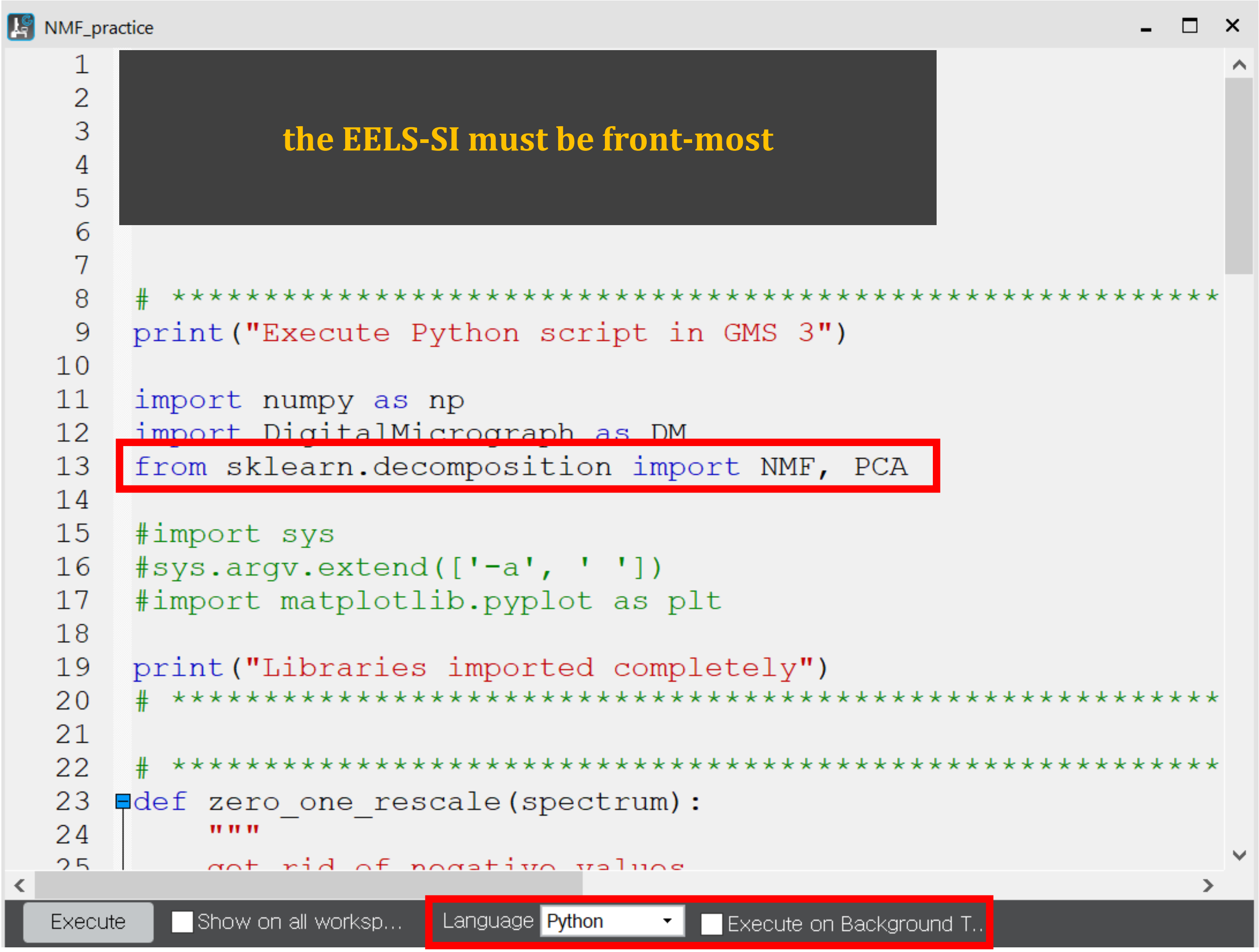
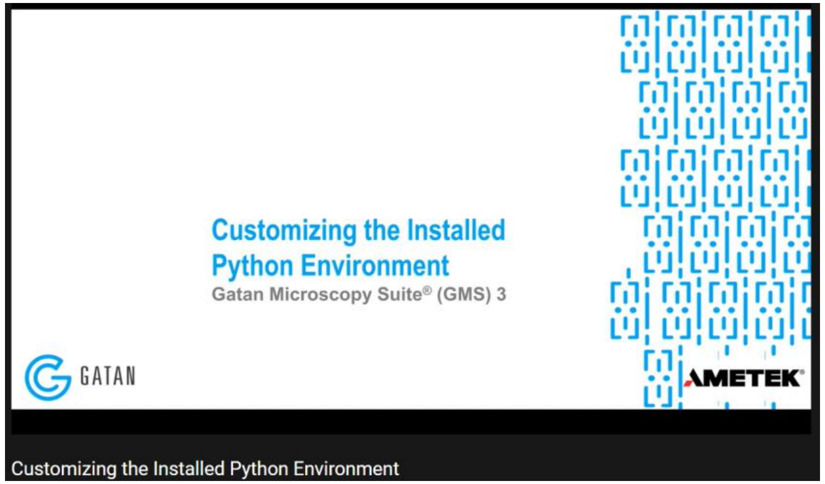
Feature extraction using dimensionality reduction methods from a spectrum image or 4D-STEM data

Requirements: Python-integrated GMS 3, Numpy, Scikit-learn



uncheck "Execute on Background Thread"



https://www.youtube.com/watch?v=-pQMytgaRVg

$$\begin{pmatrix} v_{11} & \cdots & v_{1m} \\ \vdots & \ddots & \vdots \\ v_{n1} & \cdots & v_{nm} \end{pmatrix} \approx \begin{array}{c} \mathbf{w_t} \\ \begin{pmatrix} w_{11} & \cdots & w_{1r} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nr} \end{pmatrix} \end{array} \times \begin{pmatrix} h_{11} & \cdots & h_{1m} \\ \vdots & \ddots & \vdots \\ h_{r1} & \cdots & h_{rm} \end{pmatrix} \mathbf{h_t}$$

$$\boldsymbol{V}\ (n \times m) \approx \boxed{\boldsymbol{W}\ (n \times r)} \times \boxed{\boldsymbol{H}\ (r \times m)}$$

loading vectors

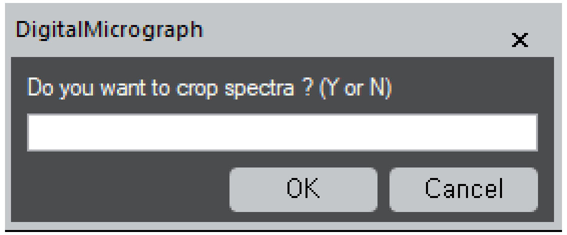
coefficients

[SI]_dimensionality_reduction.py

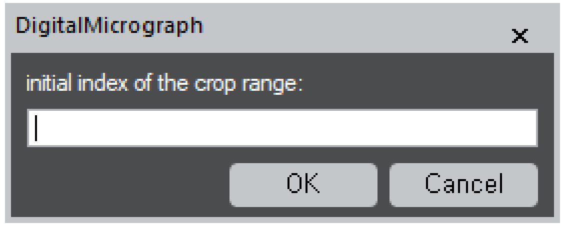
Feature extraction using dimensionality reduction methods from a spectrum image or 4D-STEM data

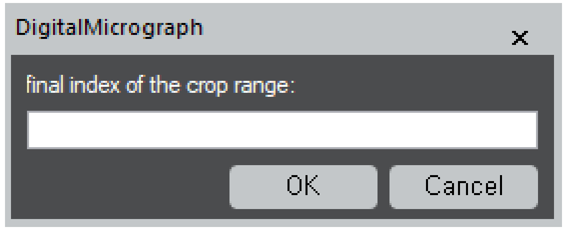
Requirements: Python-integrated GMS 3, Numpy, Scikit-learn

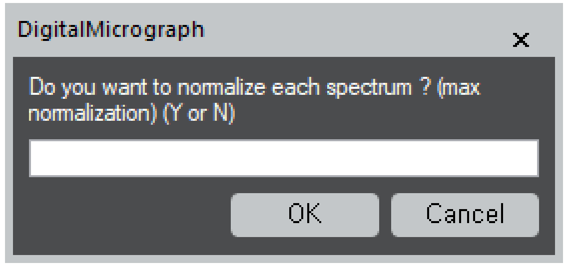
Case 1: 3D spectrum image



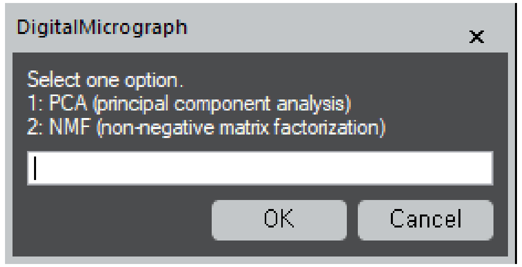
if you write "Y"



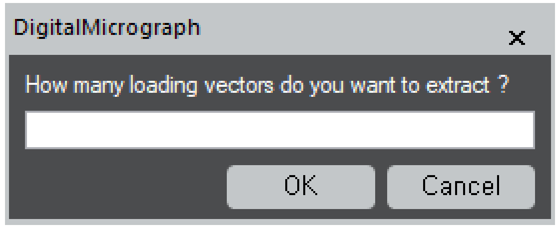




normalize or not (max-normalization)



PCA or NMF



select r

$$\begin{pmatrix} v_{11} & \cdots & v_{1m} \\ \vdots & \ddots & \vdots \\ v_{n1} & \cdots & v_{nm} \end{pmatrix} \approx \begin{array}{c} \mathbf{w_t} \\ \begin{pmatrix} w_{11} & \cdots & w_{1r} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nr} \end{pmatrix} \end{array} \times \begin{pmatrix} h_{11} & \cdots & h_{1m} \\ \vdots & \ddots & \vdots \\ h_{r1} & \cdots & h_{rm} \end{pmatrix} \begin{array}{l} \mathbf{h_t} \\ \\ \end{array}$$

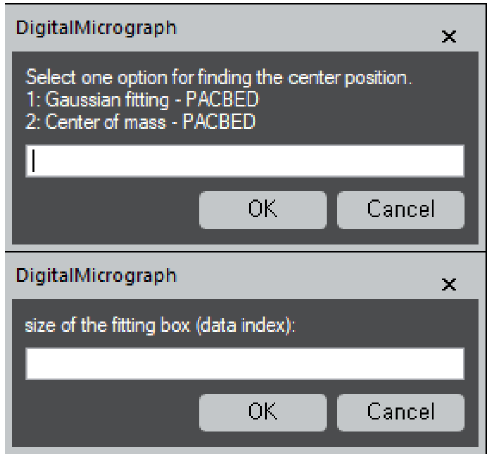
$$\boldsymbol{V}\ (n \times m) \approx \boxed{\boldsymbol{W}\ (n \times r)} \times \boxed{\boldsymbol{H}\ (r \times m)}$$

loading vectors **coefficients**

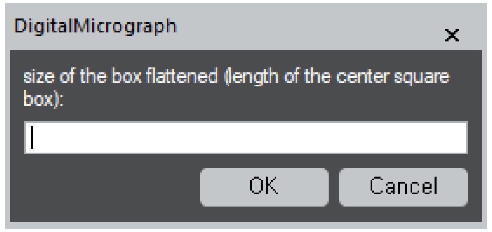
Feature extraction using dimensionality reduction methods from a spectrum image or 4D-STEM data

Requirements: Python-integrated GMS 3, Numpy, Scikit-learn

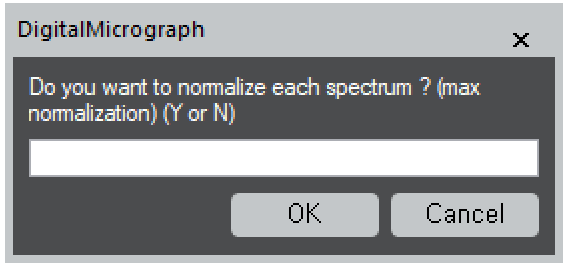
Case 2: 4D-STEM data



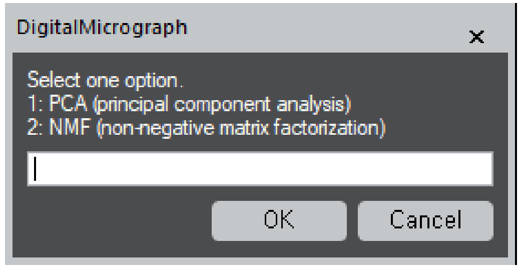
find the center position



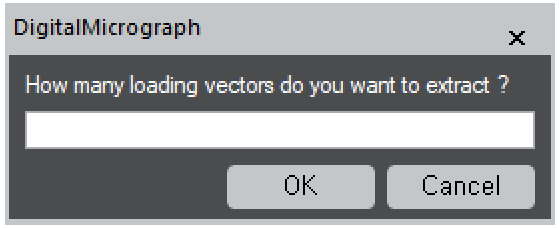
flatten the data
for a square region



normalize or not
(max-normalization)

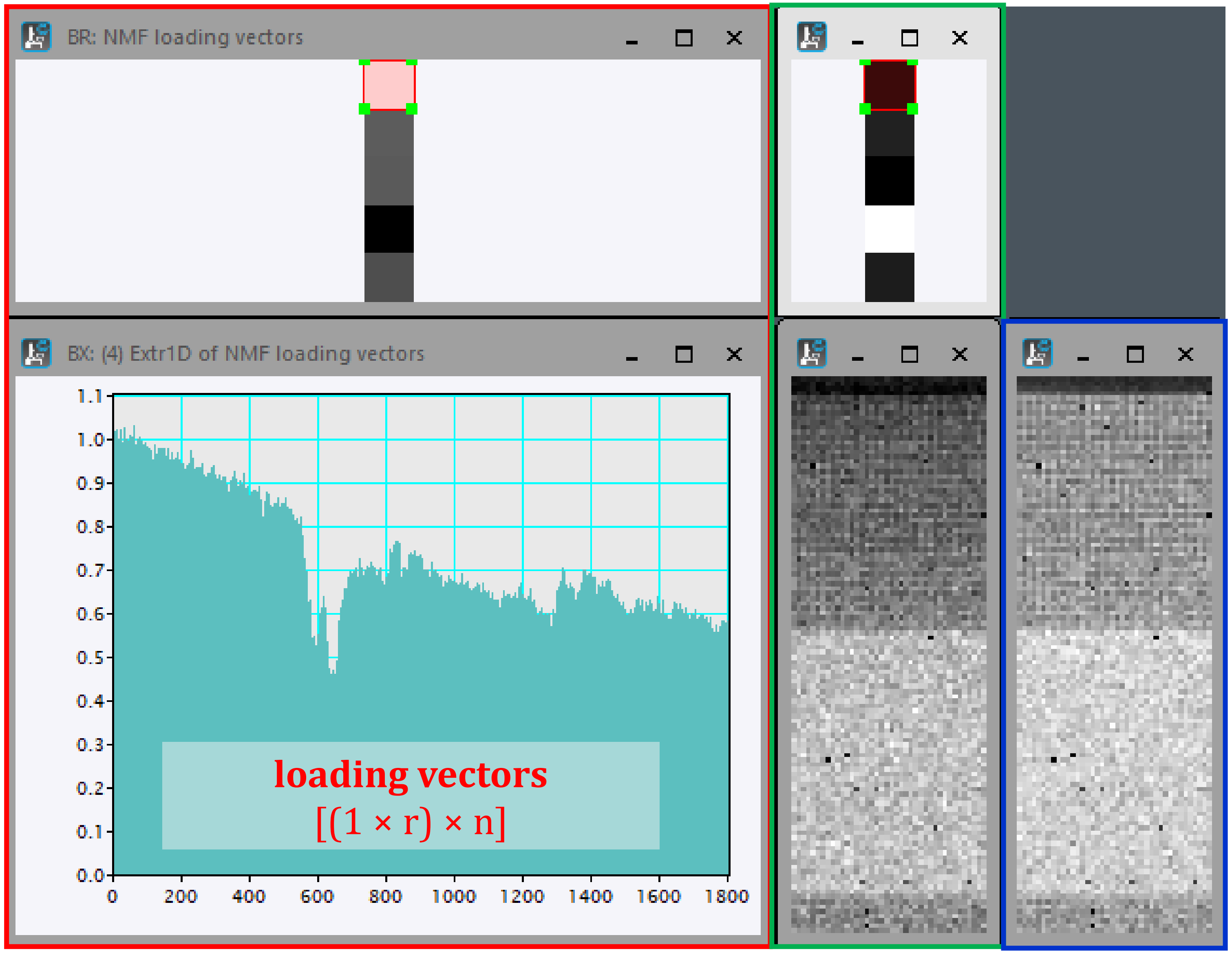


PCA or NMF



select *r*

Feature extraction using dimensionality reduction methods from a spectrum image or 4D-STEM data



Example

coefficient maps
[sx × sy × r]

Requirements:
Python-integrated GMS 3, Numpy, Scikit-learn

output example
(EELS-SI)

reconstructed EELS-SI
[sx × sy × n]

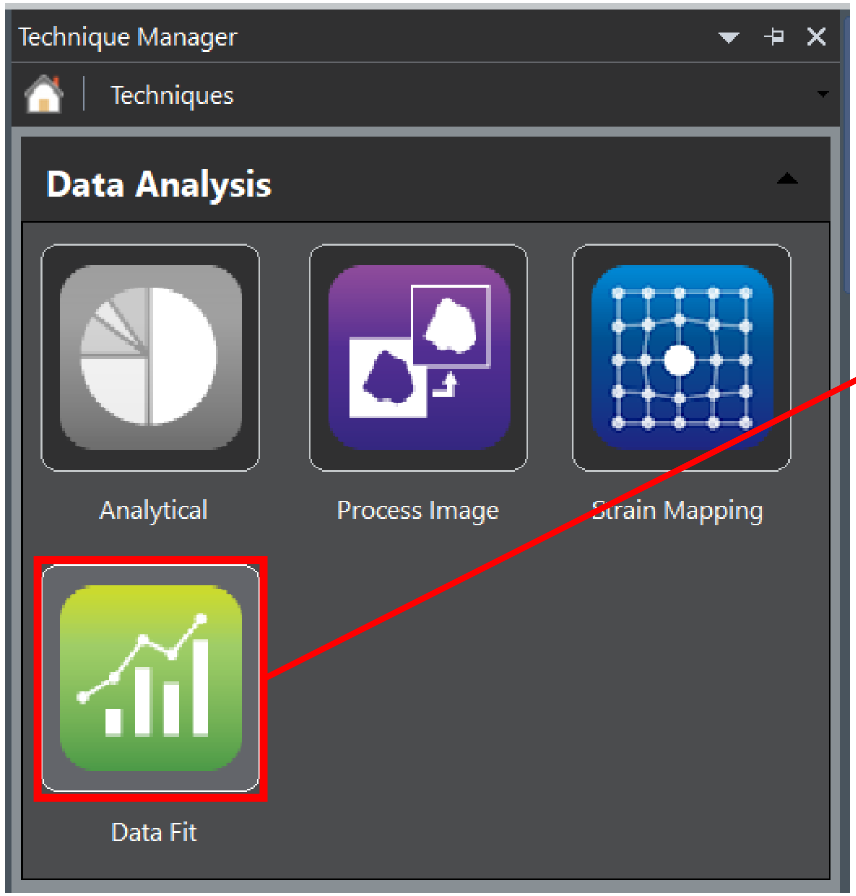
(sx, sy → STEM scanning size)

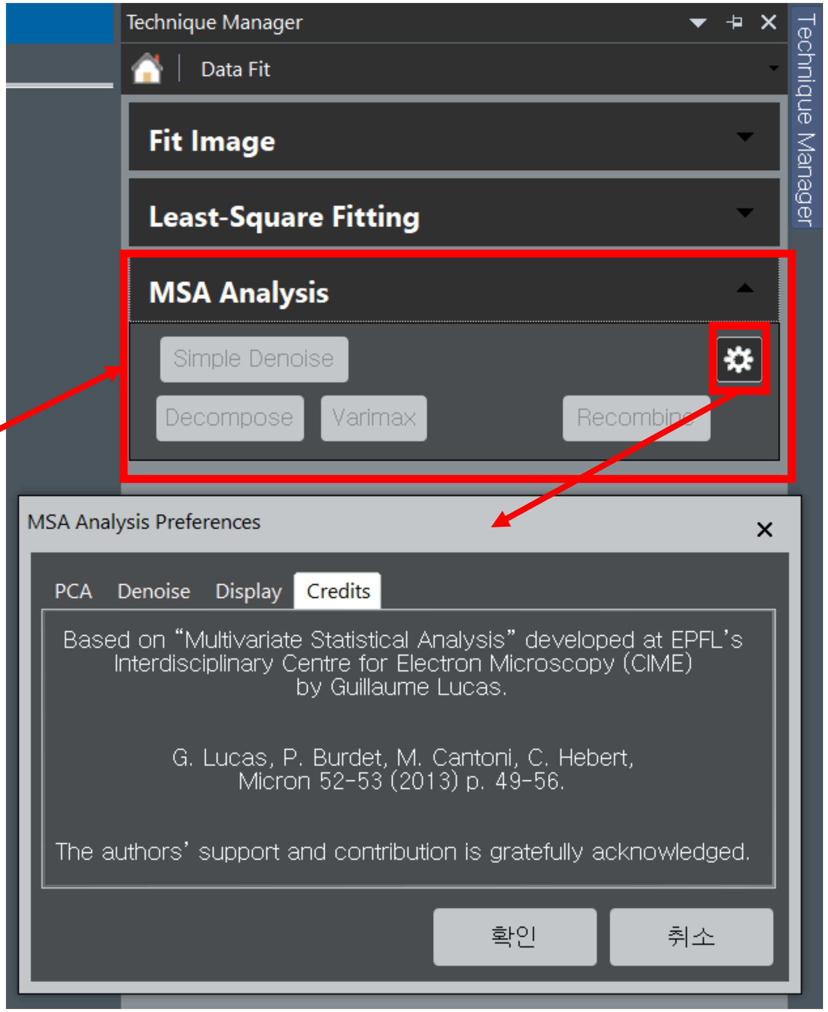
Feature extraction using dimensionality reduction methods from a spectrum image or 4D-STEM data

multivariate statistical analysis in GMS 3

it works only for an EELS-SI

https://www.youtube.com/watch?v=IgoPApTiN4I
(official tutorial)

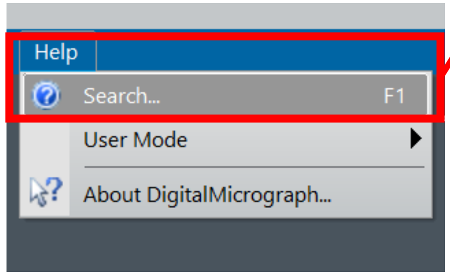


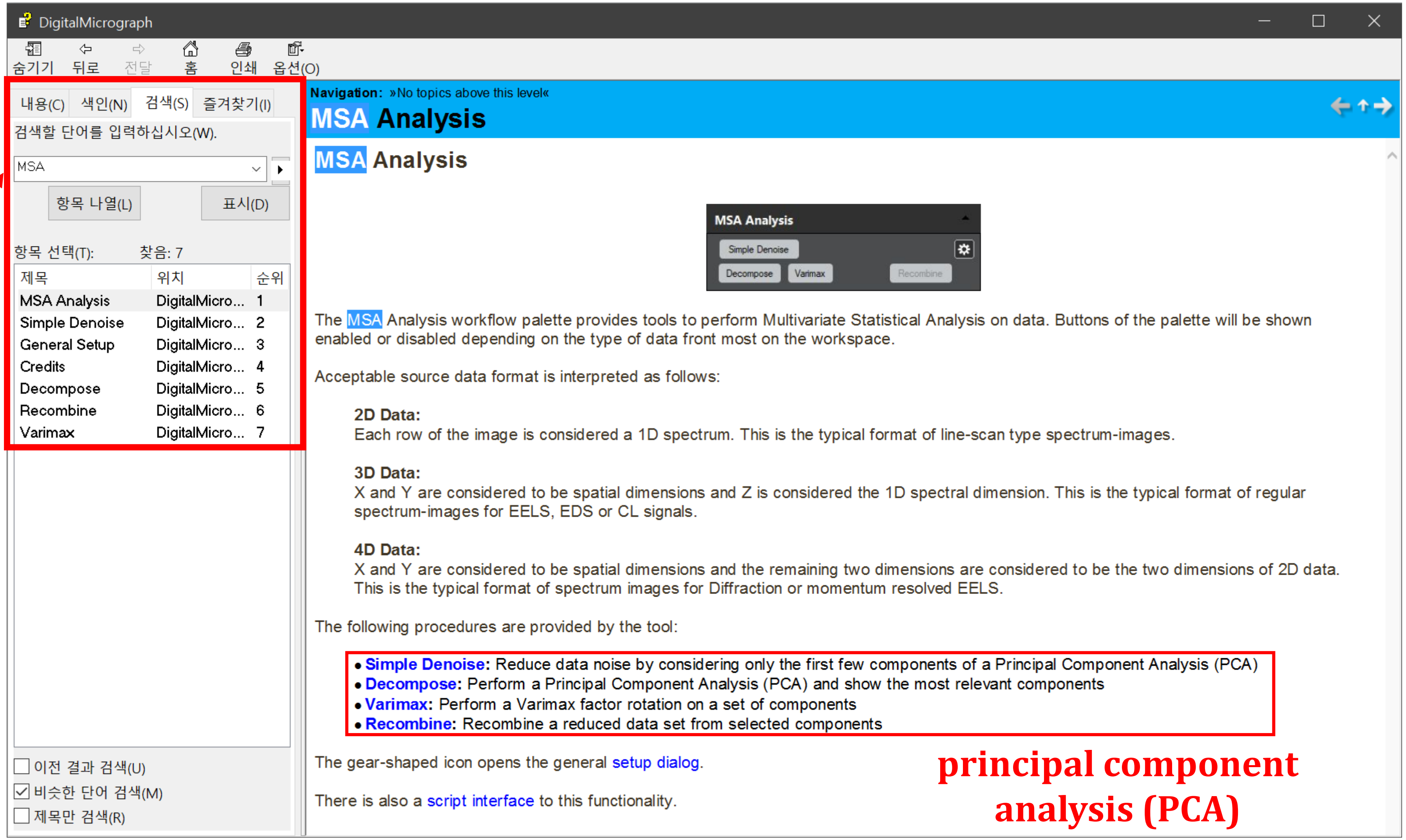


Feature extraction using dimensionality reduction methods from a spectrum image or 4D-STEM data

multivariate statistical analysis in GMS 3

https://www.youtube.com/watch?v=IgoPApTiN4I
(official tutorial)





Feature extraction using dimensionality reduction methods from a spectrum image or 4D-STEM data

multivariate statistical analysis in GMS 3

https://www.youtube.com/watch?v=IgoPApTiN4I
(official tutorial)

EEL spectrum images
n steps x **m** pixels

=

multivariate matrix
(n x m)

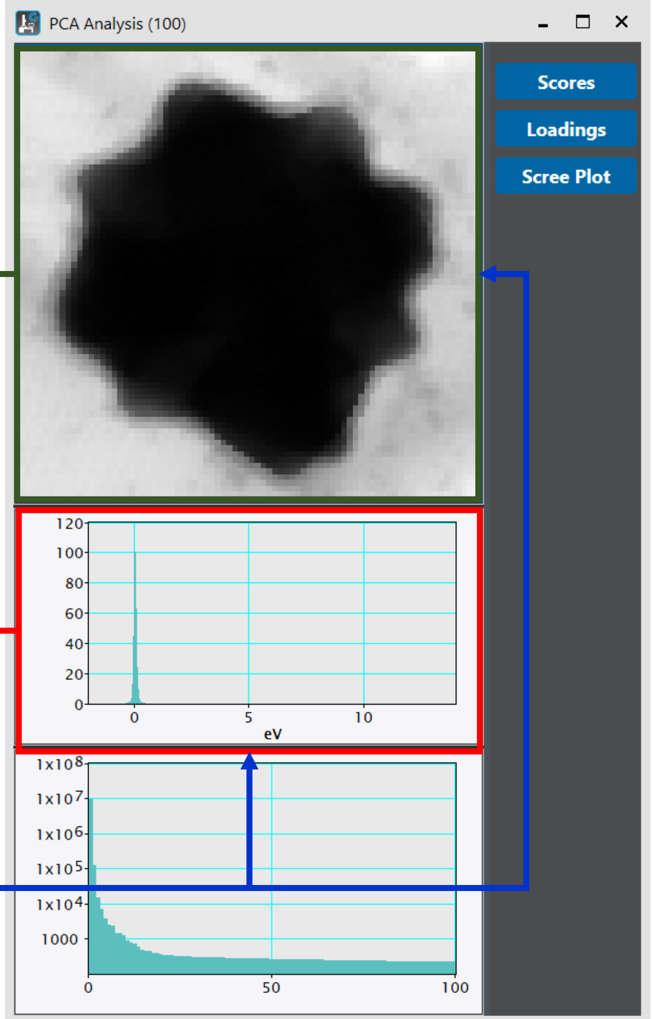
$\mathbf{w}_t$ $\mathbf{h}_t$

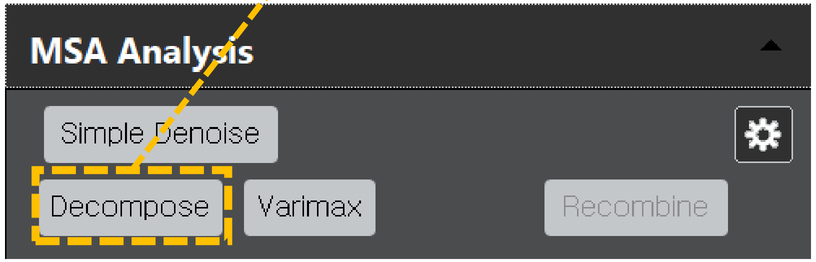
$$\begin{pmatrix} v_{11} & \cdots & v_{1m} \\ \vdots & \ddots & \vdots \\ v_{n1} & \cdots & v_{nm} \end{pmatrix} \approx \begin{pmatrix} w_{11} & \cdots & w_{1r} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nr} \end{pmatrix} \times \begin{pmatrix} h_{11} & \cdots & h_{1m} \\ \vdots & \ddots & \vdots \\ h_{r1} & \cdots & h_{rm} \end{pmatrix}$$

$$\boldsymbol{V}\ (n \times m) \approx \boxed{\boldsymbol{W}\ (n \times r)} \times \boxed{\boldsymbol{H}\ (r \times m)}$$

loading vectors **scores**

$\mathbf{h}_t$





$\mathbf{w}_t$

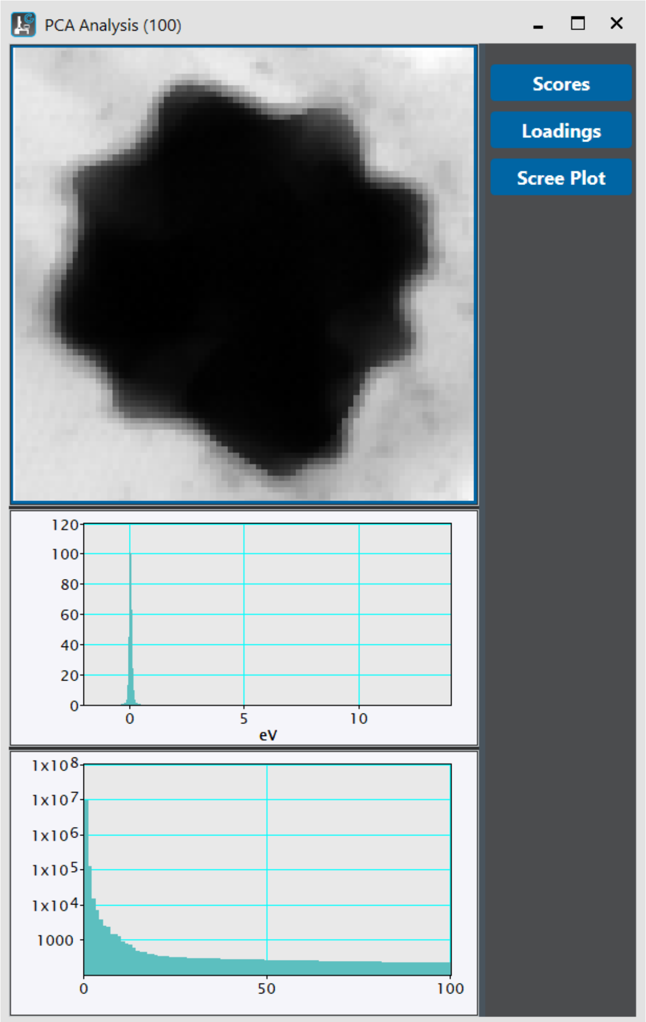
the EELS-SI must be front-most

choose a loading vector

Feature extraction using dimensionality reduction methods from a spectrum image or 4D-STEM data

multivariate statistical analysis in GMS 3

https://www.youtube.com/watch?v=IgoPApTiN4I
(official tutorial)



EEL spectrum images
n steps x **m** pixels

=

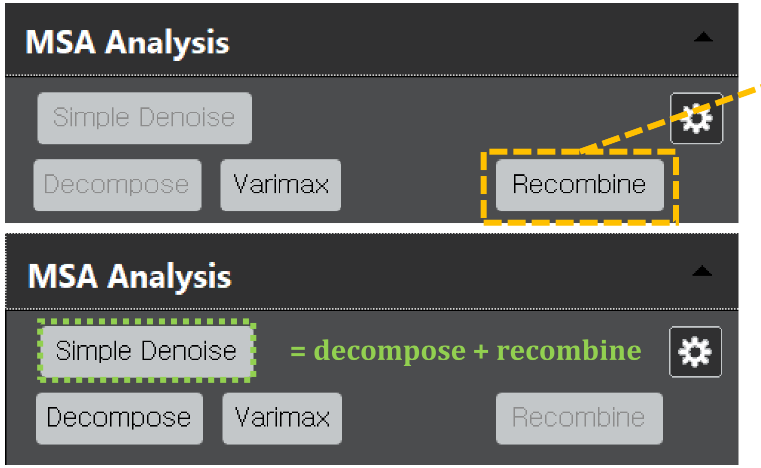
multivariate matrix
(n x m)

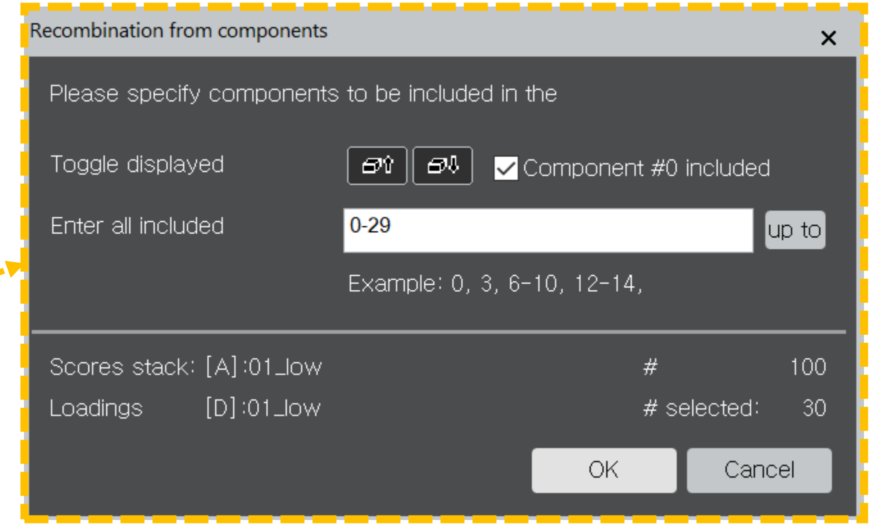
$$\begin{pmatrix} v_{11} & \cdots & v_{1m} \\ \vdots & \ddots & \vdots \\ v_{n1} & \cdots & v_{nm} \end{pmatrix} \approx \begin{pmatrix} w_{11} & \cdots & w_{1r} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nr} \end{pmatrix} \times \begin{pmatrix} h_{11} & \cdots & h_{1m} \\ \vdots & \ddots & \vdots \\ h_{r1} & \cdots & h_{rm} \end{pmatrix}$$

$\mathbf{w_t}$ $\mathbf{h_t}$

$$\boldsymbol{V}\ (n \times m) \approx \boxed{\boldsymbol{W}\ (n \times r)} \times \boxed{\boldsymbol{H}\ (r \times m)}$$

loading vectors **scores**





choose loading vectors used to reconstruct the dataset

$$\underset{n \times m}{V'} = \underset{n \times r'}{W'} \times \underset{r' \times m}{H'}$$
$$(r' \leq r)$$