HTSFilter: Data-based filtering for replicated high-throughput sequencing experiments

Andrea Rau, Mélina Gallopin, Gilles Celeux, and Florence Jaffrézic

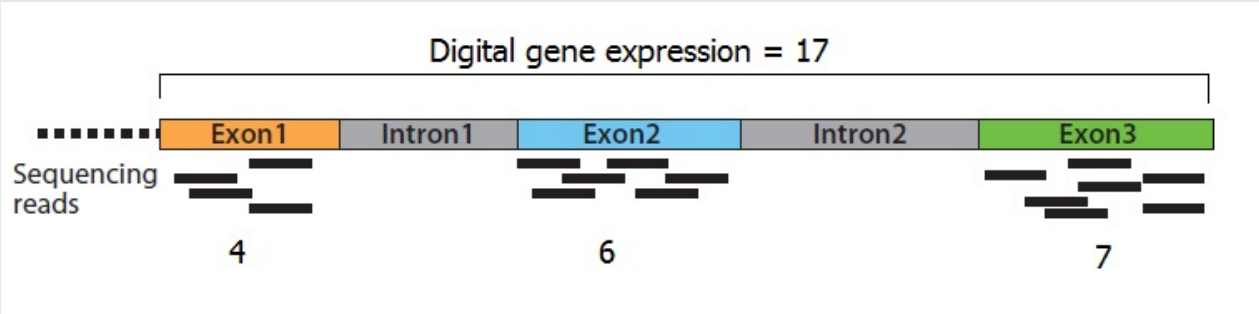
Deuxièmes rencontres R @ Lyon
June 28, 2013

RNA sequencing (RNA-seq)

- RNA-seq = Application of high-throughput sequencing (HTS) technology to the study of gene expression

Analysis of RNA-seq data

1. Short reads pre-processed and mapped onto a genome reference sequence or assembled
2. Expression level estimated for each biological entity (e.g., a gene)
   $\Rightarrow$ Here we focus on count-based measures of gene expression (number of sequenced reads mapped to a gene)



3. Data normalization and statistical analysis

Differential gene expression analysis

Differential expression (DE)

Observed change in expression between two experimental conditions is statistically significant, i.e., greater than expected just due to natural random variation.

$\Rightarrow$ Statistical tools required to make such a decision (count data, highly heterogeneous, over-dispersion, ...)

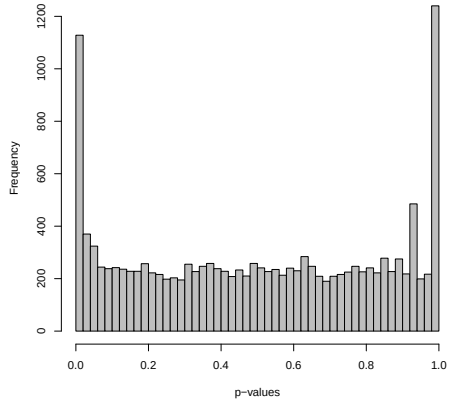
Several approaches have been proposed using over-dispersed Poisson or negative binomial models

- Bioconductor packages `DESeq`, `DESeq2`, `edgeR`, `limma` (with `voom` function), `DSS`, ...

Filtering in differential expression analysis

Differential analyses performed gene-by-gene, requiring a correction for multiple testing (e.g., FDR control):

- Stringent correction due to large number of hypothesis tests
- Usually assume p-values are uniformly distributed under H_0



Filtering for RNA-seq data

- Identify and remove genes that generate an uninformative signal
- Only test hypotheses for genes passing filter $\Rightarrow$ tempered correction for multiple testing
- Up to now, little discussion about appropriate filter & threshold

Defining a data-based filter for HTS data

Let $\mathbf{y}_j$ be the full vector of normalized read counts in a given sample $j \in \{1, \ldots, J\}$, where $\mathcal{C}(j)$ is the experimental condition of sample j.

Idea:

Find the threshold s that maximizes the filtering similarity among replicates in the same condition ($\mathcal{C}(j) = \mathcal{C}(j')$) using the Jaccard index:

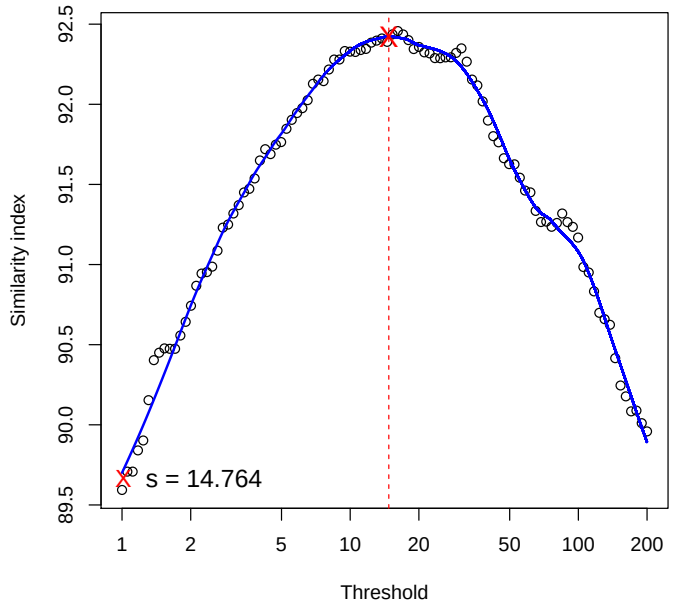
$$J_s(\mathbf{y}_j, \mathbf{y}_{j'}) = \frac{a}{a + b + c}$$

		Sample j: Normalized counts $> s$	Sample j: Normalized counts $\leq s$
Sample j'	Normalized counts $> s$	a	b
	Normalized counts $\leq s$	c	d

Data-based filtering threshold for HTS data

- Multiple replicates/conditions typically available $\Rightarrow$ define a global filtering similarity by summing the pairwise Jaccard indices within each condition:

$$J_s^\star(\mathbf{y}) = \sum_{\substack{j<j' \\ \mathcal{C}(j)=\mathcal{C}(j')}} J_s(\mathbf{y}_j, \mathbf{y}_{j'})$$



- Data-based filter threshold $s^\star = \text{argmax}_s\, J_s^\star(\mathbf{y})$

Proposed data-based Jaccard filter

Filter genes with normalized read counts $\leq s^\star$ in all samples

HTSFilter package: Primary command

```
## S4 method
HTSFilter(x, conds,
     s.min=1, s.max=200, s.len=100, loess.span=0.3,
     normalization=c("TMM", "DESeq", "none"),
     plot=TRUE, plot.name=NA)
```

Implementation of HTSFilter in the DESeq pipeline

```
> library(DESeq)
> library(HTSFilter)
> data("sultan")
> conds <- pData(sultan)$cell.line
>
> ## DESeq commands
> cds <- newCountDataSet(exprs(sultan), conds)
> cds <- estimateSizeFactors(cds)
> cds <- estimateDispersions(cds)
>
> ## HTSFilter
> cds <- HTSFilter(cds)$filteredData
>
> ## Calculate p-values
> res <- nbinomTest(cds, levels(conds)[1], levels(conds)[2])
```

Implementation of HTSFilter in the edgeR pipeline

```
> library(edgeR)
> library(HTSFilter)
> data("sultan")
> conds <- pData(sultan)$cell.line
>
> ## edgeR commands
> dge <- DGEList(counts=exprs(sultan), group=conds)
> dge <- calcNormFactors(dge)
> dge <- estimateCommonDisp(dge)
> dge <- estimateTagwiseDisp(dge)
> et <- exactTest(dge)
>
> ## HTSFilter
> et <- HTSFilter(et, DGEList=dge)$filteredData
> topTags(et)
```

Comparisons of filters made on real and simulated data

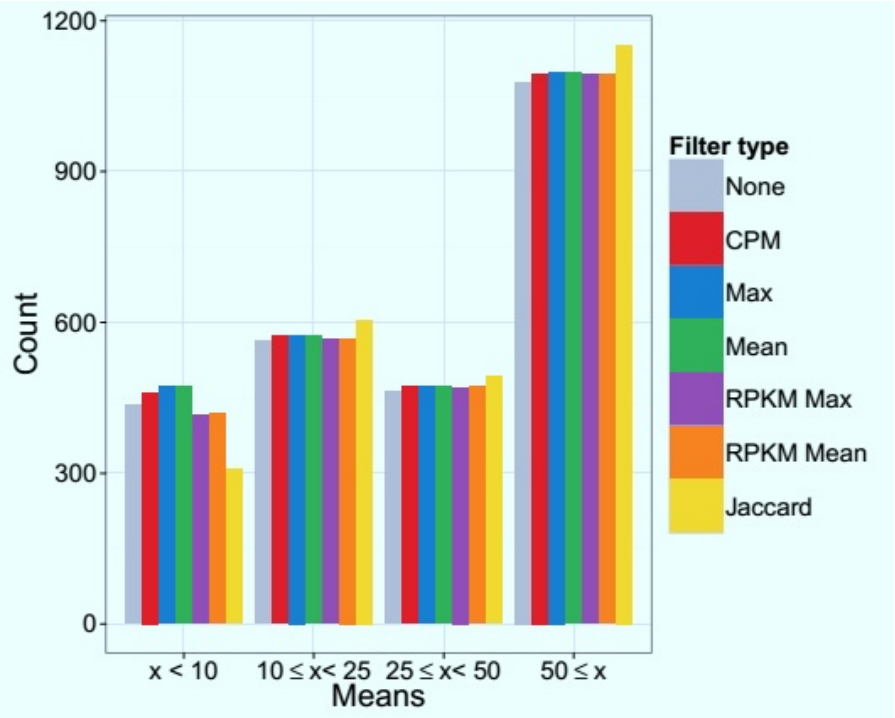
Supplementary Table 1: Characteristics for the Bottomly, Sultan, and Strub data, including the organism studied, the experimental conditions under comparison, sequencing machine used, relevant publication, number of replicates per condition, number of non-zero genes (i.e., genes with a non-zero count in at least one sample), sequencing depth (i.e., total number of uniquely mapped reads), minimum and maximum library sizes, and minimum intra-condition correlation.

	Bottomly	Sultan	Strub
Organism	Mouse	Human	Human
Comparison	C57BL/6J vs. DBA/2J strains	Embryonic kidney vs. B cell line	MiTF melanoma vs. repressed miTF cell lines
Sequencing machine	Illumina GA IIx	1G Illumina Genome Analyzer	Illumina GA IIx
Publication	Bottomly et al. (2011)	Sultan et al. (2008)	Strub et al. (2011)
Reps per condition	$\{10, 11\}$	$\{2, 2\}$	$\{3, 3\}$
Non-zero genes	13,932	9,010	27,485
Sequencing depth	102,987,446	1,793,562	147,294,269
Min library size	2.7×10^6	3.9×10^5	2.0×10^7
Max library size	7.3×10^6	5.1×10^5	2.8×10^7
Min intra-condition correlation	0.82	0.99	0.98

- Variety of real data (human, mouse) with different characteristics
- Simulations using negative binomial models and parameters fixed based on real datasets
- Alternative filters for comparisons: unfiltered, mean- and maximum-based (normalized read counts, RPKM, CPM), with thresholds chosen using 15% quantile

Selected results from filter comparisons

- Jaccard filter leads to more discoveries (increased detection power!) at all but very weak levels of expression
- Note: about half of discoveries with mean expression < 10 in unfiltered data had 0 read counts in one of the conditions



(Results for Sultan data)

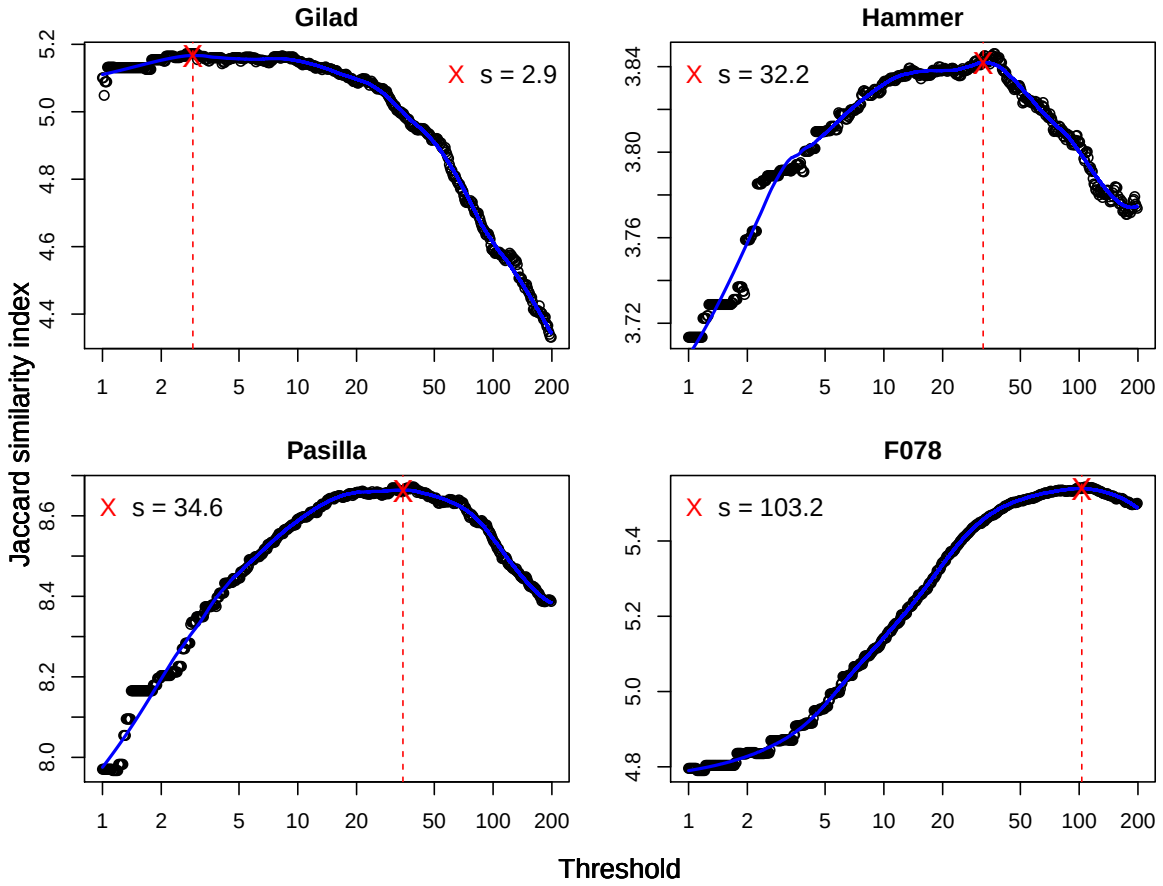
- Simulations: max-based filters more effective than mean-based filters, and the data-based `HTSFilter` threshold is a reasonable choice

Discussion

Filtering is of great practical importance for differential analysis of microarray and RNA-seq data:

- Identify and remove genes with uninformative signal prior to testing
- Until now, no clear recommendations about choice of filtering technique for RNA-seq data
- HTSFilter: a data-driven and non pre-fixed filtering threshold for replicated HTS data that was found to perform well in comparison with several other commonly used ad hoc filters

Discussion: A word on data-driven threshold values...



Filtering threshold is specific to each dataset (tissue, organism, sequencing depth, intra-condition variability ...)

R package HTSFilter:

- Release version available on Bioconductor (http://www.bioconductor.org/packages/2.12/bioc/html/HTSFilter.html) and development version available on R-Forge (http://r-forge.r-project.org/projects/htsfilter)
- Compatible with a variety of data classes and analysis pipelines: `matrix` and `data.frame` objects, the S4 class `CountDataSet` (`DESeq`), and the S3 class `DGEList` (`edgeR`), ...

Rau, Gallopin, Celeux, Jaffrézic (2013). Data-based filtering for replicated high-throughput sequencing experiments. *Bioinformatics* (to appear).

Thank you!