(EELS) Data analysis basics

Francisco de la Peña

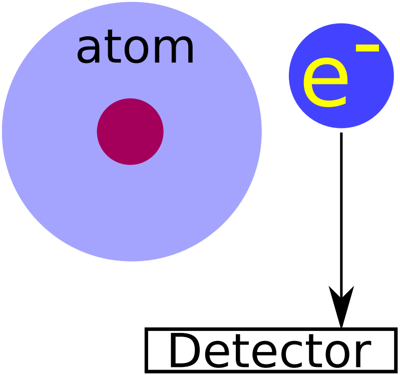
Université de Lille

Diamond Light Source
2nd of March 2020

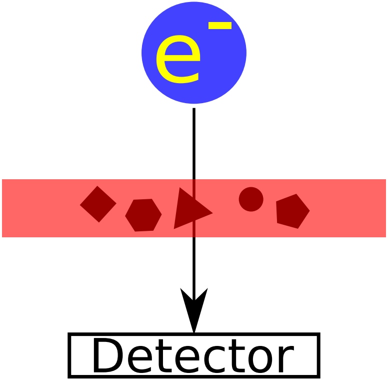
Outline

Outline

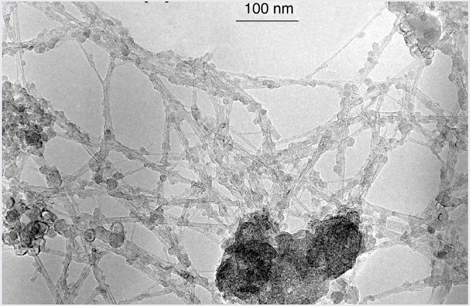
Why do we care about data processing at all?

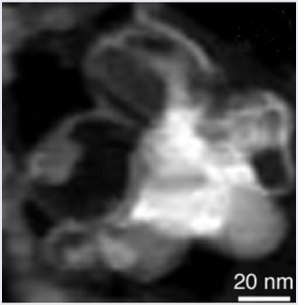


Why do we care about data processing at all?



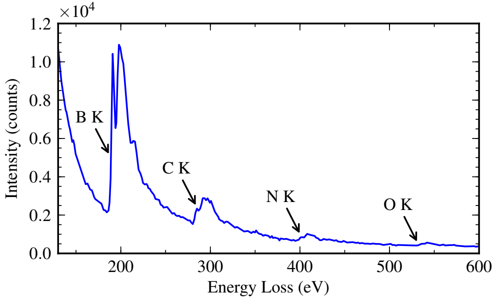
Boron-nitride nano-particles characterisation by EM



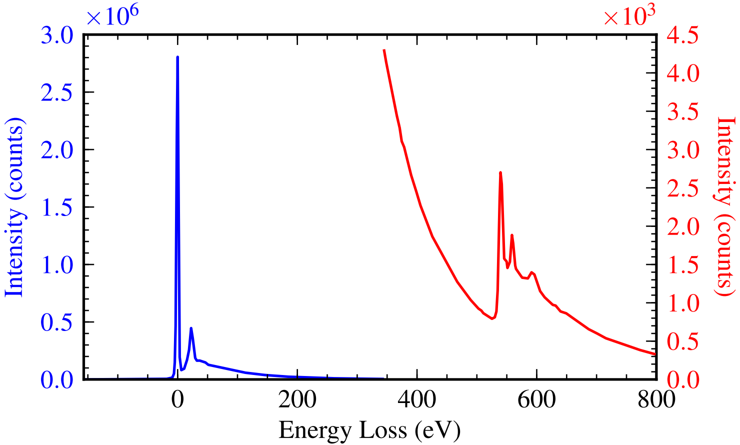


Outline

EELS spectrum from BN NPs

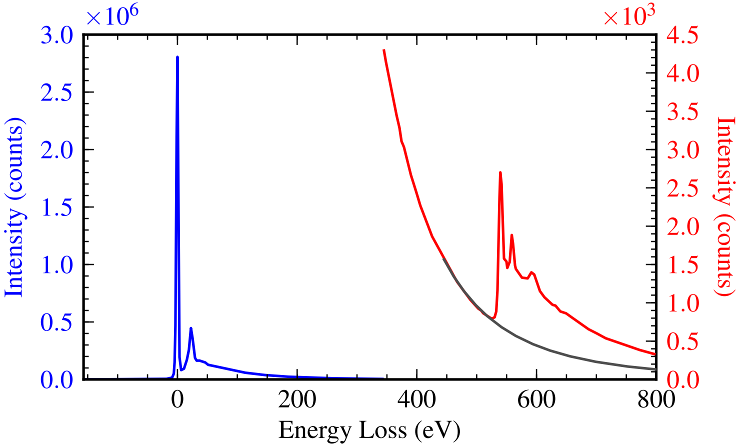


The "windows" method



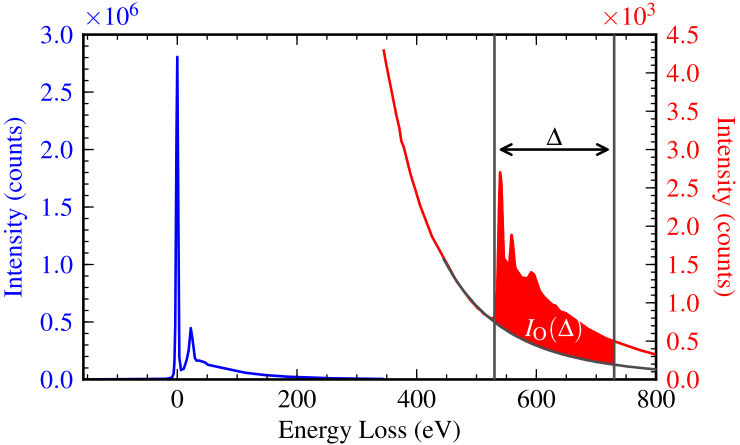
The "windows" method

$$N_{\mathrm{O}} \approx \frac{}{}$$



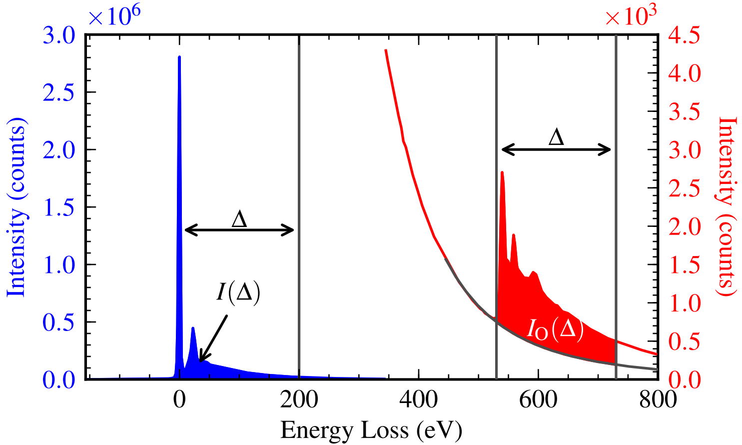
The “windows” method

$$N_{\mathrm{O}} \approx \frac{I_{\mathrm{O}}(\Delta,\beta)}{}$$



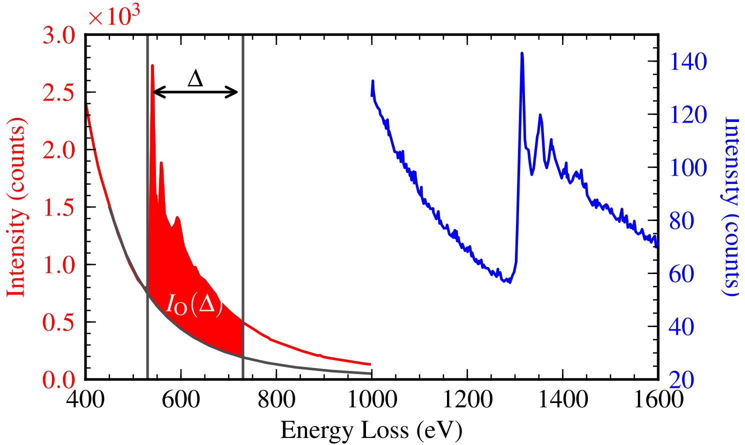
The "windows" method

$$N_{\mathrm{O}} \approx \frac{I_{\mathrm{O}}(\Delta,\beta)}{I(\Delta,\beta)} \sigma_{\mathrm{O}}^{-1}(\Delta,\beta)$$



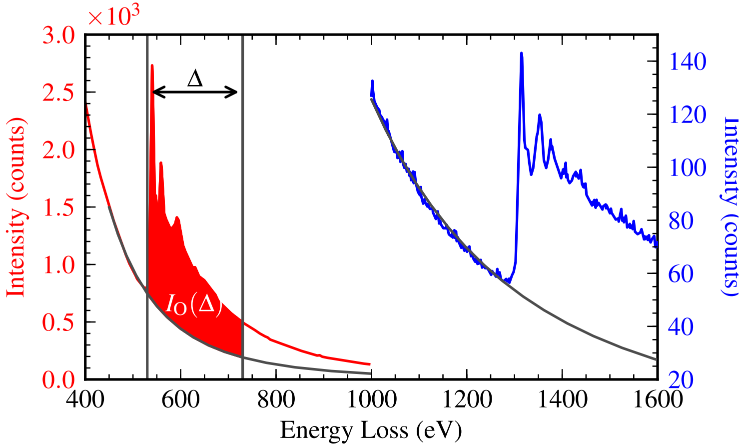
The "windows" method

$$N_{\mathrm{O}} \approx \frac{I_{\mathrm{O}}(\Delta,\beta)}{I(\Delta,\beta)}\sigma_{\mathrm{O}}^{-1}(\Delta,\beta) \qquad N_{\mathrm{Mg}} \approx \frac{I_{\mathrm{Mg}}(\Delta,\beta)}{I(\Delta,\beta)}\sigma_{\mathrm{Mg}}^{-1}(\Delta,\beta)$$



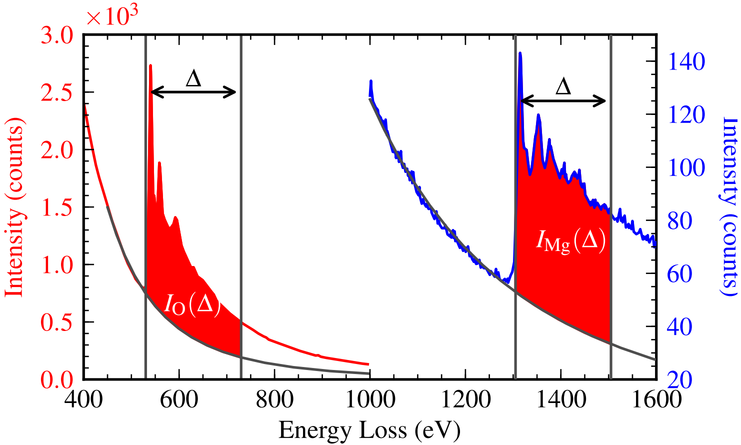
The "windows" method

$$N_O \approx \frac{I_O(\Delta,\beta)}{I(\Delta,\beta)} \sigma_O^{-1}(\Delta,\beta) \qquad N_{Mg} \approx \frac{I_{Mg}(\Delta,\beta)}{I(\Delta,\beta)} \sigma_{Mg}^{-1}(\Delta,\beta)$$



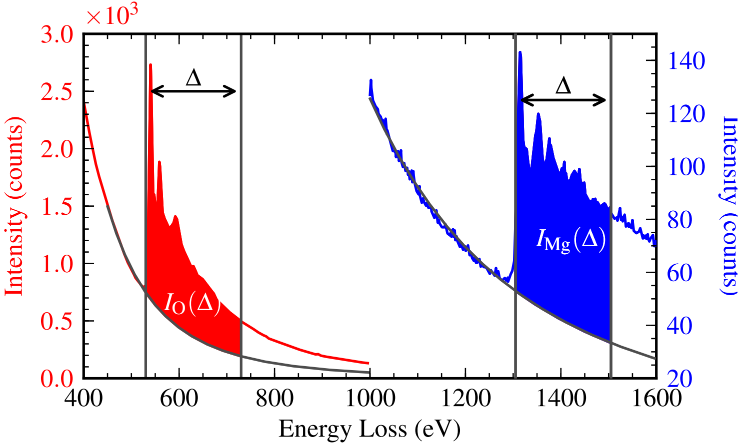
The "windows" method

$$N_{\mathrm{O}} \approx \frac{I_{\mathrm{O}}(\Delta,\beta)}{I(\Delta,\beta)}\sigma_{\mathrm{O}}^{-1}(\Delta,\beta) \qquad N_{\mathrm{Mg}} \approx \frac{I_{\mathrm{Mg}}(\Delta,\beta)}{I(\Delta,\beta)}\sigma_{\mathrm{Mg}}^{-1}(\Delta,\beta)$$



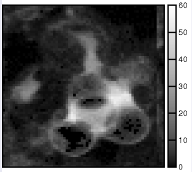
The "windows" method

$$\frac{N_{\mathrm{O}}}{N_{\mathrm{Mg}}} \approx \frac{I_{\mathrm{O}}(\Delta,\beta)}{I_{\mathrm{Mg}}(\Delta,\beta)} \frac{\sigma_{\mathrm{Mg}}(\Delta,\beta)}{\sigma_{\mathrm{O}}(\Delta,\beta)}$$

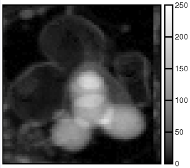


EELS elemental of BN nanoparticle

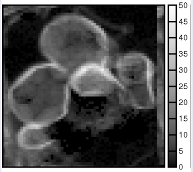
Oxygen



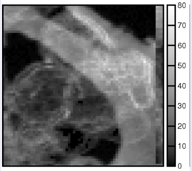
Boron



Nitrogen



Carbon



Arenal et al., Ultramicroscopy 2008

Some limitations of the "windows" method

- Overlapping edges

Some limitations of the “windows” method

- Overlapping edges
- It always returns a result (what feels good) but, how do we know that it is correct?

Some limitations of the "windows" method

- Overlapping edges
- It always returns a result (what feels good) but,
  how do we know that it is correct?
- Only analyses a fraction of the available signal (non-optimal SNR)

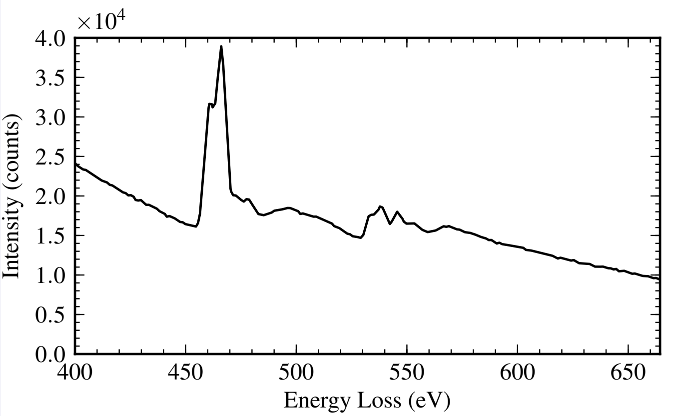
Some limitations of the "windows" method

- Overlapping edges
- It always returns a result (what feels good) but,
  how do we know that it is correct?
- Only analyses a fraction of the available signal (non-optimal SNR)
- Useful information gets lost (fine structures changes, energy onset shifts...)

Outline

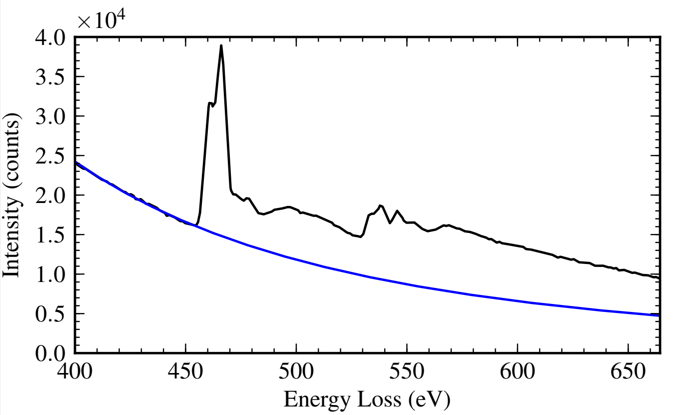
The curve fitting method: an example

$SrTiO_3$ Spectrum



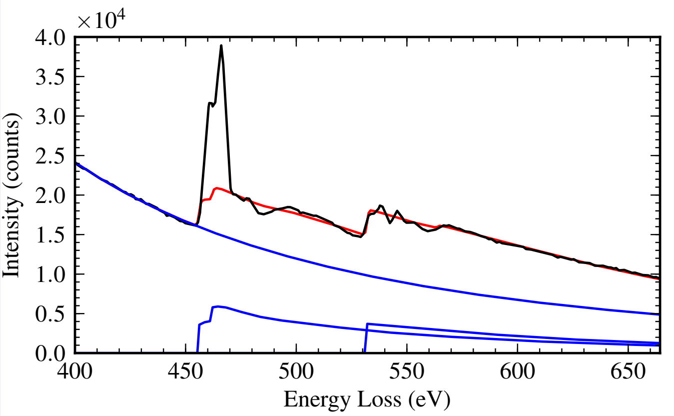
The curve fitting method: an example

$M(E) = AE^{-r}$



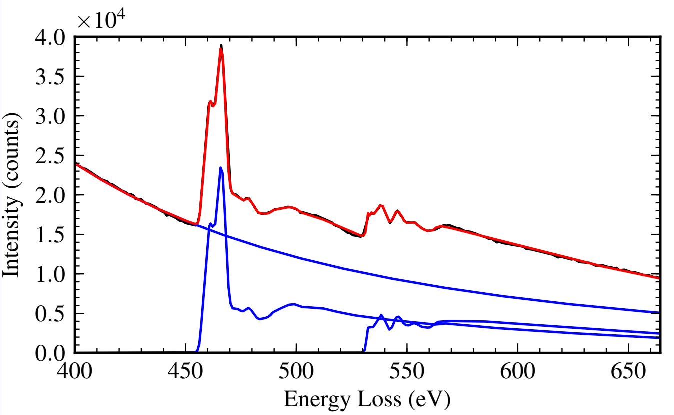
The curve fitting method: an example

$M(E) = AE^{-r} + I_{\mathrm{Ti}}\sigma_{\mathrm{Ti}}(E) + I_{\mathrm{O}}\sigma_{\mathrm{O}}(E)) * L(E)$



The curve fitting method: an example

$$M(E) = AE^{-r} + (N_{\mathrm{Ti}} f_{\mathrm{Ti}}(E)\sigma_{\mathrm{Ti}}(E) + N_{\mathrm{O}} f_{\mathrm{O}}(E)\sigma_{\mathrm{O}}(E)) * L(E)$$



Assumptions

- There is a *known* function, f, that relates the *independent variable* X and the *dependent variable* Y. $Y \approx f(X,\beta) + \varepsilon(f(X,\beta))$

Assumptions

- There is a *known* function, f, that relates the *independent variable* X and the *dependent variable* Y. $Y \approx f(X,\beta) + \varepsilon(f(X,\beta))$
- The number of unknown parameters, β is *equal or less* thant the number of different observations of the independent variable

Assumptions

- There is a *known* function, f, that relates the *independent variable* X and the *dependent variable* Y. $Y \approx f(X,\beta) + \varepsilon(f(X,\beta))$
- The number of unknown parameters, β is *equal or less* thant the number of different observations of the independent variable
- The probability distribution of the statistical error (ε) is known

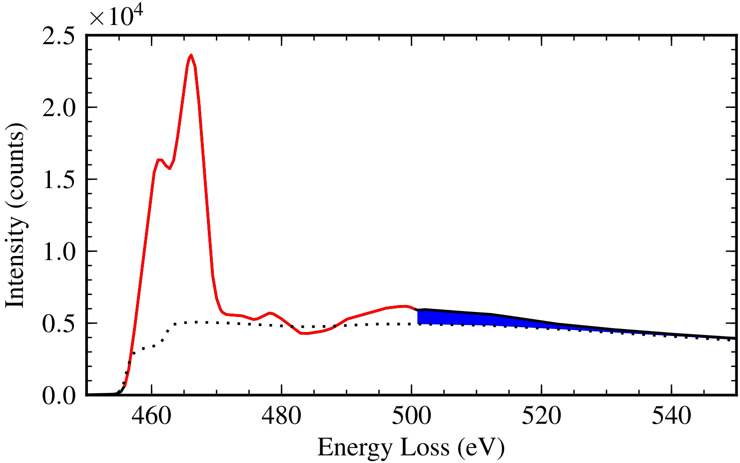
Components of the model

Parametric model of the high energy loss spectrum for elemental and bonding quantification:

$$M(E;\text{parameters}) = AE^{-r} + \left(\sum_i N_i f_i(E) \int_0^{q(\beta)} \sigma_i(E,q)\, dq \right) * L(E)$$

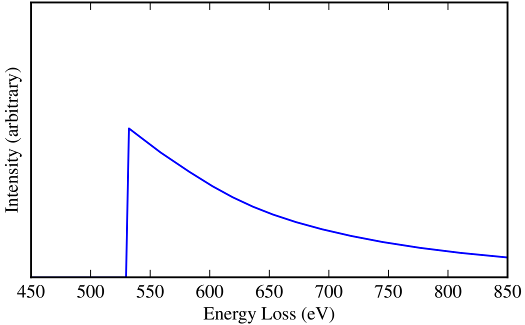
- AE^{-r}: background model
- σ_i^{FS}: cross section of each ionization edge, i
- N_i: $^{\text{atoms}}/_{\text{nm}^2}$
- $f_i(E)$: function that mimics the fine structure of each ionization edge, e.g. gaussian, fingerprints, splines...
- $L(E)$: experimental low loss spectrum.

Why adding the fine structure to the model?



Ionization edge fine structure

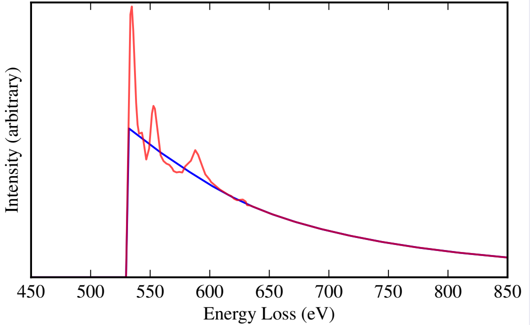
Oxygen K ionization edge from MgO



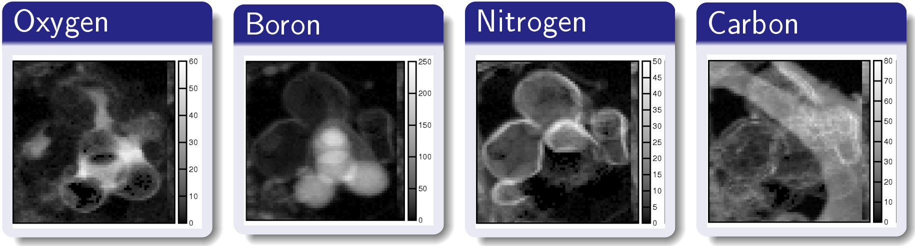
Ionization edge fine structure

- In solids, the first $\sim 40\,\mathrm{eV}$ are strongly modified by the final density of states $\Rightarrow$ carries bonding information

Oxygen K ionization edge from MgO



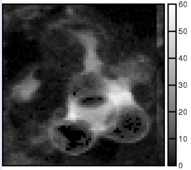
EELS elemental and bonding maps of BN nanoparticle



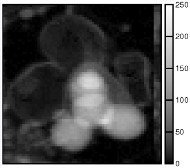
Arenal et al., Ultramicroscopy 2008

EELS elemental and bonding maps of BN nanoparticle

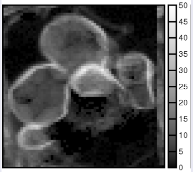
Oxygen



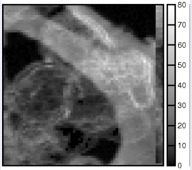
Boron



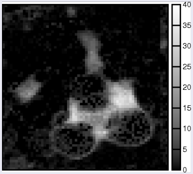
Nitrogen



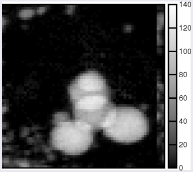
Carbon



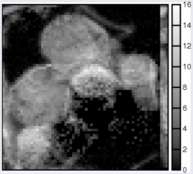
Boron oxide



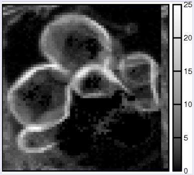
Boron pure



BN $\perp$



BN $||$



Arenal et al., Ultramicroscopy 2008

Parameter value and error estimation

- Regression analysis, in addition to estimate the value of β, can estimate the error

Parameter value and error estimation

- Regression analysis, in addition to estimate the value of β, can estimate the error
- The most common estimation method is non-linear least squares *(NLLS)*, however this *assumes that the noise is Gaussian distributed*

Parameter value and error estimation

- Regression analysis, in addition to estimate the value of β, can estimate the error
- The most common estimation method is non-linear least squares *(NLLS)*, however this *assumes that the noise is Gaussian distributed*
- If the noise distribution is known (e.g. Poissonian) we can use:

Parameter value and error estimation

- Regression analysis, in addition to estimate the value of β, can estimate the error
- The most common estimation method is non-linear least squares *(NLLS)*, however this *assumes that the noise is Gaussian distributed*
- If the noise distribution is known (e.g. Poissonian) we can use:
    - Weighted non-linear least squares (WNNLS)

Parameter value and error estimation

- Regression analysis, in addition to estimate the value of β, can estimate the error
- The most common estimation method is non-linear least squares *(NLLS)*, however this *assumes that the noise is Gaussian distributed*
- If the noise distribution is known (e.g. Poissonian) we can use:
    - Weighted non-linear least squares (WNNLS)
    - Maximum likelihood estimation (ML)

To keep in mind

- The estimation of the parameters value and error *will be wrong* if the noise probability distribution is wrong

To keep in mind

- The estimation of the parameters value and error *will be wrong* if the noise probability distribution is wrong
- In EELS the noise is a mixture of *Poisson and Gaussian noise.*

To keep in mind

- The estimation of the parameters value and error *will be wrong* if the noise probability distribution is wrong
- In EELS the noise is a mixture of *Poisson and Gaussian noise.*
- WNNLS can approximate well Poissonian noise when the number of counts is high enough (almost always in EELS)

To keep in mind

- The estimation of the parameters value and error *will be wrong* if the noise probability distribution is wrong
- In EELS the noise is a mixture of *Poisson and Gaussian noise.*
- WNNLS can approximate well Poissonian noise when the number of counts is high enough (almost always in EELS)
- Non-linear parameter estimation is an iterative process that *is very sensitive to the starting parameters*

Key articles

- Steele, J., Titchmarsh, J., Chapman, J., and Paterson, J. (1985). A single-stage process for quantifying electron energy-loss spectra. Ultramicroscopy, 17(3):273–276.

Key articles

- Steele, J., Titchmarsh, J., Chapman, J., and Paterson, J. (1985). A single-stage process for quantifying electron energy-loss spectra. Ultramicroscopy, 17(3):273–276.
- Manoubi, T., Tencé, M., Walls, M. G., and Colliex, C. (1990). Curve fitting methods for quantitative analysis in electron energy loss spectroscopy. Microscopy Microanalysis Microstructures, 1(1):23.

Key articles

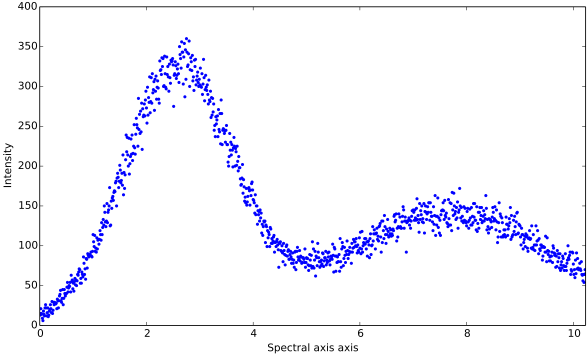
- Steele, J., Titchmarsh, J., Chapman, J., and Paterson, J. (1985). A single-stage process for quantifying electron energy-loss spectra. Ultramicroscopy, 17(3):273–276.
- Manoubi, T., Tencé, M., Walls, M. G., and Colliex, C. (1990). Curve fitting methods for quantitative analysis in electron energy loss spectroscopy. Microscopy Microanalysis Microstructures, 1(1):23.
- Verbeeck, J. and Aert, S. V. (2004). Model based quantification of EELS spectra. Ultramicroscopy, 101(2-4):207–224.

Software

- EELSModel http://www.eelsmodel.ua.ac.be/ (open source)
- HyperSpy http://hyperspy.org (open source)
- Digital Micrograph

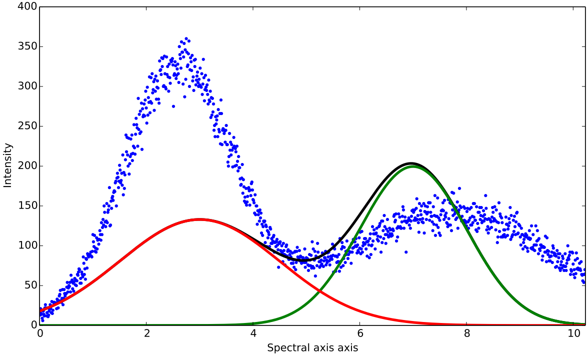
Outline

Non-linear optimisation routine



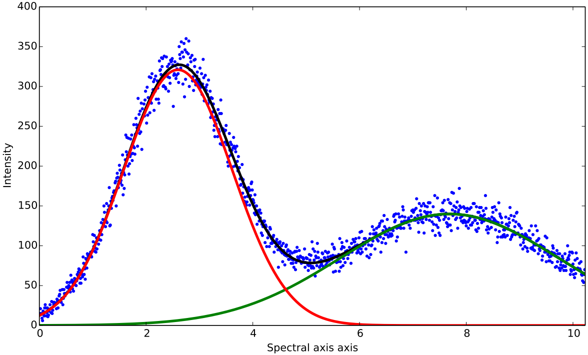
- Set stating parameters.
- Fit.

Non-linear optimisation routine



- Set stating parameters.
- Fit.

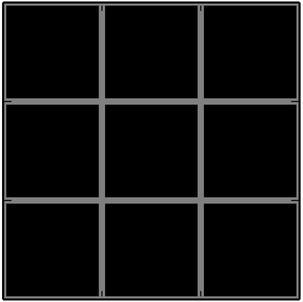
Non-linear optimisation routine



- Set stating parameters.
- Fit.

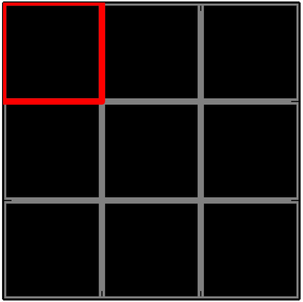
Fitting routine n-dimensions

- Set stating parameters.
- Fit.
- Move to next element.
- Copy parameter values from previous fit.



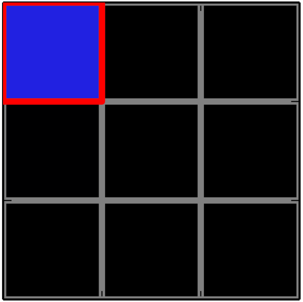
Fitting routine n-dimensions

- Set stating parameters.
- Fit.
- Move to next element.
- Copy parameter values from previous fit.



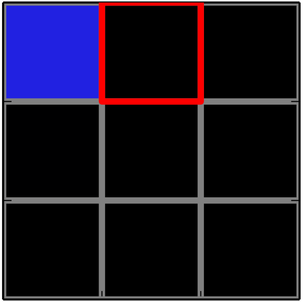
Fitting routine n-dimensions

- Set stating parameters.
- Fit.
- Move to next element.
- Copy parameter values from previous fit.



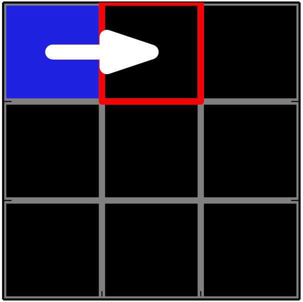
Fitting routine n-dimensions

- Set stating parameters.
- Fit.
- Move to next element.
- Copy parameter values from previous fit.



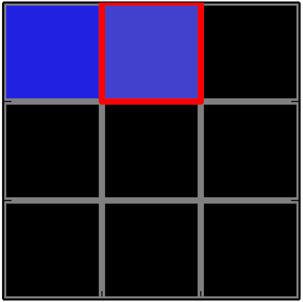
Fitting routine n-dimensions

- Set stating parameters.
- Fit.
- Move to next element.
- Copy parameter values from previous fit.



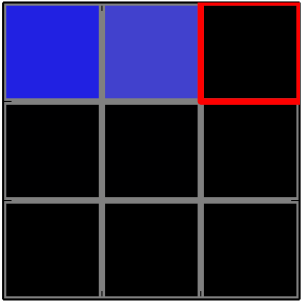
Fitting routine n-dimensions

- Set stating parameters.
- Fit.
- Move to next element.
- Copy parameter values from previous fit.



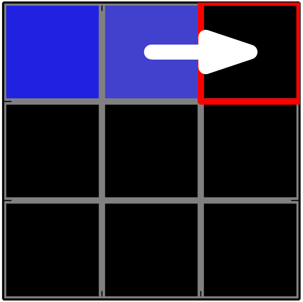
Fitting routine n-dimensions

- Set stating parameters.
- Fit.
- Move to next element.
- Copy parameter values from previous fit.



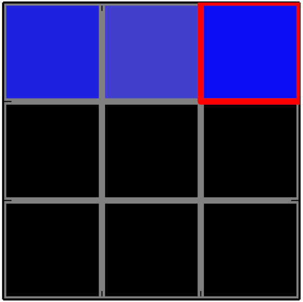
Fitting routine n-dimensions

- Set stating parameters.
- Fit.
- Move to next element.
- Copy parameter values from previous fit.



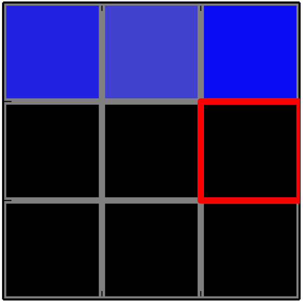
Fitting routine n-dimensions

- Set stating parameters.
- Fit.
- Move to next element.
- Copy parameter values from previous fit.



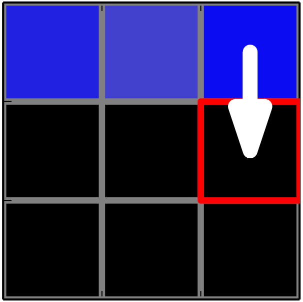
Fitting routine n-dimensions

- Set stating parameters.
- Fit.
- Move to next element.
- Copy parameter values from previous fit.

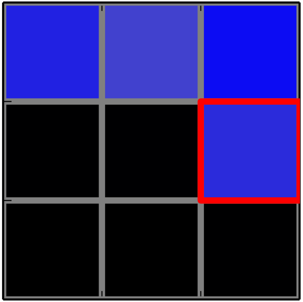


Fitting routine n-dimensions

- Set stating parameters.
- Fit.
- Move to next element.
- Copy parameter values from previous fit.

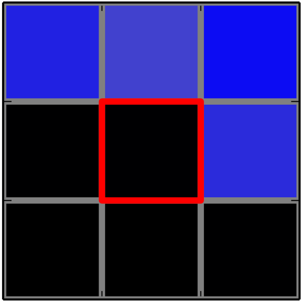


- Set stating parameters.
- Fit.
- Move to next element.
- Copy parameter values from previous fit.



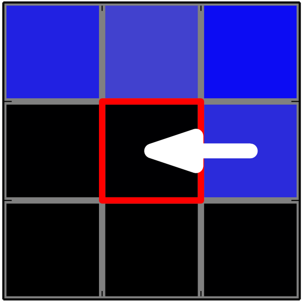
Fitting routine n-dimensions

- Set stating parameters.
- Fit.
- Move to next element.
- Copy parameter values from previous fit.



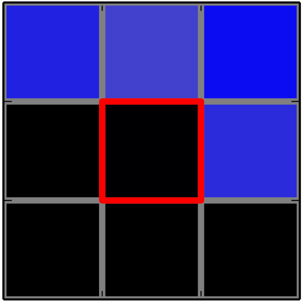
Fitting routine n-dimensions

- Set stating parameters.
- Fit.
- Move to next element.
- Copy parameter values from previous fit.



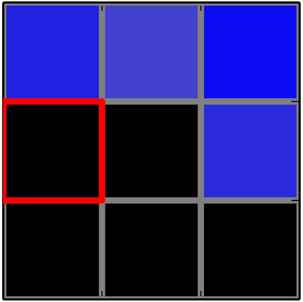
Fitting routine n-dimensions

- Set stating parameters.
- Fit.
- Move to next element.
- Copy parameter values from previous fit.



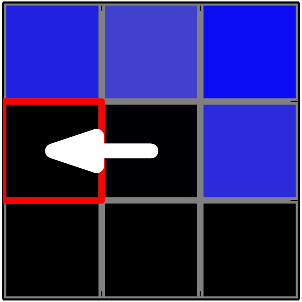
Fitting routine n-dimensions

- Set stating parameters.
- Fit.
- Move to next element.
- Copy parameter values from previous fit.



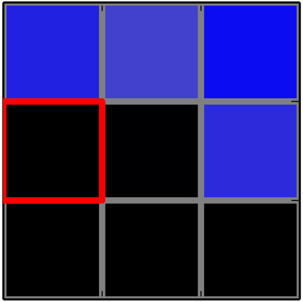
Fitting routine n-dimensions

- Set stating parameters.
- Fit.
- Move to next element.
- Copy parameter values from previous fit.



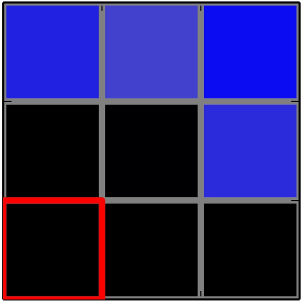
Fitting routine n-dimensions

- Set stating parameters.
- Fit.
- Move to next element.
- Copy parameter values from previous fit.



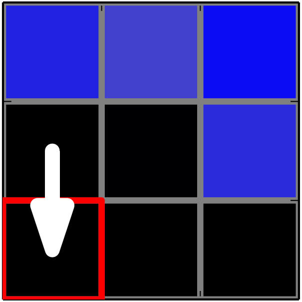
Fitting routine n-dimensions

- Set stating parameters.
- Fit.
- Move to next element.
- Copy parameter values from previous fit.



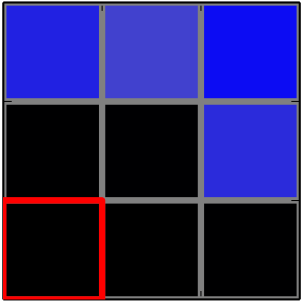
Fitting routine n-dimensions

- Set stating parameters.
- Fit.
- Move to next element.
- Copy parameter values from previous fit.



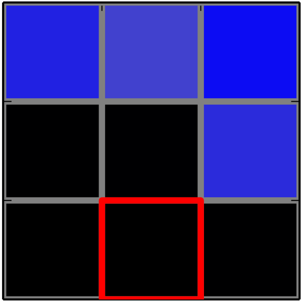
Fitting routine n-dimensions

- Set stating parameters.
- Fit.
- Move to next element.
- Copy parameter values from previous fit.



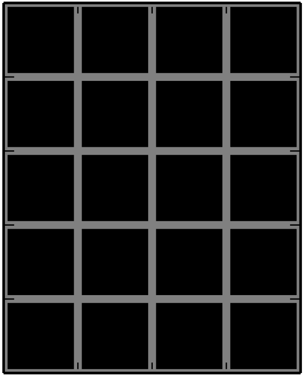
Fitting routine n-dimensions

- Set stating parameters.
- Fit.
- Move to next element.
- Copy parameter values from previous fit.



SAMFire

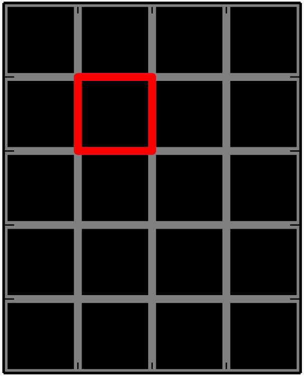
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

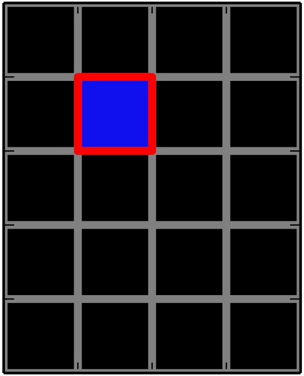
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

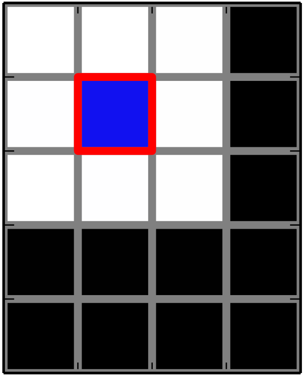
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

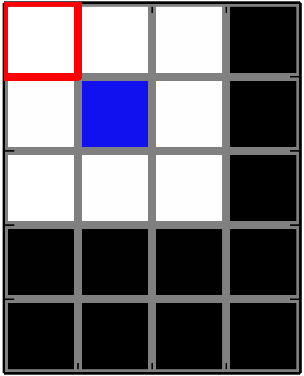
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

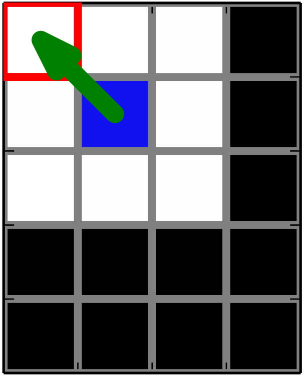
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

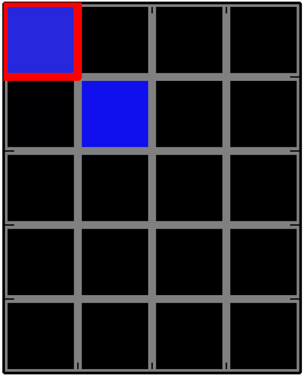
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

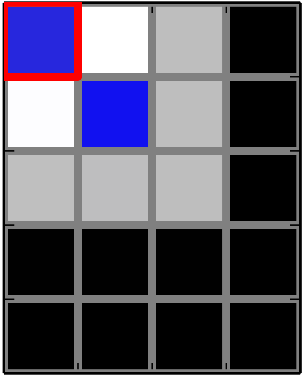
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

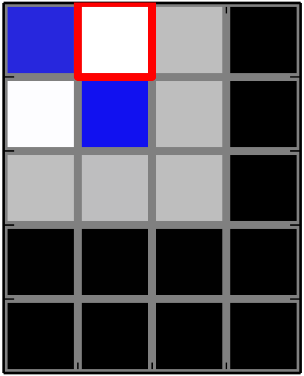
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

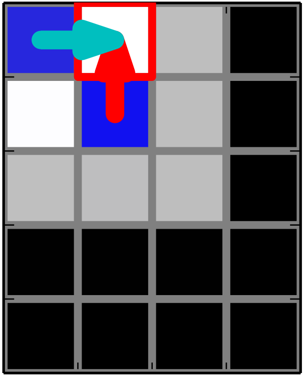
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

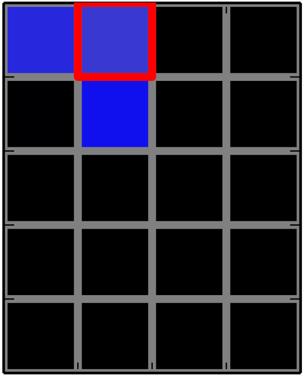
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

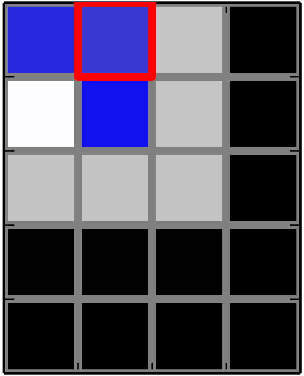
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

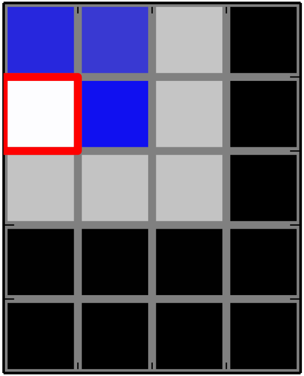
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

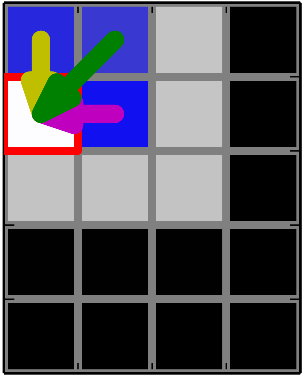
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

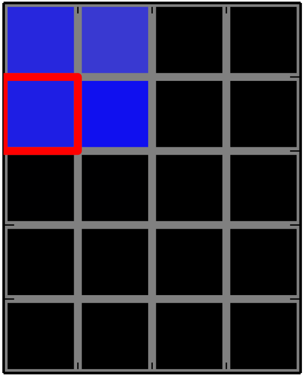
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

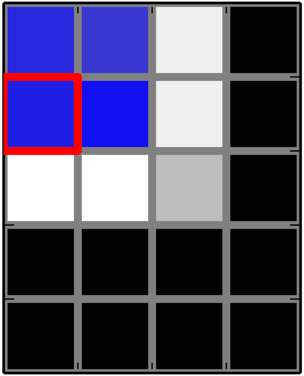
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

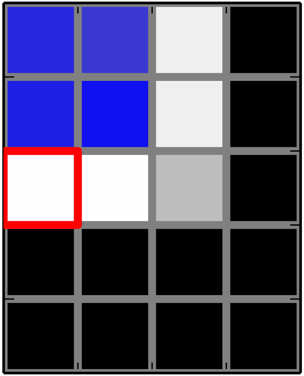
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

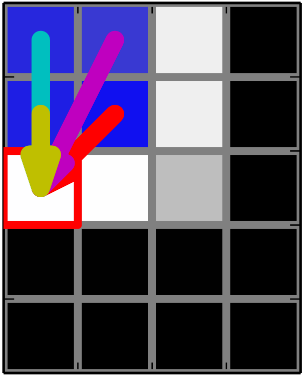
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

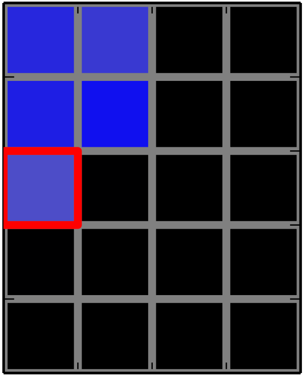
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

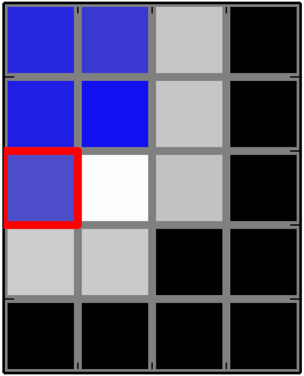
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

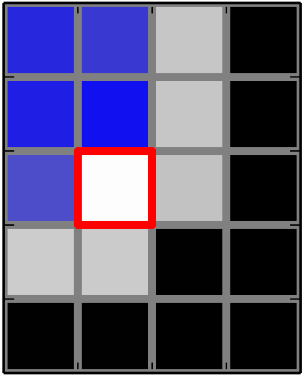
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

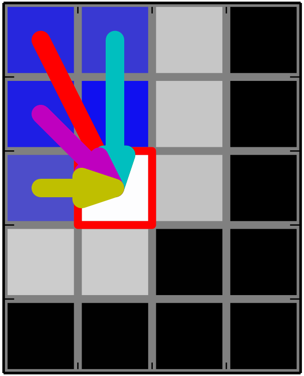
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

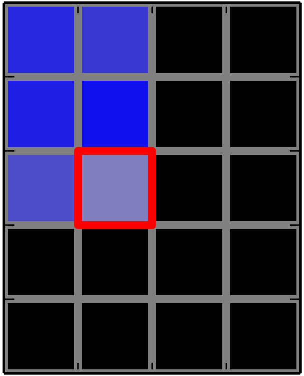
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

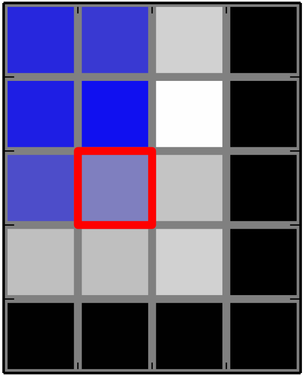
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

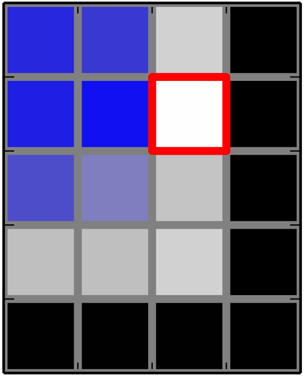
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

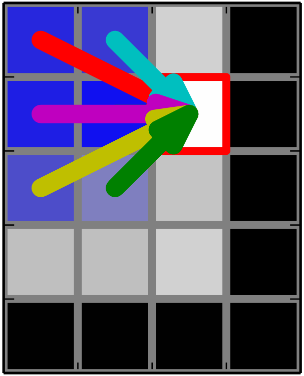
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

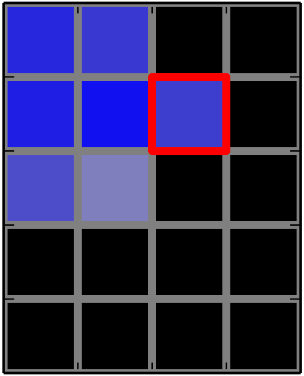
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

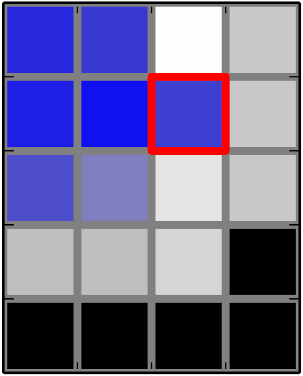
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

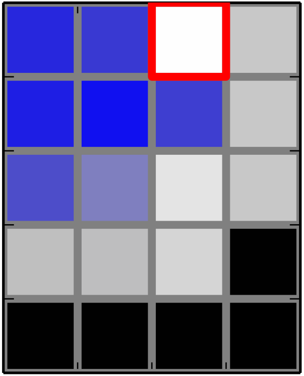
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

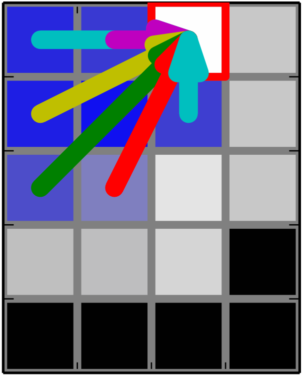
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

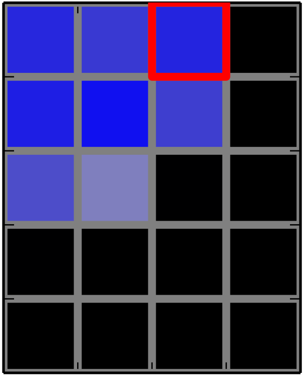
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

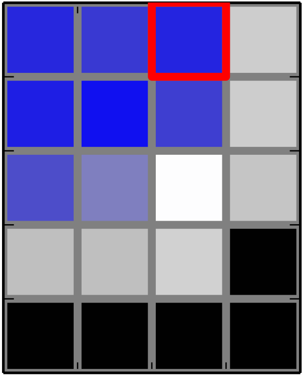
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

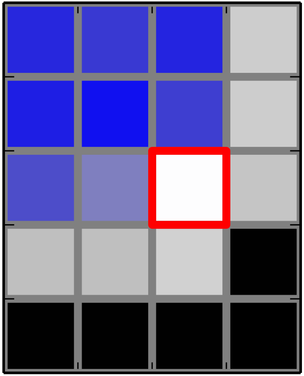
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

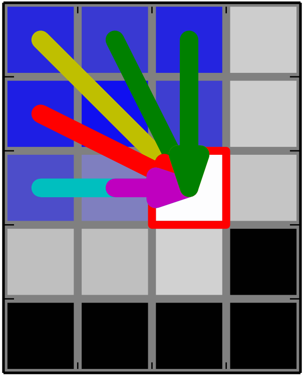
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

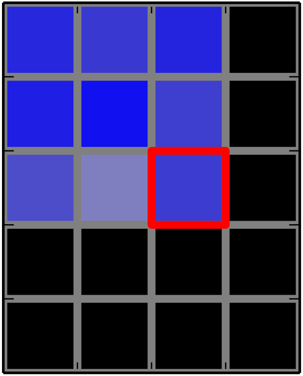
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

SAMFire

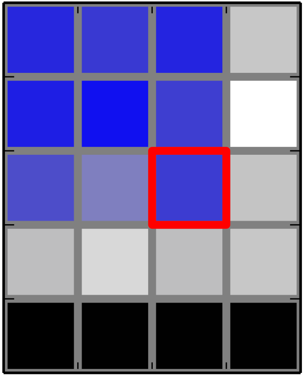
- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

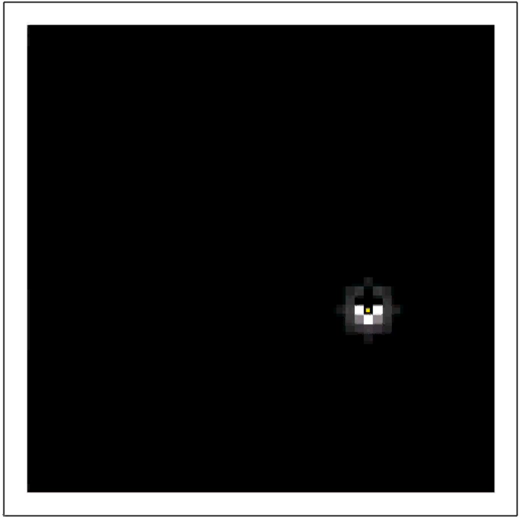
SAMFire

- Set stating parameters.
- Fit.
- Estimate success probability.
- Move to most promising element.
- Calculate starting parameters from all sucessfully fitted elements.



T. Ostasevicious *et al*, EMC2016 proceedings

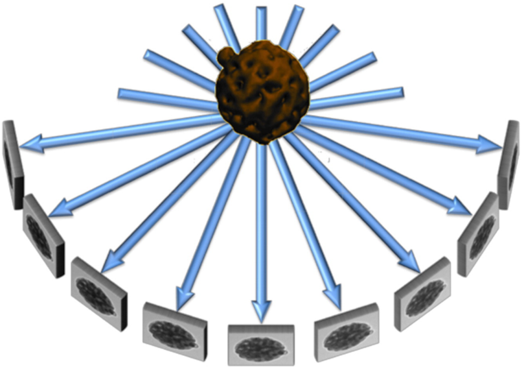
SAMFire parallel fitting example



T. Ostasevicious *et al*, EMC2016 proceedings

Outline

Transmission electron tomography



Radon transform

$Y_\theta = R_\theta(X) \quad i = -70 \ldots 70$

Figure from O. Ersen et al., *Materials Today* 18, 2015

Tomography as a constrained optimisation problem

$$Y_\theta = R_\theta(X) + \text{noise} \quad \theta = -70, \ldots, +70$$

Tomography as a constrained optimisation problem

$$Y_\theta = R_\theta(X) + \mathrm{noise} \quad \theta = -70, \ldots, +70$$

$$X^* = \arg\min_X \left\{ \| R_\theta(X) - Y_\theta \|_2^2 + \lambda \mathrm{f}(\mathrm{X}) \right\}$$

Tomography as a constrained optimisation problem

$$Y_\theta = R_\theta(X) + \text{noise} \quad \theta = -70, \ldots, +70$$

$$X^* = \underset{X}{\arg\min}\left\{\|R_\theta(X) - Y_\theta\|_2^2 + \lambda \mathrm{f}(\mathrm{X})\right\}$$

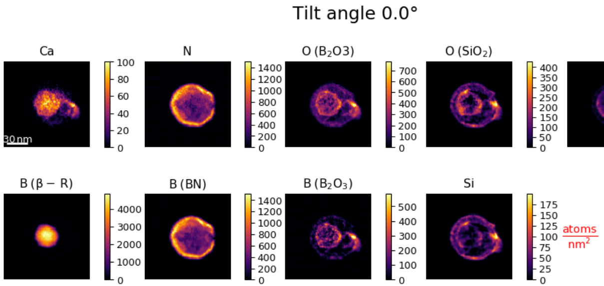
Useful regularisation functions are:

- $\mathrm{L}_1 - \text{norm}$: promotes sparsity
- Total variation: promotes sparsity in the gradient

For EM applications see:

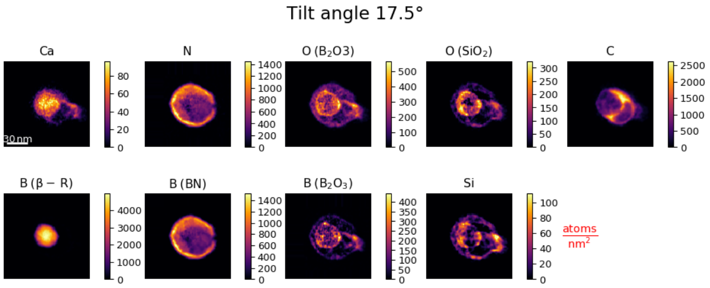
- Leary, Rowan, et al. , *Ultramicroscopy* 131 (2013)
- Goris, Bart, et al. , *Ultramicroscopy* 113 (2012)

SAMFire application - Quantitative bonding tomography



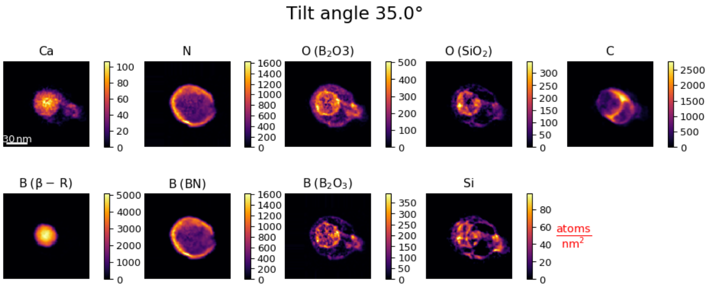
de la Peña, EMC 2016 proceedings

SAMFire application - Quantitative bonding tomography

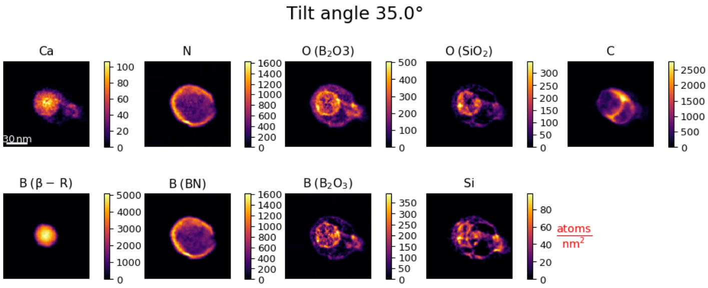


de la Peña, EMC 2016 proceedings

SAMFire application - Quantitative bonding tomography

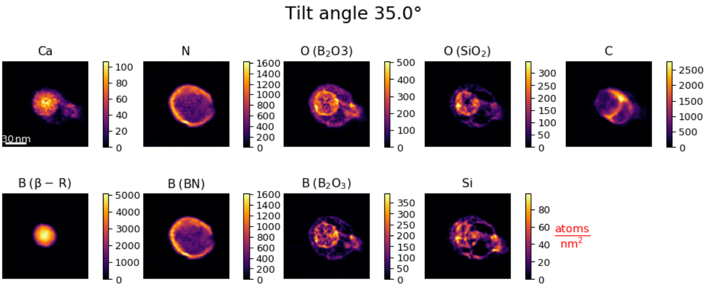


de la Peña, EMC 2016 proceedings



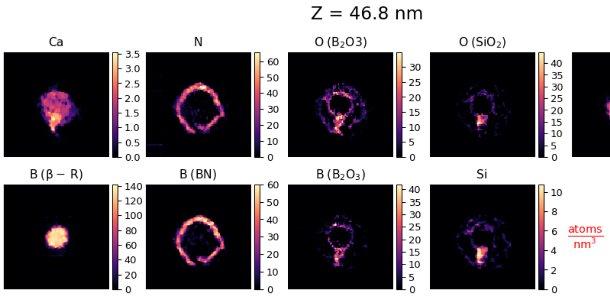
de la Peña, EMC 2016 proceedings

SAMFire application - Quantitative bonding tomography



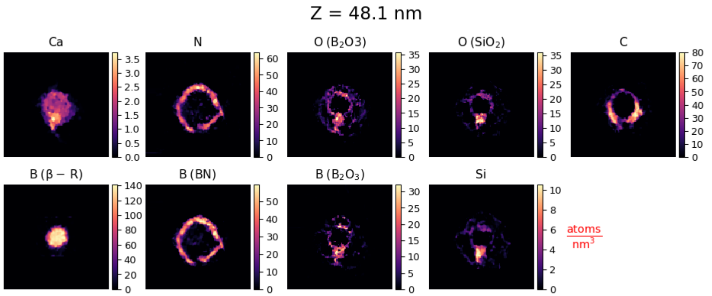
de la Peña, EMC 2016 proceedings

SAMFire application - Quantitative bonding tomography



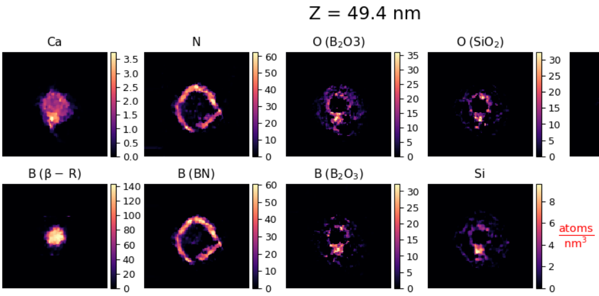
de la Peña, EMC 2016 proceedings

SAMFire application - Quantitative bonding tomography



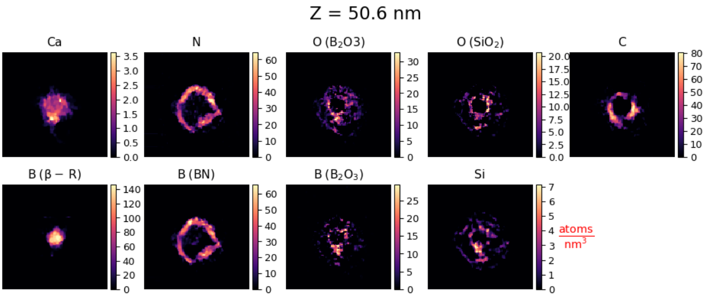
de la Peña, EMC 2016 proceedings

SAMFire application - Quantitative bonding tomography



de la Peña, EMC 2016 proceedings

SAMFire application - Quantitative bonding tomography

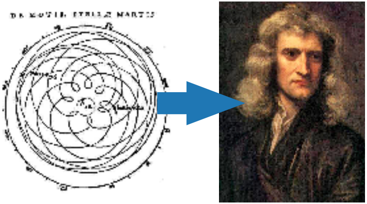


de la Peña, EMC 2016 proceedings

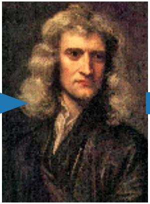
Outline

(Human / Machine) learning

(Human / Machine) learning

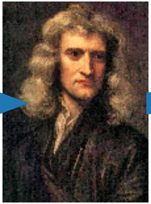


(Human / Machine) learning

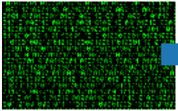


$$F = G\frac{m_1 m_2}{r^2}$$

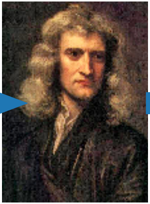
(Human / Machine) learning



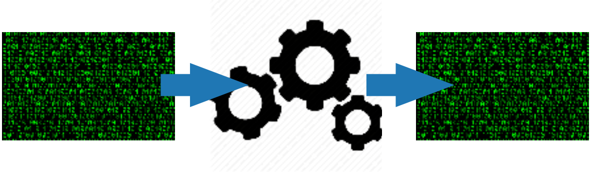
$$F = G\frac{m_1 m_2}{r^2}$$



(Human / Machine) learning



$$F = G\frac{m_1 m_2}{r^2}$$



(Human / Machine) learning electron microscopy

Human learning : $$i\hbar\frac{\partial}{\partial t}\Psi(\mathbf{r},t) = \left[\frac{-\hbar^2}{2\mu}\nabla^2 + V(\mathbf{r},t)\right]\Psi(\mathbf{r},t)$$

(Human / Machine) learning electron microscopy

Human learning : $i\hbar\frac{\partial}{\partial t}\Psi(\mathbf{r},t) = \left[\frac{-\hbar^2}{2\mu}\nabla^2 + V(\mathbf{r},t)\right]\Psi(\mathbf{r},t)$

Machine learning : $H \cdot X = Y$

(Human / Machine) learning electron microscopy

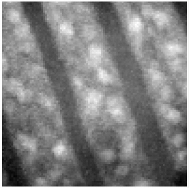
Human learning : $i\hbar\frac{\partial}{\partial t}\Psi(\mathbf{r},t) = \left[\frac{-\hbar^2}{2\mu}\nabla^2 + V(\mathbf{r},t)\right]\Psi(\mathbf{r},t)$

Machine learning : $H \cdot X = Y$

Outline

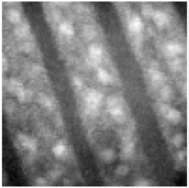
Spinodally decomposed SnO_2/TiO_2 multilayers

HAADF

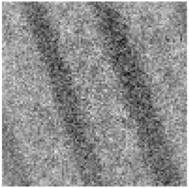


Spinodally decomposed SnO_2/TiO_2 multilayers

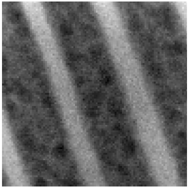
HAADF



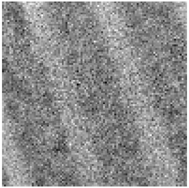
Tin



Titanium

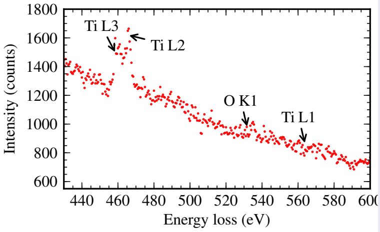


Oxygen

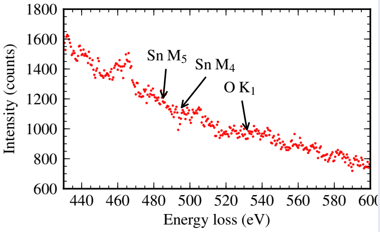


The problem: overlapping peaks and noise

EEL spectrum Ti rich



EEL spectrum Sn rich



Linear mixing
Linearity assumption

$$[a_{i,j}]_{10000\times(64\times64)} = [u_{i,j}]_{(10000)\times 4} \times [v_{i,j}]_{4\times(64\times64)}$$

Paul

Ringo

George

John

Linear mixing

Linearity assumption

$$[a_{i,j}]_{10000\times(64\times64)} = [u_{i,j}]_{(10000)\times4} \times [v_{i,j}]_{4\times(64\times64)}$$

Paul

Ringo

George

John

Mix 1

Mix 2

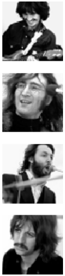
Mix 3

Mix 4

Blind source separation

$$[a_{i,j}]_{I \times I} \times S = \tilde{S}$$

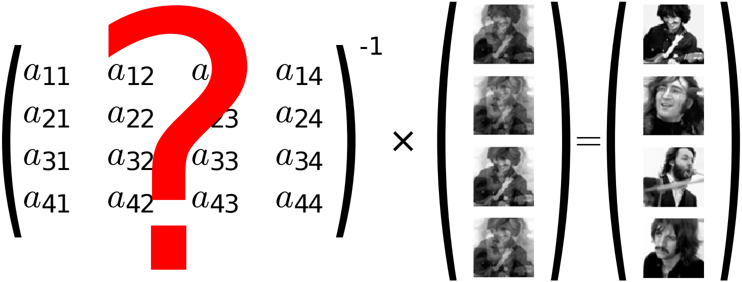
$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \times$$



$$=$$

Blind source separation

$$[a_{i,j}]_{I\times I} \times S = \tilde{S}$$



Noisy linear mixing

$$[a_{i,j}]_{10000\times(64\times64)} = [u_{i,j}]_{(10000)\times4} \times [v_{i,j}]_{4\times(64\times64)}$$

Paul

Ringo

George

John

Mix 1

Mix 2

Mix 3

Mix 4

Noisy linear mixing

$$[a_{i,j}]_{10000\times(64\times 64)} = [u_{i,j}]_{(10000)\times 4} \times [v_{i,j}]_{4\times(64\times 64)} + \text{noise}$$

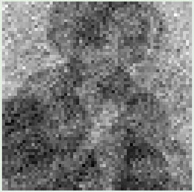
Paul

Ringo

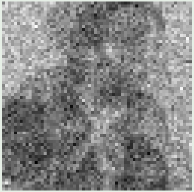
George

John

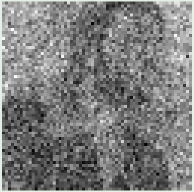
Mix 1



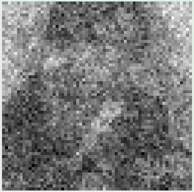
Mix 2



Mix 3



Mix 4



Theorem

Any matrix $A \in \mathbb{R}^{m \times n}$ can be factorised into a singular value decomposition (SVD),

$$A = USV^T \tag{1}$$

where $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices, and $S \in \mathbb{R}^{m \times n}$ is diagonal with $r = \mathrm{rank}(A)$ leading positive entries. The p diagonal entries of S are denoted σ_i for $i = 1, \ldots, p$ where $p = \min\{m, n\}$ and are called the singular values of A. They satisfy the property $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p$.

Theorem

Let the SVD of A be given by (1). If $k < r = \mathrm{rank}(A)$ and $A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$, then

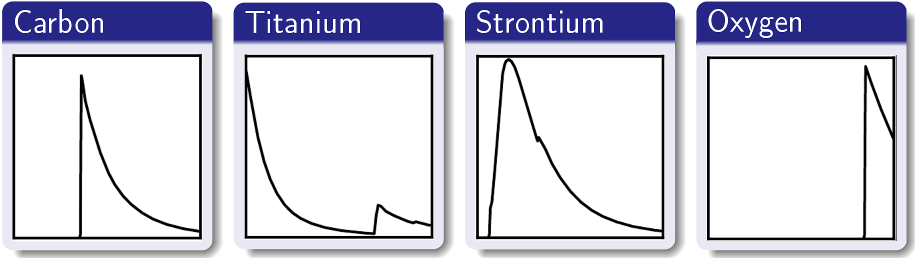
$$\min_{\mathrm{rank}(B)=k} \|A - B\|_2 = \|A - A_k\|_2 = \sqrt{\sum_{i=k+1}^{p} \sigma_i^2}$$

BSS with The Beatles

input image: 1

EELS BSS with The Beatles

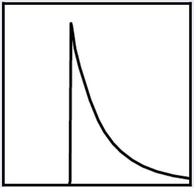
$[d_{i,j}]$ $\times 1024$ $=$ $\times 4 \times [s_{i,j}]_{4 \times 1024}$



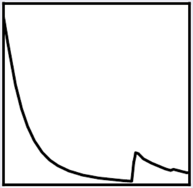
EELS BSS with The Beatles

$$[d_{i,j}]_{(134\times134)\times1024} = [p_{i,j}]_{(134\times134)\times4} \times [s_{i,j}]_{4\times1024}$$

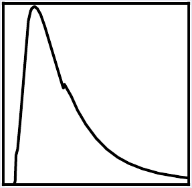
Carbon



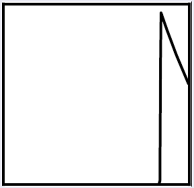
Titanium



Strontium



Oxygen



Carbon

Titanium

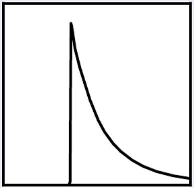
Strontium

Oxygen

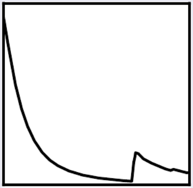
EELS BSS with The Beatles

$$[d_{i,j}]_{(134\times134)\times1024} = [p_{i,j}]_{(134\times134)\times4} \times [s_{i,j}]_{4\times1024} + \text{Poisson noise}$$

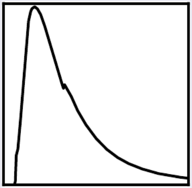
Carbon



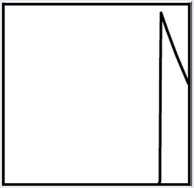
Titanium



Strontium



Oxygen



Carbon

Titanium

Strontium

Oxygen

Synthetic SIs for ICA test

- Using the synthetic SIs we will test the performance of ICA at estimating the mixing matrix when using the first and second derivative as pre-treatment

Low SNR SI

- 4 elements: C, Sr, Ti, O
- 134×134 pixels
- 1024 energy channels
- Poisson noise
- Average number of counts: $\sim 10^3$

Synthetic SIs for ICA test

- Using the synthetic SIs we will test the performance of ICA at estimating the mixing matrix when using the first and second derivative as pre-treatment

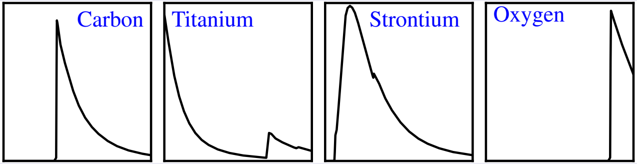
Low SNR SI

- 4 elements: C, Sr, Ti, O
- 134×134 pixels
- 1024 energy channels
- Poisson noise
- Average number of counts: $\sim 10^3$

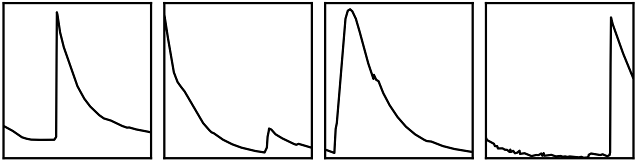
High SNR SI

- 4 elements: C, Sr, Ti, O
- 134×134 pixels
- 1024 energy channels
- Poisson noise
- Average number of counts: $\sim 10^6$

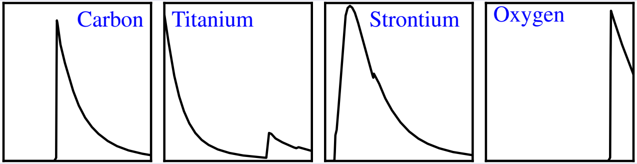
Original spectral components



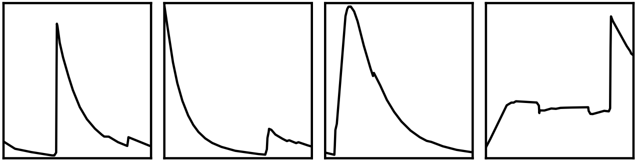
ICA result: Low SNR, first derivative



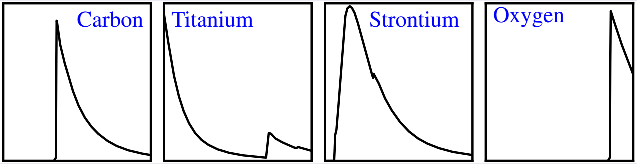
Original spectral components



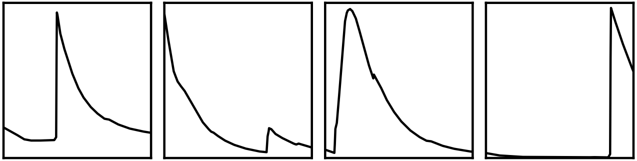
ICA result: Low SNR, second derivative



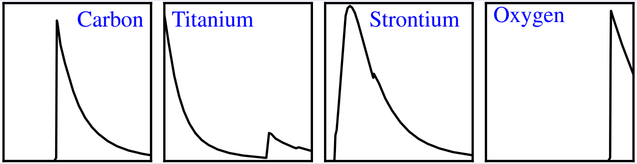
Original spectral components



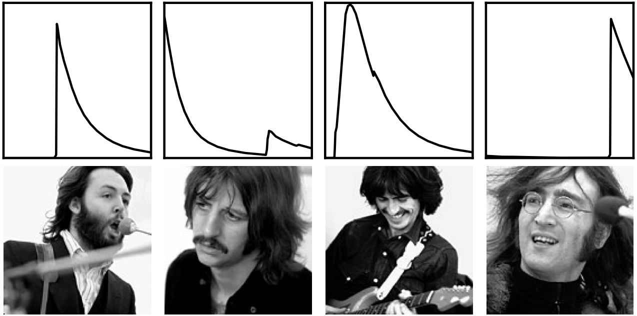
ICA result: High SNR, first derivative



Original spectral components



ICA result: High SNR, second derivative



Original spectral components

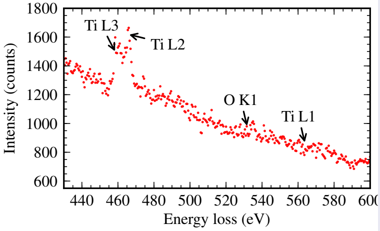
Low SNR: windows methods

ICA result: Low SNR, first derivative

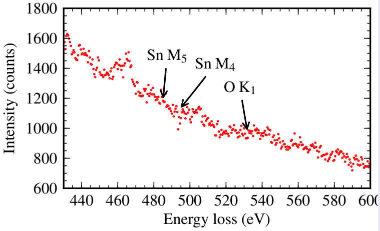
Spinodally decomposed SnO_2/TiO_2 multilayers

Noise reduction by dimensionality reduction

EEL spectrum Ti rich



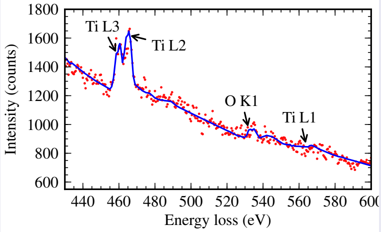
EEL spectrum Sn rich



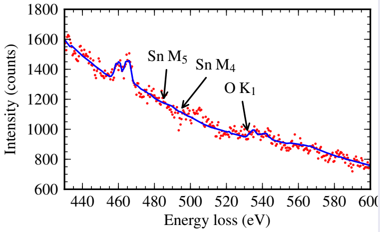
Spinodally decomposed SnO_2/TiO_2 multilayers

Noise reduction by dimensionality reduction

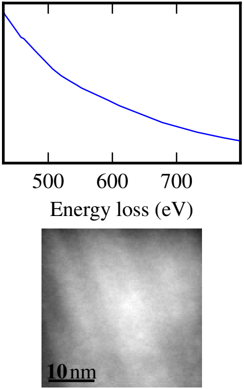
EEL spectrum Ti rich



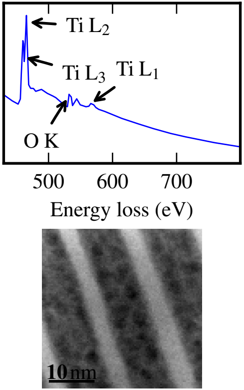
EEL spectrum Sn rich



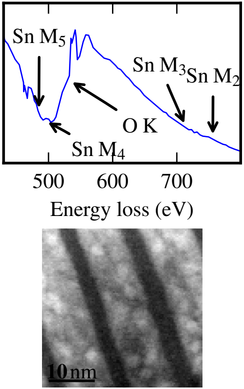
IC 1 Carbon



IC 2 Titanium oxide



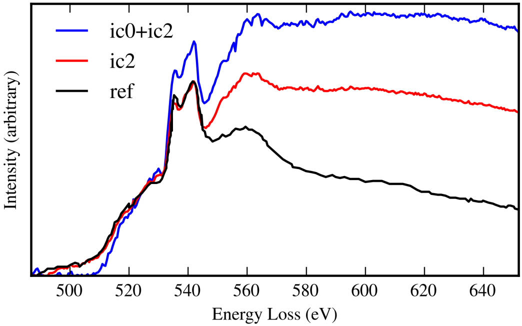
IC 3 Tin Oxide



de la Peña et al., Ultramicroscopy 11 (2011)

Spinodally decomposed SnO_2/TiO_2 multilayers

The effect of plural scatteting



Take home messages

- Singular value decomposition
  - is very useful for
    - Data denoising with no information loss
    - Rank estimation
    - Dimensionality reduction

Take home messages

- Singular value decomposition
  - is very useful for
    - Data denoising with no information loss
    - Rank estimation
    - Dimensionality reduction
  - The SNR improves with the number of trials in the dataset

Take home messages

- Singular value decomposition
  - is very useful for
    - Data denoising with no information loss
    - Rank estimation
    - Dimensionality reduction
  - The SNR improves with the number of trials in the dataset
- Independent component analysis
  - Separates sources from a mixture
  - The accuracy increases with SNR

Bibliography

- PCA: Jolliffe, Ian. Principal component analysis. John Wiley & Sons, Ltd, 2002.
- weighted PCA: Keenan, Michael R., and Paul G. Kotula. "Accounting for Poisson noise in the multivariate analysis of ToF-SIMS spectrum images." Surface and Interface Analysis 36.3 (2004): 203-212.
- ICA: Hyvärinen, A., Karhunen, J., and Oja, E. (2001). Independent Component Analysis. Wiley- Interscience

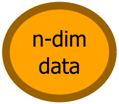
Other methods

- PCA variants: robust PCA, online PCA
- Other BSS methods: non-negative matrix factorization (NMF), vertex component analysis (VCA)
- Tensor decomposition: Spiegelberg, Jakob, Ján Rusz, and Kristiaan Pelckmans. "Tensor Decompositions for the Analysis of Atomic Resolution Electron Energy Loss Spectra." Ultramicroscopy (2017).

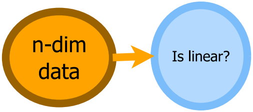
Outline

1. Introduction
   - Introduction
2. Model based quatification
   - The integration method
   - The curve fitting method
   - Multi-dimensional curve fitting
   - Practical application: Analytical tomography
3. Machine learning
   - Introduction
   - EELS core-loss analysis
4. Summary
   - Summary

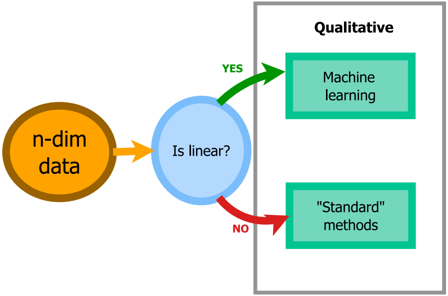
Multi-dimensional data analysis workflow



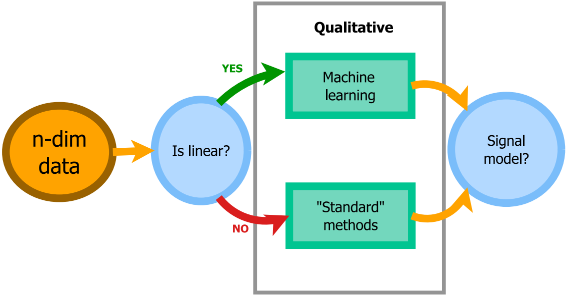
Multi-dimensional data analysis workflow



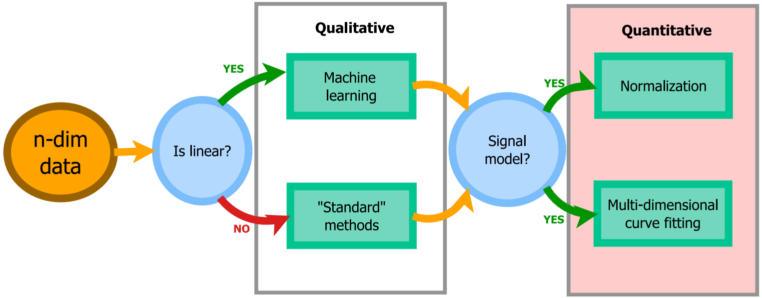
Multi-dimensional data analysis workflow



Multi-dimensional data analysis workflow



Multi-dimensional data analysis workflow



Thank you all for you attention