Affymetrix chip elemzési jegyzetek

Tartalomjegyzék

Adatok beolvasása

- a `CEL` állományok beolvasása a `ReadAffy` alkalmazásával:

```
> library(affy)
> Data <- ReadAffy()
```

- a `list.celfiles` függvénnyel a munkakönyvtárban lévő `CEL`-fájlokat tudjuk kilistázni, és meghatározhatjuk, hogy melyeket olvassa be a `ReadAffy` függvény
- a beolvasott adatok egy úgynevezett *AffyBatch* objektumban tárolódnak, ami a további folyamatok függvényeiként bemeneteként szolgál
- a `CEL` fájlok nyers adatokat tartalmaznak, pontonként egy értéket
- a pontok *probeset*-be való rendezéséhez valamint a az adott génnel való összekapcsolásához szükséges információkat a `CDF` fájl tartalmazza
- az `affy` csomag automatikusan megkeresi a megfelelő `CDF` csomagot és betölti (ha létezik)

A probe szintű adatok vizsgálata

- először töltsünk be adatokat

```
> library(affydata)
> data(Dilution)
```

- két függvény, a `pm` és a `mm` segítségével kapcsolódhatunk a probe-szintű adatokhoz

```
> pm(Dilution, "1001_at")[1:3,]
```

```
           20A   20B   10A   10B
1001_at1 128.8  93.8 129.5  73.8
1001_at2 223.0 129.0 174.0 112.8
1001_at3 194.0 146.8 155.0  93.0
```

- az Affymetrix chip mátrixok a gének expressziójának mérésére a probok egyes csoportjait használják

```
> cdfName(Dilution)

[1] "HG_U95Av2"

> length(geneNames(Dilution))

[1] 12625

> length(probeNames(Dilution))

[1] 201800
```

Probe-intenzitási ábrák

Az alábbi kódok segítségével ábrázolhatjuk a probe-intenzitásokat:

```
> matplot(pm(Dilution, "1001_at"), type = "l", xlab = "Probe No.", ylab = "PM Probe intensity")
> matplot(t(pm(Dilution, "1001_at")), type = "l", xlab = "Array No.", ylab = "PM Probe intensity")
```

A fenti függvények egy adott probeseten belüli probok mintázata a probok szerint, illetve a mintákon keresztül. Figyeljük meg a nagy probe hatásokat. A probok közötti variabilitás nagyobb, mint a mátrixok közötti variabilitás.

Leíró adatok

- a `phenoData` rész tárolja a leíró adatokat
- a `pData` függvényt használhatjuk ezen információ

```
> pData(Dilution)

    liver sn19 scanner
20A    20    0       1
20B    20    0       2
10A    10    0       1
10B    10    0       2
```

- a leíró adatok két különböző mintából származó RNS koncentrációkból állnak, májból és a központi idegrenszerből vett totál RNS-ből véve, valamint a szkennerek azonosítóját is tartalmazza

A probe-intenzitások viselkedése

- különböző mátrixokban a probe-intenzitások viselkedését vizsgálhatjuk a `hist` és a `boxplot` függvényekkel
- a boxplottok hasznosak a mátrixok közötti nyers probe-intenzitási szintek különbségének azonosítására
- a mátrixok eloszlásának alakjában és középpontjában tapasztalható különbségek gyakran rávilágítanak a normalizáció szükségességére

```
> hist(Dilution)
> boxplot(Dilution)
```

MA-plotok

- az MA-plot két Y_1 és Y_2 vektor 45 fokkal elforgatott és átskálázott tengelyű szórásdiagramja
- $j = 1, \ldots, J$ gén $Y_{2,j}$-nek $Y_{1,j}$ függvényében való ábrázolása helyett, $M_j = Y_{2,j} - Y_{1,j}$-nek $A_j = (Y_{2,j} + Y_{1,j})/2$ függvényében való ábrázolását használjuk
- ha Y_1 és Y_2 logaritmizált expressziós értékek, akkor M_j a j gén log-fold változását fejezi ki
- és A_j az adott gén átlagos logaritmizált intenzitását jelenti
- a `geneplotter` csomag `smoothScatter` függvényével tudunk simított ábrákat készíteni
- ha a legtöb gén expressziójában nincsen különbség, akkor a loess görbe az $M = 0$ vízszintes egyeneshez közelít
- ha a loess görbe nem-lineáris, az azt jelzi, hogy M és A között valamilyen kapcsolat van; vagy a különbözően expresszált és az átlagos intenzitás között
- a szórásdiagram azért van elforgatva, mert így egyszerűbb a mintázat értelmezése, a vízszintes egyeneshez való viszonyítás

$$M_j = Y_{2,j} - Y_{1,j}$$
$$A_j = (Y_{2,j} + Y_{1,j})/2$$

Amennyiben Y_1 és Y_2 logaritmikus skálájú expresszió értékekeből álló vektorok, M_j a log fold change-et, A_j pedig az átlagos log intenzitást jelenti az adott j génre vonatkozóan. Az `affy` csomagban két függvény is van az MA-plot készítésére: `mva.pairs` és az `MAplot`. Az előbbi az `AffyBatch`-ban tárolt összes tömbre elkészíti a páronkénti szórásdiagramot. Ha sok tömbbel dolgozunk, nem hasznos az összes lehetséges párt összehasonlító ábrázolás. Ebben az esetben hasznosabb egy referencia tömbhöz való hasonlítása az egyes tömböknek. Erre használhatjuk a `MAplot` függvényt. A referencia tömböt probe-szintű mediánnal hozzuk létre az összes tömb felhasználásával. A szórásdiagramon az IQR és a M mediánja is leolvasható.
A görbék, $M = 0$ tengelytől való eltérése a referencia tömbtől való intenzitás eltérést jelenítik meg, ha ahhoz közel húzódnak, akkor a legtöbb gén expressziója nem különbözik.

Modell

$$Y = B + S$$

- B: a háttér „zaj” intenzitása (az optikai hatások és a nem-specifikus kötődésből adódik)
- S: specifikus kötődés

$$log(S) = \theta + \phi + \epsilon$$

  - θ: a valódi mennyiség logaritmusa
  - ϕ: a probe hatása
  - ϵ: mérési hiba

Háttérkorrekció, normalizáció, összegzés

- az Affymetrix expressziós mátrixok előfeldolgozása általában három lépést foglal magába:
  1. háttérkorrekció
  2. normalizáció
  3. összegzés
- a Bioconductor szoftver széles választékban nyújt eszközöket erre a három lépésre
- a háttérkorrekciót és normalizációt végző függvények általában egy *AffyBatch* objektumot használnak bemenetként és eredményül egy feldolgozott *AffyBatch* objektumot adnak vissza
- az összegzésre használatos eljárások *exprSet* objektumot hoznak létre, ami összesített expressziós értékeket tartamaz

- **Háttérkorrekció**
  - RMA
  - MAS 5.0
  - Ideal Mismatch

RMA

- amikor az RMA-t fejlesztették, a készítők úgy találták, hogy az MM probok problematikusak, így egy olyan megoldást javasoltak, ami csak a PM probokat használja
- a PM értékeket mátrixról mátrixra korrigálják, egy a probe intenzitási modell segítségével, ami a probe intenzitások tapasztalati eloszlásán alapszik
- a megfigyelt PM értékeket úgy modellezik, mint a $B \sim N(\mu, \sigma^2)$ „zaj" elemek és a $S \sim Exp(\alpha)$ jel elem összegét
- az esetlegesen előforduló negatív értékek elkerülésére a normál eloszlás nullánál le van vágva
- legyen Y a megfigyelt intenzitás

$$E(S|Y = y) = a + b\frac{\phi(\frac{a}{b}) - \phi(\frac{y-a}{b})}{\Phi(\frac{a}{b}) + \Phi(\frac{y-a}{b}) - 1} ,$$

  ahol $a = y - \mu - \sigma^2\alpha$ és $b = \sigma$. A ϕ és Φ a standard normális sűrűség és eloszlás függvények.
- a `Dilution` adatállományból az alábbi kóddal készíthetünk egy háttérkorrigált *AffyBatch* objektumot

```
> Dilution.bg.rma <- bg.correct(Dilution, method = "rma")
```

GCRMA

- az RMA háttérkorrekció nem veszi figyelembe az egyes probok sajátosságait, a nem specifikus kötődést. Így a hátteret gyakran alulbecsüli.
- az egyes probok sajátosságait azok szekvenciája határozza meg
- a szekvenciára vonatkozó információt használva számítjuk az *affinitás* mértéket
- a háttér zaj eloszlását a hasonló affinitások használatával becsüljük, ahelyett, hogy minden probot felcserélhetőnek tekintenénk
- a `Dilution` adatállományból egy GCRMA háttérkorrigált *AffyBatch* objektum létrehozására a következő kódot használhatjuk:

```
> library(gcrma)
> Dilution.bg.gcrma <- bg.adjust.gcrma(Dilution)
```

MAS 5.0

- a Statistical Algorithm Description Documentumba írták le és a MAS 5.0 szoftver használja
- a chip egy k számú négyszög alaku területből álló gridre van osztva (az alaértelmezés $k = 16$)
- minden területen belül a probe intenzitások legalacsonyabb 2%-át használja a grid háttér értékének számításához
- ezután minden háttér érték súlyozott átlagával korrigálják a probe-ok intenzitását
- a súlyozás az adott probe-nak a grid középpontjától való távolságán alapszik:

$$w_k(x, y) = \frac{1}{d_k^2(x, y) + s_0} ,$$

  ahol $d_k(x, y)$ (x, y) euklédeszi távolsága a k grid középponttól. A simítási koefficiens s_0 alapértelmezett értéke 100.

- különös figyelmet fordítanak arra, hogy a negatív értékeket kiküszöböljék vagy más numerikus problémákat, amik az alacsony intenzitású területeken előfordulhatnak
- a módszer mind a PM, mind a MM probokat korrigálja
- háttérkorrigált *AffyBatch* objektumot létrehozhatunk az alábbi kóddal:

```
> Dilution.bg.mas <- bg.correct(Dilution, method = "mas")
```

Ideal Mismatch

- az MM probok használatával a PM probok korrigálhatók a segítségükkel, annak érdekében, hogy a nem-specifikus kapcsolódásokat javítandó, a PM probok intenzitás értékekből a megfelelő MM prob intenzitások kivonásával
- ez problematikus lehet, mivel agy tipikus mátrixban az MM probok 30%-ának magasabb intenzitásértékei vannak, mint a hozzá tartozó PM proboknak
- így ha a nyers MM intenzitási értékeket kivonjuk a PM intenzitási értékekből, akkor sok negatív expressziós értéket fogunk kapni, aminek viszont nincs sok értelme
- az MM értékek negatív hatásának orvoslására az Affymetrix bevezetett egy koncepciót, az *Ideal Mismatch (IM)*, ami garantálja, hogy az kisebb lesz, mint a megfelelő PM intenzitás
- a cél, hogy az MM-et használhassuk, amikor az fizikálisan lehetséges és hogy az értéke kisebb legyen mint a PM
- ha i a probe és k a probeset, akkor az i és k indexü probepárra vonatkozó IM az alábbi szerint határozható meg:

$$IM_i^{(k)} = \begin{cases} MM_i^{(k)} & \text{ha } MM_i^{(k)} < PM_i^{(k)} \\ \frac{PM_i^{(k)}}{2^{SB_k}} & \text{ha } MM_i^{(k)} \geq PM_i^{(k)} \text{ és } SB_k \leq \tau_c \\ \frac{PM_i^{(k)}}{2^{\tau_c/(1+(\tau_c - SB_k)/\tau_s)}} & \text{ha } MM_i^{(k)} \geq PM_i^{(k)} \text{ és } SB_i \leq \tau_c \end{cases},$$

  ahol τ_c és τ_s finomító konstansok, a kontraszt τ alapértelmezett értéke 0.03, és a skálázó τ alapértelmezett értéke 10
- a korrigált PM intenzitás az IM értéknek a megfigyelt PM intenzitásból való kivonással adódik
- ennek a háttérkorrekciónak a használatához vagy magunk által írt kódot, vagy az `affyPLM` csomag `threestep` függvényét használhatjuk

Normalizáció

- a normalizáció arra szolgál, hogy a különböző mátrixok adatait úgy manipuláljuk, hogy azok összehasonlíthatóak legyenek, lehetnek lineárisak vagy nem-lineárisak
  - skála-normalizáció (lineáris)
  - nem-lineáris normalizáció:
    - cross-validated splines (Schadt et al 2001)
    - running median lines (Li and Wong, Genome Biology 2001)
    - loess smoothers (Bolstad et al)
  - kvantilis normalizáció, ami mindegyik mátrixra az intenzitásoknak ugyanazt a tapasztalati eloszlását illeszti
- egy *AffyBatch* objektum normalizálására használhatjuk a `normalize` függvényt

Skála-normalizáció

Válasszuk ki az X *AffyBatch* objektum egy oszlopát, mint alapmátrixot, és nevezzük j-nek
Számítsuk ki a j oszlop (trimmelt) átlagát, ezt nevezzük $\tilde{X}_j$-nek
$i = 1$-től n-ig ciklusban úgy, hogy $i \neq j$ hajtsuk végre
  számítsuk ki az i oszlop (trimmelt) átlagát, amit nevezzünk $\tilde{X}_i$-nek
  számítsuk ki $\beta_i = \tilde{X}_j / \tilde{X}_i$-t
  szorozzuk meg az i oszlopot β_i-vel
ciklus vége
egy *AffyBatch* objektumot skála-normalizálhatunk a következő kóddal:

```
> Dilution.norm.scale <- normalize(Dilution, method = "constant")
```

Nem-lineáris normalizáció

Válasszuk ki az X *AffyBatch* objektum egy oszlopát, mint alapmátrixot, és nevezzük j-nek
$i = 1$-től n-ig ciklusban úgy, hogy $i \neq j$ hajtsuk végre
  illesszünk egy smooth nem-lineáris összefüggést i oszlop és j alapmátrixra vonatkozóan, ezt nevezzük $\hat{f}_i$-nek
  a j oszlopra vonatkozó normalizált értékek az $\hat{f}_i(X_j)$-ből adódik
ciklus vége
Nem-lineáris normalizáció hajtható végre az alábbi kóddal:

```
> Dilution.norm.nonlinear <- normalize(Dilution, method = "invariantset")
```

Kvantilis normalizáció

Az X *AffyBatch* n számú p hosszúságú vektorból áll ($p \times n$), ahol aminden mátrix egy oszlop.
Rendezzük sorba X minden oszlopát külön-külön, aminek az eredménye az X_s.
Számítsuk ki az X_s minden sorának átlagát, és készítsünk egy olyan X'_s mátrixot, aminek a dimenziói egyeznek az X dimenzióival, és minden sor egyenlő az X_s sorátlagával.
Az X'_s az eredeti sorrendenk megfelelő újrarendezésével megkapjuk az X_n-t.
Az eljárást az alábbi kóddal végezhetjük el:

```
> Dilution.norm.quantile <- normalize(Dilution, method = "quantiles")
```

Cyclic loess

```
> Dilution.norm.loess <- normalize(Dilution, method = "loess")
```

Variancia stabilizáló normalizáció (vsn)

- a vsn eljárás kombinálja a háttérkorrekciót és a normalizációt
- egy lehetséges előnye a kombinált megközelítésnek az, hogy a mátrixok között megoszthatható információk a háttérkorrekció paraméterek becslésére használható, ami egyébként mátrixonként külön-külön történik
- egy x_{ki} mátrixra, ahol k a probok, i pedig a mátrixok indexét jelöli, az alábbi normalizációs transzformációt illesztünk

$$x_{ki} \mapsto h_i(x_{ki}) = glog\left(\frac{x_{ki} - a_i}{b_i}\right) ,$$

  ahol b_i i mátrix skála paramétere, a_i a háttér kontraszt, és a *glog* pedig az ún. generalizált logaritmus vagy attenuált logaritmus
- egy előnyös tulajdonsága a *glog*-nak, hogy megfelelő a_i és b_i értékek mellett, a különböző tömbökből származó adatokat nem csak egymáshoz igazítja, hanem az ismétlések között a varianciák közelítőleg független az átlagtól
- a modell ilesztését és a transzformációt a `vsn` csomag segítségével hajthatjuk végre, az alábbi kóddal egy *AffyBatch* objektumot normalizálhatjuk:

```
> library(vsn)
> Dilution.vsn <- normalize(Dilution, method = "vsn")
```

- az új *AffyBatch* objektumban a transzformációs paramétereket a `description` tulajdonság `preprocessing` részében kapjuk vissza

- egy *AffyBatch* objektumra a `normalize.methods` függvények keresztül használható normalizációs eljárásokat az alábbi szerint listázhatjuk ki:

```
> normalize.methods(Dilution)

[1] "constant"         "contrasts"        "invariantset"     "loess"
[5] "qspline"          "quantiles"        "quantiles.robust" "vsn"
```

Összegzés

- az Affymetrix GeneChip adatok előfeldolgozásának utolsó lépése az összegzés[1]
- a folyamat során az egyes probe intenzitási értékekből létrejön a probeset szintű érték, ami az expressziós érték lesz
- a Bioconductor csomagok számos függvényt tartalmaznak az összegzés végrehajtására, a génexpressziós értékek számításához: az `expresso` és a `threestep` függvények segítségével, amelyek lehetővé teszik a felhasználó számára, hogy különböző előfeldolgozási eljárásokat állítson be
- más függvények speciális expressziós mértékek számítására szintén elérhetők, úgy mint a `rma`, a `gcrma` vagy az `expressopdnn`

expresso

- az `expresso` igen széleskörű lehetőségeket biztosít az expressziós értékek számítására
- lehetővé teszi a legtöbb háttérkorrekciós, normalizációs és összegző eljárás kombinálását
- azonban az `expresso` gyakran jelentősen lassabb, mint azok a függvények, amiket egy speciális expressziós érték számításra fejlesztettek
- a háttérkorrekcióra, a PM korrekcióra és az összegzésekre vonatkozó argumentumokhoz elérhető beállítási lehetőségeket az alábbi utasításokkal listáztathatjuk ki:

```
> bgcorrect.methods

[1] "mas"  "none" "rma"  "rma2"

> pmcorrect.methods

[1] "mas"        "pmonly"     "subtractmm"

> express.summary.stat.methods

[1] "avgdiff"      "liwong"       "mas"          "medianpolish" "playerout"
```

Előfeldolgozási példák

mas5

a MAS 5 becslést az alábbi szerint is megvalósíthatjuk:

```
> eset <- mas5(Dilution)
```

expresso

- az `expresso` függvény használata a beállítási lehetőségek többségének alkalmazásával:

```
> eset <- expresso(Dilution, bgcorrect.method = "rma",
+     normalize.method = "constant", pmcorrect.method = "pmonly",
+     summary.method = "avgdiff")
```

- vagy a csak PM-et használó modellnek az a változata, ami a Li és Wong által fejlesztett expressziós indexet használja:

```
> eset <- expresso(Dilution, bgcorrect.method = "invariantset",
+     normalize.method = FALSE, pmcorrect.method = "pmonly",
+     summary.method = "liwong")
```

[1] summarization

threestep

- az `affyPLM` csomagban található a `threestep` függvény, ami sokféle expressziós mérték számítását teszi lehetővé
- mivel a `threestep` lefordított kód ezért általában gyorsabb, mint az `expresso`
- a `threestep` mindig log_2 skálán adja vissza az expressziós értékeket
- az IM-et a PM-ből kivonásával, a tömbök közötti kvantilis normalizációval és összegzésként a robusztus átlagot használó kód példa:

```
> library(affyPLM)
> eset <- threestep(Dilution, background.method = "IdealMM",
+     normalize.method = "quantile", summary.method = "tukey.biweight")
```

RMA

- az `rma` egy expressziós értéket számító függvény, ami a következő elemekből, eljárásokból épül fel:
  - RMA háttérkorrekció
  - kvantilis normalizáció
  - medián polish összegzés (egy több tömbre alkalmazott robusztus modellillesztés)

```
> eset <- rma(Dilution)
```

- az `rma` függvény helyett használhatjuk a `justRMA` függvényt, azokban az esetekben, ha nagy számú `CEL` fájlt használunk, és nem akarunk további alacsony szintű elemzéseket végezni

GCRMA

- szintén három előfeldolgozási lépésből áll:
  - a probe szekvenciák által meghatározott *affinitás* alkalmazásával működő háttérkorrekció
  - kvantilis normalizáció
  - medián polish összegzés (egy több tömbre alkalmazott robusztus modellillesztés)
- a `gcrma` függvény kiszámítja az *AffyBatch* objektumból egy *exprSet* objektumot hoz létre

```
> library(gcrma)
> eset <- gcrma(Dilution)
```

- az affinitási információkat kiszámíthajuk egyszer, elmenthetjük és ugyanazt máskor is felhasználva a további előfeldolgozásokat gyorsíthatjuk:

```
> ai <- compute.affinities(cdfName(Dilution))
> eset <- gcrma(Dilution, affinity.info = ai)
```

- az alapértelmezett háttérkorrekció eljárás mind az affinitási információkat, mind pedig a megfigyelt MM intenzitásokat használja (`type="fullmodel"`)
- amennyiben csak az affinitási információkat szeretnénk használni, akkor a `type="affinities"`, ha csak az MM intenzitásokat, akkor a `type="mm"` beállítást használhatjuk

```
> eset <- gcrma(Dilution, affinity.info = ai, type="affinities")
```

Milyen előfeldolgozást használjunk?

A számos háttérkorrekció, normalizáció és összegző módszerkombinációk közül nem könnyü kiválasztani a helyes kombinációt. Az `affycomp` csomag függvényeit használhatjuk a megfelelő eljárás kiválasztásában.

Minőségellenőrzés

- az `affyPLM` csomag számos eszközt biztosít az Affymetrix chip-ek minőségi ellenőrzésére
- az `simpleaffy` csomag szintén rendelkezik a minőségellenőrzési eljárásokkal, ezek az Affymetrix ajánlásokon alapszanak

Példa adatok

- betöltjük az ALLMLL minta adatállományt és abból leválogatunk részeket, amiket a továbiakban használni fogunk

```
> library("RbcBook1")
> library("hgu133bcdf")
> library("affy")
> library("ALLMLL")
> data(MLL.B)
> Data <- MLL.B[,c(2,1,3:5,14,6,13)]
> sampleNames(Data) <- letters[1:8]
```

Chip képének megjelenítése

```
> palette.gray <- c(rep(gray(0:10/10),times=seq(1,41,by=4)))
> image(Data[,1],transfo=function(x) x, col=palette.gray)
> image(Data[,1],col=palette.gray)
```

Affymetrix minősítési mértékek

Az Affymetrix javasol néhány minősítési mértékeket:

- Average Background (Átlagos háttér): a 16 háttérérték átlaga
- Scale Factor (Skálázási faktor): a β_i konstans, ami az i tömb trimmelt átlagának és a referencia tömb trimmelt átlagának hányadosa
- Percent Present (Jelenléti százalék): a jelenlévő spotok, probok százalékos aránya
- $3'/5'$ ratios ($3'/5'$ arány): különböző minőségi kontrolként tekinthető probesetekre, úgy mint a β-Actin és a GAPDH, amelyek mind 3 probsettel vannak representálva, ezek közül egy transzkipt $5'$ végéről, egy a közepéről és egy a $3'$ végéről. A $3'$ vég expressziós értékének és az $5'$ vég expressziós értékének aránya szolgál az RNS minőségének kifejezésére.

Affymetrix QC

Először töltsük be a `simpleaffy` csomagot és a `qc` függvény meghívásával végezzünk számításokat.

```
> library(simpleaffy)
> Data.qc <- qc(Data)
```

```
Background correcting
Retrieving data from AffyBatch...done.
Computing expression calls...
........done.
scaling to a TGT of 100 ...
Scale factor for: a 9.76598646950149
Scale factor for: b 4.90548890944312
Scale factor for: c 10.4895287164673
Scale factor for: d 7.05332341070512
Scale factor for: e 7.5616125563141
Scale factor for: f 2.47522435896775
Scale factor for: g 13.5312382045890
Scale factor for: h 8.08945757690254
Getting probe level data...
Computing p-values
Doing PMA Calls
```

- Az átlagos háttér az egyes tömbökre vonatkozóan:

```
> avbg(Data.qc)
```

```
    a     b     c     d     e     f     g     h
 68.2  67.3  42.1  61.3  53.6 128.4  49.4  49.3
```

Az egyes chipek átlagos háttérértékei összehasonlíthatók. Az `f` tömb átlagos hátere viszont. Ez egy problémára mutathat rá.

- A skálázási faktort az alábbi módon számíthatjuk ki:

```
> sfs(Data.qc)

[1]  9.77  4.91 10.49  7.05  7.56  2.48 13.53  8.09
```

Ezeknek az értékeknek 3-szoros[2] tartományon belül kell lennie egymáshoz viszonyítva. A példában megjelenik egy probléma az `f` és `g` tömbök esetén.

- A jelenléti százalékot a következő kóddel olvashatjuk ki:

```
> percent.present(Data.qc)

a.present b.present c.present d.present e.present f.present
     21.7      26.5      25.6      23.5      23.4      25.3
g.present h.present
     18.0      24.4
```

Ismételt minták esetén ezeknek hasonlónak kell lennie. Nagyon alacsony értékek gyenge minőséget jelezhetnek.

- A $3'/5'$ arány az első kettő minőségi kontrol probsetre vonatkozóan:

```
> ratios(Data.qc)[,1:2]

  AFFX-HSAC07/X00351.3'/5' AFFX-HUMGAPDH/M33197.3'/5'
a                    0.970                     0.1639
b                    0.324                     0.0580
c                    0.466                    -0.1557
d                    1.257                     0.5755
e                    0.604                    -0.1402
f                    0.672                     0.2467
g                    0.380                    -0.0183
h                    0.485                     0.2768
```

Ezeknek 3-nál kisebbnek kell lenniük.

Affymetrix több-tömbös vizualizáció

Több tömböt tartalmazó *AffyBatch* objektum vizualizálható egyszerre. Az alábbi két példában a nem előfeldolgozott mért adatok log transzformált probe intenzitási értékeit ábrázoljuk. Boxplottal:

```
> library("RColorBrewer")
> cols <- brewer.pal(9,"Set1")[-6]
> boxplot(Data,col=cols)
```

Hisztogrammal:

```
> hist(Data,col=cols,lty=1,xlab="Log (base 2) intensities")
> legend(12,1.0,letters[1:8],lty=1,col=cols)
```

RNS-degradáció

- az RNS-degradáció rajz arról tájékoztat bennünket, hogy vannak-e nagy különbségek az RNS-degradációban az egyes tömbök között
- nem annyira a degradáció mértéke (meredekség) fontos, hanem sokkal inkább az, hogy valamely egyenes(ek) meredeksége vagy más tulajdonsága nem tér-e jelentősen a többiekétől
- egy egyedi probeseten belül a probok hatása dominál; a $3'/5'$ trend azonban csak akkor jelenik meg, ha a probesetek nagy számának átlagát vizsgáljuk

[2] within 3-fold of each other

A példában 3 amplifikált RNS-sel illetve 3 az Affymetrix ajánlása szerint hibridizált chipet hasonlítunk össze.

```
> library("AmpAffyExample")
> data(AmpData)
> sampleNames(AmpData) <- c("N1","N2","N3","A1","A2","A3")
> RNAdeg <- AffyRNAdeg(AmpData)
> plotAffyRNAdeg(RNAdeg,col=c(2,2,2,3,3,3))
```

Mindegyik görbe egy-egy chip-et jelent. Az Y tengelyen az átlagos intenzitást ábrázoljuk. Az intenzitás értékek el lettek tolva az eredetiektől a könnyeb áttekinhetőség érdekében, azonban a meredekség változatlan.

Minőségi értékelés

- az artefaktok és kisebb foltok nem ritkák a chipek elemzése során. Ezeknek nem szükségszerűen van nagy hatásuk a génexpresszió becslésére.
- de esetenként lehetnek chipek, amelyek rosszak, vagyis nagyon gyenge minőségűek illetve alavetően különböznek a többitől. Adott esetben ezek kihagyása a további elemzésekből lehet a megoldás.
- az `affyPLM` csomagban találhatók eszközök diagnosztikai eljárásokra, amelyek a Y_{gij} háttérkorrigált és normalizált probe szintű adatokra épülő modellen alapszanak

$$log(Y_{gij}) = \theta_{gi} + \phi_{gj} + \epsilon_{gij}$$

  - θ_{gi} az i tömb g génjére vonatkozó logaritmált expresszió
  - ϕ_{gj} a j probe hatása a g génre

RLE (Relative Log Expression)

- számítsuk ki minden i tömbben minden g génre a $\hat{\theta}_{gi}$ logaritmált expressziós becslést
- számítsuk ki minden génre az összes tömbből a m_g medián értéket
- definiáljuk a relatív expressziót, mint $M_{gi} = \hat{\theta}_{gi} - m_g$
- ezt a relatív expressziós értéket jelenítjük meg boxplottal minden egyes tömbre vonatkozóan külön-külön
- ha egy tömb valamiért problematikus, akkor vagy nagyobb kiterjedéssel rendelkezik, vagy pedig a közepe nem az $M = 0$ értéknél lesz, vagy mindkettőt mutatja

```
> library("affyPLM")
> library("ALLMLL")
> library("RColorBrewer")
> data(MLL.B)
> Pset2 <- cache("Pset2",fitPLM(MLL.B))
> cols <- brewer.pal(9,"Set1")[-6]
> Mbox(Pset2,ylim=c(-1,1),col=cols,names=NULL,main="RLE")
```

NUSE (Normalized Unscaled Standard Error)

- a PLM illesztés után adodó standard hibát minden génre, minden tömbön becsüljük
- a NUSE-t az alábbi szerint számítjuk:

$$NUSE(\hat{\theta}_{gi}) = \frac{SE(\hat{\theta}_{gi})}{med_i(SE(\hat{\theta}_{gi}))}$$

- a NUSE értékeket boxplottal ábrázoljuk
- a gyenge minőségű tömbök azok, amelyek szignifikánsan elemelkedettek, vagy szélesebbek, a többihez viszonyítva

```
> boxplot(Pset2,ylim=c(0.95,1.5),col=cols,names=NULL, main="NUSE",outline=FALSE)
```

Gén expressziós különbségek elemzése I.

- sok mikrochip kísérlet célja, hogy olyan géneket találjanak, amelyek különbséget tesznek kettő vagy több csoport között
- általában nagyon számú gént mérünk és kis számú mintánk van mindegyik csoportban
- a gének nem függetlenek egymástól, de általában a kapcsolataikat csak kevéssé ismerjük, itt génről-génre végzett elemzéseket végzünk

Adatok előfeldolgozása és transzformálása

- előfeldolgozás
- skála
  - log-skála: az arányok logja szimmetrikus 0 körül
  - a $log_2(4/1)$ és a $log_2(1/4)$ átlaga 0, míg 4/1 és 1/4 átlaga nagyobb, mint 2

Alapok

Tegyük fel, hogy megfigyeltünk két gént, amelyeknek a fold-change 2. Érdekes ez?

Egy gén statisztikai tesztelésének alapjai

- Null hipotézis: az A gén expressziója nem különbözik a két csoportban, $\mu_1 = \mu_2$
- megfigyelések: $x_1, \ldots, x_m,\ y_1, \ldots, y_n$
- átlagok: $\bar{x} = \frac{1}{m}\sum_{i=1}^{m} x_i,\ \bar{y} = \frac{1}{n}\sum_{i=1}^{n} yx_i$
- SD^2 vagy variancia: $s_X^2 = \frac{1}{m-1}\sum_{i=1}^{m}(x_i - \bar{x})^2\ s_Y^2 = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2$
- nincs ismétlés: nincs variancia becslés. Fold change?
- paraméteres tesztek: pl. t-teszt, ANOVA. Erősebbek, amennyiben a feltételeik legalább közelítőleg teljesünek.
- nem paraméteres tesztek: pl. Wilcoxon teszt. Kevsébé szigorúak az eloszlási feltételek, de kisebb az erejük is kevés ismétlés esetén. Mi a legkisebb p-érték, amit 3-3 minta összehasonlításával kaphatunk?

T-teszt

$$\frac{\delta}{\hat{se}(\delta)} = \frac{\bar{Y} - \bar{X}}{\sqrt{\frac{s_Y^2}{n} + \frac{s_X^2}{m}}}$$

- ha a minta nagy, akkor a *t-statisztika* normál eloszlású, 0 átlaggal és 1 szórással
- Ha X és Y normál eloszlásúak, akkor a *t-statisztika* t-eloszlást követ függetlenűl a minta méretétől (a szabadságfok a minta méretétől függ)
- kis mintával az SE becslése nem stabil

A variancia jobb becslése

- Empírikus Bayes módszerek, amik a többi géntől „kölcsönöznek erőt”
- az s_g^2 helyett s_0^2 és s_g^2 súlyozott átlaga
- kompromisszum a globális variancia és a gén-specifikus standard hiba becslés között
- gyakori eredmény a „moderált” tesztekben

Most, hogy van egy teszt-statisztikánk, mi a helyzet a „szignifikanciával”?

- a t-statisztikát mindig ki lehet számolni. Vajon a t-eloszlásból a p-érték használható-e a megfigyelt adatok normlitásának feltétele mellett.
- a moderált t-statisztikáknál, a permutációk segíthetnek a null hipotézis generálásában
  - permutáljuk a „cimkéket” $b = 1, \ldots, B$ permutációban
  - minden permutációban számítsuk ki a t_b teszt-statisztikát
  - számoljuk meg, hogy hány esetben volt legalább akkora, vagy nagyobb a t_b, mint a megfigyelt teszt-statisztika $p = \#\{b : |t_b| >= |t_{obs}\}/B$

Sokszoros összehasonlítási probléma

- Amikor egyszerre nagy számú gént tesztelünk, annak a valószínűsége, hogy néhány alacsony p-értéket kapunk nem differenciáló gének esetén magas lesz
- egy egyszerű simuláció: 6000 gén 8 kezeléssel és 8 kontrollal. Minden gén expressziós érték standard normál eloszlásból származó szimuláció, nincs differenciáló expresszió
- a valódi expressziós adatok esete komplexebb, de ugyanez a probléma ott is fenn áll

gén index	t-érték	p-érték
983	5.4	1e-04
3141	-4.95	2e-04
3986	4.4	6e-04
5081	4.25	8e-04
5607	-4.24	8e-04
3852	4.22	9e-04
2872	-4.18	9e-04
1473	-4.07	0.0012
2073	3.98	0.0014
314	-3.85	0.0018

- Mit tehetünk?
  - korrigált p-érték
    - a `multtest` csomag több lehetőséget nyújt erre, beleértve a Bonferoni (FWER) és a Benjamin-HOchberg (FDR)
    - `mt.maxT`, `mt.raw2adjp` függvények
    - 15. fejezet
  - gének listázása FDR-rel
$$FDR = E\left[\frac{\text{fals felfedezések}}{\text{összes felfedezés}}\right]$$
  - csökkentsük a tesztek számát nem specifikus szűréssel (az ábrán az látszik, hogy a magasabb IQR értékekhez magasabb abszolut értékű t-statisztikákk tartoznak)

Az érdekes génekre koncentrálva

Ahelyett, hogy azt kérdezzük, hogy „az összes gén között melyek azok, amelyek különböző expressziót mutatnak”, kérdezzük azt, hogy „azok között a gének között, amelyek thyrozin kináz aktivitással rendelkezik, melyek között találunk különböző expressziót?”

```
> library(GO)
> library(annotate)
> tykin <- unique(lookUp("GO:0004713", "hgu95av2", "GO2ALLPROBES"))
> length(tykin)
```

```
[1] 268
```

Több csoport összehasonlítása

Minden génre

$$y_{ik} = \alpha_k + \epsilon_{ik},\ i = 1, 2, \ldots, n_k;\ k = 1, 2, 3, \ldots,$$

ahol k a daganat típusára míg i pedig a mintára vonatkozik.

Gén expressziós különbségek elemzése II.

Adatok betöltése

Az adatok, amiket felhasználunk Chiaretti és mtsai. által közölt akut limfoblasztikus leukémia (ALL) esetekből származnak, amelyeket HGU95AV2 Affyimetrix chippel elemeztek. Az ALL adat-csomag tartalmazza egy exprSet formájában, amit ALL-nak neveznek, és expressziós értékeket tarttalmaz, ami rma-val lett normalizálva (az intenzitás értékek log2-skálájúak), ezen kívűl a minták annotációját is tartalmazza.

a. Töltsük be az ALL csomagot. Milyen dimenziói vannak az expressziós adatmátrixnak?

b. Próbáljuk ki a show függvényt, amikvel áttekintést kapunk az exprSet objektumról. Milyen változók írják le a mintákat a pData tulajdonságban?

```
> library(ALL)
> library(hgu95av2)
> library(annotate)
> data(ALL)
> show(ALL)

Expression Set (exprSet) with
        12625 genes
        128 samples
                 phenoData object with 21 variables and 128 cases
         varLabels
                cod:  Patient ID
                diagnosis:  Date of diagnosis
                sex:  Gender of the patient
                age:  Age of the patient at entry
                BT:  does the patient have B-cell or T-cell ALL
                remission:  Complete remission(CR), refractory(REF) or NA. Derived from CR
                CR:  Original remisson data
                date.cr:  Date complete remission if achieved
                t(4;11):  did the patient have t(4;11) translocation. Derived from citog
                t(9;22):  did the patient have t(9;22) translocation. Derived from citog
                cyto.normal:  Was cytogenetic test normal? Derived from citog
                citog:  original citogenetics data, deletions or t(4;11), t(9;22) status
                mol.biol:  molecular biology
                fusion protein:  which of p190, p210 or p190/210 for bcr/able
                mdr:  multi-drug resistant
                kinet:  ploidy: either diploid or hyperd.
                ccr:  Continuous complete remission? Derived from f.u
                relapse:  Relapse? Derived from f.u
                transplant:  did the patient receive a bone marrow transplant? Derived from f.u
                f.u:  follow up data available
                date last seen:  date patient was last seen

> dim(exprs(ALL))

[1] 12625   128

> print(summary(pData(ALL)))

     cod             diagnosis           sex          age              BT
 Length:128         Length:128         F   :42   Min.   : 5.00   B2     :36
 Class :character   Class :character   M   :83   1st Qu.:19.00   B3     :23
```

```
 Mode  :character   Mode  :character   NA's: 3   Median :29.00   B1     :19
                                                 Mean   :32.37   T2     :15
                                                 3rd Qu.:45.50   B4     :12
                                                 Max.   :58.00   T3     :10
                                                 NA's   : 5.00   (Other):13
 remission      CR              date.cr           t(4;11)
 CR  :99   Length:128         Length:128         Mode :logical
 REF :15   Class :character   Class :character   FALSE:86
 NA's:14   Mode  :character   Mode  :character   TRUE :7
                                                 NA's :35



  t(9;22)        cyto.normal        citog              mol.biol
 Mode :logical   Mode :logical   Length:128         ALL1/AF4:10
 FALSE:67        FALSE:69        Class :character   BCR/ABL :37
 TRUE :26        TRUE :24        Mode  :character   E2A/PBX1: 5
 NA's :35        NA's :35                           NEG     :74
                                                    NUP-98  : 1
                                                    p15/p16 : 1

   fusion protein     mdr                kinet       ccr
 p190     :17     Length:128         dyploid:94   Mode :logical
 p190/p210: 8     Class :character   hyperd.:27   FALSE:74
 p210     : 8     Mode  :character   NA's   : 7   TRUE :26
 NA's     :95                                     NA's :28



  relapse        transplant          f.u            date last seen
 Mode :logical   Mode :logical   Length:128         Length:128
 FALSE:35        FALSE:91        Class :character   Class :character
 TRUE :65        TRUE :9         Mode  :character   Mode  :character
 NA's :28        NA's :28
```

B-cell ALL

A B-cell ALL mintákat (az ALL exprSet pData részének BT oszlopa által határozhatjuk meg). A vizsgálat célja a BCR/ABL minták összehasonlítása citogenetikailag normális mintákkal (cimkéje NEG).

a. Hozzunk létre egy exprSet objektumot, amiben csak a B-cell ALL adatok vannak. Hány minta tartozik a cytogenetikailag meghatározott csoportokba?

```
> pdat <- pData(ALL)
> table(pdat$BT)

 B B1 B2 B3 B4  T T1 T2 T3 T4
 5 19 36 23 12  5  1 15 10  2

> table(pdat$mol)

ALL1/AF4  BCR/ABL E2A/PBX1      NEG   NUP-98  p15/p16
      10       37        5       74        1        1

> subset <- intersect(grep("^B", as.character(pdat$BT)),
+ which(as.character(pdat$mol) %in% c("BCR/ABL", "NEG")))
> eset <- ALL[, subset]
> table(eset$mol)

ALL1/AF4  BCR/ABL E2A/PBX1      NEG   NUP-98  p15/p16
       0       37        0       42        0        0
```

Nem-specifikus szűrés

A chip sok génje nem expresszálódik a B-cell limfocitákban, amiket itt vizsgálunk, vagy csak nagyon kis varianciát mutatnak a mintákban.

a. Megpróbáljuk ezeket a géneket eltávolítani (helyesebben a megfelelő probeseteket) egy intenzitás szűréssel (egy gén intenzitásának 100 felett kell lennie a minták legalább 25%-ában), és egy variancia szűréssel (a log2 intenzitás interkvartilis terjedelmének legalább 0.5-nek kell lenni). Létrehozunk egy új exprSet objektumot, ami a szűrés utáni probeseteket tartalmazza. Hány probesetet kapunk?

```
> library(genefilter)
> f1 <- pOverA(0.25, log2(100))
> f2 <- function(x) (IQR(x) > 0.5)
> ff <- filterfun(f1, f2)
> selected <- genefilter(eset, ff)
> sum(selected)

[1] 2391

> esetSub <- eset[selected,]
```

Expressziós különbségek vizsgálata

Most készen állunk arra, hogy összehasonlíthassuk a BCR/ABL és a citogenetikailag normál mintákat, abból a célból, hogy meghatározzunk olyan géneket, amelyek expressziójában különbséget lehet meghatározni a két csoport között.

a. Az eltérően expresszált gének meghatározására használjunk kétmintás t-próbát. A `multtest` csomag `mt.teststat` függvénye lehetővé teszi, hogy néhány általánosan használt teszt-statisztikát használjunk egy adott adat mátrix minden sorára – részeletek a súgójában. Először számítsuk ki a névleges p-értékeket – a `pt` fügvény a t-eloszlás eloszlás függvényét adja. Egy benyomsát szerezhetünk az eltérő expressziójú gének mennyiségéről a p-érték eloszlásának hisztogramját megtekintve.

```
> library(multtest)
> labels <- as.numeric(esetSub$mol == "BCR/ABL")
> t <- mt.teststat(exprs(esetSub), classlabel = labels, test = "t.equalvar")
> pt <- 2*pt(-abs(t), df = ncol(exprs(esetSub))-2)
> hist(pt, 50)
```

b. A második lehetőség egy permutációs teszt végrehajtása mindegyik génre. Az `mt.maxT` függvény kiszámítja a permutációs p-értékeket (`rawp`) és a Wesfall-Young féle FWER-korrigált p-értékeket (`adjp`) is egyidejüleg. Hány FWER-korrigált p-érték kisebb, mint 0.1?

```
> mT <- mt.maxT(exprs(esetSub), classlabel = labels, B = 1000)
> sum(mT$adjp < 0.1)

[1] 32
```

Nagyobb számú permutáció (B) előnybe helyezendő, azonban több időt vesz igénybe. Fontos megjegyezni, hogy a függvény a p-értékeket sorbarendezve adja vissza. Az eredeti sorrendet az alábbi kóddal kaphatjuk vissza:

```
> pPermRaw <- mT$rawp[order(mT$index)]
> pWestfallYoung <- mT$adjp[order(mT$index)]
```

c. Készítsünk egy ábrát a p-értékekről a log-arány függvényében (a két csoport log-intenzitásainak átlagaiból számolt különbség) *volcano-plot* segítségével. Figyelemre méltó a volcano-plot aszimmetriája. Hasonlítsuk össze a permutációból illetve a paraméteres t-teszt eredményeként adódó p-értékek ábráit. Miért kapunk extrémebben kis p-értékeket a paraméteres teszttel?

```
> hist(pt, 50)
> logRatio <- rowMeans(exprs(esetSub)[, labels==1]) - rowMeans(exprs(esetSub)[,labels ==0])
> plot(logRatio, -log10(pt), xlab="log-ratio", ylab="-log10(p)")
> plot(log10(pt), log10(pPermRaw), main = "log10(p-érték)", xlab = "paraméteres", ylab = "permutációs t
```

d. A `multtest` csomag `mt.rawp2adjp` függvénye tartalmaz különféle szokszoros tesztelési eljárásokat. A p-érték FDR korrekciójára a Benjamin-Hochberg féle módszert használjuk. Hány gént kapun, ha FDR 0.1-et alkalmazunk.

```
> pAdjusted <- mt.rawp2adjp(pt, proc = c("BH"))
> sum(pAdjusted$adjp[,"BH"]<0.1)
```

```
[1] 171
```

Ez a függvény szintén sorbarendezve adja vissza a p-értékeket, ezért azok eredeti rendbe való visszaállításához az alábbi kódot kell futtatnunk:

```
> pBH <- pAdjusted$adjp[order(pAdjusted$index),"BH"]
```

Annotáció

a. Most szeretnénk látni, hogy mely gének a legszignifikánsabbak, és azok eredeti illetve korrigált p-értékeit, a különböző eljárások szerint. A gének jelölését a `hgu95av2` annotációs csomag tartalmazza.

```
> diff <- pAdjusted$index[1:10]
> genesymbols <- mget(geneNames(esetSub)[diff], hgu95av2SYMBOL)
> pvalues <- cbind(pt, pPermRaw, pWestfallYoung, pBH)[diff,]
> colnames(pvalues) <- c("pt", "pPermRaw", "pWestfallYoung", "pBH")
> rownames(pvalues) <- genesymbols
> print(pvalues)
```

```
                pt pPermRaw pWestfallYoung          pBH
ABL1   3.762489e-14    0.001          0.001 8.996112e-11
ABL1   4.791997e-13    0.001          0.001 5.728833e-10
ABL1   2.445693e-10    0.001          0.001 1.949217e-07
KLF9   2.785038e-08    0.001          0.001 1.664756e-05
AHNAK  2.600957e-07    0.001          0.001 1.243778e-04
ZNF467 4.741431e-07    0.001          0.001 1.889460e-04
FYN    1.058358e-06    0.001          0.001 3.410177e-04
CASP8  1.233678e-06    0.001          0.004 3.410177e-04
TUBA1  1.283630e-06    0.001          0.001 3.410177e-04
MX1    3.005298e-06    0.001          0.006 7.185668e-04
```

b. Az első 3 probeset az ABL1 génre mutat. Most nézzük meg, hogy van-e még másik probset is, ami ehhez a génhez tartozik, és esetleg nem szelektáltuk-e ki a nem-specifikus szűrés során?

```
> geneSymbols <- mget(geneNames(ALL), hgu95av2SYMBOL)
> ABL1probes <- which(geneSymbols == "ABL1")
> selected[ABL1probes]
```

```
  1635_at 1636_g_at 1656_s_at 2040_s_at 2041_i_at  39730_at
     TRUE      TRUE     FALSE     FALSE     FALSE      TRUE
```

c. Vagyis 5 olyna probeset van, ami az ABL1-hez tartozik és ki lett szűrve vagy azért mert alacsony volt az intenzitása, vagy azért mert kis varianciát mutatott. Most szeretnénk megnézni azt, hogy vajon ezek ugyancsak mutatnak-e expressziós különbséget az ABL1 génre vonatkozóan.

```
> tABL1 <- mt.teststat(exprs(eset)[ABL1probes,], classlabel = labels, test = "t.equalvar")
> ptABL1 <- 2*pt(-abs(tABL1), df = ncol(exprs(esetSub))-2)
> sort(ptABL1)
```

```
[1] 3.762489e-14 4.791997e-13 2.445693e-10 5.486259e-02 5.842693e-01
[6] 7.570959e-01
```

Azt láthatjuk, hogy az ABL1 génhez tartozó 8 probeset-ből csak 3 jelzi azt, hogy a gén különböző mértékben expresszálódnak a két csoportban. Érdekes lehet az ABL1 probeset-jeinek további elemzése, pl. a transzkriptnek melyik részén találhatók azok.

Gén ontológia

a. A BCR/ABL transzlokáció több hatása a tirozin kináz aktivitás által szabályozott. Nézzünk meg azokat a probeseteket, amelyek a GO-ban a `protein-tyrosine kinase activity` annotációval rendelkezne, animek az azonosítója a GO:0004713.

```
> gN <- geneNames(esetSub)
> tykin <- unique(lookUp("GO:0004713", "hgu95av2", "GO2ALLPROBES"))
> str(tykin)

 chr [1:268] "1635_at" "1636_g_at" "1656_s_at" "2040_s_at" "2041_i_at" ...

> sel <- (gN %in% tykin)
```

b. Most megvizsgálhatjuk azt, hogy a tirozin-kináz gének között több-e a különbözően expresszálódó, mint a többi gén között. A Fisher exakt tesztet használjuk a kontigencia táblázat tesztelésére, abból a célból, hogy vajon a különböző expressziójú gének megoszlása a két csoportban szignifikánsan különbözik-e. Miért értelmes dolog itt az igazán liberális határérték használata (0.05 nem korrigált p-érték).

```
> tab <- table(pt < 0.05, sel, dnn = c("p < 0.05", "tykin"))
> print(tab)

        tykin
p < 0.05 FALSE TRUE
   FALSE  1913   27
   TRUE    436   15

> fisher.test(tab)

        Fisher's Exact Test for Count Data

data:  tab
p-value = 0.008645
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 1.193945 4.793148
sample estimates:
odds ratio
  2.436418
```

Limma

A t-teszt elemzések a `limma` csomag függvényei segítségével is végrehajtható.

a. Először definiáljuk a design mátrixot. Egy lehetőség az intercept használata, ami minden minta génjeire vonatkozó log intenzitások átlagát reprezentálja (az első oszlop 1-esekből áll), a második oszlopban a két csoport közötti különbség van.

```
> library(limma)
> design <- cbind(mean = 1, diff = labels)
```

b. Lineáris modell illeszthetünk az összes génre a `lmFit` függvény segítségével és Empírikus Bayes moderációt végezhetünk a standard hibán az `eBayes` fügvénnyel.

```
> fit <- lmFit(esetSub, design)
> fit2 <- eBayes(fit)
> toptable(fit2, coef = "diff", adjust.method = "fdr")

              M         t      P.Value    adj.P.Val         B
134   1.1000116  9.201234 2.512298e-14 6.006904e-11 21.765333
1787  1.1525269  8.707317 2.455619e-13 2.935693e-10 19.644093
133   1.2026753  7.389925 1.035653e-10 8.254157e-08 13.999019
1890  1.7793784  6.378922 9.506229e-09 5.682349e-06  9.767965
1193  1.3487023  5.805701 1.129901e-07 5.403189e-05  7.452028
```

```
1077  1.1457106  5.388592 6.477019e-07 2.051406e-04  5.821221
1941  0.8687145  5.381017 6.682399e-07 2.051406e-04  5.792099
595   0.9961947  5.374516 6.863757e-07 2.051406e-04  5.767118
1823  0.4757069  5.265719 1.072178e-06 2.848419e-04  5.351197
1185 -1.4137658 -5.196863 1.418928e-06 3.392657e-04  5.090057
```

c. Ha összehasonlítjuk a p-értékeket a paraméteres t-teszt eredményeivel, akkor azt láthatjuk, hogy azok majd mindegyke egyezik. Mivel sok minta van az Empírikus Bayes moderáció nem olyan releváns ebben az adatállományban – a gén-specifikus variancia jól becsülhető minden génre az adatokból.

```
> plot(log10(pt), log10(fit2$p.value[,"diff"]), xlab = "két mintás t-próba", ylab = "limma")
> abline(c(0,1), col = "red")
```

ROC görbe szűrés

a. Marker géneket szeretnénk találni, amelyek speciálisan expresszálódnak a BCR/ABL transzlokációjú leukémiákban. Legalább 0.9-es specificitási szinten, szeretnénk azonosítani géneket a legjobb specificitással a BCR/ABL fenotípusra vonatkozóan. Ez kifejezhető a ROC görbe alatti részterülettel (pAUC, $t_0 = 0.1$-et választottunk). A számítási időt csökkentendő a pAUC-statisztikát csak az első 100 probesetre végezzük el.

```
> library(ROC)
> mypauc1 <- function(x) {
+     pAUC(rocdemo.sca(truth = labels, data = x, rule = dxrule.sca), t0 = 0.1)
+ }
> pAUC1s <- esApply(esetSub[1:100,], 1, mypauc1)
```

b. Gyűjtsük ki a pAUC-statisztika maximális értékét és rajzoljuk ki a hozzá tartozó ROC görbét.

```
> j <- which(pAUC1s == max(pAUC1s))
> RC <- rocdemo.sca(truth = labels, data = exprs(esetSub)[j,], rule = dxrule.sca)
> plot(RC, main = geneNames(esetSub)[j], type = 'l')
```

Többszörös tesztelés

A probléma lényege, hogy egyszerre hipotézisek ezreit teszteljük egyszerre.

- megnőtt a fals pozitívok esélye
- pl.: képzeljük el, hogy 54,000 gént vizsgálunk egy chip-en és egyik esetében sincsen különbség a két vizsgált csoport között. Azt várjuk, hogy $54000 * 0.01 = 540$ esetben a $p < 0.01$ értéket kapunk.
- az egyedi p-értéknek, pl. 0.01 már nincsen jelentése a szignifikanciára vonatkozóan

Szükséges a kapott statisztikai szignifikancia értékek korrigálása a többszörös összehasonlítás miatt.

null-hipotézis	elfogadva	elvetve	összesen
igaz	U	V	m_0
nem-igaz	T	S	m_1
	m-R	R	m

Első fajta hiba arányok

1. FWER (Family-wise error rate). Az FWER-t úgy definiáljuk, mint annak a valószínűségét, hogy legalább egy első fajta hibát (fals pozitív) kapunk

$$FWER = Pr(V > 0)$$

2. FDR (false discovery rate). Az FDR (Benjamini & Hochber, 1995) az összes elvetett hipotézis közül az első fajta hiba várható aránya:

$$FDR = E(Q) \ ,$$

ahol

$$Q = \begin{cases} V/R, & \text{ha } R > 0, \\ 0, & \text{ha } R = 0. \end{cases}$$

Az első fajta hiba arány kontrollálása

- cél: egy adott α első fajta hiba aránya, olyan eljárás alkalmazása, ami biztosítja, hogy az első fajta hiba aránya $\leq \alpha$
- az első fajta hibát az igaz és a fals nullhipotézisek adott konfigurációjára tekintette definiáljuk
- az első fajta hiba gyenge kontrollja: csak amellett a feltételezés mellet, hogy minden null-hipotézis igaz (komplett null hipotézis H_0^C)
- az első fajta hiba erős kontrollja: az igaz és hamis null-hipotézisek összes lehetséges konfigurációjára

FWER

Bonferroni korrekció

„*Bonferroni-módszer*: Voltaképp egy elv, amelynek célja, hogy a sorozatban vagy csoportosan végzett statisztikai próbák (*hipotézisvizsgálat*) esetén fölhalmozódó hibakockázatot leszorítsa. A módszer lényege, hogy az *első fajta hibát* (általában a szokásos 5%-ot) földarabolják, és próbánként csak annak töredékét használják a szignifikancia eldöntésére. Legfőbb alkalmazási területe a *többszörös összehasonlítás*."

Tegyük fel, hogy $g = 1, \ldots, m$ génre egyenként végrehajtunk hipotézis tesztelést, aminek az eredményei:
egy megfigyelt teszt-statisztika: T_g
egy nem korrigált p-érték: p_g
A Bonferroni korrigált p-értékek:

$$\tilde{p}_g = min(mp_g, 1)$$

Azon géneket kiválasztva amelyekre igaz a $\tilde{p}_g \leq \alpha$, a FWER-t kontrolláltuk α szinten. A H_0^C komplett null hipotézis mellet, ami szerint nincsen olyan gén, amin különbözően expresszálódik:

$$\begin{aligned} FWER = Pr(V > 0) &= Pr(\text{legalább egynél } \tilde{p}_g \leq \alpha | H_0^C) \\ &= Pr(\text{legalább egynél } p_g \leq \frac{\alpha}{m} | H_0^C) \\ &= \sum_{g=1}^{m} Pr(p_g \leq \frac{\alpha}{m} | H_0^C) \\ &= m * \frac{\alpha}{m} = \alpha \end{aligned}$$

Holm step-down eljárása

- módosított Bonferroni korrekció. Azonos korrekció a legkisebb p-értékre, folyamatosan csökkenő korrekció a következőkre
- sorbarendezett nem-korrigált p-értékek: $p_{r_1} \leq p_{r_2} \leq \ldots p_{r_m}$
- a FWER kontrollálása α szinten:

$$j^* = min\{j : p_{r_j} > \frac{\alpha}{m - j + 1}$$

Vessük el a H_{r_j} hipotéziseket $j = 1, \ldots, j^* - 1$ értékeknél. Így a korrigált p-értékek:

$$\tilde{p}_{r_j} = max_{k=1,\ldots,j}\{min((m - k + 1)p_{r_k}, 1)\}$$

Westfall-Young

- a minP korrigált p-értékek
- $\tilde{p}_g = Pr(min_{k=1,\ldots,m}\ P_k \leq p_g | H_0^C)$
- a $\tilde{p}_g \leq \alpha$ géneket kiválasztva a FWER-t kontrolláljuk α szinten
- a H_0^C komplett null-hipotézis mellett:

$$\begin{aligned} FWER = Pr(V > 0) &= Pr(\text{legalább egynél } \tilde{p}_g \leq \alpha | H_0^C) \\ &= Pr(min\ \tilde{p}_g \leq \alpha | H_0^C) \\ &= Pr(min\ p_g \leq c_\alpha | H_0^C) \\ &= \alpha \end{aligned}$$

- de hogyan tehetünk szert a $\tilde{p}_g$ valószínűségekre?
  - minP-korrigált p-érték becslése resampling-el
    * $b = 1, \ldots, B$-re permutáljuk a minta cimkéit
    * minden génre számítsuk ki a permutációra alapozva a nem-korrigált p_{gb} p-értékeket
    * becsűljük a $\tilde{p}_g = Pr(min_{k=1,\ldots,m}\ P_k \leq p_g | H_0^C)$ értékét az alábi szerint:

$$\#\{b : min_g\ p_{gb} \leq p_g\}/B$$

- Példa
  - tegyük fel, hogy a $p_{min} = 0.0003$ (a legkisebb nem-korrigált p-érték)
  - a randomizált adatállományok (a permutációk után), számoljuk meg, hogy milyen gyakran fordul elő a 0.0003-nál kisebb p-érték. Ha ez pl. az esetek 4%-ában fordul elő, akkor a $\tilde{p}_{min} = 0.04$
- a Westfall-Young módszer előnye: a módszer a gének közötti függőségi struktúrát bevonja a számításokba, ami sok esetben (pozitív kapcsolat a gének között) nagyobb erőt eredményez
- Holm-féle step-down eljárás: a legkisebb p-értékre ugyanaz a korrekció, majd folyamatosan csökkenő a nagyobbakra:

$$\tilde{p}_{r_j} = max_{k=1,\ldots,j}\{min_{l \in \{r_k,\ldots,r_m\}}\ P_l \leq p_{r_k} | H_0^C\}$$

- korrigált p-értékek permutációval való számítása miatt intenzív komputációt kíván
- hasonló módszer (maxT) feltételezve, hogy a T_g teszt-statisztika egyenletes eloszlású a null-hipotézis mellett, a p_g-t T_g-vel a min-t pedig max-al helyettesítjük. Kevésbé számítás-intenzív.
- mindegyik módszer elérhető a Bioconductor `multtest` csomagban, a minP-re egy gyors algortimussal

A FWER konzervatív kritérium, ennek következtében számos érdekes gént elveszíthetünk.

FDR

- rendezzük sorba a nem korrigált p-értékeket: $p_{r_1} \leq p_{r_2} \leq \ldots p_{r_m}$
- a $FDR = E(V/R)$ kontrolja α szinten, legyen

$$j^* = max\{j : p_{r_j} \leq (j/m)\alpha\}$$

  Vessük el a H_{r_j} hipotéziseket a $j = 1, \ldots, j^*$ értékekre.
- Független teszt-statisztikákra és függőségek bizonyos típusaira. Ha sok gén expressziója különböző, akkor konzervatívan viselkedik. A Bioconductor `multtest` csomagban megtalálható.

A FDR becslése

Elmélet: A T_g teszt-statisztika tesztelése a kiválasztott cut-off értéktől függően becsüljük a fals pozitív gének arányát permutáció segítségével.

1. becsüljük a nem differenciáló gének m_0 számát
2. becsüljük az $E(V_0)$ várható fals pozitív számot a komplett null-hipotézis esetén, resampling-el. Akkor $\widehat{E(V)} = \frac{\hat{m}_0}{m}\widehat{E(V_0)}$ (mivel csak a nem különböző gének lehetnek fals pozitívok)
3. becsüljük a $FDR = E(V/R)$-t a $\widehat{E(V)}/R$ segítségével

A nem differenciáló gének m_0 számának becslése.

- vegyük szemügyre a p-értékek eloszlását: a $p > 0.5$ értékkel rendselkező gének valószínűleg nem különbözően expresszálódnak
- mivel a nem differenciáló gének uniform eloszlást $[0, 1]$ kell, hogy kövessenek, a $2 * \#\{g | p_g > 0.5\}$ értéként tekinthetjük úgy, mint m_0 becslése

FDR becslése.

- permutációval $b = 1, \ldots, B$-re számítsuk ki a T_{gb} teszt-statisztikákat a komplett null-hipotézisnek megfelelően
- valamely t_0 teszt-statiszika határértéket véve számoljuk ki a V_b gén-számot, amelyekre igaz a $T_{gb} > t_0$ (a fals pozitívok száma)
- az FDR becslése a V_b átlagán alapszik. Ugyanakkor a V_b kvantilise szintén érdekes lehet, mivel a fals pozitívok aktuális arányának mutatójaként sokkal nagyobb lehet, mint az átlag.
- az eljárás bevonja a gének közötti függőségi struktúrát is

FWER vagy FDR

- válasszuk az FWER kontrollját, ha a kigyűjtött gének esetén nagy megbízhatóságra van szükség. A nagy számú tesztelés miatt az erő csökken: sok különbözően expresszált gén nem lesz szignifikáns.
- ha a fals pozitívok bizonyos arányban való előfordulása elfogadható: az FDR-en alauló eljárások flexibilisebbek; a kutató meg tudja határozni, hogy mennyi gént szelektáljon, ami praktikus megfontolásokon nyugodhat.

Előszűrés

- hasznos lehet, de
- a szűrési feltételeket az elemzés előtt meg kell határozni
- a feltételeknek függetlennek kell lennie a null hipotézis eloszlásától

Mi egyebet használhatunk?

- kis elemszámú minták esetén használhatjuk a regularizált t-statisztikát. A gén-specifikus variancia növekszik konstans hozzáadásával (pl. SAM, limma).
- a Bioconductor `globaltest` csomagja tartalmaz egy módszert, ami lehetővé teszi génycsoportok tesztelésére, abból a célból, hogy tartalmaznak-e differenciáló géneket

FDR

Vegyünk egy példát, amiben egyszerre m (null) hipotézist tesztelünk, amelyek közül az m_0 igaz. R az elutasított hipotézisek száma. A táblázat a hagyományos formában foglalja összes a helyzeteket.

null-hipotézis	elfogadva	elvetve	összesen
igaz	U	V	m_0
nem-igaz	T	S	$m - m_0$
	m-R	R	m

Tegyük fel, hogy a m hipotéziseket előre tudjuk. Az R egy megfigyelhető valószínűségi változó; az U, V, S és T nem megfigyelhető valószínűségi változó. Ha minden egyedi null-hipotézist külön-külön tesztelünk α szinten, akkor $R = R(\alpha)$ növekszik α-ban. Kisbetűket használunk a megvalósult értékekre.
Ezen valószínűségi változók alapján a PCER $E(V/m)$, a FWER pedig $P(V \geq 1)$. Az egyes hipotézisek α szinten egyenkénti tesztelése garantálja, hogy $E(V/m) \leq \alpha$. Az egyes hipotézisek α/m szinten egyenkénti tesztelése garantálja, hogy $P(V \geq 1) \leq \alpha$.

A False Discovery Rate (FDR) definíciója

A hibák megoszlása, ami a hibásan elutasított null hipotézisekből adódik, a $Q = V/(V + S)$ valószínűségi változón keresztűl vizsgálható – a hibásan elvetett null-hipotézisek aránya az összes elvetett null-hipotézisen belül. Természetesen, $Q = 0$ ha $V + S = 0$, ha nincs fals elvetésből származó hiba. Q nem-megfigyelt valószínűségi változó, ahogy nem ismerjük v és s értékét sem, így $q = v(v + s)$. Definiáljuk a FDR Q_c-t, mint a várható Q-t,

$$Q_c = E(Q) = E\{V/(V + s)\} = E(V/R)$$

Ennek az aránynak két tulajdonságát könnyű megmutatni, mégis nagyon fontosak.

1. Ha minden null-hipotézis igaz, az FDR megegyezik a FWER-rel: ebben az esetben $s = 0$ és $v = r$, így ha $v = 0$, akkor $Q = 0$, és ha $v > 0$, akkor $Q = 1$, így $P(V \geq 1) = E(Q) = Q_c$. Ennek következtében az FDR kontrollja magában foglalja az FWER kontrollját is.

2. Ha $m_0 < m$, akkor az FDR kisebb vagy egyenlő az FWER-rel: ebben az esetben, ha $v > 0$, akkor $v/r \leq 1$, ami azt eredményezi, hogy $\chi_{(V \geq 1)} \geq Q$. Mindkét oldalon feltételezésekkel élve a $P(V \geq 1) \geq Q_c$, és a kettő igen különböző lehet. Mint eredmény az eljárás az FWER mellett kontrollálja az FDR-t is. Ugyanakkor ha egy eljárás csak a FDR-t kontrollálja, kisebb szigorúság mellet, nagyobb erővel számolhatunk. Különösen, ha a nem-igaz null-hipotézisek száma növekszik, S növekszik, és így a hiba arányok közötti különbség növekszik. Végeredményben a lehetséges erő növekedés nagyobb, ha több nem-igaz hipotézis van.

A FDR kontrollálási eljárás

Tegyük fel, hogy $H_1, H_2, \ldots, H_m$ hipotézist tesztelünk a nekik megfelelő $P_1, P_2, \ldots, P_m$ p-értékekre alapozva. Legyen $P_{(1)} \leq P_{(2)} \leq \cdots \leq P_{(m)}$ a p-értékek sorbarendezett sorozata, és $H_{(i)}$ null-hipotézishez a $P_{(i)}$ tartozik. Definiáljuk a következő Bonferroni típusú többszörös tesztelési eljárást:

$$\text{legyen } k \text{ a legnagyobb } i, \text{ amire } P_{(i)} \leq \frac{i}{m} q^*$$

ekkor elvetjük mindegyik $H_{(i)}$ $i = 1, 2, \ldots, k$ hipotézist.

Theorem I. Független teszt statisztikák és a fals null-hipotézisek egyes konfigurációja esetén a fenti eljárás kontrollálja az FDR-t a q^* szinten.

Lemma. Valamely független $0 \leq m_0 \leq m$ p-értékekre a megfeleő igaz nullhipotézishez tartozva, és valamely $m_1 = m - m_0$ p-értékekre, amelyek a fals null-hipotézishez tartoznak, a fenti többszörös tesztelési eljárás az alábbi egyenlőtlenségben foglalható össze:

$$E(Q|P_{m_{0+1}} = p_1, \ldots, P_m = p_{m_1}) \leq \frac{m_0}{m} q^*$$

Most tegyük fel, hogy $m_1 = m - m_0$ része a hipotéziseknek fals. Mindazonáltal a $P_1'', \ldots, P_{m_1}''$ joint eloszlása, ami a fals hipotézisekhez tartozik, az alábbi egyenlőtlenségben integrálható:

$$E(Q) \leq \frac{m_0}{m} q^* \leq q^*$$

és így az FDR kotrollálva van.

Megjegyzések.

FDR példa

Tegyük fel, hogy 15 összehasonlításból származó *p-értékek* alapján szeretnénk megfogalmazni következtetéseket. Amelyek alapján, ha a szignifikancia szintet 0.05-ben határozzuk meg, akkor korrekció nélkül 9 esetben vetjük el a null-hipotézist.

```
> p1 <- c(0.0001, 0.0004, 0.0019, 0.0095, 0.0201, 0.0278, 0.0298, 0.0344,
+ 0.0459, 0.3240, 0.4262, 0.5719, 0.6528, 0.7590, 1.000)
> szint <- 0.05
> idx <- which(p1 <= szint)
> orig.p <- p1[idx]
> orig.p
```

```
[1] 0.0001 0.0004 0.0019 0.0095 0.0201 0.0278 0.0298 0.0344 0.0459
```

Ha az FWER-t kontrolláljuk, a Bonferroni megközelítésben, a $0.05/15 = 0.0033$ alkalmazásával 3 olyan *p-értéket* találunk, ami kielégíti a feltételeket:

```
> FWER <- szint/length(p1)
> idx <- which(p1 <= FWER)
> adj.p <- p1[idx]
> adj.p
```

```
[1] 0.0001 0.0004 0.0019
```

Az FDR kontrolláló eljárást alkalmazva, amennyiben $q^* = 0.05$ a *p-értékek* sorozatából mindegyik $p_{(i)}$ értéket összehasonlítjuk a $q^* i/15$. A *p-értékek* sorában az utolsó a 4., ami kielégíti a feltételeket:

$$p_{(4)} = 0.0095 \leq \frac{4}{15} 0.05 = 0.013$$

Vagyis a 15 null-hipotézisből 4-et vetünk el.

```
> p2 <- sort(p1)
> p3 <- (1:length(p2)/length(p2))*szint
> idx <- which(p2<=p3)
> ptabla <- data.frame(orig.p=p2[idx], adj.p=round(p3[idx],3))
> ptabla

  orig.p adj.p
1 0.0001 0.003
2 0.0004 0.007
3 0.0019 0.010
4 0.0095 0.013
```

Gének szűrése, sorbarendezése

```
library(affy)
library(SpikeInSubset)
data(spikein95)
pd <- data.frame(population = c(1,1,1,2,2,2), replicate = c(1,2,3,1,2,3))
rownames(pd) <- sampleNames(spikein95)
v1 <- list(population = "1 is control, 2 is treatment", replicate = "arbitrary numbering")
phenoData(spikein95) <- new("phenoData", pData = pd, varLabels = v1)
eset <- rma(spikein95)
e <- exprs(eset)
dim(e)
pData(eset)
Index1 <- which(eset$population == 1)
Index2 <- which(eset$population == 2)
```

A normalizáció után az *AffyBatch* objektumból egy expressziós állományunk lesz, amiben minden j génre, minden i tömbben, mined k populációra ($k = 1, 2$) van egy x_{ijk} mérési értékünk. A sorbarendezés egyik konvencionális megoldása, hogy a differenciáló expresszió szintjének átlagát kvantifikáljuk. Egy naiv első lehetőség az átlagos fold-change használata:

$$d_j = \bar{x}_{j2} - \bar{x}_{j1}$$

A d számítására használhatjuk a `rowMeans` függvényt, ami sokkal gyorsabb mint ha az `apply` függvényt használnánk:

```
d <- rowMeans(e[, Index2]) - rowMeans(e[, Index1])
```

Különböző szerzők azt mondják, hogy a fold-change mérések variabilitása a kérdéses gének általános expressziójától függ. Ez azt jelenti, hogy a d által mutatott magas értékeket az összes expressziót jellemző mértékkel kell összevetni. Egy egyszerű példa az átlagos log expresszió:

```
a <- rowMeans(e)
```

Az MA-plot a d értékeit a függvényében mutatja. Az y tengelyt úgy limitáljuk, hogy az kisebb legyen, mint 1 log fold-change (a fold-change kisebb, mint 2).

```
library(geneplotter)
png(filename = "MA.png", width = 800, height = 600)
smoothScatter(a,d, ylim=c(-1,1), nbin = 1280, xlab ="átlag", ylab="különbség")
dev.off()
```

Ezt azért csináljuk, mivel az elemzésben azokra génekre vagyunk kíváncsiak, amelyek fold-change nagyobb, mint 2. A következő kóddal gyűjthetjük ki ezeket:

```
sum(abs(d)>1)
```

Összefoglaló statisztikák és tesztek a sorbarendezéshez

Az ábrán látható, hogy a variancia hogyan csökken a növekedésével. Az szintén látható, hogy különböző gének különböző variancia szinttel rendelkeznek. Ezek alapján a gének sorbarendezésének folyamatában figyelembe kellene vennünk a varianciát? Egy némszerű megoldása a kérdésnek, a t-statisztika. A t-statisztika egy hányados, ami a d becslést hasonlítja össze a mintára alapozott csoporton belüli standard hiba becsléssel.

$$s_j^2 = \frac{1}{2}\sum_{i=1}^{n}(x_{ij2} - \bar{x}_{j2})^2/n + \frac{1}{2}\sum_{i=1}^{n}(x_{ij1} - \bar{x}_{j1})^2/n$$

A d_j/s_j t-statisztikák kiszámolásához a `genefilter` csomag `rowttests` függvényét használhatjuk.

```
library(genefilter)
tt <- rowttests(e, factor(eset$population))
```

Megváltozik vajon a gének sorrendje, ha a t-tesztet használjuk és nem pusztán a fold-change-t? A volcano plot egy jó eszköz a két mennyiség együttes vizsgálatára. A volcano a p-értket (pontosabban a p-érték $-log_{10}$ transzformáltját) ábrázolja a d hatás mértékének függvényében. Az egyszerűség kedvéért feltételezzük, hogy a t-statisztika t-eloszlást követ. A volcano plot a következők alapján hozható létre:

```
lod <- -log10(tt$p.value)
plot(d, lod, cex=0.25, main = "B) Volcano plot for t-test")
smoothScatter(d, lod, cex=0.25, main = "B) Volcano plot for t-test", xlim=c(-1,1),nbin = 1280)
```

Az ábra azt mutatja, hogy a t-teszt és az átlagos log fold-change különböző válaszokat ad. Az ábrához hozzáadjuk a top 25 gént mind a fold-change, mind pedig a legkisebb p-érték alapján, az előzőt kék gyémánttal, az utóbbit piros körökkel jelöljük. Az alábbi kód segít ennek a kivitelezésében:

```
o1 <- order(abs(d), decreasing=T)[1:25]
o2 <- order(abs(tt$statistic), decreasing = T)[1:25]
o <- union(o1, o2)
plot(d[-o], lod[-o], cex=0.25, xlim = c(-1,1), ylim = range(lod), main="C) Close up of B)")
smoothScatter(d[-o], lod[-o], xlim = c(-1,1), ylim = range(lod), main="C) Close up of B)", nbin = 1280)
points(d[o1], lod[o1], pch=18, col="blue")
points(d[o2], lod[o2], pch=1, col="red")
```

Aránylag nagy eltérés van. A lehetséges magyarázatok:

- Néhány gén nagyobb varianciával rendelkezik, mint mások. A nagy varianciával rendelkező gének, amelyek nem expresszálódtak különbözőképpen nagyobb eséllyel rendelkeznek nagy log fold-change-el. Mivel a t-statisztika a varianciát használja, ezek nem kapnak kicsi p-értket.
- Mindössze 3 méréssel csoportonként a hatás mértékének standard hibájának becslése nem stabil, és néhány gén csak azért rendelkezik alacsony p-értékkel mert a t-statisztika nevezője nagyon kicsi.

A C ábrán látható eredményekmindkettő módon magyarázhatók.
A t-statisztikát vagy a fold change-ot hasznéljuk a gének sorbarendezésére? Mindkettőnek vannak előnyei és hátrányai. Ahogy említettük a t-tesztnél az a probléma, hogy nincs elég adat a variancia becslésére. Több szerző is javasol alternatív statisztikákat, amelyek az összes géntől kölcsönöznek erőt a geénspecifikus variancia stabilabb becslésére. Ezeket a statisztikákat módosított t-statisztikának nevezzük. Mivel csökkentik a nagy értékek valószínűségét ezért ezeket szokták nevezni *penalized*, *attenuated* vagy *regularized* t-tesztnek is. Nagyon sok módosított t-teszt van, egy ezek közül a SAM (Tusher et al., 2001) által használt statisztika.
Egy péla a módosított t-statisztikára a Smyth (2004) által leírt és a `limma` csomagban elérhető. Ezt moderált t-statisztikának hívják. A módszer empírikus Bayes megközelítséen alapszik, és az alábbi kóddal valósítható meg:

```
library(limma)
design <- model.matrix(~factor(eset$population))
fit <- lmFit(eset, design)
ebayes <- eBayes(fit)
lod <- -log10(ebayes$p.value[,2])
mtstat<- ebayes$t[,2]
o1 <- order(abs(d),decreasing=TRUE)[1:25]
o2 <- order(abs(mtstat),decreasing=TRUE)[1:25]
o <- union(o1,o2)
```

```
# plot(d[-o],lod[-o],cex=.25,xlim=c(-1,1),ylim=c(0,4), main="D) Volcano plot for moderated $t$-test")
smoothScatter(d[-o], lod[-o], xlim = c(-1,1), ylim = c(0,4), main="D) Volcano plot for moderated t-test
points(d[o1],lod[o1],pch=18,col="blue")
points(d[o2],lod[o2],pch=1,col="red")
```

A D ábra szemlélteti az empírikus Bayes megközelítés hatását a t-statisztikára, abban az értelemben, hogy nem ad magas rangot csak azért géneknek, mert alacsony a minta varianciája. Nézzük meg, hogy most kevesebb olyan génünk van alapcsony p-értékkel (nagy érték az y-tengelyen), amelyeknek ugyanakkor nagyon kicsi a log fold-change értéke (az x-tengelyen közel vannak a 0-hoz).

Határérték megválasztása

Összehasonlítás

Az elemzésből látható, hogy a moderált t-statisztika jobb megoldásnak tűnik, mint a log fold-change, vagy a t-teszt. Mivel mindegyik spike-in kísérlet adatállományát használtuk, értékelni tudjuk a 3 versengő statisztikát. Először kigyűjtjük a gének nevét a `spikein95` adatállományból. Az eredeti `phenoData` használatával ezt meg tudjuk tenni.

```
data(spikein95)
spikedin <- colnames(pData(spikein95))
spikedIndex <- match(spikedin,geneNames(eset))
```

A kísérletben összesen 16 génnek kellene különböző expressziót mutatnia. Ez azt jelenti, hogy a tökéletes sorbarendező módszerrel a 16 gén 1-től 16-ig sorrendben lenne. Az alábbi kód eredményéből úgy tűnik, hogy a moderált t-statisztika jobb rendező, mint a másik kettő módszer:

```
d.rank <- sort(rank(-abs(d))[spikedIndex])
t.rank <- sort(rank(-abs(tt$statistic))[spikedIndex])
mt.rank <- sort(rank(-abs(mtstat))[spikedIndex])
ranks <- cbind(mt.rank,d.rank,t.rank)
rownames(ranks) <- NULL
ranks
      mt.rank d.rank t.rank
 [1,]       1      1      1
 [2,]       2      2      3
 [3,]       3      3      5
 [4,]       6      4      8
 [5,]       7      5     13
 [6,]       8      6     17
 [7,]       9      9     19
 [8,]      10     11     27
 [9,]      11     12     28
[10,]      12     14     29
[11,]      16     16     45
[12,]      25     53     70
[13,]      28     86     71
[14,]      48    226     93
[15,]      77    331    390
[16,]     465    689    900
```

Significance analysis of microarrays

A gén specifikus fluktuáció kezelésére definiáltak egy statisztikát, ami a gén expresszió változásának az adott gén adatainak szórásához viszonyított arányán alapszik. A gén expresszió $d(i)$ „relatív különbsége”:

$$d(i) = \frac{\bar{x}_I(i) - \bar{x}_U(i)}{s(i) + s_0}$$

ahol $x_I(i)$ és $x_U(i)$ az i gén expressziójának átlagos szintje az I és U állapotokban. Az $s(i)$ „gén-specifikus szórás” az ismételt expressziós mérések szórása:

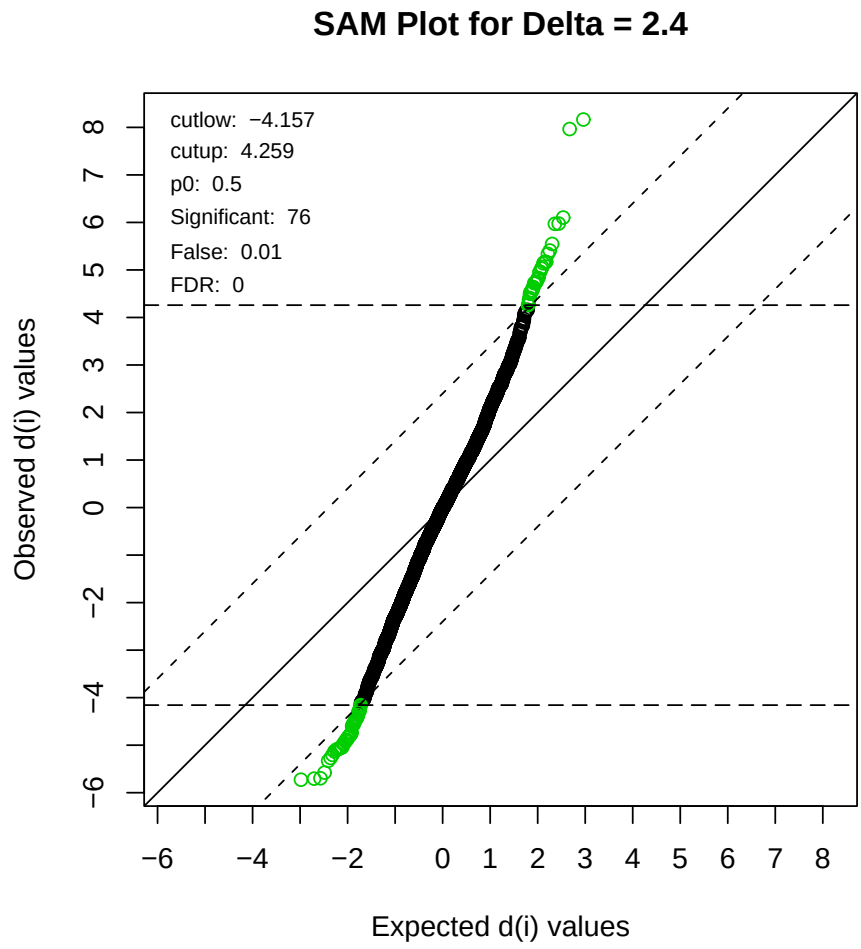
$$s(i) = \sqrt{a \left\{ \sum_m [x_m(i) - \bar{x}_I(i)]^2 + \sum_n [x_n(i) - \bar{x}_U(i)]^2 \right\}}$$

ahol $\sum_m$ és $\sum_n$ az I és U állapotokban az expressziós mérések összege, $a = (1/n_1 + 1/n_2)/(n_1 + n_2 - 2)$, és n_1 és n_2 a mérések száma az I és U állapotokban (a mi esetünkben 4).
Ahhoz, hogy összehasonlíthatók legyenek a $d(i)$ értékek sz összes génen, $d(i)$ eloszlásának függetlennek kell lennie a gén expressziós szintjétől. Alacsony expressziós szinteken a $d(i)$ varianciája nagy lehet az alacsony $s(i)$ értékek miatt. Annak érdekében, hogy biztosítsuk a $d(i)$ varianciájának függetlenségét a gén expressziótól, a nevezőhöz egy kis konstanst, a s_0 lett hozzáadva. A $d(i)$ variancia koefficiensét mint $s(i)$ függvényét számítottuk egy az adatokon mozgó ablakban. Az s_0 értékét úgy választottuk ki, hogy csökkentsük a variation koeficienst. A cikkben használt adatokra vonatkozóan $s_0 = 3.3$. A $d(i)$ $s(i)$ függvényében ábrázolt szórásdiagram a fig2.
...
Habár az A és B hibridizációkból számított relatív különbség biztosít egy kontrollt a random fluktuációhoz, további kontrolokra lenne szükség az IR hatásának statisztikai szignifikanciájának megfelelően. Ahelyett, hogy újabb kísérleteket végeztünk volna, ami drága és labor intenzív, generáltunk nagy számú kontrollt a két négyes csoport hibridizációjának permutációjából számított relatív differenciákkal. A két sejtvonal közötti különbségekből adódó zavaró hatások kiküszöbölésére az 1. és a 2. sejtvonalra kiegyenlített permutációt végeztünk.... Abból a célból, hogy szignifikáns változásokat találjunk a gén expresszióban, a géneket rangsoroltuk a $d(i)$ értékeknek megfelelően, $d(1)$ volt a legnagyobb, míg $d(i)$ a legkisebb. Mind a 36 permutáció esetén, a p permutációban a $d_p(i)$ relatív különbséget kiszámoltuk és az előzőek szerint sorbarendeztük. A $d_E(i)$ várható relatív különbséget a 36 kiegyensúlyozott permutáció átlagaként határoztuk meg: $d_E(i) = \sum_p d_p(i)/36$.
A lehetséges expressziós változások megtalálását célzóan egy szórásdiagrammot készítettünk, amin a megfigyelt $d(i)$ relatív különbséget a $d_E(i)$ várható relatív különbség függvényében ábrázoltuk. A gének túlnyomó többségében a $d(i) \cong d_E(i)$, viszont a gének egy része eltér a $d(i) = d_E(i)$ egyenestől, a Δ határértéken is túl helyezkednek el az ezeket reprezentáló pontok. Például, ha $\Delta = 1.2$, akkor 46 gént találunk „szignifikánsnak". A 46 gén vizsgálható a $d(i)$ vs. $s(i)$ ábrán, vagy az $\bar{x}_I(i)$ köbgyöke vs. $\bar{x}_U(i)$ ábrán. A $d(i)$ által meghatározott gének nem szükségszerűen a legnagyobb változást mutató gének. A SAM által meghatározott gének közül a fals szignifikánsok számánk meghatározására, horizontális hatérértékeket definiáltunk, ennek értékei a legkisebb $d(i)$ a szignifikánsan növekedett gének között valamint a legkisebb negatív $d(i)$ az elnyomott szignifikáns gének között. A fals signifikáns gének száma minden egyes permutációban meg lett számolva, mégpedig azok a gének voltak, amik a határértéket átlépték. A fals szignifikánsok számának becslése a 36 permutáció során megszámoltak átlaga volt. A $\Delta = 1.2$ esetén a permutált adatokból származó átlag 8.4 fals szignifikáns gént adott, ami alapján a 46 szignifikánsnak talált génhez viszonyítva, az FDR becsült értéke 18%. Ahogy a Δ csökkenésével a szignifikáns gének száma növekszik, azonban ez az FDR növekedésével is együtt jár. A $\Delta = 0.6,\ 0.9,\ 1.2$ esetén FDR 45, 35, 28%.

Az FDR ismeretében a p-értékeket korrigálhatjuk pl. a Benjamini – Hochberg módszerrel

Példa

```
> library(siggenes)
> library(multtest)
> data(golub)
> sam.out <- sam(golub, golub.cl, rand = 123, gene.names = golub.gnames[,
+     3])
> sam.out

SAM Analysis for the Two-Class Unpaired Case Assuming Unequal Variances

   Delta  p0   False Called      FDR
1    0.1 0.5 2424.77   2739  0.44276
2    0.7 0.5  262.21   1248  0.10508
3    1.3 0.5   12.11    507  0.01195
4    1.8 0.5    0.74    210  0.00176
5    2.4 0.5    0.01     76 6.58e-05
6    3.0 0.5       0     15        0
7    3.6 0.5       0      5        0
8    4.1 0.5       0      2        0
9    4.7 0.5       0      2        0
10   5.3 0.5       0      0        0

> plot(sam.out, 2.4)
```




```
> summary(sam.out, 2.4)

SAM Analysis for the Two-Class Unpaired Case Assuming Unequal Variances

s0 = 0.0584  (The 0 % quantile of the s values.)

 Number of permutations: 100

 MEAN number of falsely called genes is computed.

 Delta: 2.4
 cutlow: -4.157
 cutup: 4.259
 p0: 0.5
 Significant Genes: 76
 Falsely Called Genes: 0.01
 FDR: 6.58e-05


Genes called significant (using Delta = 2.4):

    Row d.value stdev     rawp  q.value R.fold             Name
1   829    8.17 0.296        0        0  7.277        M27891_at
2  2124    7.96 0.178        0        0  3.395        X95735_at
3  2600    6.10 0.191        0        0  2.669      L09209_s_at
4  2664    5.98 0.392        0        0  4.723      Y00787_s_at
5   766    5.97 0.173        0        0  2.497        M16038_at
6  2489   -5.73 0.215        0        0  0.345 U22376_cds2_s_at
```

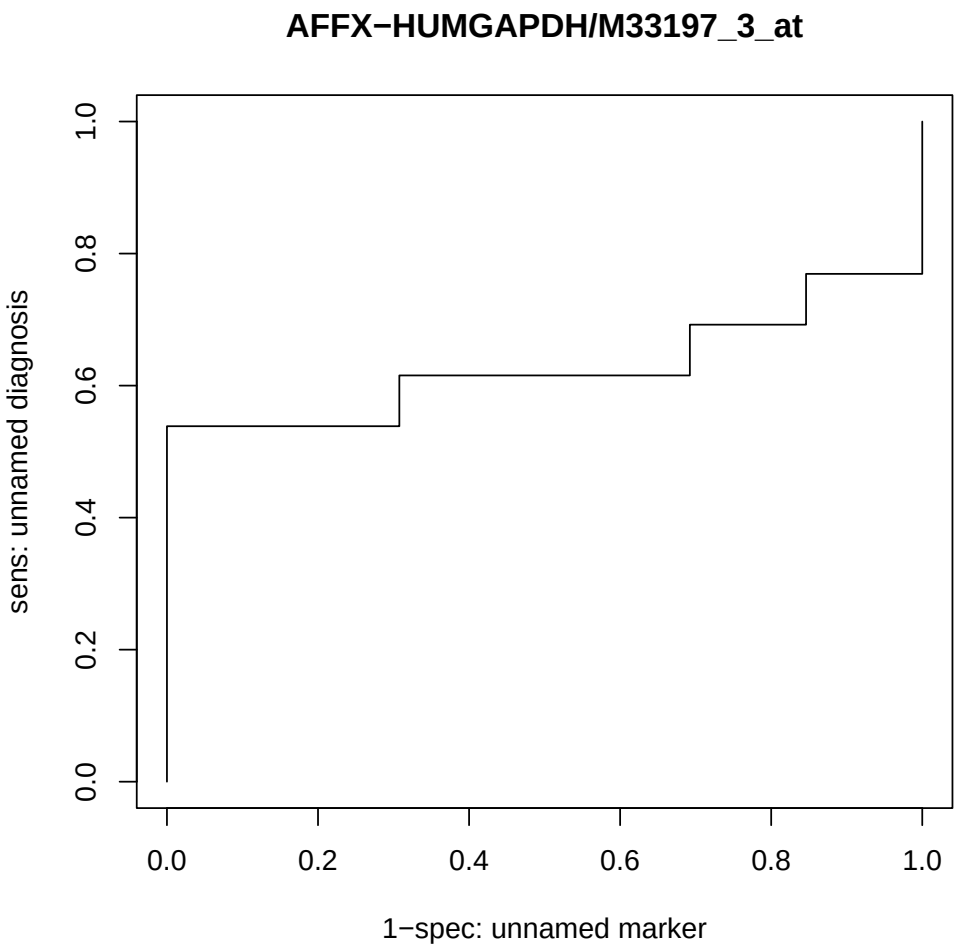
7	717	-5.70	0.207	0	0	0.345	L47738_at
8	1995	-5.70	0.193	0	0	0.374	X74262_at
9	2939	-5.58	0.165	0	0	0.413	M31523_at
10	2663	5.55	0.418	0	0	5.455	M28130_rna1_s_at
11	378	5.41	0.302	0	0	4.423	D88422_at
12	1778	5.34	0.222	0	0	2.782	X07743_at
13	523	-5.33	0.203	0	0	0.378	J05243_at
14	1037	-5.27	0.170	0	0	0.424	M91432_at
15	2065	-5.20	0.357	0	0	0.121	X82240_rna1_at
16	1911	5.17	0.190	0	0	2.357	X62654_rna1_at
17	1413	5.17	0.281	0	0	3.293	U46499_at
18	808	5.14	0.182	0	0	2.428	M23197_at
19	2386	-5.13	0.135	0	0	0.486	Z15115_at
20	1030	-5.09	0.256	0	0	0.236	M89957_at
21	1334	-5.08	0.194	0	0	0.380	U32944_at
22	738	-5.07	0.292	0	0	0.267	M11722_at
23	1665	5.06	0.168	0	0	2.083	U82759_at
24	394	-5.05	0.143	0	0	0.500	HG1612-HT1612_at
25	1162	-5.05	0.324	0	0	0.225	U05259_rna1_at
26	2198	5.03	0.152	0	0	2.064	Y12670_at
27	2851	-4.97	0.138	0	0	0.482	U72936_s_at
28	2921	4.96	0.225	0	0	2.320	M19045_f_at
29	2266	-4.96	0.265	0	0	0.320	Z69881_at
30	896	4.95	0.094	0	0	1.639	M55150_at
31	377	-4.91	0.314	0	0	0.171	D88270_at
32	2702	-4.90	0.158	0	0	0.480	M31211_s_at
33	1042	-4.89	0.199	0	0	0.424	M92287_at
34	2645	-4.85	0.148	0	0	0.460	M12959_s_at
35	1829	4.83	0.266	0	0	2.440	X17042_at
36	1909	-4.82	0.226	0	0	0.360	X62535_at
37	2801	-4.82	0.139	0	0	0.498	U26266_s_at
38	2813	4.81	0.238	0	0	2.637	X85116_rna1_s_at
39	703	-4.79	0.122	0	0	0.505	L41870_at
40	1882	-4.77	0.213	0	0	0.378	X59350_at
41	1834	-4.77	0.121	0	0	0.535	X51521_at
42	566	4.75	0.167	0	0	2.011	L08246_at
43	839	-4.73	0.193	0	0	0.392	M29696_at
44	2920	4.73	0.232	0	0	2.301	J03801_f_at
45	1754	4.73	0.202	0	0	2.167	X04085_rna1_at
46	937	4.72	0.135	0	0	1.888	M63138_at
47	1069	4.64	0.367	0	0	4.162	M96326_rna1_at
48	1448	4.63	0.115	0	0	1.745	U50136_rna1_at
49	2002	-4.60	0.114	0	0	0.560	X74801_at
50	2670	4.57	0.352	0	0	4.961	M27783_s_at
51	746	-4.57	0.213	0	0	0.424	M13792_at
52	2829	-4.57	0.156	0	0	0.448	U49020_cds2_s_at
53	803	4.56	0.220	0	0	2.365	M22960_at
54	2459	-4.55	0.307	0	0	0.190	L33930_s_at
55	2922	4.54	0.245	0	0	2.337	X14008_rna1_f_at
56	2734	4.53	0.360	0	0	2.540	M63438_s_at
57	3046	-4.51	0.148	0	0	0.492	U29175_at
58	1901	4.50	0.226	0	0	2.210	X61587_at
59	515	-4.49	0.322	0	0	0.361	J04615_at
60	345	-4.47	0.198	0	0	0.426	D87078_at
61	1817	-4.44	0.119	0	0	0.535	X15949_at
62	1086	-4.43	0.133	0	0	0.534	S50223_at
63	2950	-4.41	0.159	0	0	0.475	U27460_at
64	1009	4.41	0.482	0	0	7.689	M84526_at
65	1524	-4.41	0.113	0	0	0.583	U62136_at
66	1907	4.37	0.219	0	0	2.495	X62320_at
67	1598	-4.36	0.138	0	0	0.495	U73737_at

```
68 2289   -4.31 0.152        0        0  0.530        D38073_at
69 1585   -4.30 0.170        0        0  0.482        U72342_at
70 1920   -4.29 0.141        0        0  0.524        X63753_at
71 1883   -4.28 0.252        0        0  0.413        X59417_at
72  968    4.28 0.196        0        0  1.946        M69043_at
73 2750    4.26 0.286        0        0  2.961      M83652_s_at
74 2020   -4.23 0.191 3.28e-06 6.67e-05  0.477        X76648_at
75 1060   -4.18 0.125 3.28e-06 6.67e-05  0.537        M94633_at
76  329   -4.16 0.152 6.56e-06 0.000116  0.524        D86967_at
```

ROC

Példa

```
> library(ROC)
> library(Biobase)
> data(sample.exprSet.1)
> eset <- sample.exprSet.1
> labs <- eset$cov1 - 1
> mypauc1 <- function(x) {
+     pAUC(rocdemo.sca(truth = labs, data = x, rule = dxrule.sca),
+         t0 = 0.1)
+ }
> pAUC1s <- esApply(eset[1:100, ], 1, mypauc1)
> j <- which(pAUC1s == max(pAUC1s))
> j

AFFX-HUMGAPDH/M33197_3_at
                       48

> max(pAUC1s)

[1] 0.05384615

> RC <- rocdemo.sca(truth = labs, data = exprs(eset)[j, ], rule = dxrule.sca)

> plot(RC, main = geneNames(eset)[j], type = "l")
```



Források

1. *http://www.economia.unimi.it/projects/marray/2006/*
2. *http://www.bepress.com/bioconductor/*
3. *http://cran.r-project.org/manuals.html*
4. Robert Gentleman, Vince Carey, Wolfgang Huber, Rafael Irizarry, Sandrine Dudoit. Bioinformatics and Computational Biology Solutions Using R and Bioconductor. Springer, 2005, ISBN: 0-387-25146-4 *http://www.bioconductor.org/pub/docs/mogr/*
5. *http://affycomp.biostat.jhsph.edu/ http://compdiag.molgen.mpg.de/ngfn/docs/2006*
6. Y. Benjamini and Y. Hochberg (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society B, Vol. 57, 289–300.
7. S. Dudoit, J.P. Shaffer, J.C. Boldrick (2003). Multiple hypothesis testing in microarray experiments. Statistical Science, Vol. 18, 71– 103.
8. J.D. Storey and R. Tibshirani (2003). SAM thresholding and false discovery rates for detecting differential gene expression in DNA microarrays. In: The analysis of gene expression data: methods and software. Edited by G. Parmigiani, E.S. Garrett, R.A. Irizarry, S.L. Zeger. Springer, New York.
9. V.G. Tusher et al. (2001). Significance analysis of microarrays applied to the ionizing radiation response. PNAS, Vol. 98, 5116– 5121.
10. M. Pepe et al. (2003). Selecting differentially expressed genes from microarray experiments. Biometrics, Vol. 59, 133–142.

11. Lakner Géza, Gachályi Béla, Singer Júlia. Klinikai farmakológia. Biostatisztikai fogalomtárral. SpringMed Kiadó, Budapest, 2005