Opinionated Lessons in Statistics

by Bill Press

#49 Eigenthingies and Main Effects

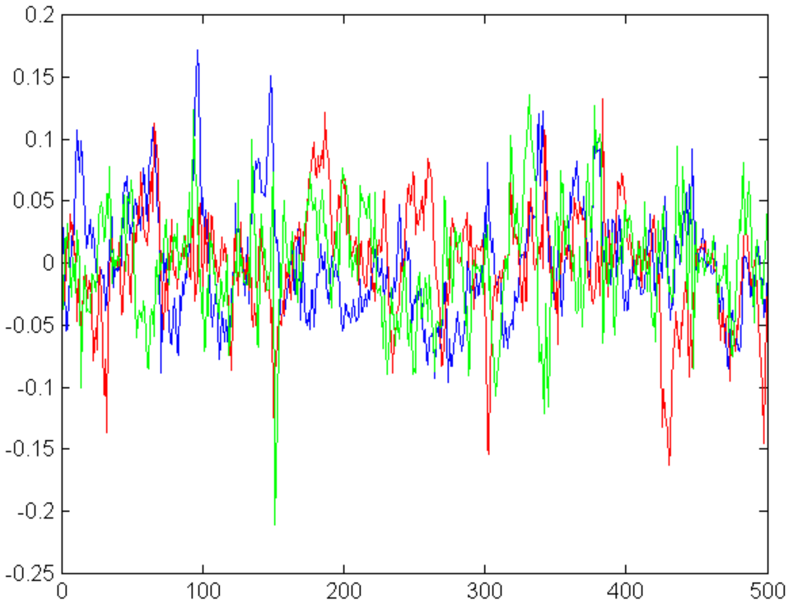
Eigengenes and Eigenarrays

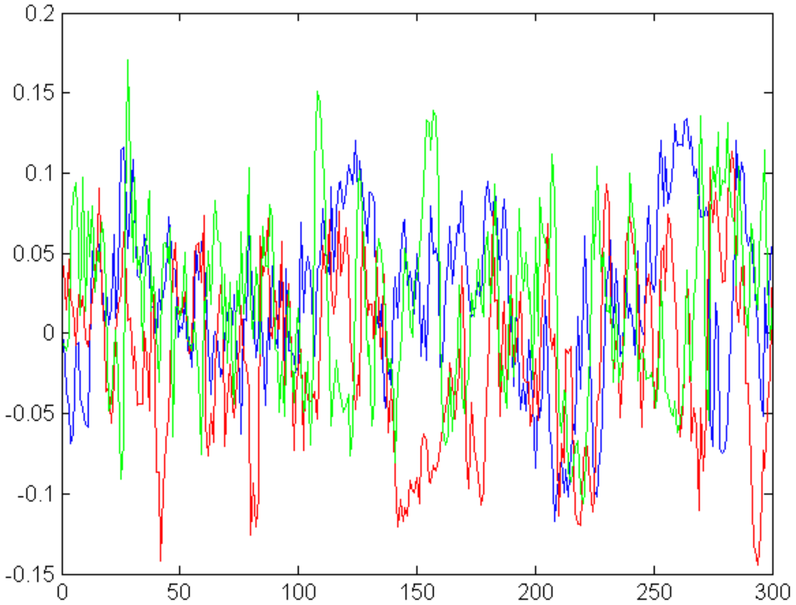
$$\mathbf{X} = \sum_{i=1}^{M} s_i\, \mathbf{U}_{\cdot i} \otimes \mathbf{V}_{\cdot i}$$

Thus far, we haven’t actually “looked at” the largest-SV orthogonal basis vectors, namely the first few columns of $\mathbf{U}$ and $\mathbf{V}$

```
plot(U(:,1),'b')
hold on
plot(U(:,2),'r')
plot(U(:,3),'g')
hold off
```

```
plot(V(:,1),'b')
hold on
plot(V(:,2),'r')
plot(V(:,3),'g')
hold off
```





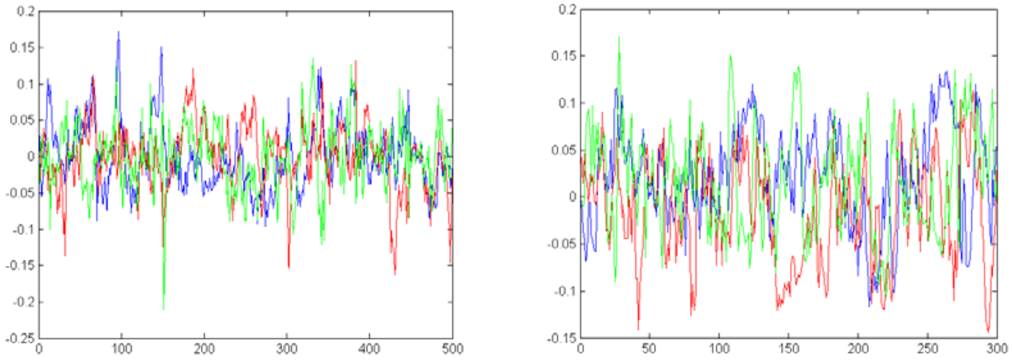
These are “eigengenes”, the linear combination of genes that explain the most data.

times s_i ↔

These are “eigenarrays”, the linear combination of experiments that explain the most data.

However, except in special cases, eigengenes and eigenarrays are not easily interpreted.

Since we can permute the order of experiments and/or genes in the data, the “shape” of the eigenfunctions has no particular meaning here.

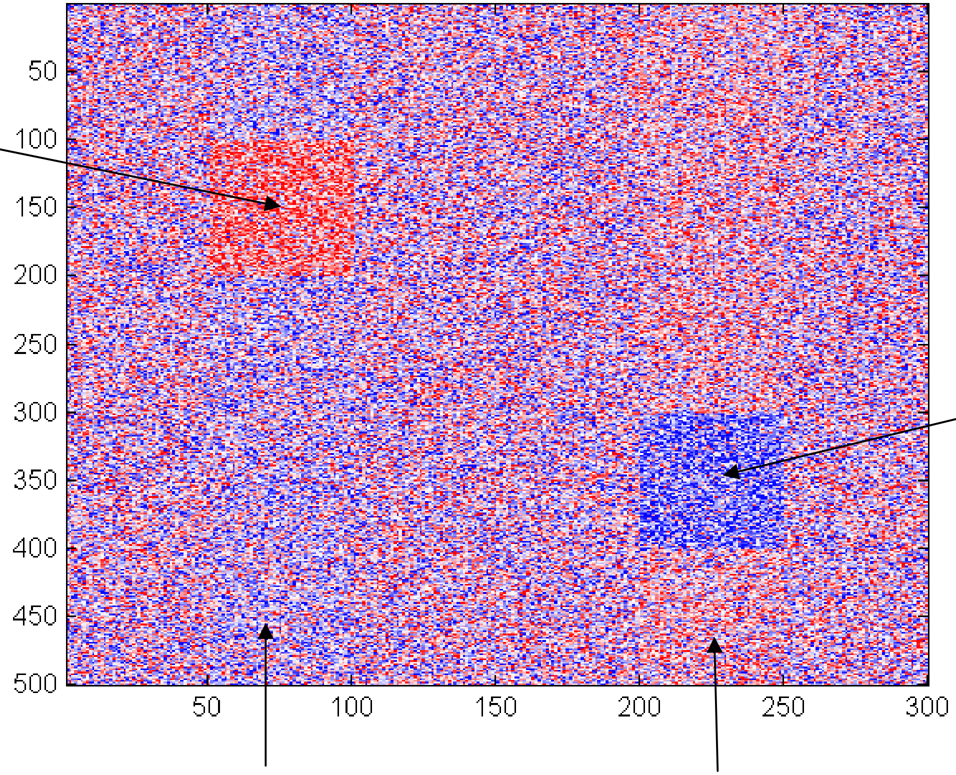


Also, as discussed in the previous segment, main effects generally don’t correspond 1-to-1 to eigenanythings. At best, ~K main effects are in ~K eigenthingies.

Let’s construct a toy gene expression example with 2 main effects, and see how they show up in eigengenes and eigenarrays.

Make a toy example with (what we would call) 2 main effects:

```
pdata = randn(500,300);
pdata(101:200,51:100) = pdata(101:200,51:100) + 1;
pdata(301:400,201:250) = pdata(301:400,201:250) - 1;
pmean = mean(pdata,1);
pstd = std(pdata,1);
pdata = (pdata - repmat(pmean,[size(pdata,1),1]))./repmat(pstd,[size(pdata,1),1]);
colormap(genecolormap)
image(20*pdata+32)
```



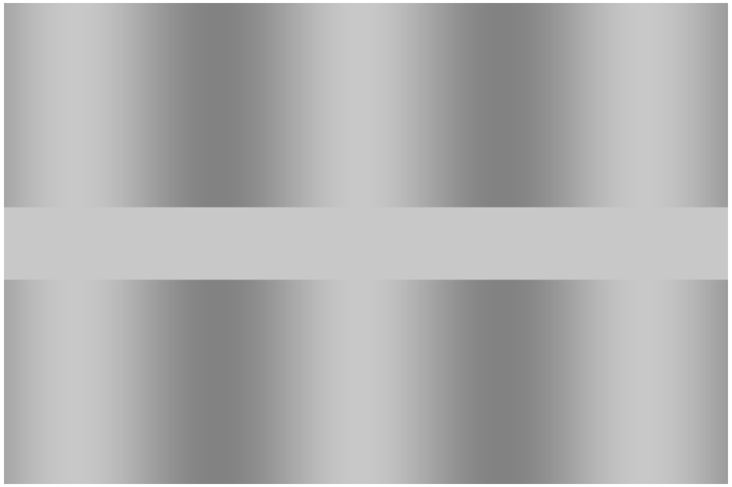
these genes overexpress in these conditions

these genes underexpress in these conditions

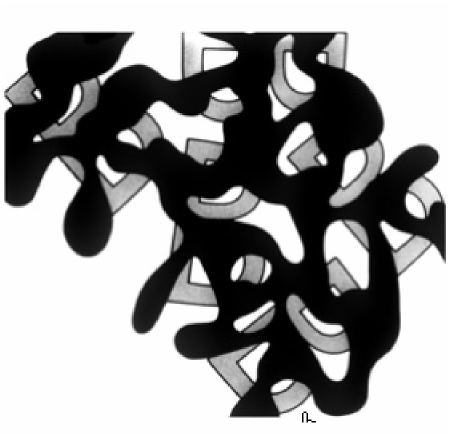
note slight vertical banding from subtracting the means

Did you see the “visual completion” or “visual phantom” illusions in the previous slide?

How about these?



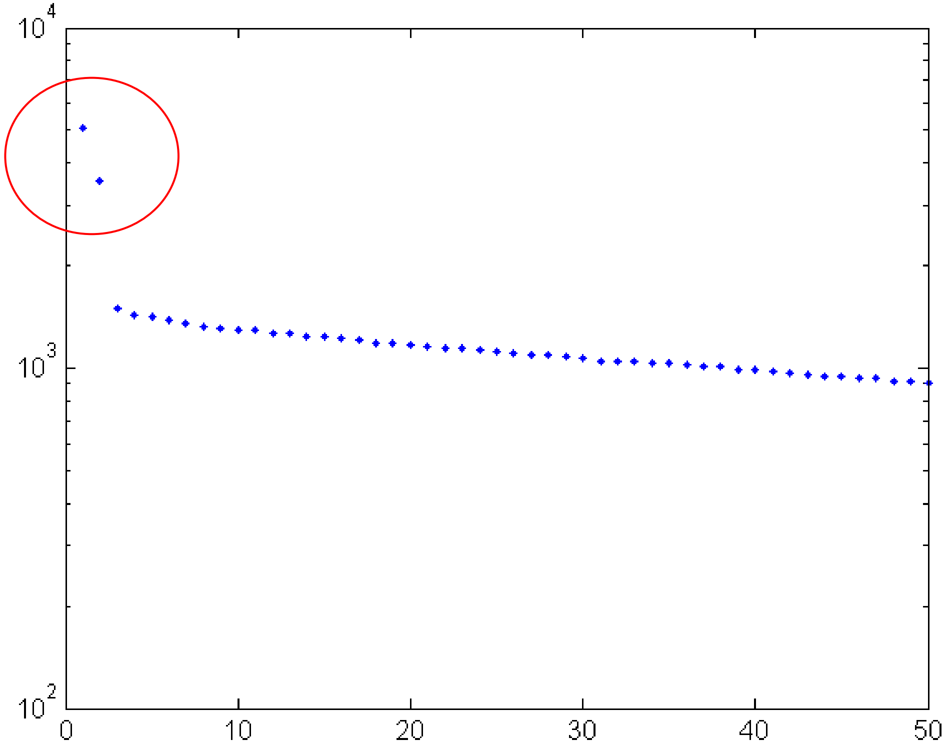
Akiyoshi Kitaoka (Ritsumeikan University)



Bregman AS (1981)

As (naively?) expected, there are exactly two large principal components (or SVs)

```
[Up Sp Vp] = svd(pdata,0);
spsq = diag(Sp).^2;
semilogy(spsq(1:50),'.b')
```



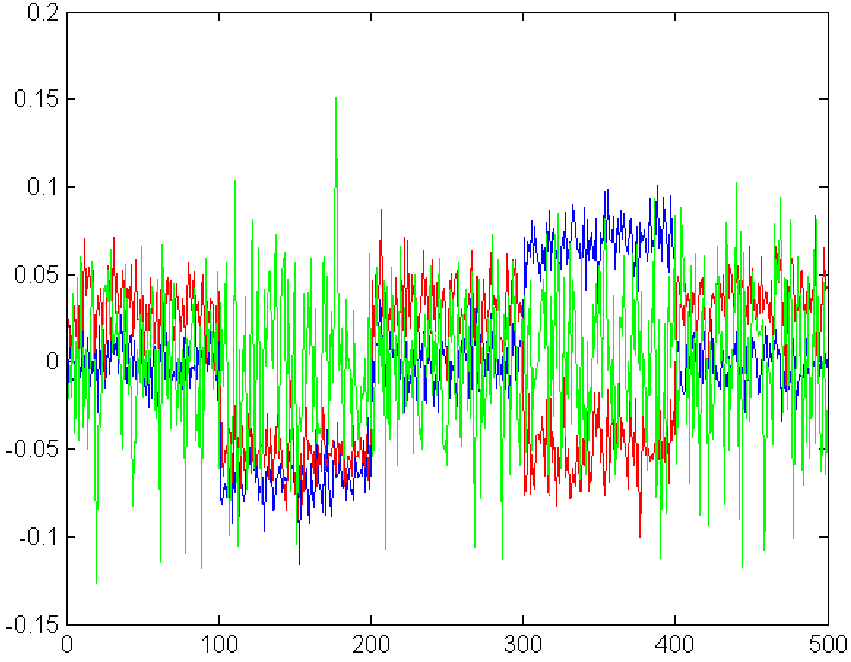
continues on to 300

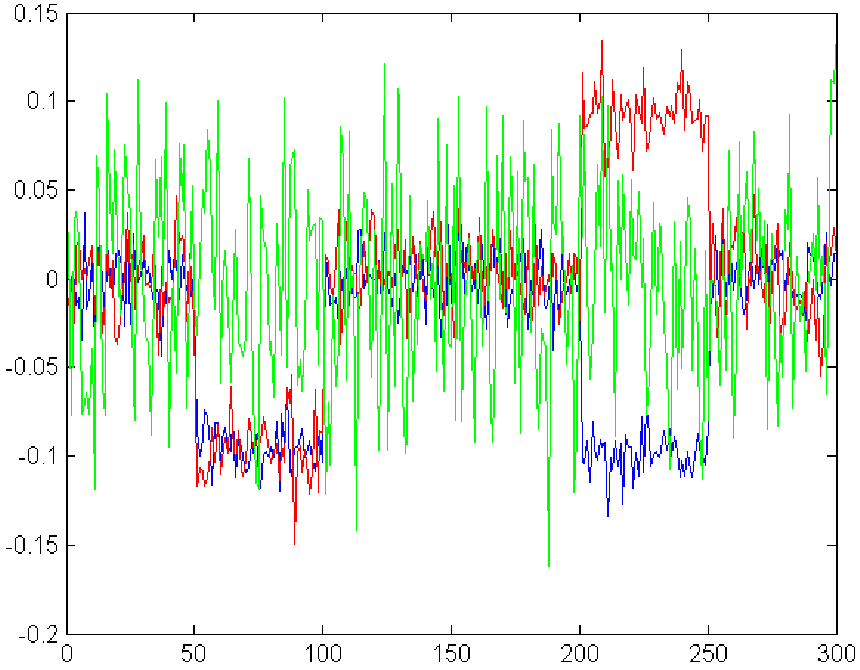
So should we expect the eigengenes/eigenarrays to show the separate main effects?

Plot 1st three eigengenes and eigenarrays

```
plot(Up(:,1),'b')
hold on
plot(Up(:,2),'r')
plot(Up(:,3),'g')
```

```
plot(Vp(:,1),'b')
hold on
plot(Vp(:,2),'r')
plot(Vp(:,3),'g')
```





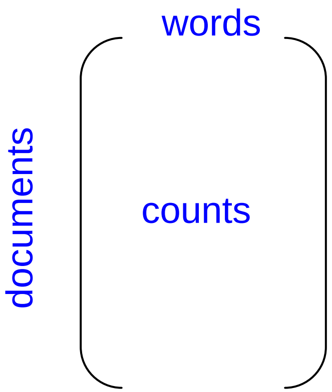
First two contain an (orthogonalized) mixture of the two main effects.
Third one is, as we expect, “random”.

If we had 20 main effects, the first 20 eigengenes/arrays would be mixtures of them.

Mention in passing:

There exist methods of Non-negative Matrix Factorization (NMF) whose purpose is to stop main effects from mixing when positivity matters.

Example: cluster text documents by word counts



= $\mathbf{X}$

vector of 0 or pos weights of documents in *i* th cluster

vector of 0 or pos weights of words in *i* th cluster

$$\mathbf{X} \approx \sum_{i=1}^{M} \mathbf{F}_{\cdot i} \otimes \mathbf{G}_{\cdot i}$$

i.e. $\mathbf{X} \approx \mathbf{F}\mathbf{G}^T$ positive matrixes

For our problem, since genes can also be under-expressed, we would need

diagonal matrix of ±1

$$\mathbf{X} \approx \mathbf{F}\mathbf{S}\mathbf{G}^T$$

The problem with these methods is

1. The factorizations are (far from) unique!
2. The computational algorithms are little better than brute-force minimization of

$$\left\|\mathbf{X} - \mathbf{F}\mathbf{G}^T\right\|$$

How to find the global minimum??

3. There are other good clustering algorithms (GMMs, hierarchical, etc.)