**Supporting Information**

**PyFitit: The software for quantitative analysis of XANES spectra using machine-learning algorithms**

A. Martini1,2\*, S. A. Guda2,3\*, A. A. Guda2, G. Smolentsev4, A. Algasov3, O. Usoltsev2, M. A. Soldatov2, A. Bugaev2, Yu. Rusalev2, C. Lamberti1,2, A. V. Soldatov2.

1Department of Physics, INSTM Reference Center and NIS and CrisDi Interdepartmental Centers, University of Torino, Via P. Giuria 1, I-10125 Torino, Italy

2The Smart Materials Research Institute, Southern Federal University, 344090 Sladkova 178/24 Rostov-on-Don, Russia

3Institute of mathematics, mechanics and computer science, Southern Federal University, 344090 Milchakova 8a, Rostov-on-Don, Russia

4Paul Scherrer Insitute, Villigen 5232, Switzerland

Corresponding authors: [andrea.martini@unito.it](mailto:andrea.martini@unito.it), [gudasergey@gmail.com](mailto:gudasergey@gmail.com)

1. **Spectral unmixing using PyFitIt**
   1. **Spectral Decomposition procedure**

How described in Section 2.1 of the main text, we start from the hypothesis that each XANES spectrum, belonging to an experimental dataset and acquired during a determined experimental conditions, can be expressed as the weighted sum of *N* uncorrelated “pure” spectra (or main components) for their related concentration profiles :

|  |  |
| --- | --- |
|  | (S.1) |

Here is characterized by rows and columns while matrices and have respectively the following dimensions ) and . In order to realize the decomposition reported in equation (S.1) three main steps are needed.

First step:

The experimental dataset is decomposed through the Singular Value Decomposition (SVD) procedure:

|  |  |
| --- | --- |
|  | (S.2) |

Where **U** () and **V** () are orthogonal matrices formed, respectively, by the eigenvectors of matrixes and . is a rectangular matrix whose elements are the positive square roots of the eigenvalues of (or , called singular values, sorted in descending order.

Second step:

After that, the correct number (i.e. *N*) of significant singular values has been identified (on the basis of the statistical techniques described in Section 2.2 of the main text) the remaining smallest elements of are set to zero. This choice determines the dimensional reduction of matrices and from () and () to () and (). For example, if two components are required (i.e. *N*=2), equation (S.2) acquires the following form:

|  |  |
| --- | --- |
| **=** | (S.3) |

Where and are respectively the dimensional reduced form of , and **V**.

According to the Eckart-Young theorem, it is possible to assert that matrix is the best approximation of **μ** among all other matrices with rank *N*. This result can be connected with equation (1) of the main text adding to equation (S.3) the residual matrix (with dimension ) given by:

If the new matrix is indistinguishable from **μ** within experimental noise (i.e. the matrix is composed by *n* spectral profiles resembling only noise) then, the experimental set of data can be described by the combination of only *N* components. Otherwise, the number of components must be increased by adding the next diagonal elements in . The decomposition showed in equation (S.3) is usually termed as *data compression* and can be also seen as a filter procedure able to remove the experimental noise from the original data. It is worth to mention that the variables and **,**here re ported, correspond to the , and **V** factors present in equation (1) of the main text.

Third step:

The third step foresees the factorization of equation (S.3) in the (S.1) form by the following relation: and. Under this representation matrix  becomes a data matrix composed by *N* spectral profiles while contains its related concentration profiles. However, how is said in the main text, matrixed and do not have any chemical/physical meaning. This problem can be overcame by the introduction of a “transformation” matrix **T** in equation (S.3) in the following way:

|  |  |  |
| --- | --- | --- |
|  |  | (S.4) |

In this last equation, it is possible to see that, globally, matrix **T** does not influence the entire decomposition reported in equation (S.3), however each element can be modified until reasonable spectral and concentration profiles are obtained, realising, in this way, equation (S.1).

* 1. **Constraints setting**

How described in the main text, the transformation matrix is a squared matrix that user can use for rotating and distorting the columns of the matrixand the rows of . Clearly, the number of elements of goes as . For this reason, in order to reduce the number of elements of matrix  (that can be directly modified by the user) some constraints must be imposed. On this basis, the first or the last experimental spectrum (or even both) can be fixed. This procedure must be performed if and only if it has been attested that they represent some *pure* species in the chemical data mixture represented by **μ**. If these requirements are satisfied, the first or the last (or both) column/s of **S** in equation (S.1) are set equal to the corresponding column values of **μ**. Consequently, the related elements of are obtained solving this set of linear equations:

|  |  |
| --- | --- |
|  | (S.5) |

Here, symbols are the elements of the matrix.

The number of unknown values in **T** can be further reduced by the normalization of the spectra. First, each experimental spectrum , constituting the input dataset, should be normalized using this equation:

|  |  |
| --- | --- |
|  | (S.6) |

Where and are the minimum and maximum values of the energy range where spectra are defined. As follows from equation (S.5) each spectrum of matrix is written as a combination of a series of abstract spectral values multiplied by related the “transformation” elements . The highest contribution to the construction of each “pure” spectrum comes from the ,which ischaracterized by the highest singular value. Actually is the only one component which resembles the XANES spectrum, while …can be considered as some difference additives to . Therefore normalization (S.6) can be applied to by calculating coefficient σ (S.7) and introducing it into matrix **T** (S.8):

|  |  |
| --- | --- |
|  | (S.7) |

|  |  |  |
| --- | --- | --- |
|  |  | (S.8) |

* 1. **Statistical criteria to determine the number of principal components: the case of IE and IND factors**

The Imbedded Error function (IE) is characterized by an expression given by:

|  |  |  |
| --- | --- | --- |
|  |  | (S.9) |

where *k* represents the number of components used to reproduce the dataset . If the experimental errors are distributed randomly and uniformly along each spectrum of , then the sum of squares of the projections of the errors, defined as (where are the elements of the residual matrix and represents the ij element of reconstructed with *k* components), onto each secondary eigenvector (i.e. noise related: , see Section 2.2 of the main text) should be approximately the same. This means that: hence, for *k > N* equation (S.9) can be rewritten as:

|  |  |
| --- | --- |
| ; with | (S.10) |

It follows that increasing the number of components, for *k < N* the IE function progressively decreases until *k = N,* where it reaches a minimum. Then, for *k > N* the IE assumes a slow growing trend [1]. Malinowski discovered an empirical function called IND-function, which seems to be more sensitive than the IE function in its ability to pick-up the proper number of components. The IND-function is defined as:

|  |  |  |
| --- | --- | --- |
|  |  | (S.11) |

It is similar to the IE by definition and reaches a minimum when the correct number of components are employed. However, it has been observed that in this function the minimum is more pronounced and can appear in a situation where the IE does not exhibit any minimum. More details about the IE and IND functions can be found in [1-3].

**2. Ridge Regression approximation**

The most popular ridge regression constructs an approximation of spectrum μ linearly with respect to the set of parameters **P**.

|  |  |
| --- | --- |
|  | (S.12) |

When linear regression is used, the unknown coefficients wj are obtained minimizing the Euclidean (L2) norm between the approximated and known theoretical XANES in the points of the training set:

|  |  |
| --- | --- |
|  | (S.13) |

In such an approach, coefficients **w** = (w1, w2, …, wk) are usually large and thus they decrease the quality of XANES prediction between the points of the training set. To avoid such large values, a regularization term is added to the minimization function. In eq. (S.14) this term is written as the sum of squares of coefficients for the linear approximation multiplied by a constant factor *C*.

|  |  |
| --- | --- |
|  | (S.14) |

The regularization procedure avoids the overfitting phenomenon and the best value of factor *C* is obtained via cross-validation procedure [4]. Approximation (S.12) can be further improved by extending structural parameters with their pairwise products, i.e. using quadratic regression:

|  |  |
| --- | --- |
|  | (S.15) |

where is a coefficient for the quadratic term .

1. **Fitting XANES Spectra using Indirect and Direct Approaches**
   1. **Adjust XANES simulations**

This introductory notebook helps to adjust convolution parameters of the theoretical spectrum relative to the experimental one. The user is supposed to provide a file with an experimental spectrum of a known compound and a file with the theoretical simulations for the corresponding structure. Function readSpectrum reads the file 'exp\_ground.txt' and then the user can specify the column numbers for energy, the values of the XANES spectrum, moreover, he canskip several first lines. In the next step, the function createPartialProject specifies which experimental file should be used and it allows the shifting procedure between the experimental and theoretical data if the simulation was performed in relative energy units. The theoretical spectrum is loaded by the function parseFdmnesFolder specifying the proper folder. When data are loaded, command *fitSmooth* initializes the interactive widget with sliders which can be adjusted for the proper convolution parameters. The button “Start auto fit” starts the automatic adjustment of all parameters which can be saved further to files.

* 1. **Create a project**

This notebook prepares the project file which will be used later by other parts of the program. The input files for this step are respectively the *xyz* formatted file of the molecule and the related file containing the experimental data. User should introduce some appropriate changes inside two functions – *moleculeConstructor* and *projectConstructor*. The function *moleculeConstructor* must have two arguments: *project* and *params*. Parameter *project* is the project in which the molecule will be used. This variable can be useful in cases where the process of creating a molecule depends on the settings of the entire project. Parameter *params* defines the geometric parameters that are used to create the molecule. These parameters are configured by the user and correspond to the sliders in the fitting spectrum function *fitBySliders*.

The body of the molecule constructor must describe the creation of the molecule, considering the parameters passed to the constructor. Function *setParts* performs grouping of atoms into parts which will be moved or rotated as a rigid fragments:

m.setParts('0','1-9','10-19','20-29','30-38','39-48','49-58')

This notation specifies that atom 0 belongs to part 0, atoms 1-9 belong to part 1, atoms 10-19 belong to part 2 and so on. It is worth noting that the numeration in Python starts always from 0 instead of 1. In order to have access to every part of the molecule, the user should specify its indices (number of the part) – *m.part[index]*. By default, all atoms in a molecule belong to part 0. If you try to access to some part > 0 without *setParts*, you would get an index out of range error. The coordinates related to an individual atom can be accessed from the list of all atoms in the molecule or from the list of atoms inside the individual part. For example, coordinates of the twelfth atom (python index 11), which belongs to the third part (python index 2), can be accessed by two equivalent commands *m.atom[11]* or *m.part[2].atom[1]*. The whole molecule or its part can be shifted or rotated:

m.rotate(axis, center, angle)

m.part[0].rotate(axis, center, angle)

m.shift(shift)

m.part[0].shift(shift)

The *Axis* command specifies the axis for rotation. It can be written as a vector in squared brackets, e.g. axis = [0,1,0] or using coordinates of atoms, e.g. axis = m.atom[0]; moreover, given two axis, it is possible to definetheir differences and even their vector product as axis = cross(v1, v2). Each deformation is described with a set of commands:

deformation = 'sideRings2\_Elong'

part1 = 5; part2 = 6

axis1 = m.atom[39] - m.atom[0]; axis1 = axis1 / norm(axis1)

axis2 = m.atom[49] - m.atom[0]; axis2 = axis2 / norm(axis2)

m.part[part1].shift(axis1\*params[deformation])

m.part[part2].shift(axis2\*params[deformation])

The user sets the name of the deformation that will be assigned to the slider name, which visualizes the changes in the spectrum along with this deformation. The following are the numbers of parts of the molecule that will move. Axes are indicated for the direction of movement or for the rotation. Finally, the commands are given for the movement of each element of the parts. The above block is repeated for each deformation of the molecule. We usually work with two till to six deformations. The second function in this notebook is the project constructor. It returns an instance of the project. The project contains settings for performing various tasks and allows one to combine them in one place.

* 1. **Calculating XANES**

After the construction of the project file, the training sample can be calculated. The appropriate functions of PyFitIt library can be found in 'Calculate XANES for a set of geometries.ipynb' example notebook. This notebook is used to load the information about the project and to generate the input files for all structures in the training sample. The function *generateInputFiles* in user define folder creates subfolders for each sampling point in the structural parameters space. The number of points for each calculation is specified by the variable *sampleCount*, and the user can select one of the three methods to distribute points in the space of structural parameters: method = 'IHS', 'grid' or 'random' (see Figure 4 of the main text for details). We used the free software FDMNES for calculations (which should be installed and added to the PATH environment variable). Parameters for calculations by FDMNES are taken from project parameter FDMNES\_calc. Function *calcSpectra* starts the sequential or parallel calculation of spectra for generated input files. Calculations can be also executed remotely. To proceed, the user should generate a script file from this notebook by the execution of the command *saveAsScript* or by saving it manually from the main menu through the command File->Download as -> Python.py. The function *collectResults* creates two files from the separate calculations. These are composed of a first file characterized by all theoretical spectra and a second file with the corresponding structural parameters.

* 1. **Structure analysis**

The fitting of structural parameters for an experimental spectrum requires two input files with the associated calculated spectra and the corresponding structural parameters, which were generated in steps of section 3.2. Users can select between different options: the automatic search of minima (notebook Inverse approach), working with sliders (notebook Fitting XANES by sliders) or making the prediction of the structural parameters via the direct method (notebook Direct approach). Furthermore, the corresponding example notebooks with the additive *diff* in a name allows using difference spectra for fitting.

The following common logic is used for these notebooks. First, a training sample is loaded using function *readSample*. Next, the convolution parameters are specified in order to perform the automatic fit. In the manual fitting procedure, these parameters can be varied along with structural parameters. Function *constructInverseEstimator* creates the inverse estimator class instance. We implemented an option to compare the quality of different approximation algorithms. This step is executed by the method *compareDifferentMethods* of the class *inverseEstimator*. This method creates a line of points in the space of structural parameters. Function *generateInputFiles* generates a set of points along the line in the multidimensional space to check the quality of the approximation among different methods. This line is automatically constructed between the initial point and a point with the largest deviations referring to all structural parameters. In the next step, the machine learning algorithm must be trained on the set of calculated spectra using method *fit*: *inverseEstimator.fit(sample)*. Finally, method *findGlobalL2NormMinimum* searches for a specified number of minima of the L2 norm. Structural parameters for the best-fit and contour plots for the pairs of structural parameters are saved to files.

In the notebook for the direct approach, the class *directEstimator* is used to train the machine learning method and predict the structural parameters.

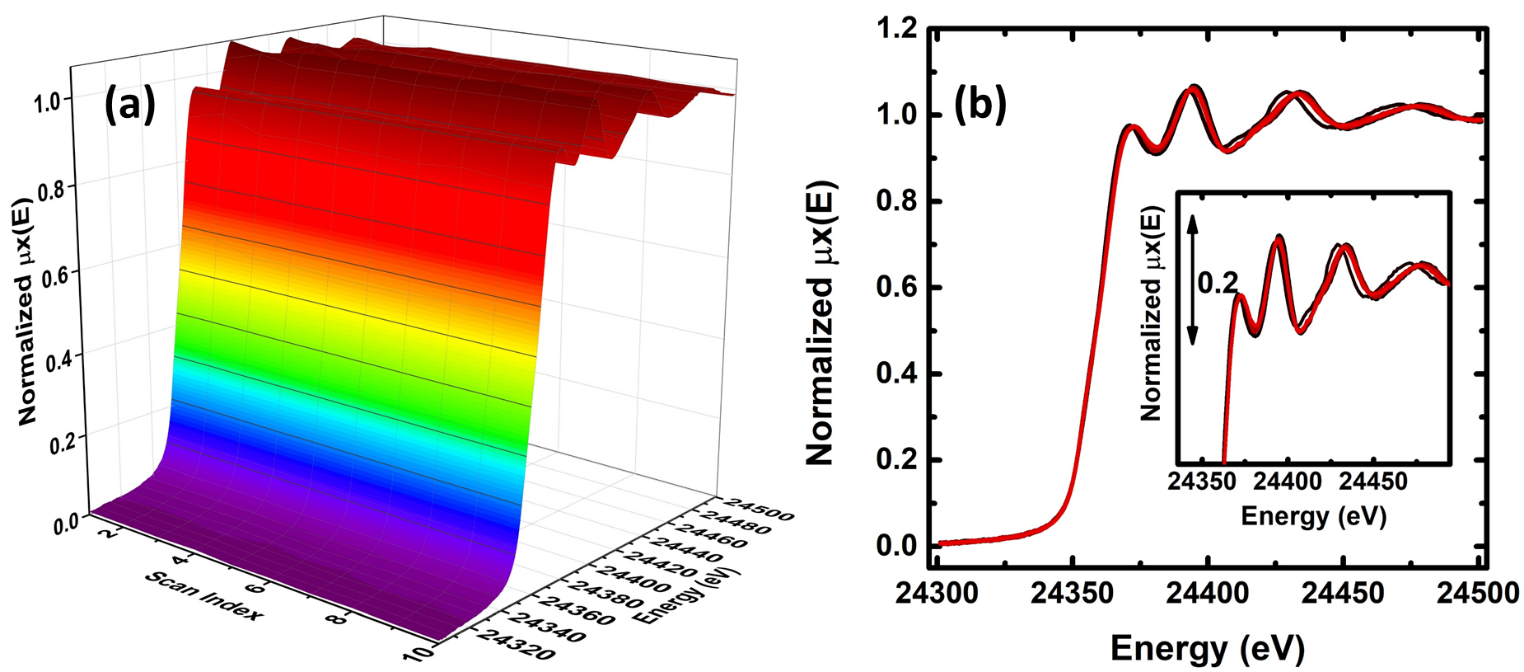
1. **Case of Studies**
   1. **Experimental Setup details for Ce L3 XANES in CeO2/Pt system under redox conditions**

The Si(111) channel-cut monochromator provided an energy resolution of ΔE/E = 2.0×10-4. High-surface-area ceria nanoparticles promoted with 1.5 wt% Pt [5] were sieved to 100-150 μm grain size and placed into an *in situ* plug-flow reactor cell [6]. The reactor was connected to the reductive (5% CO in argon) and oxidative (21% O2 in argon) gas lines and fast electrovalves provided switching between these mixtures. Due to the high concentration of the heavy elements in the substrate the fluorescence regime of measurements was selected. We utilized the time-resolved scheme described in detail in [7].

* 1. **Procedure Spectral decomposition: The case of Pd-carbide and hydride phases formation**

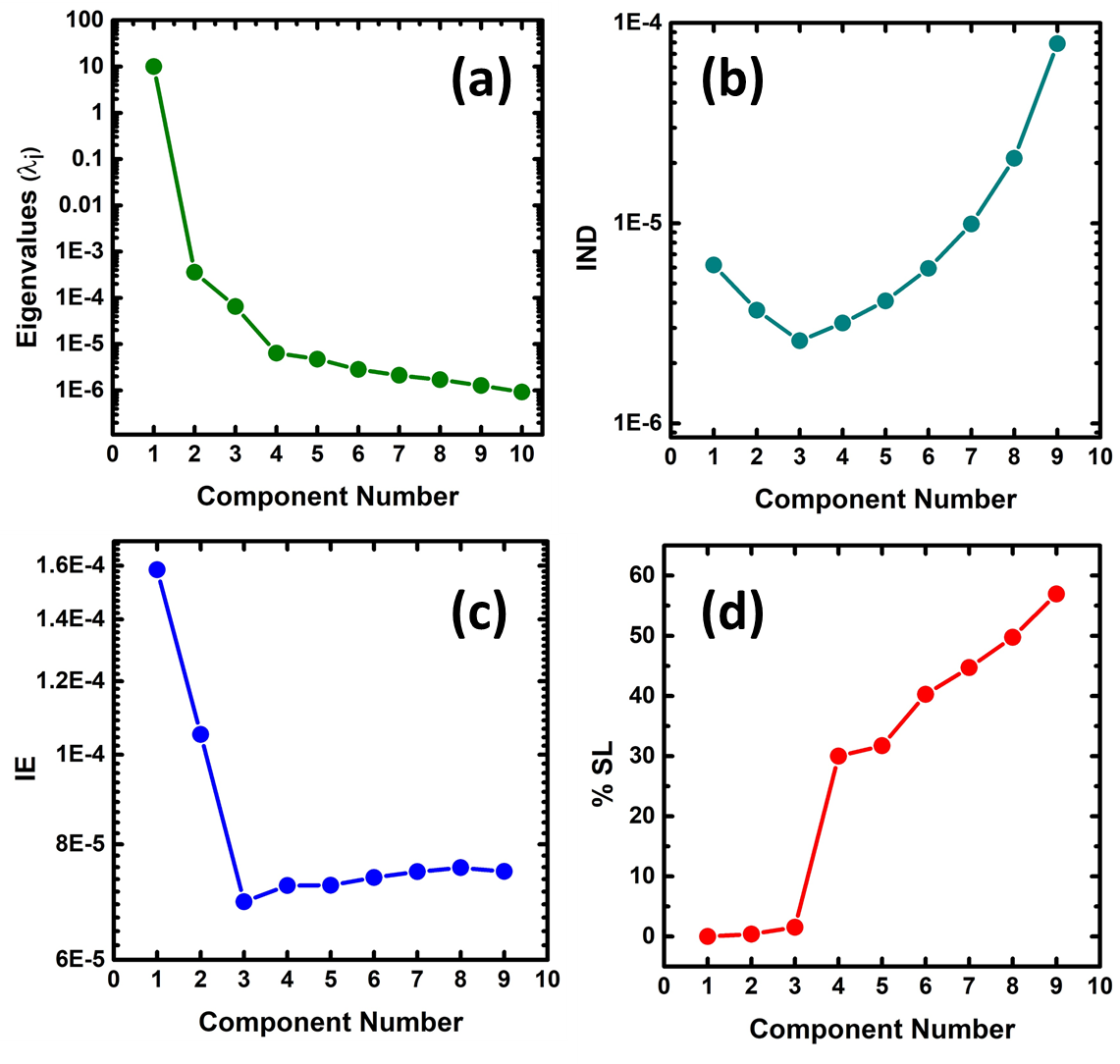
In this section, we show an example of PCA applied for a series of spectra with three independent components. Experimental X-ray absorption spectra were collected at the BM31 beamline of the European Synchrotron Radiation Facility (Grenoble, France) [8]. The main purpose of this experiment consisted to track the palladium carbide and hydride phases’ formation in the bulk and at the surface of the palladium nanoparticles [9].

The sample powder constituted by 5% of the Pd/C catalyst in form of palladium nanoparticles was stored inside a 1.5 mm thick glass capillary connected to a vacuum line allowing to control the partial pressures of acetylene and hydrogen inside it. Pd K-edge spectra were acquired in transmission mode in the energy range of 24.1−25.4 keV using ionization chambers and a double-crystal Si(111) monochromator detuned to 80% in the continuous scanning mode. Initially, the sample was activated in pure H2 at 120 °C, successively evacuated, and cooled down to 100 °C to obtain the spectrum of the clean nanoparticles (pure metallic phase). Then it was exposed to a mixture of acetylene and hydrogen with partial pressure of 350 and 650 mbar, respectively. This step showed the appearance of the palladium hydrides (Pd-H). In order to form the palladium carbide phase (Pd-C), the sample was exposed to pure acetylene, and three consecutive XAS spectra were taken. The stability and reversibility of the carbide phase were checked first by evacuation down to 10-3 mbar at 100 °C and then by treatment in hydrogen (1000 mbar at 100 °C), that caused the partial remotion of Pd-C species, with subsequent evacuation. Then, the sample was exposed again to pure acetylene at 100 °C determining a further grown of the carbide phase. The final dataset, represented in Figure 1 (a,b). It is constituted, in this way, by 10 spectra characterized by 199 energy points.

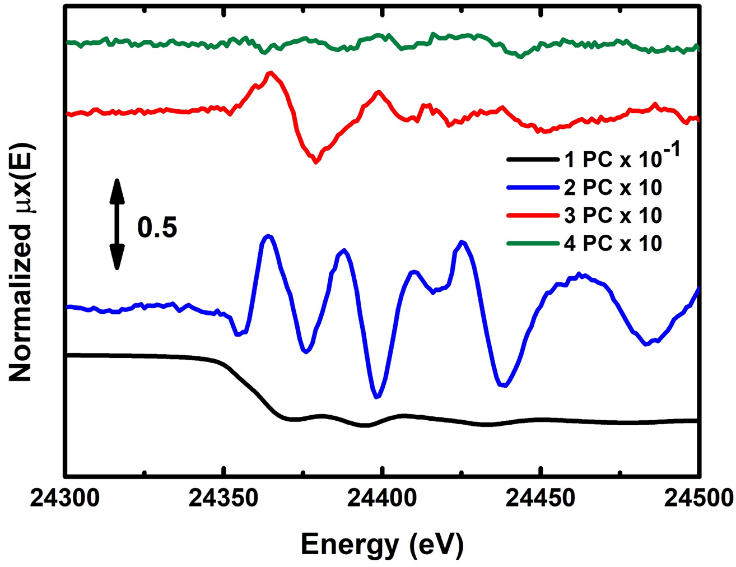


**Figure 1:** (a) 3D (energy, scan index vs μx(E)) and 2D (energy vs μx(E)) representation of Pd dataset composed by 10 spectra used as input for the pyFitIt (PyPCA) spectral decomposition.

As for the ceria dataset, reported in Section 4.1 of the main text, the entire sequence of XANES scans has been normalized using the equation (S.6). The identification of the correct number of pure species seems to be driven towards three pure components by all the statistical estimators. In the scree plot Figure .2(a), a well-defined elbow falls in the proximity of the third component. The same result is confirmed by the IE, IND, and F-test, see Figure 2(b,c,d). The first two show a minimum sited in correspondence of the third component, at the same time, in the %SL plot the number of points located below the fixed level of significance of 5% is three, suggesting the existence of three species. Finally, the plot of the abstract components (Figure 3) shows a fourth component (4 PC) characterized principally by noise and without any particular features if compared to the other three abstract components.



**Figure 2:** Statistical curves used for determining the correct number of principal components (i.e. the correct number of pure spectra in the dataset) referring to the Pd-dataset: (a) Scree Plot; (b) Percentage of significance level; (c) IND and IE factor.

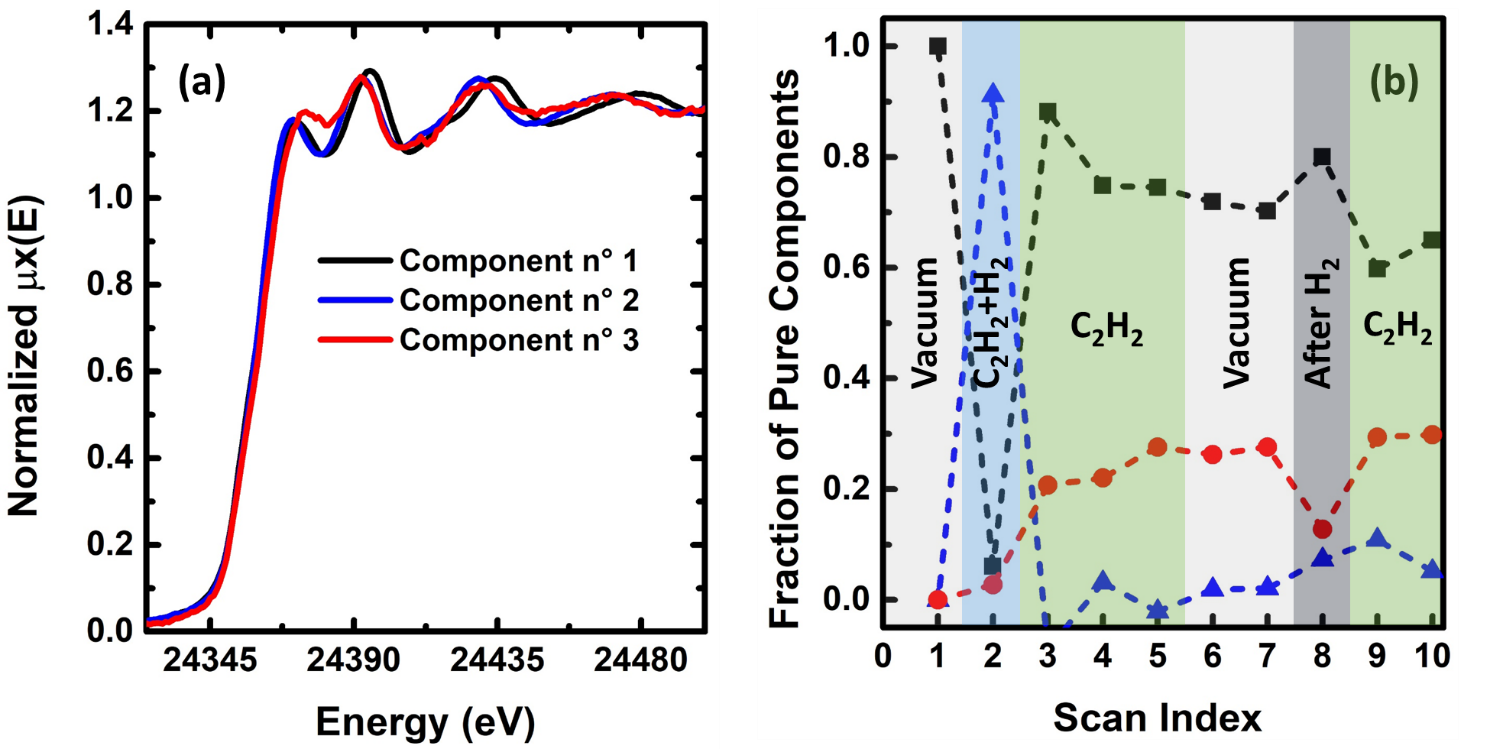


**Figure 3:** Abstract Spectral Profiles (PCs) extracted from the Pd dataset. How it is possible to see the fourth component is noisier if compared with the other abstract components that, on the contrary, are characterized by well-defined features.

Requiring the existence of three pure species, we introduced the following transformation matrix:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Herein, we used two constrained (i.e: four sliders to move): the first one was the common normalization of the spectra, the second one consisted to fix the first spectrum, considered to be pure because associated with the metallic phase of Pd. Sliders were moved in the [-5,5] range with a step of 0.05. As for the Ce case, the spectra and concentration non-negativity and the mass balance conservation condition were used to select the feasible slider values. The resulting pure spectra and concentration profiles are reported in Figure 4 (a,b).



**Figure 4:** Pure Spectra (a) and Concentration profiles (b) recovered from the Pd dataset by PyFitIt (PyPCA) which correspond to pure metallic (black), hydride (blue) and carbide (red) phases of palladium.

The three isolated spectra were assigned to a pure metallic, hydrate and carbide phase of palladium respectively, see Figure 4(a,b) . How it is possible to see from the concentration profiles, Figure 4(b), the initial state is represented by the pure metallic state of metallic Pd. Then, sending on the sample a mixture of acetylene and hydrogen (with the partial pressure described before) it is possible to assist in the Pd-H phase formation. After the exposure to acetylene, pure metallic and carbide phases coexist in the nanoparticles. At long time scales, the metallic phase shows a decreasing trend, and the opposite behaviour is observed for the carbide one. After cleaning in hydrogen, a drop in the Pd−C component and an increase of the pure metallic phase are visible. It is interesting to note that the Pd−C component was not observed, in the pure state, in any of the experimental conditions. It should be considered as a palladium carbide phase which might be formed after an infinitely long exposure to acetylene [9].

* 1. **Structural refinement of the Fe(terpy)2 excited state**

|  |  |
| --- | --- |
|  |  |
| **Figure 5:** Left panel: calculated Fe K-edge XANES spectra for Fe-N axial and equatorial distances set as following from DFT simulations. For 5E state both Fe-N axial distances were set equal to 2.1 Å and all equatorialdistances equal to 2.20 Å [10]. For 5B2 state Fe-N axial distances were set 2.16 Å and equatorial distances – 2.19 Å. Right panel: the differences were calculated with respect to crystallographic structure. | |

**Bibliography**

[1] E.R. Malinowski, Determination of the number of factors and the experimental error in a data matrix, Anal. Chem., 49 (1977) 612-617.

[2] R.G. Brereton, Chemometrics: data analysis for the laboratory and chemical plant, John Wiley & Sons, 2003.

[3] E.R. Malinowski, Factor analysis in chemistry, Wiley, 2002.

[4] R. M. Rifkin, R. A. Lippert, Notes on Regularized Least-Squares, Computer Science and Artificial Intelligence Laboratory, Technical Report, (2007) 8.

[5] R. Kopelent, J.A. van Bokhoven, J. Szlachetko, J. Edebeli, C. Paun, M. Nachtegaal, O.V. Safonova, Catalytically Active and Spectator Ce3+ in Ceria-Supported Metal Catalysts, Angewandte Chemie-International Edition, 54 (2015) 8728-8731.

[6] G.L. Chiarello, M. Nachtegaal, V. Marchionni, L. Quaroni, D. Ferri, Adding diffuse reflectance infrared Fourier transform spectroscopy capability to extended x-ray-absorption fine structure in a new cell to study solid catalysts in combination with a modulation approach, Review of Scientific Instruments, 85 (2014).

[7] A.A. Guda, A.L. Bugaev, R. Kopelent, L. Braglia, A.V. Soldatov, M. Nachtegaal, O.V. Safonova, G. Smolentsev, Fluorescence-detected XAS with sub-second time resolution reveals new details about the redox activity of Pt/CeO2 catalyst, J. Synchrot. Radiat., 25 (2018) 989-997.

[8] W. van Beek, O.V. Safonova, G. Wiker, H. Emerich, SNBL, a dedicated beamline for combined in situ X-ray diffraction, X-ray absorption and Raman scattering experiments, Phase Transit., 84 (2011) 726-732.

[9] A.L. Bugaev, O.A. Usoltsev, A.A. Guda, K.A. Lomachenko, I.A. Pankin, Y.V. Rusalev, H. Emerich, E. Groppo, R. Pellegrini, A.V. Soldatov, J.A. van Bokhoven, C. Lamberti, Palladium Carbide and Hydride Formation in the Bulk and at the Surface of Palladium Nanoparticles, J. Phys. Chem. C, 122 (2018) 12029-12037.

[10] G. Vanko, A. Bordage, M. Papai, K. Haldrup, P. Glatzel, A.M. March, G. Doumy, A. Britz, A. Galler, T. Assefa, D. Cabaret, A. Juhin, T.B. van Driel, K.S. Kjaer, A. Dohn, K.B. Moller, H.T. Lemke, E. Gallo, M. Rovezzi, Z. Nemeth, E. Rozsalyi, T. Rozgonyi, J. Uhlig, V. Sundstrom, M.M. Nielsen, L. Young, S.H. Southworth, C. Bressler, W. Gawelda, Detailed Characterization of a Nanosecond-Lived Excited State: X-ray and Theoretical Investigation of the Quintet State in Photoexcited Fe(terpy)(2) (2+), J. Phys. Chem. C, 119 (2015) 5888-5902.