Electronic Supplementary Material (ESI) for Nanoscale.
This journal is © The Royal Society of Chemistry 2016

Supplementary to:

The Mendeleev-Meyer Force Project

Sergio Santos[1], Chia-Yun Lai[1], Carlo A. Amadei[2], Karim R. Gadelrab[3], Tzu-Chieh Tang[4], Albert Verdaguer[5], Victor Barcons[6], Josep Font[6], Jaime Colchero[7], Matteo Chiesa[1]

[1]Laboratory for Energy and NanoScience (LENS), Institute Center for Future Energy (iFES), Masdar Institute of Science and Technology, Abu Dhabi, UAE

[2]John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138, USA

[3]Department of Materials Science and Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, Massachusetts 02139, USA

[4] Department of Biological Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, Massachusetts 02139, USA

[5]ICN2 – Institut Catala de Nanociencia i Nanotecnologia, CSIC – Consejo Superior de Investigaciones Cientificas, ICN2 Building, Campus UAB, 08193 Bellaterra, Barcelona, Spain

[6]Departament de Disseny i Programació de Sistemes Electrònics, UPC - Universitat Politècnica de Catalunya, Av. Bases, 61, 08242 Manresa (Barcelona), Spain

[7]Instituto de Óptica y Nanociencia (CIOyN), Campus Espinardo, Universidad de Murcia, E-30100 Murcia, Spain

Contents

Generation of models

The data was acquired from standard force versus distance curves (FDC) and imported into Matlab[1] to be processed.

Then distances from the well of the curve were measured and saved as input matrices and then normalized
as detailed in the main text. The process is described in detail in the main text and also in previous studies[2, 3].

The feature libraries in Tables I, II and II were produced by acquiring 100-1000 data points (FDCs) per sample and averaging over 40-100 data points or curves. The averages of the input features are shown in the tables for the normalized distances dF_i (i=1 …8) and were used as input features to train the artificial neural network.

The generated models consist of L layers and U unit cells as illustrated in in Fig. 1b in the main text. The output of the models consist of a K by M matrix were K is the number of substances (or families) to be identified by the model and M the number of examples to be tested by the model. K is also the number of unit cells in the last layer of the network and each cell produces the outcome for a given sample or prediction. The models were generated in Matlab following the notes of Prof. Andrew Ng[4].

In this model, the cost function J to be minimized can be written as

$$J(\Theta) = -\frac{1}{M}\left[\sum_{i=1}^{M}\sum_{k=1}^{K} y_k^{(i)} \log\left[h_\theta(x^{(i)})\right]_k + (1-y_K^{(i)})\log(1-\left[h_\theta(x^{(i)})\right]_k\right] - \frac{1}{2M}\sum_{l=1}^{L-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l+1}\left[\Theta_{ji}^{(l)}\right]^2 \quad \text{(S1)}$$

Where the matrix Θ consists of the regressors to be found for the logistic regression hypotheses of each neuron or unit cell U, $y^{(i)}$ stands for example (input feature) number i. Here an input feature is a particular normalized distance dF as illustrated in Fig. 1a(iii) and provided in the three tables in the main text. M is the number of examples to be employed to train the model, K is the number of outputs in the model, l stands for Layer, s_l is the number of neurons in layer l, L is the number of layers in the model, and h_θ is a hypothesis from logistic regression for the particular unit cell and layer.

The second term in (S1) consists of the regularization term that assists in avoiding overfitting of the parameters Θ. Large numbers of λ however might lead to underfitting as observed in the main text in Fig. 2 where the F-score is zero for the larger values of λ. In the main text we took $\lambda=10e^{-5}$ for all our models since it was the largest value of λ that was consistently giving us the largest F-score, i.e. 1, in our models.

The models were trained with training sets and cross-validated with cross validation sets, i.e. data that was not employed to generate the models. This was done to avoid overfiiting and is a

standard procedure wen implementing neural networks. All the F-score values given in the main text were produced with cross validation or test sets.

The hypotheses for each neuron or unit cell are written in terms of the Sigmoid function as $h_\theta(z)$

$$h_\theta(z) = g(z) \tag{S2}$$

Where Z is constructed as the scalar product of the input of the neuron and the regressors of the neuron as

$$z = \vec{x} \cdot \vec{\theta} \tag{S3}$$

The function g is the sigmoid function as previously employed by others in atomic force profiles[5]

$$g(z) = \frac{1}{1+e^{-z}} \tag{S4}$$

The minimization of (S1) was carried out with the standard functions optimset and fminunc from Matlab[1] (to find the coefficients θ) and the backpropagation algorithm.

Part of Fig. 3 in the main text is reproduced below as Fig. S1 where an extra figure (Fig. S1e) has been added to show the predictions of a 3L with 4U model.

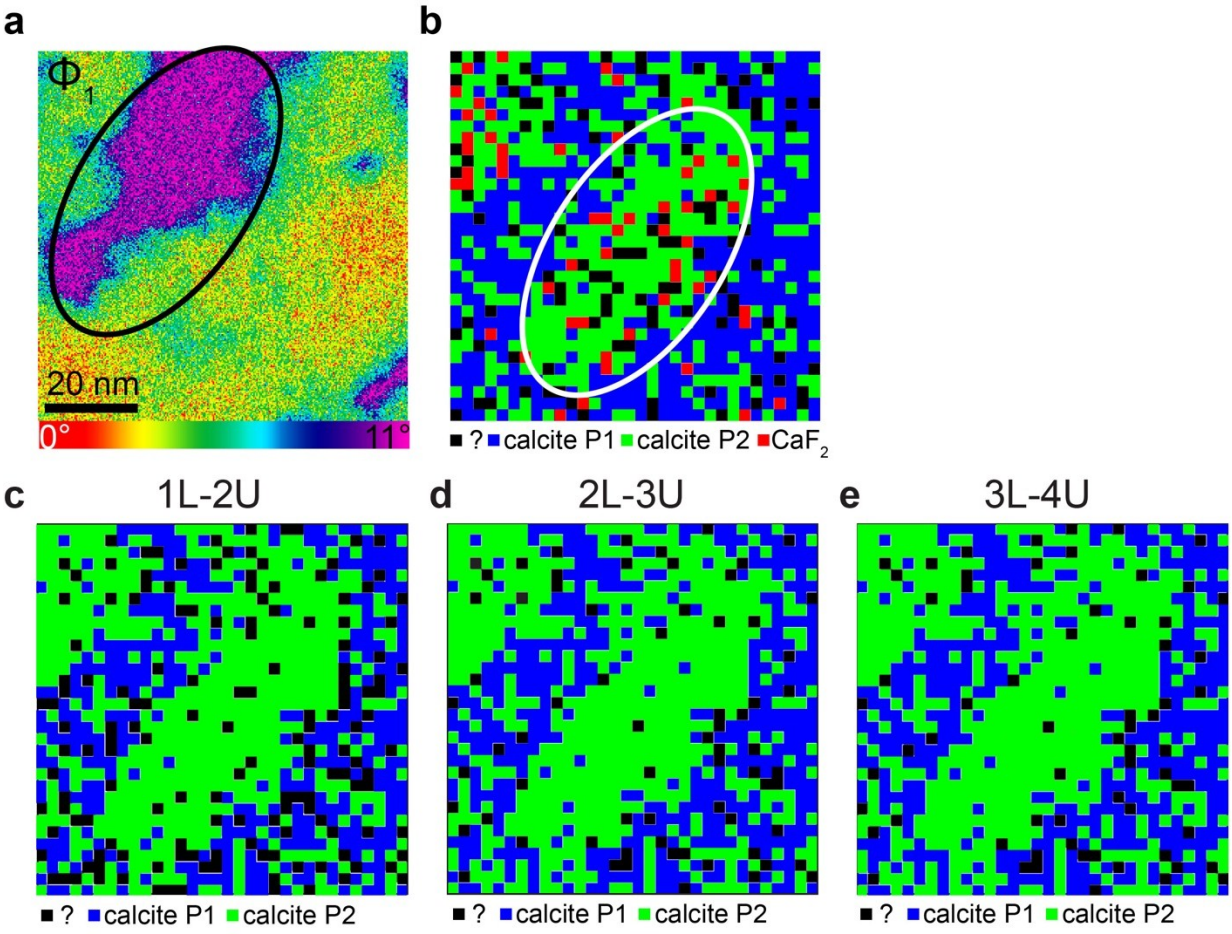

Figure S1. Two phases of calcite P2 (pink-purple) and calcite P1 (rest of the image) acquired as a standard phase image in dynamic AFM. **b**, Prediction of the model produced from a feature library consisting of calcite P1 (blue), calcite P2 (green) and CaF_2 (red). The black pixels refer to pixels where the model could not predict any output unambiguously. The two images where generated in approximately the same spot (80 nm^2) but some thermal-drift is observed. **c-e**, Predictions of the models produced from a feature library consisting of calcite P1 (blue), and calcite P2 (green) only for the same raw data as b. The black pixels refer to pixels where the model could not guess any output unambiguously. The results are shown for models consisting of (**c**) 1L-2U and (**d**) 2L-3U and (**e**) 3L-4U.

Data, models and codes

Video instructions:

We have 5 videos with instructions that explain how to use and reproduce the data and codes employed

All the data, models and codes employed in this work can be found in the (confidential and private) repositories

Dropbox:

Account: TMMFProject@gmail.com

Password: N@noscale123

Github:

Account https://github.com/TMMFProject/TMMFProject

Password: N@noscale123

The Project will be hosted and maintained by www.future-synthesis.com

The project aims to be open source and be extended into dedicated databases and search engines where the data here, and future data, algorithms and findings will also be open source mimicking the databases and search engines typically employed in bioinformatics assisted biology, i.e. MEDLARS or PRIDE.

Reference

1. R. a. Matlab and Simulink, The MathWorks, Inc. Natick The MathWorks, Inc. Natick 2010.
2. C.-Y. Lai, T. Olukan, S. Santos, A. Al Ghaferi and M. Chiesa, *Chem. Commun.*, 2015, 51, 17619-17622.

3. C.-Y. Lai, T.-C. Tang, C. A. Amadei, A. J. Marsden, A. Verdaguer, N. Wilson and M. Chiesa, *Carbon*, 2014, 80, 784-792.
4. Instructor: Andrew Ng, https://www.coursera.org/instructor/andrewng.
5. F. Lo Iacono, N. Bologna, M. V. Diamanti, Y.-H. Chang, S. Santos and M. Chiesa, *The Journal of Physical Chemistry C*, 2015, 119, 13062-13067.