UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN
INFORMACIJSKE TEHNOLOGIJE

Zaključna naloga

Primerjava modernih pristopov za identifikacijo pomembno izraženih genov za dve skupini

(Comparison of modern approaches for identifying importantly expressed genes for two groups)

Ime in priimek: Pino Simonovich
Študijski program: Bioinformatika
Mentor: doc. dr. Rok Blagus

Koper, junij 2017

Ključna dokumentacijska informacija

Ime in PRIIMEK: Pino SIMONOVICH

Naslov zaključne naloge: Primerjava modernih pristopov za identifikacijo pomembno izraženih genov za dve skupini

Kraj: Koper

Leto: 2017

Število listov: 69	Število slik: 7	Število tabel: 24
Število prilog: 5	Število strani prilog: 31	Število referenc: 14

Mentor: doc. dr. Rok Blagus

Ključne besede: Visokorazsežni podatki, izbira spremenljivk, napovedna točnost, logistična regresija, naključni gozdovi, najbližji skrčeni centroid

Izvleček:
Visokorazsežna statistika se ukvarja s podatki, kjer je število spremenljivk p veliko večje od števila enot N v vzorcu. V takih primerih ne moremo neposredno uporabiti klasičnih statističnih metod in potrebujemo določeno mero regularizacije kompleksnosti, da bi se izognili preprileganju. Posledično je potrebno zmanjšati število spremenljivk na podmnožico spremenljivk, ki imajo morebiten vpliv na obravnavani izid.
Izbor spremenljivk je postopek, s katerim izberemo podmnožico spremenljivk, ki jih bomo uporabili za gradnjo modela. Ta se uporablja iz več razlogov: (i) poenostavitev modela za lažjo interpretacijo, (ii) krajši čas učenja, (iii) da bi se izognili prekletstvu dimenzij (*angl.* curse of dimensionality; pojav v visokorazsežnih podatkih, ki povzroči zelo razpršene podatke), (iv) izboljšana posplošitev z zmanjšanjem preprileganja (z zmanjšanjem variance).
Uporabili in primerjali bomo tri glavne metode: logistična regresija, naključni gozdovi ter metoda najbližjega skrčenega centroida. Njihovo delovanje bomo ovrednotili na simuliranih podatkih in na pravih genskih podatkih.

Key words documentation

Name and SURNAME: Pino SIMONOVICH

Title of final project paper: Comparison of modern approaches to identifying important expressed genes for two groups

Place: Koper

Year: 2017

Number of pages: 69 Number of figures: 7 Number of tables: 24

Number of appendices: 5 Number of appendix pages: 31 Number of references: 14

Mentor: Assist. Prof. Rok Blagus, PhD

Keywords: High-dimensional data, variable selection, prediction accuracy, logistic regression, random forests, nearest shrunken centroid

Abstract:
High-dimensional statistics deals with data where the number of variables p is much larger than the number of observations N. This means that classical statistical methods cannot be applied directly, and one needs a certain amount of complexity regularization to avoid overfitting. As a result, it is necessary to reduce the number of variables to a subset of predictors that potentially impact the outcome of interest.
Feature selection is the process of selecting a subset of relevant features (variables) for use in model construction. Feature selection techniques are used for many reasons: (i) simplification of models to make them easier to interpret by researchers/users, (ii) shorter training times, (iii) to avoid the curse of dimensionality, (iv) enhanced generalization by reducing overfitting (formally, reduction of variance).
We will use and compare three main methods: logistic regression, random forests and nearest shrunken centroids. We will evaluate their performance on simulated data and on real genomic data.

Zahvala

Zahvaljujem se mentorju za vse popravke, pojasnila ter za hitro odzivnost.

Zahvaljujem se tudi osebju fakultete, ki mi je dovolilo uporabljati računalniško učilnico za izvedbo simulacij.

Kazalo vsebine

Kazalo tabel

Kazalo slik

Kazalo prilog

Seznam kratic

LR	Logistična Regresija (*angl.* Logistic Regression)
EN	Elastična mreža (*angl.* Elastic Net)
$Bolasso$	Lasso s Samovzorčenjem (*angl.* Bootstrapped lasso)
$Boelastic$	Elastična Mreža s Samovzorčenjem (*angl.* Bootstrapped Elastic Net)
RF	Naključni Gozdovi (*angl.* Random Forests)
RFE	Rekurzivno Odstranjevanje Spremenljivk (*angl.* Recursive Feature Elimination)
OOB	vzorci Iz Vrečke (*angl.* Out Of Bag samples)
CV	Prečno Preverjanje (*angl.* Cross-Validation)
NC	Metoda Najbližjega Centroida (*angl.* Nearest Centroid)
NSC	Metoda Najbližjega Skrčenega Centroida (*angl.* Nearest Shrunken Centroid)
PAM	Analiza Napovedi za Mikromreže (*angl.* Prediction Analysis of Microarrays)
$BoPAM$	PAM s Samovzorčenjem (*angl.* Bootstrapped PAM)
TDR	Resnična Stopnja Odkritja (*angl.* True Discovery Rate)
FDR	Napačna Stopnja Odkritja (*angl.* False Discovery Rate)
$p.corr$	Odstotek Pravilnih (*angl.* Percent Correct)
$p.false$	Odstotek Nepravilnih (*angl.* Percent False)
PA	Napovedna Točnost (*agl.* Predictive Accuracy)
$PA1$	Napovedna Točnost za prvi razred (*angl.* Predictive Accuracy 1)
$PA2$	Napovedna Točnost za drugi razred (*angl.* Predictive Accuracy 2)
GM	Geometrično Povprečje (*angl.* Geometric Mean)
AUC	Ploščina Pod ROC krivuljo (*angl.* Area Under the ROC curve)
ROC	Sprejemnikova Operacijsko Karakteristična krivulja (*angl.* Receiver Operating Characteristic curve)

1 Uvod

Nove tehnologije za merjenje izražanja genov so sprožile raziskovanje in razvoj podobnih orodij na vseh "omičnih"področjih (genomika, proteomika, ...). Glavna značilnost tako pridobljenih podatkov je njihova visoka razsežnost, kar pomeni, da je število merjenih spremenljivk mnogo večje od števila enot. V zadnjih desetih letih je produkcija in uporaba visokorazsežnih podatkov v biomedicinskem raziskovanju skokovito narasla in razvitih je bilo več statističnih metod za pravilno analizo takšnih podatkov.

Biomedicinske raziskave, ki uporabljajo visokorazsežne podatke, imajo običajno dva cilja (i) odkriti pravila, po katerih natančno napovemo razredno pripadnost novih enot (napovedovanje razreda) in (ii) identificirati spremenljivke, katerih porazdelitve se ločijo med vnaprej določenimi razredi (primerjava razredov). V zaključni nalogi se bomo osredotočili na cilj (ii).

V nalogi bomo primerjali moderne metode za izbiro spremenljivk, med katere spadajo logistična regresija, naključni gozdovi ter metode skrčenega najbližjega centroida. Te metode so se izkazale, kot zelo uspešne pri doseganju cilja (i), medtem ko njihova uporabnost za cilj (ii) še ni bila podrobno raziskana.

Učinkovitost metod bomo preverili s simulacijami, njihovo uporabnost pa bomo ilustrirali tudi na pravih genskih podatkih s področja onkologije. V simulacijah bomo preučevali vpliv velikosti vzorca, velikosti razlike med skupinami za resnično različno izražene spremenljivke ter moč korelacije med spremenljivkami, kjer bomo uporabili bločno korelacijsko strukturo.

Pomemben vidik visokorazsežnih podatkov je potreba po izboru spremenljivk (*angl.* variable selection). Izbor spremenljivk je identifikacija podmnožice spremenljivk, ki bodo uporabljene za klasifikacijsko pravilo ter za identifikacijo pomembnih spremenljivk. To lahko storimo preden razvijemo klasifikator ali pa je lahko vgrajeno v klasifikacijsko metodo [10]. Pomembnost izbora spremenljivk za visoko razsežne podatke temelji na dveh dejstvih: nekaterih klasifikacijskih pravil ne moremo izgraditi, če je število spremenljivk večje od števila opažanj in odstranjevanje spremenljivk z majhno variabilnostjo izboljša napovedno natančnost [5].

V tej nalogi se bomo osredotočili na probleme za razredno-uravnotežene (*angl.* class-balanced) podatke, tj. podatkovne množice, kjer je število opažanj pripadajoč vsakemu razredu enako. Razredno-neuravnoteženi podatki so sicer pogosti na biomedicinskem

področju in se pojavijo tudi, ko imamo visokorazsežne podatke. Za več informacij o neuravnoteženih podatkih in kako lahko pridobimo uravnotežene podatke iz neuravnoteženih podatkov, glej [2].

Za preiskovanje učinkovitosti izbora spremenljivk za visokorazsežne podatke smo ocenili delovanje treh vrst klasifikatorjev ter njihovih podvrst za razredno-uravnotežene podatke. Klasifikacijske metode so bile izbrane izmed najbolj priljubljenih za visokorazsežne podatke, za poenostavitev pa smo obravnavali zgolj klasifikacijske probleme z dvema razredoma (dvorazredni klasifikacijski problemi). Klasifikatorji so bili ocenjeni na simuliranih podatkih ter na prosto dostopnih genskih podatkih (*angl.* breast cancer gene expression microarray study [11]). Simulirali smo stanja z različnimi stopnjami razlik med razredi.

V poglavju Metode opišemo uporabljene klasifikacijske metode in dodatne spremembe le teh za izbor spremenljivk; na kratko opišemo tudi simulacije ter prave podatke iz [11].

V poglavju Rezultati predstavimo izvedene simulacije in bolj podroben pregled izbranih metod ter njihovo delovanje za visokorazsežne podatke. Nato predstavimo delovanje metod ter rezultate na pravih podatkih.

2 Metode

2.1 Logistična Regresija

Predpostavimo, da imamo N opažanj (enot) v vzorcu. Vsako opažanje i sestoji iz množice p spremenljivk $x_{1,i}, \ldots, x_{p,i}$ in povezanim binarnim izidom Y_i, tj. lahko zavzame le dve vrednosti, 0 ali 1. Cilj logistične regresije [7] je razložiti razmerje med spremenljivkami ter izidi, zato da lahko napovemo izid za novo množico spremenljivk.

Izidi Y_i imajo Bernoullijevo porazdelitev, kjer je vsak izid določen z neznano verjetnostjo p_i, ki je specifična danemu izidu in je povezana s spremenljivkami. To lahko izrazimo v naslednji obliki:

$$P(Y_i = y_i | x_{1,i}, \ldots, x_{m,i}) = p_i^{y_i}(1 - p_i)^{(1-y_i)}. \tag{2.1}$$

Ta izraz se nanaša na dejstvo, da Y_i lahko zavzame le vrednosti 0 ali 1. V vsakem primeru bo en eksponent enak 1, drugi eksponent pa bo enak 0, zato bo izid vedno ali p_i ali $1 - p_i$.

Temeljna ideja logistične regresije je uporaba mehanizmov že razvitih v linearni regresiji z uporabo linearne napovedne funkcije za modeliranje verjetnosti p_i, tj. linearna kombinacija spremenljivk in množico regresijskih koeficientov, ki so specifični za obravnavani model, vendar enaki za vse poskuse. Linearna napovedna funkcija $f(i)$ za opažanje i je zapisana kot:

$$f(i) = \beta_0 + \beta_1 x_{1,i} + \cdots + \beta_p x_{p,i} = \boldsymbol{\beta} \cdot \boldsymbol{X_i}, \tag{2.2}$$

kjer so $\beta = \beta_0, \ldots, \beta_p$ regresijski koeficienti, ki izražajo relativni učinek spremenljivk na izid, in $X_i = x_{0,i}, x_{1,i}, \ldots, x_{p,i}$ so vrednosti spremenljivk, kjer je $x_{0,i}$ dodan konstantni člen.

Model, ki ga uporablja logistična regresija, je izražen s povezavo med verjetnostjo izida in linearne napovedne funkcije:

$$\text{logit}(p_i) = \ln\left(\frac{p_i}{1 - p_i}\right) = \boldsymbol{\beta} \cdot \boldsymbol{X_i}. \tag{2.3}$$

Ta formulacija izrazi logistično regresijo v obliki splošnega linearnega modela, ki napoveduje spremenljivke z raznimi verjetnostnimi porazdelitvami.

Verjetnost p_i in regresijski koeficienti so neznani in njihovo določanje ni del modela. Ti so običajno določeni z nekim optimizacijskim postopkom, npr. z metodo največjega verjetja, ta najde vrednosti, ki najbolje opisujejo podatke.

2.1.1 Lasso

Cilj lasso regresije [13] je reševanje

$$\min_{\beta\in\mathbb{R}^p}\left\{\frac{1}{N}\|Y-\boldsymbol{X}\boldsymbol{\beta}\|_2^2\right\} \text{ pri pogoju } \|\boldsymbol{\beta}\|_1 \leq t. \tag{2.4}$$

Tukaj je t preddoločen parameter, ki določa količino regularizacije, in $\|Z\|_p = \left(\sum_{i=1}^{N}|Z_i|^p\right)^{1/p}$ je standardna ℓ^p norma.

Zgornjo enačbo lahko zapišemo bolj kompaktno v *Lagrangeovi obliki*

$$\min_{\beta\in\mathbb{R}^p}\left\{\frac{1}{N}\|Y-\boldsymbol{X}\boldsymbol{\beta}\|_2^2+\lambda\|\boldsymbol{\beta}\|_1\right\}, \tag{2.5}$$

kjer je razmerje med t in λ odvisno od podatkov.

To lahko primerjamo z ridge regresijo, kjer cilj je reševanje

$$\min_{\beta\in\mathbb{R}^p}\left\{\frac{1}{N}\|Y-\boldsymbol{X}\boldsymbol{\beta}\|_2^2+\lambda\|\boldsymbol{\beta}\|_2^2\right\}. \tag{2.6}$$

Medtem ko ridge regresija normira vse koeficiente za konstantno vrednost, lasso premakne koeficiente proti ničli za konstanto vrednost, in jih postavi na ničlo, če jo dosežejo.

2.1.2 Elastic Net

Elastic net regularizacija [14] dopolni lasso z ridge regresijsko podobno kaznijo, ki izboljša delovanje, ko je število spremenljivk večje od števila enot v vzorcu ($p > N$). Metoda je zmožna izbrati močno korelirane spremenljivke skupaj, česa z lasso tipično ne dosežemo.

V primeru, ko $p > N$, lasso lahko izbere le N spremenljivk (tudi v primeru, ko jih je več povezanih z izidom) in teži k izbiri le ene spremenljivke iz množice visoko koreliranih spremenljivk. Dodatno, tudi ko $N > p$, če so spremenljivke močno korelirane, ridge regresija deluje bolje.

Elastic net razširi lasso z dodatno ℓ^2 kaznijo

$$\min_{\beta\in\mathbb{R}^p}\left\{\frac{1}{N}\|Y-\boldsymbol{X}\boldsymbol{\beta}\|_2^2+\lambda_1\|\boldsymbol{\beta}\|_1+\lambda_2\|\boldsymbol{\beta}\|_2^2\right\}. \tag{2.7}$$

Rezultat elastic net kazni je kombinacija učinkov lasso ter ridge kazni.

Visoko korelirane spremenljivke bodo imele podobne regresijske koeficiente, kjer bo stopnja podobnosti odvisna od $\|Y\|_1$ ter λ_2, kar je precej različno od lasso. Ta pojav,

ko imajo visoko korelirane spremenljivke podobne regresijske koeficiente, je imenovan združevalni učinek (*angl.* grouping effect) in je v splošnem zaželen v raznih aplikacijah, npr. v identifikaciji vseh genov, povezanih z boleznijo, namesto da bi izbrali le en gen iz vsake množice genov z visokimi korelacijami, kar počne lasso.

2.2 Naključni gozdovi

Odločitvena drevesa [6], ki so zgrajena zelo globoko, privedejo do preprileganja podatkom, tj. imajo nizko pristranskost, vendar zelo visoko varianco. Naključni gozdovi [4] povprečijo veliko globokih odločitvenih dreves, ki so zgrajeni na različnih delih istih podatkov, z namenom zmanjšanja variance.

Algoritem za naključne gozdove uporablja splošno tehniko samovzorčenja (*angl.* bootstrap aggregating ali bagging [3]) med gradnjo dreves. Z dano učno množico $X = x_1, \ldots, x_N$ in izidi $Y = y_1, \ldots, y_N$, samovzorčenje večkrat (B-krat) izbere naključen vzorec s ponavljanjem iz učne množice in zgradi drevesa nad temi vzorci:

Za $b = 1, \ldots, B$:

1. Vzorči, s ponavljanjem, B učnih enot iz X, Y; imenuj jih X_b, Y_b.
2. Zgradi klasifikacijsko ali regresijsko drevo f_b nad X_b, Y_b.

Po gradnji, napovedi za nove vzorce x' lahko izvedemo s povprečenjem napovedi vseh individualnih regresijskih dreves na x':

$$\hat{f} = \frac{1}{B} \sum_{b=1}^{B} f_b(x') \tag{2.8}$$

ali z upoštevanjem večinskega razreda v primeru klasifikacije.

Ta samovzorčni (*angl.* bootstrap) postopek privede do boljšega delovanja modela zato, ker zniža varianco modela, ne da bi zvišal pristranskost. Kar pomeni, da čeprav so napovedi posameznih dreves zelo občutljive na šum v učni množici, povprečje mnogih dreves ni, če drevesa niso korelirana. Grajenje velikega števila dreves na isti učni množici bi ustvarilo močno korelirana drevesa; samovzorčenje je način, kako de-korelirati drevesa, tako da vsakemu drevesu pokažemo različne učne množice.

Število zgrajenih dreves, B, je prosti parameter. Običajno je zgrajenih nekaj sto ali več tisoč dreves, odvisno od velikosti in narave podatkov. Optimalno število dreves B lahko najdemo z uporabo prečnega preverjanja (*angl.* Cross-Validation) ali z uporabo OOB (*angl.* out-of-bag) napake: povprečna napovedna napaka vseh dreves; napovedno točnost b-tega drevesa ocenimo z uporabo vseh enot x_i učne množice, ki niso prisotne v samovzorčnem vzorcu b-tega drevesa.

Naključni gozdovi imajo še eno dodatno lastnost: algoritem izbira, na vsakem kandidatnem razcepu v gradnji, naključno podmnožico spremenljivk. Ta postopek je imenovan tudi samovzorčenje spremenljivk (*angl.* feature bagging). Razlog za ta postopek je prisotnost korelacije med drevesi v splošnem samovzorčnem vzorcu: če ima ena ali več spremenljivk zelo močan vpliv na izid, bodo te spremenljivke izbrane v večini izmed B dreves, kar bo privedlo do korelacije med drevesi.

Za klasifikacijske probleme s p spremenljivkami, je $\lfloor\sqrt{p}\rfloor$ (zaokroženo navzdol) spremenljivk izbranih na vsakem razcepu. Za regresijske probleme avtorji priporočajo $\lfloor p/3\rfloor$ (zaokroženo navzdol) spremenljivk z minimalnim številom enot v vozlišču enako 5.

Pomembnost spremenljivk

Naključne gozdove lahko uporabimo za izračun pomembnosti spremenljivk (*angl.* variable importance) za regresijski ali klasifikacijski problem.

Zgradimo naključni gozd nad podatki, OOB napaka za vsako enoto je zabeležena in povprečena. Nato permutiramo vrednosti j-te spremenljivke v učni množici in OOB napaka je zopet izračunana na teh permutiranih podatkih. Pomembnost j-te spremenljivke je izračunana s povprečeno razliko OOB napake pred in po permutaciji. Ocena je normalizirana s standardno deviacijo teh razlik.

Spremenljivke, ki pridobijo velike vrednosti v tej meri, so uvrščene kot bolj pomembne v primerjavi s spremenljivkami, ki imajo majhne vrednosti.

2.3 Najbližji Skrčeni Centroid

Metoda izračuna standardiziran centroid za vsak razred. To je povprečna vrednost vsake spremenljivke v vsakem razredu deljeno z znotraj-razredno standardno deviacijo te spremenljivke.

Metoda najbližjega centroida [7] sprejme vrednosti spremenljivk novega vzorca, in te primerja z vsakim razrednim centroidom. Razred, ki vsebuje centroid z najmanjšo kvadrirano razdaljo do novega vzorca, je napovedan razred za nov vzorec.

Metoda najbližjega skrčenega centroida (v biomedicinskih področjih metoda je znana pod imenom PAM - *angl.* Prediction Analysis of Microarrays [12]) ima dodatno pomembno spremembo metode najbližjega centroida. Ta skrči vsak razredni centroid proti skupnemu centroidu vseh razredov za neko vrednost, ki jo imenujemo prag (*angl.* threshold). To skrčenje sestoji iz premikanja centroida proti ničli s pragom, in postavitev na ničlo, če dosežemo ničlo.

Po skrčitvi centroidov, je nov vzorec klasificiran z običajnim pravilom najbližjega centroida, kjer uporabimo skrčene razredne centroide.

Ta skrčitev ima dve prednosti: 1) lahko privede do bolj točnega klasifikatorja z zmanjšanjem učinka šumnih spremenljivk, 2) počne samodejno izbiro spremenljivk. Konkretno, če je spremenljivka skrčena na ničlo za vse razrede, je odstranjena iz napovednega pravila. Druga možnost je, da postavimo to spremenljivko na ničlo za vse razrede razen enega, in s tem ugotovimo, da je visoka ali nizka vrednost te spremenljivke značilna za ta razred.

Uporabnik določi vrednost praga. Običajno se preuči več različnih pragov. Za usmerjanje v tej izbiri PAM naredi K-kratno prečno preverjanje (*angl.* K-Fold Cross-Validation) za različne vrednosti praga. Vzorci so razdeljeni naključno v K približno enako velikih delov. Za vsak del, je klasifikator zgrajen na ostalih $K-1$ delih in je nato ocenjen na izpuščenem delu. To je ponovljeno za različne vrednosti praga, in CV klasifikacijska napaka je zabeležena za vsako vrednost praga. Običajno bi uporabnik izbral vrednost praga, ki vrne minimalno CV napako.

2.4 Simulacije

Za vse zgoraj omenjene metode smo zgradili modele na simuliranih podatkih. Ustvarili smo sintetične podatke DNA mikromrež. Za vsak vzorec smo tudi določili pripadajoč razred, glede na diferenčno izraženost genov/spremenljivk.

Enote so bile normalizirane (*angl.* mean-centered), medtem ko spremenljivke niso bile. Tako vzorci učne množice, kot vzorci testne množice so bili uravnoteženi, tj. imeli so enako število opažanj v obeh razredih. Vsaka simulacija je bila ponovljena 750 krat.

Simulirali smo 1000 spremenljivk iz multivariatne Gaussove porazdelitve. Uporabili smo bločno izmenljivo korelacijsko strukturo, kjer so bile spremenljivke v istem bloku korelirane (parna korelacija ρ), medtem ko so bile spremenljivke ostalih blokov neodvisne; vsak blok je vseboval 10 spremenljivk in vse variance so bile enake 1. Izmed vseh 1000 spremenljivk, je bilo 100 resnično različno izraženih med razredoma.

V simulacijah smo spreminjali: velikost vzorca (`sample`), povprečno razliko med skupinami (`mde`) ter korelacijo med spremenljivkami (`rho`). Za prve simulacije smo za vzorce izbrali velikosti `50,100,250`; za povprečno razliko med skupinami `0,0.5,1`; za korelacijski koeficient med spremenljivkami `0,0.5,0.8`.

Modele ter mero razlikovalne točnosti (*angl.* variable accuracy) smo zgradili/ocenili na učni množici, medtem ko smo napovedno točnost ocenili na testni množici. Vse uporabljene mere so predstavljene v tabeli 1.

Statistična analiza in simulacije so bile izvedene z R jezikom za statistično računanje (R verzija 3.3.3) [9].

Tabela 1: Uporabljene mere v modelih.

	Mera	Opis
1	p.corr	odstotek pravilno napovedanih genov/spremenljivk
2	p.false	odstotek nepravilno napovedanih genov/spremenljivk
3	TDR	*angl.* True Discovery Rate - delež resnično različno izraženih spremenljivk
4	FDR	*angl.* False Discovery Rate - obrat TDR
5	PA	Napovedna Točnost - odstotek pravilno napovedanih enot za oba razreda
6	PA1	odstotek pravilno napovedanih enot iz prvega razreda
7	PA2	odstotek pravilno napovedanih enot iz drugega razreda
8	GM	Geometrično Povprečje - mera točnosti, pogosto uporabljena za neuravnotežene podatke, ki zajame delovanje klasifikatorjev v vseh razredih [7]
9	AUC	Ploščina Pod ROC krivuljo izračunana s funkcijo `somers2` iz paketa `hmisc`

Uporaba na pravih podatkih

Avtorji v [11] so analizirali cDNA gensko profilnih ekspresij 99 tumornih primerkov za bolnike z rakom na dojkah. Poleg 7650 predprocesiranih vrednosti izraženosti genov, kot opisano v članku, je na voljo tudi standardna prognostična informacija spremenljivk za vsakega bolnika (`https://linus.nci.nih.gov/~brb/DataArchive.html`).

Če je bila kakšna vrednost izražanja v vzorcih pod 100, so avtorji te vrednosti postavili na 100 (*angl.* thresholding) in nato izračunali log vrednost (in podobno so naredili v primeru saturacije). Če pa sta tako vzorec kot referenčni vzorec imela vrednost izražanja pod 100, so te vrednosti postavili na NA. Te vrednosti smo postavili na nič. S tem, ko smo postavili te vrednosti na ničlo, smo predpostavili, da ni razlike v izraženosti med vzorcem in referenčnim vzorcem, kar je običajen postopek v takih primerih. Ker to naredimo na enak način za vse vzorce, ne bomo uvedli pristranskosti v podatkih.

Obravnavamo dvorazredni klasifikacijski problem: prvi problem je bil napoved statusa estrogen receptorja (ER), ki je bil negativen (ER-) za 34 bolnikov in pozitiven (ER+) za 65 bolnikov; drugi problem je bil napoved stopnje tumorja, ki je bila 1 ali 2 za 54 bolnikov in 3 za 45 bolnikov.

Na teh podatkih smo ocenili delovanje klasifikatorjev, uporabljenih v simulacijah. Napovedne točnosti klasifikatorjev smo izračunali z uporabo 5-kratnega prečnega preverjanja (*angl.* 5-fold CV) ponovljene 250 krat. Za izbor spremenljivk smo si zabeležili vse uporabljene spremenljivke za model.

2.4.1 Modeli, Metode ter Parametri

Filter

Uporabimo funkcijo `filterVarImp` iz paketa `caret` [8]. Pomembnost vsakega gena je določena individualno z uporabo "filter" metode. Za klasifikacijo izvede ROC analizo za vsako spremenljivko. Za dvorazredni problem uporabi niz mejnih vrednosti nad podatki za napoved razreda. Občutljivost (*angl.* sensitivity) in specifičnost (*angl.* specificity) sta izračunani za vsako mejno vrednost in ROC je izračunana. Za izračun površine pod ROC krivuljo uporabi trapezoidno pravilo, ta je uporabljena kot mera pomembnosti spremenljivke (*angl.* variable importance).

Ker funkcija vrne vrednosti med 0 in 1 za vsako spremenljivko, metoda ne naredi nobene izbire spremenljivk. Za izbiro spremenljivk smo določili vrednost praga, nad katerim bomo obdržali spremenljivke in jih uporabili za izračun mere razlikovalne točnosti (aktivne spremenljivke), ostale spremenljivke pa bomo postavili na ničlo in jih ne bomo uporabili za izračun (neaktivne spremenljivke); vrednost praga smo fiksirali na 2/3 (metodo bomo predstavili z zapisom filter.caret).

Naključni gozdovi

Uporabimo funkcijo `randomForest` iz paketa `randomForest`. Zgradimo naključni gozd s privzetimi nastavitvami `ntree = 1000` ter `mtry` = $\sqrt{1000}$. Za izračun pomembnosti spremenljivk uporabimo tri funkcije: funkcija `varImp` iz paketa `caret` uporabi OOB meritve za izračun pomembnosti (zapis rf.caret), funkcija `varUsed` iz paketa `randomForest` vrne vse uporabljene spremenljivke v vseh drevesih (zapis rf.var.used), ter funkcija `varUsed` z dodatnim parametrom `count = TRUE` vrne frekvenco vseh uporabljenih spremenljivk v vseh drevesih in izberemo samo tiste spremenljivke, ki imajo frekvenco večjo ali enako $\sqrt{1000} = 31$ (zapis rf.var.used.freq), kjer je 1000 vrednost parametra `ntree`.

Štori

Zgradimo naključni gozd štorov (*angl.* stumps), tj. dreves, ki imajo le korenino in dva lista (globina = 1). Uporabimo privzeto nastavitev `ntree = 1000`, za `mtry` pa uporabimo RFE (Rekurzivno Odstranjevanje Spremenljivk) za pridobitev seznama vrednosti: na prvem koraku nastavimo `mtry = 1000` oz. uporabimo vse spremenljivke, nato pa na vsakem nadaljnjem koraku zmanjšamo trenutno število za 30 %. Za vsak štor izračunamo pomembnost spremenljivk in na vsakem naslednjem koraku izberemo le največje absolutne vrednosti spremenljivk, ostale vrednosti spremenljivk pa postavimo na ničlo. Na koncu izberemo tri modele, in sicer `mtry = 31` (zapis

stump.mtry.31), `mtry = 1000` (zapis stump.mtry.1000) ter tisto vrednost `mtry`, ki ima najmanjšo OOB napako med vsemi modeli (zapis stump.mtry.min.OOB). Za te tri modele uporabimo funkcijo `varUsed` za izračun mer razlikovalne točnosti.

Naključni gozdovi RFE

Podobno kot pri štorih, uporabimo RFE za postopno izločanje spremenljivk s 30 % korakom. Za izbiro spremenljivk na vsakem koraku uporabimo dve meri z uporabo funkcije `importance` iz paketa `randomForest` za dva ločena modela, `scale = TRUE` ter `scale = FALSE`. Za `scale = TRUE`, normaliziramo ocene vsake spremenljivke tako, da oceno pomembnosti spremenljivke delimo s standardno deviacijo te spremenljivke, drugače pa tega ne naredimo. Na koncu izberemo model, ki ima najmanjšo OOB oceno napake, tako za scaled (zapis rfe.scaled) kot za non-scaled (zapis rfe.non.scaled) primer.

Lasso Regresija

Uporabimo funkcijo `cv.glmnet` iz paketa `glmnet` s parametri `family = "binomial"`, `type.measure = "class"` ter privzeto nastavitev `alpha = 1` (zapis lasso). Ker lasso regresija že sama opravlja izbiro spremenljivk, uporabimo vse neničelne spremenljivke za izračun mere razlikovalne točnosti.

Ridge Regresija RFE

Uporabimo isto funkcijo kot v lasso regresiji, s tem da postavimo `alpha = 0` (zapis ridge.rfe). Podobno kot pri štorih, uporabimo RFE za postopno izločanje spremenljivk s 30 % korakom. Na koncu izberemo tisti model, ki ima najmanjšo `cvm` napako (*angl.* mean cross-validated error). Za izračun mere razlikovalne točnosti uporabimo vse uporabljene spremenljivke za izbran model.

Elastic Net

Uporabimo isto funkcijo kot v lasso regresiji in zgradimo več modelov z različnimi vrednostmi parametra `alpha`, te zavzamejo vrednosti med `0.05` in `0.95` z razmikom `0.05`. Na koncu izberemo tisti model, ki ima najmajšo `cvm` napako (zapis elastic). Ker tudi elastic net že sam opravlja izbiro spremenljivk, za izračun mere razlikovalne točnosti uporabimo vse neničelne spremenljivke izbranega modela.

Bolasso [1]

Zgradimo 100 različnih modelov lasso regresije, kjer je vsak model zgrajen na samovzorčnem vzorcu. Med gradnjo si zabeležimo frekvenco izbranih spremenljivk za vse modele in za izbiro spremenljivk uporabimo tiste spremenljivke, ki imajo frekvenco večjo ali enako 5 (zapis bolasso). Za primerjavo uporabimo tudi vse spremenljivke s frekvenco večjo ali enako ena (zapis bolasso.all).

Boelastic

Zgradimo 100 različnih Elastic Net modelov s fiksno vrednostjo `alpha = 0.05`, vsak model je zgrajen na samovzorčnem vzorcu. Med gradnjo si zabeležimo frekvenco izbranih spremenljivk za vse modele in za izbiro spremenljivk uporabimo tiste spremenljivke, ki imajo frekvenco večjo ali enako 50 (zapis boelastic.freq.50) ter 75 (zapis boelastic.freq.75).

PAM

Uporabimo funkcijo `pamr.cv` iz paketa `pamr` za izgradnjo modela. Izberemo tisti prag (*angl.* threshold), za katerega ima model najmanjšo `cv` napako (*angl.* cross-validation error). Za izbiro spremenljivk uporabimo vse uporabljene spremenljivke pri izbranem pragu (zapis pam).

BoPAM

Zgradimo 100 različnih PAM modelov, vsak model je zgrajen na samovzorčnem vzorcu. Na koncu izberemo tisti model, ki ima najmanjšo `cv` napako. V kolikor imamo več takih modelov, izberemo tistega, ki ima najmanjše število uporabljenih spremenljivk. Za izbiro spremenljivk uporabimo vse uporabljene spremenljivke pri izbranem pragu (zapis bopam).

3 Rezultati

3.1 Prvotne simulacije

V tabeli 2 so prikažani povprečni rezultati prvotnih simulacij vseh uporabljenih metod za vse primere. V prilogi A so priložene tabele z nepovprečenimi rezultati teh metod.

Tabela 2: Povprečni rezultati prvotnih simulacij.

metoda	%corr	%false	TDR	FDR	PA	PA1	PA2	GM	AUC
1 filter.caret	0.42	0.01	0.72	0.28					
2 rf.caret	0.98	0.93	0.10	0.90					
3 rf.var.used	1.00	0.99	0.10	0.90	0.75	0.75	0.75	0.75	0.79
4 rf.var.used.freq	0.34	0.04	0.63	0.37					
5 rfe.scaled	0.35	0.04	0.52	0.48	0.74	0.74	0.74	0.74	0.78
6 rfe.non.scaled	0.35	0.04	0.52	0.48	0.74	0.74	0.74	0.74	0.78
7 stump.mtry.31	0.69	0.23	0.32	0.68	0.74	0.75	0.74	0.73	0.78
8 stump.mtry.1000	0.42	0.09	0.53	0.47	0.68	0.68	0.68	0.67	0.74
9 stump.mtry.min.OOB	0.62	0.16	0.39	0.61	0.74	0.74	0.74	0.73	0.78
10 ridge.rfe	0.44	0.10	0.46	0.54	0.75	0.75	0.75	0.75	0.79
11 lasso	0.19	0.02	0.56	0.44	0.73	0.73	0.73	0.73	0.77
12 elastic	0.47	0.07	0.49	0.51	0.76	0.76	0.76	0.76	0.79
13 bolasso	0.45	0.10	0.49	0.51	0.73	0.73	0.73	0.73	0.77
14 bolasso.all	0.72	0.32	0.32	0.68					
15 boelastic.freq.50	0.59	0.10	0.48	0.52	0.76	0.76	0.76	0.76	0.79
16 boelastic.freq.75	0.42	0.03	0.59	0.41	0.76	0.76	0.76	0.76	0.79
17 pam	0.60	0.17	0.46	0.54	0.76	0.77	0.76	0.76	0.79
18 bopam	0.51	0.24	0.44	0.56	0.74	0.74	0.74	0.74	0.78

Metoda filter (1) je pridobila najvišjo vrednost TDR, podatkov o napovedi pa nimamo, saj metoda ne zgradi modela.

Naključni gozdovi (2, 3) so izbrali vse spremenljivke, zato smo dobili tudi zelo slabe, neuporabne rezultate; ko pa smo izbirali s frekvenco (4) pa smo bistveno izboljšali

model, ta je dosegel drugi najboljši rezultat.

Naključni gozdovi z RFE (5, 6) niso uspeli pridobiti višje ocene; slab rezultat glede na to, koliko modelov smo zgradili za vsak primer.

Štori (7, 9) niso pridobili visoke ocene, ko pa smo nastavili parameter `mtry = 1000` (8), je metoda pridobila višjo oceno. Sicer smo s tem delali le samovzorčenje (*angl.* bagging) in ne več naključnih gozdov.

Ridge regresija z RFE (10) na žalost ni pridobila visoke ocene, čeprav naj bi delovala še posebej dobro za visoko korelirane podatke.

Lasso regresija (11) je pridobila visoko oceno z zelo majhno oceno %false.

Elastic net (12) ima malo slabšo oceno TDR v primerjavi z lasso, vendar pa ima bistveno večjo oceno %corr.

Bolasso (13) na žalost ni uspela bistveno izboljšati TDR v primerjavi z lasso, je pa uspela bistveno izboljšati oceno %corr, ta je zelo podobna oceni elastic; glede na to, koliko modelov smo zgradili za to metodo, pa se metoda ne splača. Če pa smo izbrali vse spremenljivke z bolasso (14), metoda ni več učinkovita.

Boelastic (15, 16) je pridobila malo boljšo oceno v primerjavi z elastic, vendar glede na število zgrajenih modelov, se tudi ta metoda ne splača.

PAM (17) ima še najmanjšo TDR med glavnimi metodami, bopam (18) pa je uspela še malo poslabšati oceno.

3.2 Izbrane simulacije

Med vsemi predstavljenimi metodami smo izbrali metode, ki so delovale najboljše in naredili še bolj podrobne simulacije; izbrali smo: filter.caret, rf.var.used.freq (naključni gozdovi s frekvenčnim izbiranjem spremenljivk), stump.mtry.1000 (štori s parametrom `mtry = 1000`), lasso, elastic (elastic net s parametrom α `= 0.5`) ter pam.

Za te primere smo naredili bolj podrobne simulacije, in sicer: za velikosti vzorcev (`sample`) smo uporabili vrednosti `50, 100, 250` (nespremenjeno), za razliko med razredi (`mde`) smo uporabili vrednosti med `0` in `2` z razmikom `0.25`, za korelacijo med spremenljivkami (`rho`) smo uporabili vrednosti med `0` in `1` z razmikom `0.2`.

Za vse predstavljene metode v nadaljevanju velja nekaj splošnih pravil, in sicer: mera razlikovalne točnosti (*angl.* variable accuracy) ter napovedna točnost (*angl.* prediction accuracy) se zvišujeta z višanjem `mde`; `mde` je največji dejavnik, ki vpliva na učinkovitost modela; korelacijski koeficient ima drugi največji vpliv na delovanje modela, ta lahko izboljšuje, ali poslabša model, odvisno od modela; velikost vzorca tudi vpliva na učinkovitost modela, na nekatere modele več kot na druge.

Za vse metode velja v splošnem, ko `mde = 0` imamo skoraj ničelno mero razlikovalne točnosti ne glede na ostale dejavnike, ta se postopoma veča in pridemo do

visoke točnosti okoli `mde = 1`. Za napovedno točnost velja v splošnem, da najmanjšo vrednost imamo za `mde = 0` in tam je enaka 0.50, ta se postopoma veča in se ustali okoli `mde = 1`. Napovedna točnost je v splošnem podobna za vse metode in tudi vse izračunane mere napovedne točnosti so si zelo podobne, verjetno zaradi uravnoteženih podatkov.

Filter

Filter metoda (slika 1) zelo hitro doseže visoko TDR, obstajajo pa razlike med velikostmi vzorca. Opazimo, da ima korelacija spremenljivk zelo majhen vpliv na mero razlikovalne točnosti. Fiksirana vrednost praga (2/3) povzroči, da imamo skoraj ničelno vrednost p.false za vse primere, kar kaže na to, da je metoda zelo selektivna in izbere večinoma pravilne spremenljivke. Za `mde = 1` imamo skoraj 100 % TDR ter p.corr za večino primerov.

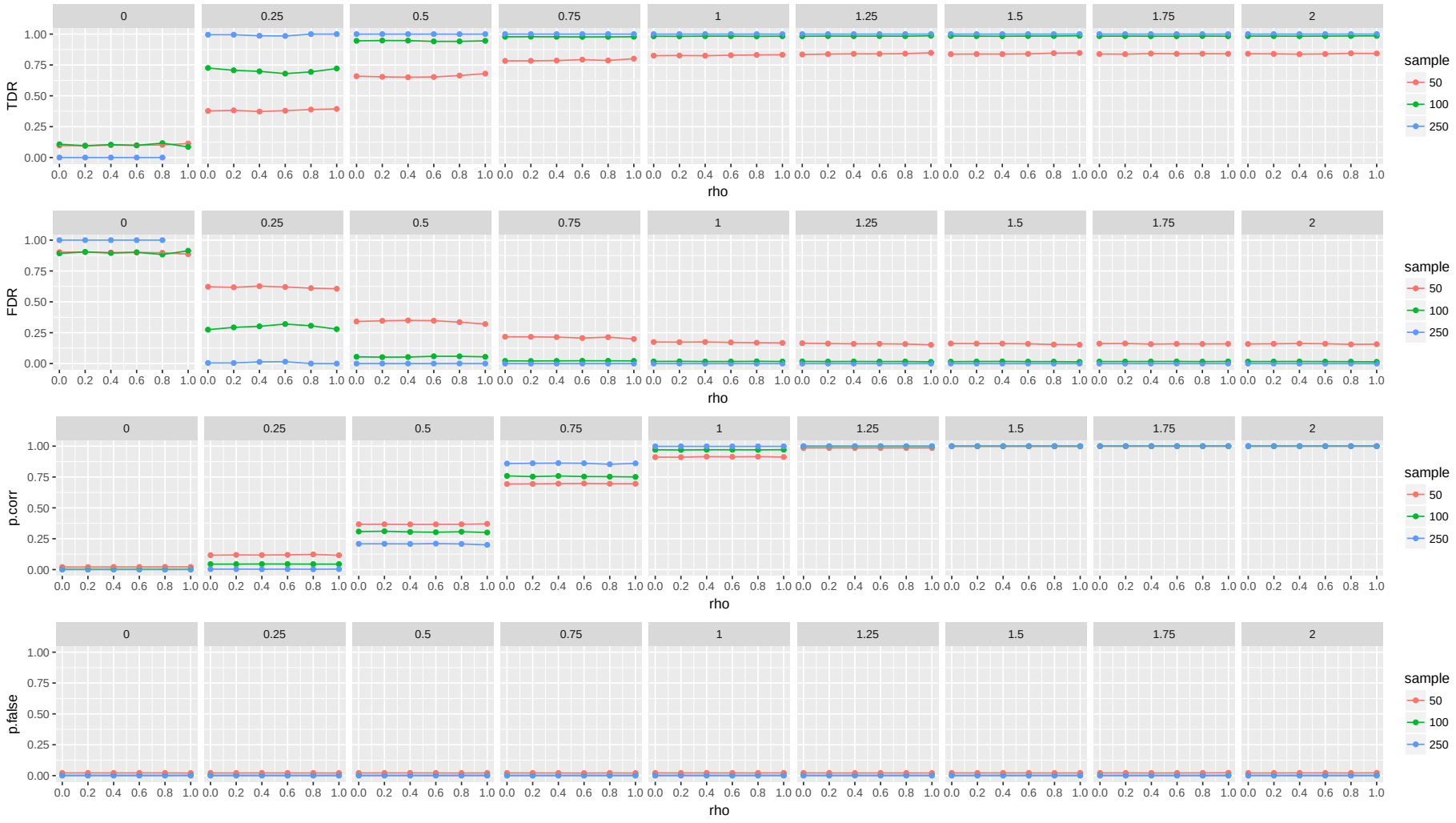
Naključni gozdovi

Naključni gozdovi (slika 2) imajo največjo razliko v razlikovalni točnosti med vsemi modeli glede velikosti vzorca. Čeprav se TDR ter p.false izboljšujeta po splošnem pravilu, p.corr postane in ostane zelo odvisen od velikosti vzorca. Velikost vzorca nam omejuje največjo možno vrednost p.corr.

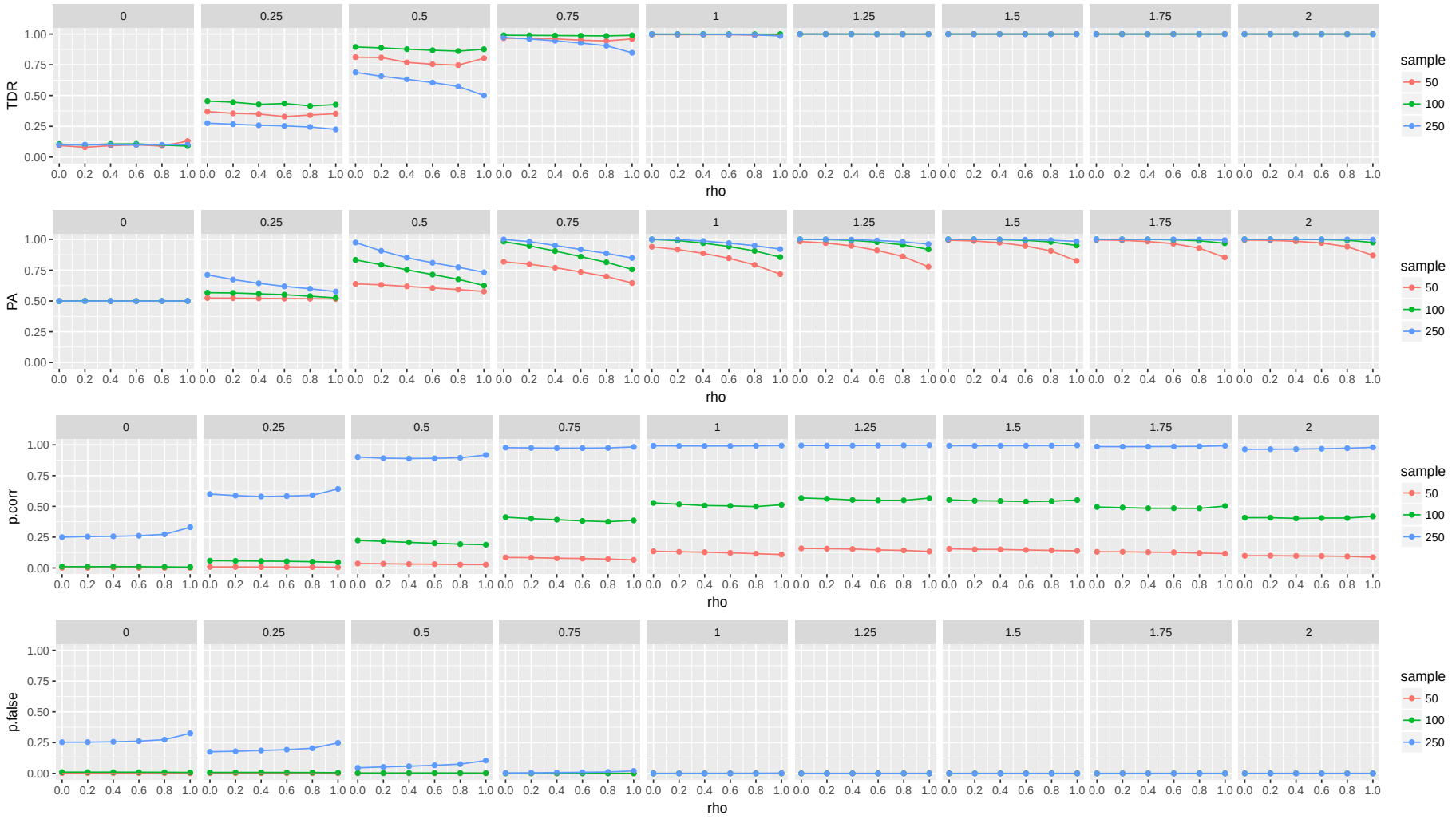
Štori

V primeru, ko izberemo vse spremenljivke uporabljene v vseh štorih (slika 3), imamo slabše stanje v primerjavi z RF glede TDR, imamo pa bolj dosledne ocene p.corr (te so si zelo podobne med sabo in so predvidljive), te se sicer nikoli ne približajo 100 %.

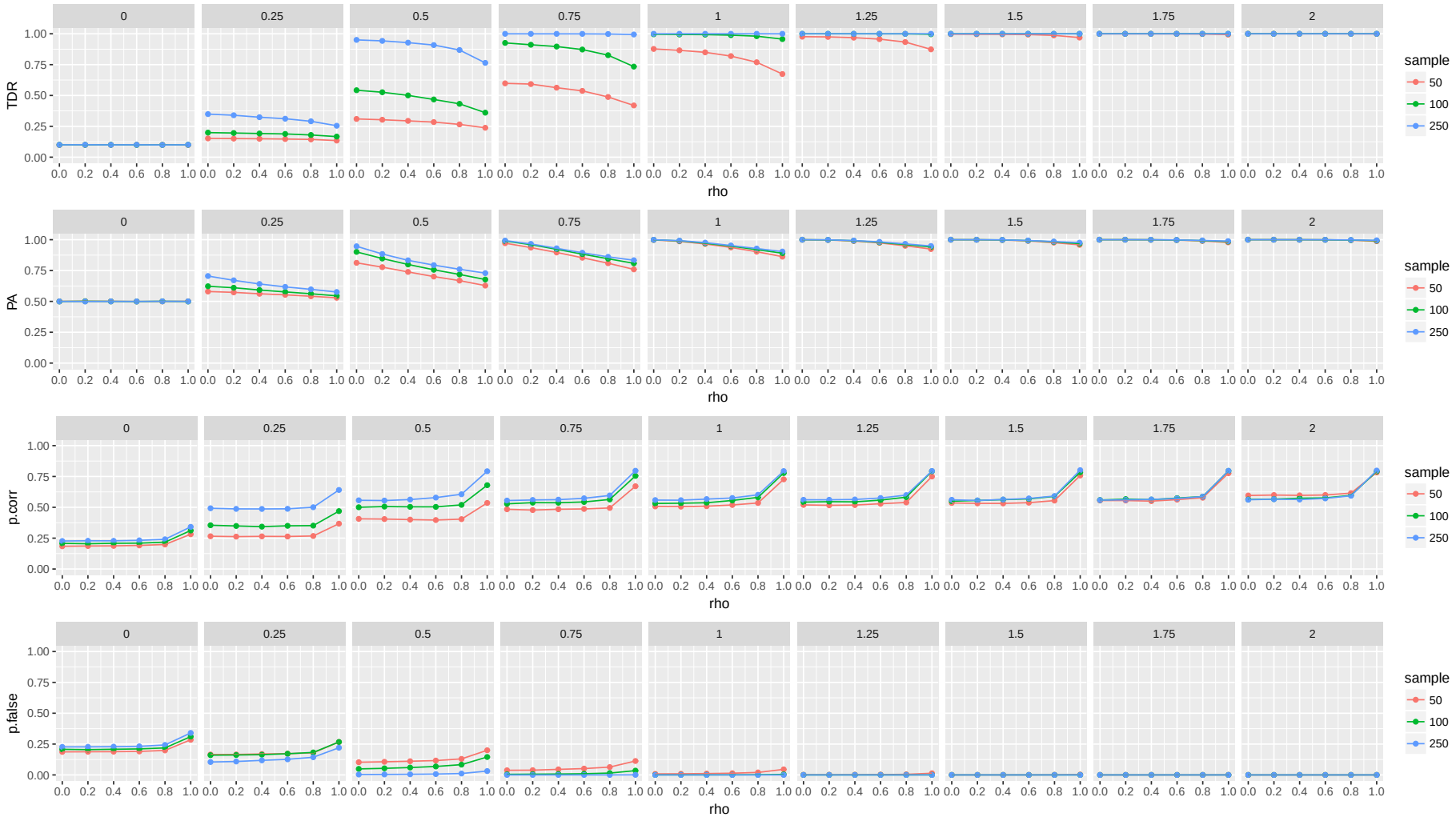
V primeru, ko pa izberemo spremenljivke s frekvenco večjo ali enako 5 (slika 4), se model obnaša bližje RF glede TDR, ima pa zelo podobno obliko ocene p.corr s prejšnjo metodo, ta je sicer še slabša (po pričakovanem).



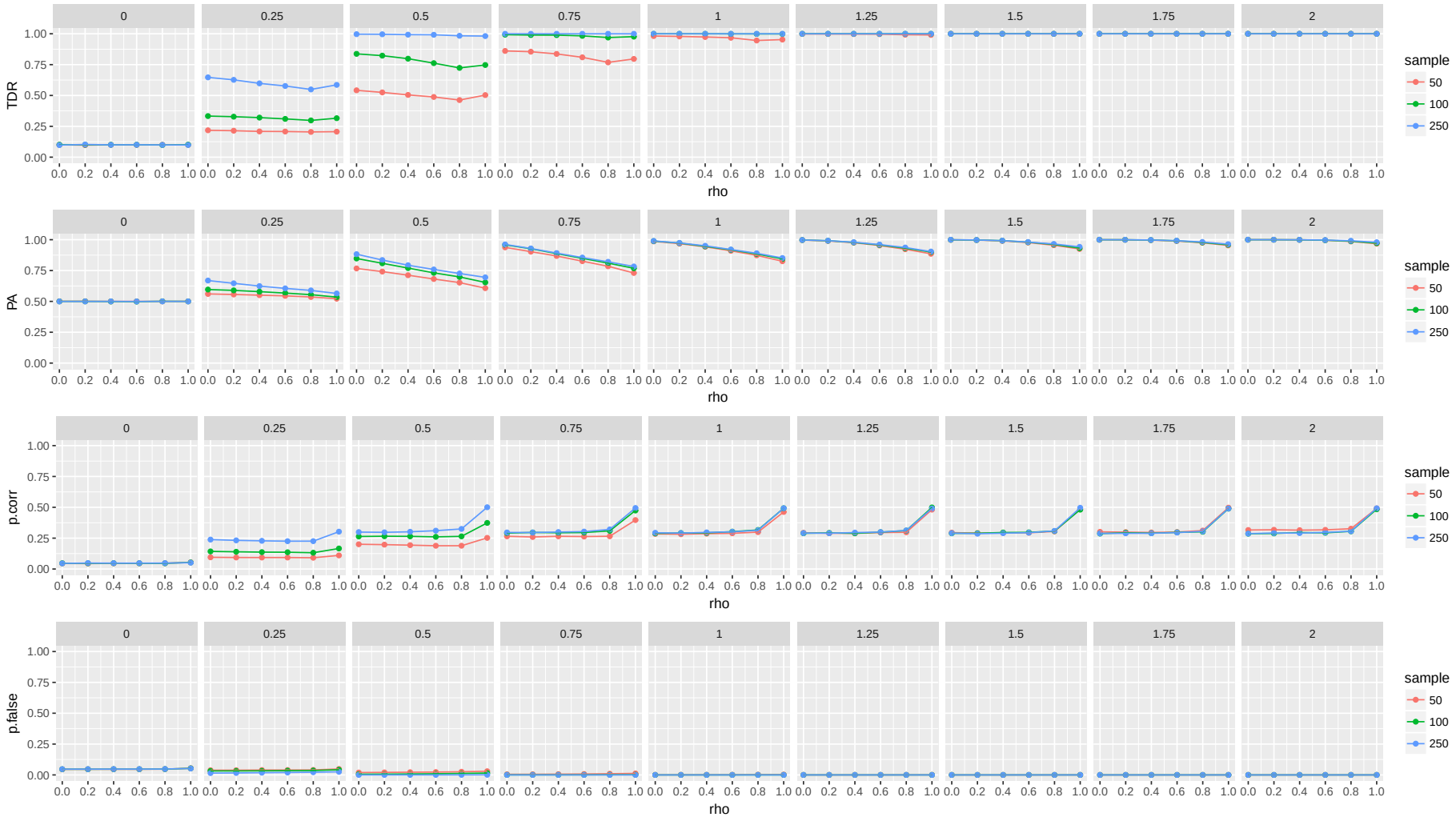
Slika 1: Rezultati filter metode. Prva vrstica slike predstavlja mere TDR, druga vrstica predstavlja mere FDR, tretja vrstica predstavlja mere p.corr, četrta vrstica predstavlja mere p.false. Vsak stolpec slike predstavlja različno vrednost `mde`, ta se giblje med 0 in 2. Na abscisni osi vsake vrstice so predstavljene različne vrednosti `rho`, ta se giblje med 0 in 1. Različne barve predstavljajo rezultate za različne velikosti vzorca.



Slika 2: Rezultati naključnih gozdov. Slika predstavlja podobne mere prejšnji sliki, edina razlika je v drugi vrstici, ta predstavlja mere PA.



Slika 3: Rezultati štorov, v primeru, ko uporabimo vse izbrane spremenljivke za izračun točnosti.



Slika 4: Rezultati štorov, v primeru, ko uporabimo zgolj spremenljivke s frekvenco večjo ali enako 5 za izračun točnosti.

Lasso regresija

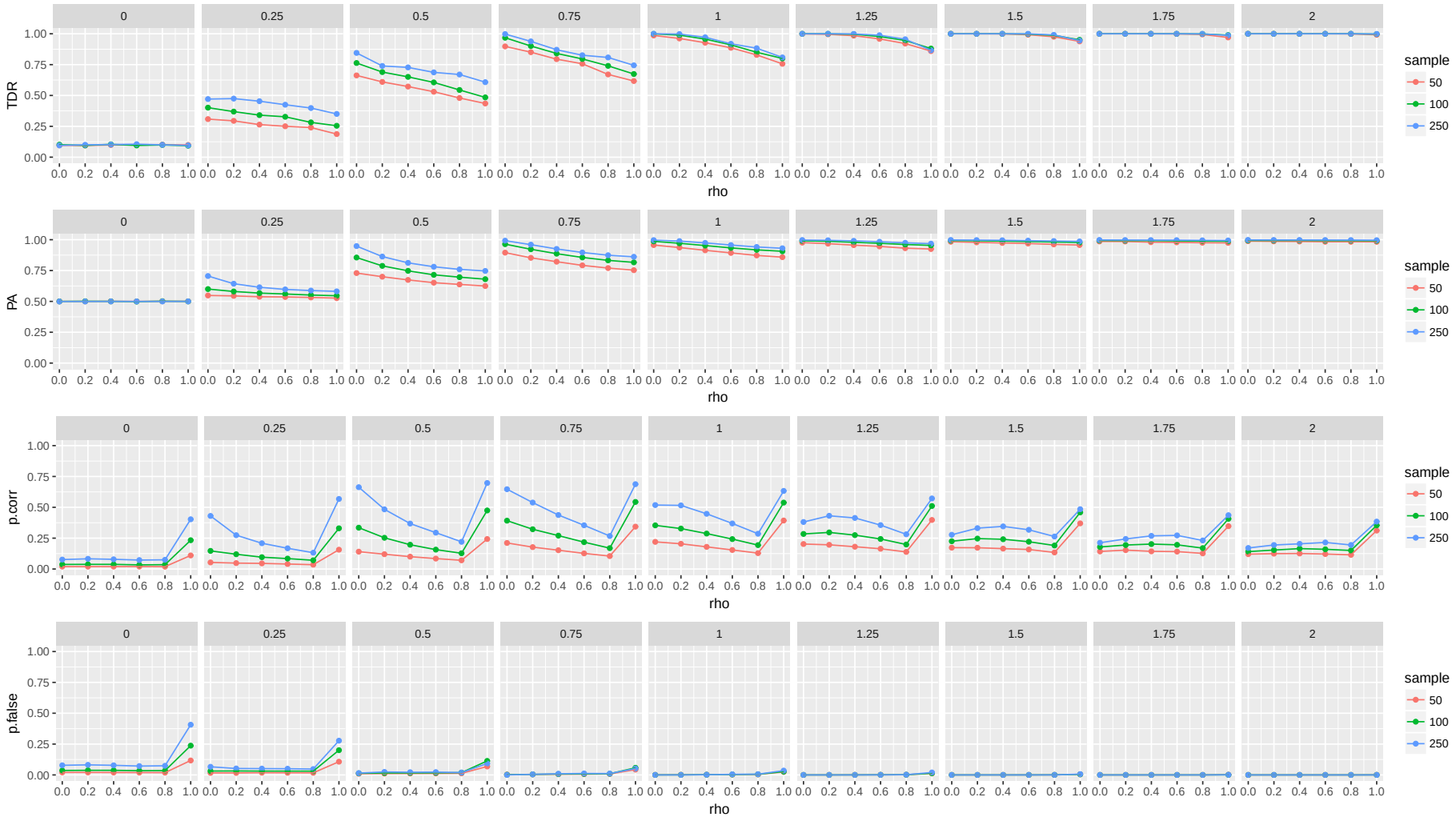
Ker lasso (slika 5) izbere le nekaj spremenljivk med koreliranimi spremenljivkami, se p.corr obnaša po pričakovanem. Nikoli ne dosežemo 100 % vrednosti ne glede na ostale dejavnike in ta je tudi precej odvisna od korelacije. Korelacija ima večji vpliv na TDR v primerjavi s prejšnjimi metodami, velikost vzorca pa ima manjši vpliv na TDR, saj so si te podobne za različne velikosti vzorca. V primeru, ko `rho = 1` (popolna korelacija), lasso izbere le eno spremenljivko izmed množice visoko koreliranih spremenljivk, in zato imamo nenadno povečanje ocene p.corr na sliki.

Elastic Net

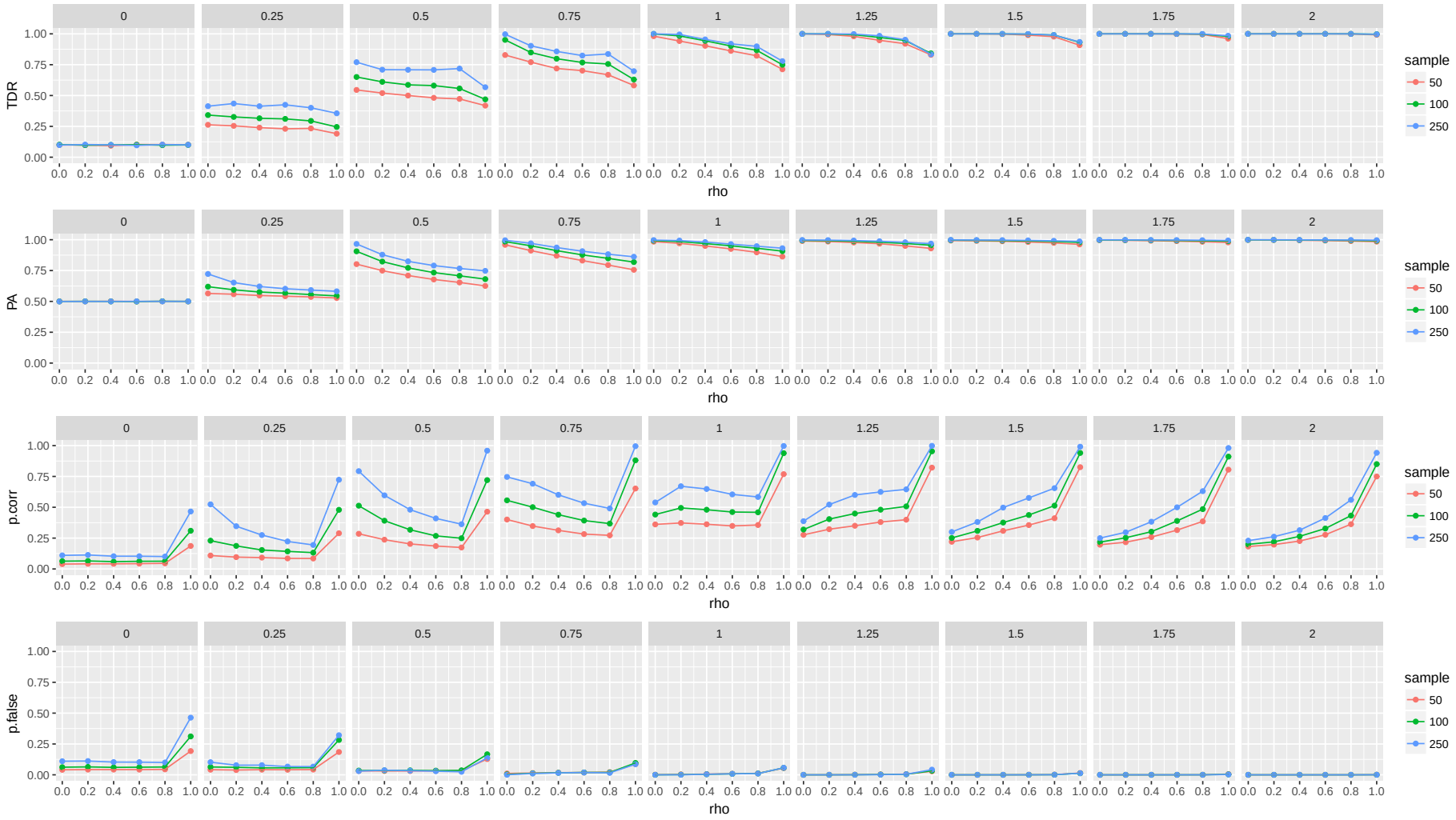
Za elastic net (slika 6) je TDR primerljiva z lasso regresijo, p.corr ocena pa je izboljšana in doseže 100 % vrednost za zelo visoke korelacije. Ta metoda je sicer še bolj odvisna od korelacijskega koeficienta in p.corr je v veliki meri vplivana s strani korelacijskega koeficienta. Dobljeni rezultati so verjetno posledica izbrane fiksne vrednosti `alpha = 0.5`, za druge vrednosti `alpha` bi verjetno dobili drugačne rezultate.

PAM

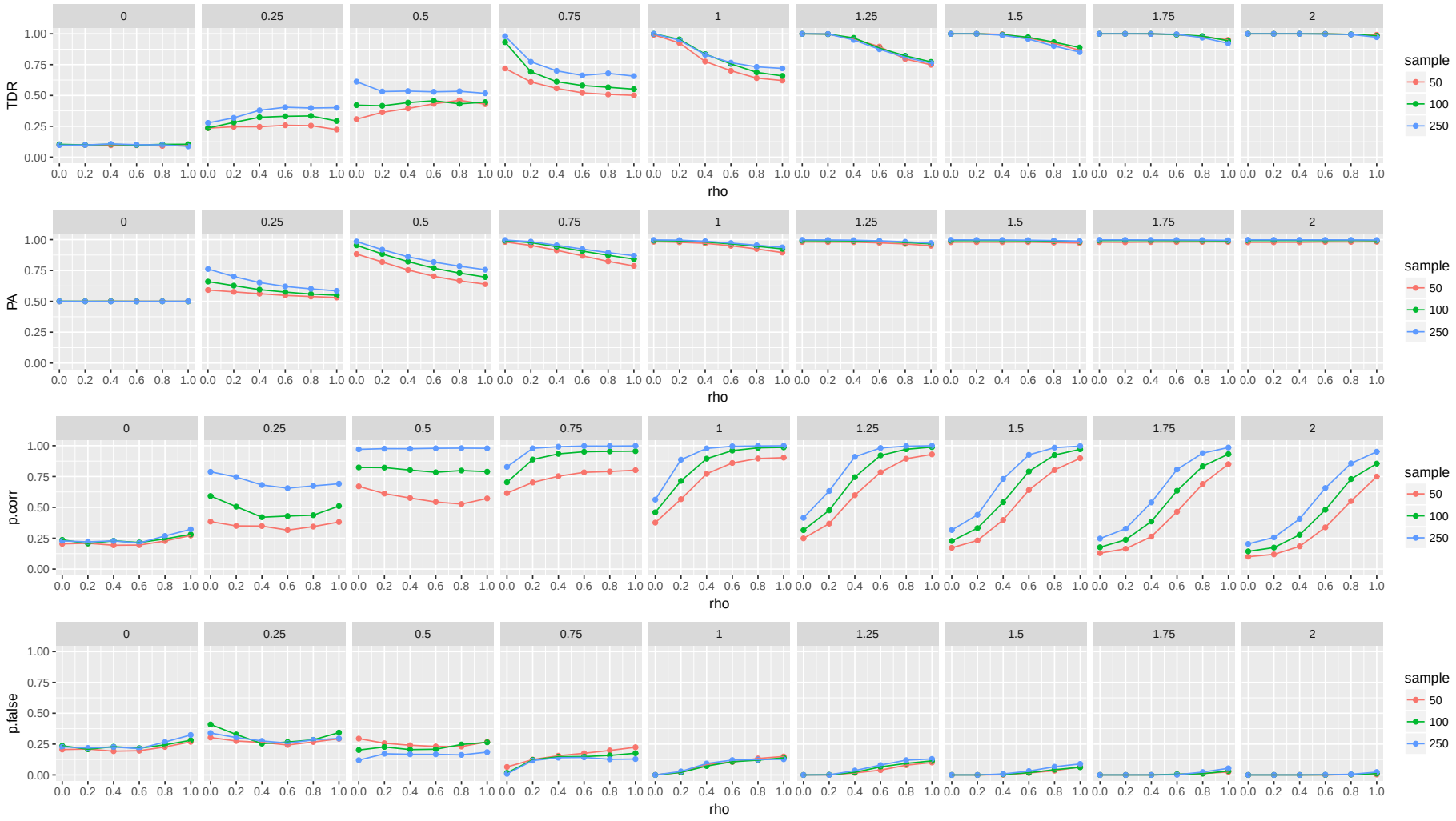
PAM (slika 7) je metoda, na katero korelacija vpliva še najbolj med vsemi metodami. TDR je najslabša izmed vseh modelov in je v veliki meri odvisna od korelacije. Prav tako sta oceni p.corr ter p.false slabši v primerjavi z ostalimi modeli. P.false se počasneje izboljšuje, p.corr pa ima ogromne razlike za različne korelacijske koeficiente. P.corr se izboljšuje z večanjem korelacije, vendar se p.false istočasno veča, kar kaže na to, da bo model v teh primerih izbral veliko spremenljivk.



Slika 5: Rezultati lasso regresije.



Slika 6: Rezultati elastic net.



Slika 7: Rezultati pam.

3.3 Uporaba na pravih podatkih

Uporabili smo prave genske podatke [11], da smo preverili moč uporabljenih modelov (tabela 3). Ker ne vemo, kateri geni so resnično različno izraženi v teh podatkih, ne moremo izračunati mere razlikovalne točnosti (*angl.* variable accuracy), zato smo si le zabeležili, koliko genov so modeli izbrali (r). Metode smo ponovili dva krat, prvič za napoved skupine ER (*angl.* estrogen receptor, ER+ in ER-) ter za skupino grade (*angl.* grade of breast cancer, 1 ali 2 in 3).

Tabela 3: Rezultati modelov na pravih podatkih.

	skupina	metoda	r	PA	PA1	PA2	GM	AUC
1	ER	filter	170					
2	ER	rf.freq	19	0.92	0.95	0.85	0.90	0.94
3	ER	stump	32	0.88	0.92	0.82	0.86	0.92
4	ER	stump.freq	15	0.88	0.92	0.82	0.86	0.92
5	ER	lasso	9	0.89	0.94	0.79	0.85	0.92
6	ER	elastic	18	0.89	0.95	0.79	0.86	0.93
7	ER	pam	6	0.90	0.92	0.85	0.88	0.91
8	grade	filter	73					
9	grade	rf.freq	12	0.75	0.77	0.74	0.75	0.88
10	grade	stump	124	0.68	0.73	0.65	0.67	0.79
11	grade	stump.freq	36	0.67	0.73	0.61	0.65	0.77
12	grade	lasso	14	0.74	0.79	0.68	0.72	0.85
13	grade	elastic	23	0.76	0.81	0.71	0.75	0.87
14	grade	pam	196	0.70	0.72	0.68	0.69	0.79

Napovedovanje receptorja (ER)

Filter metoda je v tem primeru izbrala največ spremenljivk med vsemi metodami, veliko več od drugih. PAM jih je izbrala najmanj, in sicer samo 6, in vseeno uspela pridobiti visoko napovedno točnost $PA = 0.90$. Ostale metode pa so izbrale med 9 in 32 spremenljivk, napovedne točnosti so primerljive. Za vse metode velja, da imamo boljšo oceno $PA1$ v primerjavi s $PA2$, kar je bilo pričakovano, saj imamo več enot v prvem razredu (imamo neuravnotežene podatke), kar pogosto privede do tega, da je klasifikator bolj nagnjen klasificirati nove vzorce v večji razred [2].

Napovedovanje stopnje tumorja (grade)

Filter metoda je zopet izbrala veliko spremenljivk, vendar so jih nekatere metode izbrale še več. PAM, ki je v prejšnjem primeru izbrala najmanj spremenljivk, je v tem primeru izbrala največ spremenljivk, in sicer 196. Metoda, ki je izbrala najmanjše število spremenljivk, je v tem primeru RF, in sicer je izbrala 12 spremenljivk in dosegla $PA = 0.75$, kar je druga najboljša ocena. Tudi v tem primeru je ocena $PA1$ večja od $PA2$, saj smo tudi v tem primeru imeli več enot v prvem razredu.

Delovanje metod za grade je bila precej slabša od ER, s povprečnim $PA = 0.72$ in skupno več izbranimi spremenljivkami.

3.3.1 Uravnoteženje podatkov

V tem razdelku bomo skušali uravnotežiti podatke, in sicer želeli bi dobiti podobne rezultate $PA1$ in $PA2$. Mogoče pa je druga skupina (manjša skupina) tista, ki nas res zanima. Želeli bi, da nam klasifikator rajši klasificira nove vzorce v drugi razred (npr. prisotnost bolezni) kot v prvi razred (npr. odsotnost bolezni). Z drugimi besedami, bi iz zdravstvenih razlogov želeli rajši imeti več lažno pozitivnih (*angl.* False Positive) primerov kot lažno negativnih (*angl.* False Negative) primerov.

V ta namen bomo obdelali podatke, tako da bomo skušali pridobiti uravnotežene podatke, tj. enako število vzorcev v vsakem razredu, in s tem izboljšati oceno drugega razreda ali oceno obeh razredov, če je mogoče.

Za prvi poskus uravnoteženja podatkov bomo normalizirali vrednosti spremenljivk prvega razreda (za vsako enoto v prvem razredu bomo j-temu genu odšteli povprečno vrednost j-tega gena v prvem razredu in delili s standardno variacijo j-tega gena v prvem razredu) in vrnili absolutne vrednosti normiranih spremenljivk. Nato seštejemo vse novo pridobljene vrednosti spremenljivk za vsako enoto posebej (vsota normiranih spremenljivk za enoto predstavlja celotno "razdaljo", ki jo enota pokriva) in uporabimo to mero kot (daljnosežni) vpliv dane enote na klasifikacijo. Večja vsota pomeni večji vpliv in obratno. Na koncu izberemo n enot prvega razreda, ki imajo največje vsote, kjer je n število enot v drugem razredu. Te novo pridobljene skrčene/uravnotežene podatke zdaj uporabimo za gradnjo klasifikatorjev.

Z istim razmislekom ponovimo prejšnji postopek s tem, da izberemo enote s srednjimi ter najmanjšimi vsotami.

Povzetek

Vsi rezultati so povzeti v eni tabeli 4 (brez filter metode), kjer vključimo rezultate neuravnoteženih podatkov (All - vse enote prvega razreda) in uravnotežene podatke (Max - največje vsote, Med - srednje vsote, Min - najmanjše vsote).

Tabela 4: Povprečni rezultati vseh metod na pravih podatkih. $\bar{r}$ je mediana vseh vrednosti r za dano metodo.

	skupina	metoda	$\bar{r}$	PA	PA1	PA2	GM	AUC
1	ER	All	16.5	0.89	0.93	0.82	0.87	0.92
2	ER	Max	27	0.85	0.83	0.86	0.84	0.92
3	ER	Med	27	0.85	0.83	0.86	0.84	0.92
4	ER	Min	19.5	0.92	0.97	0.86	0.91	0.96
5	grade	All	29.5	0.72	0.76	0.68	0.7	0.82
6	grade	Max	24	0.74	0.69	0.79	0.73	0.83
7	grade	Med	24	0.74	0.69	0.79	0.73	0.83
8	grade	Min	23	0.75	0.78	0.72	0.74	0.85

V prilogi B so priloženi rezultati vseh metod za vse tri poskuse uravnoteženja podatkov.

Opazimo, da smo pridobili majhno izboljšavo za nekatere primere z uravnoteženjem; najboljšo oceno smo dobili z Min metodo (tabela 25 v prilogi B). Za Min metodo sta modela, ki izstopata od drugih lasso ter elastic net. V skupini ER lasso doseže točnost $GM = 0.91$ (porast za 6 odstotnih točk) s samo dvema izbranima spremenljivkama. Elastic net pa doseže točnost $GM = 0.95$ (porast za 11 odstotnih točk) s petindvajsetimi izbranimi spremenljivkami. Za skupino grade pa tako lasso, kot elastic dosežeta točnost $GM = 0.79$ (porast za 7 ter 4 odstotnih točk) z enajstimi izbranimi spremenljivkami.

4 Zaključek

Naši rezultati so pokazali, da obstajajo razlike med uporabljenimi metodami. Vsi klasifikatorji so zelo odvisni od povprečne razlike med razredi. Za napovedno moč pa ni bilo prave razlike med klasifikatorji, verjetno zaradi zelo uravnoteženih podatkov.

Za poskus izboljšanja rezultatov smo uporabili večkratno samovzorčenje (*angl.* multiple bagging) za izgradnjo več modelov in izbrali med vsemi najboljšega oz. skupno oceno vseh modelov. Ta pristop se je izkazal za neučinkovitega in ni izboljšal ocene. Drugi pristop je bil z uporabo frekvence spremenljivk, ta pristop je izboljšal model v nekaterih merah in ga poslabšal v drugih. Uporabili smo tudi eno filter metodo, ki se je izkazala za zelo učinkovito v simulacijah, vendar pri tej smo morali predhodno izbrati optimizacijski parameter za izbiro spremenljivk, katero vrednost smo skušali uganiti. Ta metoda je bila neučinkovita na pravih podatkih.

Metode, ki so pridobile najboljše rezultate (za izbiro spremenljivk) na simuliranih ter pravih podatkih so: rf.var.used.freq (naključni gozdovi s frekvenčnim izbiranjem) ter lasso oz. elastic net (ta je malo izboljšala oceno lasso). Če bi moral izbrati le eno metodo, bi izbral elastic net, če imamo majhen vzorec ($N \leq 100$), naključne gozdove bi pa izbral, če imamo velik vzorec ($N \geq 250$).

V tej nalogi nismo skušali ugotavljati učinka v primeru neuravnoteženih podatkov, števila celotnih spremenljivk ali števila resnično različno izraženih spremenljivk, vendar smo se osredotočili na izbrane metode za primere, ko imamo uravnotežene podatke. Klasifikatorje smo izbrali izmed najbolj priljubljenih za visokorazsežne podatke. Možno je, da so ostale metode, ki jih mi nismo obravnavali, bolj učinkovite za izbiro spremenljivk in nanje manj vplivajo naše preizkušene značilnosti/lastnosti.

Na koncu smo še preizkusili izbrane metode na pravih podatkih ter ocenili njihovo delovanje. Metode so delovale odlično za en primer, za drug primer pa smo dobili slabše rezultate. Skušali smo popraviti te ocene z uravnoteženjem podatkov. Uporabili smo več načinov, kako pridobiti uravnotežene podatke, in smo uspeli malo izboljšati mere točnosti za obe skupini.

5 Literatura

[1] Francis R. Bach. Bolasso. *Proceedings of the 25th international conference on Machine learning - ICML '08*, 2008. *(Citirano na strani 11.)*

[2] Rok Blagus and Lara Lusa. Class prediction for high-dimensional class-imbalanced data. *BMC Bioinformatics*, 11(1):523, 2010. *(Citirano na straneh 2 in 23.)*

[3] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996. *(Citirano na strani 5.)*

[4] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. *(Citirano na strani 5.)*

[5] Sandrine Dudoit, Jane Fridlyand, and Terence P Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 97(457):77–87, 2002. *(Citirano na strani 1.)*

[6] A. D. Gordon, L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. Classification and regression trees. *Biometrics*, 40(3):874, 1984. *(Citirano na strani 5.)*

[7] Trevor Hastie, Robert Tibshirani, and Jerome H Friedman. *The elements of statistical learning*. Springer, 1 edition, 2016. *(Citirano na straneh 3 in 6.)*

[8] Max Kuhn. *caret: Classification and Regression Training*. Contributions from Jed Wing and Steve Weston and Andre Williams and Chris Keefer and Allan Engelhardt and Tony Cooper and Zachary Mayer and Brenton Kenkel and the R Core Team and Michael Benesty and Reynald Lescarbeau and Andrew Ziem and Luca Scrucca and Yuan Tang and Can Candan and Tyler Hunt., 2016. R package version 6.0-73. *(Citirano na strani 9.)*

[9] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. ISBN 3-900051-07-0. *(Citirano na strani 7.)*

[10] Y. Saeys, I. Inza, and P. Larranaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 2007. *(Citirano na strani 1.)*

[11] C. Sotiriou, S.-Y. Neo, L. M. McShane, E. L. Korn, P. M. Long, A. Jazaeri, P. Martiat, S. B. Fox, A. L. Harris, and E. T. Liu. Breast cancer classification and prognosis based on gene expression profiles from a population-based study. *Proceedings of the National Academy of Sciences*, 100(18):10393–10398, 2003. *(Citirano na straneh 2, 8 in 23.)*

[12] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences*, 99(10):6567–6572, 2002. *(Citirano na strani 6.)*

[13] Robert Tibshirani. Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(3):273–282, 2011. *(Citirano na strani 4.)*

[14] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005. *(Citirano na strani 4.)*

Priloge

A Rezultati prvotnih simulacij

Tabela 5: Povprečni rezultati prvotnih simulacijah za metodo filter.caret.

	sample	mde	rho	p.corr	p.false	TDR	FDR	PA	PA1	PA2	GM	AUC
1	50.00	0.00	0.00	0.02	0.02	0.10	0.90					
2	50.00	0.00	0.50	0.02	0.02	0.11	0.89					
3	50.00	0.00	0.80	0.02	0.02	0.10	0.90					
4	50.00	0.50	0.00	0.37	0.02	0.66	0.34					
5	50.00	0.50	0.50	0.36	0.02	0.65	0.35					
6	50.00	0.50	0.80	0.37	0.02	0.66	0.34					
7	50.00	1.00	0.00	0.91	0.02	0.82	0.17					
8	50.00	1.00	0.50	0.91	0.02	0.83	0.17					
9	50.00	1.00	0.80	0.91	0.02	0.83	0.17					
10	100.00	0.00	0.00	0.00	0.00	0.12	0.88					
11	100.00	0.00	0.50	0.00	0.00	0.12	0.88					
12	100.00	0.00	0.80	0.00	0.00	0.08	0.92					
13	100.00	0.50	0.00	0.31	0.00	0.95	0.05					
14	100.00	0.50	0.50	0.31	0.00	0.95	0.05					
15	100.00	0.50	0.80	0.31	0.00	0.94	0.06					
16	100.00	1.00	0.00	0.97	0.00	0.98	0.02					
17	100.00	1.00	0.50	0.97	0.00	0.98	0.02					
18	100.00	1.00	0.80	0.97	0.00	0.98	0.02					
19	250.00	0.00	0.00	0.00	0.00	1.00	0.00					
20	250.00	0.00	0.50	0.00	0.00	0.00	1.00					
21	250.00	0.00	0.80	0.00	0.00	0.00	1.00					
22	250.00	0.50	0.00	0.21	0.00	1.00	0.00					
23	250.00	0.50	0.50	0.21	0.00	1.00	0.00					
24	250.00	0.50	0.80	0.21	0.00	1.00	0.00					
25	250.00	1.00	0.00	1.00	0.00	1.00	0.00					
26	250.00	1.00	0.50	1.00	0.00	1.00	0.00					
27	250.00	1.00	0.80	1.00	0.00	1.00	0.00					

Tabela 6: Povprečni rezultati prvotnih simulacijah za metodo rf.caret.

	sample	mde	rho	p.corr	p.false	TDR	FDR	PA	PA1	PA2	GM	AUC
1	50.00	0.00	0.00	0.89	0.89	0.10	0.90					
2	50.00	0.00	0.50	0.89	0.89	0.10	0.90					
3	50.00	0.00	0.80	0.89	0.89	0.10	0.90					
4	50.00	0.50	0.00	0.94	0.86	0.11	0.89					
5	50.00	0.50	0.50	0.94	0.86	0.11	0.89					
6	50.00	0.50	0.80	0.94	0.86	0.11	0.89					
7	50.00	1.00	0.00	0.98	0.73	0.13	0.87					
8	50.00	1.00	0.50	0.98	0.74	0.13	0.87					
9	50.00	1.00	0.80	0.98	0.74	0.13	0.87					
10	100.00	0.00	0.00	0.99	0.99	0.10	0.90					
11	100.00	0.00	0.50	0.99	0.99	0.10	0.90					
12	100.00	0.00	0.80	0.99	0.99	0.10	0.90					
13	100.00	0.50	0.00	1.00	0.98	0.10	0.90					
14	100.00	0.50	0.50	1.00	0.98	0.10	0.90					
15	100.00	0.50	0.80	1.00	0.98	0.10	0.90					
16	100.00	1.00	0.00	1.00	0.90	0.11	0.89					
17	100.00	1.00	0.50	1.00	0.91	0.11	0.89					
18	100.00	1.00	0.80	1.00	0.92	0.11	0.89					
19	250.00	0.00	0.00	1.00	1.00	0.10	0.90					
20	250.00	0.00	0.50	1.00	1.00	0.10	0.90					
21	250.00	0.00	0.80	1.00	1.00	0.10	0.90					
22	250.00	0.50	0.00	1.00	1.00	0.10	0.90					
23	250.00	0.50	0.50	1.00	1.00	0.10	0.90					
24	250.00	0.50	0.80	1.00	1.00	0.10	0.90					
25	250.00	1.00	0.00	1.00	0.99	0.10	0.90					
26	250.00	1.00	0.50	1.00	1.00	0.10	0.90					
27	250.00	1.00	0.80	1.00	1.00	0.10	0.90					

Tabela 7: Povprečni rezultati prvotnih simulacijah za metodo rf.var.used.

	sample	mde	rho	p.corr	p.false	TDR	FDR	PA	PA1	PA2	GM	AUC
1	50.00	0.00	0.00	0.98	0.98	0.10	0.90	0.50	0.50	0.50	0.50	0.50
2	50.00	0.00	0.50	0.98	0.98	0.10	0.90	0.50	0.50	0.50	0.50	0.50
3	50.00	0.00	0.80	0.98	0.98	0.10	0.90	0.50	0.50	0.50	0.50	0.50
4	50.00	0.50	0.00	0.99	0.97	0.10	0.90	0.82	0.82	0.82	0.82	0.90
5	50.00	0.50	0.50	0.99	0.97	0.10	0.90	0.72	0.72	0.72	0.72	0.80
6	50.00	0.50	0.80	0.99	0.97	0.10	0.90	0.67	0.66	0.67	0.66	0.73
7	50.00	1.00	0.00	1.00	0.92	0.11	0.89	1.00	1.00	1.00	1.00	1.00
8	50.00	1.00	0.50	1.00	0.93	0.11	0.89	0.95	0.96	0.95	0.95	0.99
9	50.00	1.00	0.80	1.00	0.93	0.11	0.89	0.90	0.91	0.90	0.90	0.97
10	100.00	0.00	0.00	1.00	1.00	0.10	0.90	0.50	0.50	0.50	0.50	0.50
11	100.00	0.00	0.50	1.00	1.00	0.10	0.90	0.50	0.50	0.50	0.50	0.50
12	100.00	0.00	0.80	1.00	1.00	0.10	0.90	0.50	0.50	0.50	0.50	0.50
13	100.00	0.50	0.00	1.00	1.00	0.10	0.90	0.90	0.90	0.90	0.90	0.97
14	100.00	0.50	0.50	1.00	1.00	0.10	0.90	0.78	0.78	0.78	0.78	0.86
15	100.00	0.50	0.80	1.00	1.00	0.10	0.90	0.72	0.72	0.71	0.71	0.79
16	100.00	1.00	0.00	1.00	0.99	0.10	0.90	1.00	1.00	1.00	1.00	1.00
17	100.00	1.00	0.50	1.00	0.99	0.10	0.90	0.97	0.97	0.97	0.97	1.00
18	100.00	1.00	0.80	1.00	0.99	0.10	0.90	0.93	0.93	0.93	0.93	0.98
19	250.00	0.00	0.00	1.00	1.00	0.10	0.90	0.50	0.50	0.50	0.50	0.50
20	250.00	0.00	0.50	1.00	1.00	0.10	0.90	0.50	0.50	0.50	0.50	0.50
21	250.00	0.00	0.80	1.00	1.00	0.10	0.90	0.50	0.50	0.50	0.50	0.50
22	250.00	0.50	0.00	1.00	1.00	0.10	0.90	0.96	0.96	0.96	0.96	0.99
23	250.00	0.50	0.50	1.00	1.00	0.10	0.90	0.82	0.82	0.82	0.82	0.90
24	250.00	0.50	0.80	1.00	1.00	0.10	0.90	0.76	0.76	0.76	0.76	0.84
25	250.00	1.00	0.00	1.00	1.00	0.10	0.90	1.00	1.00	1.00	1.00	1.00
26	250.00	1.00	0.50	1.00	1.00	0.10	0.90	0.98	0.98	0.98	0.98	1.00
27	250.00	1.00	0.80	1.00	1.00	0.10	0.90	0.94	0.94	0.94	0.94	0.99

Tabela 8: Povprečni rezultati prvotnih simulacijah za metodo rf.var.used.freq.

	sample	mde	rho	p.corr	p.false	TDR	FDR	PA	PA1	PA2	GM	AUC
1	50.00	0.00	0.00	0.00	0.00	0.10	0.90					
2	50.00	0.00	0.50	0.00	0.00	0.10	0.90					
3	50.00	0.00	0.80	0.00	0.00	0.09	0.91					
4	50.00	0.50	0.00	0.04	0.00	0.82	0.18					
5	50.00	0.50	0.50	0.03	0.00	0.75	0.25					
6	50.00	0.50	0.80	0.03	0.00	0.75	0.25					
7	50.00	1.00	0.00	0.14	0.00	1.00	0.00					
8	50.00	1.00	0.50	0.12	0.00	0.99	0.00					
9	50.00	1.00	0.80	0.12	0.00	0.99	0.01					
10	100.00	0.00	0.00	0.01	0.01	0.11	0.89					
11	100.00	0.00	0.50	0.01	0.01	0.10	0.90					
12	100.00	0.00	0.80	0.01	0.01	0.09	0.91					
13	100.00	0.50	0.00	0.22	0.00	0.89	0.11					
14	100.00	0.50	0.50	0.21	0.00	0.87	0.13					
15	100.00	0.50	0.80	0.20	0.00	0.87	0.13					
16	100.00	1.00	0.00	0.53	0.00	1.00	0.00					
17	100.00	1.00	0.50	0.51	0.00	1.00	0.00					
18	100.00	1.00	0.80	0.50	0.00	1.00	0.00					
19	250.00	0.00	0.00	0.25	0.25	0.10	0.90					
20	250.00	0.00	0.50	0.26	0.26	0.10	0.90					
21	250.00	0.00	0.80	0.28	0.27	0.10	0.90					
22	250.00	0.50	0.00	0.90	0.05	0.69	0.31					
23	250.00	0.50	0.50	0.89	0.06	0.62	0.38					
24	250.00	0.50	0.80	0.90	0.07	0.58	0.42					
25	250.00	1.00	0.00	0.99	0.00	1.00	0.00					
26	250.00	1.00	0.50	0.99	0.00	1.00	0.00					
27	250.00	1.00	0.80	0.99	0.00	0.99	0.01					

Tabela 9: Povprečni rezultati prvotnih simulacijah za metodo rfe.scaled.

	sample	mde	rho	p.corr	p.false	TDR	FDR	PA	PA1	PA2	GM	AUC
1	50.00	0.00	0.00	0.05	0.05	0.10	0.90	0.50	0.50	0.50	0.49	0.50
2	50.00	0.00	0.50	0.05	0.05	0.11	0.89	0.50	0.50	0.50	0.50	0.50
3	50.00	0.00	0.80	0.04	0.04	0.10	0.90	0.50	0.50	0.50	0.49	0.50
4	50.00	0.50	0.00	0.22	0.03	0.51	0.49	0.76	0.76	0.76	0.76	0.85
5	50.00	0.50	0.50	0.22	0.04	0.46	0.54	0.70	0.70	0.69	0.69	0.77
6	50.00	0.50	0.80	0.21	0.03	0.46	0.54	0.65	0.65	0.65	0.64	0.71
7	50.00	1.00	0.00	0.15	0.00	0.99	0.01	0.94	0.94	0.94	0.94	0.98
8	50.00	1.00	0.50	0.30	0.01	0.93	0.07	0.91	0.92	0.91	0.91	0.97
9	50.00	1.00	0.80	0.44	0.02	0.85	0.15	0.88	0.88	0.88	0.88	0.95
10	100.00	0.00	0.00	0.06	0.06	0.10	0.90	0.50	0.50	0.50	0.50	0.50
11	100.00	0.00	0.50	0.06	0.06	0.10	0.90	0.50	0.50	0.50	0.50	0.50
12	100.00	0.00	0.80	0.06	0.06	0.10	0.90	0.50	0.50	0.50	0.49	0.50
13	100.00	0.50	0.00	0.47	0.05	0.58	0.42	0.90	0.89	0.90	0.90	0.96
14	100.00	0.50	0.50	0.44	0.04	0.58	0.41	0.77	0.77	0.77	0.77	0.85
15	100.00	0.50	0.80	0.44	0.05	0.56	0.44	0.71	0.71	0.71	0.71	0.79
16	100.00	1.00	0.00	0.23	0.00	1.00	0.00	0.97	0.97	0.97	0.97	0.99
17	100.00	1.00	0.50	0.56	0.02	0.90	0.10	0.95	0.95	0.95	0.95	0.99
18	100.00	1.00	0.80	0.70	0.04	0.83	0.17	0.92	0.92	0.92	0.92	0.98
19	250.00	0.00	0.00	0.07	0.07	0.10	0.90	0.50	0.50	0.50	0.49	0.50
20	250.00	0.00	0.50	0.06	0.06	0.10	0.90	0.50	0.50	0.50	0.49	0.50
21	250.00	0.00	0.80	0.07	0.06	0.10	0.90	0.50	0.50	0.50	0.49	0.50
22	250.00	0.50	0.00	0.81	0.06	0.69	0.31	0.97	0.97	0.97	0.97	1.00
23	250.00	0.50	0.50	0.76	0.06	0.67	0.33	0.82	0.82	0.82	0.82	0.90
24	250.00	0.50	0.80	0.78	0.07	0.64	0.36	0.77	0.77	0.77	0.77	0.85
25	250.00	1.00	0.00	0.37	0.00	1.00	0.00	0.99	0.99	0.99	0.99	1.00
26	250.00	1.00	0.50	0.82	0.07	0.79	0.21	0.97	0.97	0.97	0.97	1.00
27	250.00	1.00	0.80	0.90	0.08	0.74	0.26	0.94	0.94	0.94	0.94	0.99

Tabela 10: Povprečni rezultati prvotnih simulacijah za metodo rfe.non.scaled.

	sample	mde	rho	p.corr	p.false	TDR	FDR	PA	PA1	PA2	GM	AUC
1	50.00	0.00	0.00	0.05	0.05	0.10	0.90	0.50	0.50	0.50	0.49	0.50
2	50.00	0.00	0.50	0.05	0.05	0.10	0.90	0.50	0.50	0.50	0.50	0.50
3	50.00	0.00	0.80	0.04	0.04	0.10	0.90	0.50	0.50	0.50	0.49	0.50
4	50.00	0.50	0.00	0.22	0.03	0.51	0.49	0.76	0.76	0.76	0.76	0.84
5	50.00	0.50	0.50	0.21	0.03	0.47	0.53	0.69	0.69	0.69	0.69	0.77
6	50.00	0.50	0.80	0.21	0.03	0.46	0.54	0.65	0.64	0.65	0.64	0.70
7	50.00	1.00	0.00	0.15	0.00	0.99	0.01	0.94	0.94	0.94	0.94	0.98
8	50.00	1.00	0.50	0.32	0.01	0.92	0.08	0.92	0.92	0.91	0.92	0.97
9	50.00	1.00	0.80	0.46	0.02	0.85	0.15	0.88	0.88	0.88	0.88	0.95
10	100.00	0.00	0.00	0.06	0.06	0.10	0.90	0.50	0.50	0.50	0.49	0.50
11	100.00	0.00	0.50	0.06	0.06	0.10	0.90	0.50	0.50	0.50	0.50	0.50
12	100.00	0.00	0.80	0.06	0.06	0.10	0.90	0.50	0.50	0.50	0.49	0.50
13	100.00	0.50	0.00	0.46	0.04	0.59	0.41	0.89	0.89	0.90	0.89	0.96
14	100.00	0.50	0.50	0.43	0.04	0.59	0.40	0.77	0.77	0.77	0.77	0.85
15	100.00	0.50	0.80	0.43	0.05	0.57	0.43	0.71	0.71	0.71	0.71	0.78
16	100.00	1.00	0.00	0.24	0.00	1.00	0.00	0.97	0.97	0.97	0.97	1.00
17	100.00	1.00	0.50	0.59	0.03	0.88	0.12	0.95	0.95	0.95	0.95	0.99
18	100.00	1.00	0.80	0.73	0.04	0.83	0.17	0.92	0.92	0.92	0.92	0.98
19	250.00	0.00	0.00	0.06	0.06	0.10	0.90	0.50	0.50	0.50	0.49	0.50
20	250.00	0.00	0.50	0.06	0.06	0.10	0.90	0.50	0.50	0.50	0.49	0.50
21	250.00	0.00	0.80	0.06	0.06	0.10	0.90	0.50	0.50	0.50	0.49	0.50
22	250.00	0.50	0.00	0.81	0.05	0.70	0.30	0.97	0.97	0.97	0.97	1.00
23	250.00	0.50	0.50	0.76	0.06	0.68	0.32	0.82	0.82	0.82	0.82	0.90
24	250.00	0.50	0.80	0.78	0.07	0.64	0.35	0.77	0.77	0.77	0.77	0.85
25	250.00	1.00	0.00	0.37	0.00	1.00	0.00	0.99	0.99	0.99	0.99	1.00
26	250.00	1.00	0.50	0.84	0.08	0.77	0.23	0.97	0.97	0.97	0.97	1.00
27	250.00	1.00	0.80	0.93	0.09	0.69	0.31	0.94	0.94	0.94	0.94	0.99

Tabela 11: Povprečni rezultati prvotnih simulacijah za metodo stump.mtry.31.

	sample	mde	rho	p.corr	p.false	TDR	FDR	PA	PA1	PA2	GM	AUC
1	50.00	0.00	0.00	0.35	0.35	0.10	0.90	0.50	0.50	0.50	0.48	0.50
2	50.00	0.00	0.50	0.35	0.35	0.10	0.90	0.50	0.50	0.50	0.48	0.50
3	50.00	0.00	0.80	0.35	0.35	0.10	0.90	0.50	0.51	0.49	0.48	0.50
4	50.00	0.50	0.00	0.60	0.29	0.19	0.81	0.79	0.80	0.79	0.79	0.89
5	50.00	0.50	0.50	0.60	0.29	0.19	0.81	0.71	0.71	0.71	0.71	0.79
6	50.00	0.50	0.80	0.60	0.29	0.19	0.81	0.66	0.66	0.66	0.65	0.73
7	50.00	1.00	0.00	0.86	0.15	0.40	0.60	1.00	1.00	1.00	1.00	1.00
8	50.00	1.00	0.50	0.87	0.15	0.40	0.60	0.95	0.95	0.95	0.95	0.99
9	50.00	1.00	0.80	0.86	0.15	0.40	0.60	0.90	0.90	0.90	0.90	0.97
10	100.00	0.00	0.00	0.38	0.38	0.10	0.90	0.50	0.50	0.50	0.47	0.50
11	100.00	0.00	0.50	0.38	0.38	0.10	0.90	0.50	0.50	0.50	0.47	0.50
12	100.00	0.00	0.80	0.38	0.38	0.10	0.90	0.50	0.50	0.50	0.47	0.50
13	100.00	0.50	0.00	0.77	0.25	0.25	0.75	0.88	0.87	0.88	0.87	0.96
14	100.00	0.50	0.50	0.77	0.26	0.25	0.75	0.77	0.77	0.76	0.76	0.86
15	100.00	0.50	0.80	0.77	0.26	0.25	0.75	0.71	0.71	0.70	0.70	0.79
16	100.00	1.00	0.00	0.95	0.07	0.60	0.40	1.00	1.00	1.00	1.00	1.00
17	100.00	1.00	0.50	0.96	0.07	0.60	0.40	0.96	0.96	0.96	0.96	0.99
18	100.00	1.00	0.80	0.96	0.07	0.59	0.41	0.92	0.92	0.92	0.92	0.98
19	250.00	0.00	0.00	0.41	0.41	0.10	0.90	0.50	0.49	0.51	0.45	0.50
20	250.00	0.00	0.50	0.41	0.41	0.10	0.90	0.50	0.50	0.50	0.45	0.50
21	250.00	0.00	0.80	0.41	0.41	0.10	0.90	0.50	0.50	0.50	0.45	0.50
22	250.00	0.50	0.00	0.93	0.14	0.42	0.58	0.94	0.94	0.94	0.93	0.99
23	250.00	0.50	0.50	0.92	0.14	0.42	0.58	0.80	0.80	0.80	0.80	0.89
24	250.00	0.50	0.80	0.93	0.14	0.43	0.57	0.74	0.75	0.74	0.74	0.83
25	250.00	1.00	0.00	0.99	0.03	0.77	0.23	1.00	1.00	1.00	1.00	1.00
26	250.00	1.00	0.50	0.99	0.03	0.77	0.23	0.97	0.97	0.97	0.97	1.00
27	250.00	1.00	0.80	0.99	0.03	0.77	0.23	0.93	0.93	0.93	0.93	0.98

Tabela 12: Povprečni rezultati prvotnih simulacijah za metodo stump.mtry.1000.

	sample	mde	rho	p.corr	p.false	TDR	FDR	PA	PA1	PA2	GM	AUC
1	50.00	0.00	0.00	0.18	0.19	0.10	0.90	0.50	0.50	0.50	0.49	0.50
2	50.00	0.00	0.50	0.19	0.19	0.10	0.90	0.50	0.50	0.50	0.49	0.50
3	50.00	0.00	0.80	0.20	0.20	0.10	0.90	0.50	0.50	0.50	0.49	0.50
4	50.00	0.50	0.00	0.41	0.10	0.31	0.69	0.69	0.69	0.69	0.69	0.78
5	50.00	0.50	0.50	0.40	0.11	0.29	0.71	0.66	0.66	0.66	0.65	0.73
6	50.00	0.50	0.80	0.40	0.13	0.26	0.73	0.63	0.63	0.63	0.62	0.69
7	50.00	1.00	0.00	0.50	0.01	0.88	0.12	0.88	0.88	0.88	0.88	0.96
8	50.00	1.00	0.50	0.51	0.01	0.83	0.17	0.84	0.85	0.84	0.84	0.93
9	50.00	1.00	0.80	0.53	0.02	0.77	0.23	0.80	0.80	0.80	0.80	0.90
10	100.00	0.00	0.00	0.20	0.21	0.10	0.90	0.50	0.50	0.50	0.48	0.50
11	100.00	0.00	0.50	0.21	0.21	0.10	0.90	0.50	0.50	0.50	0.49	0.50
12	100.00	0.00	0.80	0.22	0.22	0.10	0.90	0.50	0.50	0.50	0.48	0.50
13	100.00	0.50	0.00	0.51	0.05	0.54	0.46	0.73	0.73	0.73	0.73	0.83
14	100.00	0.50	0.50	0.50	0.06	0.49	0.50	0.69	0.70	0.69	0.69	0.78
15	100.00	0.50	0.80	0.52	0.08	0.43	0.56	0.66	0.67	0.65	0.66	0.74
16	100.00	1.00	0.00	0.54	0.00	0.99	0.00	0.89	0.89	0.89	0.89	0.96
17	100.00	1.00	0.50	0.54	0.00	0.99	0.01	0.85	0.85	0.85	0.85	0.94
18	100.00	1.00	0.80	0.57	0.00	0.98	0.02	0.81	0.81	0.81	0.81	0.91
19	250.00	0.00	0.00	0.23	0.23	0.10	0.90	0.50	0.50	0.50	0.47	0.50
20	250.00	0.00	0.50	0.23	0.23	0.10	0.90	0.50	0.50	0.50	0.47	0.50
21	250.00	0.00	0.80	0.24	0.24	0.10	0.90	0.50	0.50	0.50	0.47	0.50
22	250.00	0.50	0.00	0.56	0.00	0.95	0.05	0.74	0.74	0.74	0.73	0.84
23	250.00	0.50	0.50	0.57	0.01	0.92	0.08	0.70	0.70	0.70	0.70	0.79
24	250.00	0.50	0.80	0.61	0.01	0.88	0.12	0.67	0.68	0.67	0.67	0.76
25	250.00	1.00	0.00	0.56	0.00	1.00	0.00	0.89	0.89	0.89	0.89	0.96
26	250.00	1.00	0.50	0.57	0.00	1.00	0.00	0.85	0.85	0.85	0.85	0.94
27	250.00	1.00	0.80	0.60	0.00	1.00	0.00	0.82	0.82	0.82	0.82	0.91

Tabela 13: Povprečni rezultati prvotnih simulacijah za metodo stump.mtry.min.OOB.

	sample	mde	rho	p.corr	p.false	TDR	FDR	PA	PA1	PA2	GM	AUC
1	50.00	0.00	0.00	0.24	0.24	0.10	0.90	0.50	0.50	0.50	0.49	0.50
2	50.00	0.00	0.50	0.24	0.24	0.10	0.90	0.50	0.51	0.49	0.49	0.50
3	50.00	0.00	0.80	0.25	0.25	0.10	0.90	0.50	0.50	0.50	0.49	0.50
4	50.00	0.50	0.00	0.52	0.19	0.24	0.76	0.75	0.76	0.75	0.75	0.84
5	50.00	0.50	0.50	0.50	0.19	0.24	0.76	0.68	0.68	0.69	0.68	0.76
6	50.00	0.50	0.80	0.50	0.19	0.23	0.76	0.64	0.64	0.64	0.64	0.71
7	50.00	1.00	0.00	0.75	0.06	0.61	0.39	0.99	0.99	0.98	0.99	1.00
8	50.00	1.00	0.50	0.80	0.09	0.52	0.48	0.94	0.94	0.94	0.94	0.99
9	50.00	1.00	0.80	0.80	0.10	0.50	0.50	0.88	0.88	0.88	0.88	0.95
10	100.00	0.00	0.00	0.27	0.27	0.10	0.90	0.50	0.50	0.50	0.48	0.50
11	100.00	0.00	0.50	0.27	0.27	0.10	0.90	0.50	0.50	0.50	0.48	0.50
12	100.00	0.00	0.80	0.28	0.28	0.10	0.90	0.50	0.50	0.50	0.48	0.50
13	100.00	0.50	0.00	0.72	0.19	0.31	0.69	0.86	0.86	0.86	0.86	0.94
14	100.00	0.50	0.50	0.71	0.18	0.32	0.68	0.75	0.76	0.75	0.75	0.84
15	100.00	0.50	0.80	0.69	0.17	0.32	0.68	0.69	0.70	0.68	0.68	0.77
16	100.00	1.00	0.00	0.84	0.01	0.89	0.11	0.99	0.99	0.99	0.99	1.00
17	100.00	1.00	0.50	0.92	0.04	0.73	0.27	0.96	0.96	0.96	0.96	0.99
18	100.00	1.00	0.80	0.94	0.05	0.68	0.32	0.91	0.91	0.91	0.91	0.97
19	250.00	0.00	0.00	0.29	0.29	0.10	0.90	0.50	0.50	0.50	0.46	0.50
20	250.00	0.00	0.50	0.30	0.30	0.10	0.90	0.50	0.50	0.50	0.46	0.50
21	250.00	0.00	0.80	0.31	0.31	0.10	0.90	0.50	0.50	0.50	0.46	0.50
22	250.00	0.50	0.00	0.90	0.11	0.50	0.50	0.93	0.93	0.93	0.93	0.98
23	250.00	0.50	0.50	0.90	0.11	0.50	0.50	0.80	0.80	0.79	0.79	0.89
24	250.00	0.50	0.80	0.91	0.11	0.50	0.50	0.74	0.74	0.74	0.73	0.83
25	250.00	1.00	0.00	0.92	0.01	0.95	0.05	1.00	1.00	1.00	1.00	1.00
26	250.00	1.00	0.50	0.97	0.02	0.84	0.16	0.96	0.96	0.96	0.96	0.99
27	250.00	1.00	0.80	0.98	0.02	0.82	0.18	0.92	0.92	0.92	0.92	0.98

Tabela 14: Povprečni rezultati prvotnih simulacijah za metodo ridge.rfe.

	sample	mde	rho	p.corr	p.false	TDR	FDR	PA	PA1	PA2	GM	AUC
1	50.00	0.00	0.00	0.07	0.07	0.10	0.90	0.50	0.50	0.50	0.50	0.50
2	50.00	0.00	0.50	0.11	0.10	0.10	0.90	0.50	0.50	0.50	0.50	0.50
3	50.00	0.00	0.80	0.12	0.12	0.10	0.90	0.50	0.50	0.50	0.50	0.50
4	50.00	0.50	0.00	0.21	0.02	0.55	0.45	0.80	0.80	0.79	0.79	0.88
5	50.00	0.50	0.50	0.35	0.08	0.39	0.61	0.72	0.72	0.73	0.72	0.80
6	50.00	0.50	0.80	0.41	0.10	0.35	0.65	0.67	0.67	0.68	0.67	0.74
7	50.00	1.00	0.00	0.15	0.00	0.97	0.03	0.96	0.96	0.96	0.96	0.99
8	50.00	1.00	0.50	0.35	0.02	0.87	0.13	0.94	0.94	0.94	0.94	0.98
9	50.00	1.00	0.80	0.63	0.06	0.71	0.29	0.91	0.91	0.91	0.91	0.97
10	100.00	0.00	0.00	0.19	0.19	0.10	0.90	0.50	0.50	0.50	0.50	0.50
11	100.00	0.00	0.50	0.14	0.15	0.10	0.90	0.50	0.50	0.50	0.50	0.50
12	100.00	0.00	0.80	0.12	0.12	0.10	0.90	0.50	0.50	0.50	0.50	0.50
13	100.00	0.50	0.00	0.54	0.04	0.69	0.31	0.94	0.94	0.94	0.94	0.98
14	100.00	0.50	0.50	0.69	0.13	0.42	0.57	0.79	0.79	0.79	0.79	0.88
15	100.00	0.50	0.80	0.65	0.12	0.44	0.56	0.73	0.73	0.73	0.73	0.81
16	100.00	1.00	0.00	0.20	0.00	1.00	0.00	0.98	0.98	0.98	0.98	1.00
17	100.00	1.00	0.50	0.64	0.06	0.81	0.19	0.96	0.96	0.96	0.96	0.99
18	100.00	1.00	0.80	0.86	0.11	0.63	0.37	0.94	0.94	0.94	0.94	0.99
19	250.00	0.00	0.00	0.21	0.21	0.10	0.90	0.50	0.50	0.50	0.50	0.50
20	250.00	0.00	0.50	0.15	0.15	0.10	0.90	0.50	0.50	0.50	0.50	0.50
21	250.00	0.00	0.80	0.11	0.10	0.10	0.90	0.50	0.50	0.50	0.50	0.50
22	250.00	0.50	0.00	0.94	0.10	0.64	0.36	0.98	0.98	0.98	0.98	1.00
23	250.00	0.50	0.50	0.95	0.17	0.43	0.57	0.83	0.83	0.83	0.83	0.91
24	250.00	0.50	0.80	0.92	0.14	0.49	0.51	0.78	0.78	0.78	0.78	0.86
25	250.00	1.00	0.00	0.29	0.00	1.00	0.00	0.99	0.99	0.99	0.99	1.00
26	250.00	1.00	0.50	0.85	0.11	0.67	0.33	0.98	0.98	0.98	0.98	1.00
27	250.00	1.00	0.80	0.93	0.14	0.56	0.44	0.95	0.95	0.95	0.95	0.99

Tabela 15: Povprečni rezultati prvotnih simulacijah za metodo lasso.

	sample	mde	rho	p.corr	p.false	TDR	FDR	PA	PA1	PA2	GM	AUC
1	50.00	0.00	0.00	0.02	0.02	0.10	0.90	0.50	0.50	0.50	0.49	0.50
2	50.00	0.00	0.50	0.02	0.02	0.10	0.90	0.50	0.50	0.50	0.50	0.50
3	50.00	0.00	0.80	0.02	0.02	0.10	0.90	0.50	0.50	0.50	0.49	0.50
4	50.00	0.50	0.00	0.14	0.01	0.67	0.33	0.73	0.74	0.73	0.73	0.81
5	50.00	0.50	0.50	0.09	0.01	0.55	0.45	0.66	0.66	0.66	0.66	0.72
6	50.00	0.50	0.80	0.07	0.01	0.49	0.51	0.63	0.63	0.64	0.63	0.69
7	50.00	1.00	0.00	0.22	0.00	0.99	0.01	0.96	0.96	0.96	0.96	0.99
8	50.00	1.00	0.50	0.16	0.00	0.91	0.09	0.90	0.90	0.90	0.90	0.96
9	50.00	1.00	0.80	0.13	0.00	0.83	0.17	0.87	0.88	0.87	0.87	0.95
10	100.00	0.00	0.00	0.04	0.04	0.10	0.90	0.50	0.49	0.51	0.49	0.50
11	100.00	0.00	0.50	0.04	0.04	0.10	0.90	0.50	0.50	0.50	0.50	0.50
12	100.00	0.00	0.80	0.04	0.04	0.10	0.90	0.50	0.50	0.50	0.50	0.50
13	100.00	0.50	0.00	0.34	0.01	0.76	0.24	0.86	0.85	0.86	0.85	0.93
14	100.00	0.50	0.50	0.18	0.02	0.64	0.36	0.73	0.73	0.73	0.73	0.81
15	100.00	0.50	0.80	0.13	0.02	0.55	0.45	0.70	0.70	0.70	0.70	0.77
16	100.00	1.00	0.00	0.35	0.00	1.00	0.00	0.99	0.99	0.99	0.99	1.00
17	100.00	1.00	0.50	0.27	0.00	0.94	0.06	0.94	0.94	0.94	0.94	0.99
18	100.00	1.00	0.80	0.20	0.00	0.86	0.14	0.92	0.92	0.92	0.92	0.98
19	250.00	0.00	0.00	0.08	0.08	0.10	0.90	0.50	0.50	0.50	0.50	0.50
20	250.00	0.00	0.50	0.08	0.08	0.10	0.90	0.50	0.50	0.50	0.50	0.50
21	250.00	0.00	0.80	0.07	0.07	0.10	0.90	0.50	0.50	0.50	0.50	0.50
22	250.00	0.50	0.00	0.66	0.01	0.85	0.15	0.95	0.95	0.95	0.95	0.99
23	250.00	0.50	0.50	0.33	0.02	0.72	0.28	0.80	0.80	0.79	0.79	0.88
24	250.00	0.50	0.80	0.22	0.02	0.67	0.33	0.76	0.76	0.76	0.76	0.84
25	250.00	1.00	0.00	0.52	0.00	1.00	0.00	1.00	1.00	1.00	1.00	1.00
26	250.00	1.00	0.50	0.41	0.00	0.94	0.06	0.96	0.96	0.96	0.96	0.99
27	250.00	1.00	0.80	0.29	0.01	0.87	0.13	0.94	0.94	0.94	0.94	0.99

Tabela 16: Povprečni rezultati prvotnih simulacijah za metodo elastic.

	sample	mde	rho	p.corr	p.false	TDR	FDR	PA	PA1	PA2	GM	AUC
1	50.00	0.00	0.00	0.07	0.08	0.10	0.90	0.50	0.50	0.50	0.50	0.50
2	50.00	0.00	0.50	0.07	0.07	0.10	0.90	0.50	0.50	0.50	0.50	0.50
3	50.00	0.00	0.80	0.07	0.07	0.10	0.90	0.50	0.50	0.50	0.50	0.50
4	50.00	0.50	0.00	0.52	0.13	0.38	0.62	0.85	0.86	0.85	0.85	0.93
5	50.00	0.50	0.50	0.32	0.08	0.42	0.58	0.70	0.70	0.71	0.70	0.78
6	50.00	0.50	0.80	0.29	0.08	0.42	0.58	0.66	0.66	0.66	0.66	0.72
7	50.00	1.00	0.00	0.62	0.00	0.96	0.04	1.00	1.00	1.00	1.00	1.00
8	50.00	1.00	0.50	0.70	0.04	0.76	0.24	0.96	0.96	0.96	0.96	0.99
9	50.00	1.00	0.80	0.66	0.05	0.70	0.30	0.91	0.92	0.91	0.91	0.97
10	100.00	0.00	0.00	0.10	0.10	0.10	0.90	0.50	0.50	0.50	0.50	0.50
11	100.00	0.00	0.50	0.09	0.09	0.10	0.90	0.50	0.50	0.50	0.50	0.50
12	100.00	0.00	0.80	0.10	0.10	0.10	0.90	0.50	0.50	0.50	0.50	0.50
13	100.00	0.50	0.00	0.76	0.13	0.46	0.54	0.94	0.94	0.95	0.94	0.99
14	100.00	0.50	0.50	0.48	0.08	0.51	0.49	0.77	0.77	0.77	0.77	0.85
15	100.00	0.50	0.80	0.39	0.07	0.51	0.49	0.72	0.72	0.71	0.71	0.79
16	100.00	1.00	0.00	0.68	0.00	1.00	0.00	1.00	1.00	1.00	1.00	1.00
17	100.00	1.00	0.50	0.81	0.03	0.84	0.16	0.97	0.97	0.97	0.97	1.00
18	100.00	1.00	0.80	0.76	0.04	0.78	0.22	0.94	0.94	0.94	0.94	0.98
19	250.00	0.00	0.00	0.15	0.15	0.10	0.90	0.50	0.50	0.50	0.50	0.50
20	250.00	0.00	0.50	0.13	0.13	0.10	0.90	0.50	0.50	0.50	0.50	0.50
21	250.00	0.00	0.80	0.12	0.12	0.10	0.90	0.50	0.50	0.50	0.50	0.50
22	250.00	0.50	0.00	0.95	0.08	0.64	0.36	0.98	0.98	0.98	0.98	1.00
23	250.00	0.50	0.50	0.69	0.06	0.66	0.34	0.82	0.82	0.82	0.82	0.90
24	250.00	0.50	0.80	0.56	0.05	0.67	0.33	0.77	0.77	0.77	0.77	0.85
25	250.00	1.00	0.00	0.75	0.00	1.00	0.00	1.00	1.00	1.00	1.00	1.00
26	250.00	1.00	0.50	0.90	0.03	0.84	0.16	0.98	0.98	0.98	0.98	1.00
27	250.00	1.00	0.80	0.82	0.04	0.81	0.19	0.95	0.95	0.95	0.95	0.99

Tabela 17: Povprečni rezultati prvotnih simulacijah za metodo bolasso.

	sample	mde	rho	p.corr	p.false	TDR	FDR	PA	PA1	PA2	GM	AUC
1	50.00	0.00	0.00	0.06	0.06	0.10	0.90	0.50	0.50	0.50	0.50	0.50
2	50.00	0.00	0.50	0.06	0.06	0.10	0.90	0.50	0.50	0.50	0.50	0.50
3	50.00	0.00	0.80	0.06	0.06	0.10	0.90	0.50	0.50	0.50	0.50	0.50
4	50.00	0.50	0.00	0.29	0.03	0.54	0.46	0.75	0.76	0.75	0.75	0.83
5	50.00	0.50	0.50	0.24	0.03	0.44	0.56	0.67	0.67	0.67	0.67	0.73
6	50.00	0.50	0.80	0.21	0.04	0.38	0.62	0.64	0.64	0.64	0.64	0.69
7	50.00	1.00	0.00	0.50	0.00	0.97	0.03	0.96	0.96	0.96	0.96	0.99
8	50.00	1.00	0.50	0.42	0.00	0.90	0.10	0.91	0.91	0.91	0.91	0.97
9	50.00	1.00	0.80	0.37	0.01	0.82	0.18	0.88	0.88	0.88	0.88	0.95
10	100.00	0.00	0.00	0.14	0.14	0.10	0.90	0.50	0.50	0.50	0.50	0.50
11	100.00	0.00	0.50	0.14	0.14	0.10	0.90	0.50	0.50	0.50	0.50	0.50
12	100.00	0.00	0.80	0.15	0.15	0.10	0.90	0.50	0.50	0.50	0.50	0.50
13	100.00	0.50	0.00	0.61	0.04	0.63	0.37	0.87	0.87	0.87	0.87	0.94
14	100.00	0.50	0.50	0.48	0.06	0.47	0.53	0.73	0.73	0.73	0.73	0.81
15	100.00	0.50	0.80	0.42	0.08	0.37	0.63	0.69	0.69	0.69	0.69	0.76
16	100.00	1.00	0.00	0.74	0.00	1.00	0.00	0.99	0.99	0.99	0.99	1.00
17	100.00	1.00	0.50	0.63	0.00	0.95	0.05	0.95	0.95	0.95	0.95	0.99
18	100.00	1.00	0.80	0.56	0.01	0.86	0.14	0.92	0.92	0.92	0.92	0.98
19	250.00	0.00	0.00	0.38	0.38	0.10	0.90	0.50	0.50	0.50	0.50	0.50
20	250.00	0.00	0.50	0.41	0.40	0.10	0.90	0.50	0.50	0.50	0.50	0.50
21	250.00	0.00	0.80	0.44	0.44	0.10	0.90	0.50	0.50	0.50	0.50	0.50
22	250.00	0.50	0.00	0.92	0.04	0.71	0.29	0.95	0.95	0.95	0.95	0.99
23	250.00	0.50	0.50	0.77	0.15	0.37	0.63	0.78	0.78	0.78	0.78	0.86
24	250.00	0.50	0.80	0.67	0.21	0.26	0.74	0.75	0.75	0.75	0.75	0.83
25	250.00	1.00	0.00	0.91	0.00	1.00	0.00	1.00	1.00	1.00	1.00	1.00
26	250.00	1.00	0.50	0.82	0.00	0.97	0.03	0.97	0.97	0.96	0.97	0.99
27	250.00	1.00	0.80	0.74	0.02	0.80	0.20	0.94	0.94	0.94	0.94	0.99

Tabela 18: Povprečni rezultati prvotnih simulacijah za metodo bolasso.all.

	sample	mde	rho	p.corr	p.false	TDR	FDR	PA	PA1	PA2	GM	AUC
1	50.00	0.00	0.00	0.29	0.29	0.10	0.90					
2	50.00	0.00	0.50	0.29	0.30	0.10	0.90					
3	50.00	0.00	0.80	0.31	0.32	0.10	0.90					
4	50.00	0.50	0.00	0.60	0.19	0.26	0.74					
5	50.00	0.50	0.50	0.54	0.22	0.22	0.78					
6	50.00	0.50	0.80	0.52	0.25	0.19	0.81					
7	50.00	1.00	0.00	0.78	0.03	0.72	0.28					
8	50.00	1.00	0.50	0.73	0.07	0.55	0.45					
9	50.00	1.00	0.80	0.69	0.10	0.44	0.56					
10	100.00	0.00	0.00	0.49	0.49	0.10	0.90					
11	100.00	0.00	0.50	0.50	0.50	0.10	0.90					
12	100.00	0.00	0.80	0.53	0.53	0.10	0.90					
13	100.00	0.50	0.00	0.84	0.23	0.29	0.71					
14	100.00	0.50	0.50	0.76	0.31	0.22	0.78					
15	100.00	0.50	0.80	0.72	0.38	0.18	0.82					
16	100.00	1.00	0.00	0.92	0.01	0.92	0.08					
17	100.00	1.00	0.50	0.86	0.06	0.64	0.35					
18	100.00	1.00	0.80	0.82	0.11	0.46	0.54					
19	250.00	0.00	0.00	0.85	0.85	0.10	0.90					
20	250.00	0.00	0.50	0.88	0.88	0.10	0.90					
21	250.00	0.00	0.80	0.90	0.90	0.10	0.90					
22	250.00	0.50	0.00	0.98	0.25	0.30	0.70					
23	250.00	0.50	0.50	0.93	0.53	0.16	0.84					
24	250.00	0.50	0.80	0.88	0.66	0.13	0.87					
25	250.00	1.00	0.00	0.98	0.00	1.00	0.00					
26	250.00	1.00	0.50	0.95	0.05	0.68	0.32					
27	250.00	1.00	0.80	0.91	0.17	0.39	0.61					

Tabela 19: Povprečni rezultati prvotnih simulacijah za metodo boelastic.freq.50.

	sample	mde	rho	p.corr	p.false	TDR	FDR	PA	PA1	PA2	GM	AUC
1	50.00	0.00	0.00	0.07	0.07	0.10	0.90	0.50	0.50	0.50	0.50	0.50
2	50.00	0.00	0.50	0.08	0.08	0.10	0.90	0.50	0.50	0.50	0.50	0.50
3	50.00	0.00	0.80	0.09	0.09	0.10	0.90	0.50	0.50	0.50	0.50	0.50
4	50.00	0.50	0.00	0.51	0.09	0.39	0.61	0.88	0.88	0.87	0.87	0.95
5	50.00	0.50	0.50	0.43	0.07	0.40	0.60	0.73	0.73	0.73	0.73	0.81
6	50.00	0.50	0.80	0.45	0.08	0.40	0.60	0.67	0.67	0.68	0.67	0.74
7	50.00	1.00	0.00	0.71	0.01	0.94	0.06	1.00	1.00	1.00	1.00	1.00
8	50.00	1.00	0.50	0.78	0.02	0.84	0.16	0.97	0.97	0.97	0.97	0.99
9	50.00	1.00	0.80	0.82	0.02	0.80	0.20	0.93	0.93	0.93	0.93	0.98
10	100.00	0.00	0.00	0.12	0.12	0.10	0.90	0.50	0.50	0.50	0.50	0.50
11	100.00	0.00	0.50	0.14	0.14	0.10	0.90	0.50	0.50	0.50	0.50	0.50
12	100.00	0.00	0.80	0.18	0.18	0.10	0.90	0.50	0.50	0.50	0.50	0.50
13	100.00	0.50	0.00	0.78	0.10	0.48	0.52	0.95	0.95	0.95	0.95	0.99
14	100.00	0.50	0.50	0.68	0.08	0.48	0.52	0.78	0.78	0.78	0.78	0.86
15	100.00	0.50	0.80	0.72	0.11	0.42	0.57	0.71	0.71	0.71	0.71	0.79
16	100.00	1.00	0.00	0.76	0.00	1.00	0.00	1.00	1.00	1.00	1.00	1.00
17	100.00	1.00	0.50	0.93	0.00	0.96	0.04	0.97	0.98	0.97	0.97	1.00
18	100.00	1.00	0.80	0.96	0.02	0.87	0.13	0.94	0.94	0.94	0.94	0.99
19	250.00	0.00	0.00	0.26	0.26	0.10	0.90	0.50	0.50	0.50	0.50	0.50
20	250.00	0.00	0.50	0.32	0.31	0.10	0.90	0.50	0.50	0.50	0.50	0.50
21	250.00	0.00	0.80	0.42	0.42	0.10	0.90	0.50	0.50	0.50	0.50	0.50
22	250.00	0.50	0.00	0.96	0.04	0.71	0.29	0.98	0.98	0.98	0.98	1.00
23	250.00	0.50	0.50	0.91	0.13	0.44	0.56	0.80	0.80	0.80	0.80	0.88
24	250.00	0.50	0.80	0.93	0.21	0.33	0.67	0.74	0.74	0.74	0.74	0.82
25	250.00	1.00	0.00	0.84	0.00	1.00	0.00	1.00	1.00	1.00	1.00	1.00
26	250.00	1.00	0.50	1.00	0.00	0.96	0.04	0.98	0.98	0.98	0.98	1.00
27	250.00	1.00	0.80	1.00	0.04	0.76	0.24	0.95	0.95	0.95	0.95	0.99

Tabela 20: Povprečni rezultati prvotnih simulacijah za metodo boelastic.freq.75.

	sample	mde	rho	p.corr	p.false	TDR	FDR	PA	PA1	PA2	GM	AUC
1	50.00	0.00	0.00	0.02	0.02	0.10	0.90	0.50	0.50	0.50	0.50	0.50
2	50.00	0.00	0.50	0.02	0.02	0.10	0.90	0.50	0.50	0.50	0.50	0.50
3	50.00	0.00	0.80	0.02	0.02	0.10	0.90	0.50	0.50	0.50	0.50	0.50
4	50.00	0.50	0.00	0.28	0.02	0.58	0.42	0.84	0.84	0.84	0.84	0.92
5	50.00	0.50	0.50	0.21	0.02	0.60	0.40	0.71	0.71	0.71	0.71	0.79
6	50.00	0.50	0.80	0.22	0.02	0.59	0.40	0.66	0.66	0.66	0.66	0.73
7	50.00	1.00	0.00	0.45	0.00	0.99	0.01	1.00	1.00	1.00	1.00	1.00
8	50.00	1.00	0.50	0.55	0.00	0.96	0.04	0.96	0.96	0.96	0.96	0.99
9	50.00	1.00	0.80	0.61	0.00	0.94	0.06	0.92	0.92	0.92	0.92	0.98
10	100.00	0.00	0.00	0.03	0.03	0.10	0.90	0.50	0.50	0.50	0.50	0.50
11	100.00	0.00	0.50	0.04	0.04	0.10	0.90	0.50	0.50	0.50	0.50	0.50
12	100.00	0.00	0.80	0.05	0.05	0.10	0.90	0.50	0.50	0.50	0.50	0.50
13	100.00	0.50	0.00	0.56	0.02	0.74	0.26	0.94	0.94	0.94	0.94	0.99
14	100.00	0.50	0.50	0.44	0.02	0.73	0.27	0.78	0.78	0.78	0.78	0.87
15	100.00	0.50	0.80	0.48	0.03	0.67	0.33	0.72	0.72	0.72	0.72	0.80
16	100.00	1.00	0.00	0.50	0.00	1.00	0.00	1.00	1.00	1.00	1.00	1.00
17	100.00	1.00	0.50	0.77	0.00	0.99	0.00	0.97	0.97	0.97	0.97	1.00
18	100.00	1.00	0.80	0.87	0.00	0.98	0.02	0.94	0.95	0.94	0.94	0.99
19	250.00	0.00	0.00	0.08	0.08	0.10	0.90	0.50	0.50	0.50	0.50	0.50
20	250.00	0.00	0.50	0.10	0.10	0.10	0.90	0.50	0.50	0.50	0.50	0.50
21	250.00	0.00	0.80	0.14	0.14	0.10	0.90	0.50	0.50	0.50	0.50	0.50
22	250.00	0.50	0.00	0.87	0.01	0.93	0.07	0.98	0.98	0.98	0.98	1.00
23	250.00	0.50	0.50	0.77	0.03	0.72	0.28	0.82	0.82	0.81	0.81	0.90
24	250.00	0.50	0.80	0.80	0.06	0.60	0.40	0.76	0.76	0.76	0.76	0.84
25	250.00	1.00	0.00	0.61	0.00	1.00	0.00	1.00	1.00	1.00	1.00	1.00
26	250.00	1.00	0.50	0.98	0.00	1.00	0.00	0.98	0.98	0.98	0.98	1.00
27	250.00	1.00	0.80	0.99	0.00	0.96	0.04	0.95	0.95	0.95	0.95	0.99

Tabela 21: Povprečni rezultati prvotnih simulacijah za metodo pam.

	sample	mde	rho	p.corr	p.false	TDR	FDR	PA	PA1	PA2	GM	AUC
1	50.00	0.00	0.00	0.21	0.21	0.10	0.90	0.50	0.51	0.49	0.48	0.50
2	50.00	0.00	0.50	0.18	0.18	0.10	0.90	0.50	0.51	0.49	0.49	0.50
3	50.00	0.00	0.80	0.22	0.22	0.09	0.91	0.50	0.52	0.48	0.48	0.50
4	50.00	0.50	0.00	0.68	0.31	0.31	0.69	0.88	0.88	0.88	0.88	0.95
5	50.00	0.50	0.50	0.55	0.22	0.42	0.58	0.72	0.72	0.73	0.72	0.80
6	50.00	0.50	0.80	0.54	0.24	0.45	0.55	0.67	0.66	0.67	0.66	0.73
7	50.00	1.00	0.00	0.37	0.00	0.99	0.01	0.98	0.98	0.98	0.98	1.00
8	50.00	1.00	0.50	0.82	0.10	0.73	0.27	0.96	0.96	0.96	0.96	0.99
9	50.00	1.00	0.80	0.90	0.14	0.63	0.37	0.92	0.92	0.92	0.92	0.98
10	100.00	0.00	0.00	0.22	0.22	0.10	0.90	0.50	0.51	0.49	0.49	0.50
11	100.00	0.00	0.50	0.23	0.23	0.10	0.90	0.50	0.52	0.48	0.48	0.50
12	100.00	0.00	0.80	0.27	0.27	0.09	0.91	0.50	0.52	0.48	0.48	0.50
13	100.00	0.50	0.00	0.84	0.22	0.40	0.60	0.95	0.95	0.95	0.95	0.99
14	100.00	0.50	0.50	0.81	0.21	0.44	0.56	0.80	0.80	0.80	0.80	0.88
15	100.00	0.50	0.80	0.79	0.23	0.45	0.55	0.73	0.73	0.73	0.73	0.81
16	100.00	1.00	0.00	0.46	0.00	1.00	0.00	0.99	0.99	0.99	0.99	1.00
17	100.00	1.00	0.50	0.94	0.11	0.77	0.23	0.97	0.97	0.97	0.97	1.00
18	100.00	1.00	0.80	0.98	0.12	0.71	0.29	0.95	0.95	0.95	0.95	0.99
19	250.00	0.00	0.00	0.24	0.24	0.10	0.90	0.50	0.51	0.49	0.48	0.50
20	250.00	0.00	0.50	0.22	0.22	0.10	0.90	0.50	0.52	0.48	0.48	0.50
21	250.00	0.00	0.80	0.26	0.25	0.10	0.90	0.50	0.51	0.49	0.49	0.50
22	250.00	0.50	0.00	0.97	0.13	0.60	0.40	0.98	0.98	0.98	0.98	1.00
23	250.00	0.50	0.50	0.98	0.18	0.51	0.49	0.84	0.84	0.84	0.84	0.92
24	250.00	0.50	0.80	0.98	0.17	0.53	0.47	0.78	0.78	0.79	0.78	0.87
25	250.00	1.00	0.00	0.57	0.00	1.00	0.00	1.00	1.00	1.00	1.00	1.00
26	250.00	1.00	0.50	0.99	0.11	0.78	0.22	0.98	0.98	0.98	0.98	1.00
27	250.00	1.00	0.80	1.00	0.13	0.72	0.28	0.95	0.95	0.95	0.95	0.99

Tabela 22: Povprečni rezultati prvotnih simulacijah za metodo bopam.

	sample	mde	rho	p.corr	p.false	TDR	FDR	PA	PA1	PA2	GM	AUC
1	50.00	0.00	0.00	0.14	0.14	0.10	0.90	0.50	0.50	0.50	0.50	0.50
2	50.00	0.00	0.50	0.18	0.18	0.10	0.90	0.50	0.50	0.50	0.50	0.50
3	50.00	0.00	0.80	0.27	0.27	0.10	0.90	0.50	0.50	0.50	0.50	0.50
4	50.00	0.50	0.00	0.30	0.11	0.37	0.63	0.80	0.80	0.80	0.80	0.88
5	50.00	0.50	0.50	0.36	0.14	0.35	0.65	0.71	0.71	0.71	0.70	0.78
6	50.00	0.50	0.80	0.50	0.26	0.28	0.72	0.65	0.65	0.65	0.65	0.71
7	50.00	1.00	0.00	0.07	0.00	0.99	0.01	0.86	0.86	0.86	0.86	0.93
8	50.00	1.00	0.50	0.14	0.00	0.97	0.03	0.85	0.85	0.85	0.85	0.93
9	50.00	1.00	0.80	0.40	0.02	0.85	0.15	0.85	0.85	0.85	0.85	0.93
10	100.00	0.00	0.00	0.33	0.33	0.10	0.90	0.50	0.50	0.50	0.50	0.50
11	100.00	0.00	0.50	0.38	0.38	0.10	0.90	0.50	0.50	0.50	0.50	0.50
12	100.00	0.00	0.80	0.49	0.49	0.10	0.90	0.50	0.51	0.49	0.49	0.50
13	100.00	0.50	0.00	0.57	0.14	0.40	0.60	0.93	0.93	0.93	0.93	0.98
14	100.00	0.50	0.50	0.73	0.30	0.29	0.71	0.80	0.80	0.80	0.80	0.88
15	100.00	0.50	0.80	0.81	0.43	0.23	0.77	0.73	0.73	0.73	0.73	0.80
16	100.00	1.00	0.00	0.14	0.00	1.00	0.00	0.95	0.95	0.95	0.95	0.99
17	100.00	1.00	0.50	0.43	0.00	0.98	0.02	0.94	0.94	0.94	0.94	0.98
18	100.00	1.00	0.80	0.85	0.08	0.76	0.24	0.93	0.93	0.93	0.93	0.98
19	250.00	0.00	0.00	0.62	0.62	0.10	0.90	0.50	0.50	0.50	0.50	0.50
20	250.00	0.00	0.50	0.58	0.58	0.10	0.90	0.50	0.50	0.50	0.50	0.50
21	250.00	0.00	0.80	0.66	0.66	0.10	0.90	0.50	0.50	0.50	0.50	0.50
22	250.00	0.50	0.00	0.80	0.08	0.62	0.38	0.98	0.98	0.98	0.98	1.00
23	250.00	0.50	0.50	0.96	0.42	0.26	0.74	0.84	0.84	0.84	0.84	0.92
24	250.00	0.50	0.80	0.98	0.47	0.23	0.77	0.78	0.78	0.78	0.78	0.86
25	250.00	1.00	0.00	0.25	0.00	1.00	0.00	0.99	0.99	0.99	0.99	1.00
26	250.00	1.00	0.50	0.89	0.06	0.89	0.11	0.97	0.97	0.97	0.97	0.99
27	250.00	1.00	0.80	0.99	0.20	0.60	0.40	0.94	0.94	0.94	0.94	0.98

B Rezultati uravnoteženih podatkov

Tabela 23: Rezultati vseh metod na pravih podatkih, po uravnoteženju podatkov z največjimi vsotami.

	skupina	metoda	r	PA	PA1	PA2	GM	AUC
1	ER	rf.freq	11	0.87	0.87	0.86	0.86	0.94
2	ER	stump	64	0.83	0.82	0.85	0.83	0.89
3	ER	stump.freq	26	0.83	0.82	0.84	0.82	0.90
4	ER	lasso	23	0.85	0.82	0.88	0.85	0.94
5	ER	elastic	28	0.87	0.84	0.89	0.86	0.94
6	ER	pam	572	0.83	0.81	0.86	0.83	0.91
7	grade	rf.freq	16	0.82	0.71	0.93	0.81	0.90
8	grade	stump	100	0.71	0.61	0.80	0.69	0.79
9	grade	stump.freq	29	0.72	0.61	0.83	0.70	0.81
10	grade	lasso	12	0.76	0.77	0.75	0.75	0.85
11	grade	elastic	19	0.77	0.78	0.76	0.76	0.85
12	grade	pam	259	0.67	0.68	0.67	0.66	0.76

Tabela 24: Rezultati vseh metod na pravih podatkih, po uravnoteženju podatkov s srednjimi vsotami.

	skupina	metoda	r	PA	PA1	PA2	GM	AUC
1	ER	rf.freq	11	0.87	0.87	0.86	0.86	0.94
2	ER	stump	64	0.83	0.82	0.84	0.82	0.90
3	ER	stump.freq	26	0.83	0.82	0.84	0.82	0.90
4	ER	lasso	23	0.85	0.81	0.89	0.84	0.94
5	ER	elastic	28	0.87	0.85	0.89	0.86	0.94
6	ER	pam	572	0.83	0.81	0.86	0.83	0.91
7	grade	rf.freq	16	0.82	0.71	0.93	0.81	0.90
8	grade	stump	100	0.71	0.61	0.81	0.69	0.80
9	grade	stump.freq	29	0.72	0.61	0.83	0.70	0.81
10	grade	lasso	12	0.76	0.77	0.75	0.75	0.85
11	grade	elastic	19	0.77	0.78	0.77	0.76	0.85
12	grade	pam	259	0.67	0.68	0.67	0.66	0.76

Tabela 25: Rezultati vseh metod na pravih podatkih, po uravnoteženju podatkov z najmanjšimi vsotami.

	skupina	metoda	r	PA	PA1	PA2	GM	AUC
1	ER	rf.freq	14	0.91	0.97	0.85	0.91	0.98
2	ER	stump	31	0.90	0.95	0.85	0.90	0.96
3	ER	stump.freq	14	0.90	0.95	0.85	0.90	0.96
4	ER	lasso	2	0.91	0.97	0.86	0.91	0.93
5	ER	elastic	25	0.95	0.99	0.91	0.95	0.98
6	ER	pam	684	0.91	0.95	0.86	0.90	0.93
7	grade	rf.freq	13	0.79	0.80	0.78	0.78	0.91
8	grade	stump	106	0.70	0.75	0.66	0.69	0.81
9	grade	stump.freq	33	0.72	0.77	0.66	0.70	0.83
10	grade	lasso	11	0.80	0.81	0.79	0.79	0.90
11	grade	elastic	11	0.80	0.83	0.78	0.79	0.90
12	grade	pam	800	0.71	0.75	0.68	0.70	0.78

C R koda za prvotne simulacije

```
1  library(class)
2  library(Hmisc)
3  library(randomForest)
4  library(pamr)
5  library(glmnet)
6  library(caret)
7  library(e1071)
8  library(doParallel)
9  library(foreach)
10
11 pas=function(fit,ctest, fit .prob) {
12   PA=(length(which(fit==0&ctest==0))+length(which(fit==1&ctest==1)))/length(fit)
13
14   pa1=length(which(fit==0&ctest==0))/length(which(ctest==0))
15   pa2=length(which(fit==1&ctest==1))/length(which(ctest==1))
16
17   gm=sqrt(pa1*pa2)
18
19   AUC=somers2(fit.prob,ctest)["C"]
20
21   c(PA,pa1,pa2,gm,AUC)
22 }
23
24 var.acc=function(true.dif,true.nd,fit.dif) {
25   perc.cor=sum(fit.dif%in%true.dif)/length(true.dif)
26   perc.false=sum(fit.dif%in%true.nd)/length(true.nd)
27
28   pv.cor.1=sum(fit.dif%in%true.dif)
29   pv.cor.2=length(fit.dif)
30   if (pv.cor.2==0) {
31     pv.cor="NaN"
32   } else {
33     pv.cor=pv.cor.1/pv.cor.2
34   }
35
36   fdr.1=sum(fit.dif%in%true.nd)
37   fdr.2=length(fit.dif)
38   if (fdr.2==0) {
39     fdr="NaN"
40   } else {
41     fdr=fdr.1/fdr.2
42   }
43
44   c(perc.cor,perc.false,pv.cor,fdr)
45 }
46
47 fun=function(nu.file,G,n.training,n.test,percent.class.training,percent.class.test,nu.different.genes,mean.difference,num.blocks,
       rho) {
48   n=n.training+n.test
49
50   n.training.class.1=n.training*percent.class.training
51   n.training.class.2=n.training-n.training*percent.class.training
52
53   n.test.class.1=n.test*percent.class.test
54   n.test.class.2=n.test-n.test*percent.class.test
55
56   num.genes=G
57   var.sigma.block=rep(1,num.genes)
58   mu.Class1=rep(0,num.genes)
59   mu.Class2=c(rep(mean.difference,nu.different.genes),rep(0,num.genes-nu.different.genes))
60   num.samples=n.training
61   n1=n.training.class.1
62   n2=n.training.class.2
63
64   #number of genes within each block
```

```
#NB! implies that the blocks have the same number of genes
#if more flexibility is needed can be changed
num.genes.per.block<-num.genes/num.blocks

#standard deviation for each gene
sd.genes<-rep(sqrt(var.sigma.block), each=num.genes.per.block)

#desired correlation between genes within each block, obtained from the specified correlation common to each block,
#can be made more flexible
cor.block<-rep(rho, num.blocks)

#sqrt of variance of the random effect to obtain a correlation of cor.block within each block
sd.RE<-sqrt(var.sigma.block*cor.block)


#standard deviation for each gene
  sd.genes<-rep(sqrt(var.sigma.block-sd.RE^2), each=num.genes.per.block)

#generation of gene expression data

#data from all the genes for samples of Class1
data.Class1<-matrix(rnorm(num.genes*n1, mu.Class1, sd.genes),
ncol=num.genes, nrow=n1, byrow=T)

#data from all the genes for samples of Class1
data.Class2<-matrix(rnorm(num.genes*n2, mu.Class2, sd.genes),
ncol=num.genes, nrow=n2, byrow=T)

#implies that the blocks have the same number of genes

#simulated RE for each of the blocks, block-specific, the same for all the subjects
#sim.RE<-rnorm(num.blocks, 0, sd.RE)

tmp<-lapply(1:num.blocks, function(x) {
  x1<-(x-1)*num.genes.per.block+1;
  x2<- x*num.genes.per.block;
  data.Class1[1:n1,x1:x2]<<-data.Class1[1:n1, x1:x2]+rnorm(n1, 0, sd.RE[x]);
  data.Class2[1:n2, x1:x2]<<-data.Class2[1:n2, x1:x2]+rnorm(n2, 0, sd.RE[x])
}
)

#full matrix of data, rows: samples, cols: genes

training<-rbind(data.Class1, data.Class2)

###data generation for test

num.samples=n.test
n1=n.test.class.1
n2=n.test.class.2

data.Class1<-matrix(rnorm(num.genes*n1, mu.Class1, sd.genes),
ncol=num.genes, nrow=n1, byrow=T)

#data from all the genes for samples of Class1
data.Class2<-matrix(rnorm(num.genes*n2, mu.Class2, sd.genes),
ncol=num.genes, nrow=n2, byrow=T)

#implies that the blocks have the same number of genes

#simulated RE for each of the blocks, block-specific, the same for all the subjects
#sim.RE<-rnorm(num.blocks, 0, sd.RE)

tmp<-lapply(1:num.blocks, function(x) {
  x1<-(x-1)*num.genes.per.block+1;
  x2<- x*num.genes.per.block;
  data.Class1[1:n1,x1:x2]<<-data.Class1[1:n1, x1:x2]+rnorm(n1, 0, sd.RE[x]);
  data.Class2[1:n2, x1:x2]<<-data.Class2[1:n2, x1:x2]+rnorm(n2, 0, sd.RE[x])
}
)

#full matrix of data, rows: samples, cols: genes

test<-rbind(data.Class1, data.Class2)

class.training=c(rep(0,n.training.class.1),rep(1,n.training.class.2))
class.test=c(rep(0,n.test.class.1),rep(1,n.test.class.2))

```

```
143
144  ###filter-caret
145   filter .caret=filterVarImp(x=data.frame(training),y=as.factor(class.training))
146  temp.sort=which(filter.caret[,2]>=2/3)
147  var.acc. filter .caret=var.acc(1:nu.different .genes,(nu. different .genes+1):G,as.numeric(temp.sort))
148  write(var.acc. filter .caret, file=paste("filter.caret.var.acc. file .",nu. file ,"samp",n.training,"mde",mean.difference,"rho",rho,".
         txt",sep=""),ncol=length(var.acc.filter.caret),append=TRUE,sep="\t")
149
150
151  ##################randomForest
152   fit .rf=randomForest(as.factor(class.training)~.,data=data.frame(training),ntree=1000,importance=TRUE)
153
154  #var imp caret
155  var.imp.rf.2=varImp(fit.rf) [,2]
156  var.imp.rf.2=which(var.imp.rf.2!=0)
157  var.acc.rf.2=var.acc(1:nu.different .genes,(nu. different .genes+1):G,as.numeric(var.imp.rf.2))
158  write(var.acc.rf.2, file=paste("rf.var.acc.caret. file ",nu. file ,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep=""),
         ncol=length(var.acc.rf.2),append=TRUE,sep="\t")
159
160  #var used rf package
161  var.used.rf=randomForest::varUsed(fit.rf,count=FALSE)
162  var.acc.rf.3=var.acc(1:nu.different .genes,(nu. different .genes+1):G,var.used.rf)
163  write(var.acc.rf.3, file=paste("rf.var.acc.var.used. file ",nu. file ,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep="
         "),ncol=length(var.acc.rf.3),append=TRUE,sep="\t")
164
165  #var used rf package freq
166  var.used.rf.2=randomForest::varUsed(fit.rf,count=TRUE)
167  var.used.rf.3=which(var.used.rf.2>=31)
168  var.acc.rf.4=var.acc(1:nu.different .genes,(nu. different .genes+1):G,var.used.rf.3)
169  write(var.acc.rf.4, file=paste("rf.var.acc.var.used.freq. file ",nu. file ,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",
         sep=""),ncol=length(var.acc.rf.4),append=TRUE,sep="\t")
170
171  #predict test
172  pred.rf.class=as.numeric(predict(fit.rf,data.frame(test),type="class"))-1
173  pred.rf.prob=predict(fit.rf,data.frame(test),type="prob")[,2]
174  pas.res.rf=pas(pred.rf.class,class.test,pred.rf.prob)
175  write(pas.res.rf, file=paste("rf.pas.res. file ",nu. file ,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep=""),ncol=
         length(pas.res.rf),append=TRUE,sep="\t")
176
177  ###stumps
178  stump.seq=vector(mode="numeric",length=14)
179  stump.seq[1]=31
180  for (i in 2:14) {
181    stump.seq[i]=round(stump.seq[i-1]*1.3)
182  }
183  stump.seq[14]=1000
184
185  stumps = foreach (i=stump.seq,.packages='randomForest') %dopar% {
186     fit .stump=randomForest(as.factor(class.training)~.,data=data.frame(training),ntree=1000,importance=TRUE,maxnodes=2,
            mtry=i)
187    return(fit.stump)
188  }
189
190  vec=matrix(nrow=14,ncol=3)
191  colnames(vec)=paste(c("OOB","mtry","nnzero"))
192  for (i in 1:length(stump.seq)) {
193    vec[i,1]=as.numeric(stumps[[i]]$err.rate[1000,1])
194    vec[i,2]=stump.seq[i]
195    vec[i,3]=nnzero(stumps[[i]]$importance[,3])
196  }
197
198  #var imp stump mtry=31
199  temp.stump=which(stumps[[1]]$importance[,3]!=0)
200  var.acc.stump=var.acc(1:nu.different.genes,(nu. different .genes+1):G,as.numeric(temp.stump))
201  write(var.acc.stump,file=paste("stump.var.acc.mtry.31.file",nu.file,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep
         =""),ncol=length(var.acc.stump),append=TRUE,sep="\t")
202
203  #predict test mtry=31
204  pred.stump.class=as.numeric(predict(stumps[[1]],data.frame(test),type="class"))-1
205  pred.stump.prob=predict(stumps[[1]],data.frame(test),type="prob")[,2]
206  pas.res.stump=pas(pred.stump.class,class.test,pred.stump.prob)
207  write(pas.res.stump,file=paste("stump.pas.res.mtry.31.file",nu.file,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep
         =""),ncol=length(pas.res.stump),append=TRUE,sep="\t")
208
209  #var imp stump mtry=1000
210  temp.stump=which(stumps[[14]]$importance[,3]!=0)
211  var.acc.stump=var.acc(1:nu.different.genes,(nu. different .genes+1):G,as.numeric(temp.stump))
```

```
212  write(var.acc.stump,file=paste("stump.var.acc.mtry.1000.file",nu.file,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",
         sep=""),ncol=length(var.acc.stump),append=TRUE,sep="\t")
213
214  #predict test mtry=1000
215  pred.stump.class=as.numeric(predict(stumps[[14]],data.frame(test),type="class"))-1
216  pred.stump.prob=predict(stumps[[14]],data.frame(test),type="prob")[,2]
217  pas.res.stump=pas(pred.stump.class,class.test,pred.stump.prob)
218  write(pas.res.stump,file=paste("stump.pas.res.mtry.1000.file",nu.file,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",
         sep=""),ncol=length(pas.res.stump),append=TRUE,sep="\t")
219
220  #var imp stump mtry=minOOB
221  #find last min OOB
222  min.oob=1
223  min.oob.index=1
224  for (i in 1:nrow(vec)) {
225    if (min.oob>=vec[i,1]) {
226      min.oob=vec[i,1]
227      min.oob.index=i
228    }
229  }
230  temp.stump=which(stumps[[min.oob.index]]$importance[,3]!=0)
231  var.acc.stump=var.acc(1:nu.different.genes,(nu. different .genes+1):G,as.numeric(temp.stump))
232  var.acc.stump=c(var.acc.stump,vec[min.oob.index,2])
233  write(var.acc.stump,file=paste("stump.var.acc.mtry.min.OOB.file",nu.file,"samp",n.training,"mde",mean.difference,"rho",rho,".
         txt",sep=""),ncol=length(var.acc.stump),append=TRUE,sep="\t")
234
235  #predict test mtry=minOOB
236  pred.stump.class=as.numeric(predict(stumps[[min.oob.index]],data.frame(test),type="class"))-1
237  pred.stump.prob=predict(stumps[[min.oob.index]],data.frame(test),type="prob")[,2]
238  pas.res.stump=pas(pred.stump.class,class.test,pred.stump.prob)
239  write(pas.res.stump,file=paste("stump.pas.res.mtry.min.OOB.file",nu.file,"samp",n.training,"mde",mean.difference,"rho",rho,".
         txt",sep=""),ncol=length(pas.res.stump),append=TRUE,sep="\t")
240
241  ###RFE
242  training.2=training
243  training.3=training
244
245  vec.scaled=matrix(nrow=14,ncol=3)
246  vec.non.scaled=matrix(nrow=14,ncol=3)
247
248  rferf.scaled=vector("list",14)
249  rferf.non.scaled=vector("list",14)
250
251  rferf.scaled[[1]]=randomForest(as.factor(class.training)~.,data=data.frame(training),ntree=1000,importance=TRUE)
252  rferf.non.scaled[[1]]= rferf.scaled[[1]]
253
254  vec.scaled[1,1]=as.numeric(rferf.scaled[[1]]$err.rate[1000,1])
255  vec.scaled[1,2]=1000
256  vec.scaled[1,3]=nnzero(rferf.scaled[[1]]$importance[,3])
257  vec.non.scaled[1,1]=as.numeric(rferf.non.scaled[[1]]$err.rate[1000,1])
258  vec.non.scaled[1,2]=1000
259  vec.non.scaled[1,3]=nnzero(rferf.non.scaled[[1]]$importance[,3])
260
261  rferf.var.imp.scaled=randomForest::importance(rferf.scaled[[1]],scale=TRUE)
262  rferf.var.imp.non.scaled=randomForest::importance(rferf.non.scaled[[1]],scale=FALSE)
263
264  rfe.seq=vector(mode="numeric",length=14)
265  rfe.seq[1]=1000
266  for (i in 2:14) {
267    rfe.seq[i]=round(rfe.seq[i-1]*0.7)
268  }
269
270  j=2
271  for (i in rfe.seq[-1]) {
272    var.imp.sort.scaled=sort(rferf.var.imp.scaled[,3], index.return=TRUE,decreasing=TRUE)
273    var.imp.sort.non.scaled=sort(rferf.var.imp.non.scaled[,3], index.return=TRUE,decreasing=TRUE)
274
275    col.sub.scaled=var.imp.sort.scaled$ix[i+1:1000]
276    col.sub.non.scaled=var.imp.sort.non.scaled$ix[i+1:1000]
277
278    training.2[, col.sub.scaled]=0
279    training.3[, col.sub.non.scaled]=0
280
281    rferf.scaled[[j]]=randomForest(as.factor(class.training)~.,data=data.frame(training.2),ntree=1000,importance=TRUE)
282    rferf.non.scaled[[j]]=randomForest(as.factor(class.training)~.,data=data.frame(training.3),ntree=1000,importance=TRUE)
283
284    vec.scaled[j,1]=as.numeric(rferf.scaled[[j]]$err.rate[1000,1])
285    vec.scaled[j,2]=i
```

```
286     vec.scaled[j,3]=nnzero(rferf.scaled[[j]]$importance[,3])
287     vec.non.scaled[j,1]=as.numeric(rferf.non.scaled[[j]]$err.rate[1000,1])
288     vec.non.scaled[j,2]=i
289     vec.non.scaled[j,3]=nnzero(rferf.non.scaled[[j]]$importance[,3])
290
291     rferf.var.imp.scaled=randomForest::importance(rferf.scaled[[j]], scale=TRUE)
292     rferf.var.imp.non.scaled=randomForest::importance(rferf.non.scaled[[j]], scale=FALSE)
293
294     j=j+1
295   }
296
297   #find last min OOB scaled
298   min.oob=1
299   min.oob.index=1
300   for (i in 1:nrow(vec.scaled)) {
301     if (min.oob>=vec.scaled[i,1]) {
302       min.oob=vec.scaled[i,1]
303       min.oob.index=i
304     }
305   }
306   temp.rfe=which(rferf.scaled[[min.oob.index]]$importance[,3]!=0)
307   var.acc.rfe=var.acc(1:nu.different.genes,(nu.different.genes+1):G,as.numeric(temp.rfe))
308   write(var.acc.rfe, file=paste("rfe.scaled.var.acc. file",nu.file,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep=""),
          ncol=length(var.acc.rfe),append=TRUE,sep="\t")
309
310   #predict min OOB scaled
311   pred.rfe.class=as.numeric(predict(rferf.scaled[[min.oob.index]],data.frame(test),type="class"))-1
312   pred.rfe.prob=predict(rferf.scaled[[min.oob.index]],data.frame(test),type="prob")[,2]
313   pas.res.rfe=pas(pred.rfe.class,class.test,pred.rfe.prob)
314   write(pas.res.rfe, file=paste("rfe.scaled.pas.res. file",nu.file,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep=""),
          ncol=length(pas.res.rfe),append=TRUE,sep="\t")
315
316   #find last min OOB non scaled
317   min.oob=1
318   min.oob.index=1
319   for (i in 1:nrow(vec.scaled)) {
320     if (min.oob>=vec.non.scaled[i,1]) {
321       min.oob=vec.non.scaled[i,1]
322       min.oob.index=i
323     }
324   }
325   temp.rfe=which(rferf.non.scaled[[min.oob.index]]$importance[,3]!=0)
326   var.acc.rfe=var.acc(1:nu.different.genes,(nu.different.genes+1):G,as.numeric(temp.rfe))
327   write(var.acc.rfe, file=paste("rfe.non.scaled.var.acc. file",nu.file,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep
          =""),ncol=length(var.acc.rfe),append=TRUE,sep="\t")
328
329   #predict min OOB non scaled
330   pred.rfe.class=as.numeric(predict(rferf.scaled[[min.oob.index]],data.frame(test),type="class"))-1
331   pred.rfe.prob=predict(rferf.scaled[[min.oob.index]],data.frame(test),type="prob")[,2]
332   pas.res.rfe=pas(pred.rfe.class,class.test,pred.rfe.prob)
333   write(pas.res.rfe, file=paste("rfe.non.scaled.pas.res. file",nu.file,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep
          =""),ncol=length(pas.res.rfe),append=TRUE,sep="\t")
334
335
336   ##################glmnet
337   cvfit=cv.glmnet(x=data.matrix(training),y=as.factor(class.training),family="binomial",type.measure="class")
338   fit.glm=glmnet(x=data.matrix(training),y=as.factor(class.training),family="binomial",lambda=cvfit$lambda.min)
339
340   #var imp coef!=0
341   glm.lasso.coef=coef(cvfit,s="lambda.min",parallel=TRUE)
342   glm.lasso.coef.noint=as.numeric(glm.lasso.coef[-1,])
343   glm.lasso.coef.nzero=which(glm.lasso.coef.noint!=0)
344   var.acc.lasso=var.acc(1:nu.different.genes,(nu.different.genes+1):G,glm.lasso.coef.nzero)
345   write(var.acc.lasso, file=paste("lasso.var.acc. file",nu.file,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep=""),
          ncol=length(var.acc.lasso),append=TRUE,sep="\t")
346
347   #predict test
348   pred.lasso.class=as.numeric(predict(cvfit,newx=data.matrix(test),s="lambda.min",type="class",parallel=TRUE))
349   pred.lasso.prob=predict(cvfit,newx=data.matrix(test),s="lambda.min",type="response",parallel=TRUE)
350   pas.res.lasso=pas(pred.lasso.class,class.test,pred.lasso.prob)
351   write(pas.res.lasso, file=paste("lasso.pas.res. file",nu.file,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep=""),
          ncol=length(pas.res.lasso),append=TRUE,sep="\t")
352
353   ###ridge rfe 30% abs elimination
354   training.2=training
355   ridge.seq=vector(mode="numeric",length=14)
356   ridge.seq[1]=1000
357   ridge.list=vector("list",14)
```

```
358 vec=matrix(nrow=14,ncol=3)
359 colnames(vec)=paste(c("variables","lambda.min","cvm"))
360
361 for (i in 2:14) {
362   ridge.seq[i]=round(ridge.seq[i-1]*0.7)
363 }
364
365 ridge.list [[1]]=cv.glmnet(x=data.matrix(training),y=as.factor(class.training),family="binomial",type.measure="class",alpha
        =0,parallel=TRUE)
366
367 vec[1,1]=1000
368 vec[1,2]=ridge.list [[1]] $lambda.min
369 vec[1,3]=min(ridge.list [[1]] $cvm)
370
371 j=2
372 for (i in ridge.seq[-1]) {
373   glm.ridge.coef=coef(ridge.list[[j-1]],s="lambda.min",parallel=TRUE)
374   glm.ridge.coef.noint=as.numeric(glm.ridge.coef[-1,])
375   glm.ridge.sort=sort(abs(glm.ridge.coef.noint),decreasing=TRUE,index.return=TRUE)
376
377   col.sub=glm.ridge.sort$ix[i+1:1000]
378   training.2[,col.sub]=0
379
380   ridge.list [[j]]=cv.glmnet(x=data.matrix(training.2),y=as.factor(class.training),family="binomial",type.measure="class",
          alpha=0,parallel=TRUE)
381
382   vec[j,1]=i
383   vec[j,2]=ridge.list [[j]] $lambda.min
384   vec[j,3]=min(ridge.list [[j]] $cvm)
385   j=j+1
386 }
387
388 #find last min cvm
389 min.cvm=1
390 min.cvm.index=1
391 for (i in 1:nrow(vec)) {
392   if (min.cvm>=vec[i,3]) {
393     min.cvm=vec[i,3]
394     min.cvm.index=i
395   }
396 }
397 temp.ridge=coef(ridge.list [[min.cvm.index]],s="lambda.min",parallel=TRUE)
398 temp.ridge=which(temp.ridge[-1]!=0)
399 var.acc.ridge.rfe=var.acc(1:nu.different.genes,(nu.different.genes+1):G,temp.ridge)
400 write(var.acc.ridge.rfe,file=paste("ridge.rfe.var.acc.file",nu.file,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep
        =""),ncol=length(var.acc.ridge.rfe),append=TRUE,sep="\t")
401
402 #predict test
403 pred.ridge.rfe.class=as.numeric(predict(ridge.list[[min.cvm.index]],newx=data.matrix(test),s="lambda.min",type="class",
        parallel=TRUE))
404 pred.ridge.rfe.prob=predict(ridge.list[[min.cvm.index]],newx=data.matrix(test),s="lambda.min",type="response",parallel=
        TRUE)
405 pas.res.ridge.rfe=pas(pred.ridge.rfe.class,class.test,pred.ridge.rfe.prob)
406 write(pas.res.ridge.rfe,file=paste("ridge.rfe.pas.res.file",nu.file,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep
        =""),ncol=length(pas.res.ridge.rfe),append=TRUE,sep="\t")
407
408 ###elastic
409 elas.seq=seq(0.05,0.95,0.05)
410 elas.glm = foreach (i=elas.seq,.packages='glmnet') %dopar% {
411   fit.elas=cv.glmnet(x=data.matrix(training),y=as.factor(class.training),family="binomial",type.measure="class",alpha=i)
412   return(fit.elas)
413 }
414
415 vec=matrix(nrow=20,ncol=3)
416 colnames(vec)=paste(c("alpha","lambda.min","cvm"))
417 for (i in 1:length(elas.seq)) {
418   vec[i,1]=elas.seq[i]
419   vec[i,2]=elas.glm[[i]] $lambda.min
420   vec[i,3]=min(elas.glm[[i]] $cvm)
421 }
422
423 #var imp min cvm
424 temp.elas=which.min(vec[,3])
425 glm.elas.coef=coef(elas.glm[[temp.elas]],s="lambda.min",parallel=TRUE)
426 glm.elas.coef.noint=as.numeric(glm.elas.coef[-1,])
427 glm.elas.coef.nzero=which(glm.elas.coef.noint!=0)
428 var.acc.elas=var.acc(1:nu.different.genes,(nu.different.genes+1):G,glm.elas.coef.nzero)
429 var.acc.elas=c(var.acc.elas,vec[temp.elas,1])
```

```
write(var.acc.elas, file =paste("elastic.var.acc. file ",nu. file ,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep=""),
      ncol=length(var.acc.elas),append=TRUE,sep="\t")

#predict test
pred.elas.class=as.numeric(predict(elas.glm[[temp.elas]],newx=data.matrix(test),s="lambda.min",type="class",parallel=
      TRUE))
pred.elas.prob=predict(elas.glm[[temp.elas]],newx=data.matrix(test),s="lambda.min",type="response",parallel=TRUE)
pas.res.elas=pas(pred.elas.class,class.test,pred.elas.prob)
write(pas.res.elas, file =paste("elastic.pas.res. file ",nu. file ,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep=""),
      ncol=length(pas.res.elas),append=TRUE,sep="\t")

###bolasso
training.2=matrix(nrow=n.training,ncol=1000)
class.training.2=vector(mode="numeric",length=n.training)

bolasso = foreach (i=1:100,.packages='glmnet') %dopar% {
  rand.seq=sample(1:n.training,n.training,replace=T)
  for (j in 1:n.training) {
    training.2[j,]=training[rand.seq[j],]
    class.training.2[j]=class.training[rand.seq[j]]
  }
  fit.bola=cv.glmnet(x=data.matrix(training.2),y=as.factor(class.training.2),family="binomial",type.measure="class")
  return(fit.bola)
}

temp.bola=vector()
freq.table=matrix(rep(0,len=1000),nrow=1000,ncol=1)
for (i in 1:100) {
  temp.hold=coef(bolasso[[i]],lambda=bolasso[[i]]$lambda.min,parallel=TRUE)
  temp.hold=which(temp.hold[-1]!=0)
  if (length(temp.hold)>=1) {
    for (k in 1:length(temp.hold)) {
      freq.table[temp.hold[k],1]=freq.table[temp.hold[k],1]+1
    }
  }
  temp.bola=c(temp.bola,temp.hold)
}
temp.bola=which(freq.table>=5)
var.acc.bolasso=var.acc(1:nu.different.genes,(nu. different .genes+1):G,as.numeric(temp.bola))
write(var.acc.bolasso, file =paste("bolasso.var.acc. file ",nu. file ,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep=""
      ),ncol=length(var.acc.bolasso),append=TRUE,sep="\t")

temp.bola.2=which(freq.table!=0)
var.acc.bolasso.2=var.acc(1:nu.different.genes,(nu. different .genes+1):G,as.numeric(temp.bola.2))
write(var.acc.bolasso.2, file =paste("bolasso.var.acc.all. file ",nu. file ,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",
      sep=""),ncol=length(var.acc.bolasso.2),append=TRUE,sep="\t")

#predict test
temp.hold=seq(1,1000,1)
temp.hold=temp.hold[!temp.hold %in% temp.bola]
training.2=training
training.2[, temp.hold]=0
bolasso.2=cv.glmnet(x=data.matrix(training.2),y=as.factor(class.training),family="binomial",type.measure="class",parallel=
      TRUE)

pred.bolasso.class=as.numeric(predict(bolasso.2,newx=data.matrix(test),s="lambda.min",type="class",parallel=TRUE))
pred.bolasso.prob=predict(bolasso.2,newx=data.matrix(test),s="lambda.min",type="response",parallel=TRUE)
pas.res.bolasso=pas(pred.bolasso.class,class.test,pred.bolasso.prob)
write(pas.res.bolasso, file =paste("bolasso.pas.res. file ",nu. file ,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep=""
      ),ncol=length(pas.res.bolasso),append=TRUE,sep="\t")

###boelastic alpha=0.05
training.2=matrix(nrow=n.training,ncol=1000)
class.training.2=vector(mode="numeric",length=n.training)

boelastic = foreach (i=1:100,.packages='glmnet') %dopar% {
  rand.seq=sample(1:n.training,n.training,replace=T)
  for (j in 1:n.training) {
    training.2[j,]=training[rand.seq[j],]
    class.training.2[j]=class.training[rand.seq[j]]
  }
  fit.boelas=cv.glmnet(x=data.matrix(training.2),y=as.factor(class.training.2),family="binomial",type.measure="class",alpha
      =0.05)
  return(fit.boelas)
}

freq.table=matrix(rep(0,len=1000),nrow=1000,ncol=1)
for (i in 1:100) {
```

```
    glm.elas.coef=coef(boelastic[[i ]], s="lambda.min",parallel=TRUE)
    glm.elas.coef.noint=as.numeric(glm.elas.coef[-1,])
    glm.elas.coef.nzero=which(glm.elas.coef.noint!=0)
    for (j in 1:length(glm.elas.coef.nzero)) {
      freq.table[glm.elas.coef.nzero[j],1]=freq.table[glm.elas.coef.nzero[j],1]+1
    }
  }

  #var imp
  temp.boelas=which(freq.table>=50)
  var.acc.boelas=var.acc(1:nu.different.genes,(nu. different .genes+1):G,as.numeric(temp.boelas))
  write(var.acc.boelas,file=paste("boelastic.var.acc.freq.50. file",nu. file ,"samp",n.training,"mde",mean.difference,"rho",rho,".txt
      ",sep=""),ncol=length(var.acc.boelas),append=TRUE,sep="\t")
  temp.boelas.2=which(freq.table>=75)
  var.acc.boelas.2=var.acc(1:nu.different.genes,(nu. different .genes+1):G,as.numeric(temp.boelas.2))
  write(var.acc.boelas.2, file=paste("boelastic.var.acc.freq.75. file",nu. file ,"samp",n.training,"mde",mean.difference,"rho",rho,".
      txt",sep=""),ncol=length(var.acc.boelas.2),append=TRUE,sep="\t")

  #predict test freq 50
  temp.hold=seq(1,1000,1)
  temp.hold=temp.hold[!temp.hold %in% temp.boelas]
  training.2=training
  training .2[, temp.hold]=0
  boelastic.2=cv.glmnet(x=data.matrix(training.2),y=as.factor(class.training),family="binomial",type.measure="class",alpha
      =0.05)

  pred.boelas.class=as.numeric(predict(boelastic.2,newx=data.matrix(test),s="lambda.min",type="class",parallel=TRUE))
  pred.boelas.prob=predict(boelastic.2,newx=data.matrix(test),s="lambda.min",type="response",parallel=TRUE)
  pas.res.boelas=pas(pred.boelas.class,class.test,pred.boelas.prob)
  write(pas.res.boelas, file=paste("boelastic.pas.res.freq.50. file",nu. file ,"samp",n.training,"mde",mean.difference,"rho",rho,".txt
      ",sep=""),ncol=length(pas.res.boelas),append=TRUE,sep="\t")

  #predict test freq 75
  temp.hold=seq(1,1000,1)
  temp.hold=temp.hold[!temp.hold %in% temp.boelas.2]
  training.2=training
  training .2[, temp.hold]=0
  boelastic.2=cv.glmnet(x=data.matrix(training.2),y=as.factor(class.training),family="binomial",type.measure="class",alpha
      =0.05)

  pred.boelas.class=as.numeric(predict(boelastic.2,newx=data.matrix(test),s="lambda.min",type="class",parallel=TRUE))
  pred.boelas.prob=predict(boelastic.2,newx=data.matrix(test),s="lambda.min",type="response",parallel=TRUE)
  pas.res.boelas.2=pas(pred.boelas.class,class.test,pred.boelas.prob)
  write(pas.res.boelas.2, file=paste("boelastic.pas.res.freq.75. file",nu. file ,"samp",n.training,"mde",mean.difference,"rho",rho,".
      txt",sep=""),ncol=length(pas.res.boelas.2),append=TRUE,sep="\t")


  ##################pamr
  mydata=list(x=t(training),y=factor(class.training),geneid=as.character(1:nrow(t(training))),genenames=paste("g",as.
      character(1:nrow(t(training))),sep=""))
  pamfit=pamr.train(mydata)
  pamcv=pamr.cv(pamfit,mydata)

  #find last min threshold
  min.cv=Inf
  min.cv.index=1
  for (i in 1:length(pamcv$error)) {
    if (min.cv>=pamcv$error[i]) {
      min.cv=pamcv$error[i]
      min.cv.index=i
    }
  }
  thr=pamcv$threshold[min.cv.index]

  #var imp threshold
  var.used.pamr=pamr.predict(pamfit,newx=NULL,threshold=thr,type="nonzero")
  var.acc.pamr=var.acc(1:nu.different.genes,(nu. different .genes+1):G,var.used.pamr)
  write(var.acc.pamr,file=paste("pam.var.acc.file",nu.file,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep=""),ncol=
      length(var.acc.pamr),append=TRUE,sep="\t")

  #predict test
  pred.pamr.class=pamr.predict(pamfit,t(test),type="class",threshold=thr)
  pred.pamr.prob=pamr.predict(pamfit,t(test),type="posterior",threshold=thr)[,2]
  pas.res.pamr=pas(pred.pamr.class,class.test,pred.pamr.prob)
  write(pas.res.pamr,file=paste("pam.pas.res.file",nu. file ,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep=""),ncol=
      length(pas.res.pamr),append=TRUE,sep="\t")

  ###bopamr
```

```
  training.2=matrix(nrow=n.training,ncol=1000)
  class.training.2=vector(mode="numeric",length=n.training)

  bopamr = foreach (i=1:100,.packages='pamr') %dopar% {
    rand.seq=sample(1:n.training,n.training,replace=T)
    for (j in 1:n.training) {
      training.2[,j,]=training[rand.seq[j],]
      class.training.2[j]=class.training[rand.seq[j]]
    }
    mydata.2=list(x=t(training.2),y=factor(class.training.2),geneid=as.character(1:nrow(t(training.2))),genenames=paste("g",as
        .character(1:nrow(t(training.2))),sep=""))
    pamfit=pamr.train(mydata.2)
    pamcv=pamr.cv(pamfit,mydata.2)
    return(list(pamfit,pamcv))
  }

  vec=matrix(nrow=100,ncol=3)
  for (i in 1:100) {
    min.cv=Inf
    min.cv.index=1
    for (j in 1:length(bopamr[[i]][[2]]$error)) {
      if (min.cv>=bopamr[[i]][[2]]$error[j]) {
        min.cv=bopamr[[i]][[2]]$error[j]
        min.cv.index=j
      }
    }
    vec[i,1]=bopamr[[i]][[2]]$threshold[min.cv.index]
    vec[i,2]=bopamr[[i]][[2]]$size[min.cv.index]
    vec[i,3]=bopamr[[i]][[2]]$error[min.cv.index]
  }
  temp.hold=vec[which.min(vec[,3]),3]
  temp.hold.2=which(vec[,3]==temp.hold)
  temp.hold.3=which.min(vec[temp.hold.2,2])
  temp.index=temp.hold.2[temp.hold.3]
  thr=vec[temp.index,1]

  #var imp threshold
  var.used.pamr=pamr.predict(bopamr[[temp.index]][[1]],newx=NULL,threshold=thr,type="nonzero")
  var.acc.pamr=var.acc(1:nu.different.genes,(nu.different.genes+1):G,var.used.pamr)
  write(var.acc.pamr,file=paste("bopam.var.acc.file",nu.file,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep=""),
      ncol=length(var.acc.pamr),append=TRUE,sep="\t")

  #predict test
  temp.hold=seq(1,1000,1)
  temp.hold=temp.hold[!temp.hold %in% var.used.pamr]
  training.2=training
  training.2[, temp.hold]=rnorm(n.training,0,1e-32)
  mydata.3=list(x=t(training.2),y=factor(class.training),geneid=as.character(1:nrow(t(training.2))),genenames=paste("g",as.
      character(1:nrow(t(training.2))),sep=""))
  pamfit.2=pamr.train(mydata.3)
  pamcv.2=pamr.cv(pamfit.2,mydata.3)

  #find last min threshold
  min.cv=Inf
  min.cv.index=1
  for (i in 1:length(pamcv.2$error)) {
    if (min.cv>=pamcv.2$error[i]) {
      min.cv=pamcv.2$error[i]
      min.cv.index=i
    }
  }
  thr=pamcv.2$threshold[min.cv.index]

  pred.bopamr.class=pamr.predict(pamfit.2,t(test),type="class",threshold=thr)
  pred.bopamr.prob=pamr.predict(pamfit.2,t(test),type="posterior",threshold=thr)[,2]
  pas.res.bopamr=pas(pred.bopamr.class,class.test,pred.bopamr.prob)
  write(pas.res.bopamr,file=paste("bopam.pas.res.file",nu.file,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep=""),
      ncol=length(pas.res.bopamr),append=TRUE,sep="\t")


}

#main
cl=makeCluster(detectCores(),type='PSOCK')
registerDoParallel(cl)

B=50 #number of repetition of the simulation

```

```
643 for(m in c(100,50,250)) { #sample size
644   for (n in c(1,0.5,0) ) { #mean difference
645     for (p in c(0.8,0.5,0) ) { #rho
646       for (w in 1:B) {
647         fun(nu.file="1",G=1000,n.training=m,n.test=1000,percent.class.training=0.5,percent.class.test=0.5,nu.different.genes=100,
                mean.difference=n,num.blocks=100,rho=p)
648       }
649     }
650   }
651 }
652
653 stopCluster(cl)
```

D R koda za izbrane simulacije

```
1  library(class)
2  library(Hmisc)
3  library(randomForest)
4  library(pamr)
5  library(glmnet)
6  library(caret)
7
8  pas=function(fit,ctest, fit .prob) {
9    PA=(length(which(fit==0&ctest==0))+length(which(fit==1&ctest==1)))/length(fit)
10
11   pa1=length(which(fit==0&ctest==0))/length(which(ctest==0))
12   pa2=length(which(fit==1&ctest==1))/length(which(ctest==1))
13
14   gm=sqrt(pa1*pa2)
15
16   AUC=somers2(fit.prob,ctest)["C"]
17
18   c(PA,pa1,pa2,gm,AUC)
19 }
20
21 var.acc=function(true.dif,true.nd,fit . dif) {
22   perc.cor=sum(fit.dif%in%true.dif)/length(true.dif)
23   perc. false =sum(fit.dif%in%true.nd)/length(true.nd)
24
25   pv.cor.1=sum(fit.dif%in%true.dif)
26   pv.cor.2=length(fit.dif)
27   if (pv.cor.2==0) {
28     pv.cor="NaN"
29   } else {
30     pv.cor=pv.cor.1/pv.cor.2
31   }
32
33   fdr.1=sum(fit.dif%in%true.nd)
34   fdr.2=length(fit.dif)
35   if (fdr.2==0) {
36     fdr="NaN"
37   } else {
38     fdr=fdr.1/fdr.2
39   }
40
41   c(perc.cor,perc. false ,pv.cor,fdr)
42 }
43
44 fun=function(nu.file,G,n.training,n.test ,percent.class. training ,percent.class. test ,nu. different .genes,mean.difference,num.blocks,
       rho) {
45   n=n.training+n.test
46
47   n. training .class.1=n.training*percent.class. training
48   n. training .class.2=n.training-n.training*percent.class.training
49
50   n. test .class.1=n.test*percent.class. test
51   n. test .class.2=n.test-n.test*percent.class. test
52
53   num.genes=G
54   var.sigma.block=rep(1,num.genes)
55   mu.Class1=rep(0,num.genes)
56   mu.Class2=c(rep(mean.difference,nu.different.genes),rep(0,num.genes-nu.different.genes))
57   num.samples=n.training
58   n1=n.training.class.1
59   n2=n.training.class.2
60
61   #number of genes within each block
62   #NB! implies that the blocks have the same number of genes
63   #if more flexibility is needed can be changed
64   num.genes.per.block<-num.genes/num.blocks
```

```
65
66  #standard deviation for each gene
67  sd.genes<-rep(sqrt(var.sigma.block), each=num.genes.per.block)
68
69  #desired correlation between genes within each block, obtained from the specified correlation common to each block,
70  #can be made more flexible
71  cor.block<-rep(rho, num.blocks)
72
73  #sqrt of variance of the random effect to obtain a correlation of cor.block within each block
74  sd.RE<-sqrt(var.sigma.block*cor.block)
75
76
77  #standard deviation for each gene
78    sd.genes<-rep(sqrt(var.sigma.block-sd.RE^2), each=num.genes.per.block)
79
80  #generation of gene expression data
81
82  #data from all the genes for samples of Class1
83  data.Class1<-matrix(rnorm(num.genes*n1, mu.Class1, sd.genes),
84  ncol=num.genes, nrow=n1, byrow=T)
85
86  #data from all the genes for samples of Class1
87  data.Class2<-matrix(rnorm(num.genes*n2, mu.Class2, sd.genes),
88  ncol=num.genes, nrow=n2, byrow=T)
89
90  #implies that the blocks have the same number of genes
91
92  #simulated RE for each of the blocks, block-specific, the same for all the subjects
93  #sim.RE<-rnorm(num.blocks, 0, sd.RE)
94
95  tmp<-lapply(1:num.blocks, function(x) {
96    x1<-(x-1)*num.genes.per.block+1;
97    x2<- x*num.genes.per.block;
98    data.Class1[1:n1,x1:x2]<<-data.Class1[1:n1, x1:x2]+rnorm(n1, 0, sd.RE[x]);
99    data.Class2[1:n2, x1:x2]<<-data.Class2[1:n2, x1:x2]+rnorm(n2, 0, sd.RE[x])
100 }
101 )
102
103 #full matrix of data, rows: samples, cols: genes
104
105 training<-rbind(data.Class1, data.Class2)
106
107 ###data generation for test
108
109 num.samples=n.test
110 n1=n.test.class.1
111 n2=n.test.class.2
112
113 data.Class1<-matrix(rnorm(num.genes*n1, mu.Class1, sd.genes),
114 ncol=num.genes, nrow=n1, byrow=T)
115
116 #data from all the genes for samples of Class1
117 data.Class2<-matrix(rnorm(num.genes*n2, mu.Class2, sd.genes),
118 ncol=num.genes, nrow=n2, byrow=T)
119
120 #implies that the blocks have the same number of genes
121
122 #simulated RE for each of the blocks, block-specific, the same for all the subjects
123 #sim.RE<-rnorm(num.blocks, 0, sd.RE)
124
125 tmp<-lapply(1:num.blocks, function(x) {
126   x1<-(x-1)*num.genes.per.block+1;
127   x2<- x*num.genes.per.block;
128   data.Class1[1:n1,x1:x2]<<-data.Class1[1:n1, x1:x2]+rnorm(n1, 0, sd.RE[x]);
129   data.Class2[1:n2, x1:x2]<<-data.Class2[1:n2, x1:x2]+rnorm(n2, 0, sd.RE[x])
130 }
131 )
132
133 #full matrix of data, rows: samples, cols: genes
134
135 test<-rbind(data.Class1, data.Class2)
136
137 class.training=c(rep(0,n.training.class.1),rep(1,n.training.class.2))
138 class.test=c(rep(0,n.test.class.1),rep(1,n.test.class.2))
139
140
141 ###filter-caret
142  filter .caret=filterVarImp(x=data.frame(training),y=as.factor(class.training))
```

```
143 temp.sort=which(filter.caret[,2]>=2/3)
144 var.acc. filter .caret=var.acc(1:nu.different .genes,(nu. different .genes+1):G,as.numeric(temp.sort))
145 write(var.acc. filter .caret, file=paste("filter.caret.var.acc. file .",nu. file ,"samp",n.training,"mde",mean.difference,"rho",rho,".
        txt",sep=""),ncol=length(var.acc.filter.caret),append=TRUE,sep="\t")
146
147
148 ###randomForest
149 fit .rf=randomForest(as.factor(class.training)~.,data=data.frame(training),ntree=1000,importance=TRUE)
150
151 #var used rf package freq
152 var.used.rf.2=randomForest::varUsed(fit.rf,count=TRUE)
153 var.used.rf.3=which(var.used.rf.2>=31)
154 var.acc.rf.4=var.acc(1:nu.different .genes,(nu. different .genes+1):G,var.used.rf.3)
155 write(var.acc.rf.4, file=paste("rf.var.acc.freq. file ",nu. file ,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep=""),
        ncol=length(var.acc.rf.4),append=TRUE,sep="\t")
156
157 if (length(var.used.rf.3)>0) {
158 #predict test
159 temp.hold=seq(1,1000,1)
160 temp.hold=temp.hold[!temp.hold %in% var.used.rf.3]
161 training.2=training
162 training .2[, temp.hold]=0
163 fit .rf2=randomForest(as.factor(class.training)~.,data=data.frame(training.2),ntree=1000,importance=TRUE)
164
165 pred.rf.class=as.numeric(predict(fit.rf2,data.frame(test),type="class"))-1
166 pred.rf.prob=predict(fit.rf2,data.frame(test),type="prob")[,2]
167 pas.res.rf=pas(pred.rf.class,class.test,pred.rf.prob)
168 write(pas.res.rf, file=paste("rf.pas.res.freq. file ",nu. file ,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep=""),
        ncol=length(pas.res.rf),append=TRUE,sep="\t")
169 } else {
170 pas.res.rf=c("NaN","NaN","NaN","NaN","NaN")
171 write(pas.res.rf, file=paste("rf.pas.res.freq. file ",nu. file ,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep=""),
        ncol=length(pas.res.rf),append=TRUE,sep="\t")
172 }
173
174 ###stumps
175 fit .stump=randomForest(as.factor(class.training)~.,data=data.frame(training),ntree=1000,importance=TRUE,maxnodes=2,
        mtry=1000)
176
177 #var imp
178 temp.stump=which(fit.stump$importance[,3]!=0)
179 var.acc.stump=var.acc(1:nu.different.genes,(nu. different .genes+1):G,as.numeric(temp.stump))
180 write(var.acc.stump,file=paste("stump.var.acc.file",nu.file,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep=""),
        ncol=length(var.acc.stump),append=TRUE,sep="\t")
181
182 if (length(temp.stump)>0) {
183 #predict test
184 temp.hold=seq(1,1000,1)
185 temp.hold=temp.hold[!temp.hold %in% temp.stump]
186 training.2=training
187 training .2[, temp.hold]=0
188 fit .stump2=randomForest(as.factor(class.training)~.,data=data.frame(training.2),ntree=1000)
189
190 pred.stump.class=as.numeric(predict(fit.stump2,data.frame(test),type="class"))-1
191 pred.stump.prob=predict(fit.stump2,data.frame(test),type="prob")[,2]
192 pas.res.stump=pas(pred.stump.class,class.test,pred.stump.prob)
193 write(pas.res.stump,file=paste("stump.pas.res.file",nu. file ,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep=""),
        ncol=length(pas.res.stump),append=TRUE,sep="\t")
194 } else {
195 pas.res.stump=c("NaN","NaN","NaN","NaN","NaN")
196 write(pas.res.stump,file=paste("stump.pas.res.file",nu. file ,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep=""),
        ncol=length(pas.res.stump),append=TRUE,sep="\t")
197 }
198
199 #var imp freq
200 temp.stump=which(randomForest::varUsed(fit.stump,count=TRUE)>=5)
201 var.acc.stump=var.acc(1:nu.different.genes,(nu. different .genes+1):G,as.numeric(temp.stump))
202 write(var.acc.stump,file=paste("stump.var.acc.freq.file",nu. file ,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep="
        "),ncol=length(var.acc.stump),append=TRUE,sep="\t")
203
204 if (length(temp.stump)>0) {
205 #predict test freq
206 temp.hold=seq(1,1000,1)
207 temp.hold=temp.hold[!temp.hold %in% temp.stump]
208 training.2=training
209 training .2[, temp.hold]=0
210 fit .stump2=randomForest(as.factor(class.training)~.,data=data.frame(training.2),ntree=1000)
211
```

```
pred.stump.class=as.numeric(predict(fit.stump2,data.frame(test),type="class"))-1
pred.stump.prob=predict(fit.stump2,data.frame(test),type="prob")[,2]
pas.res.stump=pas(pred.stump.class,class.test,pred.stump.prob)
write(pas.res.stump,file=paste("stump.pas.res.freq.file",nu.file,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep="
    "),ncol=length(pas.res.stump),append=TRUE,sep="\t")
} else {
pas.res.stump=c("NaN","NaN","NaN","NaN","NaN")
write(pas.res.stump,file=paste("stump.pas.res.freq.file",nu.file,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep="
    "),ncol=length(pas.res.stump),append=TRUE,sep="\t")
}


###glmnet
cvfit =cv.glmnet(x=data.matrix(training),y=as.factor(class.training),family="binomial",type.measure="class")

#var imp coef!=0
glm.lasso.coef=coef(cvfit,s="lambda.min")
glm.lasso.coef.noint=as.numeric(glm.lasso.coef[-1,])
glm.lasso.coef.nzero=which(glm.lasso.coef.noint!=0)
var.acc.lasso=var.acc(1:nu.different.genes,(nu.different.genes+1):G,glm.lasso.coef.nzero)
write(var.acc.lasso,file=paste("lasso.var.acc.file",nu.file,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep=""),
    ncol=length(var.acc.lasso),append=TRUE,sep="\t")

#predict test
pred.lasso.class=as.numeric(predict(cvfit,newx=data.matrix(test),s="lambda.min",type="class"))
pred.lasso.prob=predict(cvfit,newx=data.matrix(test),s="lambda.min",type="response")
pas.res.lasso=pas(pred.lasso.class,class.test,pred.lasso.prob)
write(pas.res.lasso,file=paste("lasso.pas.res.file",nu.file,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep=""),
    ncol=length(pas.res.lasso),append=TRUE,sep="\t")


###elastic
fit.elas=cv.glmnet(x=data.matrix(training),y=as.factor(class.training),family="binomial",type.measure="class",alpha=0.5)

#var imp min cvm
glm.elas.coef=coef(fit.elas,s="lambda.min")
glm.elas.coef.noint=as.numeric(glm.elas.coef[-1,])
glm.elas.coef.nzero=which(glm.elas.coef.noint!=0)
var.acc.elas=var.acc(1:nu.different.genes,(nu.different.genes+1):G,glm.elas.coef.nzero)
write(var.acc.elas,file=paste("elastic.var.acc.file",nu.file,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep=""),
    ncol=length(var.acc.elas),append=TRUE,sep="\t")

#predict test
pred.elas.class=as.numeric(predict(fit.elas,newx=data.matrix(test),s="lambda.min",type="class"))
pred.elas.prob=predict(fit.elas,newx=data.matrix(test),s="lambda.min",type="response")
pas.res.elas=pas(pred.elas.class,class.test,pred.elas.prob)
write(pas.res.elas,file=paste("elastic.pas.res.file",nu.file,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep=""),
    ncol=length(pas.res.elas),append=TRUE,sep="\t")


###pamr
mydata=list(x=t(training),y=factor(class.training),geneid=as.character(1:nrow(t(training))),genenames=paste("g",as.
    character(1:nrow(t(training))),sep=""))
pamfit=pamr.train(mydata)
pamcv=pamr.cv(pamfit,mydata)

#find last min threshold
min.cv=Inf
min.index=1
for (i in 1:length(pamcv$error)) {
  if (min.cv>=pamcv$error[i]) {
    min.cv=pamcv$error[i]
    min.index=i
  }
}
thr=pamcv$threshold[min.index]

#var imp threshold
var.used.pamr=pamr.predict(pamfit,newx=NULL,threshold=thr,type="nonzero")
var.acc.pamr=var.acc(1:nu.different.genes,(nu.different.genes+1):G,var.used.pamr)
write(var.acc.pamr,file=paste("pam.var.acc.file",nu.file,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep=""),ncol=
    length(var.acc.pamr),append=TRUE,sep="\t")

#predict test
pred.pamr.class=pamr.predict(pamfit,t(test),type="class",threshold=thr)
pred.pamr.prob=pamr.predict(pamfit,t(test),type="posterior",threshold=thr)[,2]
pas.res.pamr=pas(pred.pamr.class,class.test,pred.pamr.prob)
```

```
  write(pas.res.pamr,file=paste("pam.pas.res.file",nu.file,"samp",n.training,"mde",mean.difference,"rho",rho,".txt",sep=""),ncol=
      length(pas.res.pamr),append=TRUE,sep="\t")


}

#main
B=50 #number of repetition of the simulation

for(m in c(50,100,250)) { #sample size
  for (n in seq(0,2,0.25)) { #mean difference
    for (p in seq(0,1,0.2)) { #rho
      for (w in 1:B) {
        fun(nu.file="31",G=1000,n.training=m,n.test=1000,percent.class.training=0.5,percent.class.test=0.5,nu.different.genes
            =100,mean.difference=n,num.blocks=100,rho=p)
      }
    }
  }
}
```

E R koda za prave podatke

```
library(class)
library(Hmisc)
library(randomForest)
library(pamr)
library(glmnet)
library(caret)
library(doParallel)
library(foreach)

cl=makeCluster(detectCores(),type='PSOCK')
registerDoParallel(cl)

pas=function(fit,ctest, fit.prob) {
  PA=(length(which(fit==0&ctest==0))+length(which(fit==1&ctest==1)))/length(fit)

  pa1=length(which(fit==0&ctest==0))/length(which(ctest==0))
  pa2=length(which(fit==1&ctest==1))/length(which(ctest==1))

  gm=sqrt(pa1*pa2)

  AUC=somers2(fit.prob,ctest)["C"]

  c(PA,pa1,pa2,gm,AUC)
}

source("codeGetReal.R")
training=data.nona

for (i in c("ER","grade")) {
  if (i=="ER") {class.training=ER} else {class.training=grade}

  ###filter-caret
  filter.caret=filterVarImp(x=data.frame(training),y=as.factor(class.training))
  temp.sort=which(filter.caret[,2]>=2/3)
  write(temp.sort,file=paste("filter.caret.",i,".txt",sep=""),ncol=length(temp.sort),append=TRUE,sep="\t")
  print(temp.sort)


  ###randomForest
  fit.rf=randomForest(as.factor(class.training)~.,data=data.frame(training),ntree=1000,importance=TRUE)

  #var used rf package freq
  var.used.rf.2=varUsed(fit.rf,count=TRUE)
  var.used.rf.3=which(var.used.rf.2>=31)
  write(var.used.rf.3, file=paste("rf.var.used.freq.",i,".txt",sep=""),ncol=length(var.used.rf.3),append=TRUE,sep="\t")

  #predict test
  temp.hold=seq(1,1000,1)
  temp.hold=temp.hold[!temp.hold %in% var.used.rf.3]
  training.2=training
  training.2[, temp.hold]=0
  temp.avg=vector("list",250)

  temp.avg = foreach (i=1:250,.packages=c('Hmisc','randomForest','caret'),.combine=rbind) %dopar% {
    temp.avg[[i]]=0
    flds=createFolds(class.training,k=5,list=TRUE,returnTrain=FALSE)
    for (j in 1:5) {
      temp.hold=seq(1,99,1)
      temp.hold=temp.hold[!temp.hold %in% flds[[j]]]
      fit.rf=randomForest(as.factor(class.training[temp.hold])~.,data=data.frame(training.2[temp.hold,]),ntree=1000,importance
          =FALSE)

      pred.rf.class=as.numeric(predict(fit.rf,data.frame(training.2[flds[[j]],]),type="class"))-1
      pred.rf.prob=predict(fit.rf,data.frame(training.2[flds[[j]],]),type="prob")[,2]
      pas.res.rf=pas(pred.rf.class,class.training[flds[[j]]], pred.rf.prob)
```

```
65
66          temp.avg[[i]]=temp.avg[[i]]+pas.res.rf
67        }
68        return(temp.avg[[i]]/5)
69      }
70      temp.avg2=colMeans(temp.avg[,1:5])
71      write(temp.avg2,file=paste("rf.pas.res.",i,".txt",sep=""),ncol=length(temp.avg2),append=TRUE,sep="\t")
72
73
74      ###stumps
75      fit.stump=randomForest(as.factor(class.training)~.,data=data.frame(training),ntree=1000,importance=TRUE,maxnodes=2,
            mtry=1000)
76
77      #var imp stump mtry=1000
78      temp.stump=which(fit.stump$importance[,3]!=0)
79      write(temp.stump,file=paste("stump.var.used.",i,".txt",sep=""),ncol=length(temp.stump),append=TRUE,sep="\t")
80
81      temp.stump=which(varUsed(fit.stump,count=TRUE)>=5)
82      write(temp.stump,file=paste("stump.var.used.freq.",i,".txt",sep=""),ncol=length(temp.stump),append=TRUE,sep="\t")
83
84      #predict test mtry=1000
85      temp.hold=seq(1,1000,1)
86      temp.hold=temp.hold[!temp.hold %in% temp.stump]
87      training.2=training
88      training.2[, temp.hold]=0
89      temp.avg=vector("list",250)
90
91      temp.avg = foreach (i=1:250,.packages=c('Hmisc','randomForest','caret'),.combine=rbind) %dopar% {
92        temp.avg[[i]]=0
93        flds=createFolds(class.training,k=5,list=TRUE,returnTrain=FALSE)
94        for (j in 1:5) {
95          temp.hold=seq(1,99,1)
96          temp.hold=temp.hold[!temp.hold %in% flds[[j]]]
97          fit.stump=randomForest(as.factor(class.training[temp.hold])~.,data=data.frame(training.2[temp.hold,]),ntree=1000,
                importance=FALSE,maxnodes=2,mtry=1000)
98
99          pred.stump.class=as.numeric(predict(fit.stump,data.frame(training.2[flds[[j]],]),type="class"))-1
100         pred.stump.prob=predict(fit.stump,data.frame(training.2[flds[[j]],]),type="prob")[,2]
101         pas.res.stump=pas(pred.stump.class,class.training[flds[[j ]]], pred.stump.prob)
102
103         temp.avg[[i]]=temp.avg[[i]]+pas.res.stump
104       }
105       return(temp.avg[[i]]/5)
106     }
107     temp.avg2=colMeans(temp.avg[,1:5])
108     write(temp.avg2,file=paste("stump.pas.res.mtry.1000.",i,".txt",sep=""),ncol=length(temp.avg2),append=TRUE,sep="\t")
109
110
111     ###glmnet
112     cvfit=cv.glmnet(x=data.matrix(training),y=as.factor(class.training),family="binomial",type.measure="class")
113
114     #var imp coef!=0
115     glm.lasso.coef=coef(cvfit,s="lambda.min")
116     glm.lasso.coef.noint=as.numeric(glm.lasso.coef[-1,])
117     glm.lasso.coef.nzero=which(glm.lasso.coef.noint!=0)
118     write(glm.lasso.coef.nzero,file=paste("lasso.var.used.",i,".txt",sep=""),ncol=length(glm.lasso.coef.nzero),append=TRUE,sep
            ="\t")
119
120     #predict test
121     temp.hold=seq(1,1000,1)
122     temp.hold=temp.hold[!temp.hold %in% glm.lasso.coef.nzero]
123     training.2=training
124     training.2[, temp.hold]=0
125     temp.avg=vector("list",250)
126
127     temp.avg = foreach (i=1:250,.packages=c('Hmisc','glmnet','caret'),.combine=rbind) %dopar% {
128       temp.avg[[i]]=0
129       flds=createFolds(class.training,k=5,list=TRUE,returnTrain=FALSE)
130       for (j in 1:5) {
131         temp.hold=seq(1,99,1)
132         temp.hold=temp.hold[!temp.hold %in% flds[[j]]]
133         cvfit=cv.glmnet(x=training.2[temp.hold,],y=as.factor(class.training[temp.hold]),family="binomial",type.measure="class")
134
135         pred.lasso.class=as.numeric(predict(cvfit,training.2[flds[[j ]],], s="lambda.min",type="class"))
136         pred.lasso.prob=predict(cvfit,training.2[flds[[j ]],], s="lambda.min",type="response")
137         pas.res.lasso=pas(pred.lasso.class,class.training[flds[[j ]]], pred.lasso.prob)
138
139         temp.avg[[i]]=temp.avg[[i]]+pas.res.lasso
```

```
140     }
141     return(temp.avg[[i]]/5)
142   }
143   temp.avg2=colMeans(temp.avg[,1:5])
144   write(temp.avg2,file=paste("lasso.pas.res.",i,".txt",sep=""),ncol=length(temp.avg2),append=TRUE,sep="\t")
145
146
147   ###elastic
148   fit.elas=cv.glmnet(x=data.matrix(training),y=as.factor(class.training),family="binomial",type.measure="class",alpha=0.5)
149
150   #var imp min cvm
151   glm.elas.coef=coef(fit.elas,s="lambda.min")
152   glm.elas.coef.noint=as.numeric(glm.elas.coef[-1,])
153   glm.elas.coef.nzero=which(glm.elas.coef.noint!=0)
154   write(glm.elas.coef.nzero,file=paste("elas.var.used.",i,".txt",sep=""),ncol=length(glm.elas.coef.nzero),append=TRUE,sep="
          \t")
155
156   #predict test
157   temp.hold=seq(1,1000,1)
158   temp.hold=temp.hold[!temp.hold %in% glm.elas.coef.nzero]
159   training.2=training
160   training.2[, temp.hold]=0
161   temp.avg=vector("list",250)
162
163   temp.avg = foreach (i=1:250,.packages=c('Hmisc','glmnet','caret'),.combine=rbind) %dopar% {
164     temp.avg[[i]]=0
165     flds=createFolds(class.training,k=5,list=TRUE,returnTrain=FALSE)
166     for (j in 1:5) {
167       temp.hold=seq(1,99,1)
168       temp.hold=temp.hold[!temp.hold %in% flds[[j]]]
169       fit.elas=cv.glmnet(x=training.2[temp.hold,],y=as.factor(class.training[temp.hold]),family="binomial",type.measure="class",
              alpha=0.5)
170
171       pred.elas.class=as.numeric(predict(fit.elas,training.2[flds[[j]],], s="lambda.min",type="class"))
172       pred.elas.prob=predict(fit.elas,training.2[flds[[j]],], s="lambda.min",type="response")
173       pas.res.elas=pas(pred.elas.class,class.training[flds[[j]]], pred.elas.prob)
174
175       temp.avg[[i]]=temp.avg[[i]]+pas.res.elas
176     }
177     return(temp.avg[[i]]/5)
178   }
179   temp.avg2=colMeans(temp.avg[,1:5])
180   write(temp.avg2,file=paste("elastic.pas.res.",i,".txt",sep=""),ncol=length(temp.avg2),append=TRUE,sep="\t")
181
182
183   ###pamr
184   mydata=list(x=t(training),y=factor(class.training),geneid=as.character(1:nrow(t(training))),genenames=paste("g",as.
          character(1:nrow(t(training))),sep=""))
185   pamfit=pamr.train(mydata)
186   pamcv=pamr.cv(pamfit,mydata)
187
188   #find last min threshold
189   min.cv=Inf
190   min.cv.index=1
191   for (h in 1:length(pamcv$error)) {
192     if (min.cv>=pamcv$error[h]) {
193       min.cv=pamcv$error[h]
194       min.cv.index=h
195     }
196   }
197   thr=pamcv$threshold[min.cv.index]
198
199   #var imp threshold
200   var.used.pamr=pamr.predict(pamfit,newx=NULL,threshold=thr,type="nonzero")
201   write(var.used.pamr,file=paste("pam.var.used.",i,".txt",sep=""),ncol=length(var.used.pamr),append=TRUE,sep="\t")
202
203   #predict test
204   temp.hold=seq(1,1000,1)
205   temp.hold=temp.hold[!temp.hold %in% var.used.pamr]
206   training.2=training
207   training.2[, temp.hold]=rnorm(99,0,1e-32)
208   temp.avg=vector("list",250)
209
210   temp.avg = foreach (i=1:250,.packages=c('Hmisc','pamr','caret'),.combine=rbind) %dopar% {
211     temp.avg[[i]]=0
212     flds=createFolds(class.training,k=5,list=TRUE,returnTrain=FALSE)
213     for (j in 1:5) {
214       temp.hold=seq(1,99,1)
```

```
215         temp.hold=temp.hold[!temp.hold %in% flds[[j]]]
216
217         mydata=list(x=t(training.2[temp.hold,]),y=factor(class.training[temp.hold]))
218         pamfit=pamr.train(mydata)
219         pamcv=pamr.cv(pamfit,mydata)
220
221         #find last min threshold
222         min.cv=Inf
223         min.cv.index=1
224         for (k in 1:length(pamcv$error)) {
225           if (min.cv>=pamcv$error[k]) {
226             min.cv=pamcv$error[k]
227             min.cv.index=k
228           }
229         }
230         thr=pamcv$threshold[min.cv.index]
231
232         pred.pamr.class=pamr.predict(pamfit,t(data.frame(training.2[flds[[j]],])),type="class",threshold=thr)
233         pred.pamr.prob=pamr.predict(pamfit,t(data.frame(training.2[flds[[j]],])),type="posterior",threshold=thr)[,2]
234         pas.res.pamr=pas(pred.pamr.class,class.training[flds[[j ]]], pred.pamr.prob)
235
236         temp.avg[[i]]=temp.avg[[i]]+pas.res.pamr
237       }
238       return(temp.avg[[i]]/5)
239     }
240     temp.avg2=colMeans(temp.avg[,1:5])
241     write(temp.avg2,file=paste("pam.pas.res.",i,".txt",sep=""),ncol=length(temp.avg2),append=TRUE,sep="\t")
242   }
243
244   stopCluster(cl)
```