DATA 501 - Project Presentation

Baseline correction using asymmetrically reweighted penalized least squares smoothing

Corvin Idler

1 **Raman spectra**

some contextual information

2 **Baseline estimation**

some contextual information and methods

3 **painful lesson learnt**

right algorithm is more important than fast language

4 **Details arPLS**

details about the method and the why

5 **Some results**

some pictures of end result

6 **Questions and Demo**

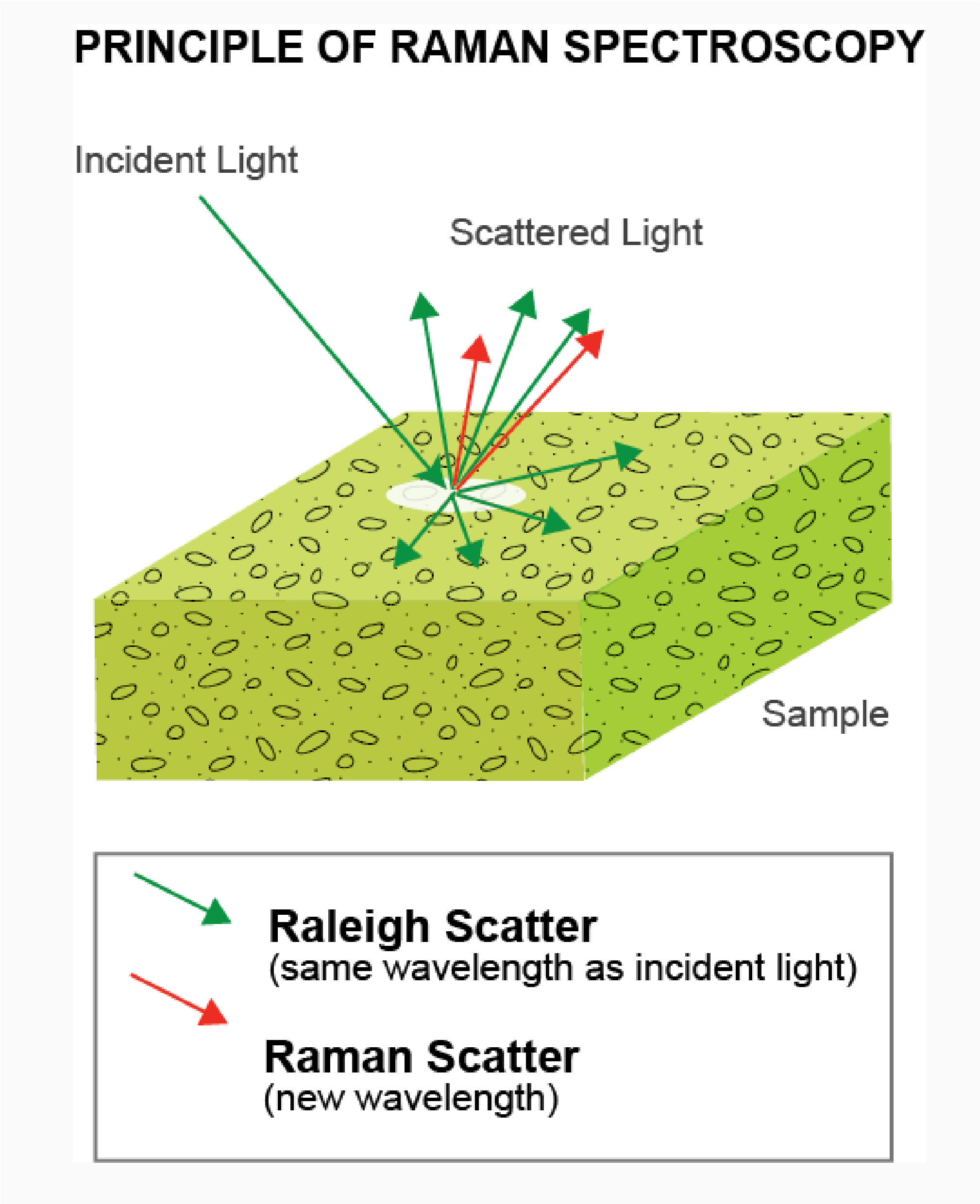
questions and demo

What are we going to talk about

Chandrasekhara Venkata Raman

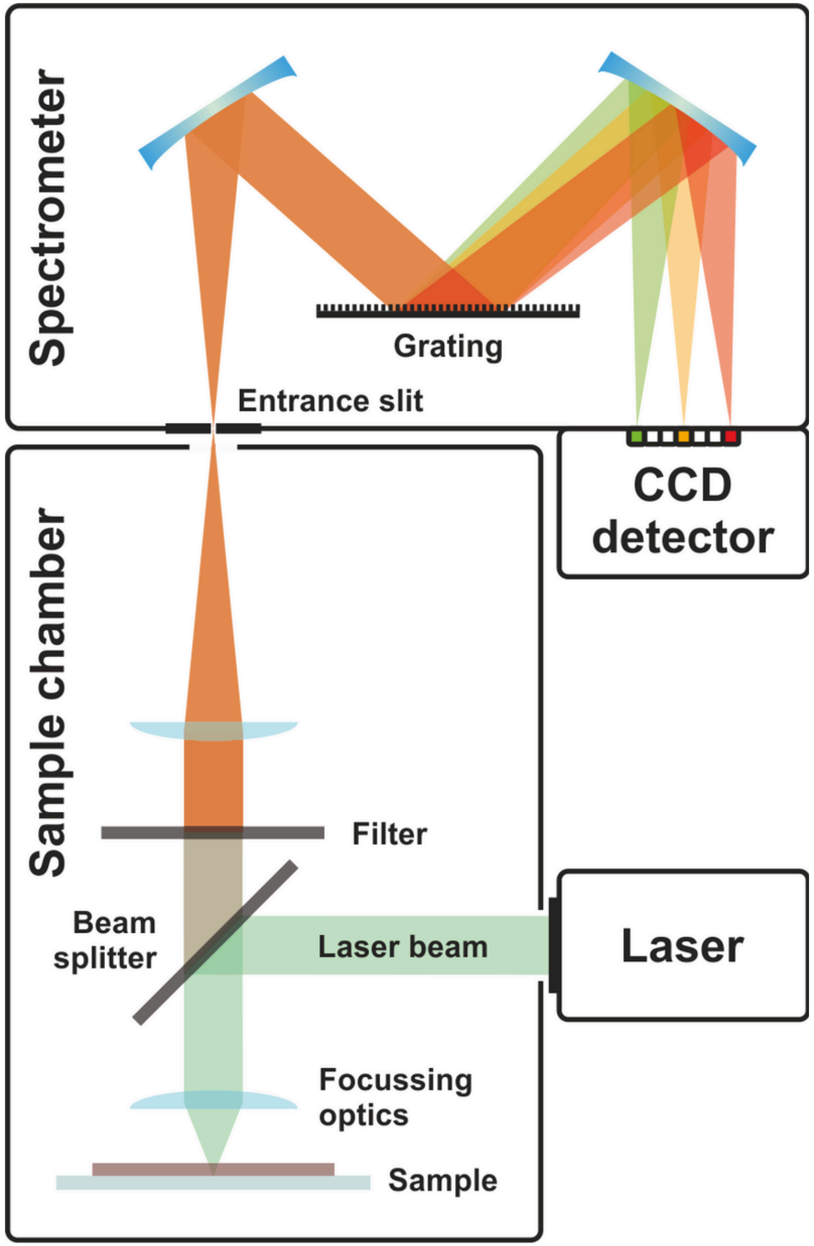
7 November 1888 - 21 November 1970

Discoverred the Raman effect or Raman scattering and received the 1930 Nobel Prize in Physics for this.



"Spontaneous Raman scattering is an inelastic process where incident laser light interacts with molecular vibrations in the sample. […] Raman spectroscopy measures the change in energy of the Raman spectral lines relative to the energy of the excitation laser."

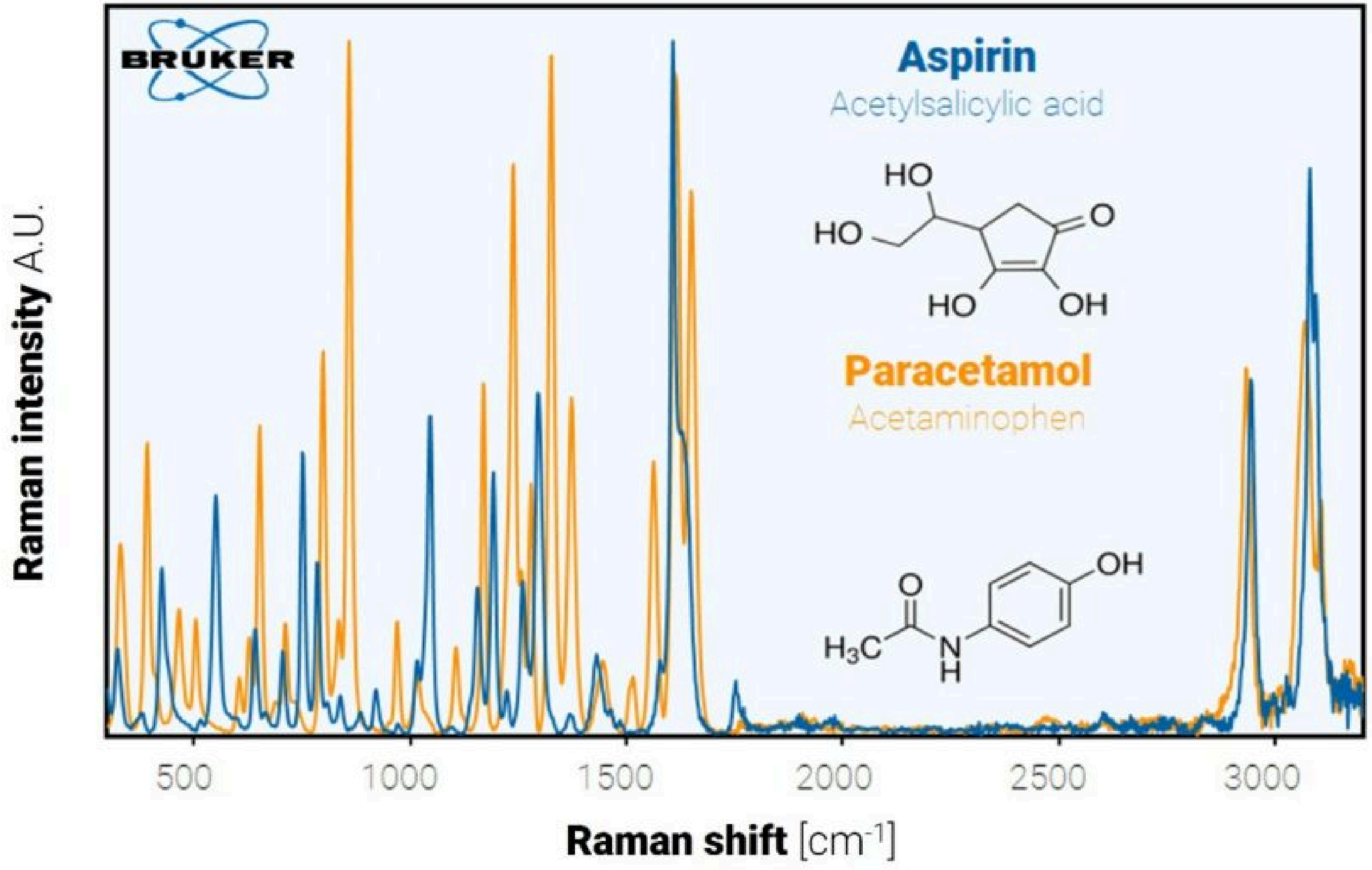
https://www.princetoninstruments.com/learn/raman/overview-raman-spectroscopy-life-sciences



"Raman scattering or the Raman effect is the inelastic scattering of photons by matter, meaning that there is both an exchange of energy and a change in the light's direction."

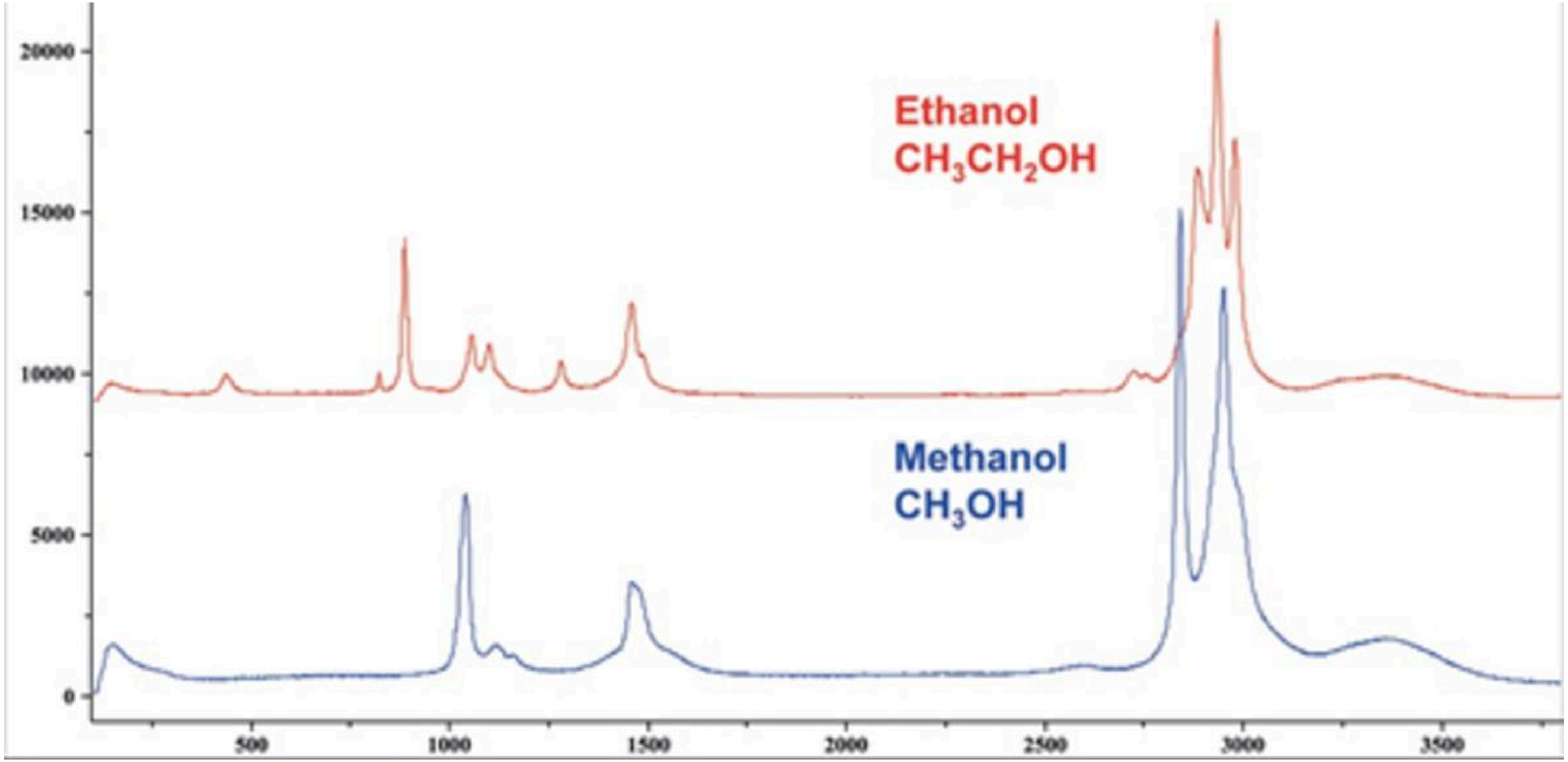
https://en.wikipedia.org/wiki/Raman_scattering

Allows for detection of substances



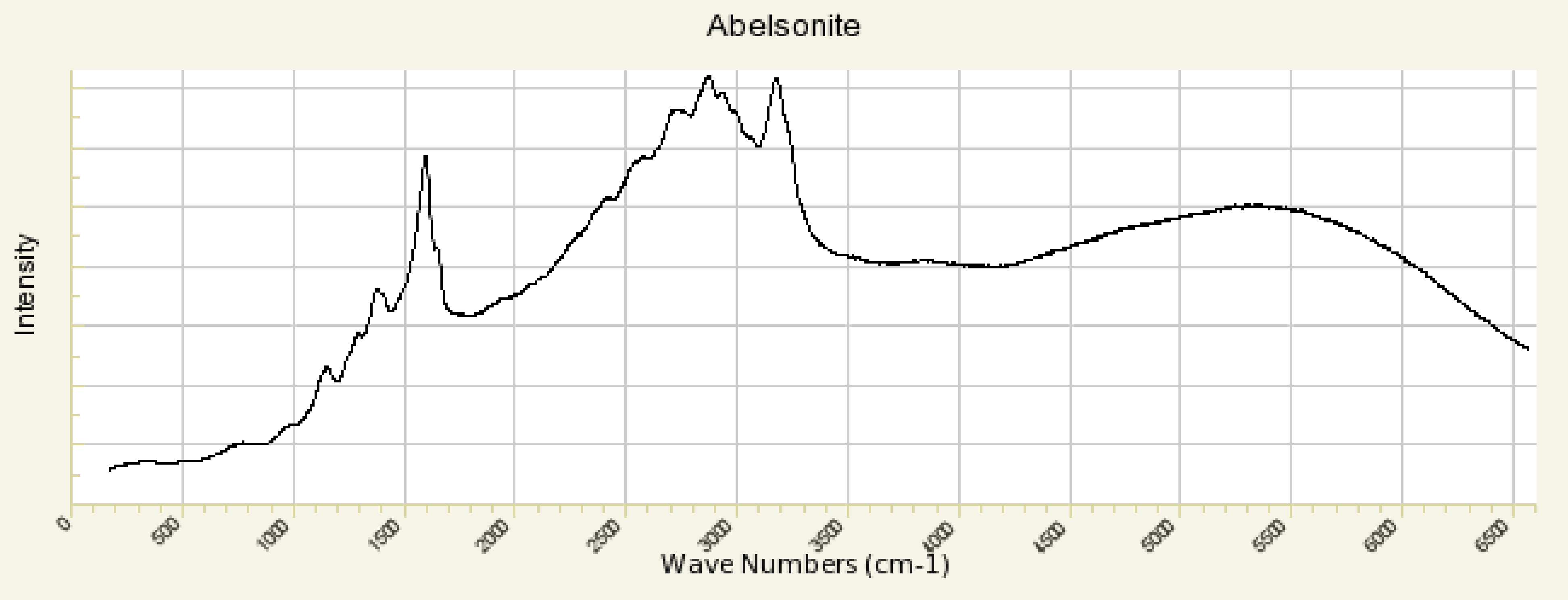
www.bruker.com/en/products-and-solutions/infrared-and-raman/raman-spectrometers/what-is-raman-spectroscopy.html

Allows for detection of substances



https://www.horiba.com/fra/scientific/technologies/raman-imaging-and-spectroscopy/raman-spectroscopy/

What is this baseline story about?



https://rruff.info/abelsonite/display=default/

How to do it?

$$\underbrace{S(\mathbf{z})}_{\text{Smoothed signal=baseline}} = \underbrace{(\mathbf{y}-\mathbf{z})^T\mathbf{W}(\mathbf{y}-\mathbf{z})}_{\substack{\text{weighted squared difference}\\ \text{between baseline and signal}}} + \underbrace{\lambda\mathbf{z}^T\mathbf{D}^T\mathbf{D}\mathbf{z}}_{\substack{\text{smoothness penalty}\\ \text{for baseline}}}$$

$$\frac{\partial S(\mathbf{z})}{\partial \mathbf{z}^T} = 2\mathbf{W}(\mathbf{y}-\mathbf{z}) + 2\lambda\mathbf{D}^T\mathbf{D}\mathbf{z} = 0$$

$$\mathbf{z} = \underbrace{(\mathbf{W}+\lambda\mathbf{D}^T\mathbf{D})^{-1}}_{\text{Matrix inversion :-(}} \mathbf{W}\mathbf{y}$$

Algorithms mainly differ in how W is chosen and how it evolves

Lesson: not all matrices are the same (banded matrix!)

$$(\mathbf{W} + \lambda \mathbf{D}^T \mathbf{D})^{-1}$$

$$D = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \end{bmatrix}$$

$$H = D^T D \begin{bmatrix} 1 & -2 & 1 & 0 & 0 \\ -2 & 5 & -4 & 1 & 0 \\ 1 & -4 & 6 & -4 & 1 \\ 0 & 1 & -4 & 5 & -2 \\ 0 & 0 & 1 & -2 & 1 \end{bmatrix}$$

$$W = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.9 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0.7 \end{bmatrix}$$

basis package matrix inversion/solver
base::solve=32.14445 mins

RcppArmadillo C++ compiled inversion/solver
arma::inv=12.8244 mins

limSolve Fortran compiled inversion/solver
(Linpack) limSolve::Solve.banded=53.76809s

Compiled code beats non-compiled code, but thinking about the problem (banded matrix) and choosing the most appropriate algorithm beats everything else. O(n^3) vs. O(n*k^2) with k=2

Various alternatives

$$\frac{\partial S(\mathbf{z})}{\partial \mathbf{z}^T} = 2\mathbf{W}(\mathbf{y} - \mathbf{z}) + 2\lambda \mathbf{D}^T \mathbf{D}\mathbf{z} = 0$$

$$\mathbf{z} = \underbrace{(\mathbf{W} + \lambda \mathbf{D}^T \mathbf{D})^{-1}}_{\text{Matrix inversion :-(}} \mathbf{W}\mathbf{y}$$

$$\text{AsLS: } w_i = \begin{cases} p & \text{if } y_i > z_i \\ 1 - p & \text{if } y_i \leq z_i \end{cases} \text{ with p e.g. } 0.01$$

$$\text{airPLS: } w_i = \begin{cases} 0 & \text{if } y_i \geq z_i \\ \exp^{t(y_i - zi)/|d|} & \text{if } y_i \leq z_i \end{cases}$$

$$\text{with d = vector where } y_i - z_i \text{ is negative}$$

$$\text{and t is the number of iteration}$$

If peaks were known we would set weights to zero around peaks. But we don't want to do peak finding!

Asymmetric Least Squares Smoothing (AsLS) [Eilers and Boelens, 2005]: Asymmetry parameter p (e.g. 0.001). Focus on where the signal is below the estimated baseline). Two parameters p and lambda and "constant" weights :(

Adaptive iteratively reweighted penalized least squares (airPLS) [Zhang et al., 2010]: Weights dependent on magnitude of difference between signal and baseline where signal below baseline

Goes wrong in both cases with lots of additive noise

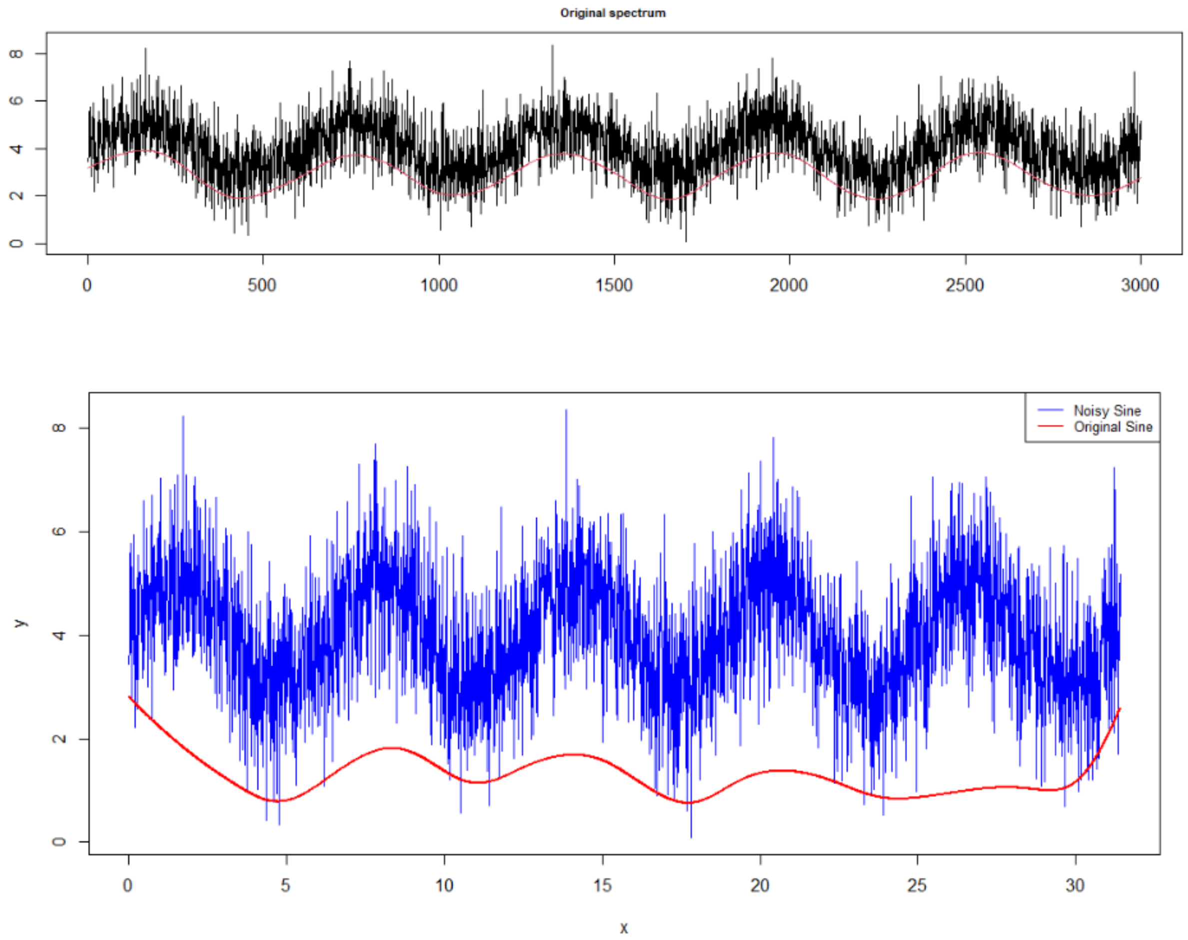
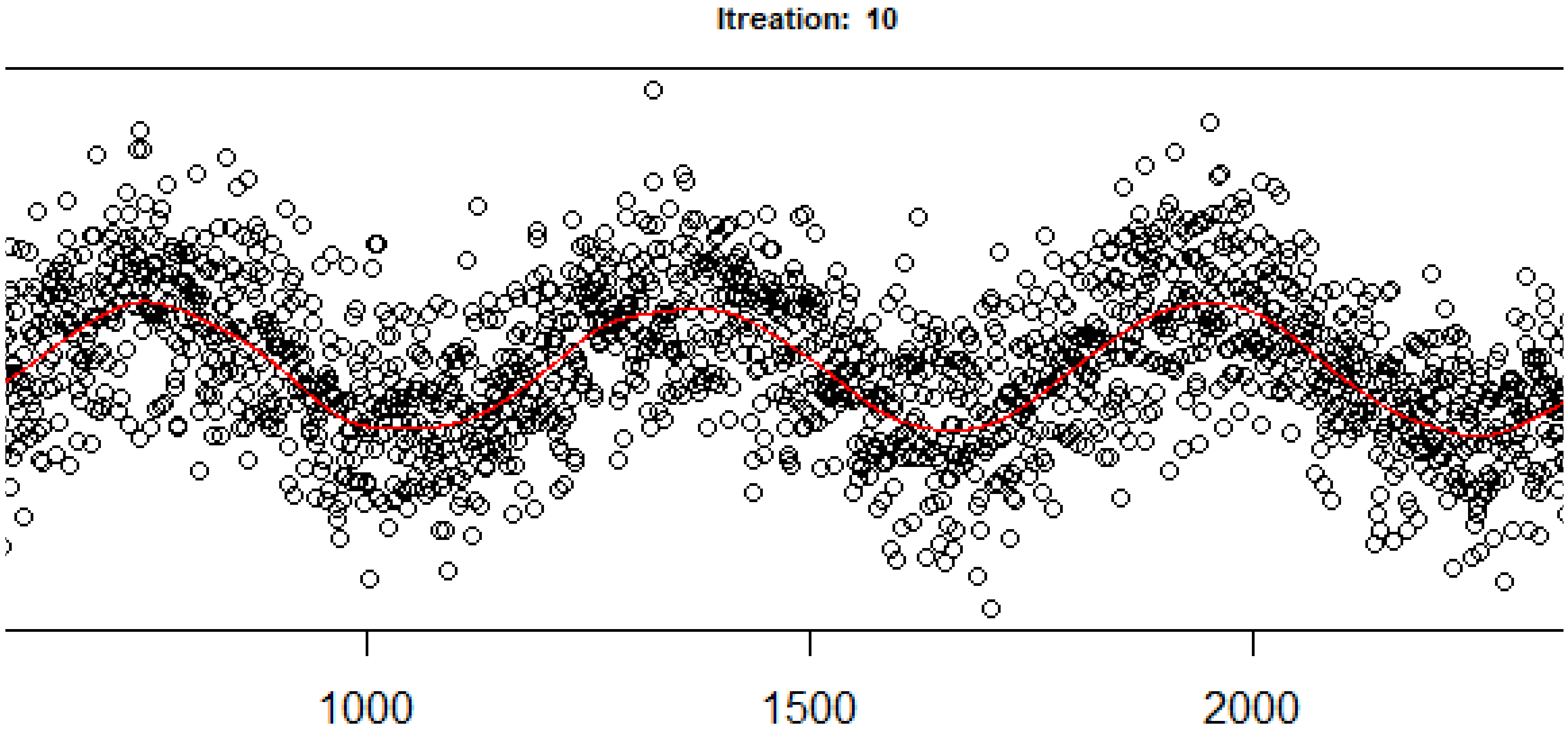


Figure 1: Baseline estimation problems due to additive noise AsLS and airPLS

arPLS to the rescue



Negative values of additive noise isn't corrupting the baseline estimate

arPLS to the rescue

$$\frac{\partial S(\mathbf{z})}{\partial \mathbf{z}^T} = 2\mathbf{W}(\mathbf{y}-\mathbf{z}) + 2\lambda \mathbf{D}^T\mathbf{D}\mathbf{z} = 0$$

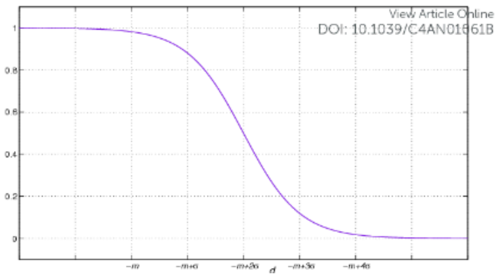
$$\mathbf{z} = \underbrace{(\mathbf{W}+\lambda\mathbf{D}^T\mathbf{D})^{-1}}_{\text{Matrix inversion :-(}} \mathbf{W}\mathbf{y}$$

$$\text{ArPLS: } w_i = \begin{cases} \text{logistic}(y_i - zi, m_{d^-}, \sigma_{d^-}) & \text{if } y_i \geq z_i \\ 1 & \text{if } y_i < z_i \end{cases}$$

with $logistic(d_i, m, \sigma) = (1 + \exp(2(d_i - (-m + 2\sigma))/\sigma))^{-1}$

and d^- the vector of points where y-z is negative.

and m_{d^-} andσ_{d^-} the mean and standard deviation of that vector

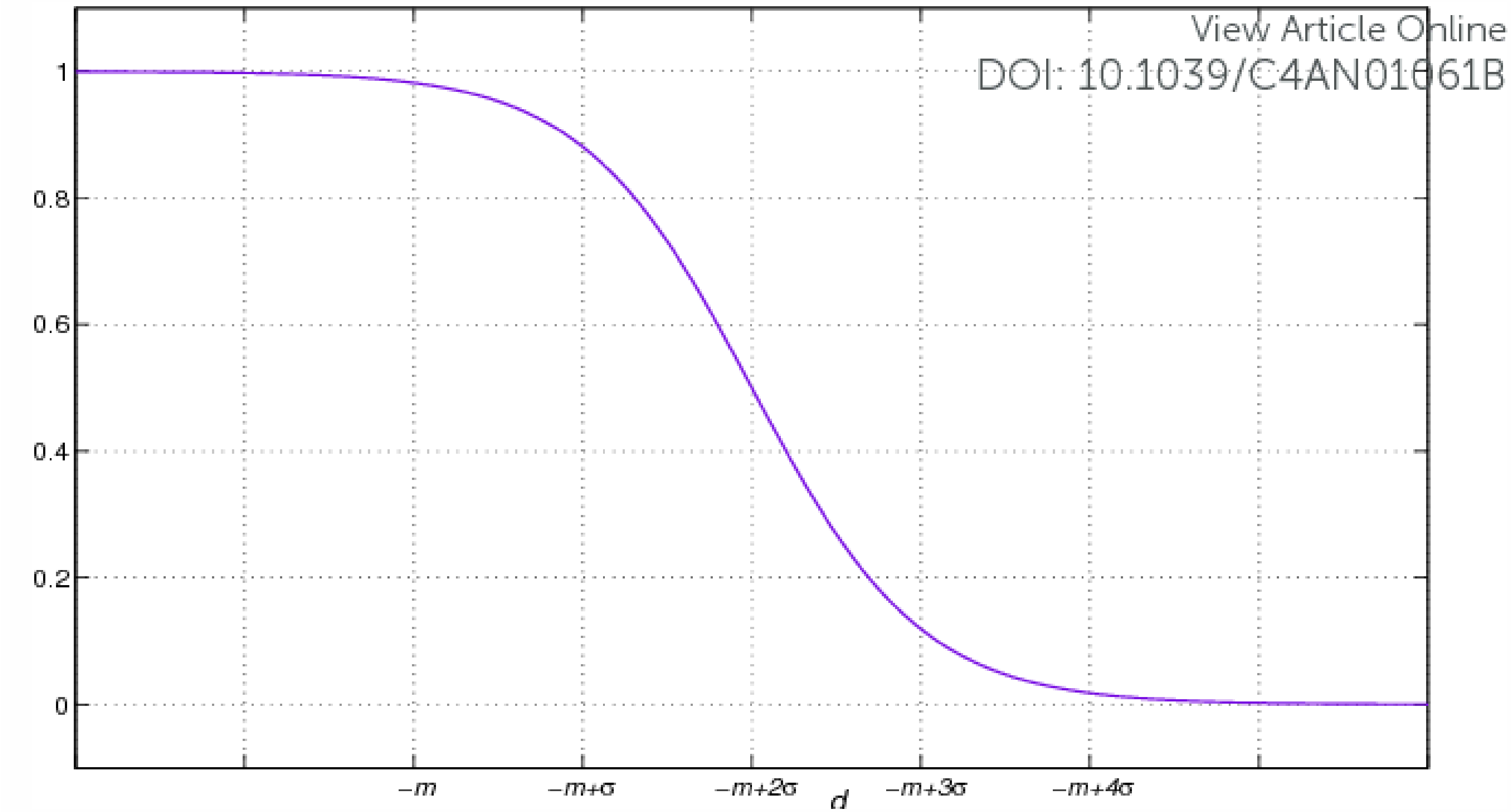


Asymemetrically reweighted penalized least squares (arPLS) [Baek et al., 2015]:

In baseline region without peaks noise is assumed to be equally distributed below and above baseline = similar weights to the signals in that region. (when difference between signal and baseline smaller than estimated noise mean)

If a signal is much greater than the baseline the weight is set to zero (as it is part of a peak)

arPLS to the rescue



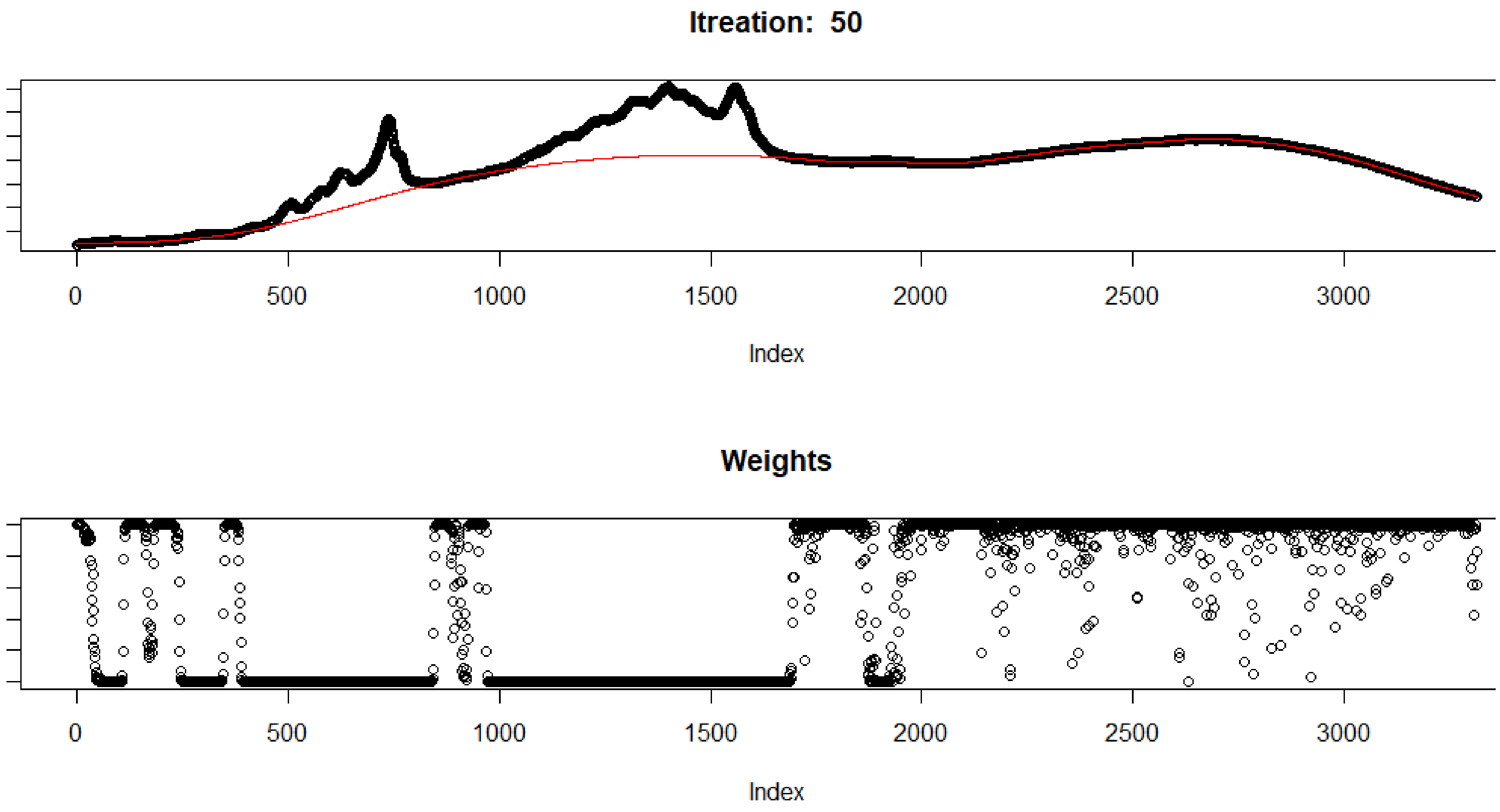
when difference between signal and baseline smaller than estimated noise mean. Once the distance is >4sigma the weights become close to 0

arPLS to the rescue

Data: measured spectrum y, smoothness parameter λ, termination condition *ratio*, fall back termination condition *max_iter*
Result: smoothed baseline z
$\mathbf{H} \leftarrow \lambda \mathbf{D}^T\mathbf{D}$
$w^1 \leftarrow [1, 1, 1, ..., 1]$
for *t=1,2,...,max_iter* **do**
  make a diagonal matrix $\mathbf{W}$ with $W_{i,i} = w_i^t$;
  $\mathbf{z} \leftarrow (\mathbf{W}+\mathbf{H})^{-1}\mathbf{W}\mathbf{y}$;
  $\mathbf{d}^- \leftarrow \mathbf{y} - \mathbf{z}$;
  make $\mathbf{d}^-$ only with $d_i < 0$;
  m = mean of $\mathbf{d}^-$;
  s = standard deviation of $\mathbf{d}^-$;
  for *i=1,2,...,N* **do**
    $w_i^{t+1} = 1/(1 + e^{2(d_i-(-m+2s))/2})$;
  end
  if $\|w^t - w^{t+1}\|/\|w^t| < ratio$ **then**
    break out of the loop and stop;
  end
end

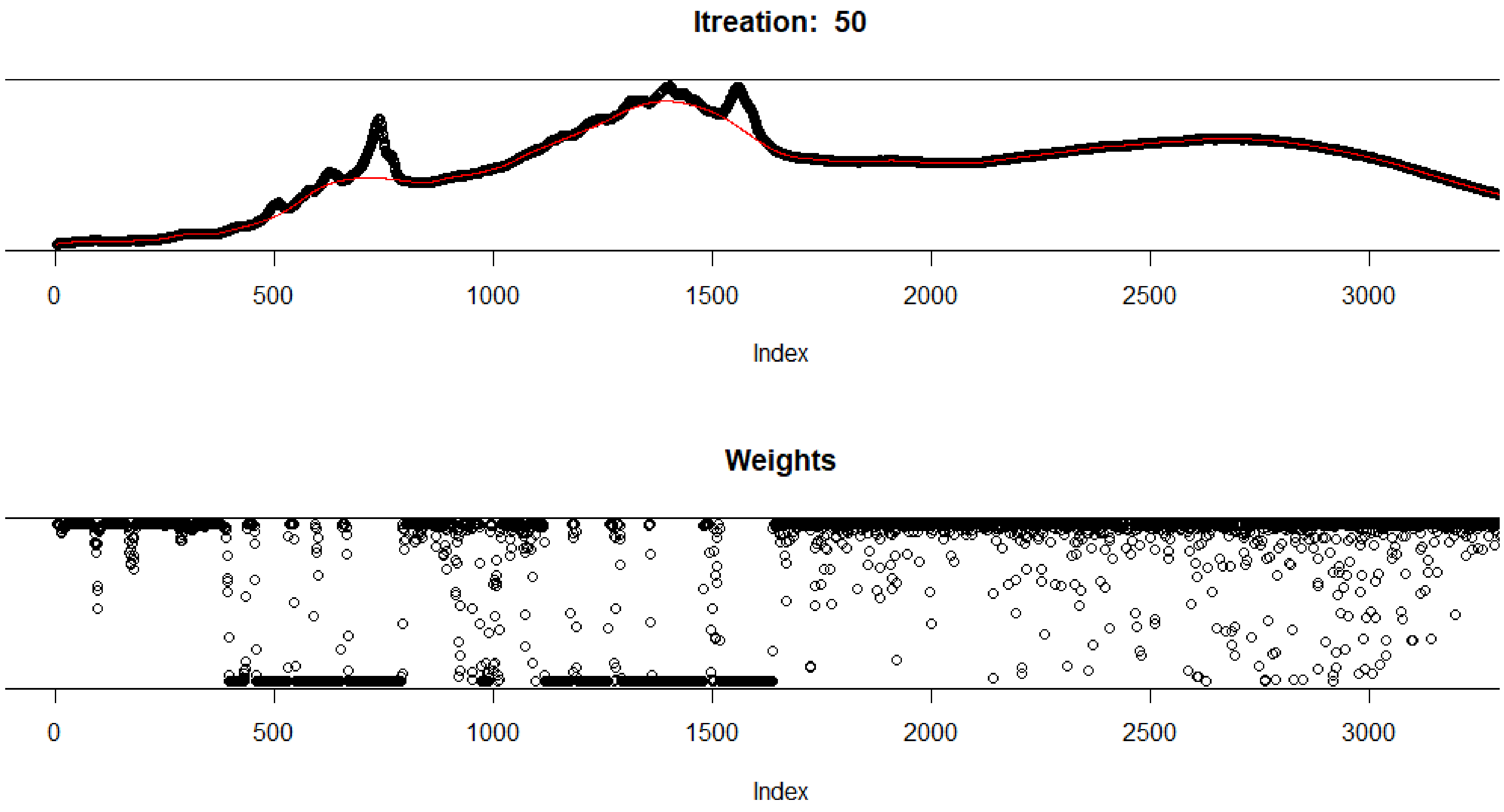
Algorithm 1: Schematics of arPLS algorithm

arPLS to the rescue



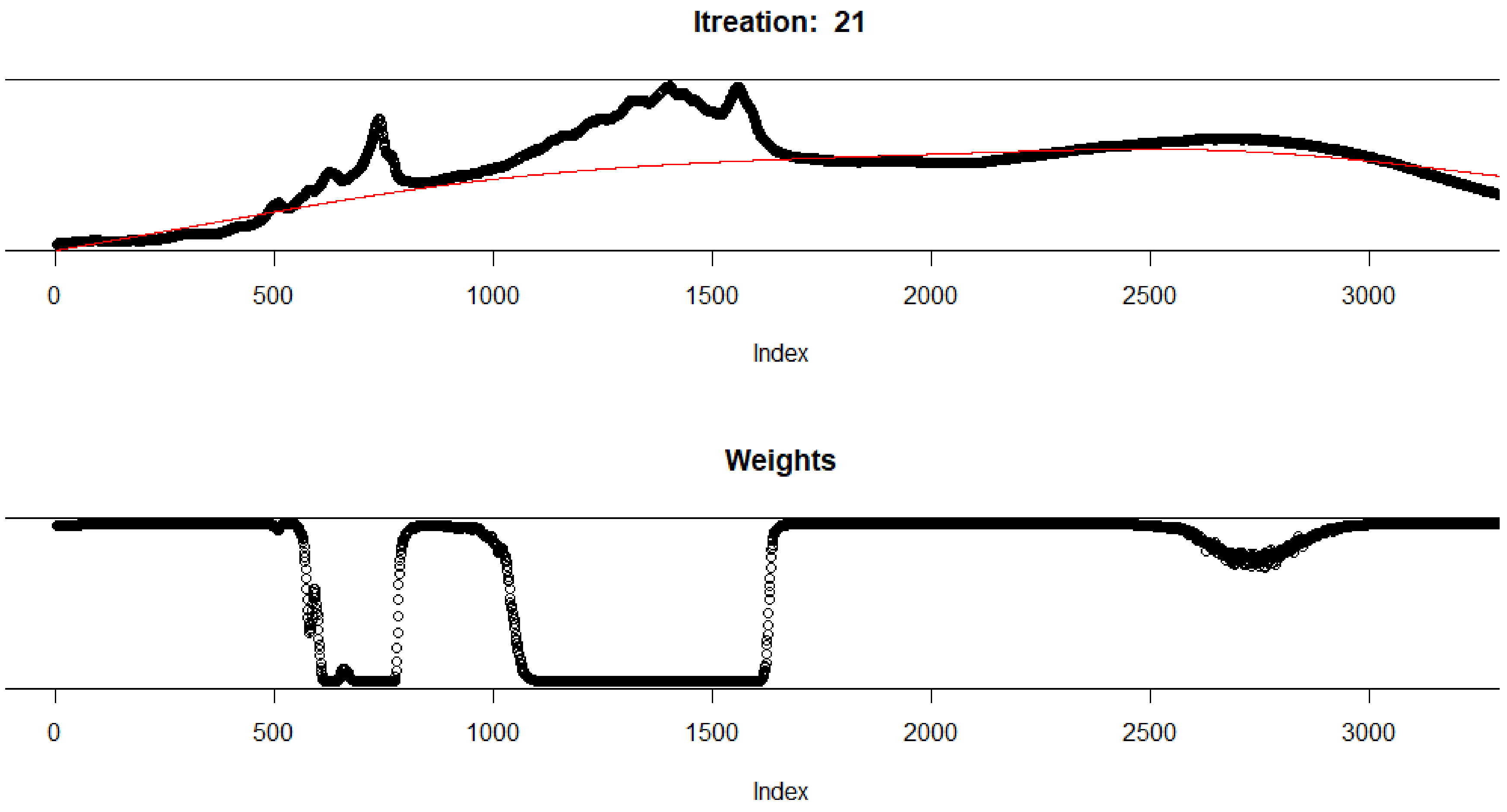
arPLS result after 50 iteration on Abelsonite spectrum, lambda=1e+6

arPLS to the rescue



arPLS result after 50 iteration on Abelsonite spectrum, lambda=1e+4

arPLS to the rescue



arPLS result after 50 iteration on Abelsonite spectrum, lambda=1e+10

demo time