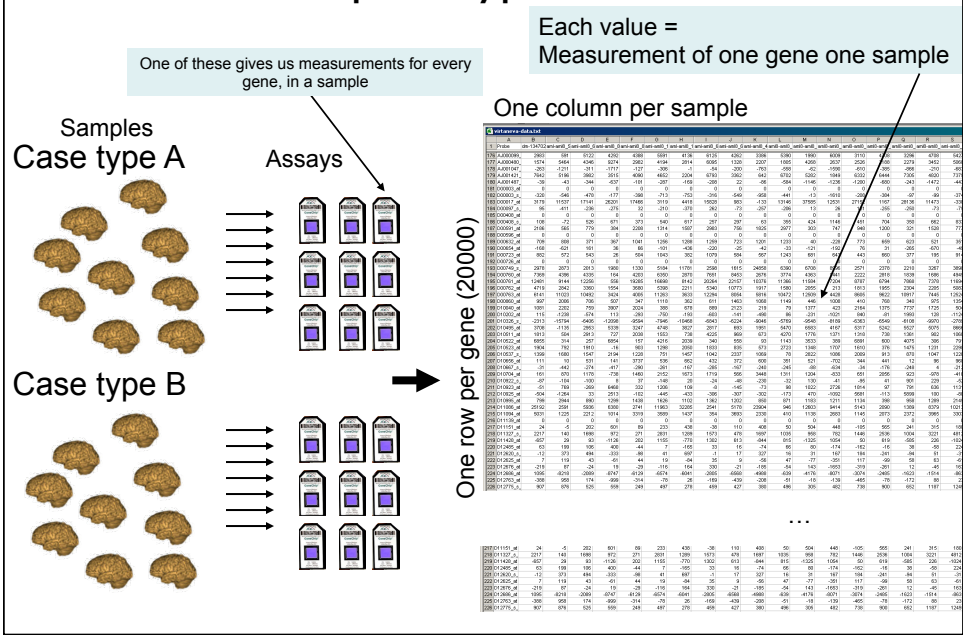
Statistical methods for high-dimensional biology

STAT 540 2015

Data exploration

Paul Pavlidis

Our prototype data set



What you usually get

Data file: contains the measured data

Metadata file: describes the samples

Pat	Patient	Sample	Affected_	Diagnosis
1.1	1	1	Yes	1: Chronic
1.2	1	2	Yes	1: Chronic
1.4	1	4	No	1: Chronic
2.1	2	1	Yes	2: Aggressive
2.2	2	2	Yes	2: Aggressive
2.4	2	4	No	2: Aggressive
3.1	3	1	Yes	2: Aggressive
3.2	3	2	Yes	2: Aggressive
3.3	3	3	No	2: Aggressive
4.1	4	1	Yes	1: Chronic
4.2	4	2	Yes	1: Chronic
4.3	4	3	No	1: Chronic
5.1	5	1	No	1: Chronic
5.2	5	2	Yes	1: Chronic
5.3	5	3	Yes	1: Chronic
7.1	7	1	Yes	2: Aggressive
7.2	7	2	Yes	2: Aggressive
7.3	7	3	Yes	2: Aggressive
8.1	8	1	Yes	1: Chronic
8.2	8	2	Yes	1: Chronic
8.3	8	3	No	1: Chronic

a.k.a. design file, sample information file

Now what?

Ready for exploration

- Understand/get a feel for the data
- Formulate hypotheses / develop models
- Identify problems

The biggest mistake in data analysis

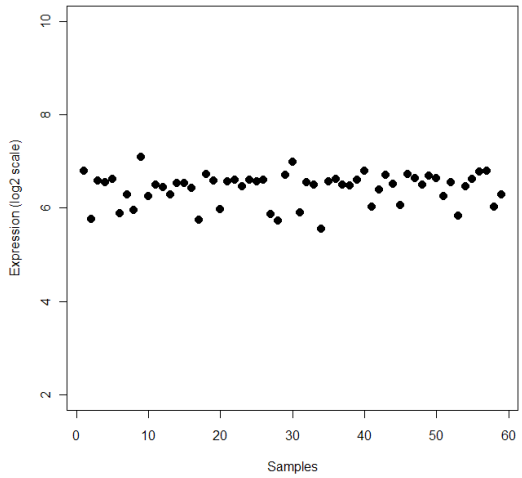
If you don't look at the data … you are going to get in trouble.

- Not just at the beginning, but at every stage.
- That could mean making graphs, or staring at numbers. Probably both.
- "Sanity checks" should make up a lot of your effort.

What are some "first questions" you might ask?

Data for one gene

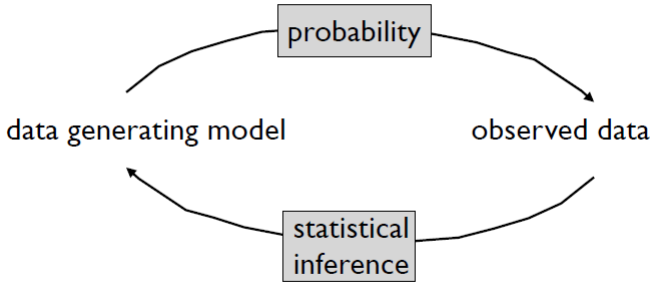
Why do the values vary?



Order of samples is arbitrary at this point

Making sense of the data

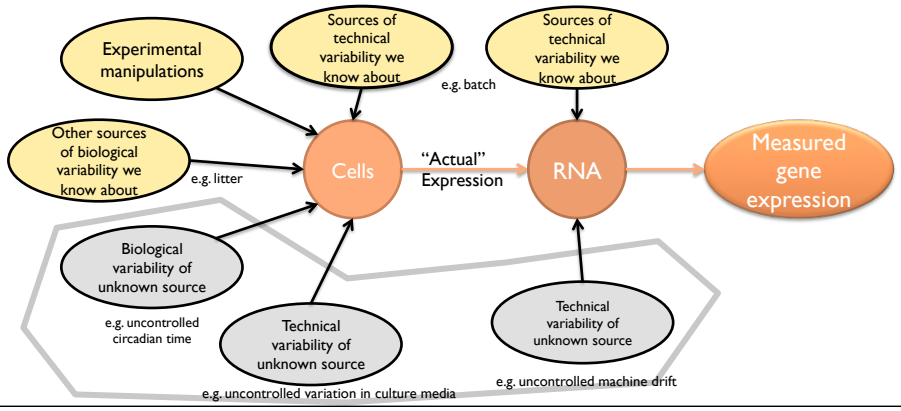
- Review: Data is what we observe, we want to infer something about "where it came from"



Jenny Bryan

Model of where expression data comes from

- The measured expression level of $gene_i$ is the combination of many effects.
- Analysis goal is often to determine relative role of effects, separate interesting from "uninteresting"
- One person's noise is another's signal.



Variability: friend and foe

- First line of defense: Know the enemy
  - You can only "correct" for things you know about.
  - Keep track of potential sources of variance: Batches of reagents, slides, personnel
- Design experiments to minimize impact of technical variability.
  - Avoid batches / minimize batch differences
  - Randomize design with respect to batches
- Replication
  - Biological (important)
  - Technical (*usually* less important but might need to convince yourself)
- Much more later in course

Exploratory numbers

Use as a rough description of the data

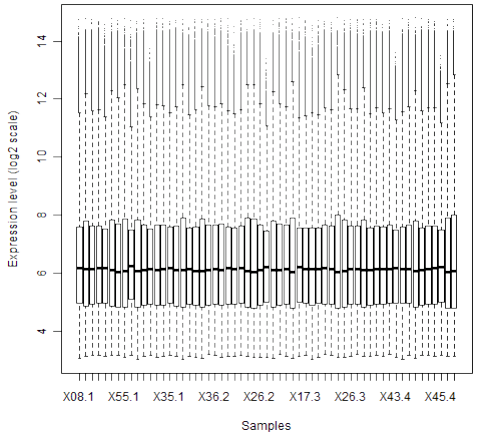
- Range
- Mean, Mode, Variance
- Number of missing values

Examine these values per-sample and per-element (e.g. gene)!

Exploratory graphics

- "Exploratory analysis" is "compute a little and graph a lot"
- I'll show a few simple approaches that are common in genomics
- Basis for examples: a data set of 59 Affymetrix microarrays, each of which have 54675 measurements. RNA was from blood cells of patients being treated for periodontal infection.

Boxplots to compare samples



Quick and dirty; Reasonable tool for summarizing large amounts of data.

Not so great if your distribution is multimodal.

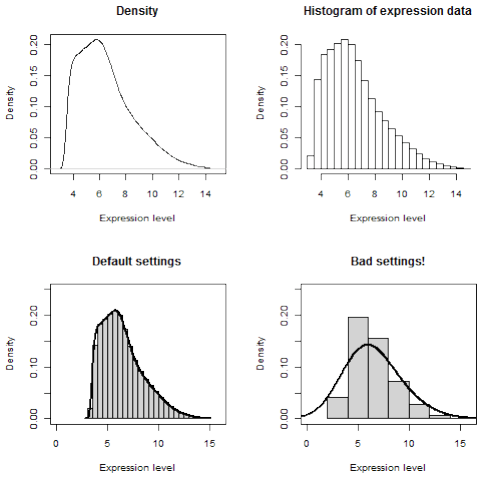
Don't use boxplots if you don't have to – show the actual data (esp. for small sample sizes!)

```
boxplot(dat, pch='.', xlab="Samples", ylab="Expression level (log2 scale)")
dev.print("basic.boxplot.png", device=png, width=500)
boxplot(dat[,1], pch='.', xlab=names(dat)[1], ylab="Expression level (log2 scale)")
```

Histograms

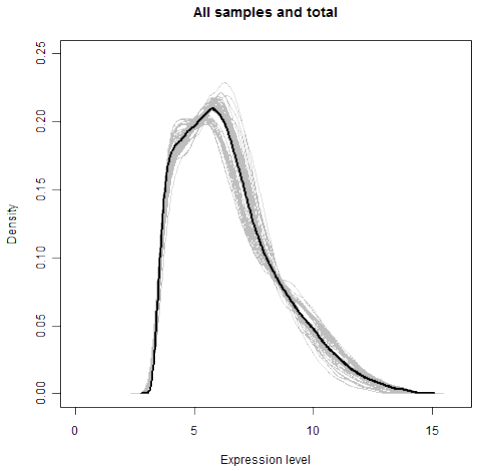
- Box plot's big brother but takes up more space.
- Consider using "density()" instead
- Choose sensible bin sizes and bandwidth

Histogram and density examples

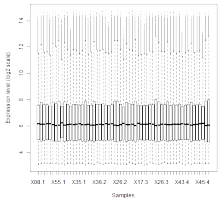



```
par(mfrow=c(2,2))
plot(density(as.matrix(dat)), xlab="Expression level", main="Density")
hist(as.matrix(dat), xlab="Expression level", main="Histogram of expression data", freq=F)
plot(0,xlim=c(0,16), ylim=c(0,0.25), xlab="Expression level", type="n", ylab="Density", main="Combined")
hist(as.matrix(dat), add=T, freq=F, col="lightgrey")
lines(density(as.matrix(dat)), xlab="Expression level", main="Density", lwd=2)
plot(0,xlim=c(0,16), ylim=c(0,0.25), xlab="Expression level", type="n", ylab="Density", main="Bad settings!")
hist(as.matrix(dat), add=T, freq=F, col="lightgrey", breaks=5)
lines(density(as.matrix(dat), bw=2), xlab="Expression level", main="Density", lwd=2)
```

Overlay densities of all samples



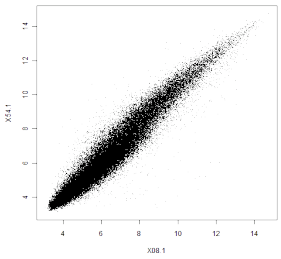
Contrast this with the boxplots.



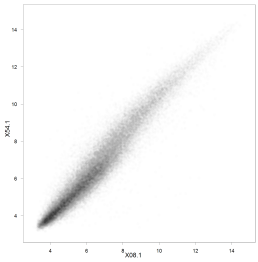
```
plot(0,xlim=c(0,16), ylim=c(0,0.25), xlab="Expression level", type="n", ylab="Density", main="All samples and total")
apply(dat, 2, function(x) { lines(density(x), col="grey") })
lines(density(as.matrix(dat)), xlab="Expression level",  lwd=2)
```

Scatter plots

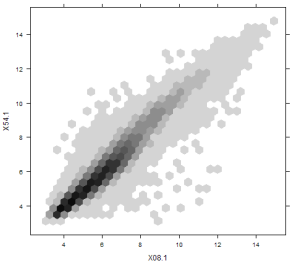
plot()



qplot(… alpha=I(0.01))



hexbinplot()

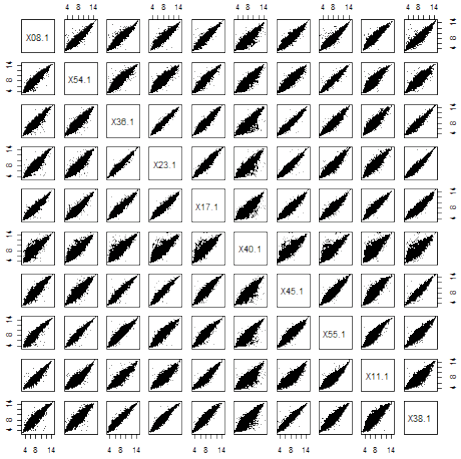



```
plot(dat[,1],dat[,2], xlab=names(dat)[1], ylab=names(dat)[2]  )
plot(dat[,1],dat[,2], xlab=names(dat)[1], ylab=names(dat)[2], pch='.' )
dev.print("basic.dotplot.png", device=png, width=500)
dev.off()

library(ggplot2)
p<-qplot(dat[,1],dat[,2], alpha=I(0.01),  xlab=names(dat)[1], ylab=names(dat)[2])
p+theme_bw()
last_plot() + opts(panel.grid.minor = theme_blank())
last_plot() + opts(panel.grid.major = theme_blank()) ggsave("fancy.dotplot.png")
ggsave("fancy.dotplot.ps") # doesn't work with alpha transparency

library(hexbin) # requires lattice
hexbinplot(dat[,1] ~ dat[,2], xlab=names(dat)[1], ylab=names(dat[2]), colorkey=F)
dev.print("hexbin.dotplot.png", device=png, width=500)
```

Alpha transparency looks cool but output is a bitmap, can't convert to postscript. Hexbin is a good alternative.

See also ggplot2 for more sophisticated options

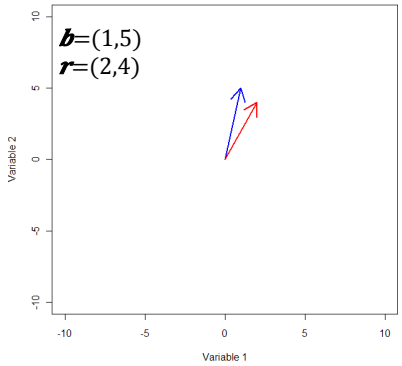
Comparing many pairs of samples



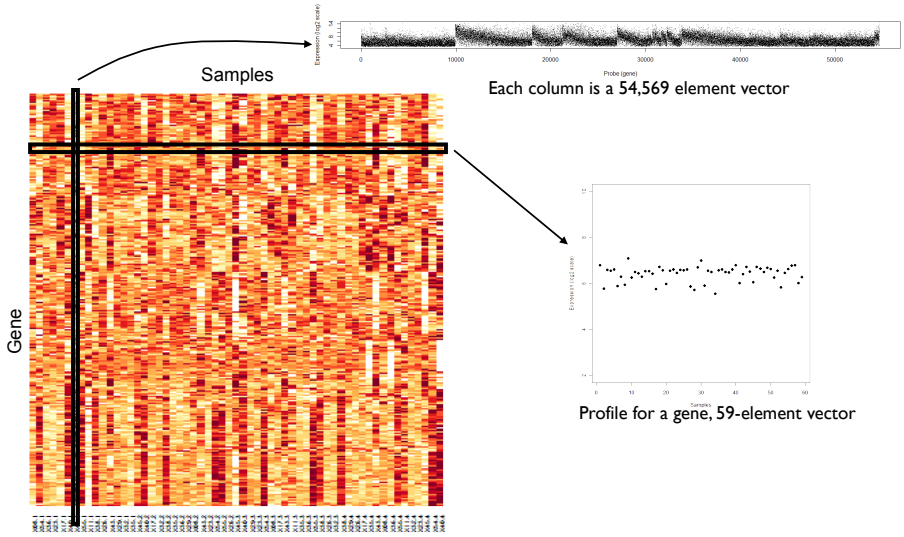

```
# plotting too much data:
pairs(dat, pch='.')
# get "Error in plot.new() : figure margins too large"
pairs(dat[,1:10], pch='.')
```

This is nice but unwieldy (or won't work) for large data sets

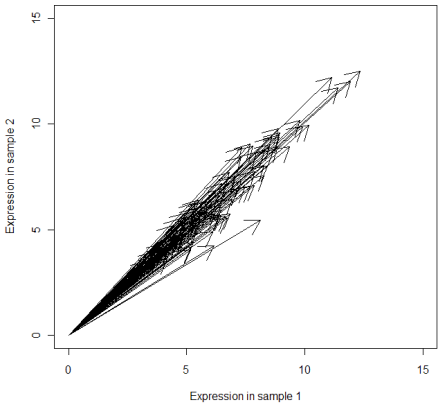
Digression into vectors



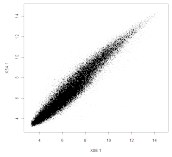
Rows and columns of your data matrix are vectors



First two dimensions of a bunch of gene vectors



Each vector is a gene (just 100 shown).
Variables/Coordinates are "assayed samples"

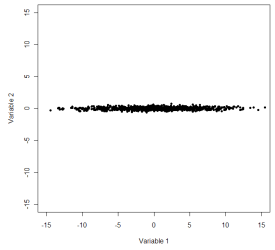


Without the arrows you can plot all the genes

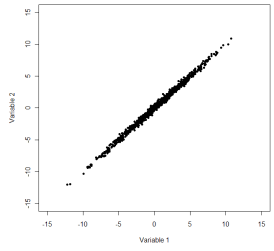
Do the genes form "interesting patterns" in "sample space"?

Data as points in space

- Distributed with a certain shape (and orientation.)
- Want to find "interesting patterns" in these "clouds"
- Do we need all 59 (or 54569) dimensions?
  - Often we are interested in finding interesting lower-dimensional structure in the (or representations of) data (regression, clustering, PCA)
- Motivating examples:



One of these dimensions is useless

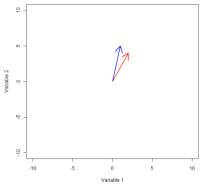


Dimensions are redundant

Comparing vectors: Euclidean distance

- The length of the difference between two vectors.
- In high dimensions, distances get big.

$$d(x, y) = \|x - y\| = \sqrt{\sum (x_i - y_i)^2}$$



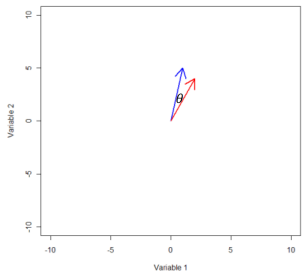
$\boldsymbol{b}=(1,5)$
$\boldsymbol{r}=(2,4)$
$\|\boldsymbol{r}-\boldsymbol{b}\|=?$

For matrices in R: `dist()`

Comparing vectors: Dot product

- Also known as **inner product**
- If two vectors are "nearer" each other, it's bigger
- Can be negative, unlike Euclidean distance

$$x \cdot y = \sum x_i y_i = \cos\theta \|x\| \|y\|$$



$\boldsymbol{b} = (1,5)$
$\boldsymbol{r} = (2,4)$
$\boldsymbol{b} \cdot \boldsymbol{r} = 2 + 20 = 22$

In R: %*%

Covariance

Recall: Sample variance $$s^2(\mathbf{x}) = \frac{\sum_i^N (x_i - \overline{x})^2}{N-1}$$

Sample covariance $$\mathrm{Cov}(\mathbf{x}, \mathbf{y}) = \frac{\sum_i^N (x_i - \overline{x})(y_i - \overline{y})}{N-1}$$

"Are you far from your mean when I am far from mine?"

What happens if the data are mean centered?

Comparing vectors: Pearson correlation

- Two vectors are pos. correlated if they point "in the same direction"
  - Equivalently, if the angle between them is small
- Always between 1 and -1, no matter how many dimensions.
- What happens if the data is **standardized** (mean 0, var 1)?

Covariance:

$$r(x, y) = \frac{\sum (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum (x_i - \overline{x})^2 \sum (y_i - \overline{y})^2}}$$

Variances: "normalizes" the covariance

In R: `cov()` and `cor()`; `scale()` for standardizing

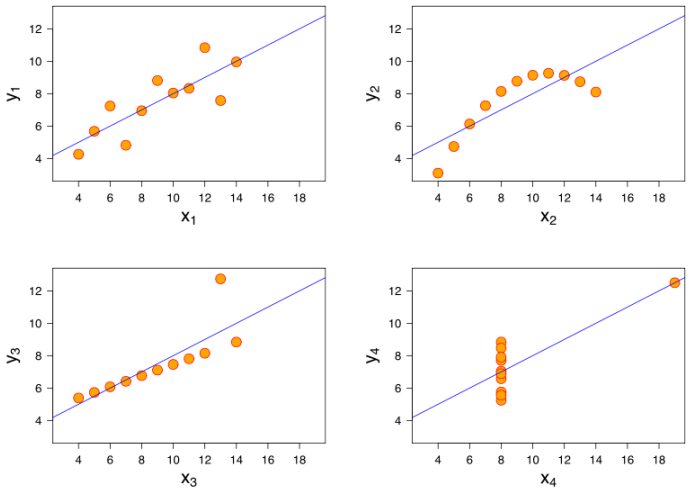
Self-study exercise: think about the metrics on the last few slides for cases of:

- orthogonal vectors
- vectors with opposite direction
- vectors with same direction
- High-dimensional data vs. lower-dimensional data
- What happens if variance of one vector is zero
- etc…

Supplement with sketches and/or plots in R

Look at the data!

Correlation does not imply "interesting", and lack of correlation does not mean "boring"

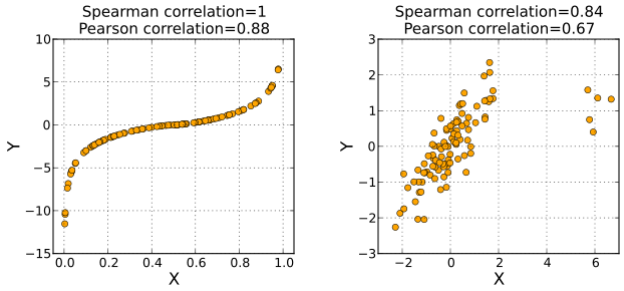


http://en.wikipedia.org/wiki/File:Anscombe%27s_quartet_3.svg

Remedy for some problems: ranks

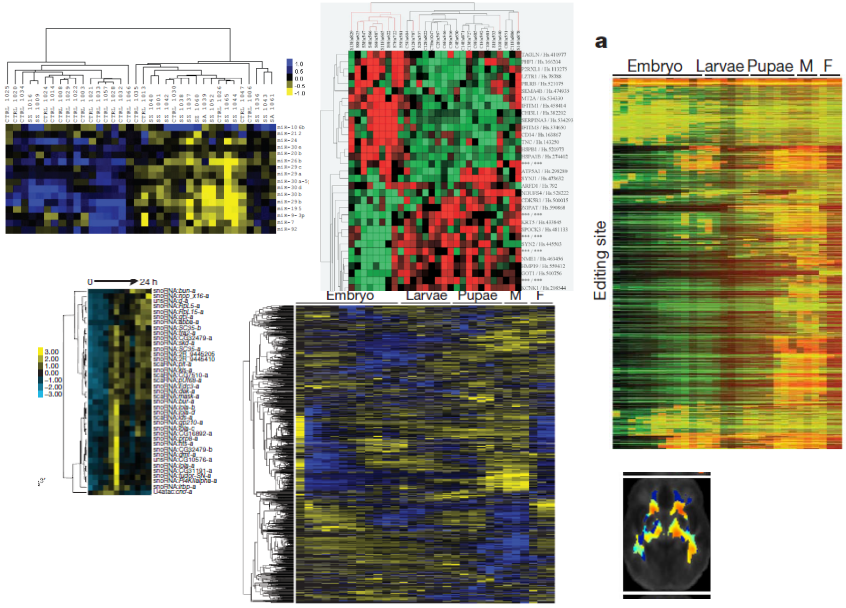
The correlation of the ranks is robust to "non-linear" relationships and outliers.

- Annoyance: dealing with ties



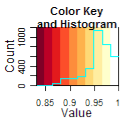
Wikipedia

Heat maps

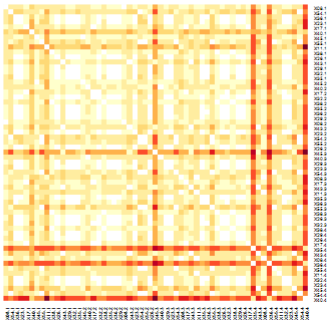


Using heat maps to compare assays

- Pairwise Pearson correlations of entire sample expression profiles



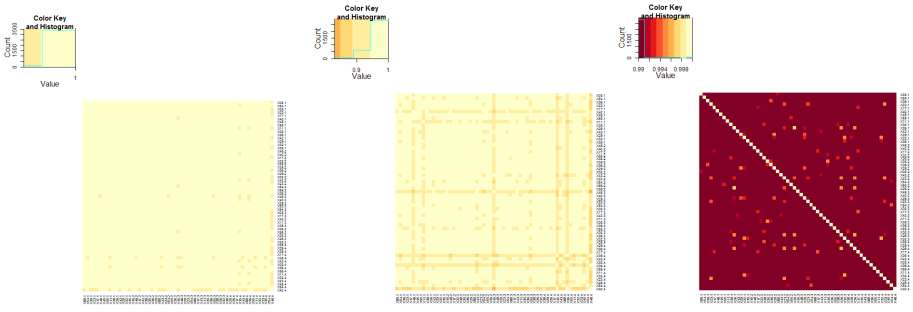
- This method loses some information, but is suitable for quite large data sets.
- Not for comparing features (genes)!



```
library(gplots)
library(RColorBrewer)
cols<-c(rev(brewer.pal(9,"YlOrRd")), "#FFFFFF")
heatmap.2(cor(dat), Rowv=NA, Colv=NA, symm=T,
trace="none", dendrogram="none", col=cols, cexCol=0.5,
cexRow=0.5)
dev.print("heatmap.png", device=png, width=500)
```

Alternative: matrix2png (http://www.chibi.ubc.ca/matrix2png)

Choose an appropriate range

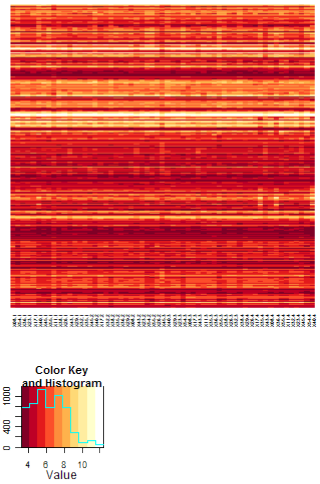


R sets "reasonable" ranges for you; but you can control it (`breaks`) and will often want to.

Setting up data heat maps

- I've taken 500 rows at random from my big data set
- The heat map here was created using this data "as is".

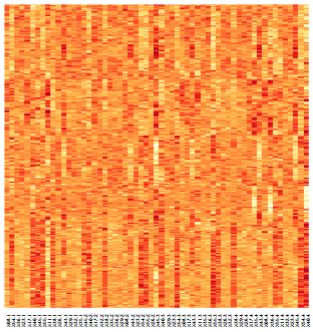
- I did cheat a little: The row ordering is not random.
- Note that this lecture doesn't get into dealing with missing data – how to evaluate it and visualize it



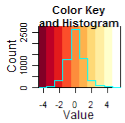

```
heatmap.2(bitOfData, Rowv=NA, Colv=NA, scale=NULL, trace="none", dendrogram="none", col=cols, cexCol=0.5, labRow=NA)
```

Revision 1

- Same input data, but each row is scaled to have mean 0 and variance 1 (z-scores)
  - Subtract the mean; divide by the standard deviation. use `scale()` on the data rows.
- It is now easier to compare the rows and seem some structure.
- But looks kind of bland compared to ones you often see in the literature



Note: R heatmap scales rows by default. You must disable this to give yourself control over how it turns out.



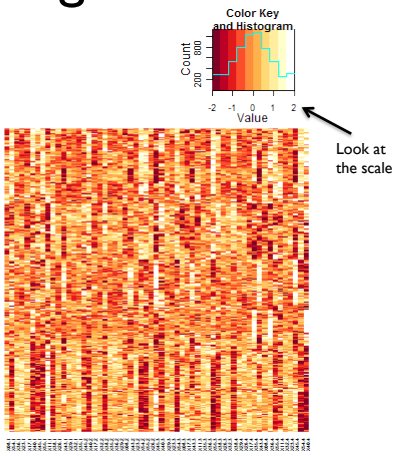

```
heatmap.2(scaledBit, Rowv=NA, Colv=NA, scale=NULL, trace="none", dendrogram="none", col=cols, cexCol=0.5,
cexRow=0.5, labRow=NA)
```

Revision 2: Adjusting contrast

- Same input data again, but after scaling I clipped the range of values to {-2,2}
- This means "anything more than two standard deviations from the row mean is set to 2"
- Now extremes have no effect; "Higher contrast"

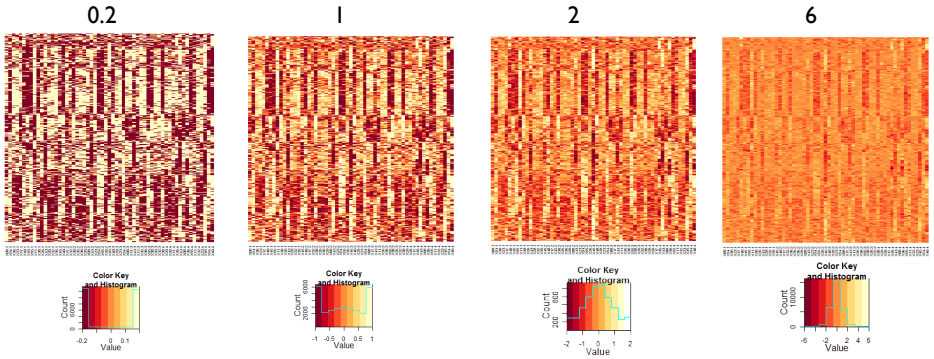
- Limit values of the range limit of 2 or 3 are usually good.

Heat maps you see in the literature are almost always set up this way. Be aware of what's really going on.



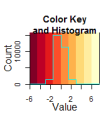

```
limit<-2
trim.scaledBit<-scaledBit
trim.scaledBit[which(trim.scaledBit < -limit)]<- -limit
trim.scaledBit[which(trim.scaledBit > limit)]<- limit
heatmap.2(trim.scaledBit, Rowv=NA, Colv=NA, scale=NULL, trace="none", dendrogram="none",
col=cols, cexCol=0.5, cexRow=0.5, labRow=NA)
```

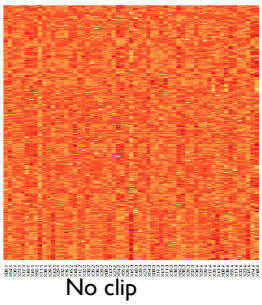
Varying clip limit



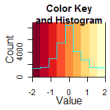
Another reason why clipping is and isn't good

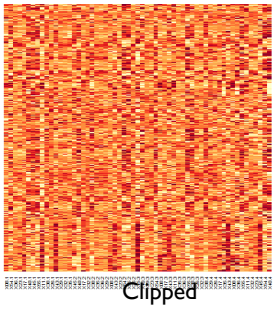
I added random "spiky" values to the data (I just multiplied them by 3)
Clipping hides these "outliers" but allows us to see variation in the bulk of the data





No clip





Clipped

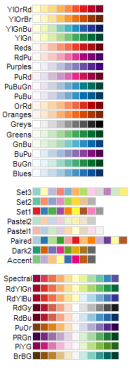
Plotting too much data

The entire data set.

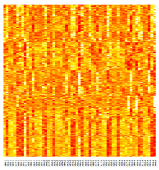
If the cells are less than 1 pixel, everything starts to turn to mush and can even be misleading.

(This won't work in R unless you print directly to a file.)

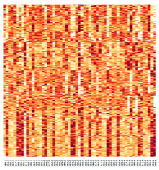
Choice of colours



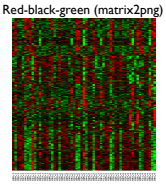
R default



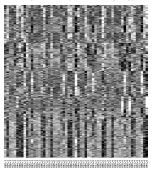
RColorBrewer YlOrRd



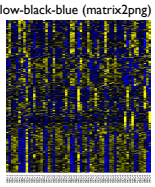
Red-black-green (matrix2png)



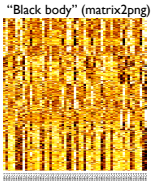
Greyscale (matrix2png)



Yellow-black-blue (matrix2png)

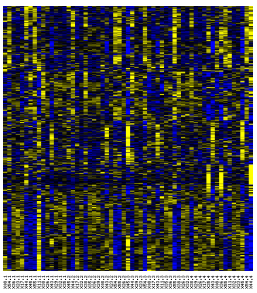


"Black body" (matrix2png)

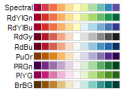


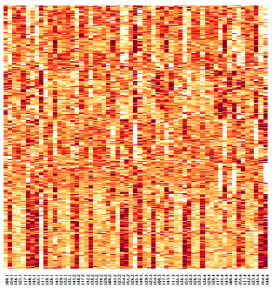
- R defaults: ketchup and mustard
- RColorBrewer: good maps based on work of visualization expert (matrix2png uses them too)
- Red-black-green you often see in papers is **bad choice** for colour blind individuals.
- Blue-black-yellow is better
- Greyscale: loss of dynamic range, but cheaper to publish!
- My favourite: matrix2png "black body"
- Humans can't really tell the difference between a 8 and 16 colour scale.

Divergent vs. sequential maps



Colours pass through black at (in this case) zero. Yellow="Below the mean", Blue="Above the mean" for the row. This can be the right thing to do, especially if your original data are naturally "symmetric" around zero (or some other value). Otherwise it might just be confusing.

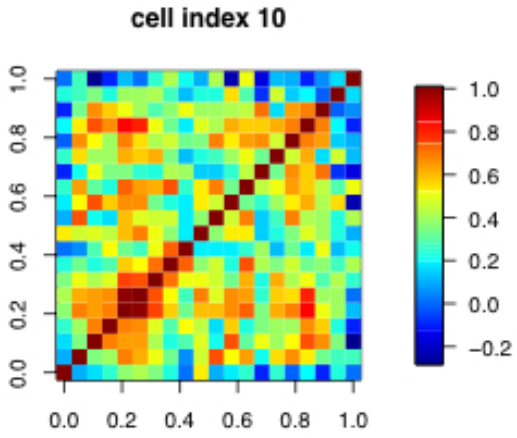




Colours go from light to dark. Darker colours mean "more".

A confusing heat map

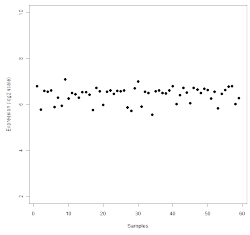
(at least for me; one can get used to anything)



http://mikelove.wordpress.com/category/visualization/

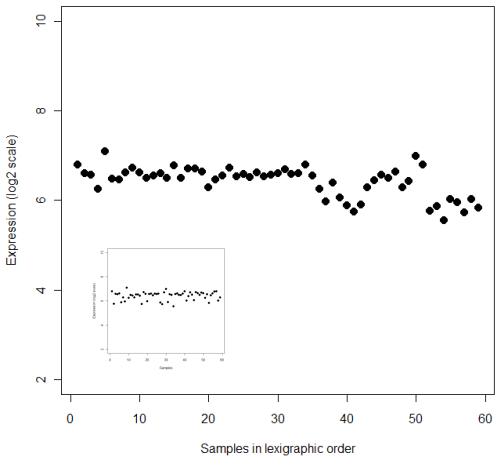
Slicing and dicing

- I mean: Viewing data arranged or grouped by factors of interest
- In my example, I have multiple "visits" for each "patient"
  - Any interesting trends we can spot?
  - Are intra-subject samples more similar than inter-subject samples?



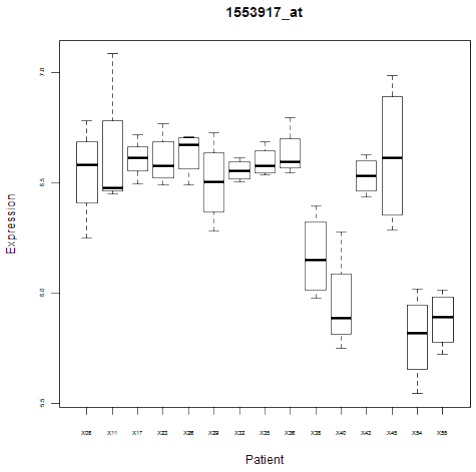
Playing around with one gene

- I sorted the samples by their names
- Is this structure meaningful? Is it interesting?
- Maybe we need more information.
- (And some statistics)



Explicitly group by patient

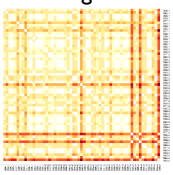
- We're developing the concept that we can look for "factors" that "explain" *some* of the variation.
- Is this interesting? We don't know, but it suggests that accounting for Patient differences might be reasonable.
- Need to get more comprehensive view …

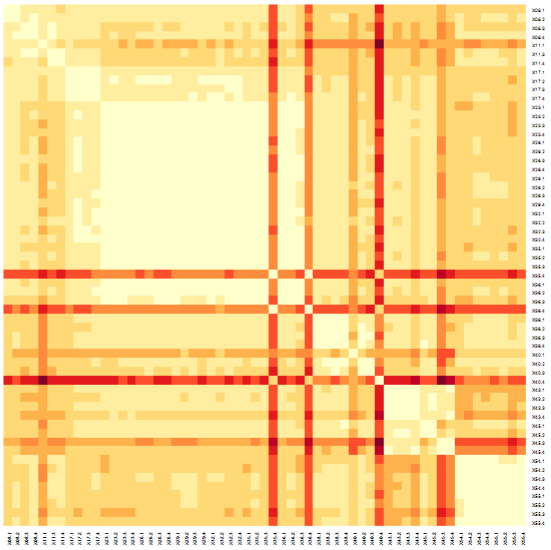


(Boxplots not ideal for this, but easy)

Arrange data by Patient

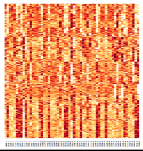
Original

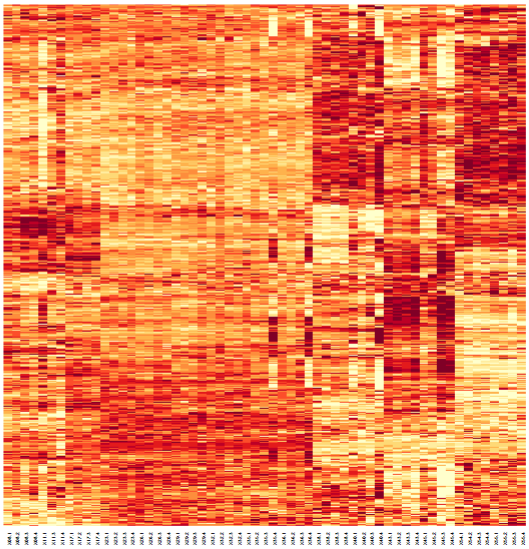




View of (part of) the data organized by Patient

(Again, row order is not random either)





Summary

- Sanity checks
- Use graphs to look at the data, slicing and dicing – build up a library of techniques that work for your data
- Additional exploratory techniques will be discussed later in the course
  - Clustering
  - PCA
- Also: missing values