Differential expression analysis of time-series RNA-seq experiments

Part II

Joyce Hsiao and Lauren Blake
Gilad and Lynch lab meeting
August 17, 2016

Motivating study: “iPSC differentiation into endoderm”

4 chimpanzees, all replicated once

6 Humans, 2 are replicated once

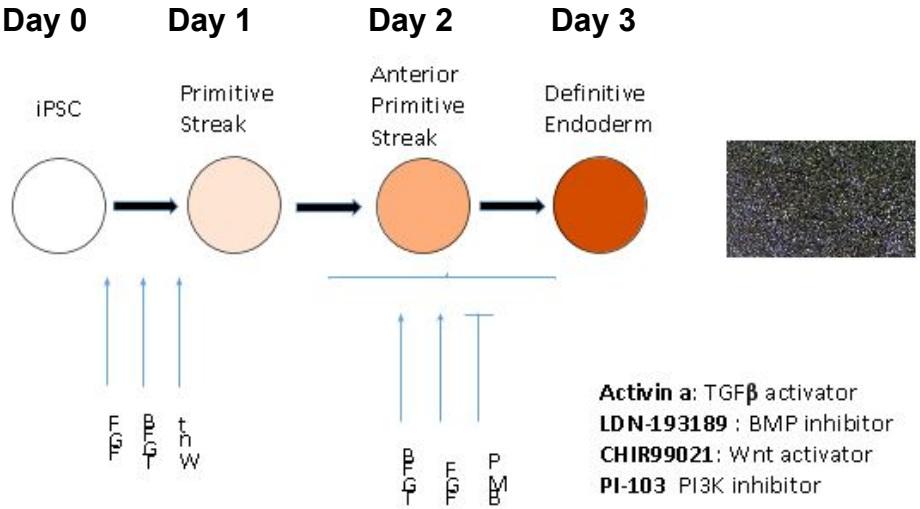


Figure Credit: Sammy Thomas

Our big questions:

1. For each species, identify significant gene expression change since Day 0.
2. Between species, identify significant species difference in gene expression pattern during the experiment.
3. Among genes with significant species difference, characteristics and groupings of expression curves.

Modeling time series gene expression data

Functional data analysis (FDA) models individual trajectories as a sample of random functions, which are not parametrically specified (Muller, 2006). Examples of parametric trend are linear or quadratic pattern, and an example of non-parametric trend is piecewise linear trend. For example, for gene i and individual j at time k,

$$y_{ijk} = \mu_i(t_{jk}) + \epsilon_{ijk}$$

Two methods to represent these functions

1. Basis expansion methods: represent the mean function as a linear combination of basis function. An example is polynomial function.

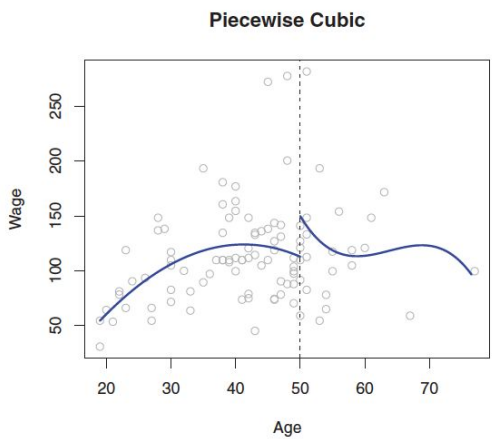
$$s(t) = \sum_{k=1}^{P} \beta_k s_k(t)$$

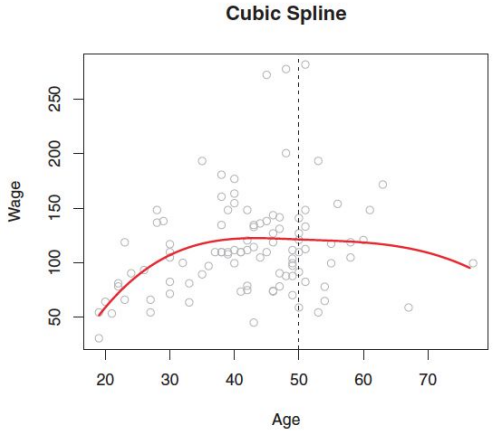
2. Smoothing splines: represent the mean function as a non-parametric function with smoothness constraint to control the shape of the curve.

$$\sum_{j=1}^{N} (y_{ij} - g_i(t_j))^2 + \lambda \int g_i''(t)^2 dt$$

EDGE (Extracting Differential Gene Expression)

Storey et al, 2005: For every gene i and individual j, model each time series using a natural cubic spline basis, specifically fitting a piecewise third-order polynomial with continuous second-order derivative at the knots.





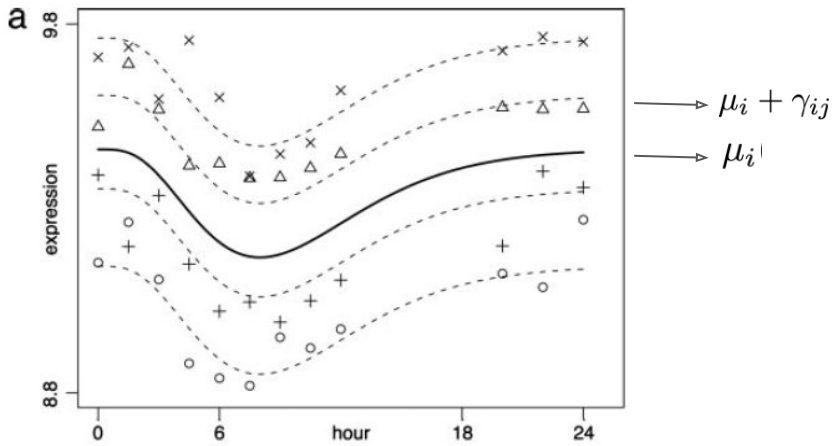
Introducing the EDGE model

Storey et al, 2005: For every gene i and individual j at time point k,

$$y_{ijk} = \mu_i(t_{jk}) + \gamma_{ij} + \epsilon_{ijk}$$

Individual random variation is model by γ_{ij}, with zero mean and gene-dependent variance.

Gene i



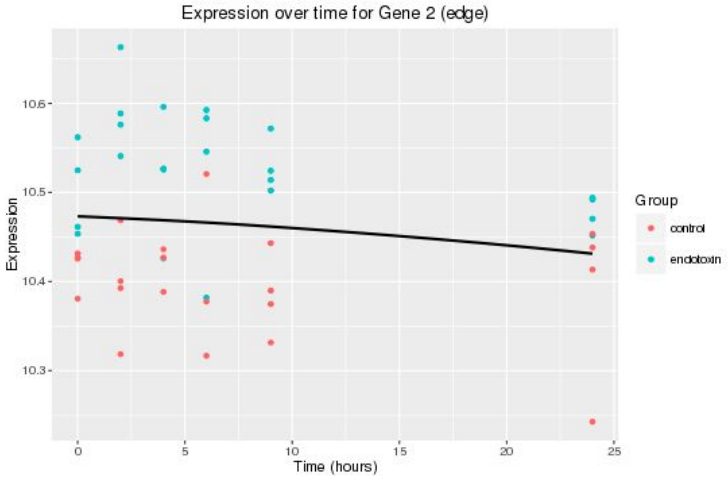
Key analytical concerns

- Choosing shape of the fitted curve
  - How many degrees of polynomial?
  - Same or different for every gene?
- Stating the null hypothesis
  - Consider changes within a biological condition and changes between biological conditions
  - Within a biological condition
    - Compare to baseline or the average
    - Is there a control condition?
  - Between biological conditions
    - Condition-specific variation?
- Testing the null hypothesis
  - Bootstrapping or not

Hypothesis testing: comparing expression trajectories

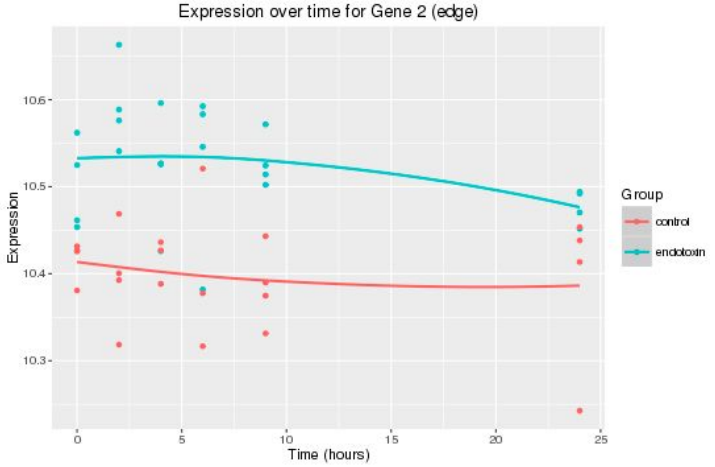
Under null hypothesis:
Single trajectory across time
(no group difference)

$$\hat{y}_{ij}^{0} = \sum_{k=1}^{p} \hat{\beta}_{ik}^{0} s(t_j)$$



Under alternative hypothesis:
Group-specific trajectories

$$\hat{y}_{ij}^{H} = \sum_{k=1}^{p} \hat{\beta}_{ik}^{H} s(t_j) \qquad \hat{y}_{ij}^{C} = \sum_{k=1}^{p} \hat{\beta}_{ik}^{C} s(t_j)$$



Computing test statistic

1. Compute sum of squares of the residuals under and under alternative.

$$SS_i^{null} = \sum_{j=1}^{N} [y_{ij} - \hat{y_{ij}}^{0}]^2 \qquad SS_i^{alt} = \sum_{j=1}^{N} [y_{ij} - \hat{y_{ij}}^{H}]^2 + \sum_{j=1}^{N} [y_{ij} - \hat{y_{ij}}^{C}]^2$$

2. Construct a statistic that is proportional to the typical F statistic.

$$F_i = \frac{SS_i^{null} - SS_i^{alt}}{SS_i^{alt}}$$

Obtaining null distribution of the test statistic

Bootstrap method

1. Compute residuals from the alternative model fit, so for human samples, we get within-individual residuals $\hat{\epsilon_{ijk}} = y_{ijk} - \hat{y}^H_{ijk}$ and between-individual residuals $\hat{\gamma}^H_{ij}$

2. Make bootstrapped data $y^*_{ijk} = (\hat{y}^0_{ijk} - \hat{\gamma}^0_{ij}) + \epsilon^*_i + \gamma^*_i$

   where ϵ^*_i and γ^*_i are randomly sampled with replacement from fitted residuals.

3. For each of the B iterations, compute F null statistic.

Choosing p

$$\mu_i(t) = \sum_{k=1}^{p} \beta_{ik} s_k(t)$$

$$\beta_{i1} s_1(t) + \beta_{i2} s_2(t) + \cdots + \beta_{ip} s_p(t)$$

p is determined by the number of regions, degree of polynomial and the number of constraints at the endpoints. For natural cubic splines over 3 regions,

p = 3 regions x 3 x 2 constraints = 12

1. If p is too large, then we lose power because degrees of freedom are wasted.
2. If p is too small, then power is lost because expression is not properly modeled.
3. Varying p by genes resulted in over-fitting the data and artificially inflating the significance.

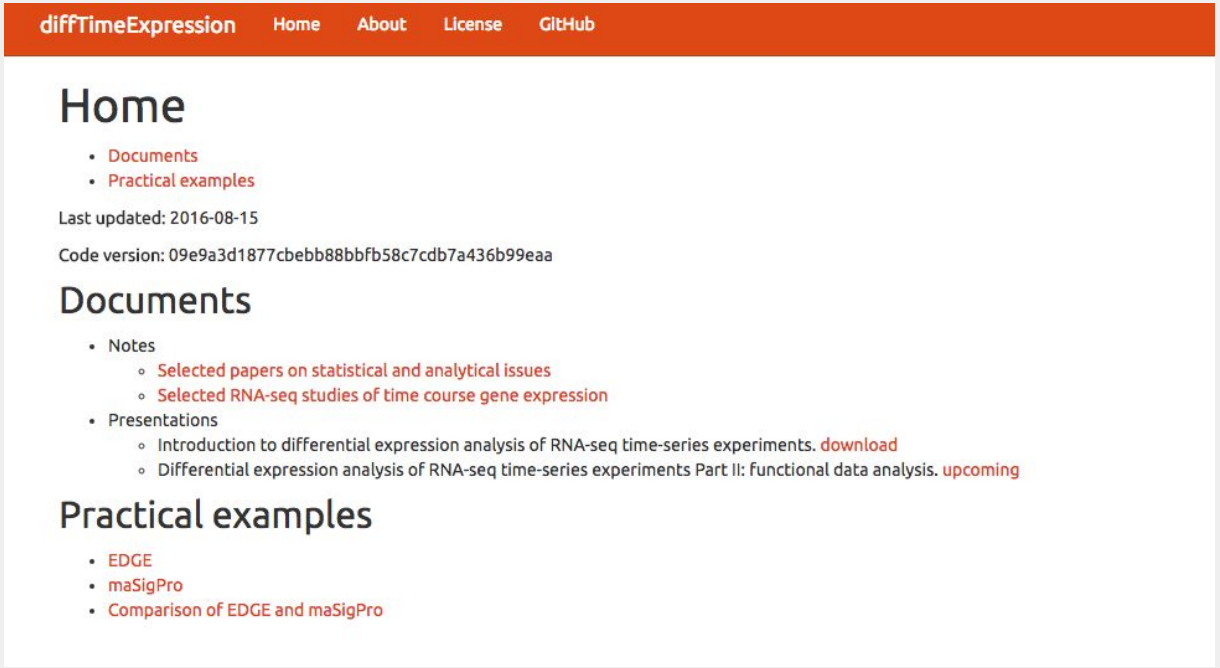
EDGE versus maSigPro

	EDGE	maSigPro
Input measurement	Continuous	Continuous
Curve function	Natural cubic splines	Polynomial
Null hypothesis	Same trajectory across groups	Same trajectory across groups
Allow individual replicates	No	No
Test statistic	Proportion to F	F
Bootstrapped null distribution	Yes	No

Note. Next maSigPro is a RNA-seq version of maSigPro, which assumes negative binomial distribution of the data.

Edge and maSigPro pipeline

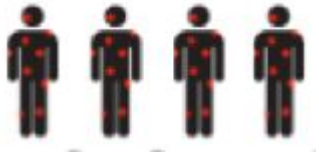
https://jhsiao999.github.io/diffTimeExpression/analysis/



Running edge and maSigPro on the same dataset

Interested in differences bet. 2 groups; Study design (from Calvano et al. 2005)

Cases w/ endotoxin

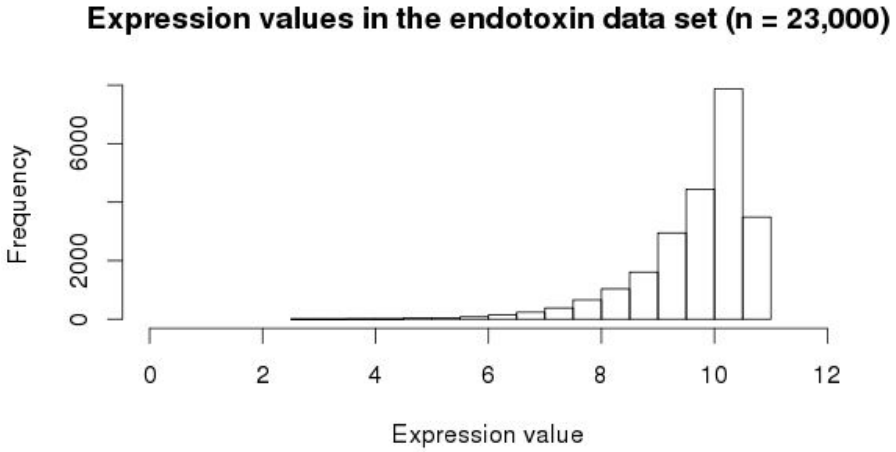


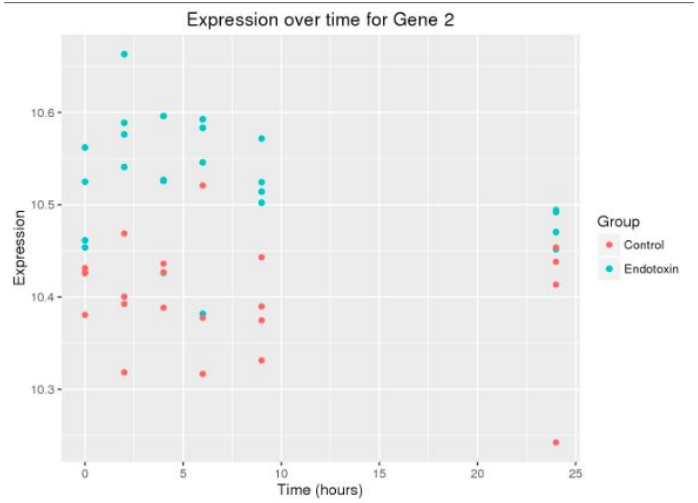
Controls w/ placebo

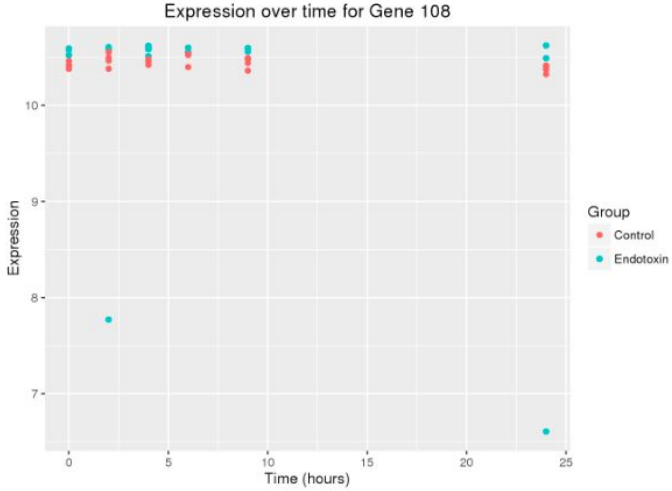
Blood draws at time 0 (before infusion), 2, 4, 6, 9, and 24 hours

Leukocyte isolation / RNA-Seq

Normalized expression data for 500 genes
Cov: ind., time, group.


Expression values in the endotoxin data set (n = 23,000)
Frequency
0
2000
6000
0
2
4
6
8
10
12
Expression value


Expression over time for Gene 2
Expression
10.6
10.5
10.4
10.3
0
5
10
15
20
25
Time (hours)
Group
Control
Endotoxin


Expression over time for Gene 108
Expression
10
9
8
7
0
5
10
15
20
25
Time (hours)
Group
Control
Endotoxin

Model comparison

EDGE: For every gene, person j, time k,

H_o : $exp_{jk} = \boldsymbol{\alpha}_j$ x group + $\boldsymbol{\beta}_j$ * spline basis + $\boldsymbol{\epsilon}_{jk}$

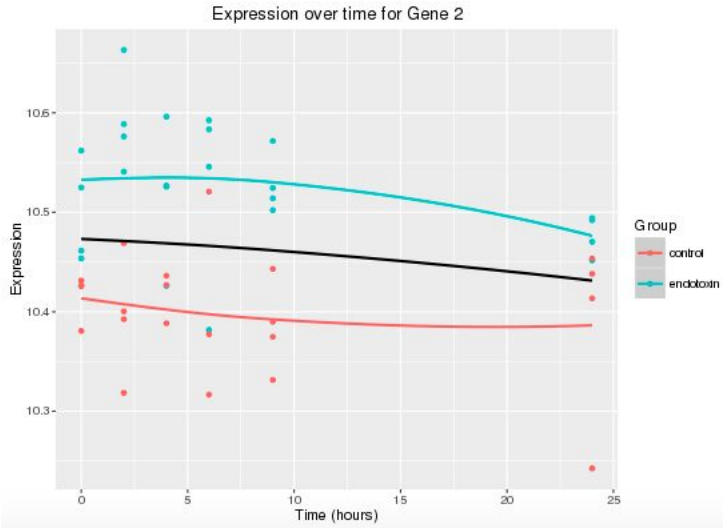
H_A : $exp_{jk} = \boldsymbol{\alpha}_j$ x group + $\boldsymbol{\beta}_j$ * spline basis + $\boldsymbol{\gamma}_j$ * $\boldsymbol{\beta}_j$ * spline basis (interaction) + $\boldsymbol{\epsilon}_{jk}$

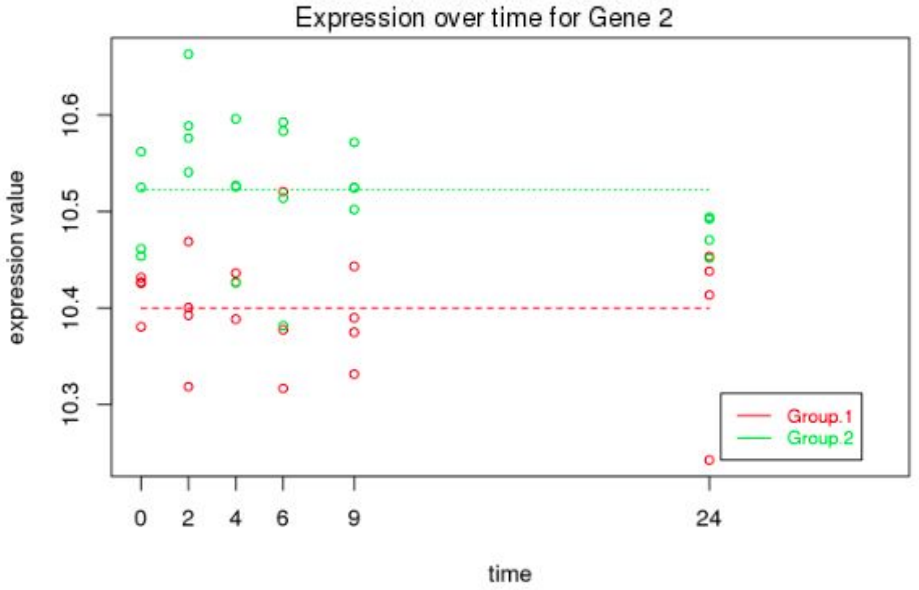
maSigPro: For every gene person j, time k,

H_o : $\boldsymbol{u}_o + \boldsymbol{u}_1 t_j + \boldsymbol{u}_2 t_j^2 + \boldsymbol{\epsilon}_{jk}$

H_A : $exp_{jk} = \boldsymbol{\beta}_o + \boldsymbol{\beta}_1 t_j + \boldsymbol{\beta}_2 t_j^2 + \boldsymbol{d}_o + \boldsymbol{d}_1 t_j + \boldsymbol{d}_2 t_j^2 + \boldsymbol{\epsilon}_{jk}$

Comparison of results

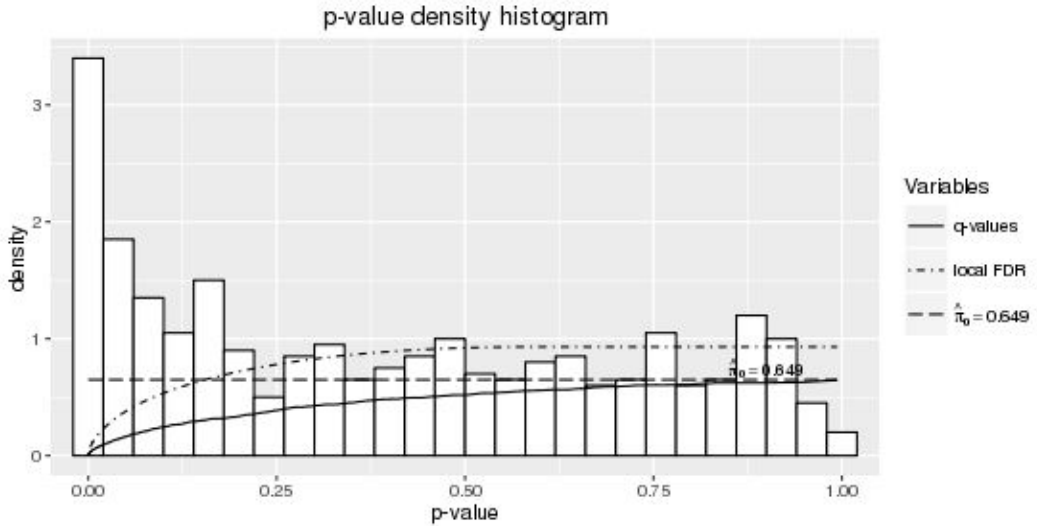




Edge

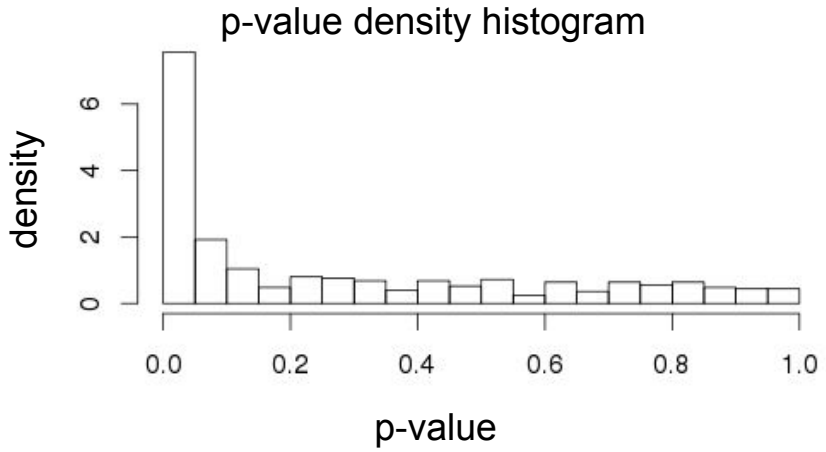
```
Cumulative number of significant calls:

           <1e-04 <0.001 <0.01 <0.025 <0.05 <0.1  <1
p-value        14     24    51     75   100  132 500
q-value         0      0    22     25    42   72 500
local FDR       0      0    14     21    24   35 500
```

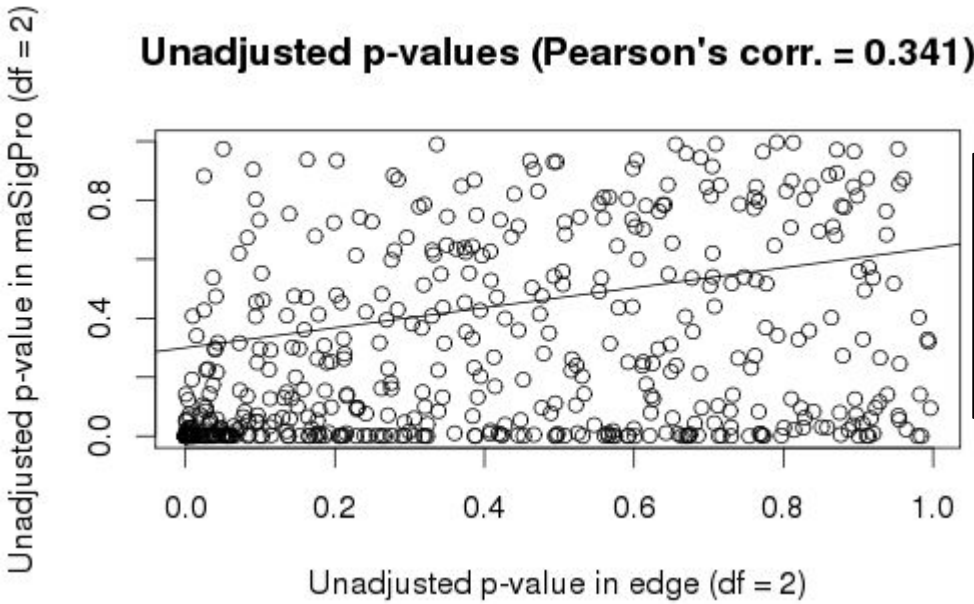


maSigPro

	<1e-4	<1e3	<0.01	<0.025	<0.05	<0.1	<1
p-val	61	96	142	162	189	237	500

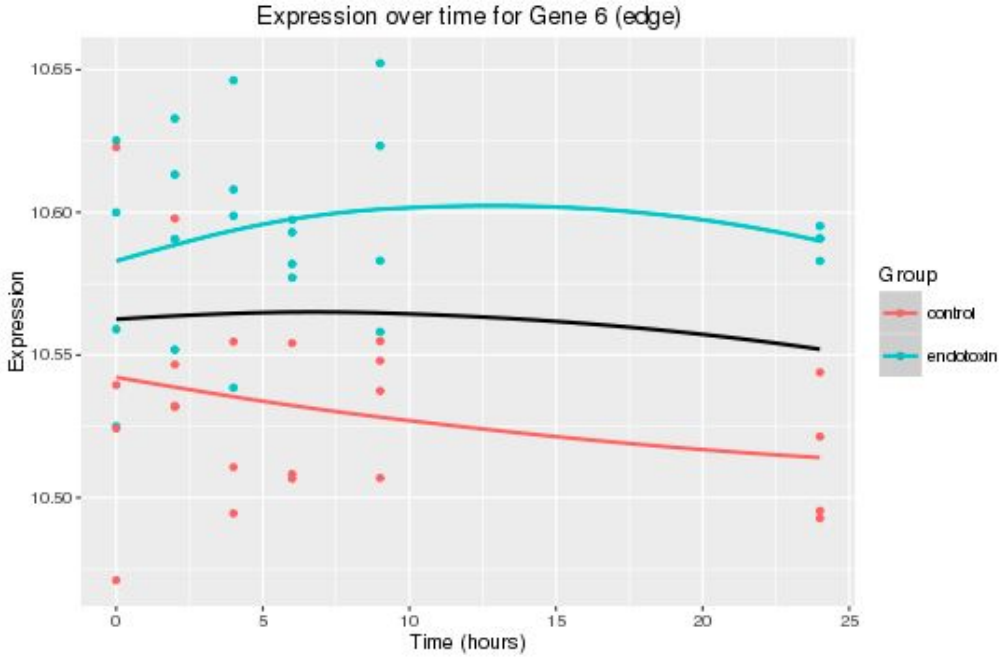


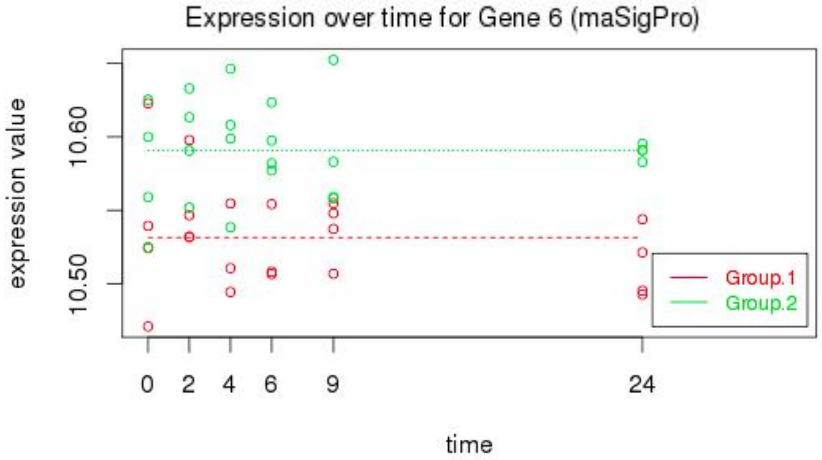
Comparison of unadjusted p-values in 2 programs



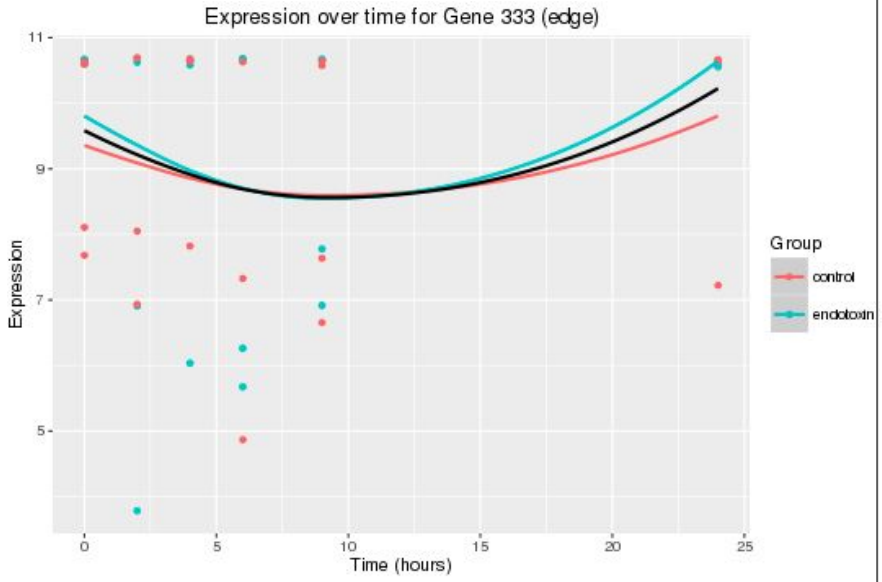
Program	unadj_pvalues_0.01	unadj_pvalues_0.1
edge	53	131
maSigPro	142	237
Genes in common	38	100

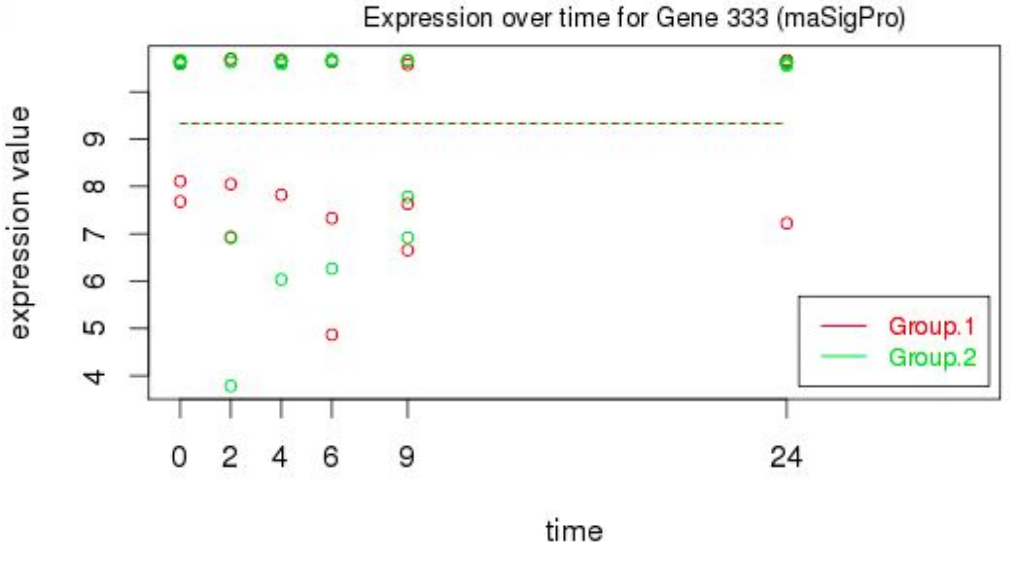
When maSigPro calls a gene significant (adj. p-val. = $6.2*10^{-4}$) and edge does not (adj. p-val = 0.90)





When edge calls a gene significant (adj. p-val. = 0.093) and maSigPro does not (adj. p-val. = 0.589)





Edge Pipeline Part I

1) Create splines

```
# Null model: make the basis matrix for natural cubic splines
null_model <- ~grp + ns(tme, df = 2, intercept = FALSE)

# Full model: make the basis matrix for natural cubic splines
full_model <- ~grp + ns(tme, df = 2, intercept = FALSE) + (grp):ns(tme, df = 2, intercept = FALSE)
```

```
de_obj <- build_models(data = endoexpr, cov = cov, full.model = full_model, null.model = null_model)
```

2) Fit the null and full models using least sqs.

```
ef_obj <- fit_models(de_obj, stat.type = "odp")
```

Edge Pipeline Part II

3) Significance testing

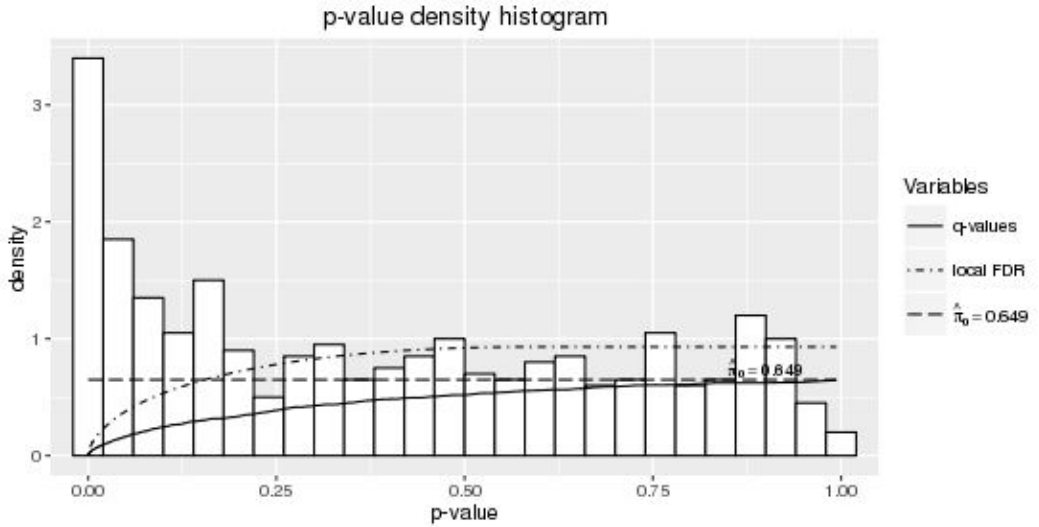
```
# Run odp

de_odp <- odp(de_obj, bs.its = 50, verbose = FALSE, n.mods = 50)

# See the results
summary(de_odp)
```

```
Cumulative number of significant calls:

           <1e-04 <0.001 <0.01 <0.025 <0.05 <0.1  <1
p-value        14     24    51     75   100  132 500
q-value         0      0    22     25    42   72 500
local FDR       0      0    14     21    24   35 500
```



maSigPro Pipeline Part I

1) Make the experimental design matrix

	Time	Replicate	Group.1	Group.2
G1.T1.1	0	1	1	0
G1.T2.1	0	1	1	0
G1.T6.1	0	1	1	0
G1.T3.1	0	1	1	0
G1.T4.1	2	2	1	0
G1.T5.1	2	2	1	0

2) Make a regression matrix for the full regression model. **Discussion: degrees!**

```
matrix_endo_design <- make.design.matrix(final_endo_design, degree = 5)
```

	Group.2vsGroup.1	Time	TimexGroup.2	Time2	Time2xGroup.2	Time3	Time3xGroup.2	Time4
G1.T1.1	0	0	0	0	0	0	0	0
G1.T2.1	0	0	0	0	0	0	0	0
G1.T6.1	0	0	0	0	0	0	0	0
G1.T3.1	0	0	0	0	0	0	0	0
G1.T4.1	0	2	0	4	0	8	0	16
G1.T5.1	0	2	0	4	0	8	0	16

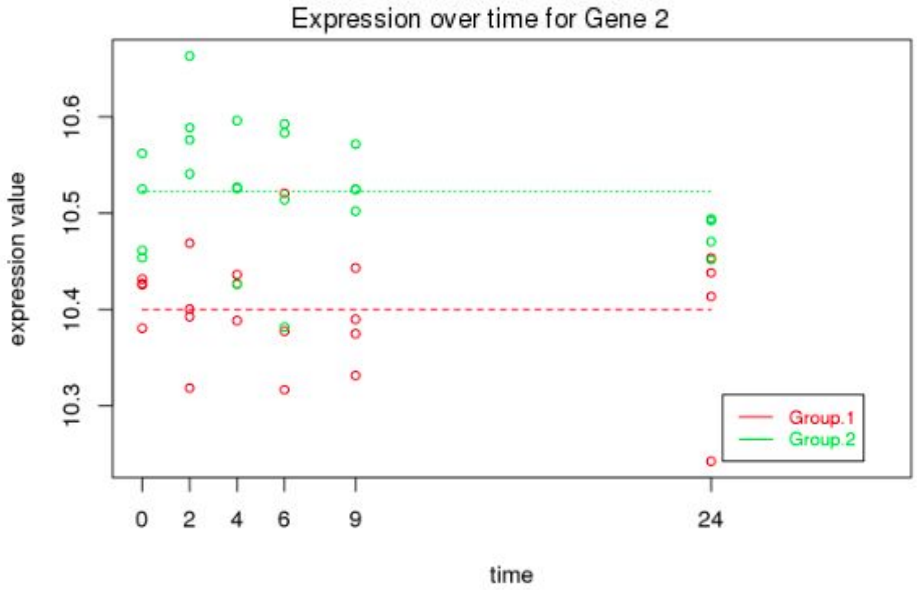
	Time4xGroup.2	Time5	Time5xGroup.2
G1.T1.1	0	0	0
G1.T2.1	0	0	0
G1.T6.1	0	0	0
G1.T3.1	0	0	0

maSigPro Part II

3) Compute a regression fit for each gene. It will compute an unadjusted p-value and a p-value associated with the F-statistic of the model (here, FDR = 10%)

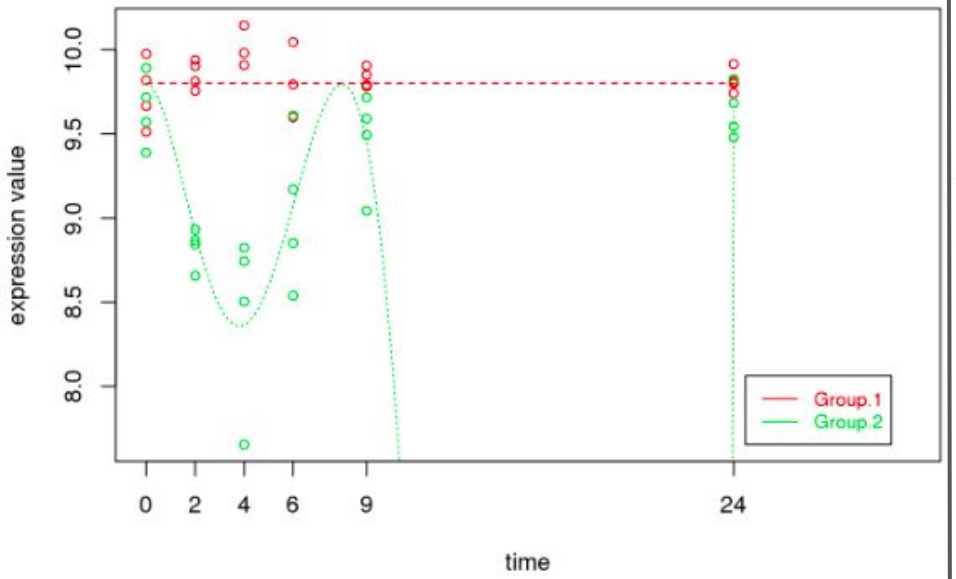
```
endo_p_vector <- p.vector(t_endo_data, matrix_endo_design, counts = TRUE, theta = 10, Q = 0.10, MT.adjust = "BH")
```

- Unadjusted and adjusted p-values for each gene
- Genes significant at FDR 10%

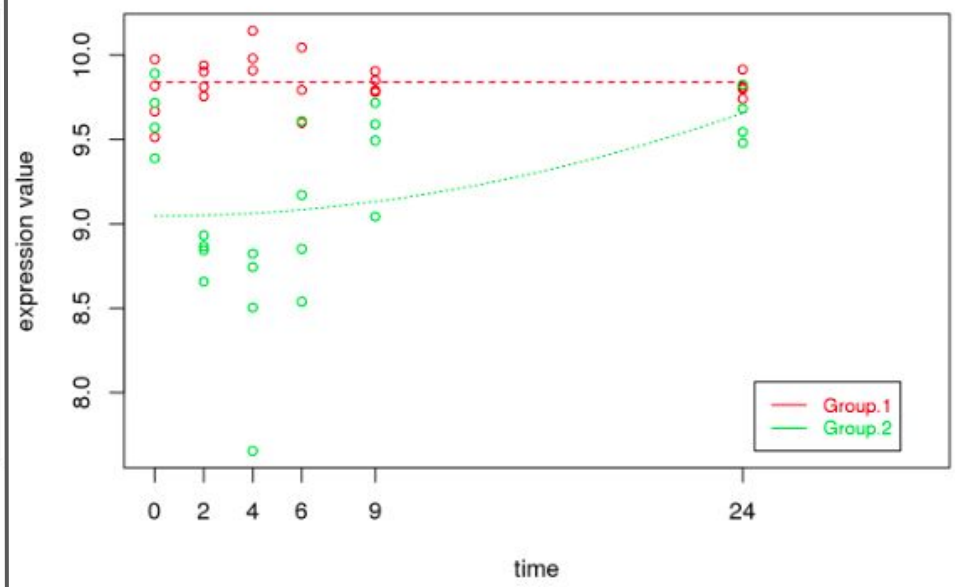


Aside: degrees does matter for some genes

Regression for Gene 108 when degree = 5



Regression for Gene 108 when degree = 2



Correlation between unadjusted p-values when degrees = 2 and degrees = 5: 0.721

maSigPro Part IV

4) For the significant genes in step 3, use forward stepwise regression to find which coefficients are statistically significant → different genes have different coefficients in the “best” model

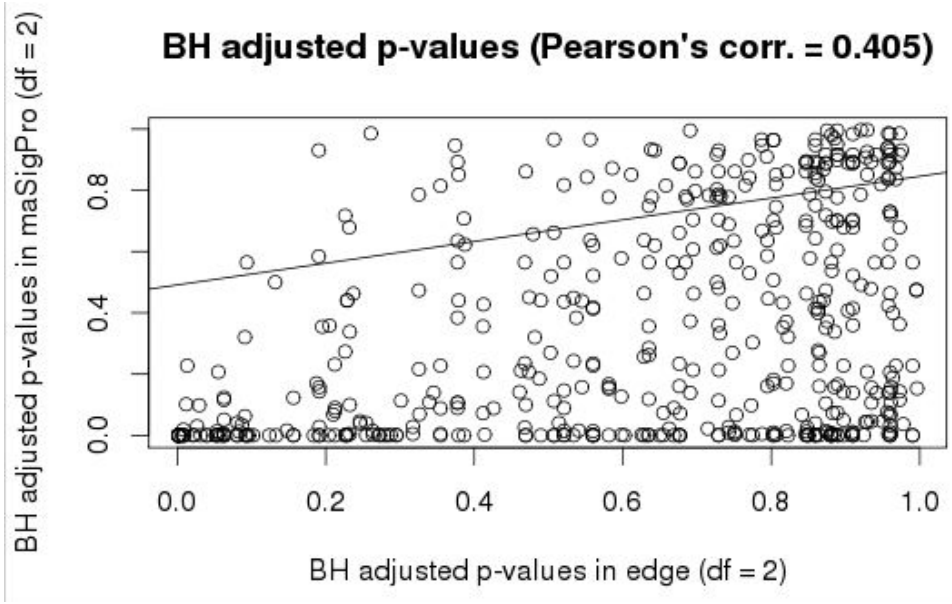
```
endo_t_stat <- T.fit(endo_p_vector)
```

```
endo_sig_genes <- get.siggenes(endo_t_stat, rsq = 0.2, var = "groups")
```

```
$group.coeffs
    Group.1_beta0 Group.1_beta1 Group.1_beta2 Group.2_beta0 Group.2_beta1 Group.2_beta2
2        2.341797  0.0000000000  0.000000e+00      2.353548  0.0000000000  0.000000e+00
4        2.332467  0.0000000000  0.000000e+00      2.332467  0.0046917019 -1.970172e-04
6        2.354382  0.0000000000  0.000000e+00      2.359995  0.0000000000  0.000000e+00
7        2.334190  0.0000000000  0.000000e+00      2.345320  0.0000000000  0.000000e+00
8        2.287567  0.0000000000  0.000000e+00      2.258974  0.0000000000  5.367678e-05
```

References

1. Storey et al 2015
2. Muller 2006
3. Conesa 2006



Program	unadj_pvalues_0.01	unadj_pvalues_0.1	adj_pvalues_FDR_0.01	adj_pvalues_FDR_0.1
edge	53	131	19	54
maSigPro	142	237	119	177
Genes in common	38	100	18	46