasta latentista muuttujamallista Poisson- ja negatiivibinomijakaumaoletuksilla. Mallin parametrit saatiin sovittamalla latentti muuttujamalli hämähäkkiaineistoon kuten luvussa 5.2. Simulointeja tehtiin jokaiselle mallille 500 kertaa. Kullekin aineistolle sovelletaan klassisia menetelmiä NMDS, PCoA ja DCA sekä sovitetaan latentti muuttujamalli eri alkuarvoilla ordinaatiokuvan muodostamiseksi. Alkuarvoina latenteille muuttujille käytetään yhtä klassisista menetelmistä, sillä ne antoivat hyvin samanlaisia tuloksia testatessa. Lisäksi käytetään satunnaisia alkuarvoja. Ne saadaan, kun ensin generoidaan kaksiulotteisesta standardista normaalijakaumasta $n$ havaintoa, jotka ajatellaan vastaavan koordinaatteja ordinaatiossa. Tämä tehdään kymmenen kertaa, ja kuhunkin simulaatioon sovitetaan yleistetty lineaarinen regressiomalli jokaiselle lajille, selittäjinä nämä koordinaatit. Ne koordinaatit, joilla saadaan mallille pienin AIC-arvo, valitaan latenttien muuttujien alkuarvoiksi. Kyseisen regressiomallin parametreista saadaan alkuarvot myös parametreille. R-koodissa (Liite C) tämän toteuttaa `start.values`-niminen funktio.

Estimoidut ordinaatiokuvat rotatoidaan ja skaalataan todellista ordinaatiokuvaa vastaavaan asetelmaan R-ohjelmiston `vegan`-paketin `procrustes`-funktiolla. Näin estimoitua ordinaatiota verrataan todelliseen ordinaatioon Procrustes-virheen avulla, joka on

$$\text{Procrustes Error} = \sum_{i=1}^{n}\sum_{r=1}^{q}(z_{ir,fitted} - z_{ir,true})^2. \tag{22}$$

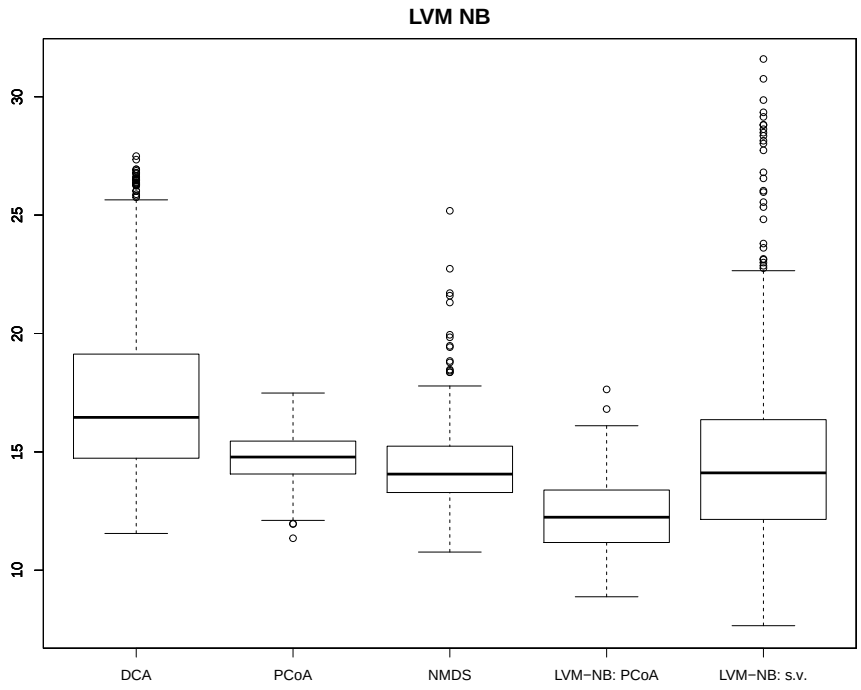
Tässä $z_{ir,fitted}$ on sovitetun mallin Procrustes-rotatoidun ordinaation koordinaatti paikalla $i$ latentille muuttujalle $r$ ja $z_{ir,true}$ on vastaavan koordinaatin ja latentin muuttujan todellinen arvo.

Ensimmäisenä simuloidaan latenttiin muuttujamalliin perustuvasta Poisson-jakaumasta 500 aineistoa. Verrataan klassisia menetelmiä MDS, PCoA ja DCA LVM-P-malliin PCoA ja normaalijakauman alkuarvoilla. Kuvasta 10 havaitaan, että latentti muuttujamalli PCoA menetelmän alkuarvoilla antaa parempia tuloksia kuin mikään klassinen menetelmä. Kun alkuarvoina on normaalijakaumasta generoidut luvut, vaihtelua on hyvin paljon. Enimmäkseen kyllä saavutetaan yhtä hyviä tuloksia kuin PCoA-alkuarvoilla, mutta välillä huonompia tuloksia kuin mikään klassinen menetelmä tuottaa.

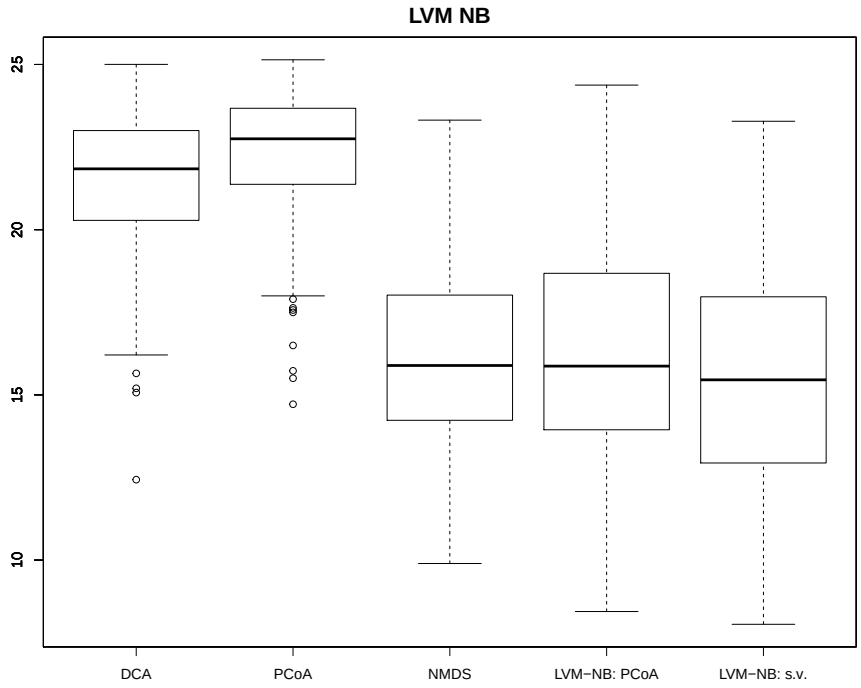



Kuva 10: Hämähäkkiaineiston LVM-P-mallista simuloinnin tuottamien Procrustes-virheiden laatikkokuviot vertailtaville menetelmille. LVM-P: PCoA tarkoittaa LVM-P-menetelmää PCoA-menetelmän antamat ordinaatiopisteet alkuarvoina ja LVM-P: s.v. vastaavaa normaalijakaumasta generoiduilla alkuarvoilla.

Vastaavasti simuloidaan negatiivisesta binomijakaumasta 500 kertaa sekä hämähäkki- että muurahaisaineistoon perustuvista malleista ja verrataan samoja klassisia menetelmiä kuin edellisessä simuloinnissa LVM-NB-malliin eri alkuarvoilla. Kun tarkastellaan hämähäkkiaineistoon perustuvaa simulointia, kuvasta 11 nähdään, että klassisista ordinaatiomenetelmistä DCA tuottaa jonkin verran huonompia tuloksia kuin muut menetelmät. Latentti muuttujamalli PCoA-alkuarvoilla näyttäisi olevan hieman parempi kuin PCoA- ja NMDS-menetelmät. Latentti muuttujamalli normaalijakaumasta generoiduilla alkuarvoilla tuottaa hyvin vaihtelevia tuloksia. Muurahaisaineistoon perustuvassa simuloinnissa latentti muuttujamalli molemmilla alkuarvoilla on parhaiden menetelmien joukossa (Kuva 12). DCA- ja PCoA-menetelmät ovat selkeasti huonompia, mutta tällä kertaa NMDS-menetelmä näyttäisi toimivan yhtä hyvin kuin latentti muuttujamalli.

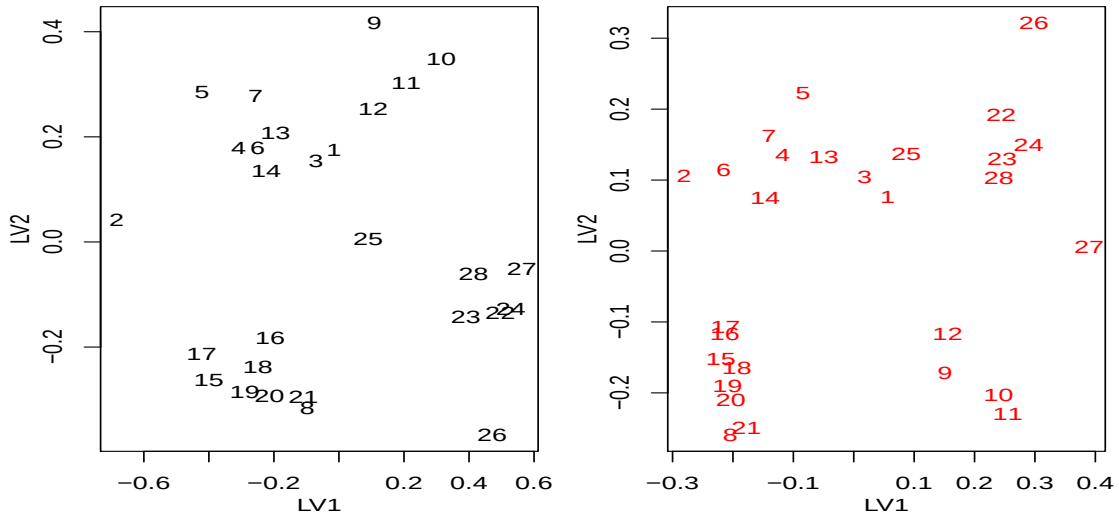



Kuva 11: Hämähäkkiaineiston LVM-NB-mallista simuloinnin tuottamien Procrustes-virheiden laatikkokuviot vertailtaville menetelmille.



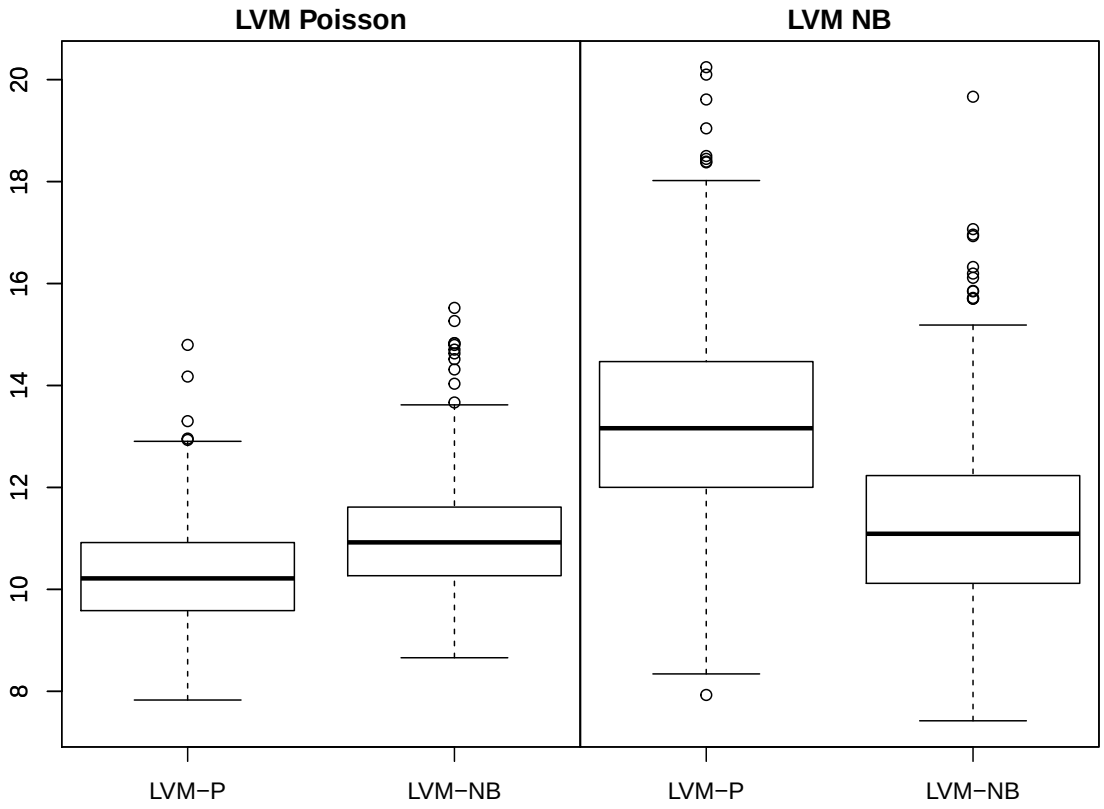

Kuva 12: Muurahaisaineiston LVM-NB-mallista simuloinnin tuottamien Procrustes-virheiden laatikkokuviot vertailtaville menetelmille.

Edellisistä kuvista huomattiin, että etenkin hämähäkkiaineistoon perustuvissa simuloinneissa latentti muuttujamalli normaalijakaumasta generoiduilla alkuarvoilla antaa hyvin vaihtelevia tuloksia. Tämä voisi johtua siitä, että välillä näillä alkuarvoilla päädytään kuvan 13 oikeanpuoleisen ordinaation kaltaiseen tilanteeseen. Molemmissa ordinaatiokuvista on löydettävissä vastaavat ryhmät, mutta niiden sijainnit suhteessa muihin ryhmiin eroavat. Tätä procrustes-rotatointi ja skaalaus ei osaa huomioida, sillä paikat $9-12$ sisältävä ryhmä sekä $22-28$ ovat ordinaatikuvissa sijainneiltaan vaihtaneet keskenään paikkaa suhteessa muihin ryhmiin.




Kuva 13: Latentin muuttujamallin Poissonjakaumaoletuksella hämähäkkiaineistolle tuottamat kaksi erilaista ordinaatiokuvaa `procrustes`-rotatoituna normaalijakaumasta generoiduilla alkuarvoilla.

Simuloidaan lopuksi vaste Poisson-jakaumasta ja negatiivisesta binomijakaumasta, ja verrataan molemmissa LVM-P- sekä LVM-NB-menetelmiä procrustes-virheiden avulla. Kuvasta 14 nähdään, että negatiivibinomijakaumaoletus on lähes yhtä hyvä kuin Poisson-jakaumaoletus, kun simuloidaan Poisson-jakaumasta. Tämä johtuu siitä, että negatiivisen binomijakauman skaalaparametrin $\phi$ lähestyessä nollaa koko jakauma lähestyy Poisson-jakaumaa. Sen sijaan, kun simuloidaan negatiivisesta binomijakaumasta, Poisson-jakaumaoletus vasteille toimii selvästi huonommin kuin negatiivibinomijakaumaoletus.




Kuva 14: Vasemmalla Poisson-jakautuneille vasteille on vertailtu LVM-P- ja LVM-NB-menetelmiä simuloimalla. Oikealla vastaavasti negatiivibinomijakautuneille vasteille.

# 7 Pohdintaa

Edellisissä luvuissa tarkastelimme latenttia muuttujamallia ordinaatiomenetelmänä. Menetelmää sovellettiin lukumääräaineistoille eri jakaumaoletuksin. Lisäksi tarkasteltiin menetelmän tarjoamia etuja ja sen toimivuutta yleisimpiin klassisiin menetelmiin verrattuna.

Simulointikokeiden perusteella havaittiin latentin muuttujamallin toimivan ordinaatiomenetelmänä hyvin. Klassisiin menetelmiin NMDS, PCoA ja DCA verrattuna se ei ole ehdoton ykkönen, mutta aina kuitenkin parhaiden joukossa. Selkeänä etuna klassisiin menetelmiin verrattuna on kuitenkin mahdollisuus tarkistaa mallin sopivuutta ja oletusten paikkansapitävyyttä. Klassisten menetelmien taas havaittiin toimivan vaihtelevalla menestyksellä aineistosta riippuen, eikä mitenkään voida tarkastella ja testata niiden hyvyyttä.

Latentti muuttujamalli vaatii kuitenkin melko hyvät alkuarvot latenteille muuttujille. Luvussa 6 tehtyjen simulointikokeiden tuloksissa huomattiin normaalijakaumasta generoitujen alkuarvojen tuottavan vaihtelevia tuloksia, mikä osaltaan johtunee siitä, että huonommilla alkuarvoilla uskottavuus konvergoi alkuarvoja lähellä olevaan lokaaliin maksimiin. Yksi tapa löytää riittävän hyvät alkuarvot on käyttää jonkin klassisen menetelmän antamia koordinaatteja alkuarvoina latenteille muuttujille. Koska klassisten menetelmien toteutus on nopeaa, ainakin R-ohjelmiston valmiilla funktioilla, tämä ei kasvata kokonaislaskenta-aikaa. Malliparametrien alkuarvoilla ei ollut yhtä suurta vaikutusta lopputuloksiin. Parametrien estimoiminen EM-algoritmin, (Hui et al., 2014), sijaan Laplacen approksimaatiota apuna käyttäen toimii hyvin, ja laskenta-aika on selvästi lyhyempi.

# Viitteet


C. G. Broyden. Quasi-newton methods and their application to function minimisation. *Mathematics of Computation*, 21:368–381, 1967.

A. C. Davison. *Statistical Models*. Cambridge University Press, New York, 2003.

A. J. Dobson. *An Introduction to Generalized Linear Models*. Chapman & Hall/CRC, 2002.

P. K. Dunn ja G. K. Smyth. Randomized quantile residuals. *Journal of Computational and Graphical Statistics*, 5:236–244, 1996.

M. O. Hill ja H. G. Gauch Jr. Detrended correspondence analysis: an improved ordination technique. *Vegetatio*, 42:47–58, 1980.

P. Huber, E. Ronchetti, ja M. P. Victoria-Feser. Estimation of generalized linear latent variable models. *Journal of the Royal Statistical Society*, 66:893–908, 2004.

F. K. C. Hui, S. Taskinen, S. Pledger, S. D. Foster, ja D. I. Warton. Model-based approaches to unconstrained ordination. *Methods in Ecology and Evolution*, 2014. doi: 10.1111/2041-210X.12236.

J. B. Kruskal. Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, 29:115–129, 1964.

G. P. Quinn ja M. J. Keough. *Experimental Design and Data Analysis for Biologists*. Cambridge University Press, Cambridge, 2002.

M. D. Sammel, L. M. Ryan, ja J. M. Legler. Latent variable models for mixed discrete and continuous outcomes. *Royal Statistical Society: Series B*, 59:667–678, 1997.

A. Skrondal ja S. Rabe-Hesketh. *Generalized Latent Variable Modeling: Multilevel, Longitudinal and Structural Equation Models*. Chapman & Hall, 2004.

J. Stoklosa, H. Gibb, ja D. I. Warton. Fast forward selection for generalized estimating equations with a large number of predictor variables. *Biometrics*, 70:110–120, 2014.

P. J. M. van der Aart ja N. Smeenk-Enserink. Correlations between distributions of hunting spiders (lycosidae, ctenidae) and environmental characteristics in a dune area. *Netherlands Journal of Zoology*, 25:1–45, 1974.

# Liite A Derivaatat

Logaritmisen uskottavuusfunktion Laplacen approksimaation maksimoimiseksi, lasketaan derivaatat parametrien suhteen.

## A.1 Bernoulli-jakautuneille vastemuuttujille

Derivoidaan logaritmista uskottavuusfunktiota

$$\tilde{l}(\boldsymbol{\theta};\boldsymbol{y}) = \sum_{i=1}^{n}\Bigg( -\frac{1}{2}\log\det\left(\sum_{j=1}^{p}\frac{e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j}}{(1+e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j})^2}\boldsymbol{\gamma}_j\boldsymbol{\gamma}_j' + \boldsymbol{I}_q\right) + \sum_{j=1}^{p}\left[y_{ij}(\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j) - \log\left(1+e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j}\right)\right] - \frac{\boldsymbol{z}_i'\boldsymbol{z}_i}{2}\Bigg),$$

$$\frac{\partial\tilde{l}(\boldsymbol{\theta};\boldsymbol{y})}{\partial\alpha_k} = -0.5\cdot\operatorname{tr}\left(\boldsymbol{\Gamma}(\boldsymbol{\theta},\boldsymbol{z}_k)^{-1}\sum_{j=1}^{p}\frac{e^{\alpha_k+\beta_j+\boldsymbol{z}_k'\boldsymbol{\gamma}_j}(1-e^{\alpha_k+\beta_j+\boldsymbol{z}_k'\boldsymbol{\gamma}_j})}{(1+e^{\alpha_k+\beta_j+\boldsymbol{z}_k'\boldsymbol{\gamma}_j})^3}\boldsymbol{\gamma}_j\boldsymbol{\gamma}_j'\right) + \sum_{j=1}^{p}\left(y_{kj} - \frac{e^{\alpha_k+\beta_j+\boldsymbol{z}_k'\boldsymbol{\gamma}_j}}{(1+e^{\alpha_k+\beta_j+\boldsymbol{z}_k'\boldsymbol{\gamma}_j})}\right),$$

$k = 1,\ldots,n.$

$$\frac{\partial\tilde{l}(\boldsymbol{\theta};\boldsymbol{y})}{\partial\beta_r} = \sum_{i=1}^{n}\Bigg[ -0.5\cdot\operatorname{tr}\left(\boldsymbol{\Gamma}(\boldsymbol{\theta},\boldsymbol{z}_i)^{-1}\frac{e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r}(1-e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r})}{(1+e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r})^3}\boldsymbol{\gamma}_r\boldsymbol{\gamma}_r'\right) + y_{ir} - \frac{e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r}}{1+e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r}}\Bigg],$$

$r = 1,\ldots,p.$

$$\frac{\partial\tilde{l}(\boldsymbol{\theta};\boldsymbol{y})}{\partial\gamma_{rs}} = \sum_{i=1}^{n}\Bigg[ -0.5\cdot\operatorname{tr}\Big(\boldsymbol{\Gamma}(\boldsymbol{\theta},\boldsymbol{z}_i)^{-1}\Big\{\frac{e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r}(1-e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r})}{(1+e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r})^3}z_{is}\boldsymbol{\gamma}_r\boldsymbol{\gamma}_r' + \frac{e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r}}{(1+e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r})^2}(e_s\otimes\boldsymbol{\gamma}_r' + e_s'\otimes\boldsymbol{\gamma}_r)\Big\}\Big) + \left(y_{ir} - \frac{e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r}}{1+e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r}}\right)z_{is}\Bigg],$$

$r = 1,\ldots,p,\ \ s = 1,\ldots,q,$ missä $\otimes$-merkki tarkoittaa Kroneckerin tuloa ja $e_s$ on $q$-vektori, jonka $s$. alkio on 1 ja muut ovat nollia.

## A.2 Poisson-jakautuneille vastemuuttujille

Derivoidaan logaritmista uskottavuusfunktiota

$$\tilde{l}(\boldsymbol{\theta};\boldsymbol{y}) = \sum_{i=1}^{n}\Bigg( -\frac{1}{2}\log\det\left(\sum_{j=1}^{p} e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j}\boldsymbol{\gamma}_j\boldsymbol{\gamma}_j' + \boldsymbol{I}_q\right) + \sum_{j=1}^{p}\left[y_{ij}(\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j) - e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j} - \log(y_{ij}!)\right] - \frac{\boldsymbol{z}_i'\boldsymbol{z}_i}{2}\Bigg).$$

$$\frac{\partial\tilde{l}(\boldsymbol{\theta};\boldsymbol{y})}{\partial\alpha_k} = -0.5\cdot\operatorname{tr}\left(\boldsymbol{\Gamma}(\boldsymbol{\theta},\boldsymbol{z}_k)^{-1}\sum_{j=1}^{p} e^{\alpha_k+\beta_j+\boldsymbol{z}_k'\boldsymbol{\gamma}_j}\boldsymbol{\gamma}_j\boldsymbol{\gamma}_j'\right) + \sum_{j=1}^{p}\left(y_{kj} - e^{\alpha_k+\beta_j+\boldsymbol{z}_k'\boldsymbol{\gamma}_j}\right),$$

$k = 1,\ldots,n.$

$$\frac{\partial\tilde{l}(\boldsymbol{\theta};\boldsymbol{y})}{\partial\beta_r} = \sum_{i=1}^{n}\left[-0.5\cdot\operatorname{tr}\left(\boldsymbol{\Gamma}(\boldsymbol{\theta},\boldsymbol{z}_i)^{-1}e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r}\boldsymbol{\gamma}_r\boldsymbol{\gamma}_r'\right) + y_{ir} - e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r}\right],$$

$r = 1,\ldots,p.$

$$\frac{\partial\tilde{l}(\boldsymbol{\theta};\boldsymbol{y})}{\partial\gamma_{rs}} = \sum_{i=1}^{n}\Bigg[ -0.5\cdot\operatorname{tr}\Big(\boldsymbol{\Gamma}(\boldsymbol{\theta},\boldsymbol{z}_i)^{-1}e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r}\Big\{z_{is}\boldsymbol{\gamma}_r\boldsymbol{\gamma}_r' + (e_s\otimes\boldsymbol{\gamma}_r' + e_s'\otimes\boldsymbol{\gamma}_r)\Big\}\Big) + \left(y_{ir} - e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r}\right)z_{is}\Bigg],$$

$r = 1,\ldots,p,\ s = 1,\ldots,q.$

## A.3 Negatiivibinomijakautuneille vastemuuttujille

Derivoidaan logaritmista uskottavuusfunktiota

$$\tilde{l}(\boldsymbol{\theta};\boldsymbol{y}) = \sum_{i=1}^{n}\Bigg( -\frac{1}{2}\log\det\left(\sum_{j=1}^{p}\left(y_{ij}+\frac{1}{\phi_j}\right)\frac{\phi_j e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j}}{(1+\phi_j e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j})^2}\boldsymbol{\gamma}_j\boldsymbol{\gamma}_j' + \boldsymbol{I}_q\right) + \sum_{j=1}^{p}\Bigg[y_{ij}(\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j) - \left(y_{ij}+\frac{1}{\phi_j}\right)\log\left(1+\phi_j e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j}\right) + y_{ij}\log\phi_j + \log\Gamma\left(y_{ij}+\frac{1}{\phi_j}\right) - \log y_{ij}! - \log\Gamma\left(\frac{1}{\phi_j}\right)\Bigg] - \frac{\boldsymbol{z}_i'\boldsymbol{z}_i}{2}\Bigg).$$

$$\frac{\partial\tilde{l}(\boldsymbol{\theta};\boldsymbol{y})}{\partial\alpha_k} = -0.5\cdot\operatorname{tr}\left(\boldsymbol{\Gamma}(\boldsymbol{\theta},\boldsymbol{z}_k)^{-1}\sum_{j=1}^{p}\left(y_{kj}+\frac{1}{\phi_j}\right)\frac{\phi_j e^{\alpha_k+\beta_j+\boldsymbol{z}_k'\boldsymbol{\gamma}_j}(1-\phi_j e^{\alpha_k+\beta_j+\boldsymbol{z}_k'\boldsymbol{\gamma}_j})}{(1+\phi_j e^{\alpha_k+\beta_j+\boldsymbol{z}_k'\boldsymbol{\gamma}_j})^3}\boldsymbol{\gamma}_j\boldsymbol{\gamma}_j'\right) + \sum_{j=1}^{p}\left(y_{kj} - \left(y_{kj}+\frac{1}{\phi_j}\right)\frac{e^{\alpha_k+\beta_j+\boldsymbol{z}_k'\boldsymbol{\gamma}_j}}{\frac{1}{\phi_j}+e^{\alpha_k+\beta_j+\boldsymbol{z}_k'\boldsymbol{\gamma}_j}}\right),\quad k = 1,\ldots,n.$$

$$\frac{\partial \tilde{l}(\boldsymbol{\theta};\boldsymbol{y})}{\partial \beta_r} = \sum_{i=1}^{n}\Bigg[ -0.5\cdot \operatorname{tr}\left(\boldsymbol{\Gamma}(\boldsymbol{\theta},\boldsymbol{z}_i)^{-1}\Big(y_{ir}+\frac{1}{\phi_r}\Big)\frac{\phi_r e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r}(1-\phi_r e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r})}{(1+\phi_r e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r})^3}\boldsymbol{\gamma}_r\boldsymbol{\gamma}_r'\right)$$
$$+ y_{ir} - \Big(y_{ir}+\frac{1}{\phi_r}\Big)\frac{e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r}}{\frac{1}{\phi_r}+e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r}}\Bigg], \quad r=1,\ldots,p.$$

$$\frac{\partial \tilde{l}(\boldsymbol{\theta};\boldsymbol{y})}{\partial \gamma_{rs}} = \sum_{i=1}^{n}\Bigg[ -0.5\cdot \operatorname{tr}\left(\boldsymbol{\Gamma}(\boldsymbol{\theta},\boldsymbol{z}_i)^{-1}\Big(y_{ir}+\frac{1}{\phi_r}\Big)\frac{\phi_r e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r}}{(1+\phi_r e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r})^2}\right.$$
$$\left.\left\{\frac{1-\phi_r e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r}}{1+\phi_r e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r}} z_{is}\boldsymbol{\gamma}_r\boldsymbol{\gamma}_r' + (e_s\otimes\boldsymbol{\gamma}_r' + e_s'\otimes\boldsymbol{\gamma}_r)\right\}\right)$$
$$+\left(y_{ir}-\Big(y_{ir}+\frac{1}{\phi_r}\Big)\frac{e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r}}{\frac{1}{\phi_r}+e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r}}\right) z_{is}\Bigg], \quad r=1,\ldots,p,\ s=1,\ldots,q.$$

$$\frac{\partial \tilde{l}(\boldsymbol{\theta};\boldsymbol{y})}{\partial \phi_r} = \sum_{i=1}^{n}\Bigg[ -0.5\cdot \operatorname{tr}\left(\boldsymbol{\Gamma}(\boldsymbol{\theta},\boldsymbol{z}_i)^{-1}\frac{e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r}\left(y_{ir}-y_{ir}\phi_r e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r}-2e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r}\right)}{(1+\phi_r e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r})^3}\boldsymbol{\gamma}_r\boldsymbol{\gamma}_r'\right)$$
$$+\frac{1}{\phi_r^2}\left(\log\left(1+\phi_r e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r}\right)-\psi_0\Big(y_{ir}+\frac{1}{\phi_r}\Big)+\psi_0\Big(\frac{1}{\phi_r}\Big)\right)$$
$$-\Big(y_{ir}+\frac{1}{\phi_r}\Big)\frac{e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r}}{1+\phi_r e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r}}+\frac{y_{ir}}{\phi_r}\Bigg], \quad r=1,\ldots,p,$$

missä $\psi_0$ on digammafunktio.

## A.4 Derivaatat selittävät muuttujat sisältävälle mallille

Kun latenttiin muuttujamalliin lisätään selittävät muuttujat, Laplacen approksimaatiot logaritmiselle uskottavuusfunktiolle, kun vastemuuttujat ovat Poisson-jakautuneet on

$$\tilde{l}_P(\boldsymbol{\theta};\boldsymbol{y}) = \sum_{i=1}^{n}\Bigg( -\frac{1}{2}\log\det\left(\sum_{j=1}^{p} e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j+\boldsymbol{x}_i'\boldsymbol{\delta}_j}\boldsymbol{\gamma}_j\boldsymbol{\gamma}_j' + \boldsymbol{I}_q\right)$$
$$+\sum_{j=1}^{p}\Big[y_{ij}(\alpha_i+\beta_j+\hat{\boldsymbol{z}}_i'\boldsymbol{\gamma}_j+\boldsymbol{x}_i'\boldsymbol{\delta}_j) - e^{\alpha_i+\beta_j+\hat{\boldsymbol{z}}_i'\boldsymbol{\gamma}_j+\boldsymbol{x}_i'\boldsymbol{\delta}_j} - \log(y_{ij}!)\Big] - \frac{\hat{\boldsymbol{z}}_i'\hat{\boldsymbol{z}}_i}{2}\Bigg),$$

ja kun vastemuuttujat ovat negatiivibinomijakautuneet

$$\begin{aligned}\tilde{l}_{NB}(\boldsymbol{\theta};\boldsymbol{y}) = \sum_{i=1}^{n}\Bigg( &-\frac{1}{2}\log\det\left(\sum_{j=1}^{p}\left(y_{ij}+\frac{1}{\phi_j}\right)\frac{\phi_j e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j+\boldsymbol{x}_i'\boldsymbol{\delta}_j}}{(1+\phi_j e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j+\boldsymbol{x}_i'\boldsymbol{\delta}_j})^2}\boldsymbol{\gamma}_j\boldsymbol{\gamma}_j' + \boldsymbol{I}_q\right)\\ &+\sum_{j=1}^{p}\Bigg[y_{ij}(\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j+\boldsymbol{x}_i'\boldsymbol{\delta}_j) - \left(y_{ij}+\frac{1}{\phi_j}\right)\log\left(1+\phi_j e^{\alpha_i+\beta_j+\boldsymbol{z}_i'\boldsymbol{\gamma}_j+\boldsymbol{x}_i'\boldsymbol{\delta}_j}\right)\\ &+ y_{ij}\log\phi_j + \log\Gamma\left(y_{ij}+\frac{1}{\phi_j}\right) - \log y_{ij}! - \log\Gamma\left(\frac{1}{\phi_j}\right)\Bigg] - \frac{\hat{\boldsymbol{z}}_i'\hat{\boldsymbol{z}}_i}{2}\Bigg),\end{aligned}$$

missä $\boldsymbol{\theta} = \{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\phi}, \boldsymbol{\delta}\}$. Kun näitä derivoidaan muuttujien $\alpha_k$, $\beta_r$, $\gamma_{rs}$ ja negatiivisen binomijakauman tapauksessa vielä muuttujan $\phi_r$ suhteen, derivaatat ovat kuten kappaleissa A.1 ja A.2, kun lisätään vain termi $\boldsymbol{x}_i'\boldsymbol{\delta}$ malliin. Lasketaan vielä derivaatat muuttujien $\delta_h$, $h = 1, \ldots, m$ suhteen, Poisson-jakaumalle:

$$\begin{aligned}\frac{\partial\tilde{l}_P(\boldsymbol{\theta};\boldsymbol{y})}{\partial\delta_{hr}} = \sum_{i=1}^{n}\Bigg[ &-0.5\cdot\operatorname{tr}\left(\boldsymbol{\Gamma}(\boldsymbol{\theta},\boldsymbol{z}_i)^{-1}e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r+\boldsymbol{x}_i'\boldsymbol{\delta}_r}\boldsymbol{\gamma}_r\boldsymbol{\gamma}_r' x_{ih}\right)\\ &+\left(y_{ir} - e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r+\boldsymbol{x}_i'\boldsymbol{\delta}_r}\right)x_{ih}\Bigg],\end{aligned}$$

negatiiviselle binomijakaumalle:

$$\begin{aligned}\frac{\partial\tilde{l}_{NB}(\boldsymbol{\theta};\boldsymbol{y})}{\partial\delta_{hr}} = \sum_{i=1}^{n}\Bigg[ &-0.5\cdot\operatorname{tr}\Bigg(\boldsymbol{\Gamma}(\boldsymbol{\theta},\boldsymbol{z}_i)^{-1}\\ &\left(y_{ir}+\frac{1}{\phi_r}\right)\frac{\phi_r e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r+\boldsymbol{x}_i'\boldsymbol{\delta}_r}(1-\phi_r e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r+\boldsymbol{x}_i'\boldsymbol{\delta}_r})}{(1+\phi_r e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r+\boldsymbol{x}_i'\boldsymbol{\delta}_r})^3}\boldsymbol{\gamma}_r\boldsymbol{\gamma}_r' x_{ih}\Bigg)\\ &+\left(y_{ir} - \left(y_{ir}+\frac{1}{\phi_r}\right)\frac{e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r+\boldsymbol{x}_i'\boldsymbol{\delta}_r}}{\frac{1}{\phi_r}+e^{\alpha_i+\beta_r+\boldsymbol{z}_i'\boldsymbol{\gamma}_r+\boldsymbol{x}_i'\boldsymbol{\delta}_r}}\right)x_{ih}\Bigg], \quad r = 1,\ldots,p.\end{aligned}$$

# Liite B Etäisyysmittoja

Olkoon paikalta $h$ mitattujen muuttujien arvot $\boldsymbol{y}_h = (y_{h1}, \ldots, y_{hp})$. Seuraavassa taulukossa on esitetty paikkojen $h$ ja $i$ välisiä etäisyyksiä.

| Etäisyys | Määritelmä | |
|---|---|---|
| Bray-Curtis | $\frac{\sum_{k=1}^{p} \lvert y_{hk} - y_{ik} \rvert}{\sum_{k=1}^{p} (y_{hk} + y_{ik})}$ | |
| Canberra | $\frac{1}{p-u} \sum_{k=1}^{p} \frac{\lvert y_{hk} - y_{ik} \rvert}{(y_{hk} + y_{ik})}$ | $u$ on muuttujien lukumäärä, jotka saavat arvon nolla paikoilla $h$ ja $i$ |
| Euclidean | $\sqrt{\sum_{j=1}^{p} (y_{hj} - y_{ij})^2}$ | |
| Chi-square | $d_{hi} = \sqrt{\sum_{j=1}^{p} \frac{(y_{hj}/r_h - y_{ij}/r_i)^2}{c_j}}$ | $r_h$ on muuttujien summa paikalla $h$ ja $c_j$ on j. muuttujan summa kaikilta paikoilta |

Wisconsin-kaksoisstandardointi havainnolle $y_{kl}$ voidaan kirjoittaa:

$$\frac{\frac{y_{kl}}{\max\{y_{1l}, \ldots, y_{nl}\}}}{\sum_{j=1}^{p} \frac{y_{kj}}{\max\{y_{1j}, \ldots, y_{nj}\}}}, \tag{23}$$

eli ensin jaetaan havainnot kunkin sarakkeen maksimilla, ja sitten syntyneen matriisin alkiot jaetaan vielä kunkin rivin summalla.

# Liite C R-koodi

```
# R-koodi
# Jenni Niku
# 10.12.2014

library(psych)
library(mvabund)
library(mvtnorm)
library(vegan)
library(numDeriv)


# LVM -mallin sovitus Laplacen approksimaation avulla Poisson-jakaumalle
LAMLE<-function(data,num.lv,max.iter=50,z0="s",site=TRUE,method="MAP"){
# z0={"s","mds","dca","pcoa"}
# method={"MAP","Qmax"}
n<-dim(data)[1]
p<-dim(data)[2]
q<-num.lv<-2
ptm <- proc.time()

# Initial values for model parameters and latent variables
para<-start.values(data,att.dist="p")
bs<-para$params[,1]
gs<-c(para$params[,2],para$params[,3])
params<-c(bs,gs)
if(site){
aas<-rep(0,n)
params<-c(aas,params)
st=0
} else {st=n}
if(z0=="mds"){
bray <- vegdist(data)
bray <- metaMDS(bray,k=2)
Z=t(bray$points)
} else{ if(z0=="dca"){
CA.res<-decorana(data)
Z=t(CA.res$rproj[,1:2])
} else{ if(z0=="pcoa"){
PCoA.res<-capscale(data~1,distance="bray")
Z=t(scores(PCoA.res,display="sites"))
}else{Z<-t(para$index)} } }

## log- likelihood
lLpois<-function(param,Z,data){
# dat is nxp data matrix
# ai=alfa_i
# b=t([beta_1,...,beta_p])
# G is qxp matrix, (q=2)
# z is qxp matrix, where z[,i]=t([z_i1,...,z_iq])
# n is number of sites
# p is number of species
# q is number of latent variables
if(site){
a=param[1:n]
} else{a=rep(0,n);}
b=param[(n+1-st):(n+p-st)]
g.vec=param[(n+p+1-st):(n+p+p*q-1-st)]
g.vec<-c(g.vec[1:p],0,g.vec[(p+1):(2*p-1)])
G<-matrix(g.vec,nrow=q,byrow=TRUE)

zg=t(Z)%*%G
mu.apu<-t(t(zg)+b)+a
SGam=0
fsum=0
for(i in 1:n){
```

```
  SGam=SGam-0.5*log(det(Gamma(a[i],b,G,Z[,i])))
  fsum=fsum+sum(log(dpois( data[i,], lambda=exp(mu.apu[i,]) )+1e-323))
}
L=SGam+fsum-sum(diag(t(Z)%*%Z))/2
L
}

# gradients of parameters alfa, beta, gamma
param.gradient<-function(param,Z,data){
if(site){
a=param[1:n];
} else{a=rep(0,n);}
b=param[(n+1-st):(n+p-st)]
g.vec=param[(n+p+1-st):(n+p+p*q-1-st)]
g.vec<-c(g.vec[1:p],0,g.vec[(p+1):(2*p-1)])
G<-matrix(g.vec,nrow=q,byrow=TRUE)
zg=t(Z)%*%G
mu.apu<-t(t(zg)+b)+a

Gam=list()
grad.a=vector(length=n);
grad.b=vector(length=p);
grad.G=matrix(0,nrow=q,ncol=p)
for(i in 1:n){
  Ga=Gamma(a[i],b,G,Z[,i])
  grad.a[i]<- -0.5*tr(solve(Ga)%*%(Ga-diag(2))) + sum(data[i,]-exp(mu.apu[i,]))
  for(r in 1:p){
    grad.b[r]=grad.b[r] - 0.5*tr( solve(Ga)%*%(exp(mu.apu[i,r])*G[,r]%*%
        t(G[,r])) ) + data[i,r]-exp(mu.apu[i,r]);
    grad.G[1,r]=grad.G[1,r] - 0.5*tr( solve(Ga)%*%(exp(mu.apu[i,r])*Z[1,i]*
        G[,r]%*%t(G[,r]) + exp(mu.apu[i,r])*(kronecker(c(1,0),
        t(G[,r]))+kronecker(t(c(1,0)),G[,r])) )+
        (data[i,r]-exp(mu.apu[i,r]))*Z[1,i];
    grad.G[2,r]=grad.G[2,r] - 0.5*tr( solve(Ga)%*%(exp(mu.apu[i,r])*Z[2,i]*
        G[,r]%*%t(G[,r]) + exp(mu.apu[i,r])*(kronecker(c(0,1),
        t(G[,r])) + kronecker(t(c(0,1)),G[,r])) )+
        (data[i,r]-exp(mu.apu[i,r]))*Z[2,i];
  }
}
grads<-c(grad.b,grad.G[1,],grad.G[2,-1])
if(site){
grads<-c(grad.a,grads)
}
grads
}

# Q-function
Qfun<-function(params,Zi,datai,i){
if(site){
a=params[1:n]
} else{a=rep(0,n);}
b=params[(n+1-st):(n+p-st)]
g.vec=params[(n+p+1-st):(n+p+p*q-1-st)]
g.vec<-c(g.vec[1:p],0,g.vec[(p+1):(2*p-1)])
G<-matrix(g.vec,nrow=q,byrow=TRUE)
oa<-(a[i]+b+Zi%*%G)
i/p*( sum(log(dpois(datai,lambda=exp(oa)))) - (t(Zi)%*%Zi)[1,1]/2 - log(2*pi) )
}

# Gamma-matrix for Poisson distribution
Gamma<-function(ai,b,G,zi){
# ai=alfa_i
# b=t([beta_1,...,beta_p])
# G is qxp matrix, (q=2)
# zi= z_i=t([z_i1,...,z_iq])
q=2
p=length(b)
apu<-exp(t(zi)%*%G+b+ai)
```

```
s<-0
for(j in 1:p){
  s=s+ apu[1,j]*G[,j]%*%t(G[,j])
}
s+diag(q)
}

current.loglik <- -1e6;
iter <- 1;
err <- 10;
new.par=params

while((err > 1.0005 || err < 0.9995) && iter <= max.iter){
ptm <- proc.time()
# updating of \alpha, \beta and \gamma parameters
fitgr<-try(optim(new.par[-(n+2*p+1-st)],fn=lLpois,gr=param.gradient,method = "BFGS",
    control = list(trace = 0, fnscale = -1),Z=Z,data=data),silent=T)
#old.par=new.par
new.par<-c(fitgr$par[1:(n+2*p-st)],0,fitgr$par[(n+2*p-st+1):(n+p+2*p-st-1)])
print((proc.time()-ptm)[3])
print(fitgr$value)

new.loglik <- fitgr$value
err <- abs(new.loglik/current.loglik);
cat("New Loglik:", new.loglik, "Current Loglik:", current.loglik, "Ratio", err,"\n")
current.loglik <- new.loglik;

# set new parameters:
if(site){
a=new.par[1:n]
} else{a=rep(0,n)}
b=new.par[(n+1-st):(n+p-st)]
g.vec=new.par[(n+p+1-st):(n+p+p*q-st)]
G<-matrix(g.vec,nrow=q,byrow=TRUE)

## updating of Z
cat("Working out index for each spp\n")
#        ptm <- proc.time()
index <- matrix(rep(0,num.lv*n),n,num.lv);
if(method=="MAP"){
big.simlv <- rmvnorm(20000,rep(0,num.lv)); X.bigb <- cbind(rep(1,20000),rep(1,20000),
    big.simlv)
  for(i in 1:n) {
          spp.f <- matrix(NA, nrow = 20000, ncol = p)
          for(j in 1:p) {
                      beta <- c(a[i],b[j],G[,j]); eta <- X.bigb %*% beta;
              spp.att.mu <- exp(eta); spp.f[,j] <- dpois(data[i,j], spp.att.mu) }
  spp.posterior <- apply(spp.f, 1, prod) * dnorm(big.simlv[,1],0,1)
        if (num.lv==2) {  spp.posterior <- spp.posterior * dnorm(big.simlv[,2],0,1) }
  index[i,] <- big.simlv[which(spp.posterior == max(spp.posterior)),] }
} else{
# Maximize Q
for(i in 1:n){
fitz<-try(optim(Z[,i],fn=Qfun,method = "BFGS",control = list(trace = 0, fnscale = -1)
    ,params=new.par[-(n+2*p+1-st)],datai=data[i,],i=i),silent=T)
index[i,]<-fitz$par
}
}
        cat("Parameters calculated in time:\n")
        print((proc.time()-ptm)[3])

# lets plot new and old Z to the same picture
plot(index,type="n")
text(index,labels=seq(1,n))
points(t(Z),type="n")
text(t(Z),labels=seq(1,n),col=2)
Z=t(index)
cat("Number of iterations",iter,"\n")
```

```
iter = iter + 1
}
list(a=a,b=b,g=t(G),z=t(Z),ll=current.loglik,pars=new.par)
}

# LVM -mallin sovitus Laplacen approksimaation avulla Negatiiviselle binomijakaumalle
LVM_L_nbp<-function(data,num.lv,max.iter=50,z0="s",estimf=TRUE,site=TRUE,method="MAP"
    ){
n=dim(data)[1]
p=dim(data)[2]
q=num.lv
# Initial values for model parameters and latent variables
if(estimf){
para<-start.values(data,att.dist="nb")
f<-para$params[,4]
} else { para<-start.values(data,att.dist="p") }
bs<-para$params[,1]
gs<-c(para$params[,2],para$params[,3])
params<-c(bs,gs)
# Initials to \alpha
if(site){
aas<-rep(0,n) #rnorm(n)
params<-c(aas,params)
st=0
} else {st=n}

if(z0=="mds"){
bray <- metaMDS(data,distance="bray",k=2)
Z=t(bray$points)
} else{ if(z0=="dca"){
CA.res<-decorana(data)
Z=t(CA.res$rproj[,1:2])
} else{ if(z0=="pcoa"){
PCoA.res<-capscale(data~1,distance="bray")
Z=t(scores(PCoA.res,display="sites"))
} else{ Z<-t(para$index)} } }

# Initial values for scale parameters
if(!estimf){
f<-rep(0.001,p)
}

## log-likelihood
lLnb<-function(param,Z,data,f){
# dat is nxp data matrix
# ai=alfa_i
# b=t([beta_1,...,beta_p])
# G is qxp matrix, (q=2)
# z is qxp matrix, where z[,i]=t([z_i1,...,z_iq])
# f is p-vector
# n is number of sites
# p is number of species
# q is number of latent variables
if(site){
a=param[1:n]
} else{a=rep(0,n);}
b=param[(n+1-st):(n+p-st)]
g.vec=param[(n+p+1-st):(n+p+p*q-1-st)]
g.vec<-c(g.vec[1:p],0,g.vec[(p+1):(2*p-1)])
#f=param[(n+p+p*q+1):(n+p+p*q+p)]
G<-matrix(g.vec,nrow=q,byrow=TRUE)
#Z<-matrix(nrow=q,ncol=n)

zg=t(Z)%*%G
mu.apu<-apply(t(apply(zg,1,function(x,w) x+w,w=b)),2,function(x,w) x+w,w=a)
#yf<-t(t(data)+1/f)
SGam=0
```

```
fsum=0
for(i in 1:n){
  SGam=SGam-0.5*log(det(Gammanb(data[i,],a[i],b,f,G,Z[,i])))
  fsum=fsum+sum(log(dnbinom(data[i,],mu=exp(mu.apu[i,]),size=1/f)))
}
# log-likelihood
L= SGam + fsum - sum(diag(t(Z)%*%Z))/2
L
}

Qfun<-function(param,Zi,datai,f,i){
if(site){
a=param[1:n]
} else{a=rep(0,n);}
b=param[(n+1-st):(n+p-st)]
g.vec=param[(n+p+1-st):(n+p+p*q-1-st)]
g.vec<-c(g.vec[1:p],0,g.vec[(p+1):(2*p-1)])
G<-matrix(g.vec,nrow=q,byrow=TRUE)
oa<-a[i]+b+Zi%*%G
1/p*( sum(log(dnbinom(datai,mu=exp(oa),size=1/f)))-(t(Zi)%*%Zi)[1,1]/2 - log(2*pi))
}

# Gamma matrix
Gammanb<-function(yi,ai,b,f,G,zi){
# ai=alfa_i
# b=t([beta_1,...,beta_p])
# G is qxp matrix, (q=2)
# zi= z_i=t([z_i1,...,z_iq])
q=2
p=length(b)
my<-exp(t(zi)%*%G+b+ai)
s<-0
for(j in 1:p){
  s=s+ my[1,j]*f[j]*(yi[j]+1/f[j])/(1+f[j]*my[1,j])^2*(G[,j]%*%t(G[,j]))
}
s+diag(q)
}

# Gradients for \alpha, \beta and \gamma parameters
param.gradient<-function(param,Z,data,f){
if(site){
a=param[1:n];
} else{a=rep(0,n);}
b=param[(n+1-st):(n+p-st)]
g.vec=param[(n+p+1-st):(n+p+p*q-1-st)]
g.vec<-c(g.vec[1:p],0,g.vec[(p+1):(2*p-1)])
G<-matrix(g.vec,nrow=q,byrow=TRUE)
zg=t(Z)%*%G
mu.apu=apply(t(apply(zg,1,function(x,w) x+w,w=b)),2,function(x,w) x+w,w=a)
ex=exp(mu.apu)
yf=t(t(data)+1/f)
Gam=list();
grad.a=vector(length=n);
for(i in 1:n){
  Ga=Gammanb(data[i,],a[i],b,f,G,Z[,i])
  Gam[[length(Gam)+1]] <- Ga
sa=matrix(0,ncol=2,nrow=2)
for(j in 1:p){
sa=sa+f[j]*ex[i,j]*(data[i,j]+1/f[j])*(1-f[j]*ex[i,j])/(1+f[j]*ex[i,j])^3*G[,j]%*%t(G
    [,j])
}
grad.a[i]=-0.5*tr(solve(Gam[[i]])%*%sa)+sum( data[i,]-yf[i,]*ex[i,]/t(1/f+ex[i,]) )
}
grad.b=vector(length=p)
grad.G=matrix(0,nrow=q,ncol=p)
for(r in 1:p){
  for(i in 1:n){
  grad.b[r]= grad.b[r] -0.5*tr( solve(Gam[[i]])%*%(f[r]*ex[i,r]*(yf[i,r])*
```

```
      (1-f[r]*ex[i,r])/(1+f[r]*ex[i,r])^3*G[,r]%*%t(G[,r])) ) +
      data[i,r] - yf[i,r]*ex[i,r]/(1/f[r]+ex[i,r]);
#
  grad.G[1,r]= grad.G[1,r] - 0.5*tr( solve(Gam[[i]])%*%(f[r]*ex[i,r]*(yf[i,r])*
      (1-f[r]*ex[i,r])/(1+f[r]*ex[i,r])^3*Z[1,i]*G[,r]%*%t(G[,r]) + f[r]*ex[i,r]
      *(yf[i,r])/(1+f[r]*ex[i,r])^2*(kronecker(c(1,0),t(G[,r]))+
      kronecker(t(c(1,0)),G[,r]))) ) +
      (data[i,r] - yf[i,r]*ex[i,r]/(1/f[r]+ex[i,r]))*Z[1,i];
#
  grad.G[2,r]= grad.G[2,r] - 0.5*tr( solve(Gam[[i]])%*%(f[r]*ex[i,r]*(yf[i,r])*
      (1-f[r]*ex[i,r])/(1+f[r]*ex[i,r])^3*Z[2,i]*G[,r]%*%t(G[,r]) + f[r]*ex[i,r]
      *(yf[i,r])/(1+f[r]*ex[i,r])^2*(kronecker(c(0,1),t(G[,r]))+
      kronecker(t(c(0,1)),G[,r]))) ) +
      (data[i,r] - yf[i,r]*ex[i,r]/(1/f[r]+ex[i,r]))*Z[2,i];
  }
}
grads<-c(grad.b,grad.G[1,],grad.G[2,-1])
if(site){
grads<-c(grad.a,grads)
}
grads
}

# Gradients for scale parameters \phi
f.gradient<-function(param,Z,data,f){
if(site){
a=param[1:n];
} else{a=rep(0,n);}
b=param[(n+1-st):(n+p-st)]
g.vec=param[(n+p+1-st):(n+p+p*q-1-st)]
g.vec<-c(g.vec[1:p],0,g.vec[(p+1):(2*p-1)])
G<-matrix(g.vec,nrow=q,byrow=TRUE)
zg=t(Z)%*%G
mu.apu=apply(t(apply(zg,1,function(x,w) x+w,w=b)),2,function(x,w) x+w,w=a)
ex=exp(mu.apu)
yf=t(t(data)+1/f)
grad.f=vector(length=p)
for(r in 1:p){
  for(i in 1:n){
  grad.f[r]= grad.f[r] - 0.5*tr( solve(Gammanb(data[i,],a[i],b,f,G,Z[,i]))%*%
      (ex[i,r]*(data[i,r]-data[i,r]*ex[i,r]*f[r]-2*
      ex[i,r])/(1+f[r]*ex[i,r])^3*G[,r]%*%t(G[,r])) ) +
      1/f[r]^2*( log(1+f[r]*ex[i,r])-digamma(yf[i,r])+
      digamma(1/f[r]) ) - ex[i,r]*yf[i,r]/(1+f[r]*ex[i,r]) + data[i,r]/f[r];
}}
grad.f
}

current.loglik <- -1e6;
iter <- 1;
err <- 10;
new.par=params

while((err > 1.0005 || err < 0.9995) && iter <= max.iter){
ptm <- proc.time()
# updating of \alpha, \beta and \gamma parameters
fitgr<-try(optim(new.par[-(n+2*p+1-st)],fn=lLnb,gr=param.gradient,method = "BFGS",
    control = list(trace = 0, fnscale = -1),Z=Z,data=data,f=f),silent=T)
#
new.par<-c(fitgr$par[1:(n+2*p-st)],0,fitgr$par[(n+2*p+1-st):(n+p+p*q-1-st)])
# updating of \phi -parameters
if(estimf){
fitgr<-try(optim(f,fn=lLnb,gr=f.gradient,method = "BFGS",control = list(trace = 0,
    fnscale = -1),Z=Z,data=data,param=new.par[-(n+2*p+1-st)]),silent=T)
f<-fitgr$par
}

print((proc.time()-ptm)[3])
```

```
print(fitgr$value)
new.loglik <- fitgr$value
err <- abs(new.loglik/current.loglik);
cat("New Loglik:", new.loglik, "Current Loglik:", current.loglik, "Ratio", err, "\n")
current.loglik <- new.loglik;

# set new parameters:
if(site){
a=new.par[1:n]
} else{a=rep(0,n)}
b=new.par[(n+1-st):(n+p-st)]
g.vec=new.par[(n+p+1-st):(n+p+p*q-st)]
G<-matrix(g.vec,nrow=q,byrow=TRUE)

## updating of Z
  cat("Working out index for each spp\n")
  ptm <- proc.time()
index <- matrix(rep(0,num.lv*n),n,num.lv);
if(method=="MAP"){
big.simlv <- rmvnorm(20000,rep(0,num.lv)); X.bigb <- cbind(rep(1,20000),rep(1,20000),
    big.simlv)
  for(i in 1:n) {
          spp.f <- matrix(NA, nrow = 20000, ncol = p)
          for(j in 1:p) {
                      beta <- c(a[i],b[j],G[,j]); eta <- X.bigb %*% beta;
              spp.att.mu <- exp(eta); spp.f[,j] <- dnbinom(data[i,j], mu=spp.att.mu,
                  size=1/f[j]) }
  spp.posterior <- apply(spp.f, 1, prod) * dnorm(big.simlv[,1],0,1)
        if (num.lv==2) {  spp.posterior <- spp.posterior * dnorm(big.simlv[,2],0,1) }
  index[i,] <- big.simlv[which(spp.posterior == max(spp.posterior)),] }
} else{
# Maximize Q
for(i in 1:n){
fitz<-try(optim(Z[,i],fn=Qfun,method = "BFGS",control = list(trace = 0, fnscale = -1)
    ,param=new.par[-(n+2*p+1-st)],datai=data[i,],f=f,i=i),silent=T)
index[i,]<-fitz$par
}
}
        cat("Parameters calculated in time:\n")
        print((proc.time()-ptm)[3])
# lets plot new and old Z to the same picture
plot(index,type="n")
text(index,labels=seq(1,n))
points(t(Z),type="n")
text(t(Z),labels=seq(1,n),col=2)
Z=t(index)
cat("Number of iterations",iter,"\n")
iter = iter + 1
}
list(a=a,b=b,g=t(G),f=f,z=t(Z),ll=current.loglik,pars=new.par)
}

# LVM -mallin sovitus Laplacen approksimaation avulla Bernoulli-jakaumalle
LAMLEb<-function(dat,num.lv,max.iter=50,z0="s",site=TRUE){
n<-dim(dat)[1]
p<-dim(dat)[2]
q<-num.lv<-2
ptm <- proc.time()

# Initial values for model parameters and latent variables
para<-start.values(dat,att.dist=rep("b",12))
bs<-para$params[,1]
gs<-c(para$params[,2],para$params[,3])
params<-c(bs,gs)
if(site){
aas<-rep(0,n)
params<-c(aas,params)
st=0
```

```
} else {st=n}
if(z0=="mds"){
bray <- vegdist(dat)
bray <- metaMDS(bray,k=2)
Z=t(bray$points)
}else{Z<-t(para$index)}

##    log-likelihood
lLbern2<-function(param,Z,dat){
# dat is nxp data matrix
# a=[alfa_1,...,alfa_n)]'
# b=[beta_1,...,beta_p]'
# G is qxp matrix, (q=2)
# z is qxp matrix, where z[,i]=t([z_i1,...,z_iq])
# n is number of sites
# p is number of species
# q is number of latent variables
if(site){
a=param[1:n]
} else{a=rep(0,n);}
b=param[(n+1-st):(n+p-st)]
g.vec=param[(n+p+1-st):(n+p+p*q-1-st)]
g.vec<-c(g.vec[1:p],0,g.vec[(p+1):(2*p-1)])
G<-matrix(g.vec,nrow=q,byrow=TRUE)
zg=t(Z)%*%G
mu.apu<-apply(t(apply(zg,1,function(x,w) x+w,w=b)),2,function(x,w) x+w,w=a)

SGam=0
for(i in 1:n){
  SGam=SGam-0.5*log(det(Gammab(a[i],b,G,Z[,i])))
}
L=SGam+sum(dat*mu.apu)-sum(log(1+exp(mu.apu)))-sum(diag(t(Z)%*%Z))
L
}

# Gamma matrix
Gammab<-function(ai,b,G,zi){
# ai=alfa_i
# b=t([beta_1,...,beta_p])
# G is qxp matrix, (q=2)
# zi= z_i=t([z_i1,...,z_iq])
q=2
p=length(b)
apu<-exp(t(zi)%*%G+b+ai)/(1+exp(t(zi)%*%G+b+ai))^2
s<-0
for(j in 1:p){
  s=s+ apu[1,j]*G[,j]%*%t(G[,j])
}
s+diag(q)
}

# gradients of parameters alfa, beta, gamma
param.gradient<-function(param,Z,dat){
if(site){
a=param[1:n];
} else{a=rep(0,n);}
b=param[(n+1-st):(n+p-st)]
g.vec=param[(n+p+1-st):(n+p+p*q-1-st)]
g.vec<-c(g.vec[1:p],0,g.vec[(p+1):(2*p-1)])
G<-matrix(g.vec,nrow=q,byrow=TRUE)
#Z<-matrix(z.vec,nrow=q)
zg=t(Z)%*%G
mu.apu<-apply(t(apply(zg,1,function(x,w) x+w,w=b)),2,function(x,w) x+w,w=a)
ex<-exp(mu.apu)

Gam=list(Gammab(a[1],b,G,Z[,1]))
for(i in 2:n){
  Ga=Gammab(a[i],b,G,Z[,i])
```

```
  Gam[[length(Gam)+1]] <- Ga
}
grad.a=vector(length=n);
for(i in 1:n){
sa=matrix(0,ncol=2,nrow=2)
for(j in 1:p){
sa=sa+ex[i,j]*(1-ex[i,j])/(1+ex[i,j])^3*(G[,j]%*%t(G[,j]))
}
grad.a[i]<- -0.5*tr(solve(Gam[[i]])%*%sa) + sum(dat[i,]-ex[i,]/(1+ex[i,]))
}

grad.b=vector(length=p);
grad.G=matrix(0,nrow=q,ncol=p)
for(r in 1:p){
  for(i in 1:n){
    grad.b[r]=grad.b[r] - 0.5*tr( solve(Gam[[i]])%*%(ex[i,r]*(1-ex[i,r])/(1+ex[i,r])
        ^3*(G[,r]%*%t(G[,r]))) )
        + dat[i,r]-ex[i,r]/(1+ex[i,r])

    grad.G[1,r]=grad.G[1,r] - 0.5*tr( solve(Gam[[i]])%*%
        ( ex[i,r]*(1-ex[i,r])/(1+ex[i,r])^3*G[,r]%*%t(G[,r])*Z[1,i])+
        ex[i,r]/(1+ex[i,r])^2*(kronecker(c(1,0),t(G[,r]))+kronecker(t(c(1,0)),G[,r]))
             )+
        (dat[i,r]-ex[i,r]/(1+ex[i,r]))*Z[1,i]

    grad.G[2,r]=grad.G[2,r] - 0.5*tr( solve(Gam[[i]])%*%(ex[i,r]*(1-ex[i,r])/(1+ex[i,
        r])^3*G[,r]%*%t(G[,r])*Z[2,i])+ex[i,r]/(1+ex[i,r])^2*(kronecker(c(0,1),t(G[,r
        ]))+kronecker(t(c(0,1)),G[,r])) )+
        (dat[i,r]-ex[i,r]/(1+ex[i,r]))*Z[2,i]
  }
}
grads<-c(grad.b,grad.G[1,],grad.G[2,-1])
if(site){
grads<-c(grad.a,grads)
}
grads
}

current.loglik <- -1e6;
iter <- 1;
err <- 10;
new.par=params

while((err > 1.0002 || err < 0.9998) && iter <= max.iter){
ptm <- proc.time()
# updating of \alpha, \beta and \gamma parameters
fitbern<-optim(par=new.par[-(n+2*p+1-st)],fn=lLbern2,gr=param.gradient,method = "BFGS
    ",control=list(trace=0,fnscale=-1), Z=Z,dat=dat)

new.par<-c(fitbern$par[1:(n+2*p-st)],0,fitbern$par[(n+2*p+1-st):(n+p+p*q-1-st)])
print((proc.time()-ptm)[3])
print(fitbern$value)
new.loglik <- fitbern$value
err <- abs(new.loglik/current.loglik); cat("New Loglik:", new.loglik, "Current Loglik
    :", current.loglik, "Ratio", err, "\n")
current.loglik <- new.loglik;

# set new parameters:
if(site){
a=new.par[1:n]
} else{a=rep(0,n)}
b=new.par[(n+1-st):(n+p-st)]
g.vec=new.par[(n+p+1-st):(n+p+p*q-st)]
G<-matrix(g.vec,nrow=q,byrow=TRUE)

## updating of Z
cat("Working out index for each spp\n")
        ptm <- proc.time()
```

```
index <- matrix(rep(0,num.lv*n),n,num.lv); big.simlv <- rmvnorm(20000,rep(0,num.lv));
     X.bigb <- cbind(rep(1,20000),rep(1,20000),big.simlv)
  for(i in 1:n) {
          spp.f <- matrix(NA, nrow = 20000, ncol = p)
          for(j in 1:p) {
                        beta <- c(a[i],b[j],G[,j]); eta <- X.bigb %*% beta;
      spp.att.mu <- exp(eta)/(1+exp(eta)); spp.f[,j] <- dbinom(dat[i,j],1, spp.att.mu
          ) }
  spp.posterior <- apply(spp.f, 1, prod) * dnorm(big.simlv[,1],0,1)
        if (num.lv==2) {  spp.posterior <- spp.posterior * dnorm(big.simlv[,2],0,1) }
  index[i,] <- big.simlv[which(spp.posterior == max(spp.posterior)),] }
        cat("Parameters calculated in time:\n")
        print((proc.time()-ptm)[3])

plot(index,type="n")
text(index,labels=seq(1,n))
points(t(Z),type="n")
text(t(Z),labels=seq(1,n),col=2)
Z=t(index)
cat("Number of iterations: ",iter,"\n")
iter = iter + 1
}
list(a=a,b=b,g=t(G),z=t(Z),ll=current.loglik,pars=new.par)
}


# Funktio joka sovittaa LVM-mallin, jossa selittajat mukana:
LVM_L_nb_pred<-function(data,X,num.lv,max.iter=50,z0="s",estimf=TRUE,site=FALSE,
    method="MAP"){
# estimf=TRUE jos \phi-parametrit estimoidaan
# site=TRUE, jos \alpha-parametri mallissa
n=dim(data)[1]
p=dim(data)[2]
q=num.lv<-2
m=dim(X)[2]
# Initial values for model parameters and latent variables
if(estimf){
para<-start.values(data,att.dist="nb")
f<-para$params[,4]
} else {
para<-start.values(data,att.dist="p")
}
bs<-para$params[,1]
gs<-c(para$params[,2],para$params[,3])
params<-c(bs,gs)
# Initials to \alpha
if(site){
aas<-rep(0,n)
params<-c(aas,params)
st=0
} else {st=n}
# Initials to \delta
d=rep(0,m*p)
params<-c(params,d)

#if(sum(Z)==0){
if(z0=="mds"){
bray <- vegdist(data)
bray <- metaMDS(bray,k=2)
Z=t(bray$points)
} else{ if(z0=="dca"){
CA.res<-decorana(data)
Z=t(CA.res$rproj[,1:2])
} else{ if(z0=="pcoa"){
PCoA.res<-capscale(data~1,distance="bray")
Z=t(scores(PCoA.res,display="sites"))
} else{ Z<-t(para$index)} } }
#}
```

```
# Initial values for scale parameters
if(!estimf){
f<-rep(0.001,p)
}

## log-likelihood
lLnb<-function(param,Z,data,f,X){
# dat is nxp data matrix
# ai=alfa_i
# b=t([beta_1,...,beta_p])
# G is qxp matrix, (q=2)
# z is qxp matrix, where z[,i]=t([z_i1,...,z_iq])
# f is p-vector
# n is number of sites
# p is number of species
# q is number of latent variables
if(site){
a=param[1:n]
} else{a=rep(0,n);}
b=param[(n+1-st):(n+p-st)]
g.vec=param[(n+p+1-st):(n+p+p*q-1-st)]
g.vec<-c(g.vec[1:p],0,g.vec[(p+1):(2*p-1)])
#f=param[(n+p+p*q+1):(n+p+p*q+p)]
G<-matrix(g.vec,nrow=q,byrow=TRUE)
#Z<-matrix(nrow=q,ncol=n)
d=param[(n+p+p*q-st):(n+p+p*q-1+m*p-st)]
D=matrix(d,ncol=p) #mxp matrix
zg=t(Z)%*%G
dX=X%*%D
mu.apu<-apply(t(apply(zg,1,function(x,w) x+w,w=b)),2,function(x,w) x+w,w=a)+dX

SGam=0
fsum=0
for(i in 1:n){
  SGam=SGam-0.5*log(det(Gammanb(data[i,],a[i],b,f,G,Z[,i],d,X[i,])))
  fsum=fsum+sum(log(dnbinom(data[i,],mu=exp(mu.apu[i,]),size=1/f)))
}
# log-likelihood
L= SGam + fsum - sum(diag(t(Z)%*%Z))/2
L
}

# Gamma matrix
Gammanb<-function(yi,ai,b,f,G,zi,d,x){
# ai=alfa_i
# b=[beta_1,...,beta_p]'
# G is qxp matrix, (q=2)
# zi= z_i=[z_i1,...,z_iq]'
# d=[d_1,...,d_p]
# dj=[d_j1,...,d_jm]'
# x=[x_i1,...,x_im]'
q=2
p=length(b)
D=matrix(d,ncol=p) #mxp matrix
my<-exp(t(zi)%*%G+b+ai+(t(x)%*%D))
s<-0
for(j in 1:p){
  s=s+ my[1,j]*f[j]*(yi[j]+1/f[j])/(1+f[j]*my[1,j])^2*(G[,j]%*%t(G[,j]))
}
s+diag(q)
}

# Q-function
Qfun<-function(param,Zi,datai,f,Xi,i){
if(site){
a=param[1:n]
} else{a=rep(0,n);}
b=param[(n+1-st):(n+p-st)]
```

```
g.vec=param[(n+p+1-st):(n+p+p*q-1-st)]
g.vec<-c(g.vec[1:p],0,g.vec[(p+1):(2*p-1)])
#f=param[(n+p+p*q+1):(n+p+p*q+p)]
G<-matrix(g.vec,nrow=q,byrow=TRUE)
#Z<-matrix(nrow=q,ncol=n)
d=param[(n+p+p*q-st):(n+p+p*q-1+m*p-st)]
D=matrix(d,ncol=p) #mxp matrix

zg=Zi%*%G
dX=Xi%*%D
oa<-a[i]+b+Zi%*%G+Xi%*%D
1/p*( sum(log(dnbinom(datai,mu=exp(oa),size=1/f)))-(t(Zi)%*%Zi)[1,1]/2 - log(2*pi))
}

# Gradients for \alpha, \beta, \gamma and \delta parameters
param.gradient<-function(param,Z,data,f,X){
if(site){
a=param[1:n];
} else{a=rep(0,n);}
b=param[(n+1-st):(n+p-st)]
g.vec=param[(n+p+1-st):(n+p+p*q-1-st)]
g.vec<-c(g.vec[1:p],0,g.vec[(p+1):(2*p-1)])
G<-matrix(g.vec,nrow=q,byrow=TRUE)
d=param[(n+p+p*q-st):(n+p+p*q-1+m*p-st)]
D=matrix(d,ncol=p) #mxp matrix
zg=t(Z)%*%G
dX=X%*%D
mu.apu=apply(t(apply(zg,1,function(x,w) x+w,w=b)),2,function(x,w) x+w,w=a)+dX
ex=exp(mu.apu)
yf=t(t(data)+1/f)

Gam=list();
grad.a=vector(length=n);
for(i in 1:n){
  Ga=Gammanb(data[i,],a[i],b,f,G,Z[,i],d,X[i,])
  Gam[[length(Gam)+1]] <- Ga
sa=matrix(0,ncol=2,nrow=2)
for(j in 1:p){
sa=sa+f[j]*ex[i,j]*(data[i,j]+1/f[j])*(1-f[j]*ex[i,j])/(1+f[j]*ex[i,j])^3*G[,j]%*%t(G
    [,j])
}
grad.a[i]=-0.5*tr(solve(Gam[[i]])%*%sa)+sum( data[i,]-yf[i,]*ex[i,]/t(1/f+ex[i,]) )
}

grad.b=vector(length=p)
grad.G=matrix(0,nrow=q,ncol=p)
grad.d=matrix(0,nrow=m,ncol=p);
for(r in 1:p){
  for(i in 1:n){
  grad.b[r]= grad.b[r] -0.5*tr( solve(Gam[[i]])%*%(f[r]*ex[i,r]*(yf[i,r])*
    (1-f[r]*ex[i,r])/(1+f[r]*ex[i,r])^3*G[,r]%*%t(G[,r])) ) +
    data[i,r] - yf[i,r]*ex[i,r]/(1/f[r]+ex[i,r]);
#
  for(h in 1:m){
  grad.d[h,r]=grad.d[h,r] -0.5*tr( solve(Gam[[i]])%*%(f[r]*ex[i,r]*(yf[i,r])*
    (1-f[r]*ex[i,r])/(1+f[r]*ex[i,r])^3*G[,r]%*%t(G[,r])*X[i,h]) ) +
    ( data[i,r] - yf[i,r]*ex[i,r]/(1/f[r]+ex[i,r]) )*X[i,h];
  }
#
  grad.G[1,r]= grad.G[1,r] - 0.5*tr( solve(Gam[[i]])%*%(f[r]*ex[i,r]*(yf[i,r])*
    (1-f[r]*ex[i,r])/(1+f[r]*ex[i,r])^3*Z[1,i]*G[,r]%*%t(G[,r]) + f[r]*ex[i,r]
    *(yf[i,r])/(1+f[r]*ex[i,r])^2*(kronecker(c(1,0),t(G[,r]))+
    kronecker(t(c(1,0)),G[,r])) ) +
    (data[i,r] - yf[i,r]*ex[i,r]/(1/f[r]+ex[i,r]))*Z[1,i];
#
  grad.G[2,r]= grad.G[2,r] - 0.5*tr( solve(Gam[[i]])%*%(f[r]*ex[i,r]*(yf[i,r])*
    (1-f[r]*ex[i,r])/(1+f[r]*ex[i,r])^3*Z[2,i]*G[,r]%*%t(G[,r]) + f[r]*ex[i,r]
    *(yf[i,r])/(1+f[r]*ex[i,r])^2*(kronecker(c(0,1),t(G[,r]))+
```

```
    kronecker(t(c(0,1)),G[,r])) ) +
    (data[i,r] - yf[i,r]*ex[i,r]/(1/f[r]+ex[i,r]))*Z[2,i];
  }
}
grads<-c(grad.b,grad.G[1,],grad.G[2,-1],c(grad.d))
if(site){
grads<-c(grad.a,grads)
}
grads
}

# Gradients for scale parameters \phi
f.gradient<-function(param,Z,data,f,X){
if(site){
a=param[1:n];
} else{a=rep(0,n);}
b=param[(n+1-st):(n+p-st)]
g.vec=param[(n+p+1-st):(n+p+p*q-1-st)]
g.vec<-c(g.vec[1:p],0,g.vec[(p+1):(2*p-1)])
G<-matrix(g.vec,nrow=q,byrow=TRUE)
d=param[(n+p+p*q-st):(n+p+p*q-1+m*p-st)]
D=matrix(d,ncol=p) #mxp matrix
zg=t(Z)%*%G
dX=X%*%D
mu.apu=apply(t(apply(zg,1,function(x,w) x+w,w=b)),2,function(x,w) x+w,w=a)+dX
ex=exp(mu.apu)
yf=t(t(data)+1/f)

grad.f=vector(length=p)
for(r in 1:p){
  for(i in 1:n){
  grad.f[r]= grad.f[r] - 0.5*tr( solve(Gammanb(data[i,],a[i],b,f,G,Z[,i],d,X[i,]))%*%
    (ex[i,r]*(data[i,r]-data[i,r]*ex[i,r]*f[r]-2*
    ex[i,r])/(1+f[r]*ex[i,r])^3*G[,r]%*%t(G[,r])) ) +
    1/f[r]^2*( log(1+f[r]*ex[i,r])-digamma(yf[i,r])+
    digamma(1/f[r]) ) - ex[i,r]*yf[i,r]/(1+f[r]*ex[i,r]) + data[i,r]/f[r];
}}
grad.f
}

current.loglik <- -1e6;
iter <- 1;
err <- 10;
new.par=params

while((err > 1.0002 || err < 0.9998) && iter <= max.iter){
ptm <- proc.time()
# updating of \alpha, \beta and \gamma parameters
fitgr<-try(optim(new.par[-(n+2*p+1-st)],fn=lLnb,gr=param.gradient,
method = "BFGS",control = list(trace = 0, fnscale = -1),
Z=Z,data=data,f=f,X=X),silent=T)
#old.par=new.par
new.par<-c(fitgr$par[1:(n+2*p-st)],0,fitgr$par[(n+2*p+1-st):(n+p+p*q+m*p-1-st)])
# updating of \phi -parameters
if(estimf){
fitgr<-try(optim(f,fn=lLnb,gr=f.gradient,method = "BFGS",
control = list(trace = 0, fnscale = -1),Z=Z,data=data,
param=new.par[-(n+2*p+1-st)],X=X),silent=T)
f<-fitgr$par
}

print((proc.time()-ptm)[3])
print(fitgr$value)
new.loglik <- fitgr$value
err <- abs(new.loglik/current.loglik);
cat("New Loglik:", new.loglik, "Current Loglik:", current.loglik, "Ratio", err, "\n")
current.loglik <- new.loglik;
# set new parameters:
```

```
if(site){
a=new.par[1:n]
} else{a=rep(0,n)}
b=new.par[(n+1-st):(n+p-st)]
g.vec=new.par[(n+p+1-st):(n+p+p*q-st)]
d=new.par[(n+p+p*q+1-st):(n+p+p*q+m*p-st)]
G<-matrix(g.vec,nrow=q,byrow=TRUE)
D=matrix(d,ncol=p) #mxp matrix

## updating of Z
  cat("Working out index for each spp\n")
  ptm <- proc.time()
index <- matrix(rep(0,num.lv*n),n,num.lv);
if(method=="MAP"){
big.simlv <- rmvnorm(20000,rep(0,num.lv));
X.bigb <- cbind(rep(1,20000),rep(1,20000),big.simlv)
  for(i in 1:n) {
       spp.f <- matrix(NA, nrow = 20000, ncol = p)
        for(j in 1:p) {
            beta <- c(a[i],b[j],G[,j]);
    eta <- X.bigb %*% beta+(t(X[i,])%*%D[,j])[1,1];
    spp.att.mu <- exp(eta); spp.f[,j] <- dnbinom(data[i,j], mu=spp.att.mu,size=1/f[j
        ]) }
  spp.posterior <- apply(spp.f, 1, prod) * dnorm(big.simlv[,1],0,1)
        if (num.lv==2) {  spp.posterior <- spp.posterior * dnorm(big.simlv[,2],0,1) }
  index[i,] <- big.simlv[which(spp.posterior == max(spp.posterior)),] }
} else{
# Maximize Q
for(i in 1:n){
fitz<-try(optim(Z[,i],fn=Qfun,method = "BFGS",
control = list(trace = 0, fnscale = -1),param=new.par[-(n+2*p+1-st)],
datai=data[i,],f=f,Xi=X[i,],i=i),silent=T)
index[i,]<-fitz$par
}
}
        cat("Parameters calculated in time:\n")
        print((proc.time()-ptm)[3])

# lets plot new and old Z to the same picture
plot(index,type="n")
text(index,labels=seq(1,n))
points(t(Z),type="n")
text(t(Z),labels=seq(1,n),col=2)
Z=t(index)
cat("Number of iterations",iter,"\n")
iter = iter + 1
}
# Hessian matrix
H1<-hessian(function(x){ lLnb(x,Z=Z,data,f,X)},x=new.par[-(n+2*p+1-st)])
list(a=a,b=b,g=t(G),d=D,f=f,z=t(Z),ll=current.loglik,pars=new.par,H=H1)
}


# Normaalijakaumasta generoidut alkuarvot latenteille muuttujille tuottava funktio
start.values <- function(data, att.dist="p", num.lv = 2) {
        N <- nrow(data)
        p <- ncol(data)
        data <- as.matrix(data)
        allb<-NULL
        par <- array(0, dim=c(p,num.lv+2,10))
        for (j in 1:10) { allb <- cbind(allb,rmvnorm(N,rep(0,num.lv))) }
        AIC <- NULL
            if(att.dist=="b") {
                  for (i in 1:10) {
                        fit <- manyglm(data~allb[,(2*i-1):(2*i)],family="binomial")
                        par[,,i] <- cbind(t(fit$coef),rep(NA,p))
                        AIC <- c(AIC,sum(fit$AIC)) }
                        m <- (which(AIC == min(AIC)))[1]
```

```
                        params <- par[,,m]
                        index <- allb[,(2*m-1):(2*m)] }
            if(att.dist=="p") {
                  for (i in 1:10) {
                        fit <- manyglm(data~allb[,(2*i-1):(2*i)],family="poisson")
                        par[,,i] <- cbind(t(fit$coef),rep(NA,p))
                        AIC <- c(AIC,sum(fit$AIC)) }
                        m <- (which(AIC == min(AIC)))[1]
                        params <- par[,,m]
                        index <- allb[,(2*m-1):(2*m)] }
            if(att.dist=="n") {
                  for (i in 1:10) {
                        fit <- manyglm(data~allb[,(2*i-1):(2*i)],family="normal")
                        get.phi <- fit$phi
                        par[,,i] <- cbind(t(fit$coef),get.phi)
                        AIC <- c(AIC,sum(fit$AIC)) }
                        m <- (which(AIC == min(AIC)))[1]
                        params <- par[,,m]
                        index <- allb[,(2*m-1):(2*m)] }
            if(att.dist=="nb") {
                  for (i in 1:10) {
                        fit <- manyglm(data~allb[,(2*i-1):(2*i)],family="negative.
                            binomial")
                        get.phi <- fit$phi; get.phi[get.phi == 0] <- 1e-5
                        par[,,i] <- cbind(t(fit$coef),get.phi)
                        AIC <- c(AIC,sum(fit$AIC)) }
                        m <- (which(AIC == min(AIC)))[1]
                        params <- par[,,m]
                        index <- allb[,(2*m-1):(2*m)] }
        list(params=params,index=index)
}


# Funktio, joka laskee Dunn-Smyth-residuaalit
pit.res <- function(data, index, params, att.dist, site.params) {
        pitres <- data
        d <- dim(params)[2]; p <- dim(data)[2]; n <- dim(data)[1]
        if(length(att.dist) == 1) att.dist <- rep(att.dist[1],ncol(data))
        palette(rainbow(p))
        for (a in 1:p) { pitres[,a] <- pit.site(data[,a],index,params[a,],site.params
            ,family = att.dist[a]) }
        ## linear predictor
        lin.pred <- cbind(rep(1,dim(data)[1]),index) %*% t(params[,1:(d-1)])+site.
            params
        par(mfrow=c(2,2), mar=c(2,2,2,2), cex=1)
        matplot(lin.pred, pitres, ylab="Dunn-Smyth Residuals", xlab="Linear
            Predictors", type="n", xlim=c(-4,max(lin.pred)))
        for(i in 1:p) { points(lin.pred[,i],pitres[,i],col=palette()[i]) }
        abline(0,0,lty=2)
        matplot(t(pitres),ylab="Dunn-Smyth Residuals",xlab="Species",type="n")
        for(i in 1:p) { points(rep(i,n),pitres[,i],col=palette()[i]) }
        #for(i in 1:p) { points(apply(y,2,sum),pitres[,i],col=palette()[i]) }
        abline(0,0,lty=2)

        matplot(pitres,ylab="Dunn-Smyth Residuals",xlab="Sites Total",type="n")
        for (i in 1:p) { points(seq(1,n),pitres[,i],col=palette()[i]) }
        abline(0,0,lty=2)
        #return(pitres)
ind<-as.vector(unlist(pitres))<100000
        qqnorm(as.vector(unlist(pitres))[ind], main = "Normal Quantile Plot"); abline
            (0,1)
#         qqnorm(as.vector(unlist(pitres)), main = "Normal Quantile Plot"); abline
    (0,1)

        palette("default")

        return(list(res = pitres, linpred = lin.pred))
        }
```

```
pit.site <- function(y, index, params, site.params, family) {
        N <- length(y)
        Z <- cbind(rep(1,N),index)
        d <- length(params)

        mu <- exp(Z%*%params[1:(d-1)]+site.params)
        if(family == "poisson") { a <- ppois(y - 1, mu); b <- ppois(y, mu) }
        if(family == "negative.binomial") { a <- pnbinom(y-1,size=1/params[d],mu=mu);
             b <- pnbinom(y,size=1/params[d],mu=mu) }
        if(family == "binomial") {
                p <-  exp(Z%*%params[1:(d-1)]+site.params)/(1+exp(Z%*%params[1:(d-1)
                    ]+site.params))
                a <- pbinom(y - 1, 1, p); b <- pbinom(y, 1, p) }
        u <- runif(n = length(y), min = a, max = b)
        qnorm(u)
        }
```

./rkoodi.R