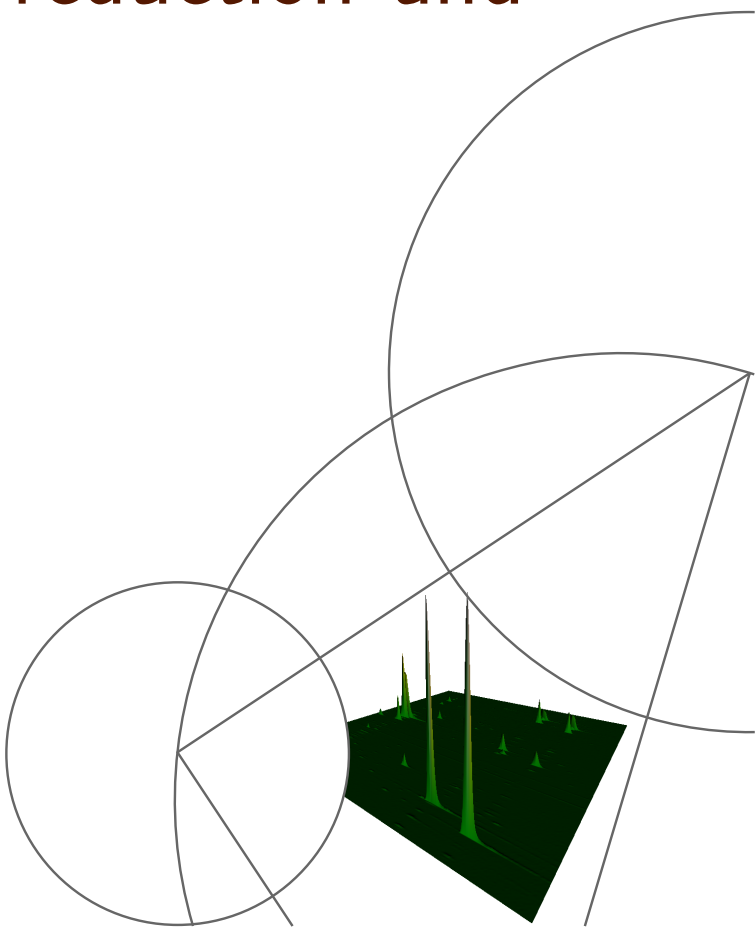
Faculty of Life Sciences

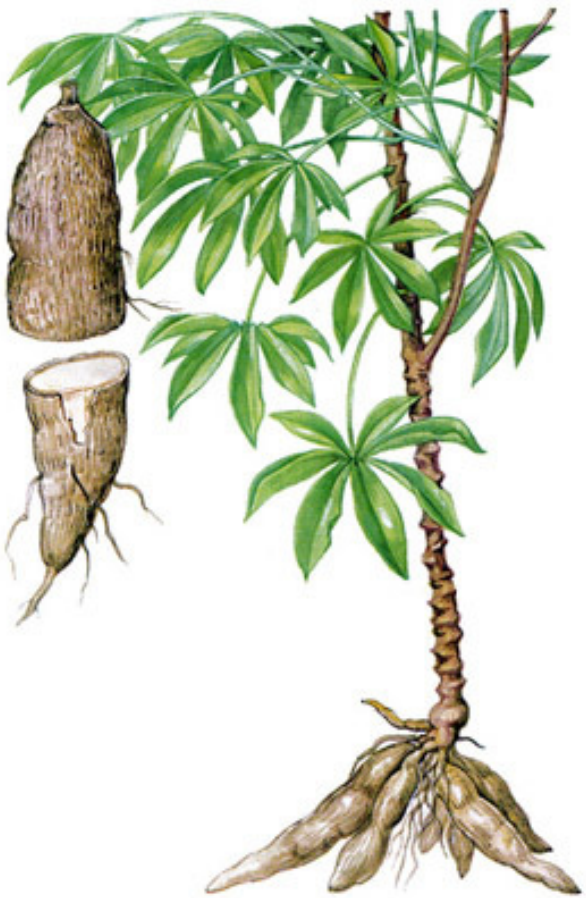
Getting a little from a lot, reduction and modeling of -omics data

Kasper Brink
IGM



Intro

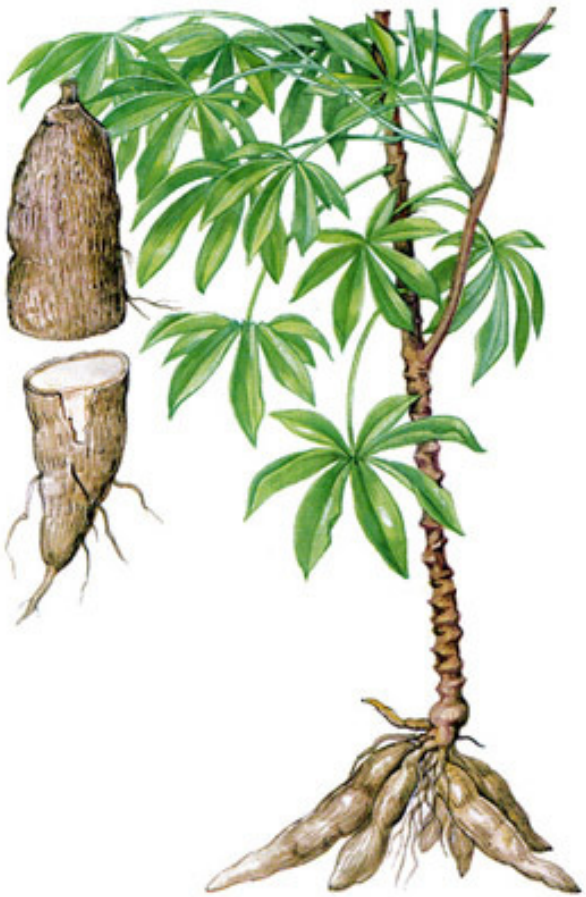
Cassava (*Manihot esculenta*)



- Important source of nutrition in Africa, Asia and South America
- Contains Cyanogenic Glucosides

Intro

Cassava (*Manihot esculenta*)



- Important source of nutrition in Africa, Asia and South America
- Contains Cyanogenic Glucosides

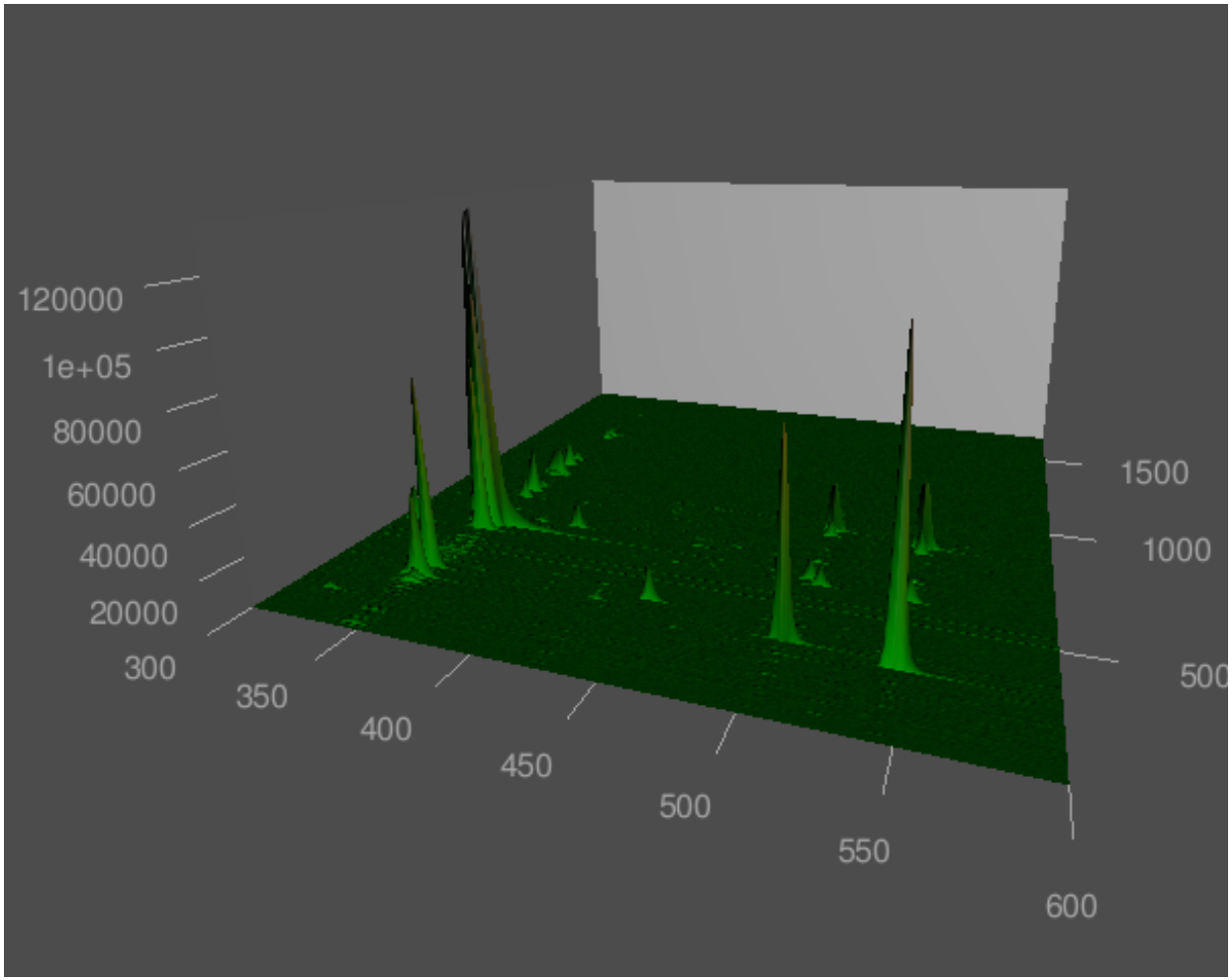
Intro

- 32 plants
- LC-MS spectra from the 32 plants
- Arrays for 13865 genes from the same plants
- Construct a model on the form
  $Spectra = expression \times coefficients + noise$
- Problems: dimension reduction, variable selection, validation

Numbers

1. 109431241

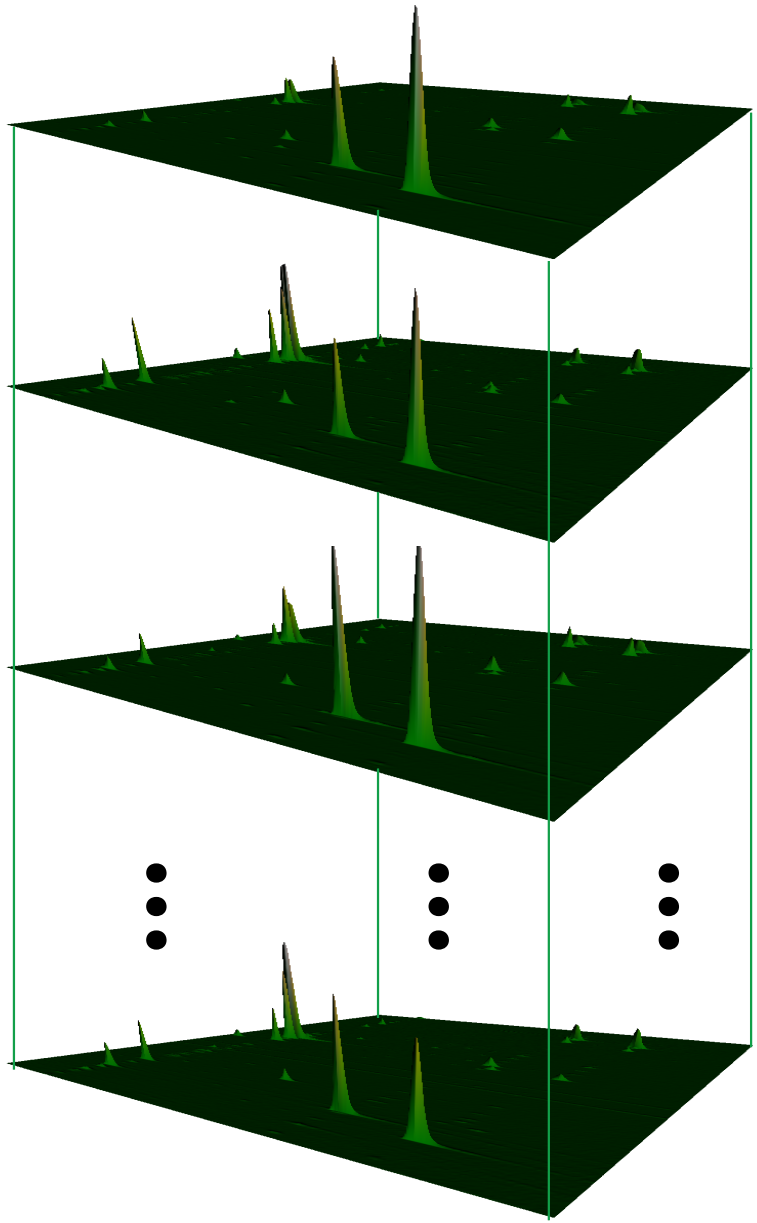
Data types: LC-MS



Data types: LC-MS

	time	mz	intensity
[1,]	5.205	124.7990	20
[2,]	5.205	125.1057	17
[3,]	5.205	125.5501	12
[4,]	5.205	126.6448	27
[5,]	5.205	127.3360	13
[6,]	5.205	127.7387	22
[3682288,]	1802.489	1202.724	398
[3682289,]	1802.489	1203.069	309
[3682290,]	1802.489	1203.419	178
[3682291,]	1802.489	1204.025	70
[3682292,]	1802.489	1204.426	48
[3682293,]	1802.489	1204.741	80

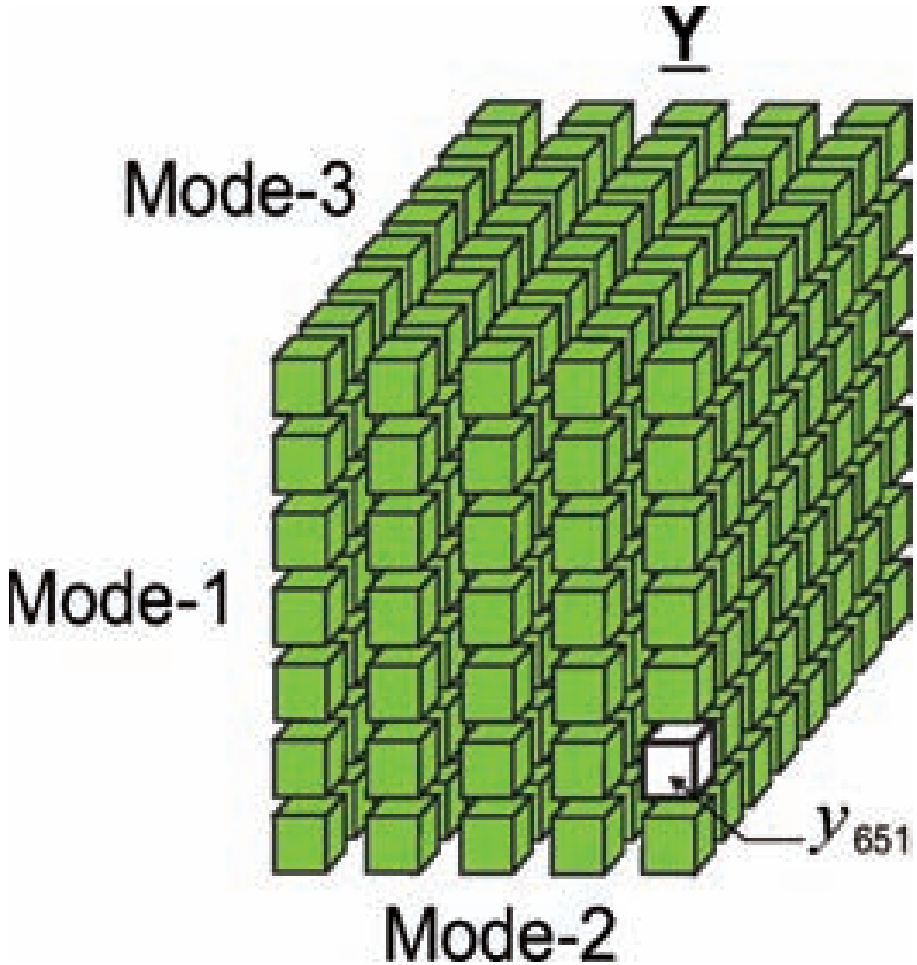
Data types: LC-MS



Numbers

1. 109431241
2. 17463549

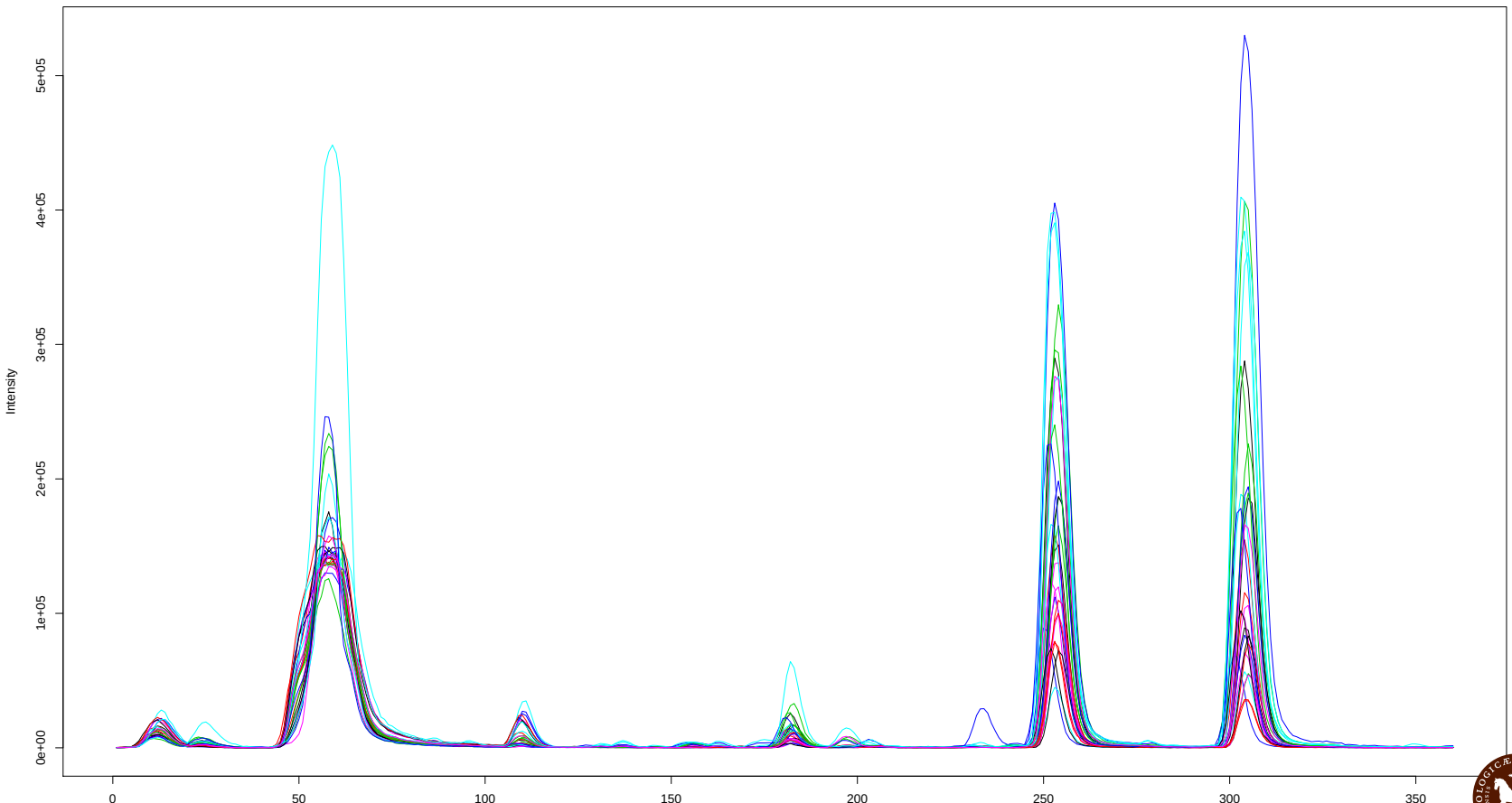
Data types: LC-MS



Extract signals from 3D tensor

Data types: LC-MS

The easy reduction: Average in the required direction

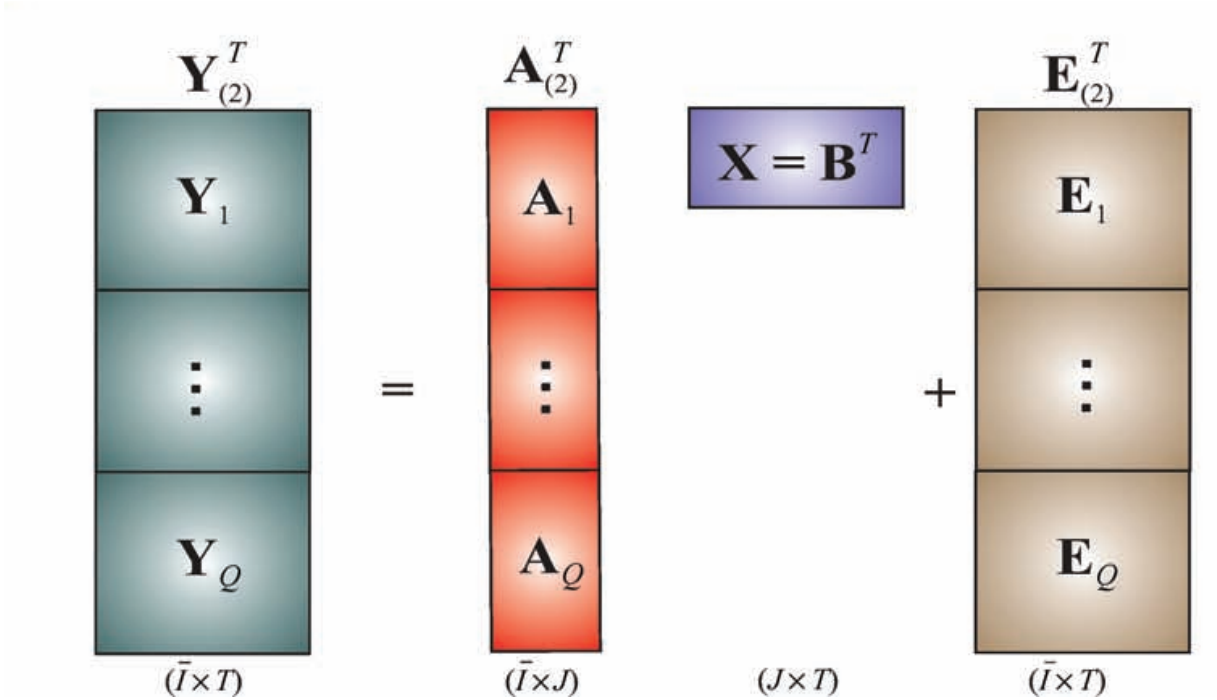


Numbers

1. 109431241
2. 17463549
3. 11872

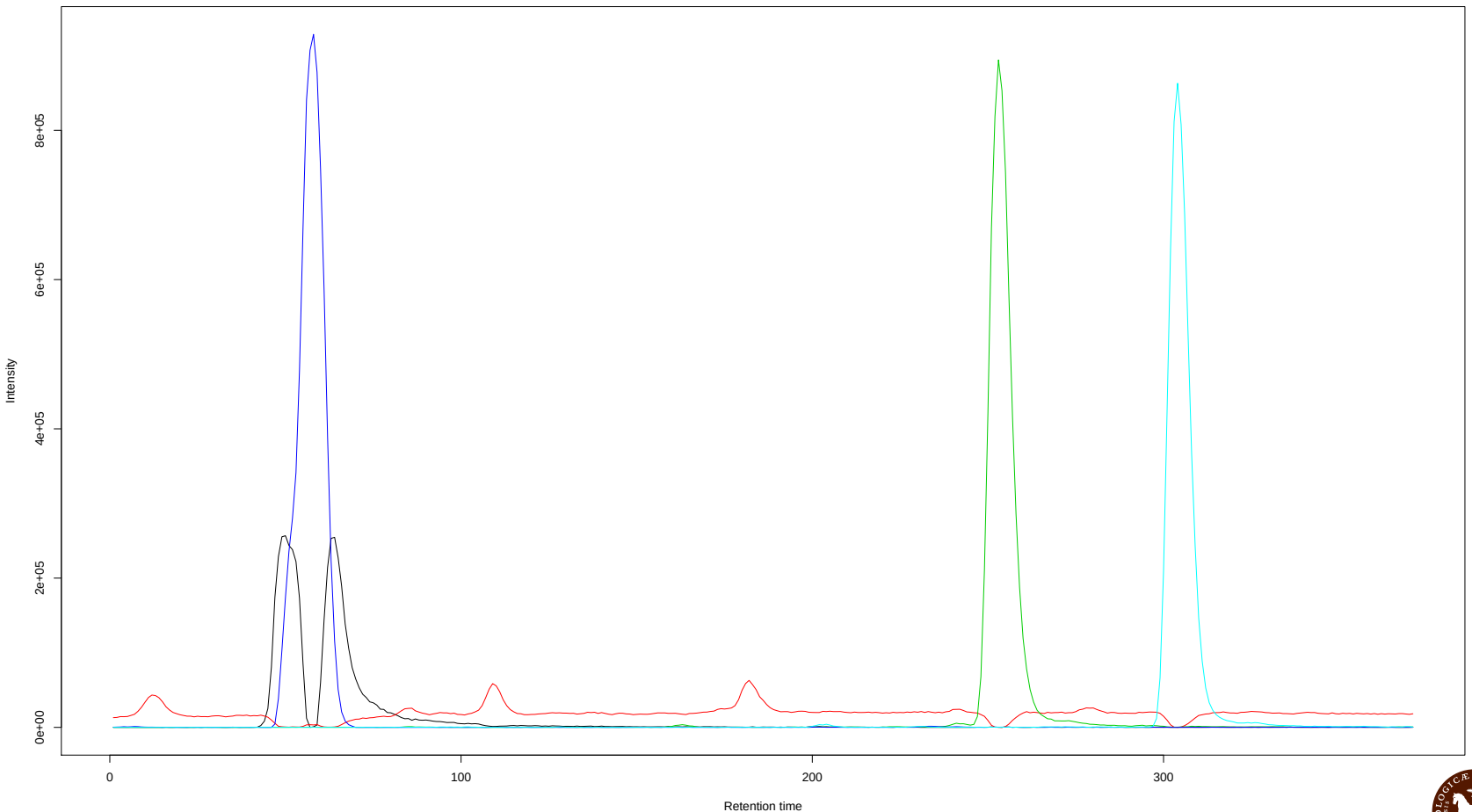
Data types: LC-MS

The more meaningful reduction: Decomposition into mixing matrices and underlying components with some restraints (non-negativity, sparsity)



This requires that the individual spectra are exactly the same dimension to construct the tensor $\underline{\mathbf{Y}}$

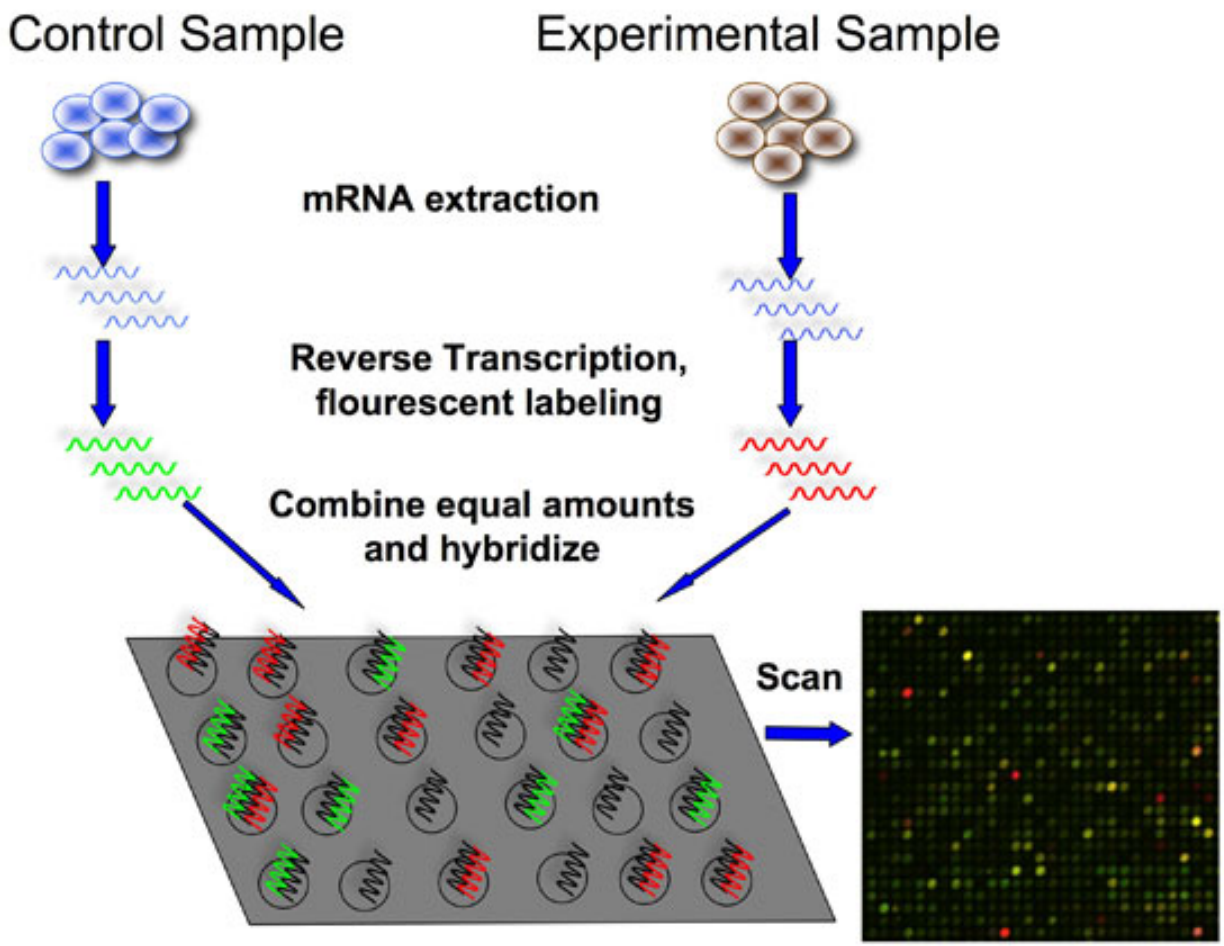
Data types: LC-MS



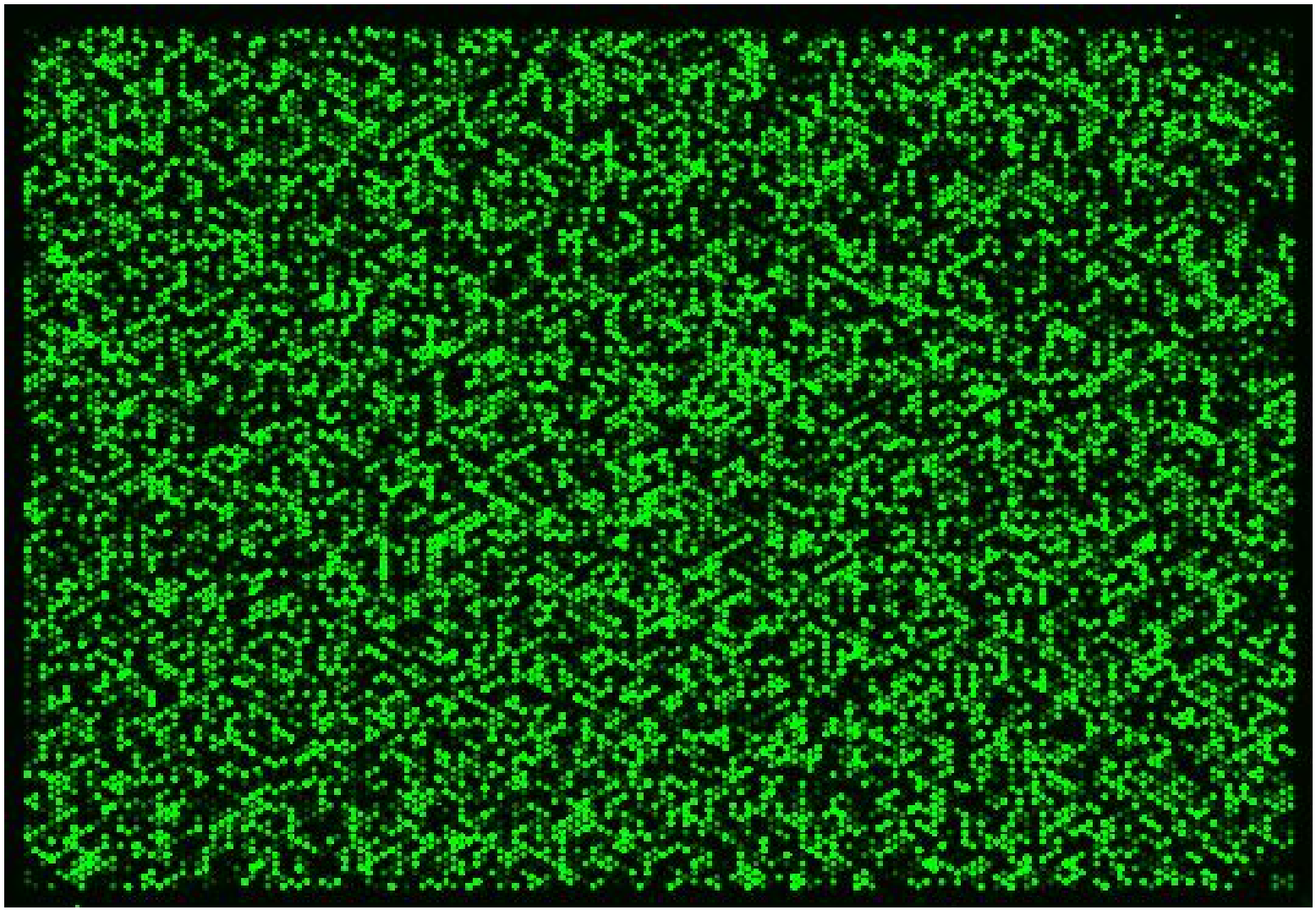
Numbers

1. 109431241
2. 17463549
3. 11872
4. 1855

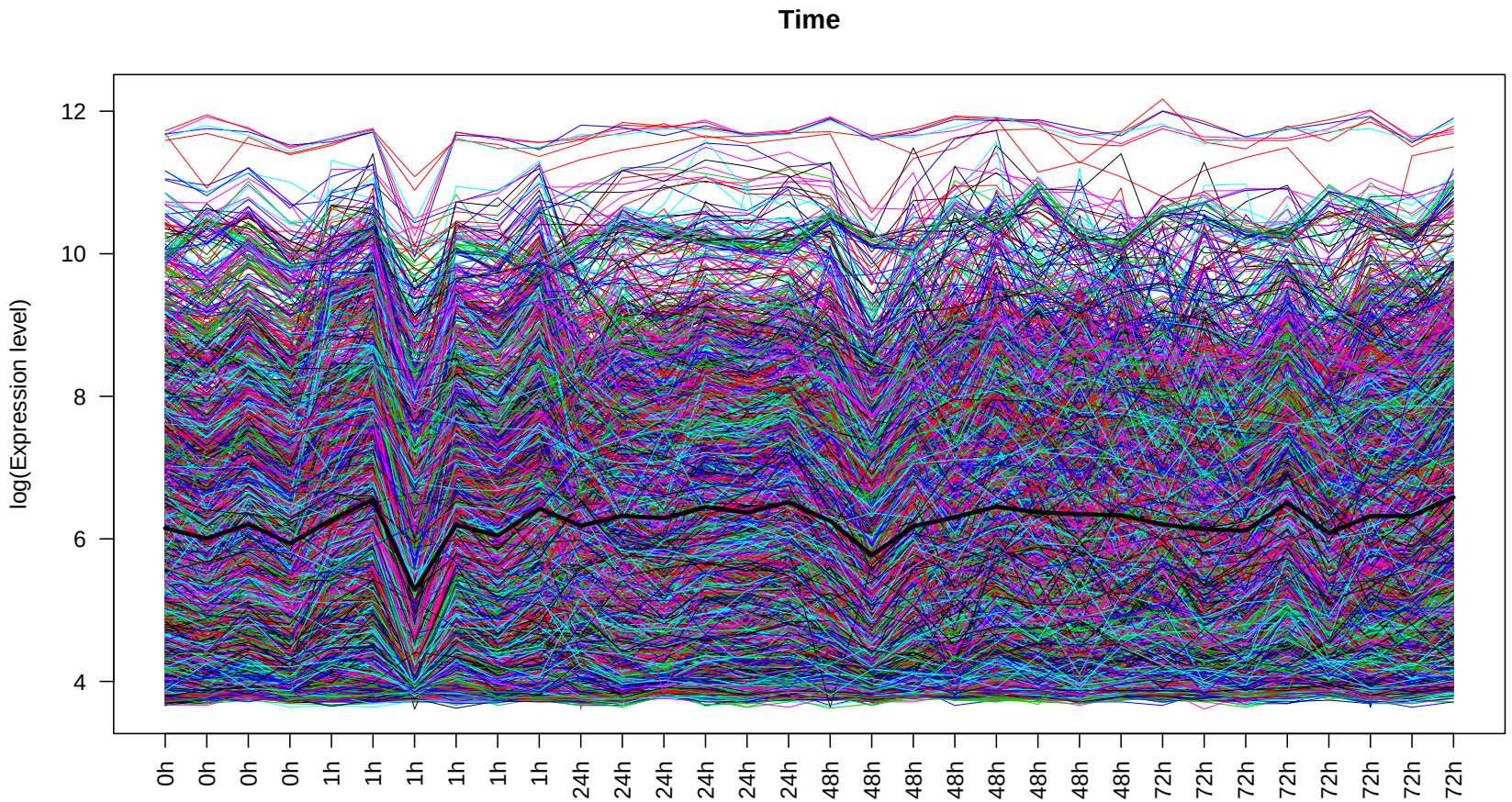
Data types: Array



Data types: Array



Data types: Array



Data types: Array

```
          151_1     151_2     151_3      151_4      152_1
[1,]   243.5156  170.1290  137.9091   131.3433   138.5385
[2,]  1813.1180 2502.2420 1337.7850  1903.2860  3155.5690
[3,]  6471.4770 1927.5710 1923.8660  1023.3690  1088.7270
[4,] 12794.1900 5849.8330 8092.0160 12470.7900 17349.1100
[5,]   875.2462  649.8030  516.7879   451.6719   694.3333
[6,]   324.4375  139.4516  157.3231   121.8254   260.1695
[13860,]  257.98460  265.31250  299.25000  399.62120  149.66100
[13861,]   96.68852  257.23440  291.28130  178.21540  108.80650
[13862,]  596.07690  264.51560  143.07940   66.92424   52.63768
[13863,]   56.67742   71.33871   60.19672   88.40984   52.62903
[13864,] 2112.12500 1995.82100 1702.69400 2240.42200 1716.50000
[13865,] 1259.39400 3086.60700 1551.72900  881.36360 2081.33300
```

Numbers

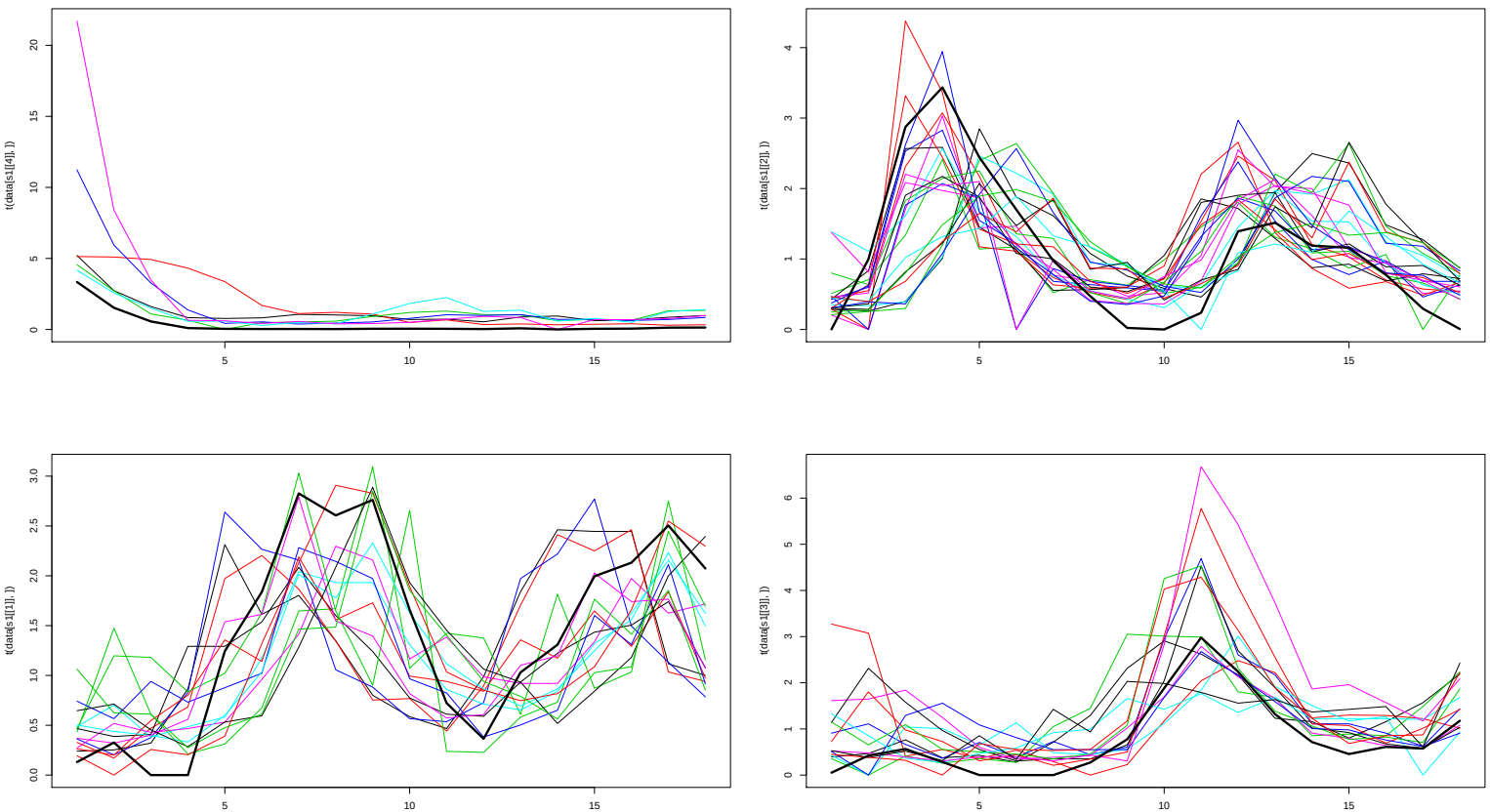
1. 109431241
2. 17463549
3. 11872
4. 1855
5. 11872+443680

Data types: Array

How to choose relevant genes? The easy way: Fit linear model with multiple testing correction (`limma`) and select the genes with the lowest p-values.
Very fast and simple, but is it meaningful?

Data types: Array

A more interesting approach, clustering by profile. Use NMF algorithm



Modelling

At some point data will look like this

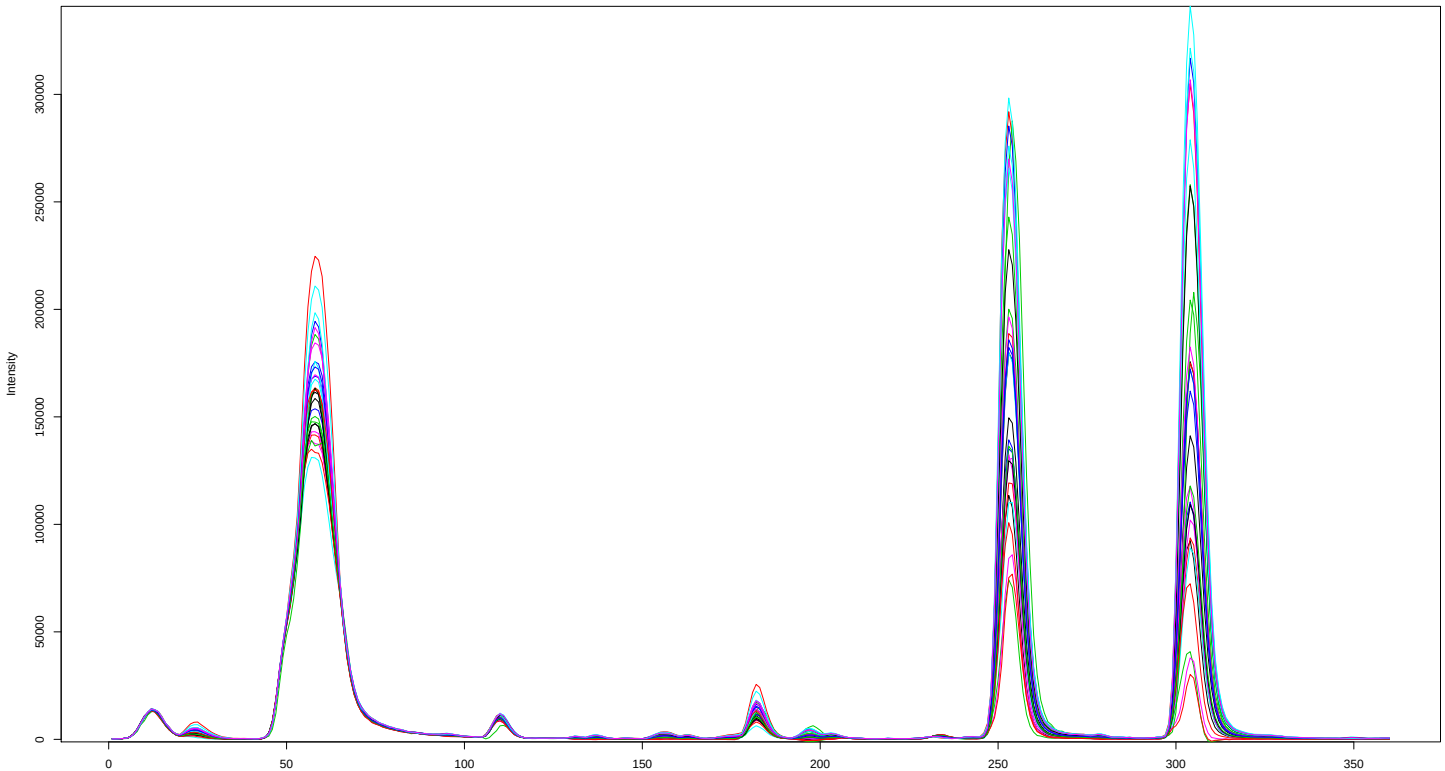
$$\begin{pmatrix} obs_{1,1} & \cdots & samples \\ \vdots & \ddots & \\ timepoints & \cdots & \end{pmatrix} = \begin{pmatrix} obs_{1,1} & \cdots & samples \\ \vdots & \ddots & \\ genes & \cdots & \end{pmatrix} \times (\beta) + (\epsilon)$$

Multivariate regression with PLS

Numbers

1. 109431241
2. 17463549
3. 11872
4. 1855
5. 11872+443680
6. 11872+3200

Predictions from PLS



Biological results

- The coefficient matrix can be seen as the effect of each gene at each time point
- At a given time point (peak) this can be used to construct a ranked list of genes
- Compare with existing knowledge of genes

Numbers

1. 109431241
2. 17463549
3. 11872
4. 1855
5. 11872+443680
6. 11872+3200
7. 300

Genetic algorithms

- How to determine the number of genes that go into the model?
- Possible solution is to use a genetic algorithm for the selection
- Allows selection from a larger set of genes, not just the top 100
- Fitness measure is the same as before

Genetic algorithms

- GA is an optimization algorithm
- Chromosomes 11110010101010101111000 each have a fitness measure
- High fitness chromosomes are more likely to have offspring
- Crossover 11110010101010101111000
  00101010111010101111000
- Mutation 01110010101010101111000

Genetic algorithms