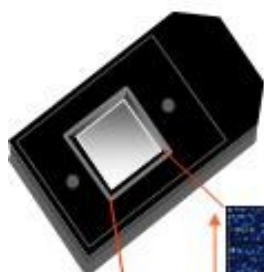


Kapitola II.1.2

Vznik a charakter dat -> Affymetrix čipy

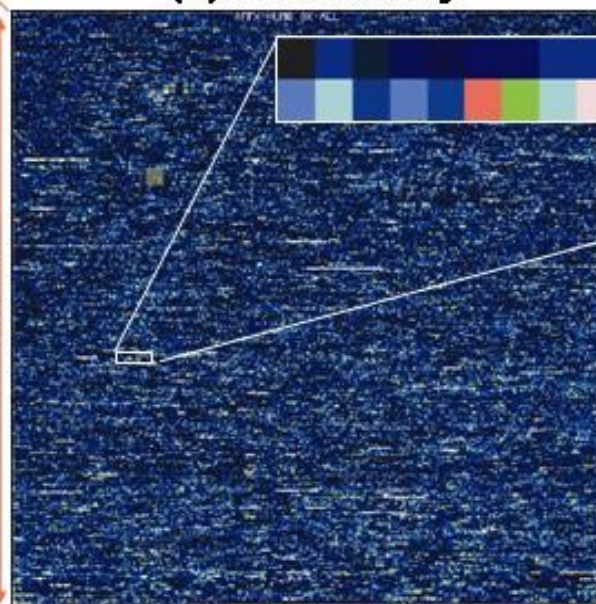
Anatomie GeneChipu® I.

Human Genome U133A GeneChip® Array



1.28cm

(1) Probe Array



(2) Probe Set

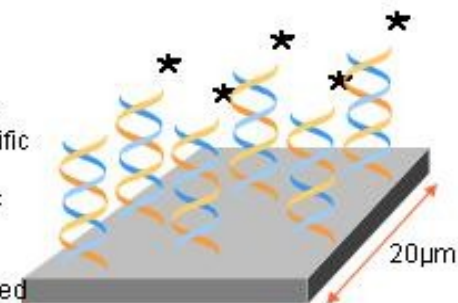
Each Probe Set contains
11 Probe Pairs (PM:MM)
of different probes

(3) Probe Pair

Each Perfect Match
(PM) and Mismatch
(MM) Probe Cells are
associated by pairs

(4) Probe Cell

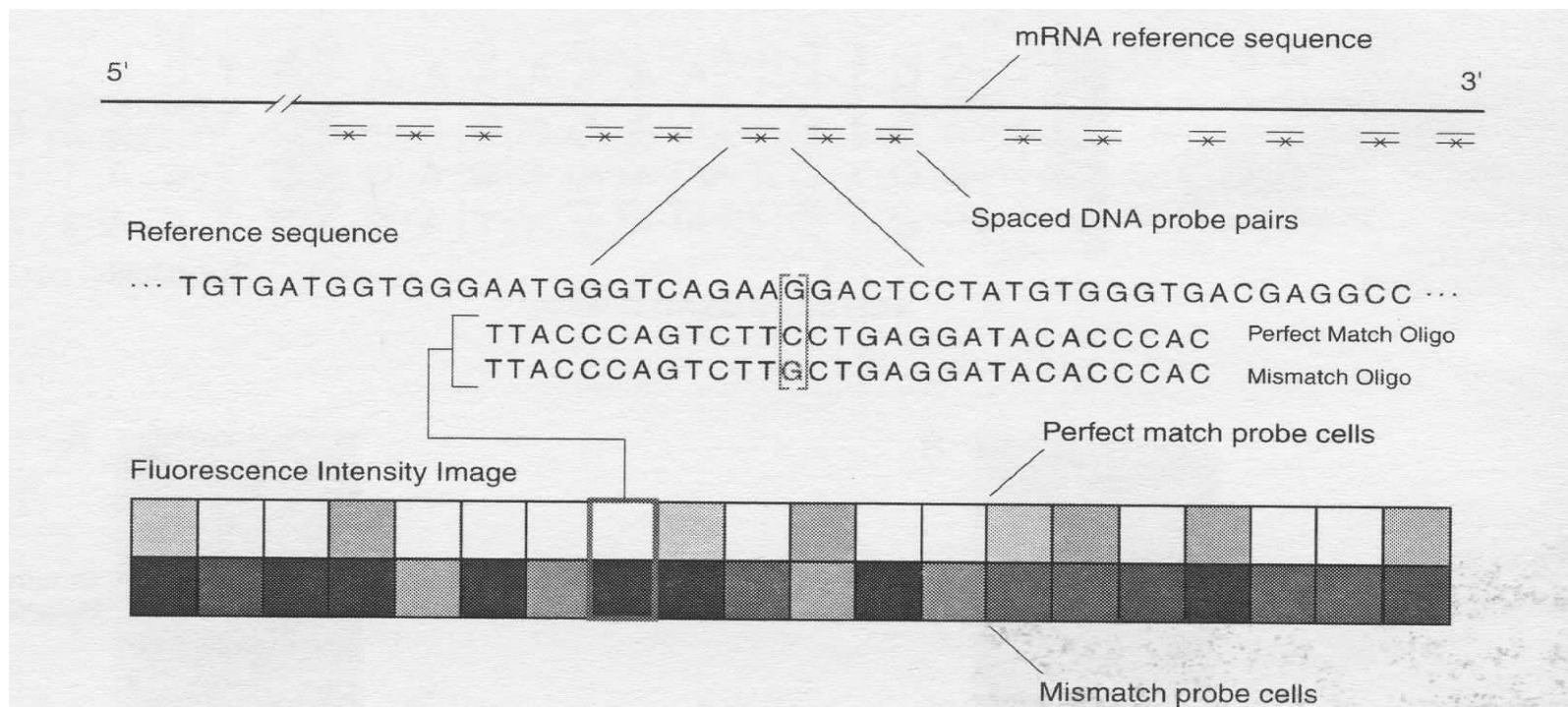
Each Probe Cell contains
 $\sim 40 \times 10^7$ copies of a specific
probe
complementary to genetic
information of interest
probe: single stranded,
sense, fluorescently labeled
oligonucleotide (25 mers)

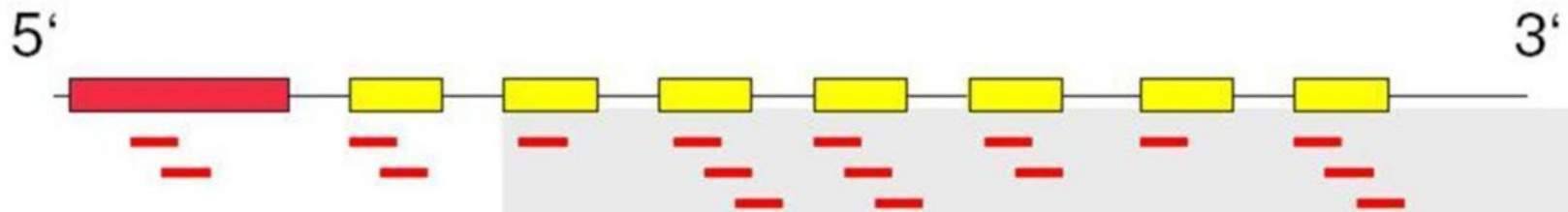


The Human Genome U133 A
GeneChip® array represents
more than 22,000 full-length
genes and EST clusters.

Anatomie GeneChipu® II.

- Sonden = oligonukleotidy, jednořetězcové, délky 25 bp (AGCATGACTAG.....)
- Každý gen reprezentovaný sadou 11-20 párů sond (**probeset**)
- Každý pár sond se skládá z Perfect Match (PM) a Mismatch (MM) sondy
 - PM je perfektní komplementární sekvence genu
 - MM – jako PM, kromě prostřední (13^{té}) báze
 - MM je interní kontrola, měřící nespecifické vazby





several *probe pairs*
(perfect match PM
and mismatch MM)
per *probeset*

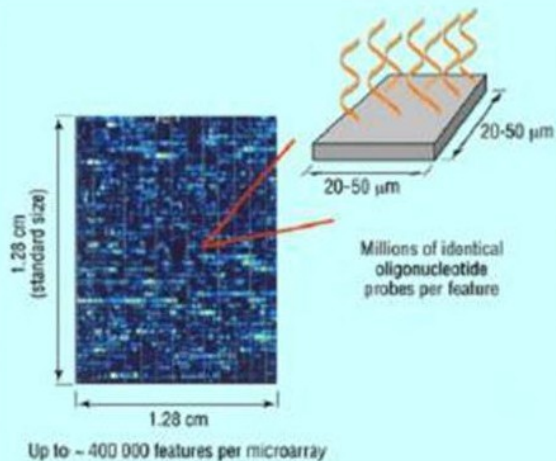
PM: ATGAGCTGTACCAATGCCAACCTGG
MM: ATGAGCTGTACCTATGCCAACCTGG



64 pixels; Signal intensity is upper
quartile of the 36 inner pixels

16-20 probe pairs: HG-U95a
11 probe pairs: HG-U133

Stored in CEL file



Skenování a analýza obrazu Affymetrix

- U jednokanálových oligonukleotidových mikročipů je použita pouze jedna vlnová délka a pomocí UV skeneru je vytvořený jen jeden obraz
- U Affymetrix mikročipů je tento obraz ve formátu *DAT*, a je zpracovaný v software firmy Affymetrix

Affymetrix vs cDNA

- Vzhledem k odlišnému kontextu sond, odlišné úpravy dat než u cDNA
- 11-20 sond na gen - nutná sumarizace, je potřebná jediná hodnota reprezentující gen!
- Rozlišujeme dvě úrovně základních datových matic – **úroveň sondy** (anglicky *probe level*) a **úroveň sady sond** (anglicky *probeset level*)

Kontrola kvality a normalizace

- Jen jeden kanál => většina kontroly kvality a normalizace se vykonává vzhledem k ostatním čipům v experimentu
- Některé nástroje kontroly kvality využívají statistiky, které jsou výsledkem modelování **normalizovaných** intenzit sond
- Kontrolu kvality a normalizaci proto nebudeme dělit na uvnitř čipu a mezi čipy, jako u dvoukanálových cDNA experimentů, ale na **kontrolu sond a kontrolu a normalizaci celých mikročipů.**

AffyBatch

- třída pro uskladnění a analýzu Affymetrix GeneChip dat v Bioconductoru
- Tvoří se s pomocí `read.affybatch()` nebo `ReadAffy()`
- Sloty: `cdfName`, `nrow`, `ncol`, `assayData`, `phenoData`, `annotation`, `protocolData`, `featureData`, `experimentData`

Příkladová data pro ilustraci

- Zde si načteme další datový soubor, na kterém budeme demonstrovat kontrolu kvality. Jedná se o data akutní lymfoblastické leukemie (Ross a kol., 2004). Soubor je součástí balíku ALLMLL a již je ve formátu AffyBatch.

```
install.packages(ALLMLL)
```

```
library(ALLMLL)
```

```
data(MLL.B)
```

- Pro ilustraci z dat vybereme pouze osm mikročipů a jejich názvy změníme na čísla.

```
Data = MLL.B[, c(1:7, 14)]
```

```
sampleNames(Data) = c(1:7, 14)
```

Kontrola kvality na úrovni sady sond I

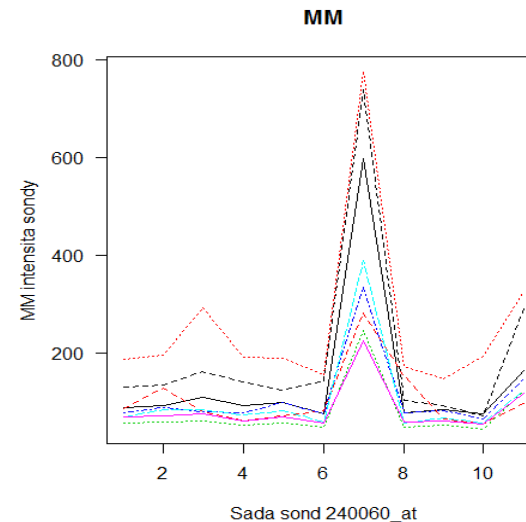
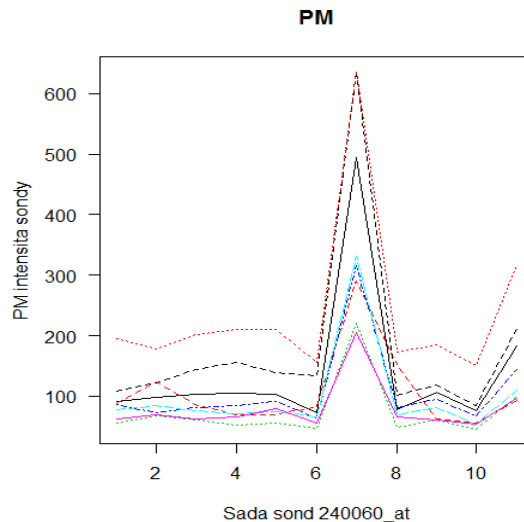
- Najčastejší v prípade, pokud potrebujeme vedieť, zda je určitá sada sond funkčná ve smyslu správnej reprezentácie cílovej sekvencie.

```
pm(Data, "240060_at")
```

```
par(mfrow=c(1,2))
```

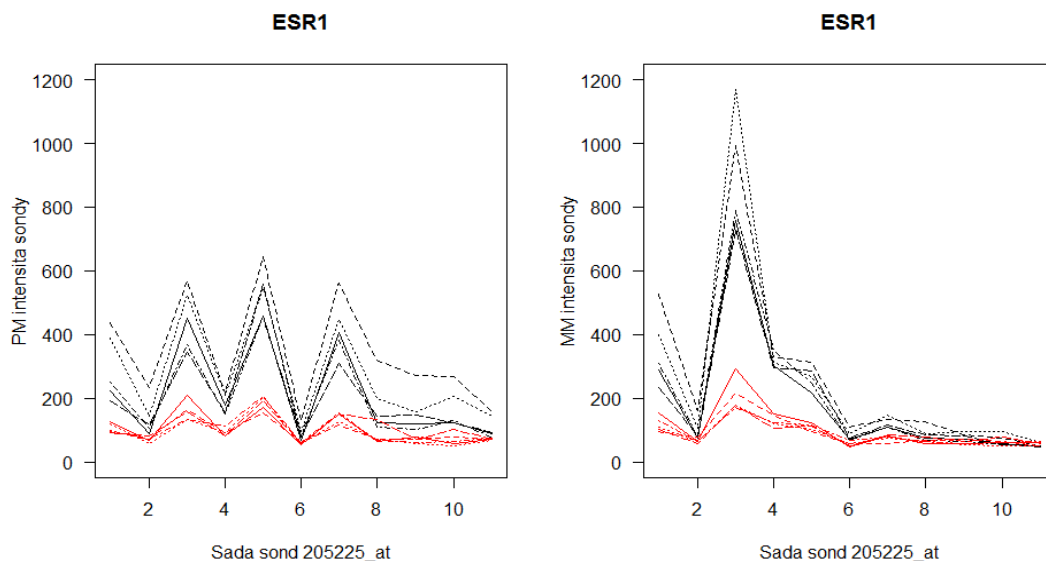
```
matplot(pm(Data, "240060_at"), type="l", ylab="PM intensita sondy", xlab="Sada sond 240060_at", las=1, main="PM")
```

```
matplot(mm(Data, "240060_at"), type="l", ylab="MM intensita sondy", xlab="Sada sond 240060_at", las=1, main="MM")
```



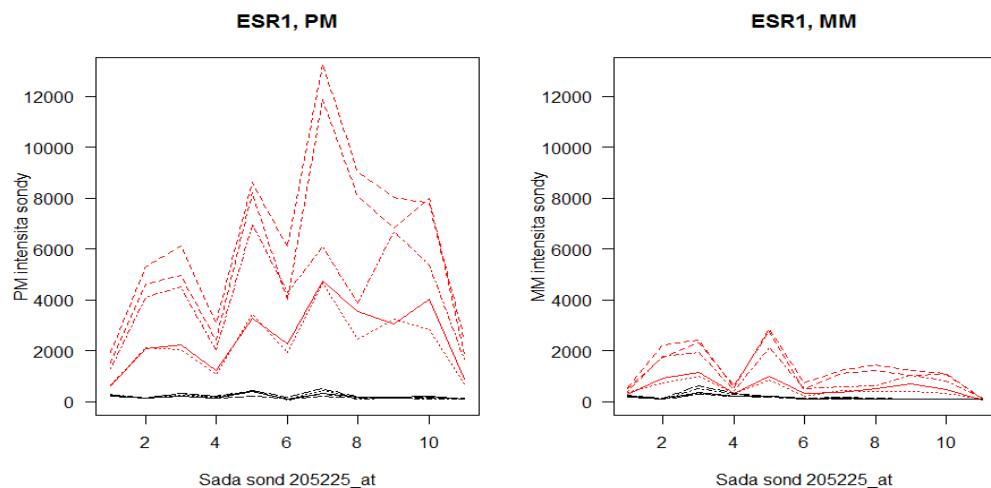
Kontrola kvality na úrovni sady sond II

- Efekt dávky, gen ESR1, data karcinom kolorekta



Dávka 1
Dávka 2

- Porovnání ESR1 MM a PM intenzit u ER+ a ER- karcinomu prsu



ER+
ER-

Kontrola kvality na úrovni mikročipu

Rozlišujeme 3 hlavní způsoby kontroly kvality na úrovni mikročipu:

- Kontrola kvality na základě **parametrů Affymetrix**
- Kontrola kvality s pomocí **základních diagnostických grafů**
- Kontrola kvality na základě **modelu úrovně sondy (PLM – probe level model)**

Efekt barviva není problémem, protože máme pouze jeden kanál.

Kontrola kvality na úrovni mikročipu

Rozlišujeme 3 hlavní způsoby kontroly kvality na úrovni mikročipu:

- Kontrola kvality na základě **parametrů Affymetrix**
- Kontrola kvality s pomocí **základních diagnostických grafů**
- Kontrola kvality na základě **modelu úrovně sondy (PLM – probe level model)**

Efekt barviva není problémem, protože máme pouze jeden kanál.

Kontrola kvality na základě parametrů Affymetrix

Affymetrix vydal sadu doporučení k analýze dat GeneChip mikročipu
”GeneChip® Expression Analysis Data Analysis Fundamentals”

http://media.affymetrix.com/support/downloads/manuals/data_analysis_fundamentals_manual.pdf

Kontrola kvality na základě parametrů Affymetrix I

Balík `simpleaffy` implementuje základní funkce, které počítají sumarizace parametrů kvality Affymetrix GeneChip mikročipu

```
library(simpleaffy)
```

```
Data.qc = qc(Data) #funkce qc()
```

- Podle návodu Affymetrixu by **průměrné hodnoty pozadí měly být porovnatelné** (a mezi 20 a 100)

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

```
  1      2      3      4      5      6      7     14  
67.34494 68.18425 42.12819 61.31731 53.64844 49.39112 75.14030 128.41264
```

- Škálové faktory by se neměly lišit více než trojnásobně mezi čipy:

```
> sfs(Data.qc)
```

```
4.905489 9.765986 10.489529 7.053323 7.561613 13.531238 3.394921 2.475224
```

Kontrola kvality na základě parametrů Affymetrix II

- **Procento nalezených (present) sond** by mělo být **porovnatelné**, přičemž **extrémně nízké hodnoty** jsou znakem **nízké kvality**. V našem případě je na tom nejhůř čip 6.

> `percent.present (Data.qc)`

```
1.present 2.present 3.present 4.present 5.present 6.present 7.present 14.present
26.53124 21.65158 25.58181 23.53279 23.35615 17.96423 25.98808 25.25061
```

- Nakonec, **3'/5' poměry interních kontrolních genů (beta actin a GADPH)** by neměly překročit hranici tří, v našem příkladu tedy nenalzáme problém s degradací RNA.

> `ratios (Data.qc)`

Kontrola kvality na úrovni mikročipu

Rozlišujeme 3 hlavní způsoby kontroly kvality na úrovni mikročipu:

- Kontrola kvality na základě parametrů Affymetrix
- Kontrola kvality s pomocí **základních diagnostických grafů**
- Kontrola kvality na základě modelu úrovně sondy (PLM – probe level model)

Efekt barviva není problémem, protože máme pouze jeden kanál.

Kontrola kvality na základě základních diagnostických grafů I

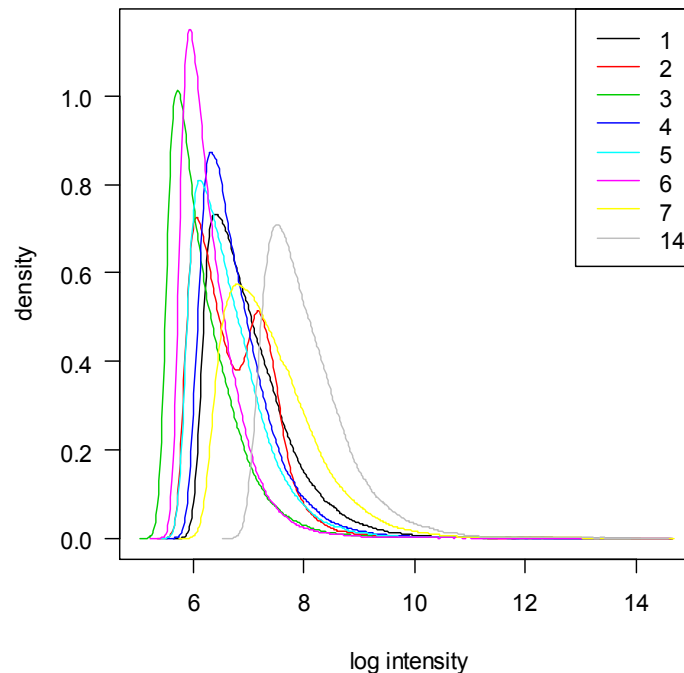
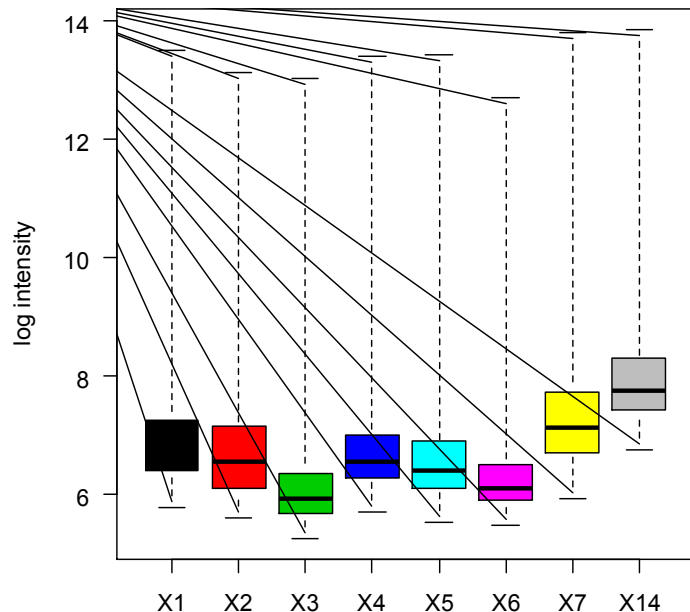
- Krabicové grafy a hustoty rozložení **logaritmovaných** hodnot intenzit sond u všech mikročipů

```
par(mfrow=c(1,2))
```

```
boxplot(Data, las=1, ylab="log intensity")
```

```
hist(Data, las=1, col=c(1:8), lty=1)
```

```
legend("topright", col=c(1:8), lty=1, legend=c(1:7,14))
```



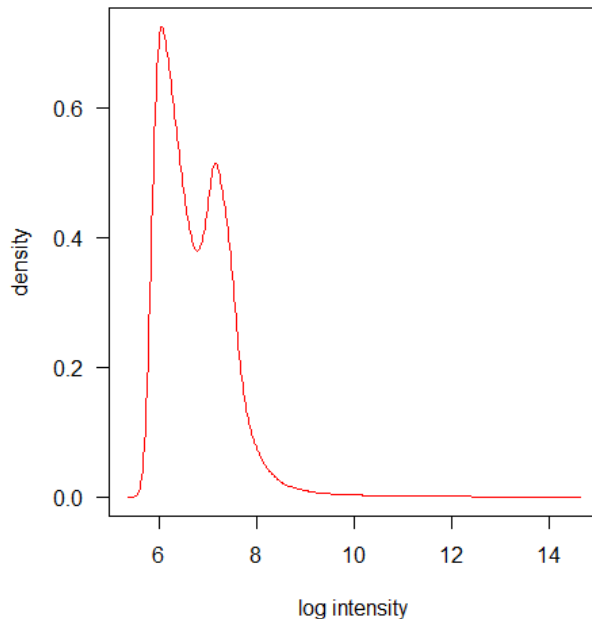
Kontrola kvality na základě základních diagnostických grafů II

- Podobně jako u cDNA mikročipů, i u oligonukleotidových čipů může dojít k prostorovému efektu nerovnoměrné hybridizace, která se pak také odhaluje pomocí heatmapy virtuálně zrekonstruovaného mikročipu

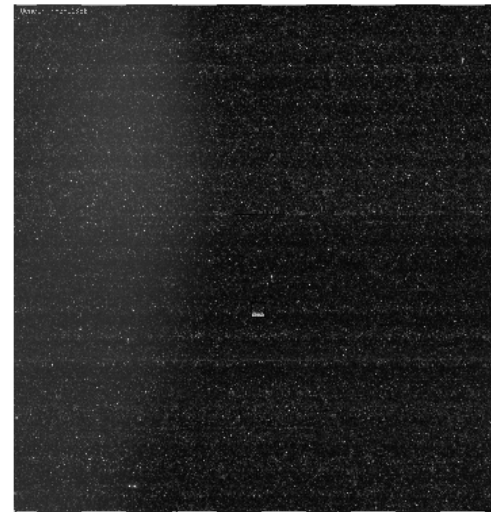
```
par(mfrow=c(1,2))
```

```
hist(Data[,2], las=1, col=2, lty=1)
```

```
image(Data[,2])
```



2



Kontrola kvality na základě základních diagnostických grafů III

- Jako další lze podobně jako u cDNA čipů vykreslit **MA graf**
- *M* a *A* hodnoty se buď počítají mezi dvěma mikročipy, nebo úlohu referenčního kanálu zastoupí referenční pseudo-mikročip (medián)

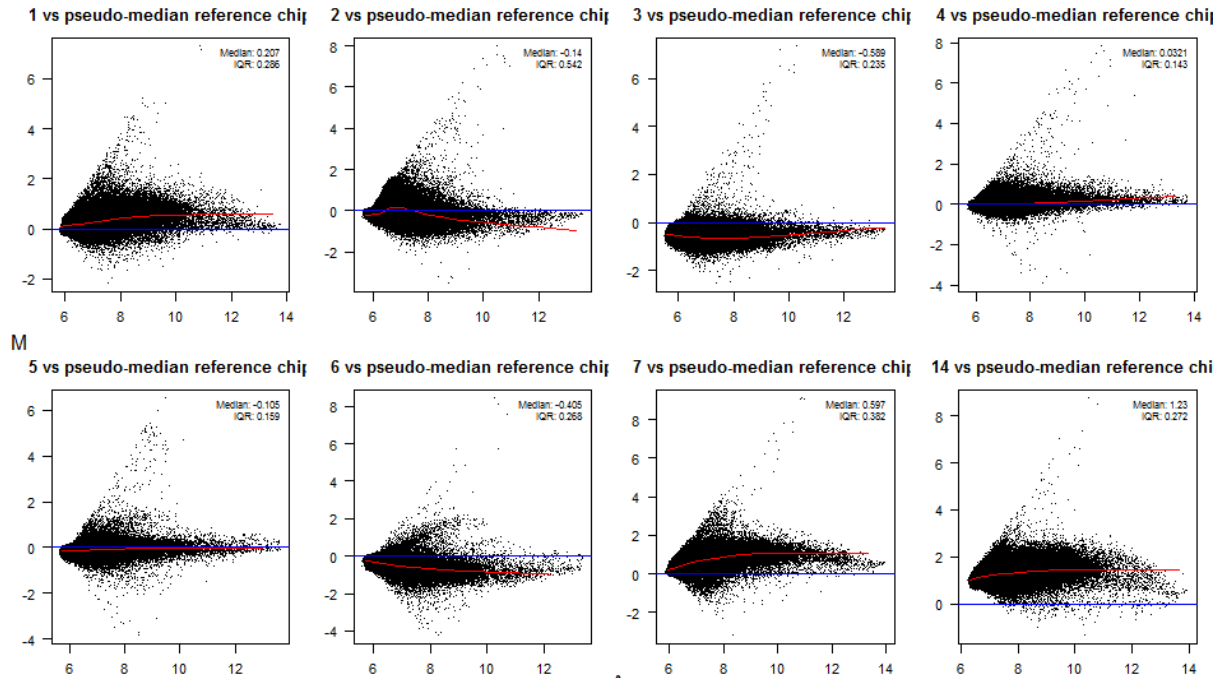
```
windows(12, 7)
```

```
par(mfrow=c(2, 4), mar=c(2, 2, 3, 1))
```

```
MAplot(Data, cex=0.75, las=1)
```

```
mtext("M", 2, outer=T, line=-1.5, las=1)
```

```
mtext("A", 1, line=2, at=-6)
```



Kontrola kvality na základě

modelu úrovně sondy (PLM – probe level model) I.

Tento typ kontroly kvality staví na lineárním modelu Y_{gik} - intenzit normalizovaných na pozadí pomocí RMA, který se nazývá PLM model a je definován následovně:

$$\log(Y_{gik}) = \theta_{gi} + \vartheta_{gk} + \epsilon_{gik},$$

θ_{gi} - logaritmovaná hladina exprese transkriptu (genu) g na mikročipu i

ϑ_{gk} - efekt k -té sondy reprezentující transkript g a ϵ_{gik} je chyba měření

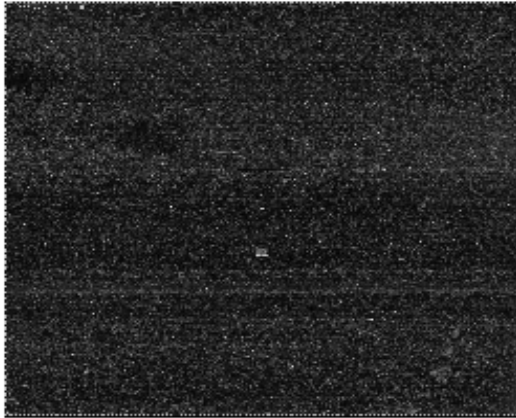
θ_{gi} je tedy již **sumarizovaná hodnota signálů všech sond ze sady** reprezentující gen g a odhaduje se buď pomocí mediánového vyhlazování, nebo pomocí robustní lineární regrese

```
> library(affyPLM)
```

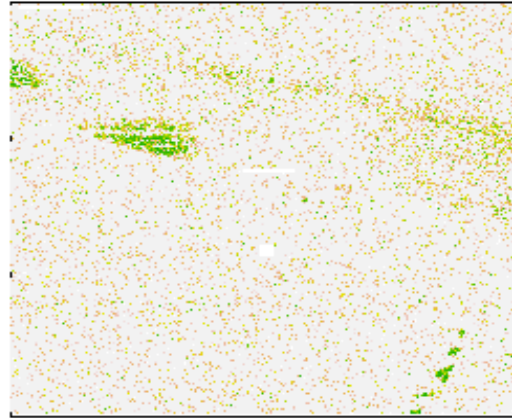
```
PLMres <- fitPLM(Data)
```

Kontrola kvality na základě modelu úrovně sondy (PLM – probe level model) II.

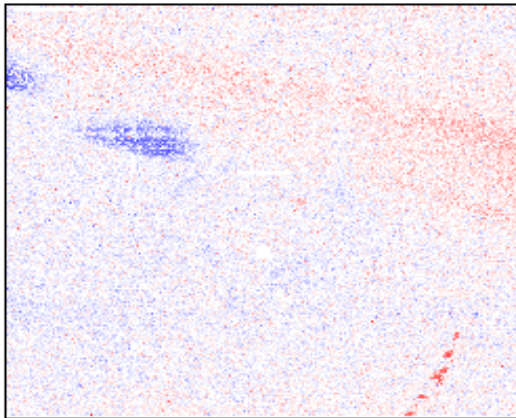
intenzita signálu



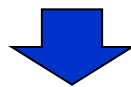
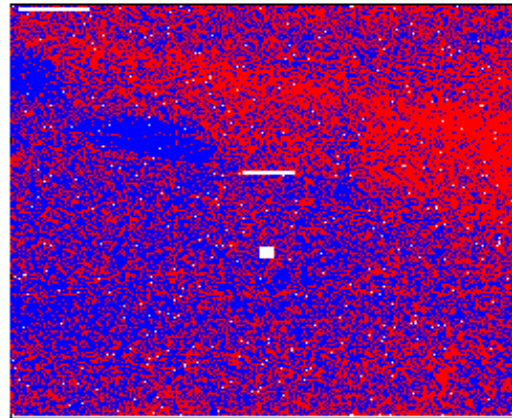
váhy



rezidua



znaménka rezidui



Jak kvantifikovat kvalitu?

Kontrola kvality na základě

modelu úrovně sondy (PLM – probe level model) III.

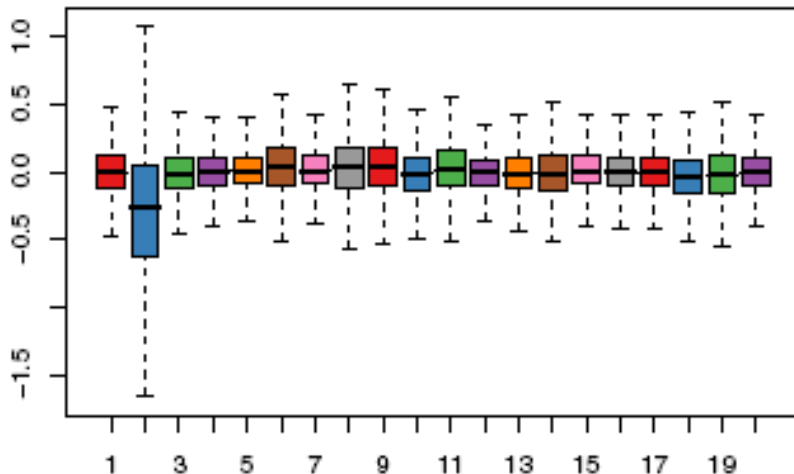
- Relative Log Expression (RLE) $RLE_{gi} = \hat{\theta}_{gi} - m_g$
- Normalized Unscaled Standard Error (NUSE)

$$\text{NUSE}(\hat{\theta}_{gi}) = \frac{\text{SE}(\hat{\theta}_{gi})}{\text{med}_i(\text{SE}(\hat{\theta}_{gi}))}$$

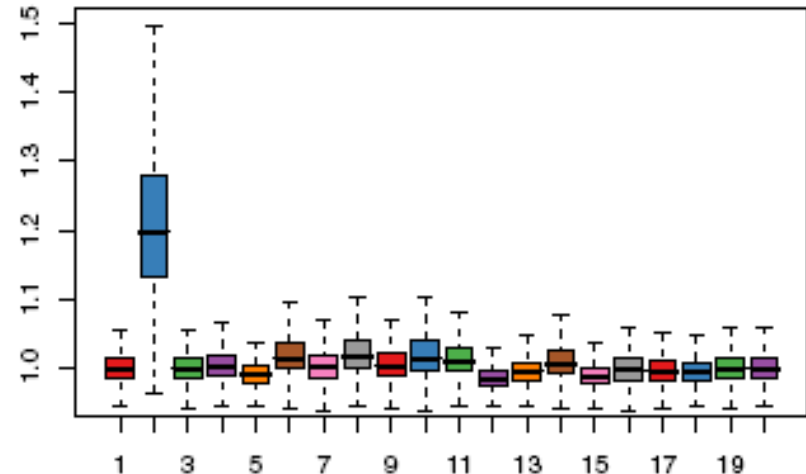
kde θ_{gi} představuje intenzitu genu g na sklíčku i a m_g medián genu i počítaný přes všechny sklíčka

- Počítané pro každý gen, mohou se využít jako kontrola kvality sond a sklíček

RLE



NUSE



Kontrola kvality na základě

modelu úrovně sondy (PLM – probe level model) IV.

- Pokud vzhledem k druhu experimentu a mikročipu můžeme očekávat, že platí předpoklad o nezměněné expresi většiny transkriptů, můžeme odstranit čip jako nekvalitní, pokud má výrazně posunuté *RLE* hodnoty mimo 0, a *NUSE* hodnoty nad 1 (>1.02)

```
> nuse.stat = nuse(PLMres, type="stats")
```

```
> W = nuse.stat["median", ] < 1.02
```

```
> W
```

```
      1      2      3      4      5      6      7     14
TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

```
> Data.clean = Data[,W]
```

Funkce *Mbox* vykreslí krabicové grafy *RLE* hodnoty pro všechny čipy a funkce *NUSE* vykreslí krabicové grafy hodnot *NUSE* :

```
> Mbox(PLMres, main="RLE", las=1)
```

```
> NUSE(PLMres, ylim=c(0.9,2), las=1, main="NUSE")
```


Normalizace a sumarizace

- Mnoho metod pro úpravy dat oligonukleotidových mikročipů představuje algoritmy, které provedou komplexní normalizaci a sumarizaci dat.
- V případě, že tyto metody poprvé představily některou z metod, na tuto metodu se pak odkazuje jménem algoritmu.
- 2 nejznámější algoritmy
 - MAS 5.0 (Microarray Suite 5.0)
 - <http://www.affymetrix.com/products/software/specific/mas.affx>
 - RMA (log scale Robust Multi-array Analysis)
 - Methods for Affymetrix Oligonucleotide Arrays R package
 - <http://www.bioconductor.org>

MAS 5.0 algoritmus

- Používá PM i MM sondy

1. Odečtení intensity pozadí od každé sondy (*PM* i *MM*)

- Metoda odhadu signálu pozadí: Rozdělení čipu na K čtvercových oblastí ($K=16$), označme je Z . 2% sond s nejnižší intenzitou je pak použito pro odhad signálu pozadí u každé oblasti (b_{Z_k}). Odhad pozadí pro sondu na pozici (x, y) pak je vypočten váženým průměrem odhadů signálů všech zón

> `Data.bg.mas5 = bg.correct(Data, method="mas")`

2. Odečtení signálu nespecifické hybridizace sondy i v sadě j

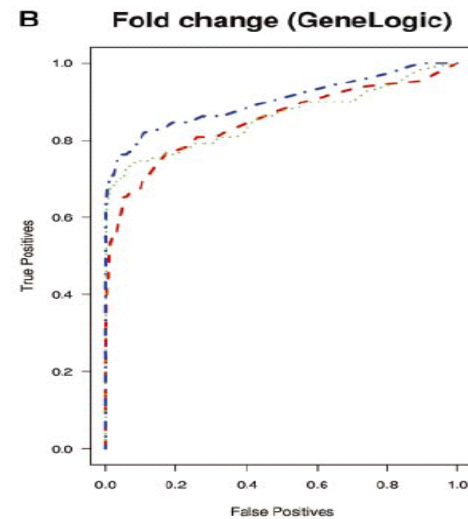
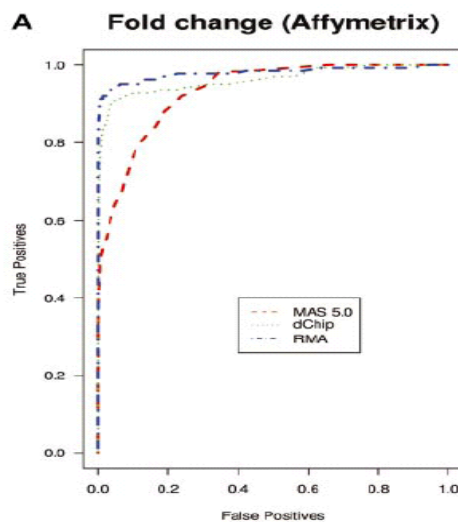
$$V_{i,j} = PM_{i,j} - IM_{i,j}$$

- IM je "ideal mismatch". Je to vlastně MM , ale v případě, že $MM > PM$, MM se odhadne na základě ostatních sond ze sady.

> `threestep(Dilution, background.method = "MASIM")`

RMA algoritmus

- Robust Multichip Average:
 1. Odpočet hodnoty pozadia (odhadnutá zo všetkých MM)
 2. Kvantilová normalizace
 3. Sumarizace
- Používá už všetky microarray sklíčka, počítá jen s PM hodnotami, všetky MM používa na odhad pozadí



> Data.bg.rma = bg.correct(Data, method="rma")

Normalizace mezi mikročipy

- Podobně jako u cDNA mikročipů hlavně:
 - **Centrování mediánem**
 - **Loess**
 - **Kvantilová normalizace**
- Funkce `normalize` implementuje několik normalizačních metod.
Centrování průměrem:

```
> Data.norm.scale = normalize(Data, method="constant")
```

Kvantilová normalizace:

```
> Data.norm.quant = normalize(Data, method="quantiles")
```

Cyklická loess:

```
> Data.norm.loess = normalize(Data, method="loess")
```

Také funkce `threestep` balíku `affyPLM` implementuje několik druhů normalizace. Jak již bylo řečeno výše, tato funkce vrací již sumarizované hodnoty .

Příklad 2

- Načteme knihovnu `affy` pro základní práci s Affymetrix GeneChip daty:

```
library(affy)
```

- Vytvoření datové struktury `AffyBatch` budeme demonstrovat na příkladu mikročipů z experimentu porovnávajícího ER (estrogen receptor) pozitivní a ER negativní karcinomy prsu.
- Pomocí funkce `ReadAffy` načteme základní datové matice (CEL soubory) našeho příkladu do datové struktury `AffyBatch`.

```
breast = ReadAffy(celfile.path="Raw/")
```

Názvy čipů upravíme, odstraníme koncovku ".CEL":

```
ns = length(sampleNames(breast))
```

```
nm = unlist(strsplit(sampleNames(breast), split=".",  
  fixed=TRUE))[seq(1, 2*ns, 2)]
```

```
sampleNames(breast) = nm
```

Konečná podoba dat

mRNA vzorky

	vzorka1	vzorka2	vzorka3	vzorka4	vzorka5	...
1	0.46	0.30	0.80	1.51	0.90	...
2	-0.10	0.49	0.24	0.06	0.46	...
3	0.15	0.74	0.04	0.10	0.20	...
4	-0.45	-1.03	-0.79	-0.56	-0.32	...
5	-0.06	1.06	1.35	1.09	-1.09	...

Gén

M hodnota genu i vzorky j

M = $\left\{ \begin{array}{l} \text{Log}_2(\text{Cy5} / \text{Cy3}) - \text{cDNA arrays} \\ \text{Funkce}(\text{PM}, \text{MM}) \text{ z MAS, dchip alebo RMA} \end{array} \right.$