Removing Unwanted Variation in Machine Learning for Personalized Medicine

with Johann Gagnon-Bartsch and Laurent Jacob

European Marie Curie Network for MLPM. Barcelona, 20 May 2016

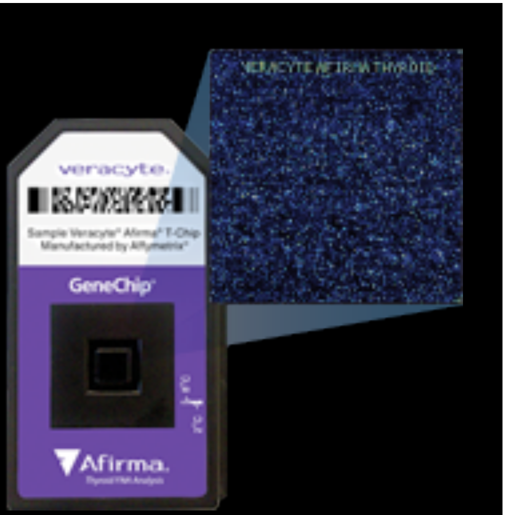
Photo: Bernard Gagnon

Apology, Motivation and Declaration of Conflict of Interest

SBA73 from Sabadell, Catalunya

Afirma®

Thyroid FNA Analysis



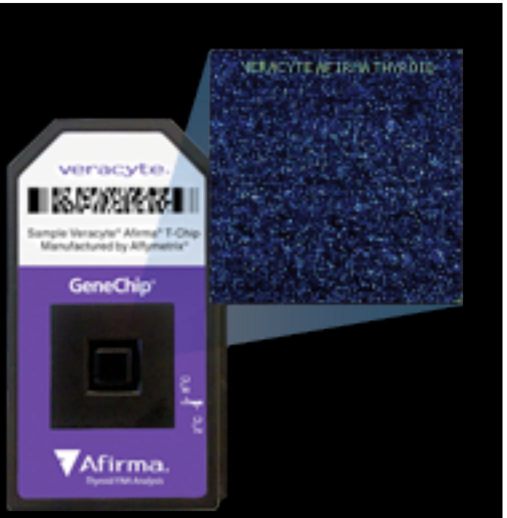
Over 500,000 thyroid nodule fine needle aspiration (FNA) procedures were performed in the US in 2011. FNA samples can be challenging to interpret and produce indeterminate results in 15% to 30% of cases.

Guidelines recommended that most of these patients undergo a diagnostic thyroid surgery to assess whether the nodules are benign or malignant. 70%-80% of the time, the nodules prove to be benign.

The Afirma Gene Expression Classifier (GEC), helps physicians reduce the number of surgeries by preoperatively identifying benign nodules among those that were classified by cytopathology as indeterminate.

Afirma®

Thyroid FNA Analysis



Over 500,000 thyroid nodule fine needle aspiration (FNA) procedures were performed in the US in 2011. FNA samples can be challenging to interpret and produce indeterminate results in 15% to 30% of cases.

I'm on the Scientific Advisory Board of Veracyte and receive money from them.

The Afirma Gene Expression Classifier (GEC), helps physicians reduce the # of avoidable surgeries by preoperatively identifying benign nodules among those that were classified by cytopathology as indeterminate.

Introduction to our RUV methods

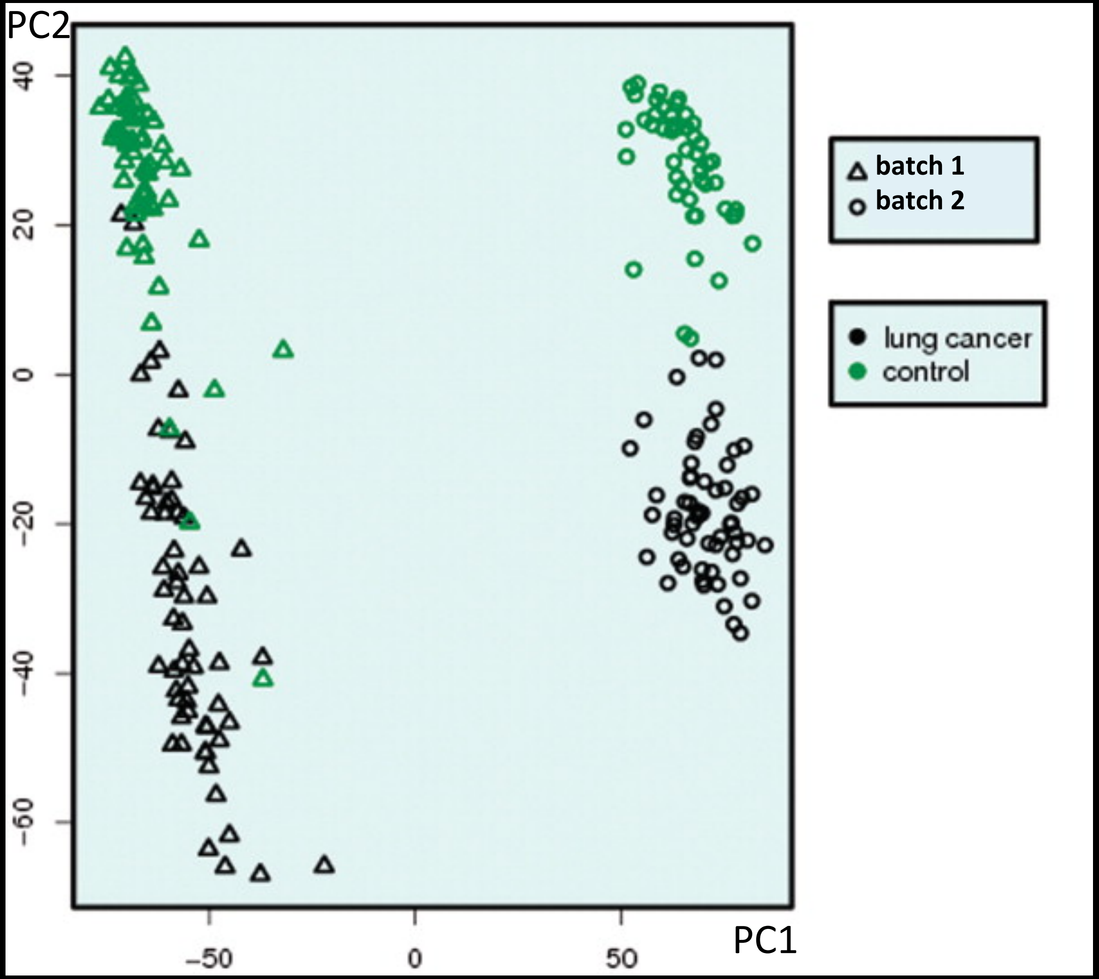
The problem

High-dimensional (e.g. omic or fMRI) data can be affected by unwanted variation.

For example, batch effects due to time, space, equipment, operators, reagents, sample source, sample quality, environmental conditions,…the list goes on…

Artifact can overwhelm biology

Sample principal component scores



Gene expression data. Adapted from Lazar C *et al.* ***Brief Bioinform*** *2013*

Some scientific goals sought using gene expression microarrays

Differential Expression

Classification

Clustering

Unwanted variation can reduce precision and add bias (via confounding), leading to false positives and false negatives, poor classifiers and artificial clusters.

Aim for today

To discuss some new ways of

- identifying and removing (i.e. adjusting for) unwanted factors, when the goal is **classification**, and
- telling whether or not it helped.

“Our” model (brief refs later)

m (10s-1,000s) samples, *n (10s of 1,000s)* genes, *k (≤ m-p)* UV factors

$$Y_{m\times n} = X_{m\times p}\beta_{p\times n} + W_{m\times k}\alpha_{k\times n} + \varepsilon_{m\times n}$$

where

Y is a matrix of gene expression measurents, observed,

X carries the factors of interest, observed in a training set, unobserved in a test set

β are gene coefficients, unobserved,

W carries unwanted variation factors, unobserved,

α are gene coefficients, unobserved,

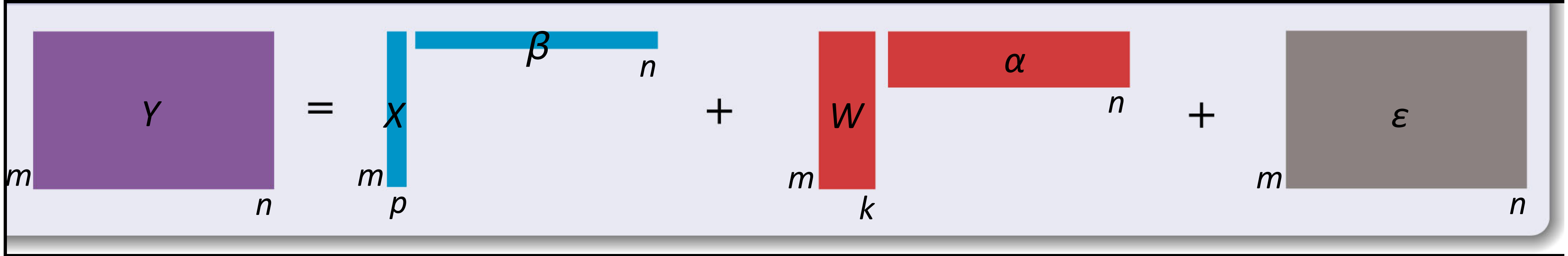
ε are errors, unobserved.

Concrete example

With our *Afirma-T* example, we could put x_i=*-1* if sample *i* is benign, x_i *= +1* if sample *i* is malignant.

The w_i for this example could capture batch effects in reagents, in chips, processing dates, operators, and other things (remember: we're treating them as unobserved.

Our model in pictures



$$y_{ij} = x_i\beta_j + w_i\alpha_j + \varepsilon_{ij}$$

The ε_{ij} are all $(0, \sigma^2_j)$, uncorrelated with each other and all else. We resist the temptation to make assumptions about the $\{\alpha_j\}$.

Our goal: classification

That is, we have *y* but ***don't know*** *X (or W)* for our test and target set samples.

Before we get there, we'll discuss estimating *β* as we would in a training set ***with known*** *X*.

Our model, 2

$$Y_{m\times n} = X_{m\times p}\beta_{p\times n} + W_{m\times k}\alpha_{k\times n} + \varepsilon_{m\times n}$$

Initial goal: to estimate β

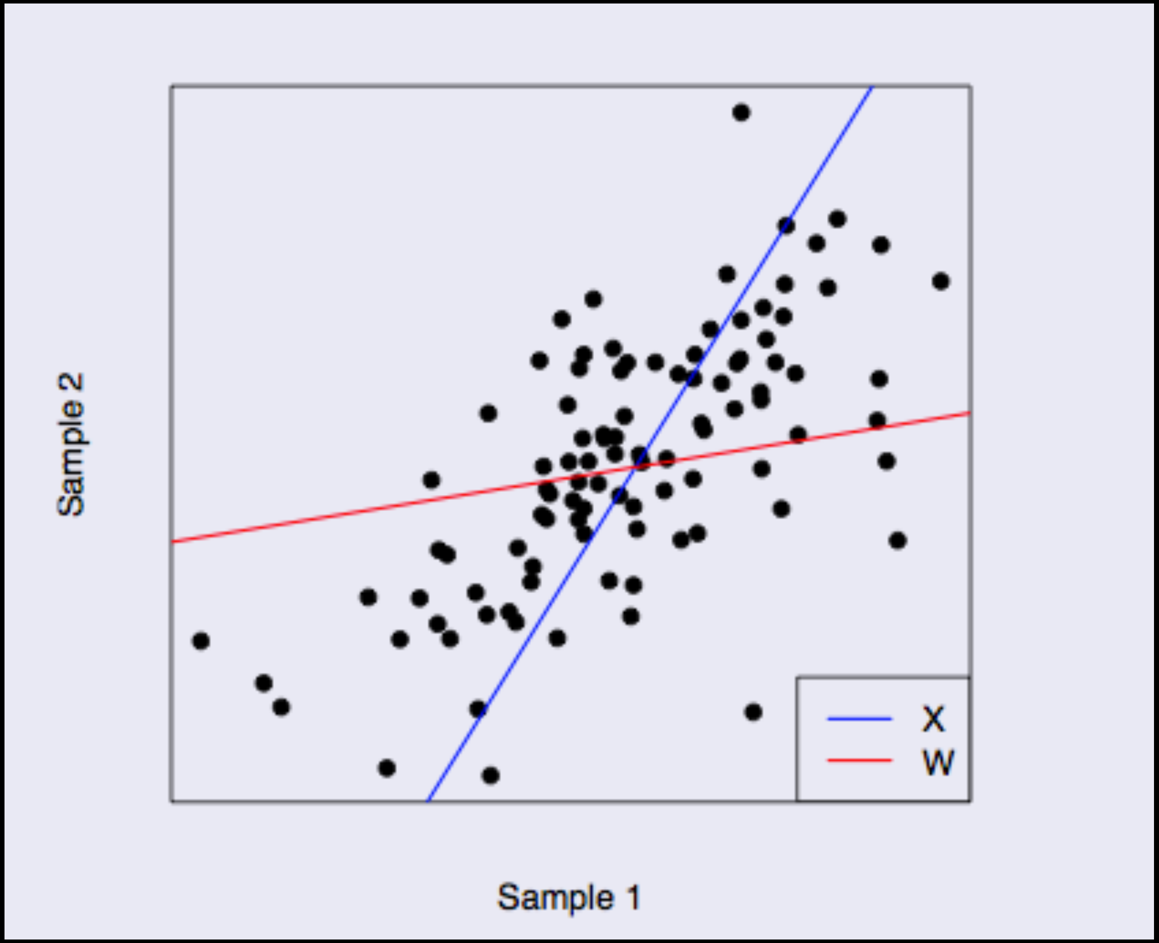
Note: W unobserved, o/w standard linear model

“Our” strategy: use factor analysis to estimate W

Some ways of dealing with these and related problems with microarrays

- Standard linear regression (many)
- EB linear regression (ComBat, Johnson *et al*, 2007)
- Naïve factor analysis (SVD, several)
- Bayes (Lucas *et al*, 2006, Stegle *et al*, 2008)
- Surrogate Variable Analysis (Leek & Storey, 2007)
- Mixed model analysis (Kang *et al,* 2008, Listgarten *et al,* 2012)

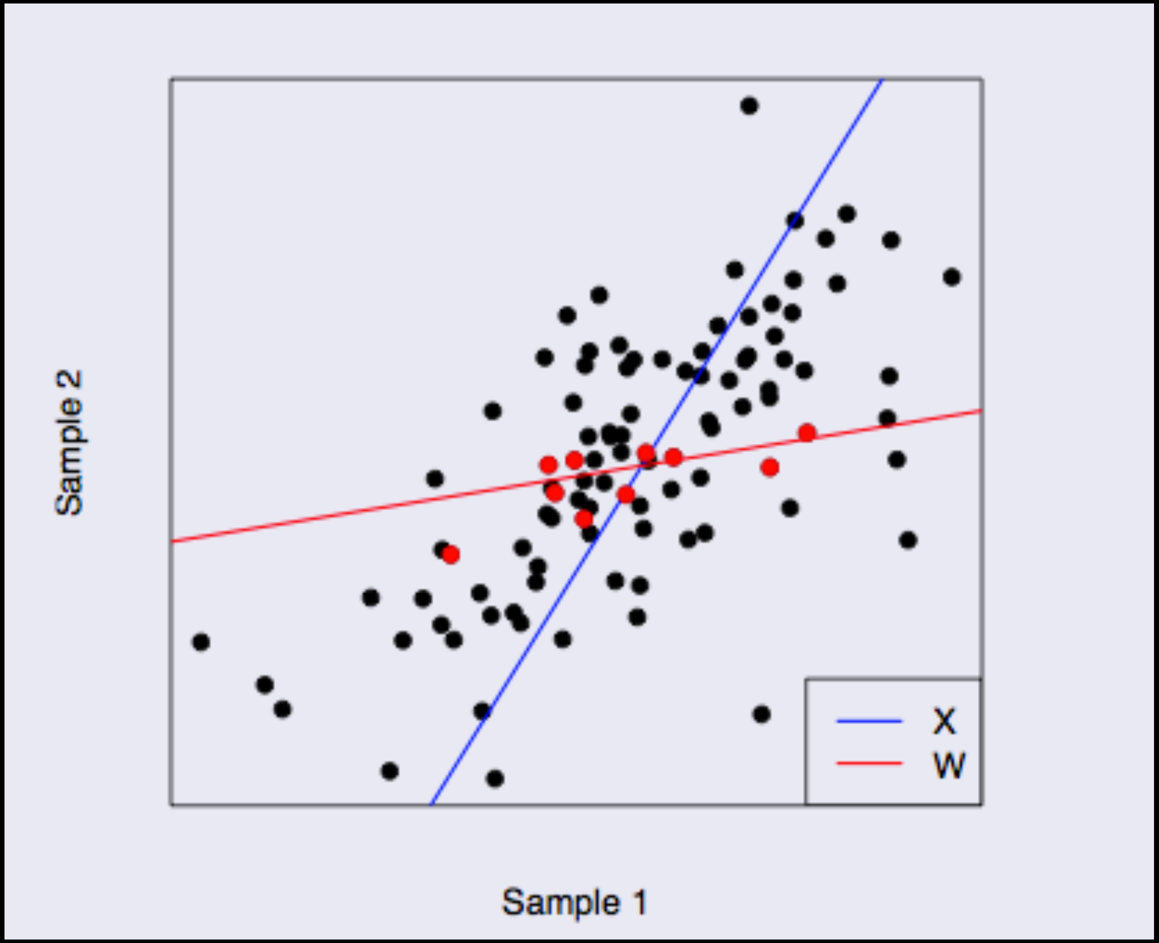
Identifiability: we don't know the correlation of *W* (*k=1*) with *X*



Two samples
$x_1 = w_1 = 1$
$x_2 = x,\ w_2 = w$
Dots are genes

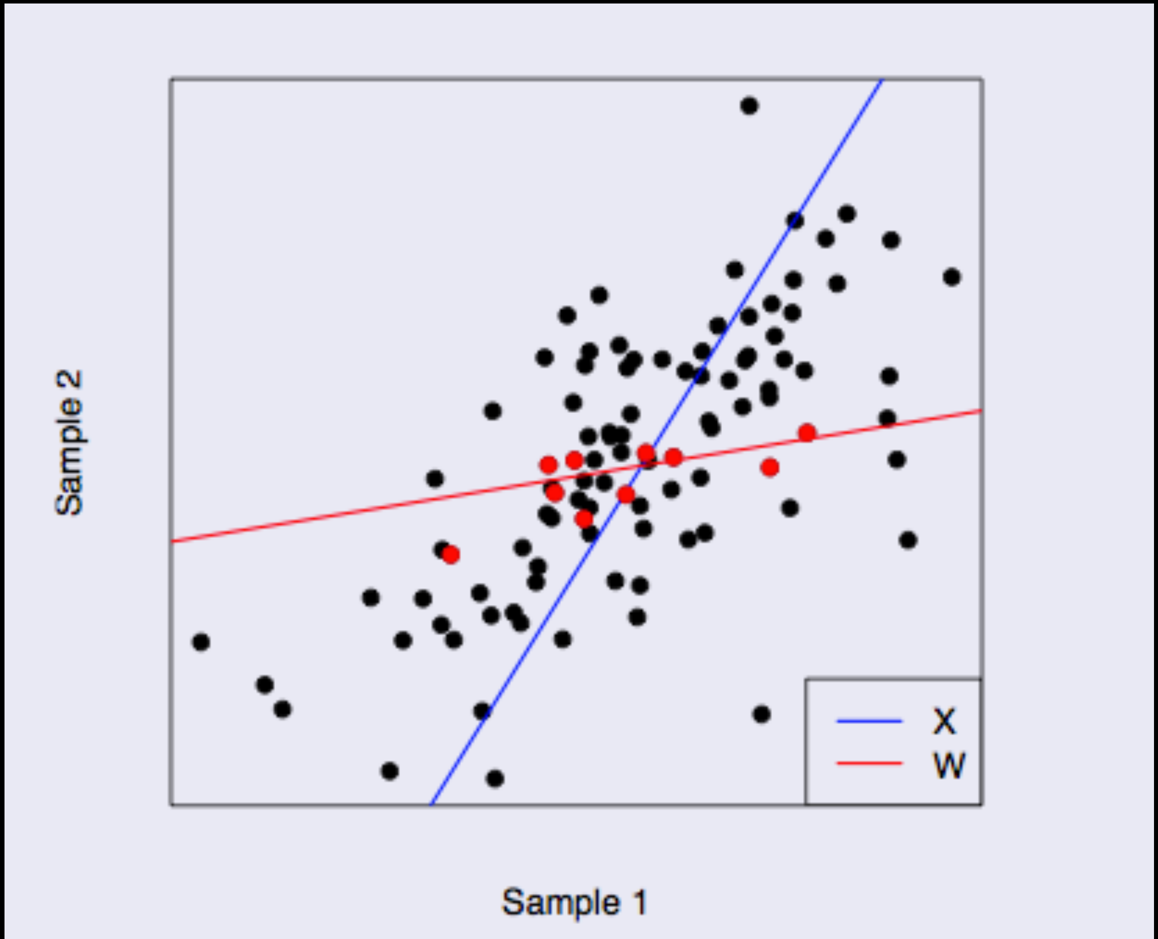
$$(y_{lj}, y_{2j}) = (\beta_j + a_j + \varepsilon_{1j}\ ,\ x\beta_j + w a_j + \varepsilon_{2j})$$

We might have genes *j* not affected by *X*



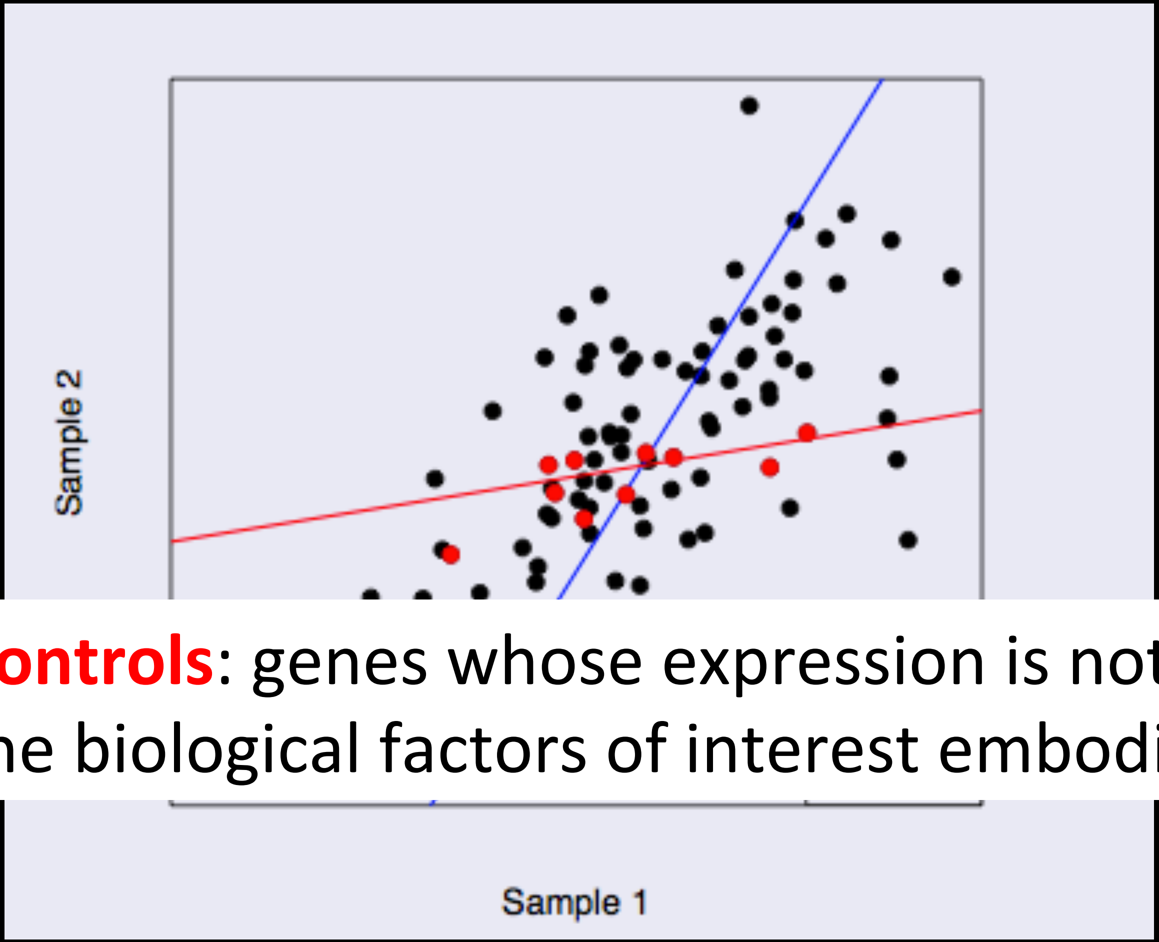
$$(y_{lj}, y_{2j}) = (a_j + \varepsilon_{1j}\,,\; wa_j + \varepsilon_{2j})$$

We might have genes *j* not affected by *X*



$$(y_{lj}, y_{2j}) = (a_j + \varepsilon_{1j}\ ,\ wa_j + \varepsilon_{2j})$$

We might have genes *j* not affected by *X*

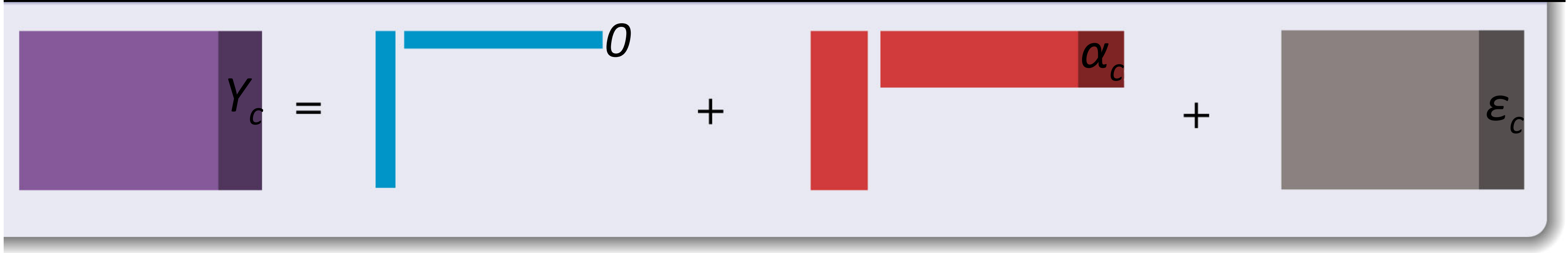


Negative controls: genes whose expression is not associated with the biological factors of interest embodied in *X*

$$(y_{lj}, y_{2j}) = (a_j + \varepsilon_{1j}\ ,\ wa_j + \varepsilon_{2j})$$

"Our" solution: Use control genes

Negative controls: Assume $\beta_j = 0$.



Positive controls: Assume $\beta_j \neq 0$.

"controls" in this context means
"controls w.r.t. differential expression"

Using the negative controls $_c$

$$Y_c = W\alpha_c + \varepsilon_c$$

Just do a factor analysis on the negative controls!

Examples of negative controls

- housekeeping (HK) genes,
- spiked-in controls
- suitable empirical controls

This works!

Introducing the two-step: RUV-2

1. Do a factor analysis on Y_c to estimate W.
2. Then regress Y on X and $W^\wedge$, the estimated W, to get an estimate of β adjusted for $W^\wedge$.

There are many ways to do the factor analysis, but we just use

SVD: Write $Y_c = U \Lambda V^T$, then put $W^\wedge = U^{(k)}$ *(first k columns)*

Issues: choice of *k,* and can we do better? Yes: RUV-4

Introducing RUV-inv

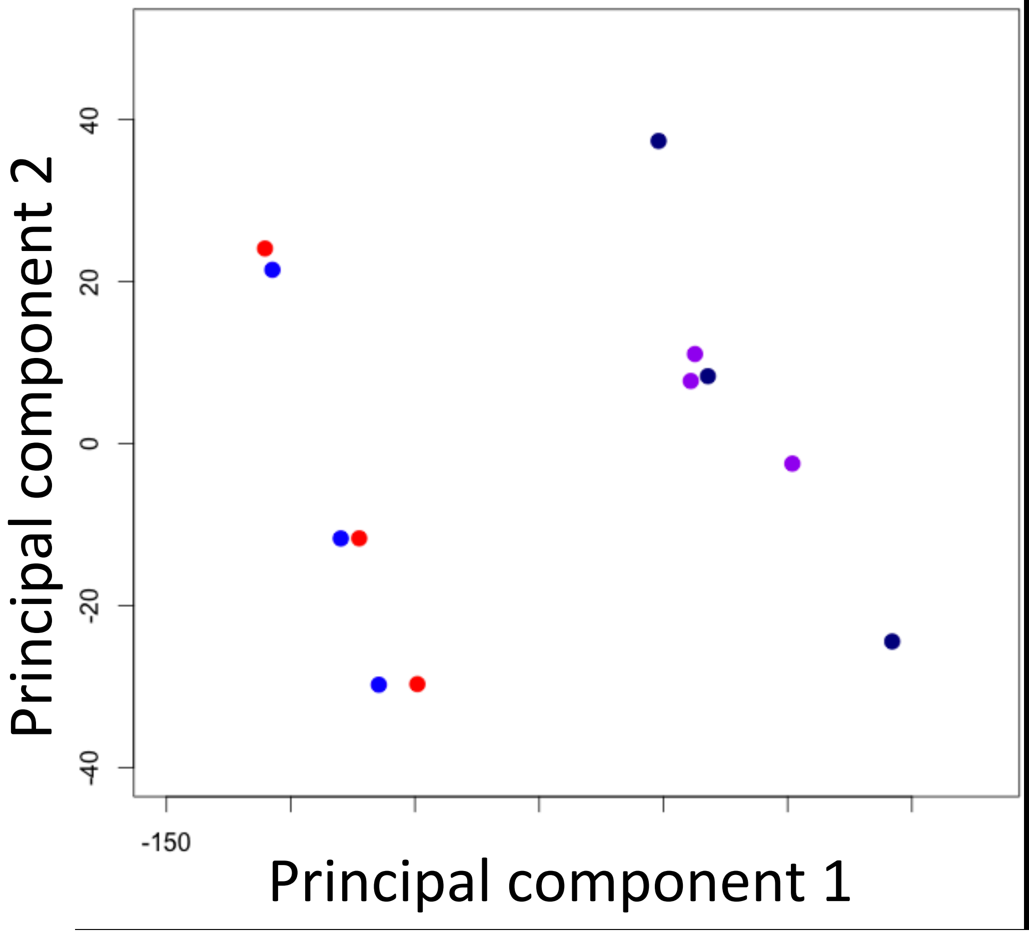
We start with RUV-4 (UCB Stat Tech Rep 820), and put *k=m-1* (the largest possible value when *p=1*). We don't need an SVD, and we find

$$\hat{\beta}^{RUV-inv} = [X^t(Y_cY_c^t)^{-1}X]^{-1}X^t(Y_cY_c^t)^{-1}Y$$

This is the generalized least squares estimator using a covariance matrix based on data from the negative control genes (others use all genes), but we estimate SEs differently.

A microarray experiment with central retina tissue from the *rd1* mouse: *4 times x 3*

rd1 is a mouse model of *retinitis pigmentosa:* loss of rod photoreceptors, followed by that of cone photoreceptors



Light blue: 2 months

Dark blue: 4 months

Purple: 6 months

Red: 8 months

Very severe batch effects

Ideally we would have seen 4 tight groups of 3 ●, ●, ● and ● resp.

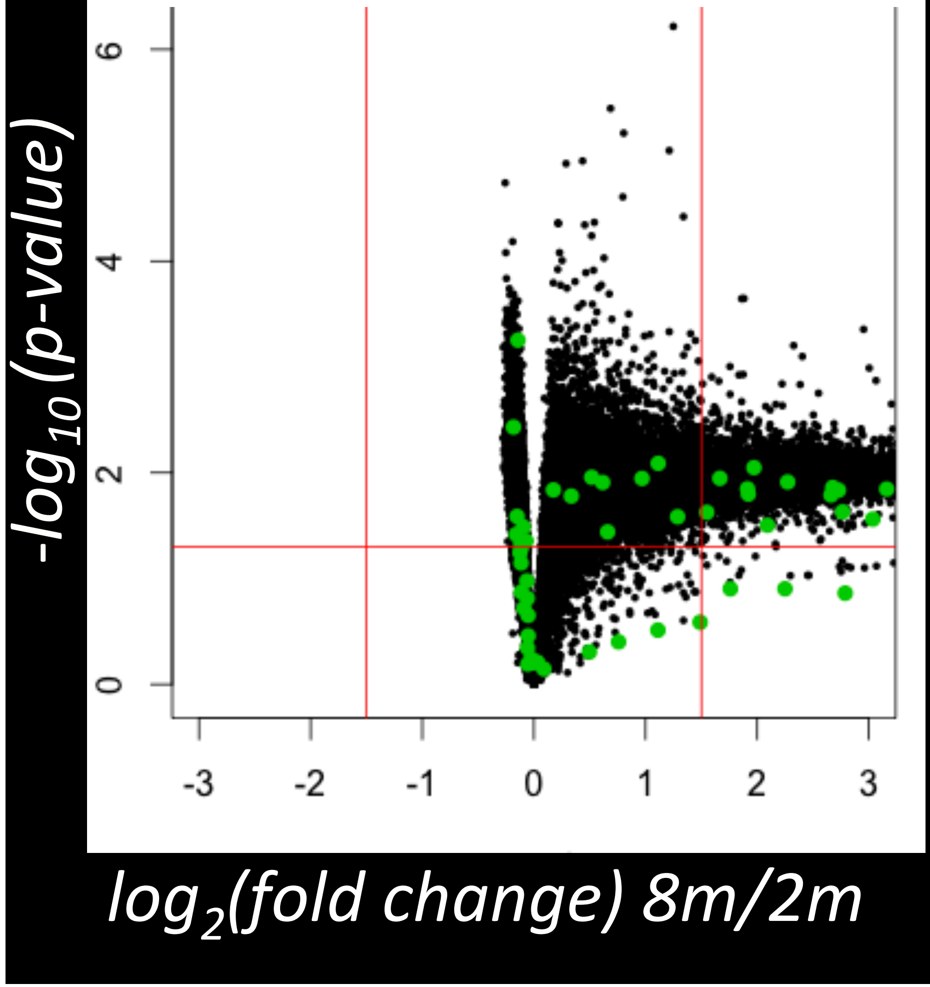
Removing severe batch effects

- Initially no significantly downregulated retinal genes were found between 2 and 8 months (left *volcano plot* on the next slide).

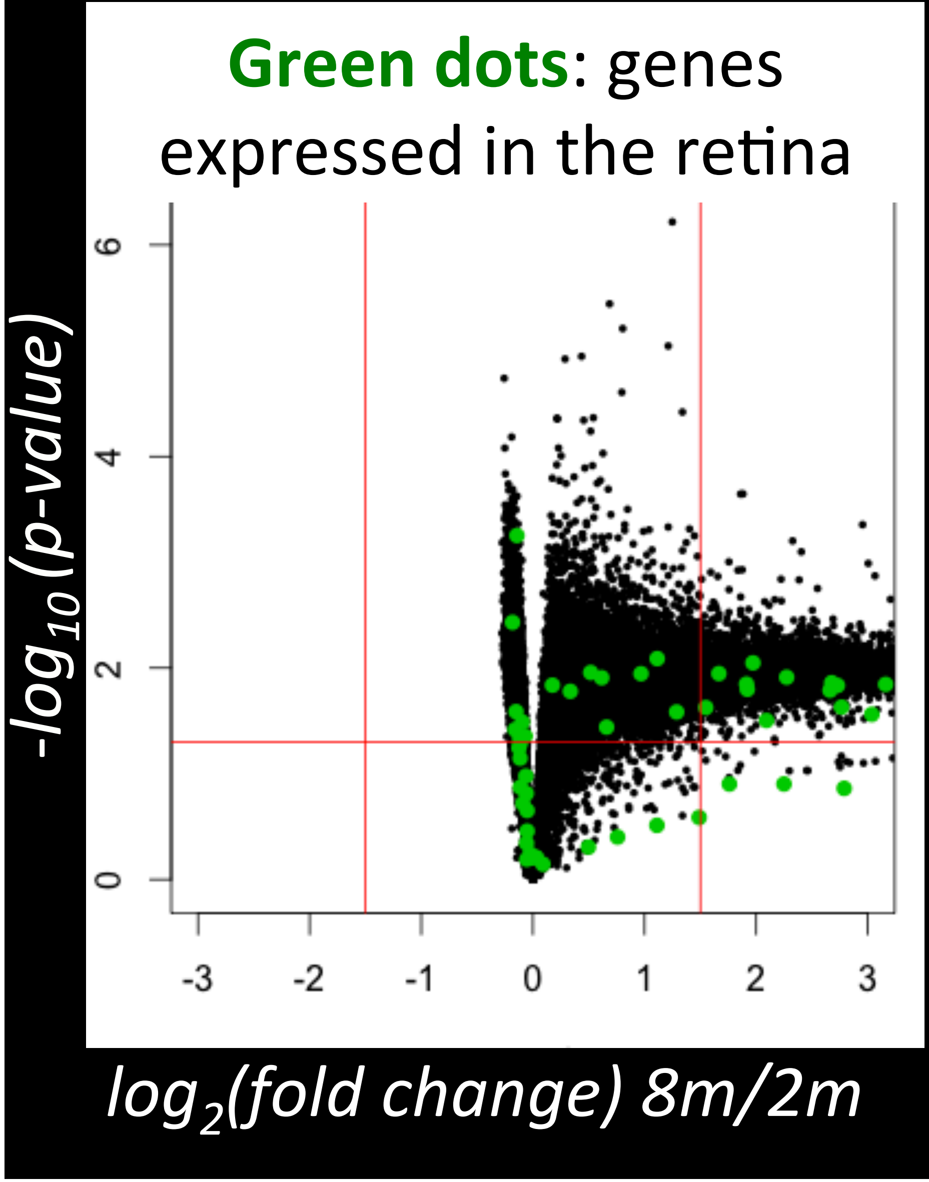
- Using RUV-inv (right plot), we were able to find several significantly down-regulated retinal, even cone-specific genes, which were later confirmed.

Standard analysis

Green dots: genes expressed in the retina

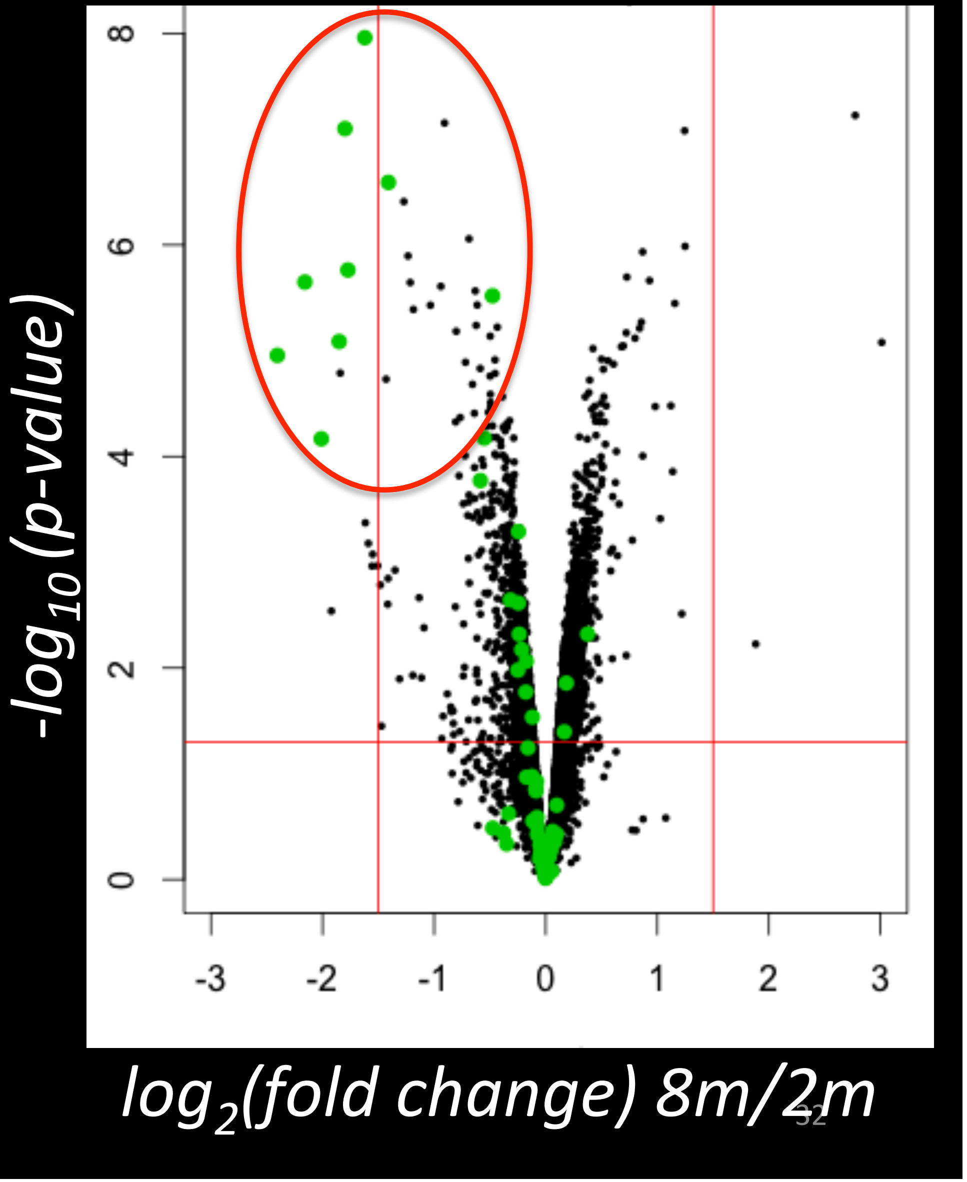


Standard analysis



Green dots: genes expressed in the retina

Analysis with RUVinv



Are there any questions?

Classification

training, test and target sets

What is unwanted variation?

What is unwanted variation?

- Variation that has no predictive power

Hypothetical example: Suppose we want to classify tumors into one of two types, A or E.

Suppose Asians tend to get type A, and Europeans type E.

Is ethnicity “wanted” or “unwanted"?

What if it is easy to classify by ethnicity, but hard / impossible to classify by tissue type per se?

Same question, but now the “unwanted variation” is a lab effect – and one lab is in Agra and the other is in Essen.

- Variation that cannot be assumed to be stationary
- Redundant variation

It depends on how the classifier will be used, and how similar the target set is to the training set.

The challenge of non-stationarity

In most realistic applications, the new samples to be classified will come from a different “batch” than the original samples used to build the classifier.

What, if anything, can we do to guard against or deal with the possibility that new sources of unwanted variation will affect the new samples?

More comments later.

An interesting point

The fact that the choice of negative controls depends on the purpose of the classifier is obviously important for applied work. But it is also interesting on a conceptual level.

We see that the negative controls may be used not just to identify unwanted variation, but, in some sense, to define it.

Removing unwanted variation from the test (target) set

Model for the training set data

$$Y_{mxn} = X_{mxp}\beta_{pxn} + W_{mxk}\alpha_{kxn} + \varepsilon_{mxn}$$

Assumptions as before; here *X* is known

Model for the test and target set

$$\tilde{Y}_{\tilde{m}\mathrm{x}n} = \tilde{X}_{\tilde{m}\mathrm{x}p}\beta_{p\mathrm{x}n} + \tilde{W}_{\tilde{m}\mathrm{x}k}\alpha_{k\mathrm{x}n} + \tilde{\varepsilon}_{\tilde{m}\mathrm{x}n}$$

Analogous assumptions; here $\tilde{X}$ is unobserved.

The shared *α* and *β* (and *p*, *k* and *n*) constitute the weak stationarity assumption. Note that *m* and $\tilde{m}$ will in general differ. We assume *ε* and $\tilde{\varepsilon}$ are independent.

Goal

To estimate $W\alpha$ and $\tilde{W}\alpha$, and subtract off the estimate from Y and $\tilde{Y}$ respectively, to produce matrices P and $\tilde{P}$ (predictors) for the classification.

P should be ≈ $X\beta$ and $\tilde{P}$ should be ≈ $\tilde{X}\beta$.

How to proceed?

We know* how to remove the unwanted variation from Y when X is known: we can use RUV-2, RUV-4 or RUV-inv to estimate W and α, and subtract $\hat{W}\hat{\alpha}$.

How can we estimate and subtract $\tilde{W}\alpha$ when $\tilde{X}$ is not known?

We will describe two ways.

* We *think* we know. There is a catch!

Method A, start with $\hat{\alpha}$

Since $\tilde{Y}_c = \tilde{W}\alpha_c + \tilde{\varepsilon}_c$

we can estimate $\tilde{W}$ by regressing $\tilde{Y}_c'$ on $\hat{\alpha}'_c$, and so

$$\tilde{W} \approx \tilde{Y}_c\hat{\alpha}'_c(\hat{\alpha}_c\hat{\alpha}'_c)^{-1}.$$

This leads to

$$\tilde{P}^{(A)} \approx \tilde{Y} - \tilde{Y}_c\hat{\alpha}'_c(\hat{\alpha}_c\hat{\alpha}'_c)^{-1}\hat{\alpha}.$$

Digression: some calculations

Let $UDV_c{}'$ be the SVD of Y_c.

Note that $U \approx W$, and is an RUV-2 estimator of W, and that $DV_c{}' \approx \alpha_c$, though it is not the RUV-2 estimator.

Now $U = Y_c V_c D^{-1}$, and so analogously, we define

$$\tilde{U} = \tilde{Y}_c V_c D^{-1}.$$

Then we find that $\tilde{U} \approx \tilde{W}$.

Method B, start with $\hat{\beta}$

$$\text{Define } \hat{\alpha} = (U'U)^{-1}U'(Y - X\hat{\beta}),$$
$$\text{and write } \tilde{P}^{(B)} = \tilde{Y} - \tilde{U}\hat{\alpha}.$$

We can show that this expression is quite insensitive to violation of the negative control gene assumption, and so we can use **all** genes, giving

$$\tilde{P}^{(B)} \approx \tilde{Y} - \tilde{Y}Y'(YY')^{-1}(Y - X\hat{\beta}).$$

Comparing and contrasting Methods A and Method B

Method A starts with $\hat{\alpha}$ based on Y_c, and leads to $P^{(A)}$. Method B starts with $\hat{\beta}$, which might, but need not be based on Y, and uses the $\tilde{U} \approx \tilde{W}$ based on $\tilde{Y}$ to get $P^{(B)}$.

If we get both $\hat{\alpha}$ and $\hat{\beta}$ from RUV-inv, *with the same controls*, then we find that $P^{(A)} = P^{(B)}$.

But if things are done differently, they will diverge.

Advantage of $P^{(B)}$

If we don’t need to worry about control genes, we can use all genes. (We may still need control genes to get $\hat{\beta}$ in the first place.) If we do take all genes, we find that

$$\tilde{P} \approx \tilde{Y} - \tilde{Y}Y'(YY')^{-1}(Y - X\hat{\beta}).$$

Using all genes gives us a richer estimate of *W*. If there are unwanted (e.g. biological) factors that affect a subset of genes, but not the negative control genes, these will not be adjusted for if we limit ourselves to control genes. But by using all genes as above, we can adjust for these factors.

Cleaning up the training set

$P^{(B)}$ is our test/target set prediction data.
What is the analogus for the training set data?

Do the same thing with our training data. We find that

$$P \approx Y - YY'(YY')^{-1}(Y - X\hat{\beta}).$$

simplifies to $X\hat{\beta}$! Way too optimistic to be a realistic training set. We need another way.

Cross Normalization

$$P_i = Y_i - Y_i Y'_{-i}(Y_{-i}Y'_{-i})^{-1}(Y_{-i} - X_{-i}\hat{\beta}^{(-i)}),$$

where

- Y_i = the ith row of Y
- $Y_{-i} = Y$ with the ith row removed
- $X_{-i} = X$ with the ith row removed
- $\hat{\beta}^{(-i)}$ = the estimate of β using X_{-i} and Y_{-i}

Now the P_i, $i=1,\ldots m$, are “not too clean”.

How does it work? Simulations

Simulations have been carried out to compare Methods A and B, using “good”, “bad” and “too good” controls, X and W uncorrelated or correlated, β and α uncorrelated or correlated, stationarity or not (column of W_b not in W_a), using RUV-2, RUV-4 (with varying k) and RUV-inv for estimating β and α, and housekeeping (HK) or all genes as controls.

Overall, Methods A and B using RUV-inv are pretty similar, and best, but when the going gets tough, B wins. Not surprisingly, full outperforms HK in sims.

Choice of classifier

Removing Unwanted Variation makes it possible to use "simple" classifiers

Here "simple" includes linear discriminant analysis (LDA) and diagonal linear discriminant analysis (ΔLDA).

Below we compare them to support vector machines (SVM) and the elastic net logistic regression package, `glmnet` (not so simple classifiers).

We also use these classifiers with the only other method which we know deals with unwanted variation: fSVA (Leek *et al*, 2012).

Why we might be able to stick to “simple” classifiers?

Suppose that β and α are fixed, and that X, W, ε and their ~ counterparts are all random, and mutually independent. Assume that W and $\tilde{W}$ are iid $N(0,\Lambda)$, and that X and $\tilde{X}$ are single column matrices with entries -1 or $+1$ with probability π. Define Σ to be the $n \times n$ diagonal matrix whose diagonal entries are the variances σ_j^2. For this illustration, we assume strong stationarity: that the pairs (X_i, W_i) and $(\tilde{X}_i . \tilde{W}_i)$ are iid. Then we find that

$Y_i \,|\, \{X_i=-1\} \sim N(-\beta,\alpha'\Lambda\alpha+\Sigma)$ and $Y_i \,|\, \{X_i=+1\} \sim N(\beta,\alpha'\Lambda\alpha+\Sigma)$,

and $P_i \,|\, \{X_i=-1\} \approx \sim N(-\beta,\Sigma)$, and $P_i \,|\, \{X_i=+1\} \approx \sim N(\beta,\Sigma)$,

and the same for $\tilde{Y}$. Now Σ is diagonal. Discuss!

Example

Gender differences in the brain

(Vawter *et al*, **Neuropsychopharmacology** 2004)

- 5 men, 5 women
- 3 brain regions (AnCing, DLPFC, Cb)
- Each sample done in 3 labs
- 2 Affymetrix chip types: HGU95a, HGU95av2
- There should be (5+5) × 3 × 3 = 90 arrays, but 6 are missing, so there are just 84.

We'll focus on gender, i.e. sex.

Ex: gender differences in the brain

(Vawter *et al*, **Neuropsychopharmacology** 2004)

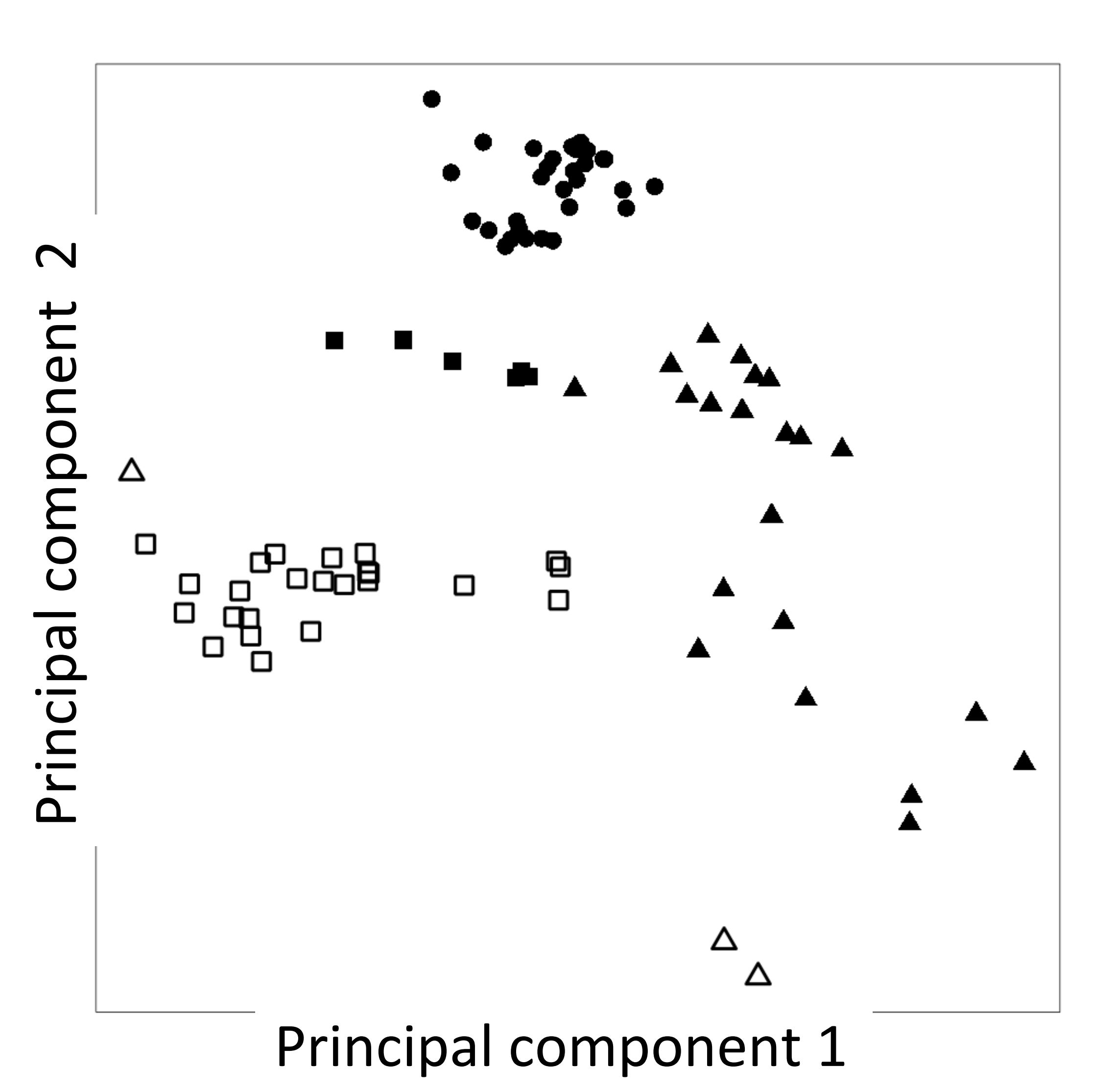
- 5 men, 5 women
- 3 brain regions (AnCing, DLPFC, Cb)
- Each sample done in 3 labs
- 2 Affymetrix chip types: HGU95a, HGU95av2
- There should be (5+5) × 3 × 3 = 90 arrays, but 6 are missing, so there are just 84.

We'll focus on gender, i.e. sex.

There is lots of UV!

(5♀ + 5♂) x 3 regions x 3 labs on chips v1, v2

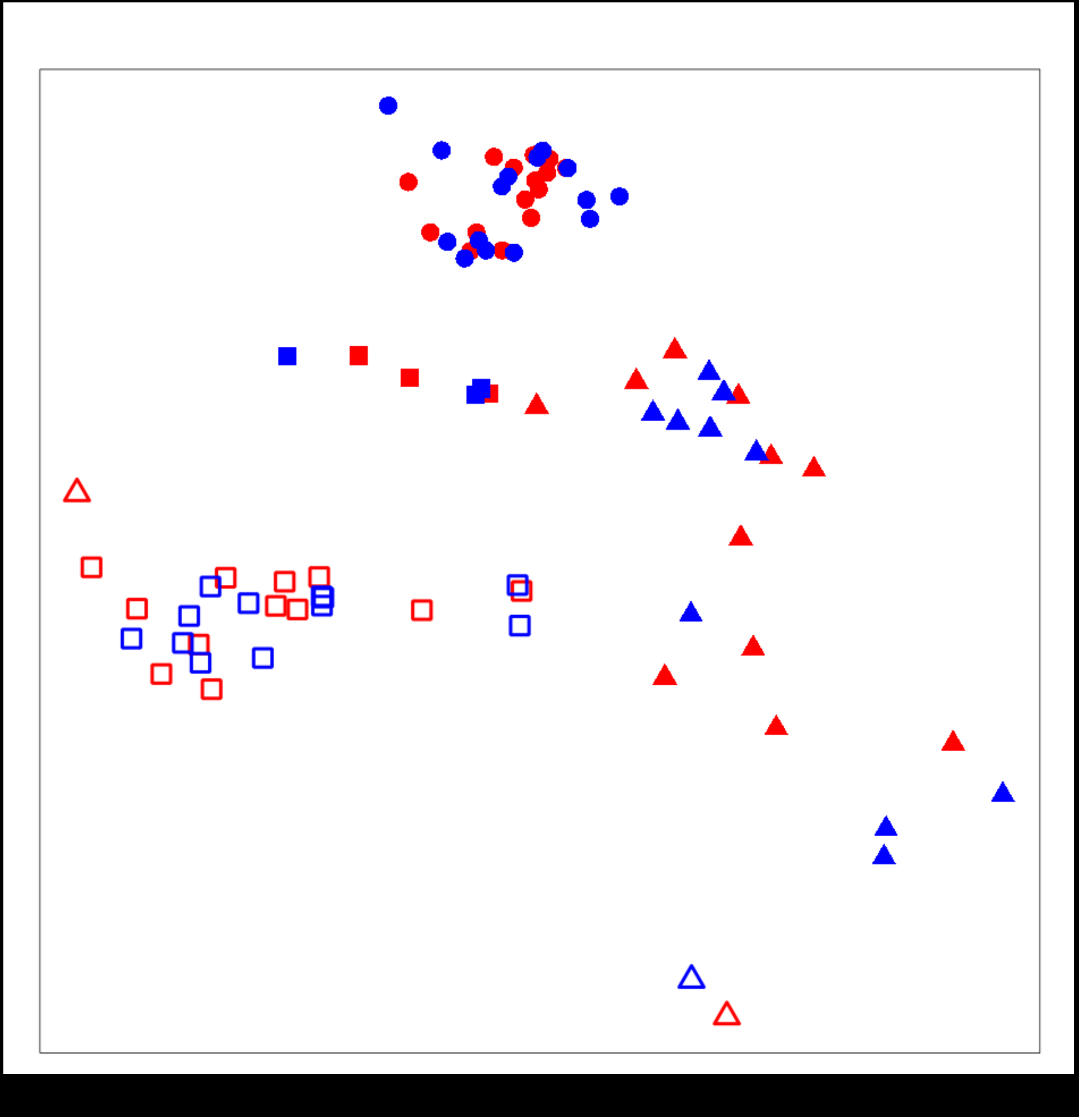
84 Affy chips. PCs based on all genes. No preprocessing



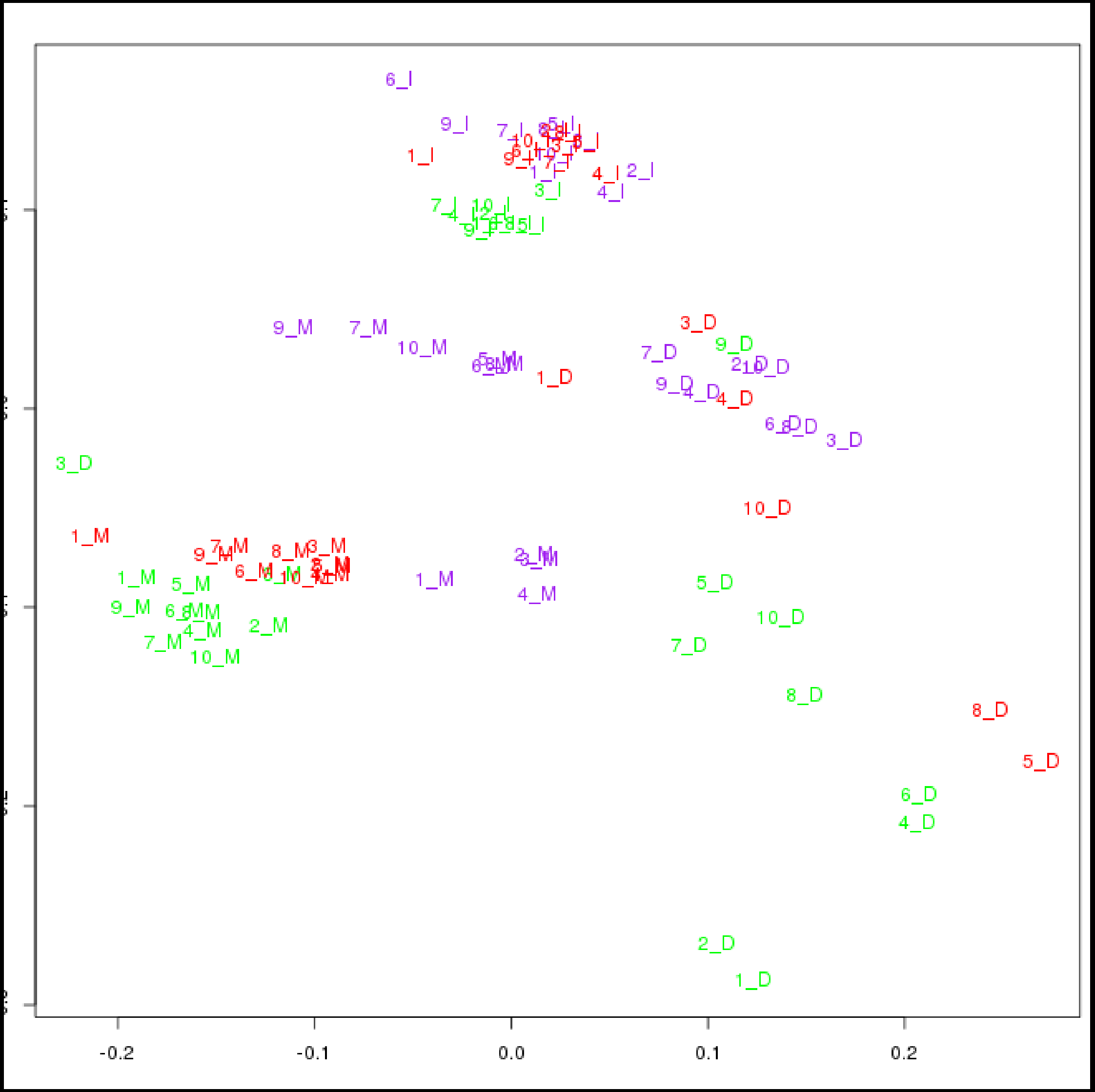
Shape = lab

Closed/open = chip v1/v2

Same plot with gender indicated



Same plot with brain regions indicated



Ex: gender differences in the brain, 2

- 12,685 probe sets
- 799 housekeeping (HK) genes, 33 spike-in negative controls

We remove genes on the Y chromosome, XIST and DDX3X, for otherwise, predicting gender is too easy.

Training set 60. Validation set 24. Results are averages over 100 random training/validation splits. In each case*, the classifiers are based on the top 10 ranked differentially expressed genes in the training set.

Estimated accuracy rates

Avgs of 100 random	SVM	LDA	ΔLDA	Glmnet*
Unadjusted	.57	.58	.57	.71
fSVA (Leek *et al*, 2012)	.64	.64	.64	.72
RUV-inv only	.67	.68	.64	-
RUVBinv (HK)	.87	.85	.88	.83
RUVBinv (full)	.85	.83	.85	.84

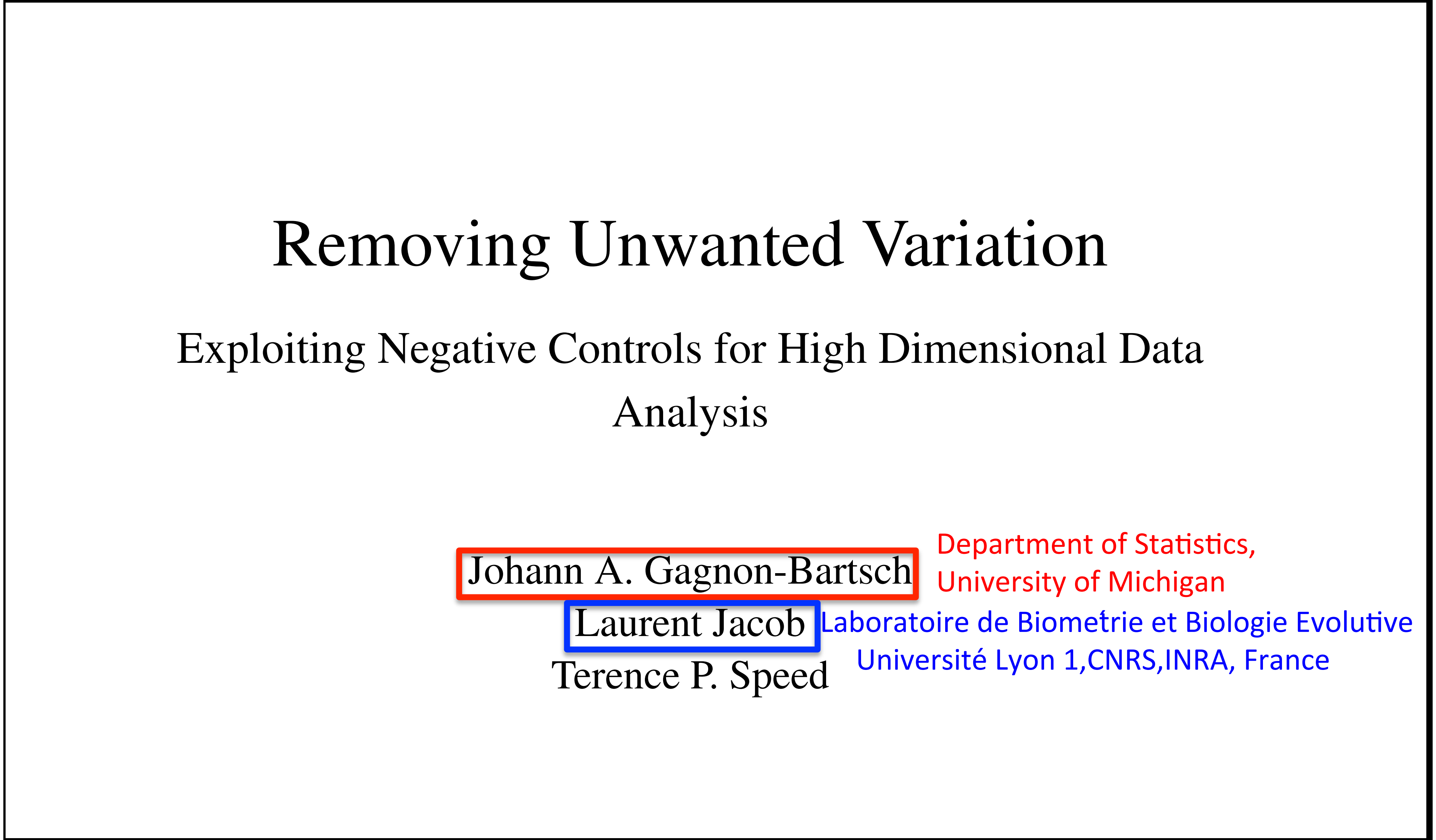
*α = 1, own variable selection

ΔLDA = Δiagonal LDA

Making this work for personalized medicine

Here are a couple of thoughts

- Veracyte's Afirma-T removes unwanted variation by normalizing a set of reference genes to a fixed distribution, a common strategy. This aspect of their algorithm, along with all others, is *locked down* as a Food & Drug Administration requirement.
- RUV-B begins with an estimate of *β,* and takes it from there. If this is locked down, then the whole process can be locked down.
- If the "truth" associated with some target samples becomes known, the estimate of *β* can be improved, but this would violate the locking.



Removing Unwanted Variation

Exploiting Negative Controls for High Dimensional Data Analysis

Johann A. Gagnon-Bartsch
Department of Statistics,
University of Michigan

Laurent Jacob
Laboratoire de Biometrie et Biologie Evolutive
Université Lyon 1,CNRS,INRA, France

Terence P. Speed

10/12 written book, to be completed in the next few months, CUP-IMS monograph
This lecture was part of chapter 11.