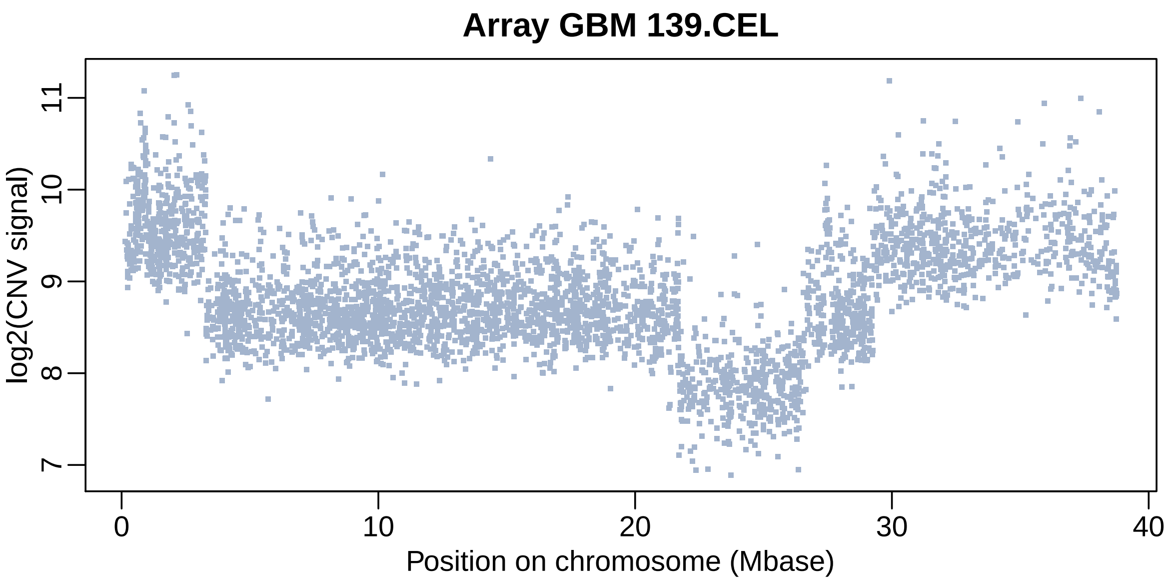
Sharp Piece-wise Linear Smoothing and Deconvolution with an L_0 Penalty

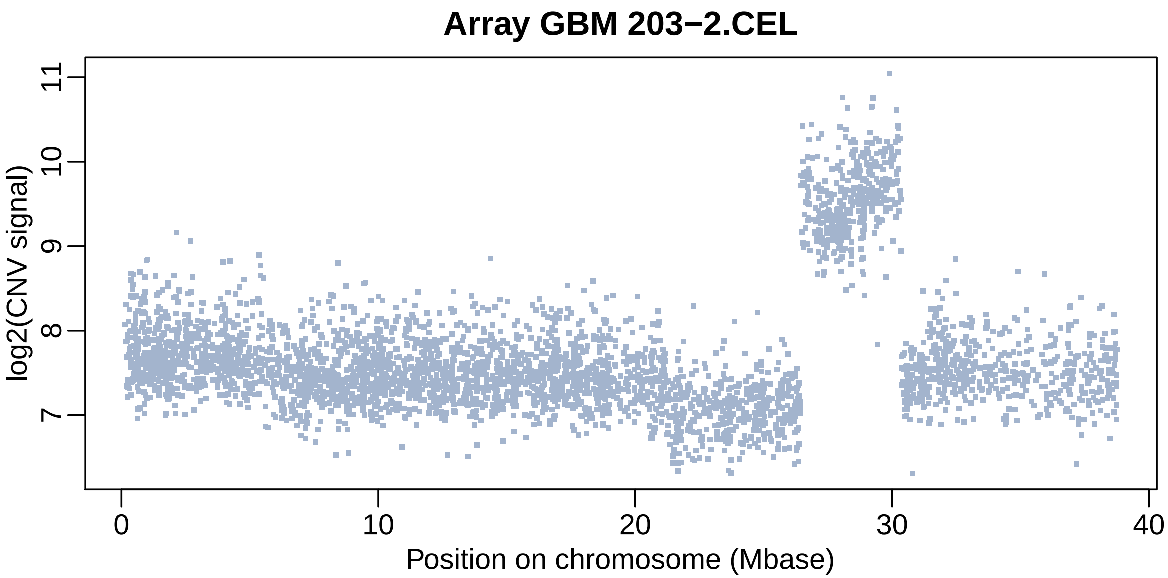
Paul Eilers[1] and Ralph Rippe[2]

[1]*Erasmus Medical Center, Rotterdam, The Netherlands*
[2]*Leiden University Medical Center, Leiden, The Netherlands*

The University of Warwick, March 2012

Data from genetics: DNA copy numbers in tumors





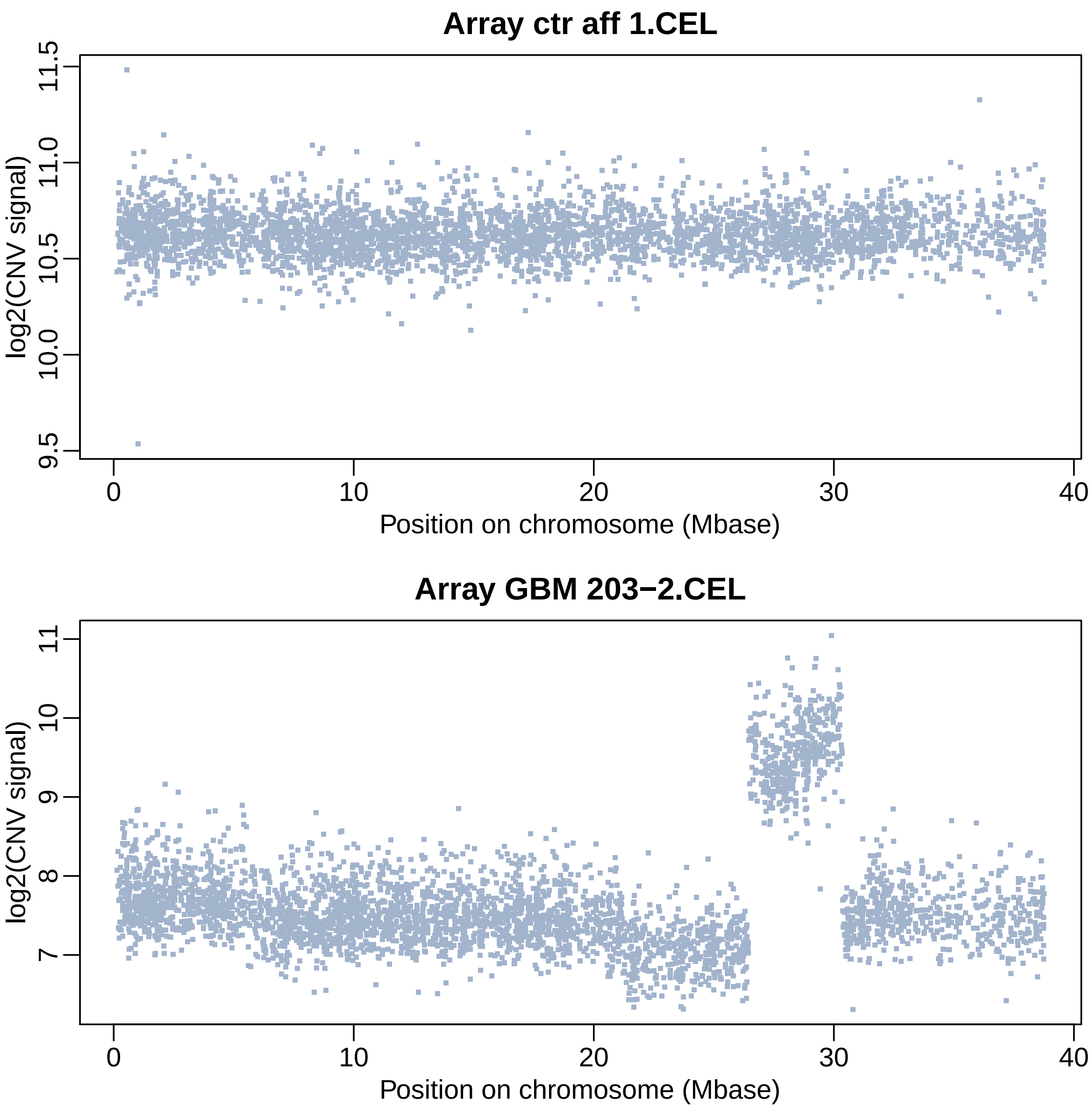
Biological background, simplified

- We have two copies of each chromosome
- Most of our DNA is identical for everyone
- But not at millions of SNPs (single nucleotide polymorphisms)
- A SNP has two possible states, called alleles, say A and B
- Possible combinations (genotypes): AA, AB, BB
- Microarrays allow detection of the states of very many SNPs
- Using selective hybridization and fluorescence technology
- For each SNP we get two signals, a for A, b for B

Copy number variations

- Two signals, a proportional to A, b proportional to B
- Signal b is 0, c, $2c$ for AA, AB, BB genotypes
- Signal a is $2c$, c, 0 for AA, AB, BB genotypes
- So $a + b = 2c$ (save for noise, background, non-linearities, ...)
- Normal DNA is quite boring, but tumor DNA is not
- DNA segments may be deleted or multiplied (amplified)
- So we can have A, B, AA, AB, BB, AAA, AAB, ABB, and so on
- Copy number changes (CNV); $a + b \neq 2c$ will reflect that

CNV in normal (top) and tumor (bottom) DNA



A simple smoother: Whittaker

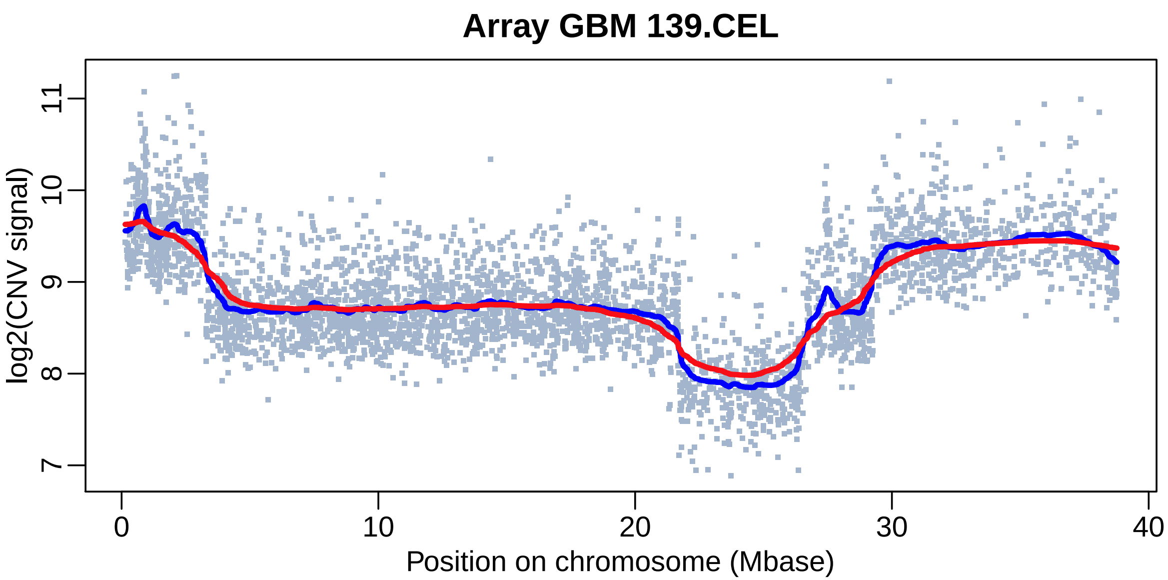
- Whittaker (1923) proposed "graduation": minimize

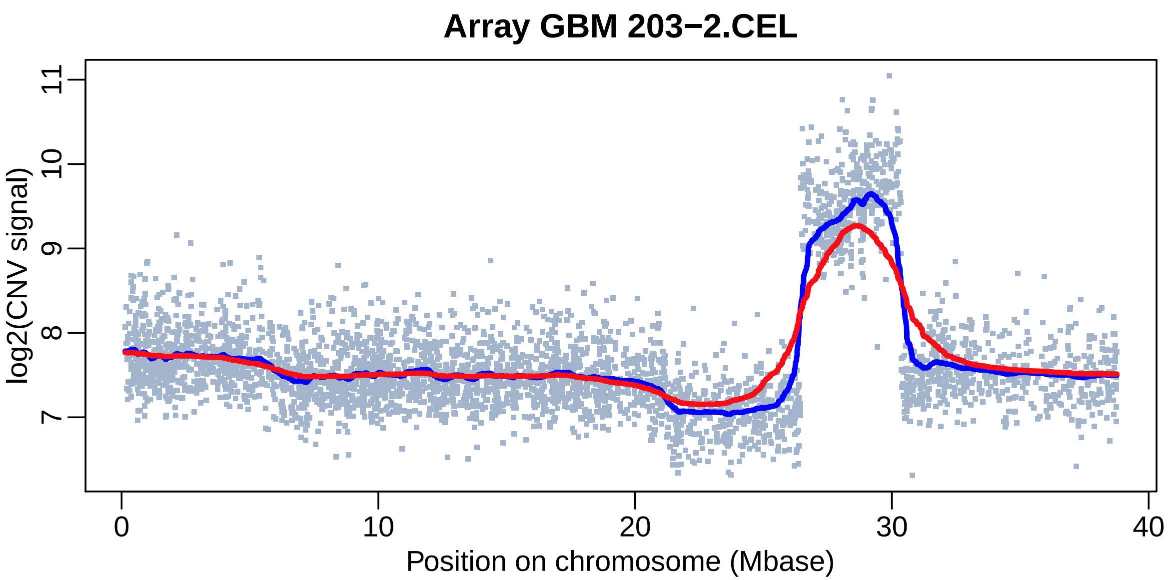
$$S_2 = \sum_i (y_i - z_i)^2 + \lambda \sum_i (\Delta^d z_i)^2$$

- Given a noisy data series y, it finds a smoother series z
- Operator Δ^d forms differences of order d
- Today we call this penalized least squares
- Explicit solution, with matrix D, such that $\Delta^d z = Dz$:

$$(I + \lambda D'D)\hat{z} = y$$

The Whittaker smoother ($d = 2$) in action





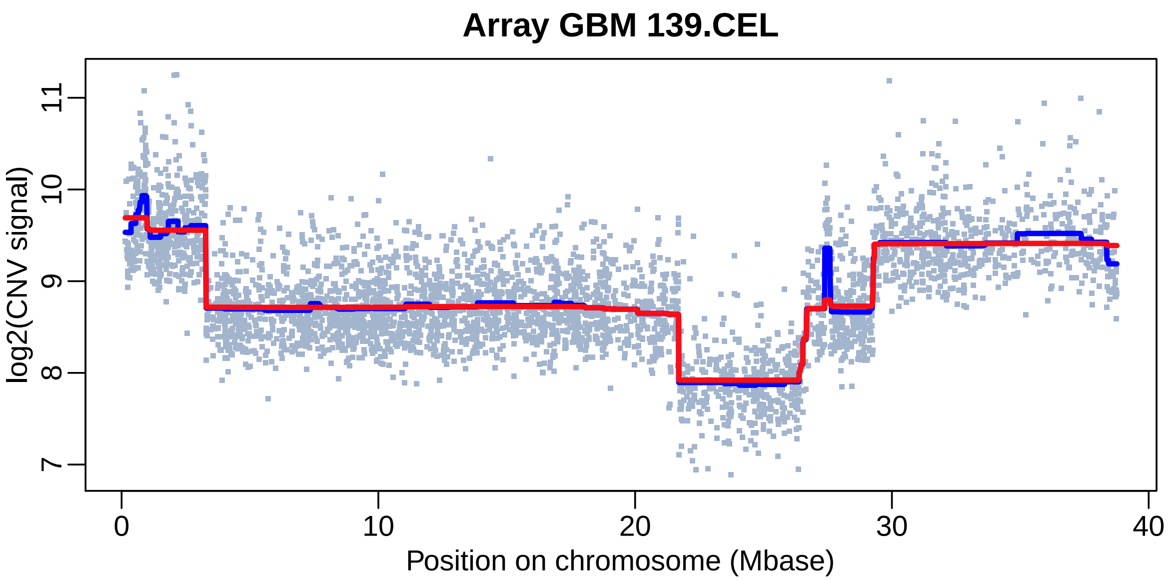
Critique of the Whittaker smoother, and a solution

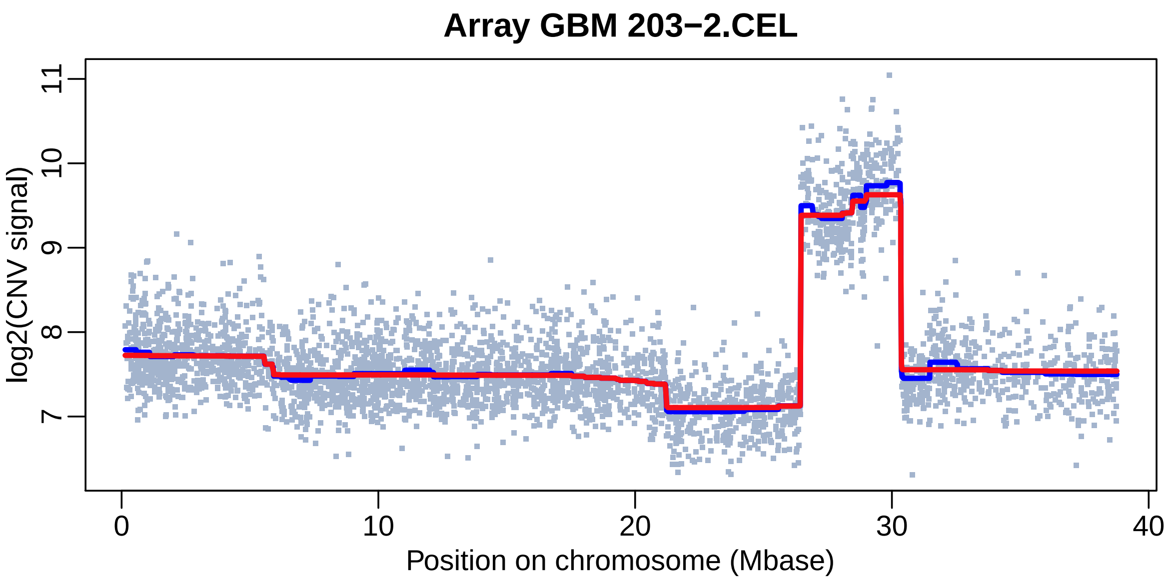
- Noise is effectively reduced
- But jumps are rounded
- Alternative approach, inspired by LASSO, minimizes

$$S_1 = \sum_i (y_i - z_i)^2 + \lambda \sum_i |\Delta z_i|$$

- Total variation penalty or "fused LASSO"
- Notice the first differences
- The L_1 norm (instead of the L_2 norm) in the penalty
- It is a big improvement

The L_1 penalty in action





Computation for the L_1 penalty

- We could try quadratic programming techniques
- But there is an easier solution
- For any x and approximation $\tilde{x}$ we have $|x| = x^2/|x| \approx x^2/|\tilde{x}|$
- Use weighted L_2 penalty, with $v_i = 1/|\Delta \tilde{z}_i|$:

$$S_1 = \sum_i (y_i - z_i)^2 + \lambda \sum_i v_i (\Delta z_i)^2$$

- Iteratively update v and $\tilde{z}$
- Solve $(I + D'VD)z = y$ repeatedly, with $V = \text{diag}(v)$
- Some smoothing near 0: use $v_i = 1/\sqrt{(\Delta \tilde{z}_i)^2 + \beta^2}$, with small β

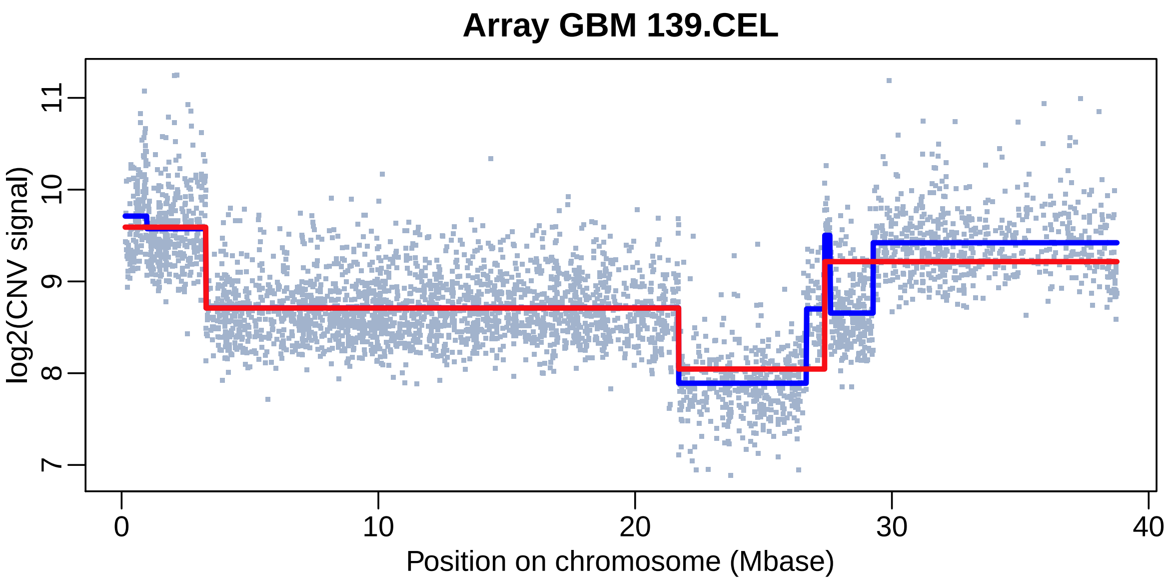
Critique of the L_1 penalty

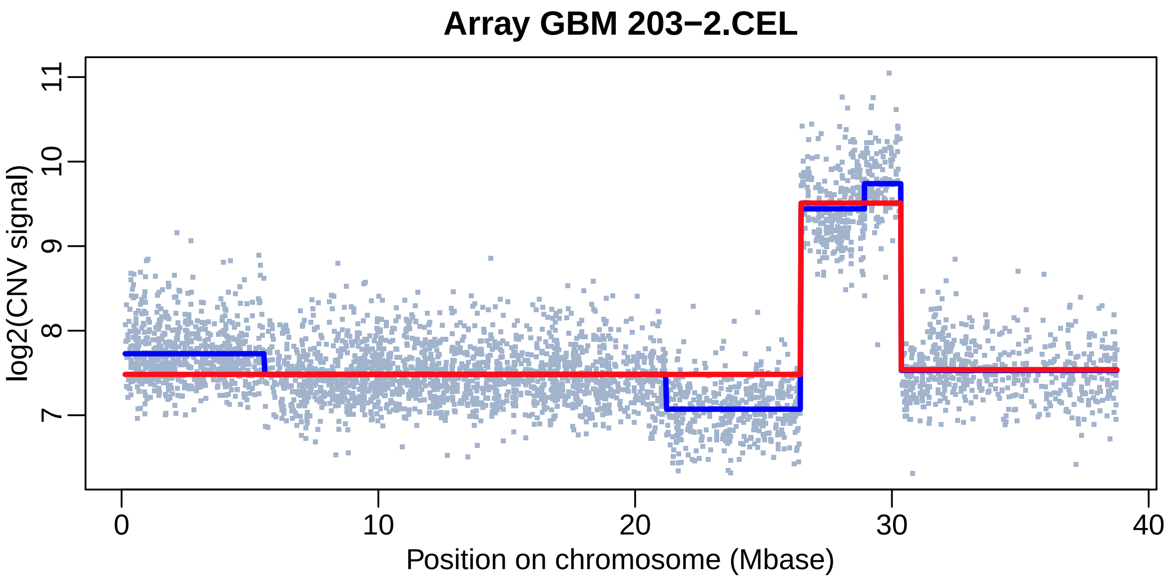
- We certainly get a big improvement
- But the jumps are not completely crisp
- Solution: a penalty with (implicitly) the L_0 norm
- Iterate as above with weighted quadratic penalty

$$S_0 = \sum_i (y_i - z_i)^2 + \lambda \sum_i v_i (\Delta z_i)^2$$

- With $v_i = 1/(\tilde{x}^2 + \beta^2)$ instead of $v_i = 1/\sqrt{(\Delta \tilde{z}_i)^2 + \beta^2}$

The L_0 penalty in action





Sparseness

- We have many equations (4032 here), but a banded system
- Computation time linear in data length
- R package `spam` is great (sparse matrices, Matlab-style)
- L_2 system solved in 20 milliseconds

```
# Whittaker smoother
m = length(y)
E = diag.spam(m)
D = diff(E, diff = 2)
P = lambda * t(D) %*% D
z = solve(E + P, y)
```

Optimal smoothing

- Interactive use in the hands of biologists is our main goal
- But we can try “optimal” smoothing, using AIC
- $\mathrm{AIC} = 2m \log \hat{\sigma} + 2 * \mathrm{ED}$
- With $\hat{\sigma}^2 = \sum (y_i - \hat{z}_i)^2 / m$ (ML estimate of error SD)
- Effective dimension ED
- $\mathrm{ED} = \mathrm{tr}(I + D'VD)$, with $V = \mathrm{diag}(v)$
- Serial correlation in errors can spoil AIC (undersmoothing)

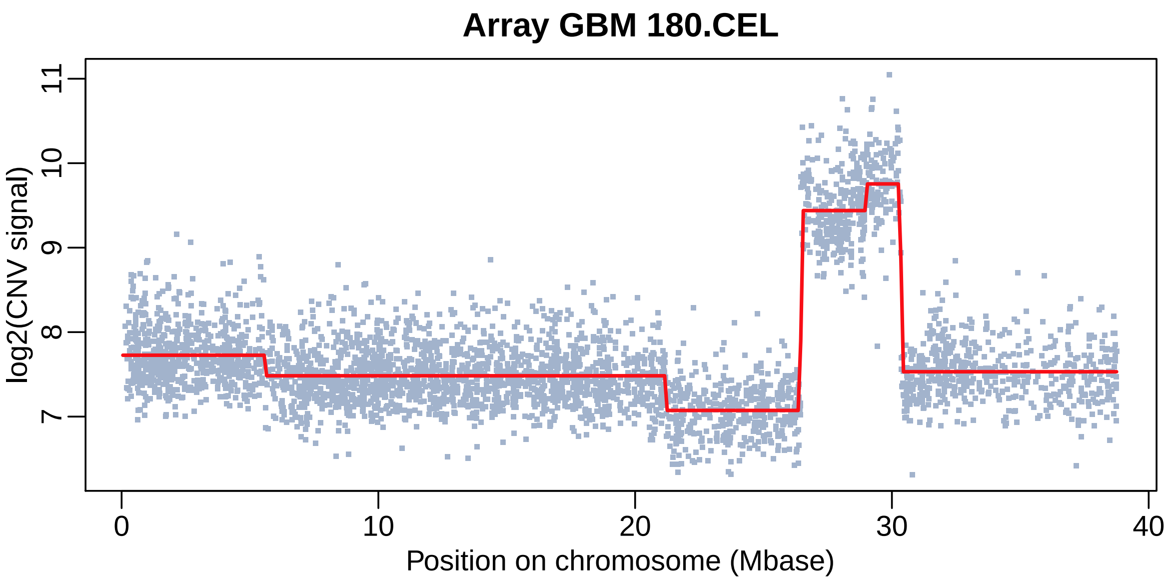
Computational details for AIC

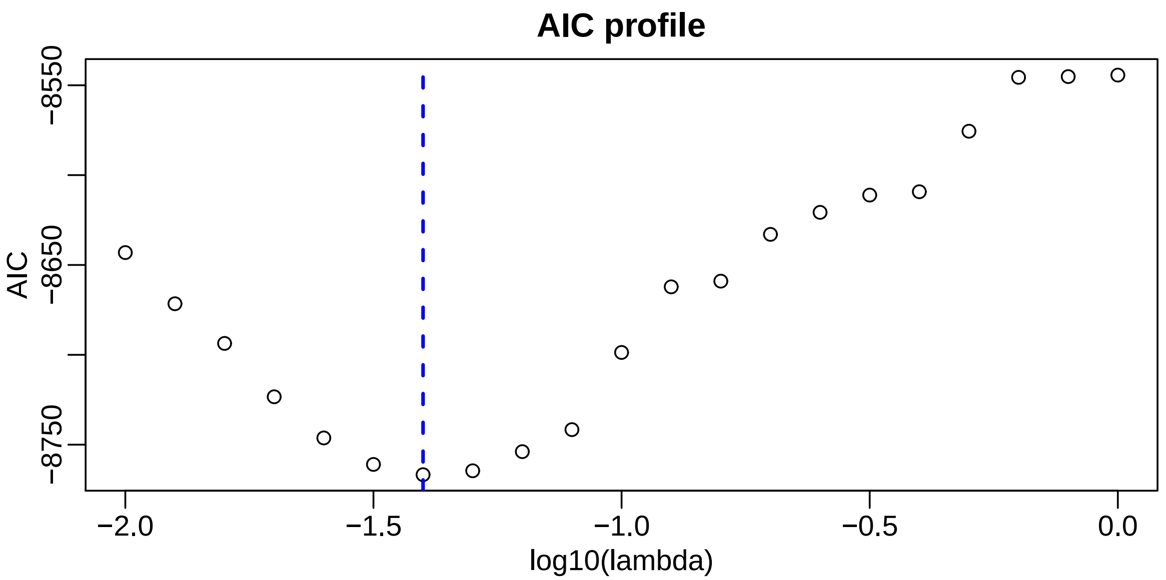
- We need to compute $\mathrm{ED} = \mathrm{tr}(I + D'VD)$, with $V = \mathrm{diag}(v)$
- But this is a large (4032 squared) matrix
- Beautiful solution: Hutchinson and De Hoog algorithm
- But not yet implemented
- For now: use 400 intervals and indicator basis R

$$S_i = \sum_i (y_i - \sum_j r_{ij} a_j)^2 + \sum_j v_j (\Delta a_j)^2$$

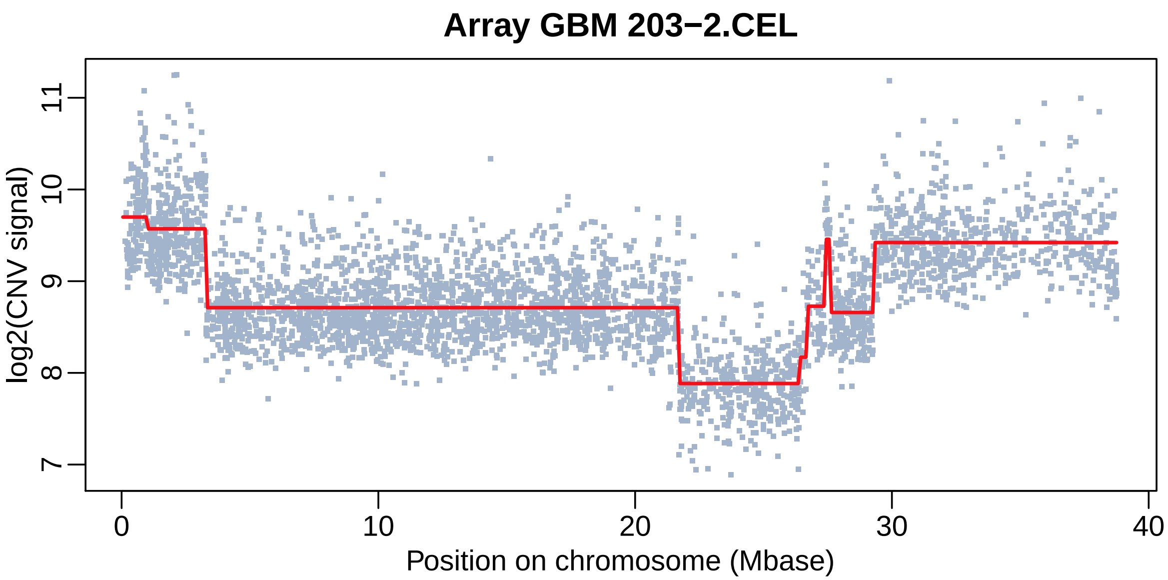
- Adaptive weights $v_j = 1/((\Delta \tilde{a}_j)^2 + \beta^2)$

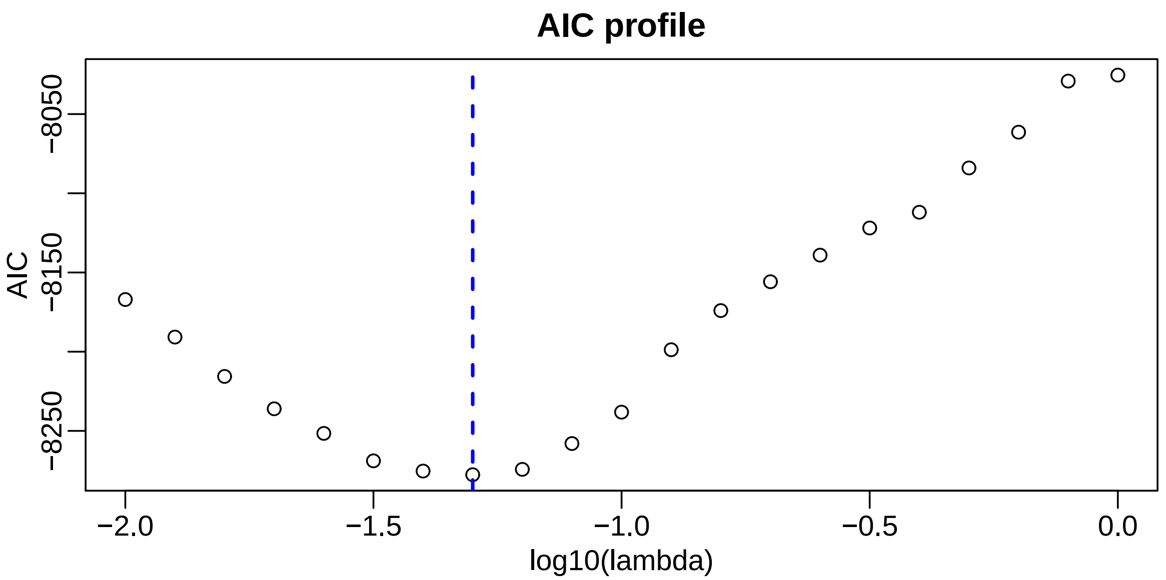
AIC for one array





AIC for the other array

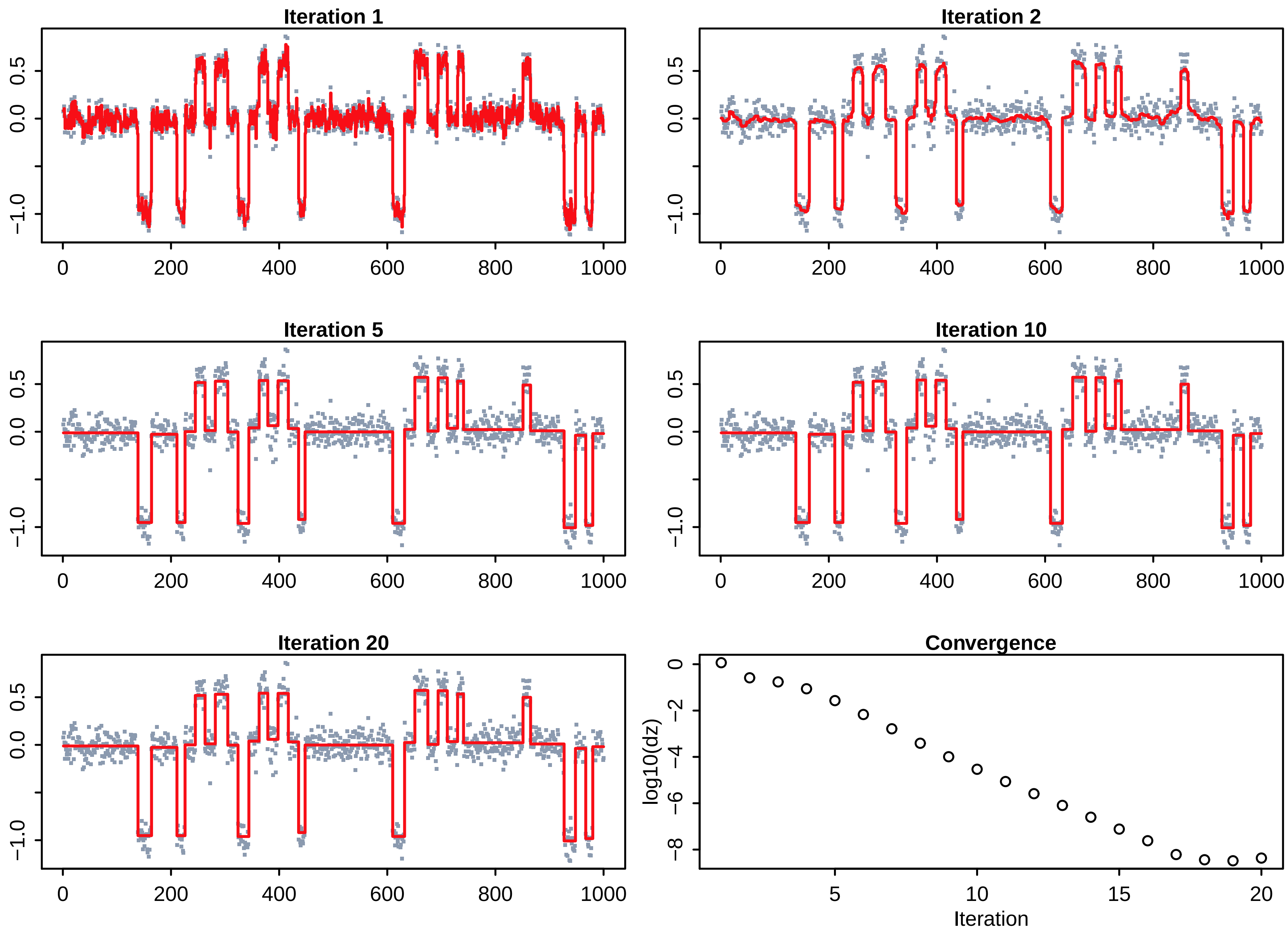




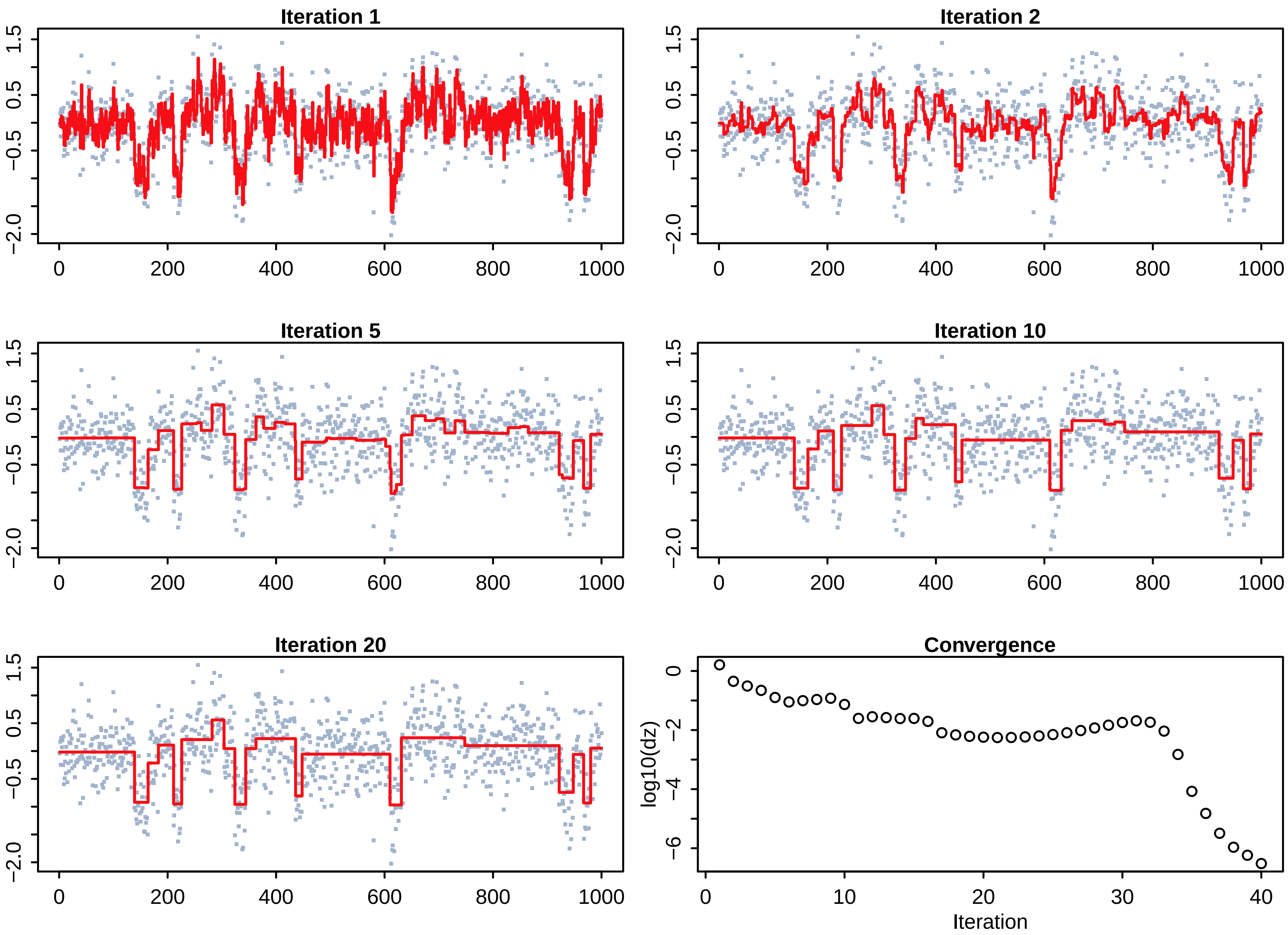
Tests on simulated data

- We did some tests on simulated data
- Using results in the Bioconductor package `VEGA`
- Starting with adaptive weights $v_i \equiv 1$
- Graphs will show intermediate results
- And the size of changes per iteration
- The smoother starts with a “wild” solution
- It gradually removes more and more details

Example of convergence, little noise



Example of convergence, more noise



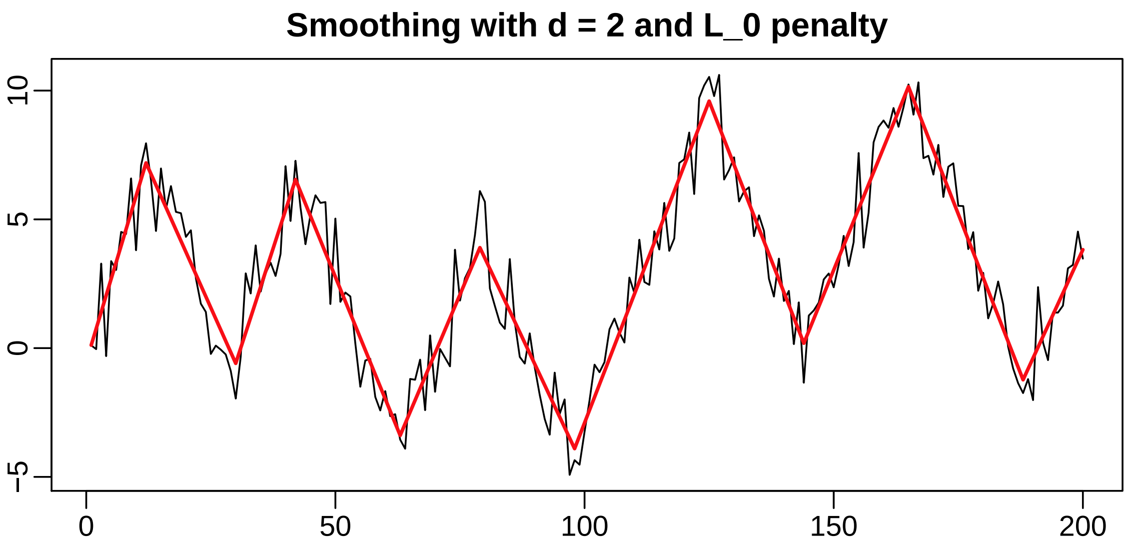
Can we trust it?

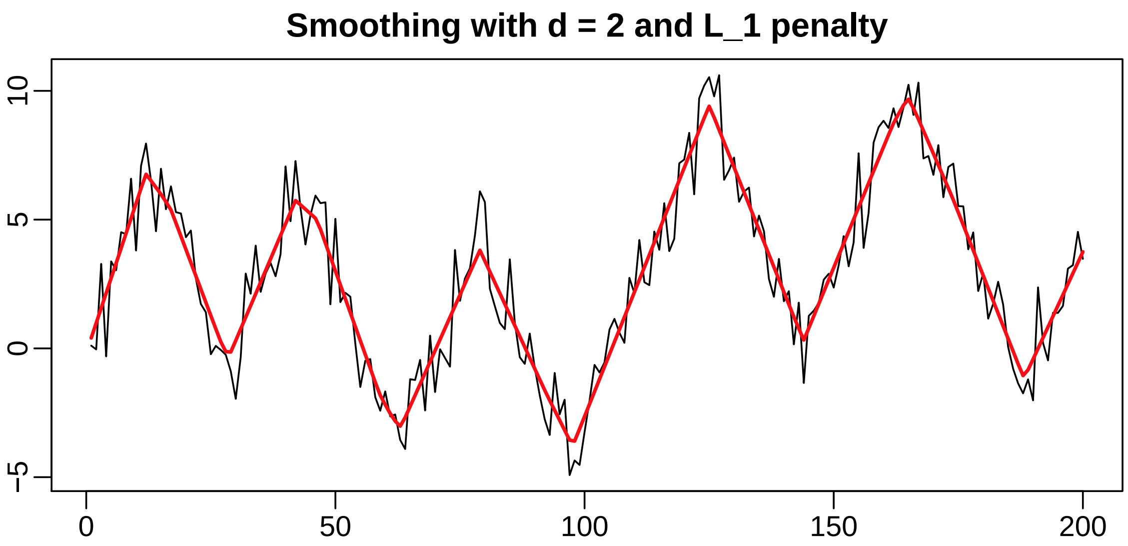
- The objective function is non-convex
- In contrast to penalties with L_2 or L_1 norm
- Yet we consistently get quite good results
- Convergence history looks OK
- We cannot be sure that we found a global minimum
- But should we care?

A variation on our theme

- Using first order differences in an L_0 penalty we get
  - constant segments
  - jumps between segments
- With second order differences we get
  - linear segments
  - kinks between segments

A kinky simulation

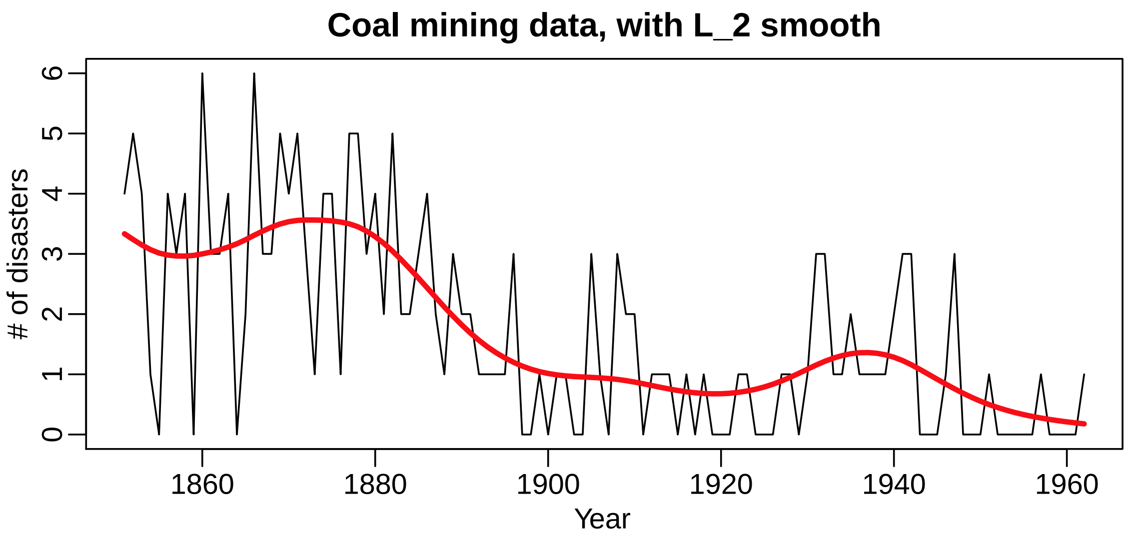


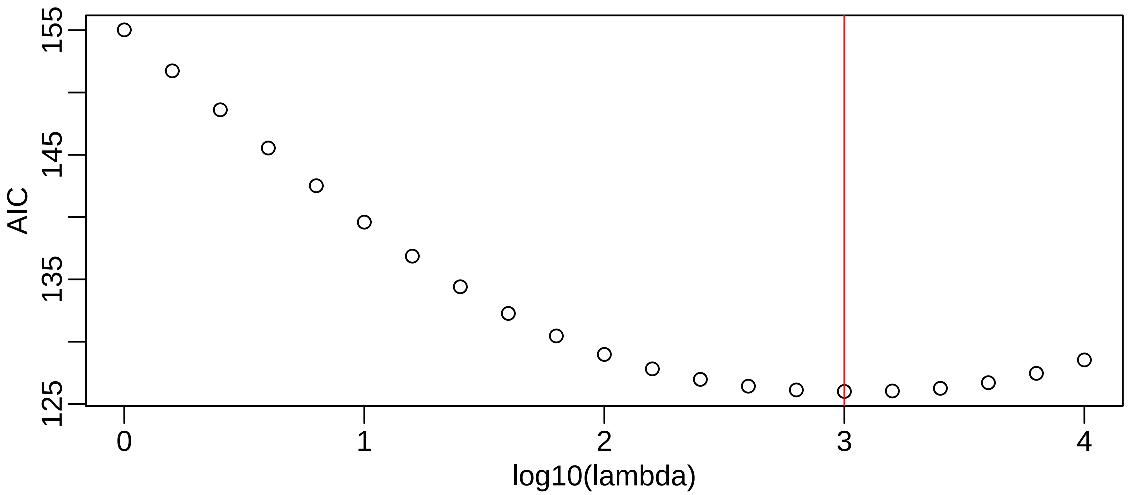


Non-normal data

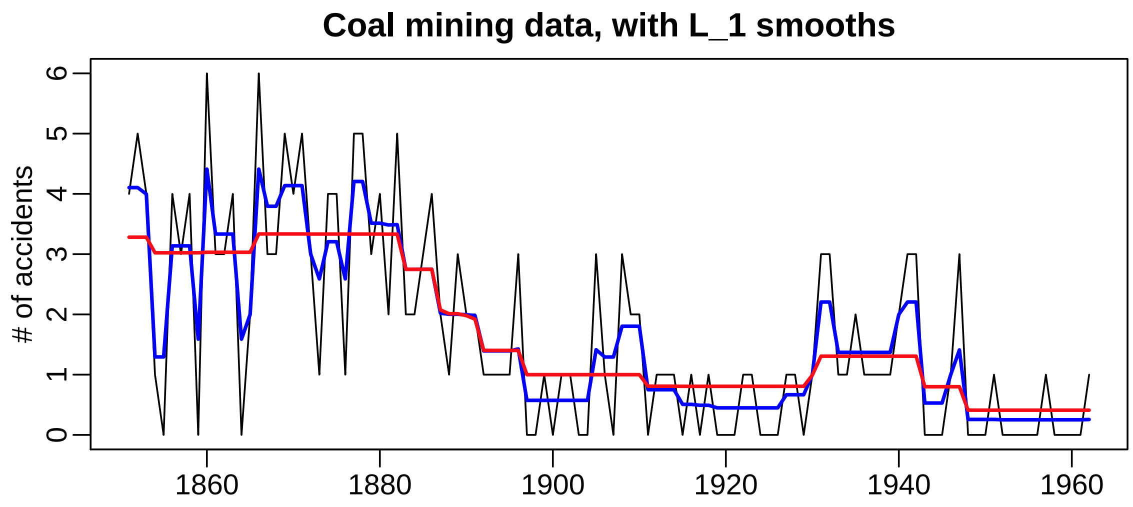
- We are not limited to a sums of squares
- The L_0 penalty can be combined with any log-likelihood
- Example: Poisson distribution
- Observed y, expected values $\mu = e^{\eta}$
- Extend the IWLS algorithm of the GLM with L_0 penalty on $\Delta\eta$
- No technical complications
- Example: the famous coal mining disaster time series

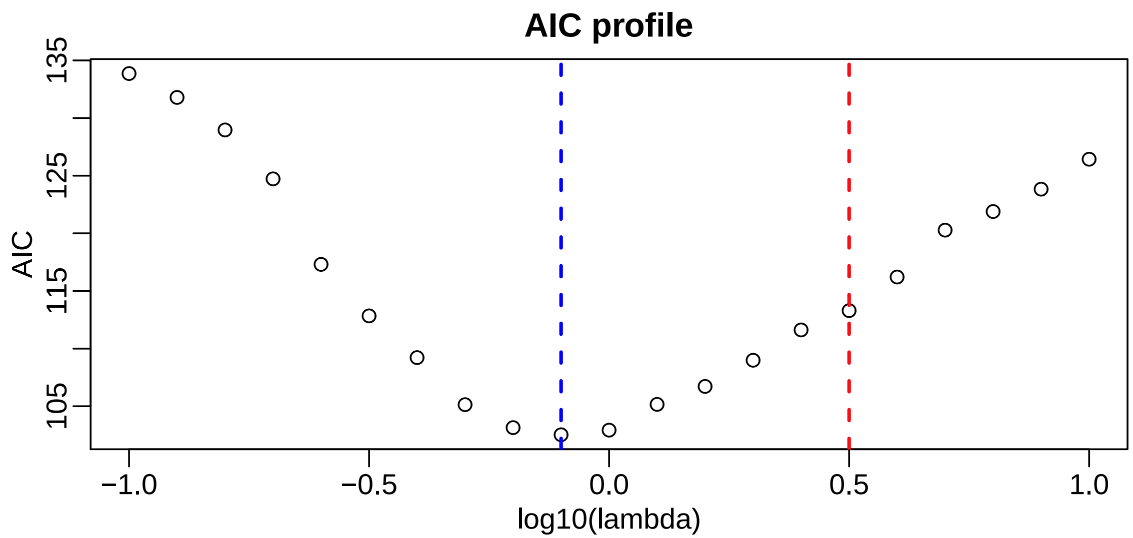
Smoothing of coal mining data with L_2 penalty



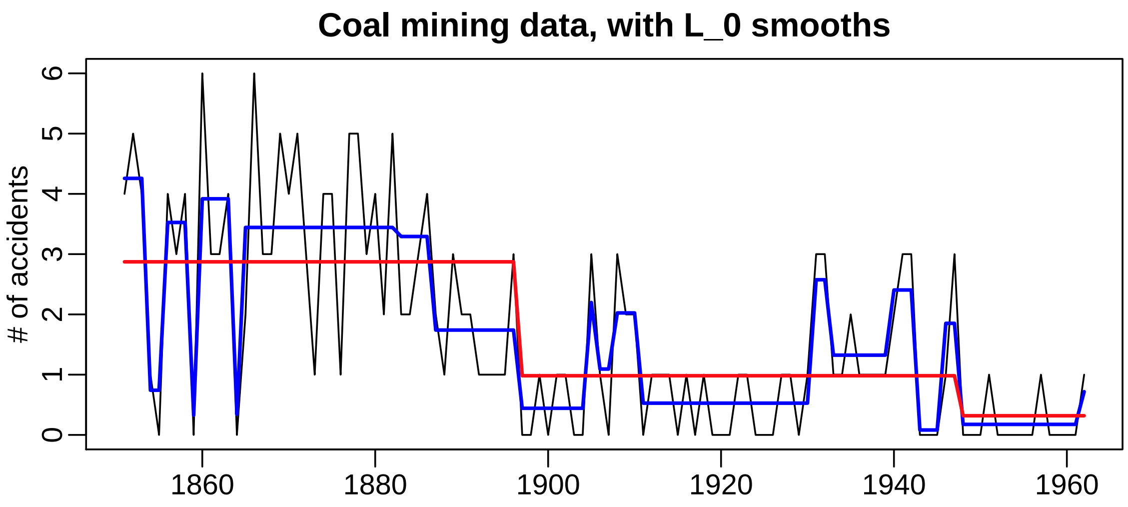


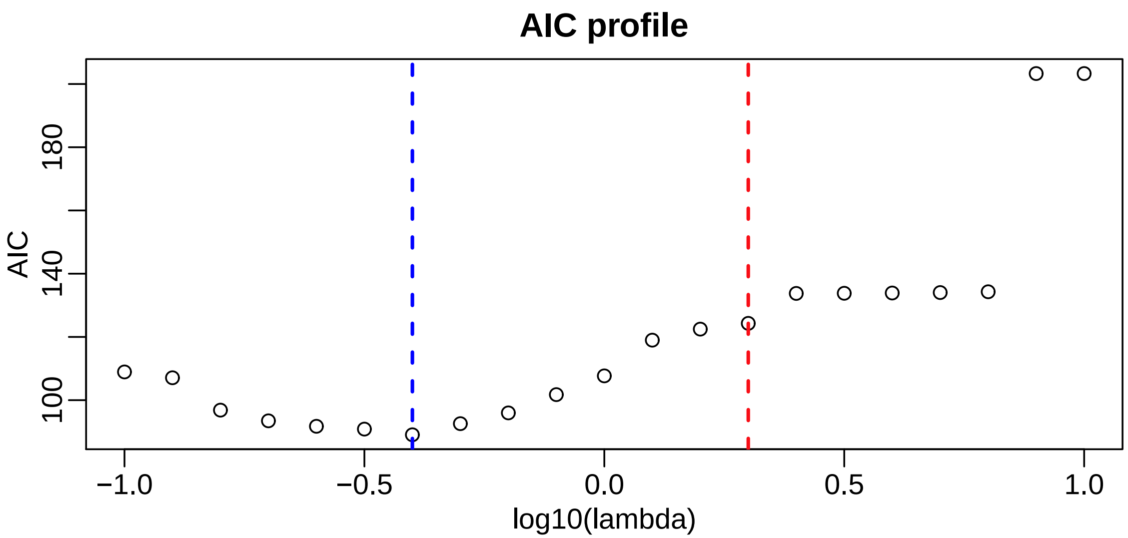
Smoothing of coal mining data with L_1 penalty





Smoothing of coal mining data with L_0 penalty

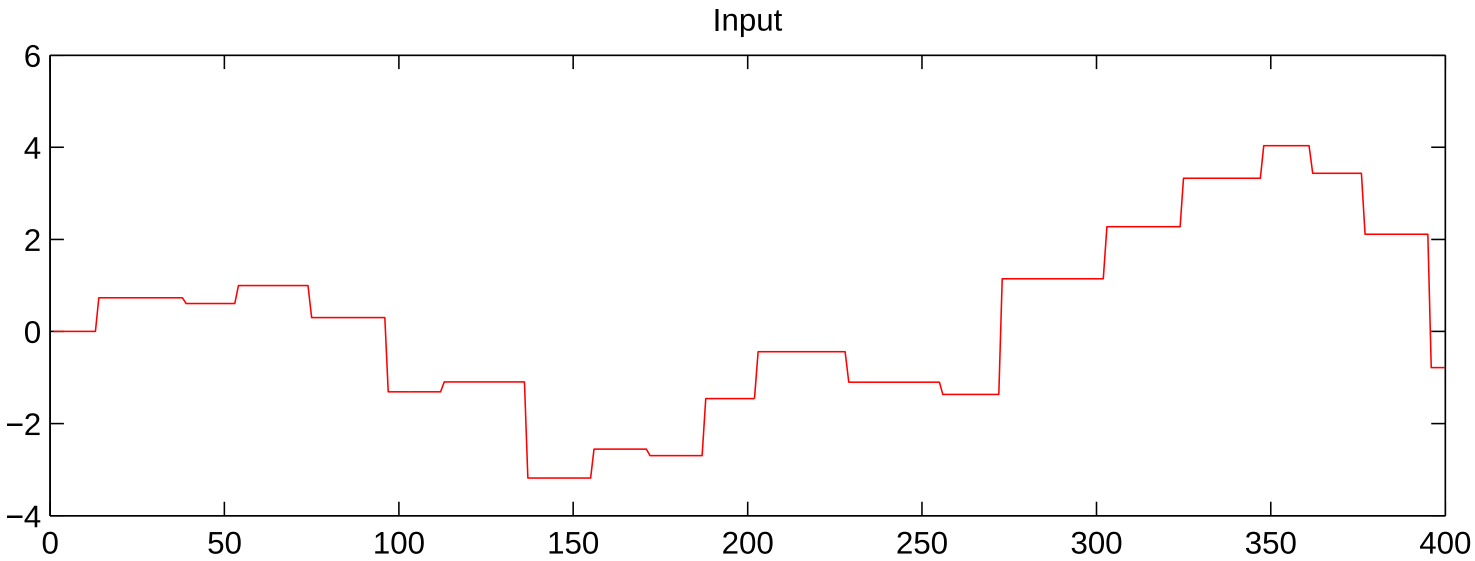


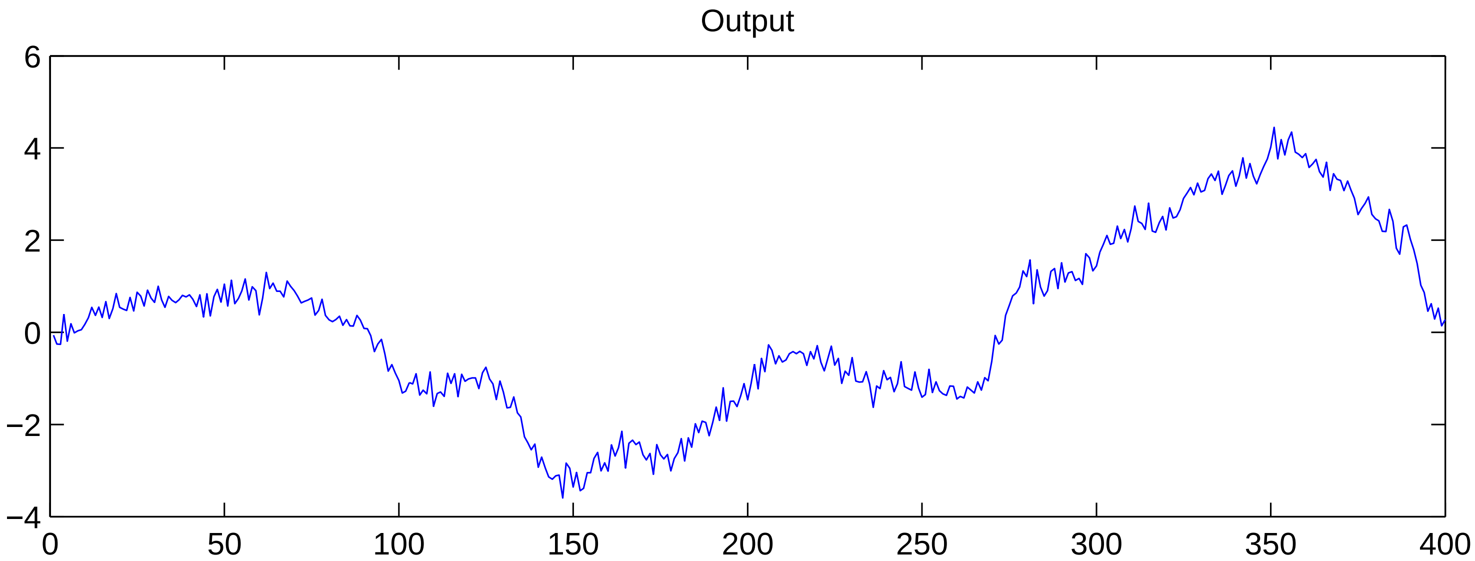


Deconvolution

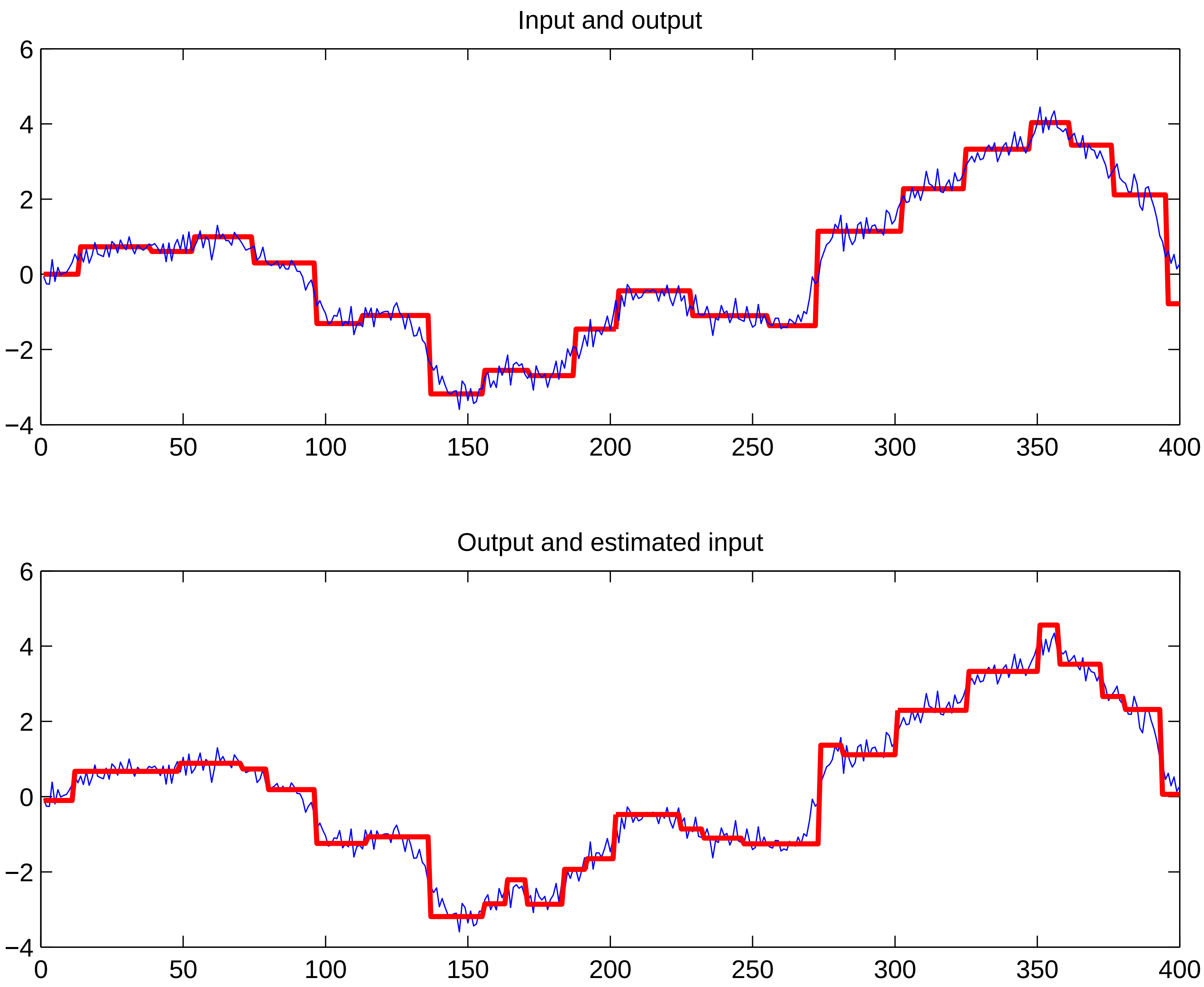
- We observed a "crisp" signal plus noise
- Sometimes the signal has been filtered before
- A crisp input, run through a filter
- So we have penalized deconvolution
- This is not hard to do
- Model: $y = Cx + e$
- Regression with penalty on x

Illustrating convolution with step input





Deconvolution with step input



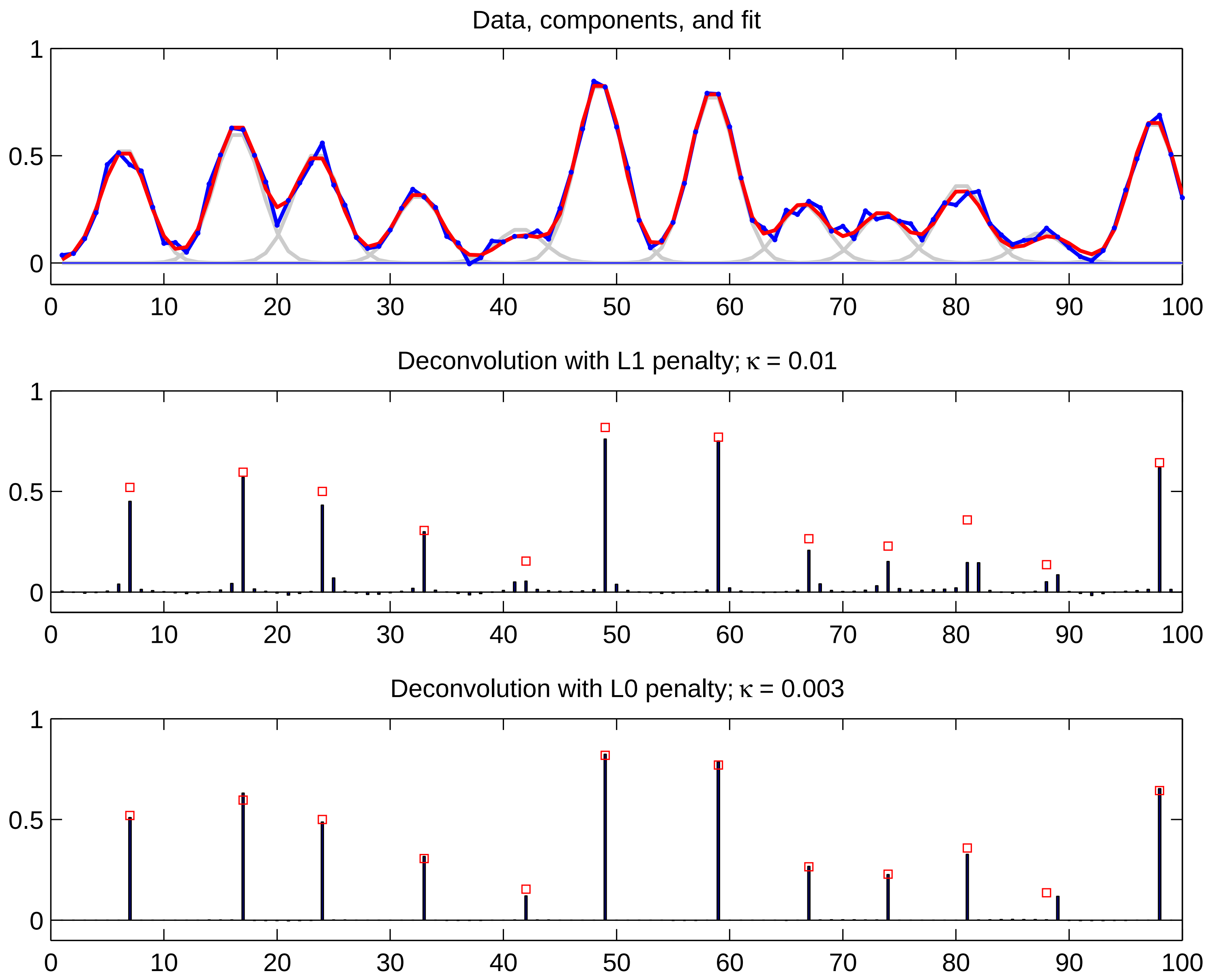
Another application: spike deconvolution

- Many (technical) signals consists of peaks
- Often they can be described by convolution
- Spikes as input, convolution with peaked impulse response
- Penalized regression: minimize

$$S = ||y - Cx|| + \kappa||x||^p$$

- Ridge regression, $p = 2$, is no use
- LASSO, $p = 1$ is an improvement
- But $p = 0$ works best

Spike deconvolution (simulated data)



Summary

- We can implement the L_0 norm as a weighted L_2 norm
- We get surprisingly good results in segmentation and deconvolution
- Non-convex objective function seems no problem in practice
- Fast, sparse, computations, linear in data length
- Easily combined with any likelihood
- SCALA software for CNV smoothing
- And for much more

SCALA software (r.c.a.rippe@lumc.nl)