Lecture 10

Differential Expression (DE) Analysis on Microarray Data

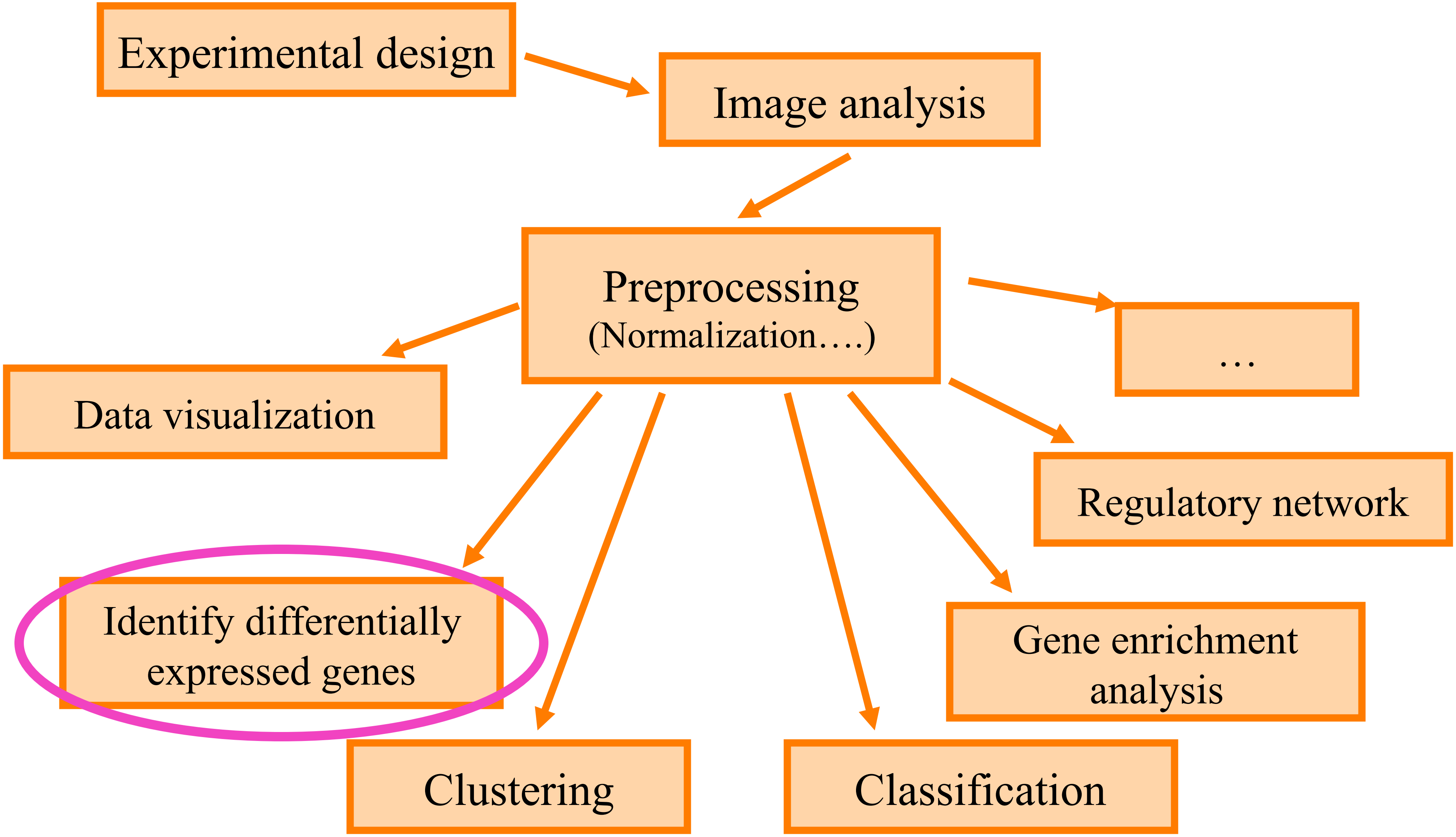
MCB 416A/516A
Statistical Bioinformatics and Genomic Analysis

Prof. Lingling An
Univ of Arizona

Outline

- Measuring differential expression
- Multiple hypothesis testing
- R code for DE on Affymetrix data

Statistical Issues in Microarray Analysis



Select DE genes

	Tumor	Tumor	Tumor	Tumor	Normal	Normal	Normal	Normal	
31308_at	21.0199	29.1547	17.9257	20.3766	19.8673	18.4821	17.9005	20.863	No
31309_r_at	46.0512	48.7559	43.1192	46.5921	25.2423	33.0099	30.2182	27.3594	Yes
31310_at	20.3716	27.6846	20.7468	18.5927	16.1071	15.4484	16.9989	16.1746	Yes
31311_at	75.6513	94.4134	80.6328	84.4216	71.8248	69.553	78.4236	71.5484	??
31312_at	97.5175	154.163	90.5806	118.928	115.495	130.89	100.678	89.8753	No
31313_at	50.9551	58.7498	54.0995	46.8968	61.6732	62.3931	64.7219	57.7332	??
31314_at	52.2138	62.3064	59.9553	54.8983	77.118	61.0678	84.1336	82.37	Yes
31315_at	315.543	252.801	204.426	265.601	224.804	225.89	139.36	177.225	
31316_at	12.2335	12.163	8.8393	10.0476	13.2467	13.3113	12.7941	10.0831	
31317_r_at	361.66	423.547	331.67	404.61	260.041	295.872	235.307	209.306	
31318_at	19.4059	26.4248	17.1136	16.5311	12.6095	15.2638	13.262	15.527	
31319_at	159.305	120.841	120.867	117.889	124.751	122.684	116.257	123.107	
31320_at	309.203	273.927	226.194	342.061	267.247	299.116	269.536	240.244	

Which genes are differentially expressed between tumor and normal?

Measuring differential expression

- One common goal is to rank all the genes in a study in order of evidence for differential expression
- Ways to score genes:
  - – Fold change
  - – T-statistic, p-value
  - – Another statistic (nonparametric, etc.)
  - – A combination of several scores

  …

Fold change

$$\text{Fold change} = \frac{\text{expression value in sample 1}}{\text{expression value in sample 2}}$$

- Advantage: Fold change makes sense to biologists

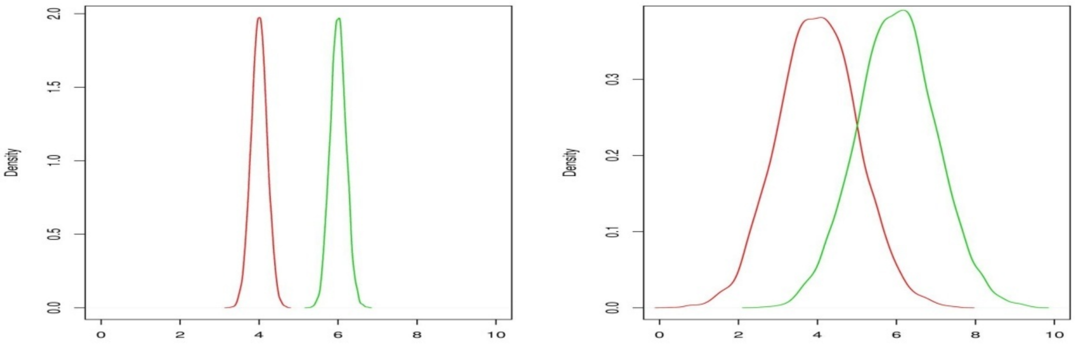
- What cutoff should be used?
- Should it be the same for all genes?
- Disadvantages:
  – Only mean values – not variability – are considered

Hypothesis testing

- We may want to test …
  - – Is the expression of my gene different in a set in one condition compared to another condition?
  - – How big is the difference?
  - – Is the mean of one set of values different from the mean of another set of values?
  - – If we say "yes", how much confidence do we have that the means are truly different?
- Assumptions:
  - – Data are normally distributed or with large sample size
  - – Samples are randomly chosen

Two-sample t-test

- Advantages: Considers mean values and variability



- Disadvantages:
  – Genes with small variances are more likely to make the cutoff
  – Works best with larger data sets than one usually has
  – Normal distribution assumption may not be held

Select DE genes

Statistical significance:

the observed differential expression is unlikely to be due to chance.

Scientific (biological) significance:

the observed level of differential expression is of sufficient magnitude to be of biological relevance.

Flavors of the tests

- Are we only considering up-regulated or down-regulated genes, or both?
  – If both, perform a 2-tailed test
- Can we assume that the variance of the gene is similar in both samples?
  – Yes (e.g., pooled t-test)
  – No (e.g., unequal variance t-test)
- Moderated t-tests: pool data for many genes
  – Significance Analysis of Microarrays (SAM)
  – limma (Bioconductor)

$$t = \frac{\overline{x}_1 - \overline{x}_2}{s + s_0}$$

- If assumption of normal distribution is not hold, using nonparametric way – Wilcoxon test

DE from multiple-group comparison: ANOVA

- Analysis of variance – like a multidimensional t-test
- Measure effect of multiple treatments and their interactions
- A thoughtful ANOVA design can help answer several questions with one analysis
- ANOVA generally identifies genes that are influenced by some factor – but then post-hoc tests must be run to identify the specific nature of the influence
  - Ex: t-tests between all pairs of data

ANOVA (Analysis of Variance): Model in limma package

Let y_{ijkg} be the log-transformed fluorescent intensity measured from Array i, Dye j, Variety k, and Gene g. A typical analysis of variance (ANOVA) model is:

$$y_{ijkg} = \mu + A_i + D_j + V_k + G_g + (AG)_{ig} + (DG)_{jg} + \mathbf{(VG)}_{kg} + \varepsilon_{ijkgl}$$

- μ: overall mean
- G are the overall gene effects
- AG’s are “spot” effects
- DG’s are gene-specific dye effects
- VG’s are the effects of interest. The capture the expression of genes specifically attributable to varieties.
- ε is random error

Multiple tests/comparison

	Tumor	Tumor	Tumor	Tumor	Normal	Normal	Normal	Normal	
31308_at	21.0199	29.1547	17.9257	20.3766	19.8673	18.4821	17.9005	20.863	No
31309_r_at	46.0512	48.7559	43.1192	46.5921	25.2423	33.0099	30.2182	27.3594	Yes
31310_at	20.3716	27.6846	20.7468	18.5927	16.1071	15.4484	16.9989	16.1746	Yes
31311_at	75.6513	94.4134	80.6328	84.4216	71.8248	69.553	78.4236	71.5484	??
31312_at	97.5175	154.163	90.5806	118.928	115.495	130.89	100.678	89.8753	No
31313_at	50.9551	58.7498	54.0995	46.8968	61.6732	62.3931	64.7219	57.7332	??
31314_at	52.2138	62.3064	59.9553	54.8983	77.118	61.0678	84.1336	82.37	Yes
31315_at	315.543	252.801	204.426	265.601	224.804	225.89	139.36	177.225	
31316_at	12.2335	12.163	8.8393	10.0476	13.2467	13.3113	12.7941	10.0831	
31317_r_at	361.66	423.547	331.67	404.61	260.041	295.872	235.307	209.306	
31318_at	19.4059	26.4248	17.1136	16.5311	12.6095	15.2638	13.262	15.527	
31319_at	159.305	120.841	120.867	117.889	124.751	122.684	116.257	123.107	
31320_at	309.203	273.927	226.194	342.061	267.247	299.116	269.536	240.244	

Which genes are differentially expressed between tumor and normal?

So far we discuss hypothesis testing applied for each individual gene.

Multiple hypothesis testing

- We need both sensitivity and specificity:
  - Sensitivity: probability of successfully identifying a real effect
  - Specificity: probability of successfully rejecting a nonexistent effect
  - These are inversely related.
- The problem
  - The number of false positives greatly increases as one performs more and more t-tests
  - How seriously do you want to limit false positives?

Why correct for multiple hypothesis testing?

Number of genes tested (N)	Probability of ≥1 FP $=100(1-0.95^N)$
1	5%
10	40.1%
100	99.4%

FP = false positive

Multiple comparison correction-FWER

Controlling Family-wise Error Rate (FWER)

control Pr(at least one false positives)

e.g., Bonferroni: α/m

- If $\alpha=0.05$, $m=6000$ genes, then the p-value threshold for individual test is selected as 0.05/6000=0.0000083.
- The method is too conservative. Almost no genes will be selected in microarray study.

Other methods to control FWER have better bound. But normally trying to control FWER selects only very few genes.

So we'll use false discovery rate (FDR) method -

Multiple comparison correction -FDR

What a scientist cares is: Among the detected DE gene list, how many are real and how many are false positives? **=> concept of false discovery rate (FDR)**

	# of non-rejected hypotheses	# of rejected hypotheses	
# of true non-DE genes	U	V **(false positives)**	m_0
# of true DE genes	T **(false negatives)**	S	m_1
	m-R	R	m

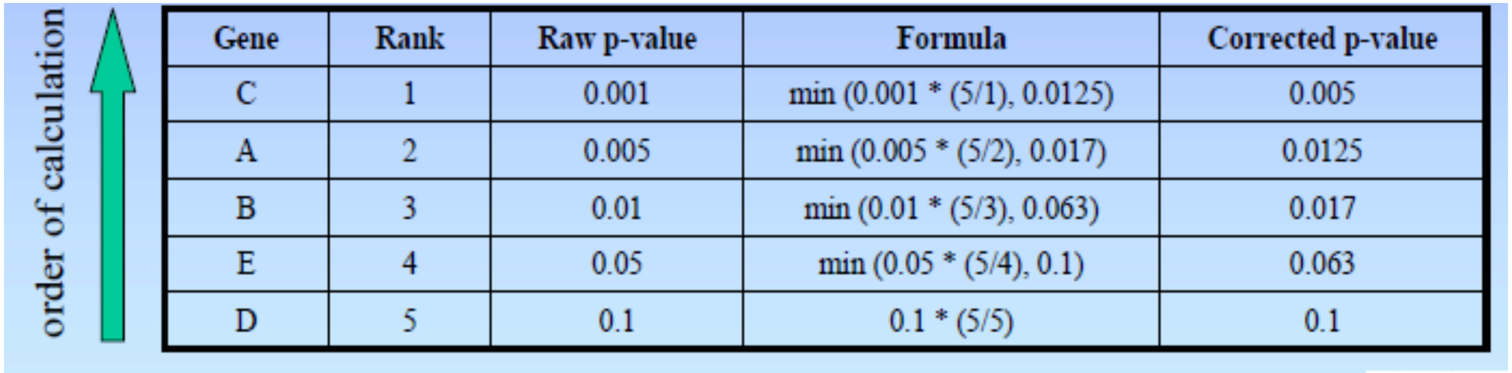
False discovery rate

- False Discovery Rate (FDR)
  - (Benjamini and Hochberg, JRSS-B, 1995) proposed the idea of FDR: the proportion of type I errors among rejected hypotheses.

$$FDR=E(Q)=E(V/R)$$

Performing a FDR correction

- Sort list of p-values in increasing order
- Starting at the bottom row, corrected p-value = the minimum between
  - 1: raw p-value * (m/rank)
  - 2: corrected p-value below
    - m is the number of tests
    - rank is the position in the sorted list
- Example: a microarray assays 5 genes for differential expression


order of calculation

Gene	Rank	Raw p-value	Formula	Corrected p-value
C	1	0.001	min (0.001 * (5/1), 0.0125)	0.005
A	2	0.005	min (0.005 * (5/2), 0.017)	0.0125
B	3	0.01	min (0.01 * (5/3), 0.063)	0.017
E	4	0.05	min (0.05 * (5/4), 0.1)	0.063
D	5	0.1	0.1 * (5/5)	0.1

When to use FWER and when to use FDR?

FWER=Pr(V>0)=Pr(at least one false positive)
FDR=E(Q)=E(V/R)

1. Choose FWER if high confidence in “ALL” selected genes is desired (for example, selecting candidate genes for RT-PCR validation). Loss of power due to strong control of type-I error.

2. Use more flexible FDR procedures if certain proportions of false positives are tolerable (e.g. gene discovery, selecting candidate co-regulated gene sets for GO/pathway analysis).

And more

- If false positives are not tolerated
  - Perform Bonferroni correction
    - If you perform 100 t-tests, multiply each p-value by 100 to get corrected (adjusted) values p = 0.0005 => p = 0.05
- If false positives can be tolerated
  - Use False Discovery Rate (FDR)
    - If you can tolerate 15% false positives, calculate FDR p-values and then select 0.15 as your threshold
- FDR method is less conservative than Bonferroni and usually more appropriate for microarrays.

R code for DE on Affymetrix array data

Load the required packages

```
>library(affy)
>library(limma)
>library(hgu95acdf)
```

Download .CEL datasets from “data_DE” in D2L

Save them into a new folder in your computer …

Linear model for microarray (limma)

- limma is an R package to find differentially expressed genes:
  - for comparisons between two or more groups
  - for multifactorial designs
- it uses linear models
  - fitted to normalized intensities (one-color)
  - or log-ratios (two-color)
- It uses empirical Bayes analysis to improve power in small sample sizes
  - borrowing information across genes
- assumption: normal distribution

My experience

- limma has excellent documentation and many examples
- integration with preprocessing and exploratory data analysis makes it possible to test different options for background subtraction and normalization – *for two color array data*
- makes it possible for a non-statistician to fit linear models and find differentially expressed genes

Linear models

- Limma approach requires two matrices to be specified:
- design matrix
  - indicates which RNA samples have been applied to each array
  - rows: arrays; columns: coefficients
- contrast matrix
  - specifies which comparisons you would like to make between the RNA samples
  - for very simple experiments, you may not need a contrast matrix

Steps of using limma

- create design matrix
- fit a linear model to estimate all the (fold) changes
- [make contrasts matrix]
- apply Bayesian smoothing to the standard errors (very important)
- output: moderated- t-statistics, f-statstics, logFC, p-value, etc.

Example Data

- tissues from fetal and normal human liver and brain
- Hybridized to Affymetrix HU95A arrays
- each hybridization is performed duplicate
- Aim:

-- identify genes which are differentially expressed between fetal and normal brain tissue, and between fetal and normal liver tissue

-- and classify them

Load and visualize data

- Download the 8 .CEL files into your computer
- Change the working directory to the folder where your data are saved
- Read in data:

```
> raw = ReadAffy()
```

- Check the data file names:

```
>pData(raw)
```

- Data visualization:

```
>probeNames(raw)[1:20]
>geneNames(raw)[1:10]
>image(raw[,1])
>hist(raw)
>boxplot(raw, col=c(2,2,3,3,4,4,5,5))
> par(mfrow=c(2,4))
> MAplot(raw, plot.method="smoothScatter" )
```

Data preprocessing

Three options:

- 1) step by step:

```
>raw.bg=bg.correct(raw, “rma”)
>raw.norm=normalize(raw.bg, “quantiles”)

>hist(raw.norm, main = “ raw data after RMA  BG & quantile normalization ”)
>par(mfrow=c(2,4))
>MAplot(raw.norm, plot.method= “smoothScatter”)

>raw.pm=pmcorrect.pmonly(raw.norm)
>eset <- computeExprSet(raw.norm, “pmonly”,“medianpolish”)
```

- 2) expresso:

```
> eset <- expresso(raw, bgcorrect.method = “rma”, normalize.method =
“quantiles”, pmcorrect.method = “pmonly”,  summary.method =“medianpolish”)
```

- 3) RMA wrapper:

```
> eset=rma(raw) ##(background correcting, normalizing, calculating Expression)
```

Construct a design matrix

- There are many ways to construct a design matrix for this experiment.

```
> f <- factor(c(1,1,2,2,3,3,4,4), labels=c(″brain″,
″f.brain″, ″f.liver″, ″liver″ ))

>design <- model.matrix(~ 0 + f)
```

What my design matrix looks?

```
> design
  fbrain ff.brain ff.liver fliver
1      1        0        0      0
2      1        0        0      0
3      0        1        0      0
4      0        1        0      0
5      0        0        1      0
6      0        0        1      0
7      0        0        0      1
8      0        0        0      1
attr(,"assign")
[1] 1 1 1 1
attr(,"contrasts")
attr(,"contrasts")$f
[1] "contr.treatment"
```

```
> colnames(design) <-c("brain", "f.brain", "f.liver", "liver")
> design
  brain f.brain f.liver liver
1     1       0       0     0
2     1       0       0     0
3     0       1       0     0
4     0       1       0     0
5     0       0       1     0
6     0       0       1     0
7     0       0       0     1
8     0       0       0     1
attr(,"assign")
[1] 1 1 1 1
attr(,"contrasts")
attr(,"contrasts")$f
[1] "contr.treatment"
```

Linear fitting

```
fit <- lmFit(eset, design)
```

Construct contrast:

- Given that we are interested in the comparison of fatal brain and brain, the comparison of fatal liver and liver, we can choose a parametrization which includes these two contrasts.

```
> contrast.matrix <- makeContrasts(f.brain-
  brain, f.liver-liver, levels = design)
```

Given the linear model fit to microarray data, compute the estimated coefficients and standard errors for the previous specified set of contrasts.

```
>fit1 <- contrasts.fit(fit, contrast.matrix)
```

Given a series of related parameter estimates and standard errors, compute moderated t-statistics, moderated F-statistic, and log-odds of differential expression by empirical Bayes shrinkage of the standard errors towards a common value.

```
> fit2 <- eBayes(fit1)
```

Look at the results

- A list of top 10 genes for overall effect of two comparisons (f.liver – liver, f.brain – brain):

```
> topTable(fit2, n=10, adjust=″fdr″)
```

- A list of top 10 genes for f.brain versus brain can be obtained from

```
> topTable(fit2,coef=1,n=10, adjust=″fdr″)
```

- A list of top 10 genes for f.liver versus liver can be obtained from

```
> topTable(fit2,coef=2,n=10, adjust=″fdr” )
```

More …

```
> modt=topTable(fit2, coef=1,sort.by=″t″,
adjust=″fdr″,number=10)[,c(1,3)]
> logodds= topTable(fit2, coef=1,sort.by=″B″,
adjust=″fdr″,number=10)[,c(1,6)]
> logFC=topTable(fit2, coef=1,sort.by=″logFC″,
adjust=″fdr″,number=10)[,c(1,3)]
```

– *modt is preferred as it is most robust, in particular, for the case of missing values*

Acceptance or rejection of each hypothesis test can be decided by

```
> results <- decideTests(fit2,
  adjust=″fdr″, p=0.05)
> summary(results)
```

Note: `decideTests`

- makes a matrix with 0 (not selected) and -1/1 (selected for a specific p-value)
- visualize by Venn diagram

- We can examine which genes respond to either the change between f.brain and brain or the change between f.liver and liver:

```
> p.value <-fit2$F.p.value
```

- What p-value cutoff should be used? One guide is to examine control probe-clusters which are known not to be differentially expressed. We find that the smallest p-value amongst these is about 0.0000027. A cutoff p-value of 0.000001, say, would be conservatively below this.

```
> i <- grep("AFFX",geneNames(raw))
> summary(p.value[i])
```

Note: if you are not given any control gene info like above, skip this step.

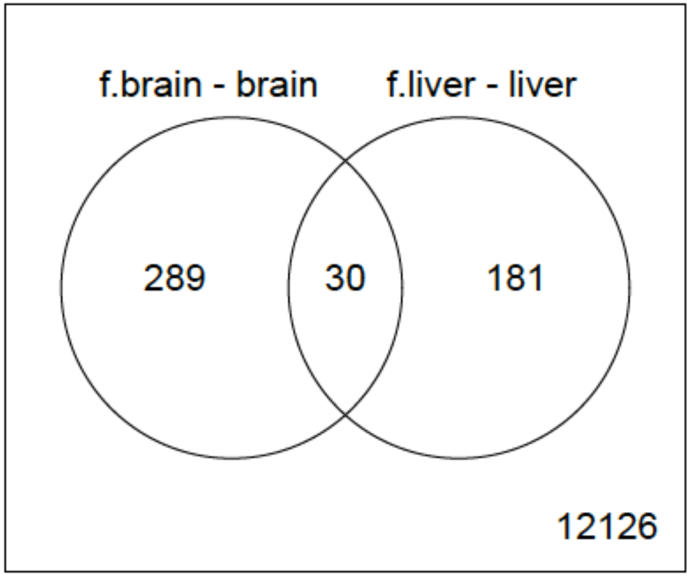
- Now we consider those genes with moderated F-statistics with p-values below 0.000001, and classify these according to whether they are significantly up or down regulated in either comparison:

```
>results <- classifyTestsF(fit2,
  p.value=0.000001)
```

#(Note: you may just use p.value=0.001 or 0.01 or 0.05 if you are not given any control gene info.)

```
> head(results)
           Contrasts
            f.brain - brain f.liver - liver
  100_g_at                0               0
  1000_at                 0               0
  1001_at                 0               0
  1002_f_at               0               0
  1003_s_at               0               0
  1004_at                 0               0
```

```
>summary(results)
f.brain - brain f.liver - liver
-1             136              69
0            12307           12415
1              183              142
> vennDiagram(results)
```



Detailed results – up/down regulation

```
>table(brain.comp=results[,1],liver.comp=results[,
  2])
```

```
             liver.comp
brain.comp      -1      0     1
         -1      4    122    10
         0      65  12126   116
         1       0    167    16
```

Visualize the results of classification & save a file:

```
> jpeg(″my_vennDiagram.jpg″, width =
  960, height = 480, pointsize = 10,
  quality = 100, bg = ″white″ )
> par(mfrow=c(1,2))
> vennDiagram(results,include=″up″,
  main=″up″)
> vennDiagram(results,include=″down″,
  main=″down″)
> dev.off()
```

Save the significant genes to a file

- To select all up-regulated genes for brain:

```
>brain.up <- results[results[,1]==1,  1]
```

To select all down-regulated genes for brain:

```
>brain.down <- results[results[,1]== -1,1 ]
```

To output the gene lists to files:

```
>write.csv(data.frame(brain.up),file=”myresults_brain_up.csv”)
>write.csv(data.frame(brain.down),file=”myresults_brain_down.csv”)
```

Volcano plot

```
>par(mfrow=c(1,2))
>volcanoplot(fit2,coef=1, main="brain",
   names=row.names(fit2$coefficients), highlight=10)
>volcanoplot(fit2,coef=2, main="liver",
   names=row.names(fit2$coefficients), highlight=10)


## highlight the top significant 10 genes for each
   plot
```