HORIZON2020 European Centre of Excellence
Grant Agreement n. 824143

Deliverable D5.7
Final report on results verification and validation of codes and on the data analytics pilots

D5.7

Final report on verification and validation of codes and on the data analytics pilots

Nicola Marzari, Marnik Bercx, Michele Ceriotti, Stefaan Cottenier, Stefano de Gironcoli, Thierry Deutsch, Alessandro Laio

Due date of deliverable: 31/01/2022
Actual submission date: 01/02/2022
Final version: 01/02/2022

Lead beneficiary: EPFL (participant number 6)
Dissemination level: PU - Public

Document information

Project acronym:	MaX
Project full title:	Materials Design at the Exascale
Research Action Project type:	European Centre of Excellence in materials modelling, simulations and design
EC Grant agreement no.:	824143
Project starting / end date:	01/12/2018 (month 1)/31/05/2022 (month 42)
Website:	www.max-centre.eu
Deliverable No.:	D5.7

Authors: Nicola Marzari, Marnik Bercx, Michele Ceriotti, Stefaan Cottenier, Stefano de Gironcoli, Thierry Deutsch, Alessandro Laio

To be cited as: Nicola Marzari, Marnik Bercx, Michele Ceriotti, Stefaan Cottenier, Stefano de Gironcoli, Thierry Deutsch, Alessandro Laio, (2022): Final report on verification and validation of codes and on the data analytics pilots. Deliverable D5.7 of the H2020 project MaX (final version as of 01/02/2022). EC grant agreement no: 824143, EPFL, Lausanne, Switzerland.

Disclaimer:

D5.7 Final report on results verification and validation of codes and on the data analytics pilots

1. Executive Summary

Key objectives of work package 5 ("Ecosystem for HPC, HTC and HPDA convergence") are the identification of protocols for the verification of materials science codes, and the development of algorithms and tools for high-performance data analytics in materials space. In this respect, we:

- have developed a comprehensive protocol to verify any density-functional theory (DFT) code under a broad range of diverse chemical environments, using automated AiiDA common workflows (https://www.nature.com/articles/s41524-021-00594-6). All the material is available at https://github.com/aiidateam/aiida-common-workflows, and involves a large community.

- have delivered 4 pilot projects on high-performance data analytics (HPDA), in the form of

  1) two accurate predictors targeted pre-exascale architectures, for the optimization of parallelization parameters, and for the prediction of the computational time-per-call;

  2) a configurational explorer https://chemiscope.org, integrated into the Materials Cloud: https://chemiscope.materialscloud.io. This is complemented by novel data algorithms for kernel principal covariate regression (also accessible online on the Materials Cloud at https://www.materialscloud.org/discover/kpcovr), and in the data analysis software DULy, with algorithms for intrinsic dimension estimation, density estimation, density-based clustering and metric comparison;

  3) automated, turn-key quantum mechanical simulations, resulting in 36,000 inorganic crystal structures published on the Materials Cloud (https://www.materialscloud.org/discover/mc3d), and a hero run on the new pre-exascale machine LUMI (LUMI-C, the currently available CPU version) running for 12 hours on 196,608 cores, with 55,704 Quantum ESPRESSO calculations orchestrated by AiiDA, fully optimizing the geometry of 15,324 compounds;

  4) demonstrated edge computing in the case of SARS-CoV-2 peptide inhibitors, integrating substrate and ligand binding data from X-ray and neutron crystallography postprocessed by BiGDFT calculations on thousands of structures (https://pubs.rsc.org/en/content/articlelanding/2021/SC/D1SC03628A).

2. Verification and validation of codes

2.1 The test set

The goal of this task is to perform a quality assessment of MaX codes in an automated fashion, exploiting AIIDA plugins and workflows according to established community protocols, as well as to provide curated datasets of crystal structures, pseudopotentials and other relevant input that can enable the turn-key workflows. A minimal requirement for any simulation is that it must be verified: further improvement of numerical approximations should not alter the result. Five years ago, the most stringent test ever for the verification of electronic structure methods and codes was published, comparing the results of 40 different flavours of electronic structure codes on a test set of 71 elemental crystals. In the present project, we undertook the more ambitious attempt of verifying electronic structure codes in more diverse and realistic circumstances. Rather than examining 71 elemental crystals, we consider a test set of 570 crystals, selected or created with the aim of covering as much as possible chemical diversity in a set that is overall as small as possible. We selected 6 prototype oxides, that should fulfil the following criteria:

- The unit cell is cubic.
- The unit cell does not contain atom sites with symmetry-conserving degrees of freedom.
- There is a unique lattice site available for one other element that is not oxygen.
- The stoichiometry of these 6 crystals is such that the formal oxidation state of the other element covers all positive integers from +1 to +6.

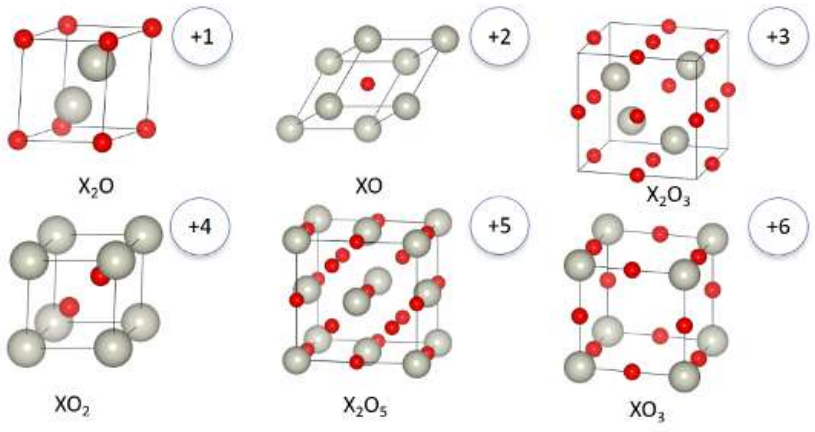

Figure 1: *The 6 prototype structures chosen for the verification and validation of codes.*

The figure shows these 6 prototypes. The grey/large atom site is populated subsequently by all elements from hydrogen (Z=1) to Americium (Z=95). This results in 6x95=570 crystals, with the volume being the only geometrical degree of freedom, and with the non-oxygen atom being in 6 different formal oxidation states. By calculating the total energy (E) as a function of the volume (V) by an electronic structure code, the equation of state or E(V)-curve is obtained. Doing this for two different electronic structure codes, allows to inspect differences between the curves. If both codes give exactly the same E(V)-curve, the differences are zero and the codes are concluded to give identical results (for this test set). This procedure of pairwise comparison can be performed to any

number of methods and codes. The present test is particularly suited to detect imperfections in pseudopotentials (a formalism used by a major class of codes where the interaction between the nucleus and the electrons in the region deep inside the atom is replaced by an interaction between a user-constructed potential and the electrons, where the potential is made in such a way that it does not alter the behaviour of the electrons outside this inner region and leads to much faster computations). Codes that make use of pseudopotentials typically have a library of pseudopotentials, one for every element. The potentials have been tested to yield results that are nearly identical to the ones given by all-electron codes (where no pseudopotential is used). These tests are typically done on a limited set of standard situations, and may not capture the diversity of chemically different environments these atoms may end up in. The present 6 oxides provide 6 very different and sometimes exotic chemical environments, and offer in this way a strong and independent test of the quality of a pseudopotential.

2.2 The protocol

In order to minimize as much as possible ambiguities to run these calculations, a strict protocol has been agreed upon, and has subsequently be implemented in automated workflows (see also item 2.4):

- For each of the 570 crystals, a volume is singled out that is within 0.5% of the equilibrium volume for the given crystal (for the XC-functional for which the test will be run, which is in this case PBE). This will be called the reference volume for this crystal from here on. It has no particular physical meaning, and its exact value does not matter as long as it is sufficiently close to the exact (and at this stage not precisely known) equilibrium volume.
- For every electronic structure code, 7 total energies are calculated, at 7 different volumes: +6%, +4%, +2%, 0%, -2%, -4% and -6% away from the reference volume. This ensures that every code calculates the total energies at exactly the same volumes, with the minimal energy lying close to the central (reference) volume.
- It is the responsibility of the person who runs the test to make sure that the code settings that determine the numerical precision are sufficient to reduce the numerical noise below a threshold that is deemed to be acceptably small.

2.3 First exploratory run

In order to explore the relevance of this procedure, we apply it first for 3 different electronic structure methods/codes:

- WIEN2k: an all-electron code (i.e., no pseudopotential choice has to be made) that is considered to be numerically one of the most precise codes available.
- VASP with the std potential library: a code based on the PAW formalism, that involves a concept similar to the pseudopotential. Every element comes with a unique potential, constructed by the code developers, and collected in a library. The ‘std’ library was the default library up to a few years ago.

- VASP with the GW potential library: the GW library has been introduced a few years ago, and is claimed to be better (=more precise) than the std library.

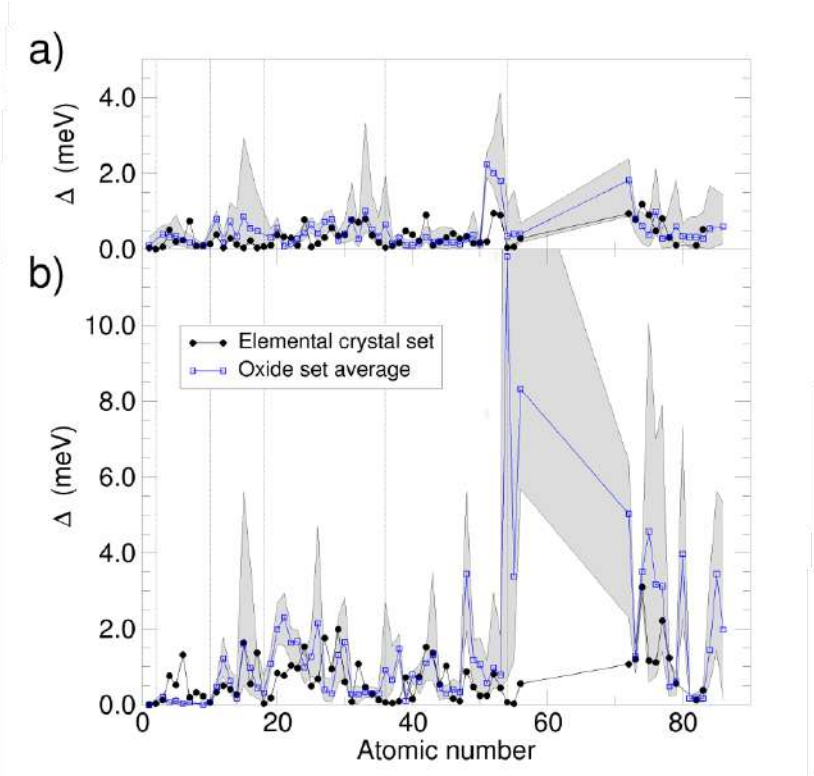

Figure 2: *verification of VASP-GW against WIEN2k.*

The upper part of the picture shows the level of agreement between WIEN2k and VASP-GW, expressed as a quantity D that is zero when two E(V)-curves perfectly coincide. The black dots give this D for every element as part of a unary crystal for the previous test set of 71 crystals. The blue open squares are the average D for all 6 oxides based on that same element, with the grey area indicating the spread over these 6 oxides. For some elements there is no data because there are no VASP-GW potentials available for them. This picture shows that the D-values for the oxide set are similar in magnitude as for the 71-crystal set. This implies that the conclusion reached in the past – VASP-GW potentials allow for the same level of precision as all-electron WIEN2k – is further corroborated by the new test.

This is different for the comparison VASP-std vs. VASP-GW, which is reported in the lower part of the figure. They were in reasonable agreement for the 71-crystal set (black dots, small D). For the oxide set, however, there are elements with a very large D. This indicates that the low D observed for those elements in the 71-crystal set was fortuitous. The VASP-std set considerably underperforms with respect to VASP-GW and WIEN2k.

For all oxides for which VASP-GW calculations were performed, the Hirschfeld-I charges were calculated. These are effective charges that tend to be close to the formal oxidation states, and that can therefore be used to find in a semi-quantitative way in which actual oxidation state the oxides end up (in HO3, the H-atom with its single electron will never appear in a +6 oxidation state, where 6 electrons should be removed from every H-atom).

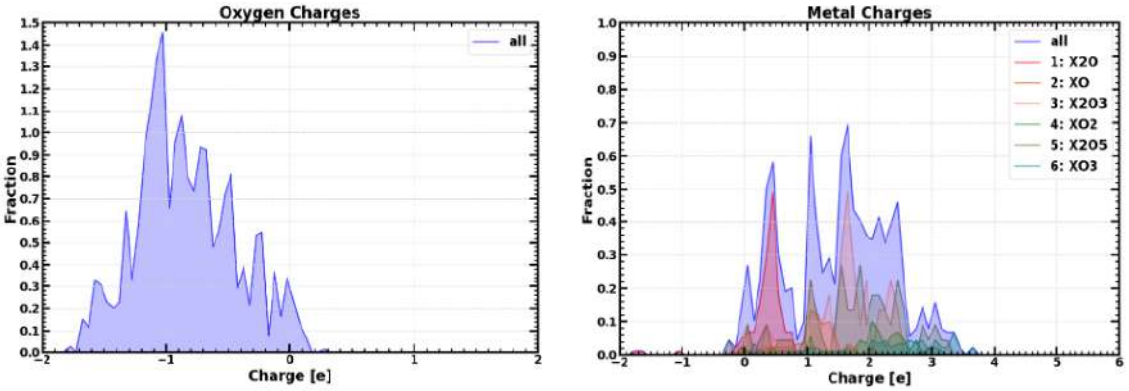

Figure 3: *Hirshfeld-I charges for the 570 oxides studied: oxygen (left) and the other elements (right).*

The Hirshfeld-I charges for the 570 oxides are shown in the figures for oxygen (left) and for the other elements (right). The oxygen charge should formally be always -2, but is spread over a wide range between -2 and 0, with the majority around -1. The charges for the non-oxygen element should be between +1 and +6, but are rather between 0 and +3. This illustrates the creative solutions the electronic structure finds to accommodate these atoms within an imposed structure that may in several cases not be stable in nature.

2.4 Automation towards all MaX codes

Having the test set and the protocol developed and tested, the next step is extending this analysis to all MaX codes. To this end, a "common workflow" turnkey solution has been developed for the task of running the oxide test. A user specifies the code-independent input in a generic interface, and this is translated into code-specific input for each of the tested codes. The AiiDA workflow manager takes care of running all calculations. Jupyter notebooks visualize the results for all codes and their pairwise comparison according to different measures for the difference. Figure 4 shows an example where two codes (Quantum ESPRESSO and Wien2K) produce E(V) curves that are nearly identical for a set of 6 oxides, whereas for another element the agreement is far from good – indicating a problematic pseudopotential for that element. All code is available at https://github.com/aiidateam/aiida-common-workflows, and the resulting paper at https://www.nature.com/articles/s41524-021-00594-6 .

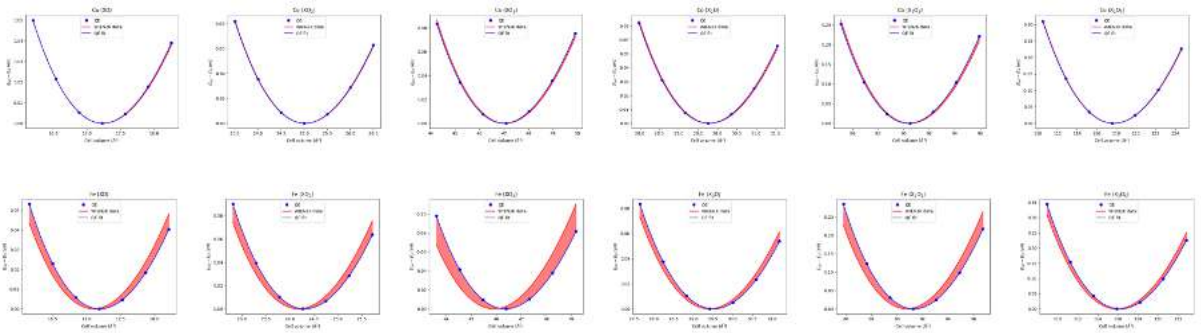

Figure 4: *Comparison of the equations of state for two different elements entering in the 6 oxide binaries/prototype structures of Figure 1.*

3. High performance data analytics pilots

3.1 Pilot 1: Predicting code performance

We worked on a pipeline that allows for the optimization of computational resources and the reduction of queue time for density functional theory (DFT) calculations run on high-performance computing facilities.

To this end, we developed two algorithms, one for the optimization of parallelization parameters, and the second for the prediction of computational time-per-call. Figure 5 illustrates a scheme of the proposed pipeline for the combined use of these two algorithms in a use-case scenario.

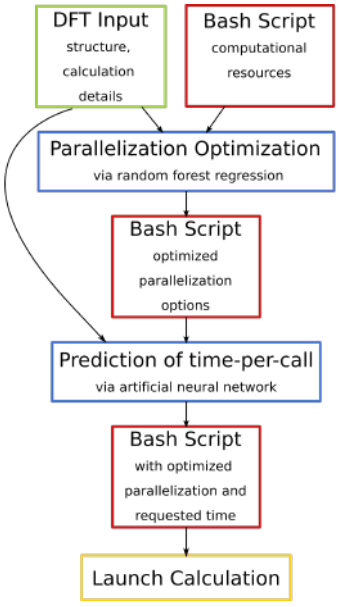

Figure 5*: Schematic representation of the pipeline for the optimization of parallelization parameters and requested time in DFT calculations. The algorithms developed are marked in blue, the input file containing non-optimizable parameters is in green, and the launch script on which the algorithms act is highlighted in red.*

When launching DFT calculations, choosing the optimal parallelization parameters (number of threads, number of pools) can have a dramatic effect on the efficiency, and therefore the cost and energy consumption, of the calculation.

We developed an algorithm based on random forest regression that informs the optimal choice for the number of threads and pools for DFT computations, as a function of the computational resources available, and of the computation input parameters (number of electrons, number of k points, etc.). The algorithm was trained on a set of 1500 calculations run for the case study of Quantum ESPRESSO on the Marconi cluster for 5 chemical systems and with varying system sizes, computational resources, and parallelization options. The algorithm is able to choose the optimal number of threads and pools in 88% of cases for validation sets disjointed from the training set in a 10-fold cross validation setup. In the 12% of cases where the algorithm does not choose the optimal

number of threads and/or pools, the proposed parallelization options lead to computational times that are 5% higher, on average, than the ones reported for an optimal choice of the parameters. Figure 6 reports a comparison of the computational times required for DFT calculations where the number of threads and pools are the ones for which the lowest computational cost has been found (x-axis), and of the times for the same calculations where the number of threads and pools have been chosen by the algorithm (y-axis).

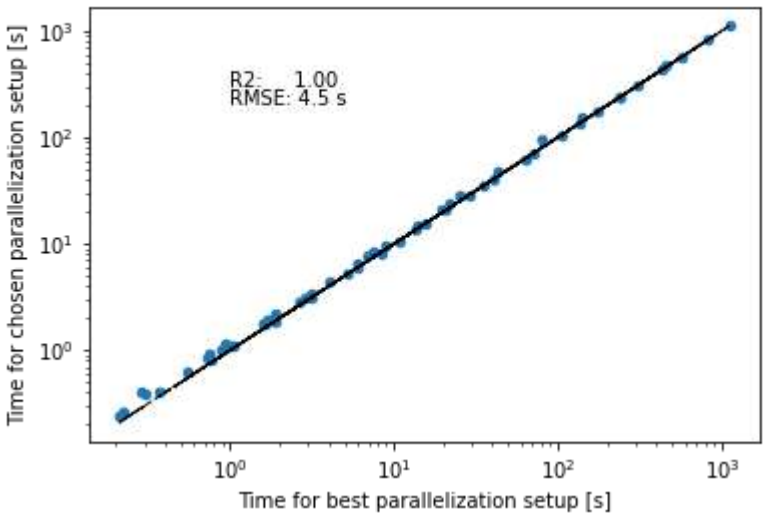

***Figure 6**: Scatter plot reporting the time required to launch a Quantum ESPRESSO calculation using the best possible combination of number of threads and number of pools (x-axis), and using the number of threads and pools proposed by the optimization algorithm (y-axis). The R2 score and root mean square error are reported in the graph.*

The algorithm has been developed in Python 3, and we plan to export it into a script that can support multiple input file formats and can be easily called before submitting a DFT calculation.

Prediction of time per call in Quantum ESPRESSO self-consistent calculations

We developed an algorithm that aims to predict the time-per-call required to run a DFT self-consistent calculation. The predictor employs a shallow artificial neural network to infer the time-per-call as a function of the input parameters to DFT calculations. The algorithm was trained on a data set containing more than 40.000 entries gathered from calculations performed using Quantum ESPRESSO as a case study on the Marconi and Piz Daint supercomputers. The data set spans more than 20000 chemically-diverse systems, and consists of calculations run using different parallelization setups and computational resources.

Figure 7 reports the time-per-call true values (x-axis) and time-per-call predictions (y-axis) on a validation set containing 8000 calculations. The root-mean-squared-error for this model is ~15 s, corresponding to a mean relative error of 30%, and the R2 score is 0.89.
The accuracy of the model is sufficient to optimize the time requested to supercomputers for calculations, therefore potentially reducing the queue time and optimizing the management of computational resources.

The algorithm has been developed in Python 3, and we plan to export it into a script that can support multiple input file formats and can be easily called before submitting a DFT calculation.

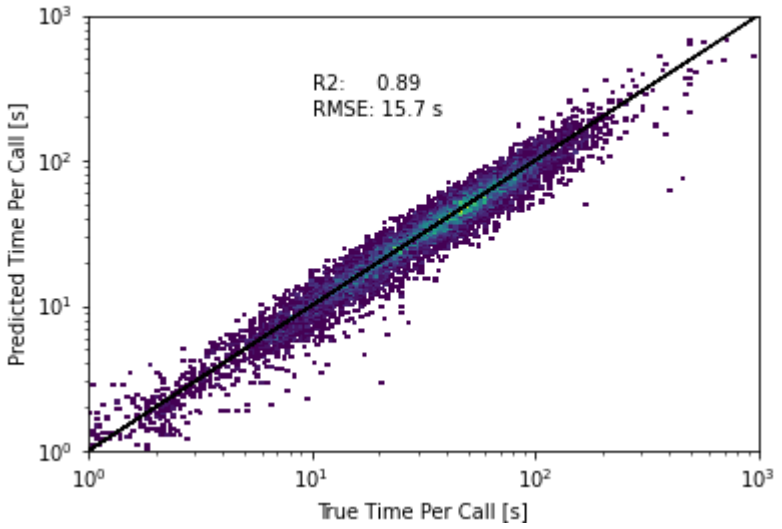

***Figure 7**: 2D histogram for the true (x-axis) vs predicted (y-axis) time-per-call for a validation set containing ~8000 Quantum Espresso calculations. The color is proportional to the density of points, going from blue (low density) to yellow (high density). The black line is a visual aid that indicates a perfect fit.*

3.2 Pilot 2: Configuration explorer/Data explorer toolkit

Chemiscope: interactive visualization of large datasets

Since the last report, we continued development of the chemiscope (available at https://chemiscope.org) library for HPDA using online interactive visualization and exploration. We continued to improve chemiscope performance when displaying large datasets, in particular by using the 3Dmol.js (3Dmol.csb.pitt.edu) library to display molecules and materials in the structure panel instead of JSmol. This cut down the initial loading time of the structure from seconds to milliseconds. This change was done while keeping full backward compatibility with input files and a very similar visual output.

We also added multiple features to chemiscope to make it more useful for data exploration. In particular, chemiscope now supports multiple structure viewers at the same time, to allow the user to compare different structures directly. Additionally, we will now support storing data for a subset of atoms in the input file, again improving loading time of very large datasets where researchers are only interested in some points out of the full data.

Finally, we integrated chemiscope with the materials cloud website, first by adding a discover section which uses chemiscope to display and allow users to explore the data stored in the materials cloud archive entry about Kernel Principal Covariate Regression (visible at https://www.materialscloud.org/discover/kpcovr). Following this, we added a new materials cloud tool accessible to anyone at https://chemiscope.materialscloud.io, and integrated this tool with the materials cloud archive. Now if a chemiscope input file is stored in the materials cloud archive,

visitors to the archive are prompted to use the above-mentioned tool to visualize the corresponding data. This increases interactivity and ease exploration of data stored in the materials cloud archive.

Structure-property correlations with Principal Covariate Regression

One challenge of high-performance data analytics for materials science is the very high dimensionality of the configuration and composition space to explore. One way to tackle this dimensionality problem is to project the different systems of interest in a small dimension map (using an algorithm like PCA, t-SNE, or UMAP on atomic scale representation of the material). Unfortunately, this low dimension projection often only contains data related to the structures of the materials; which doesn't correlate well with the physical properties of interest (energy, magnetism, …). One possible solution to this is to use the Principal Covariate Regression framework (PCovR), which we extended to full and sparse kernel methods as Kernel Principal Covariate Regression (KPCovR). This method finds a low dimensional projection of the dataset which correlates both with structural characteristics of the materials, and the physical properties of interest.

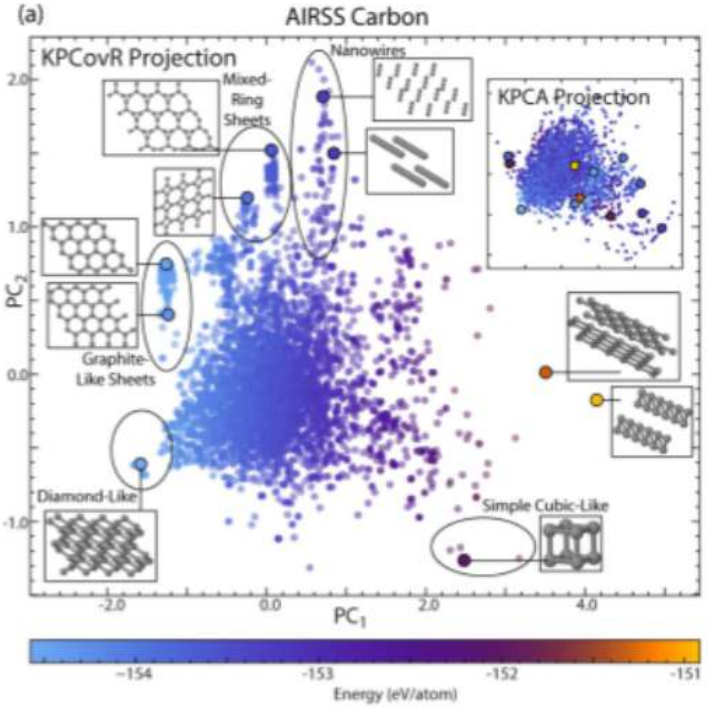

Figure 8: *KPCovR map created from a dataset of carbon structures. Notice how the map contains structural information (shown by molecular structure inserts) and the energy increases linearly from left to right. Adapted from https://doi.org/10.1088/2632-2153/aba9ef)*

The KPCovR dimensionality reduction algorithm can also be used without any human input to produce maps of very large datasets containing the most important structural features related to the properties of interest, enabling better data analysis of these large datasets.

We also extended PCovR to both feature and sample selection in Machine Learning applications (https://doi.org/10.1088/2632-2153/abfe7c). By pre-selecting features or training samples which correlate well with the properties of interest, we are able to improve the computational efficiency of machine learning models, requiring fewer training points and smaller feature vectors for prediction.

An efficient and user-friendly software for data analysis

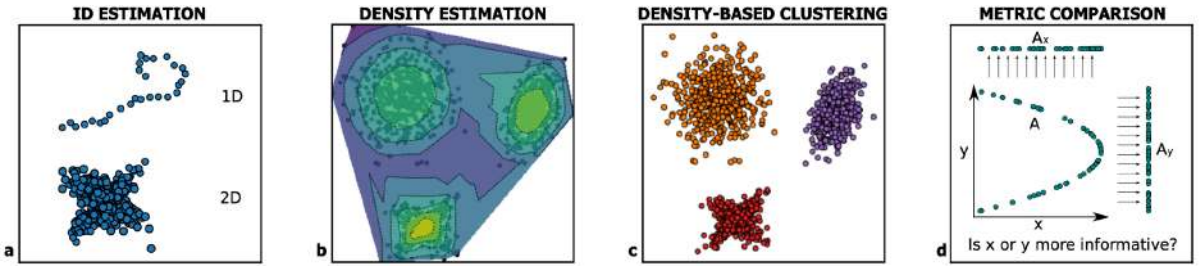

***Figure 9**: An illustration of the four main classes of algorithms implemented in DULy: Intrinsic dimension estimators, density estimators, density-peaks estimators (density-based clustering), and tools of manifold comparison.*

The need to analyze large volumes of data is rapidly becoming ubiquitous in all branches of computational materials science, as HPC/HTC calculations now have the capability of generating an extremely large number of configurations, each including many atoms and molecules in different conformations and electronic structures.

These configurations and structures typically exhibit non-trivial correlations, to be quantified and understood. Deriving a scientifically meaningful picture from this large amount of information is highly non trivial. It requires exploiting state-of-the-art tools for data analysis, which are not yet mature from the points of view of user friendliness and of computational efficiency.

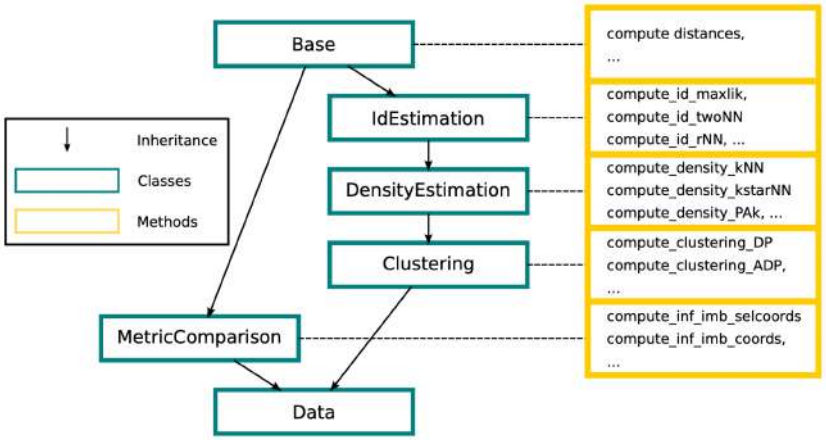

***Figure 10**: The class structure of the package. Classes are marked by green boxes, and the main methods of each class are reported in yellow boxes. Relationships of inheritance are indicated as black arrows. The class Data inherits from all other classes, thus providing easy access to all of the algorithms implemented in the package.*

Our work in this review period focused on the implementation of the algorithms for data analysis developed in our group in a single and user-friendly software provisionally called DULy. This software is designed from scratch in such a way that it can be easily interfaced with AiiDA, becoming one of its

plugins. DULy implements algorithms of *intrinsic dimension estimation* (see Figure 9a), *density estimation* (Figure 9b), *density-based clustering* (Figure 9c) and *metric comparison* (Figure 9d).

DULy is organised in six main classes: *Base*, *IdEstimation*, *DensityEstimation*, *Clustering*, *MetricComparison* and *Data*. The relationships of inheritance between these classes, as well as the main methods (i.e., algorithms) available in each class are summarised in Figure 10. The *Base* class contains basic methods of data cleaning and manipulation which are inherited in all other classes. Then, in a train of inheritance: *IdEstimation* inherits from *Base*; *DensityEstimation* inherits from *IdEstimation* and *Clustering* inherits from *DensityEstimation*.

Each of these classes contains as methods the algorithms of the corresponding type. The inheritance structure of these classes is well motivated by the fact that to perform a density based clustering one first needs to estimate the density, and to perform a density estimation one first needs to estimate the intrinsic dimension, which itself needs the distances to be computed. The *MetricComparison* class contains methods able to compare two manifolds using the distances between points. It also contains the "information imbalance", a novel tool to measure asymmetric information content developed within this project (see previous report).

The class *Data* does not implement any extra method but, importantly, it inherits all methods from the other classes. As such, *Data* contains easy access to all available algorithms of the package and it is the main class to be used in practice.

DULy is written entirely in Python, to help interfacing it with the other program suites developed in the project. Python is however a notoriously inefficient language for large scale computation. In DULy, we circumvent this known shortcoming by implementing all the heavy numerical routines using the "Cython" extension, which essentially generates C-compilable routines of very high efficiency. In this way we are able to maintain the user friendliness of Python without sacrificing the computational efficiency of fully compiled languages.

Just to give an example, processing a data set of 100.000 points takes only minutes on a laptop. Moreover, many routines are linear-scaling and parallelized, and as such they are ideally suited for HPDA applications.

DULy can be used for a wide variety of applications in materials and molecular physics.

For example, the routines of density estimation and density based clustering can be used to divide a large and dishomogeneous atomistic dataset into of few groups of physically similar configurations as well as to represent the hierarchical relationship of "similarity" of these groups and the topography of the data space, in a human-readable and intuitive graphical representation. Moreover, the routines implementing the newly developed information imbalance tool can be used to select, among a large set of materials descriptors, the one that is most informative for a specific dataset.

The code is available through the following GitHub repository (https://github.com/sissa-data-science/DULY), we are now working on the release of a first stable version and on the publication of the package on a dedicated journal.

Using the information imbalance to efficiently compress materials descriptors

In the first 18 months of the project we developed a measure of missing information between different numerical representations of a given material dataset. We called this measure of missing information from a given representation A to another representation B the *information imbalance* from A to B.

We further investigated the theoretical and practical relevance of the information imbalance. In particular we demonstrated the empirical utility of the information imbalance as a tool to select materials descriptors from a pool of candidates (see Figure 11a), and as a tool to sparsify a given materials descriptor by selecting its most informative coordinates (see Figure 11b-d). For example, in Figure 11d, one can observe that the compact descriptors obtained by retaining only the most informative coordinates can be used to efficiently learn materials properties.

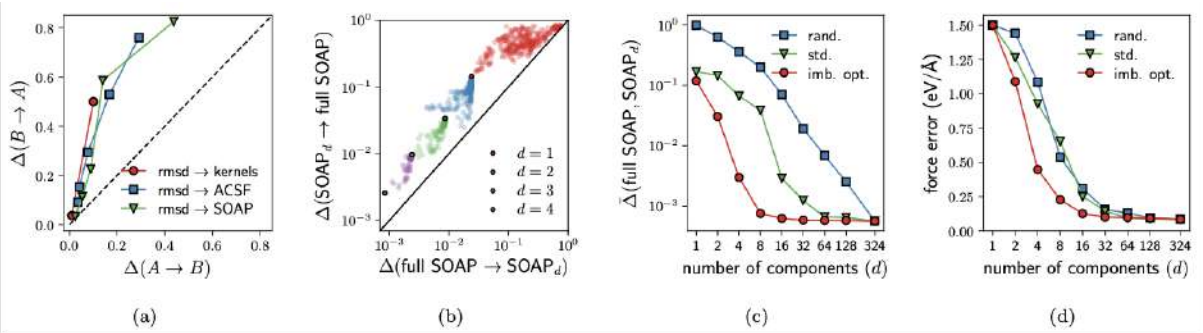

***Figure 11**: Use of the information imbalance for the compression atomistic descriptors. a): Information imbalances between ground truth "rmsd" distance metric and standard atomistic descriptors. All descriptors are seen to converge to the ground truth. b): Information imbalances between a full SOAP descriptor and the most informative d-plet of components (d=1,…,4). c): Convergence of the information imbalance of a SOAP descriptor with the number of components for three different compression strategies. The compression based on the information imbalance is seen to provide the fastest convergence d): Force error on a validation set of a machine learning potential energy model built on the compressed descriptors. The potential built on the information imbalance selected coordinates achieves the lowest test errors.*

3.3 Pilot 3: Dissemination of highly-curated computational material data

Since the previous report, we have been running the automated "turn-key" workflows developed for the aiida-quantumespresso plugin for the dataset of experimental structures obtained from the ICSD, MPDS and COD. In the following sections we describe the full procedure starting from the experimental data sets.

Processing the raw experimental data

The experimental structures are first imported into the AiiDA database as CifData nodes. They are subsequently cleaned and parsed into StructureData nodes using the CifCleanWorkChain (Fig. 12),

which relies on the cod-tools package (https://wiki.crystallography.net/cod-tools/) to clean syntax errors and remove unnecessary information from the cif file. Next, the parsed structures are normalized and primitivized using the SeeK-path library (https://www.materialscloud.org/work/tools/seekpath).

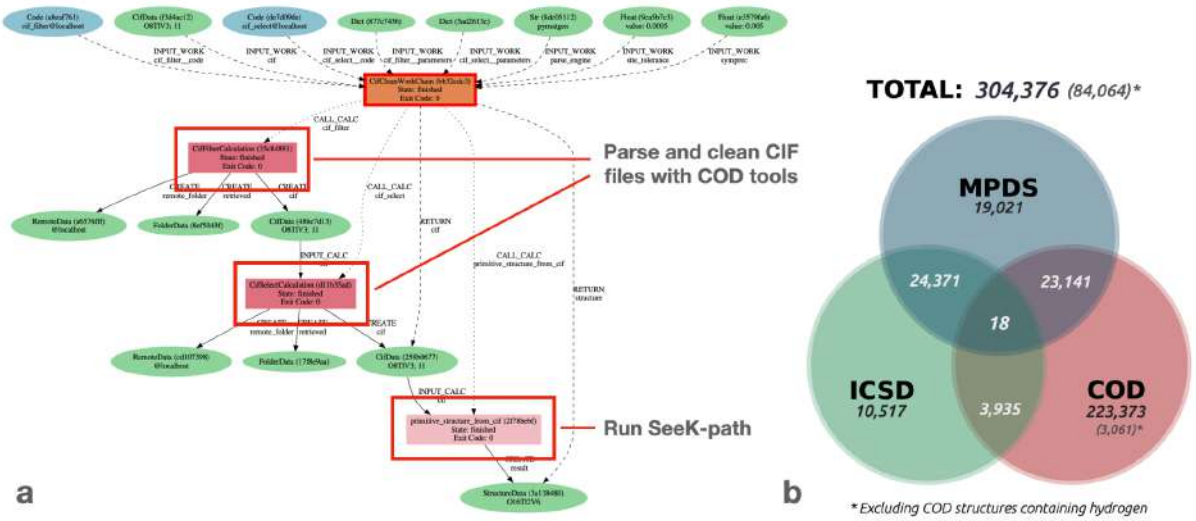

Figure 12: *(a) Provenance graph of the CifCleanWorkChain used to clean and parse the experimental CIF files into StructureData, as well as normalize and primitivitize the structure with SeeK-path. (b) Venn-diagram of the overlap of the experimental databases based on the uniqueness analysis.*

Since calculating structures with partial occupancies is beyond the scope of this pilot, we remove these from the list of parsed StructureData. Finally, we filter out duplicates from the structure set by comparing the primitivized structures using the StructureMatcher utility of pymatgen (https://pymatgen.org/). The resulting venn-diagram of structures is shown in Fig. 12 (b). The large number of unique structures in the COD is mainly due to the many molecular crystals present in this database. After removing the COD structures containing hydrogen, we obtain a list of 84,064 structures for which we aim to optimize the geometry and band structure.

Calculating the ground-state geometry

The structures obtained from the uniqueness analysis are optimized in several steps, shown schematically in Fig. 13 (a). First a reconnaissance SCF (rSCF) is performed to determine if the structure should be treated magnetically or not. Here, we simply assign a high-spin ferromagnetic configuration to all elements with partially occupied *d* or *f* orbitals, as well as a small magnetic moment to other elements (10% spin surplus in the up channel). A suitable k-point mesh is constructed based on the density specified by the “moderate” protocol (see section 5.2 of report D5.4). Next, a static PwCalculation is performed, wrapped inside a BaseRestartWorkChain (see section

2.2 of report D5.3) to allow recovery from common errors that can occur for the pw.x code of Quantum ESPRESSO. The final magnetic moments are used to construct a new StructureData node with the correct kinds, which is passed onto the next step in the procedure. If all of the magnetic moments are below a certain threshold (0.05 μ_B), the structure is subsequently treated as non-magnetic.

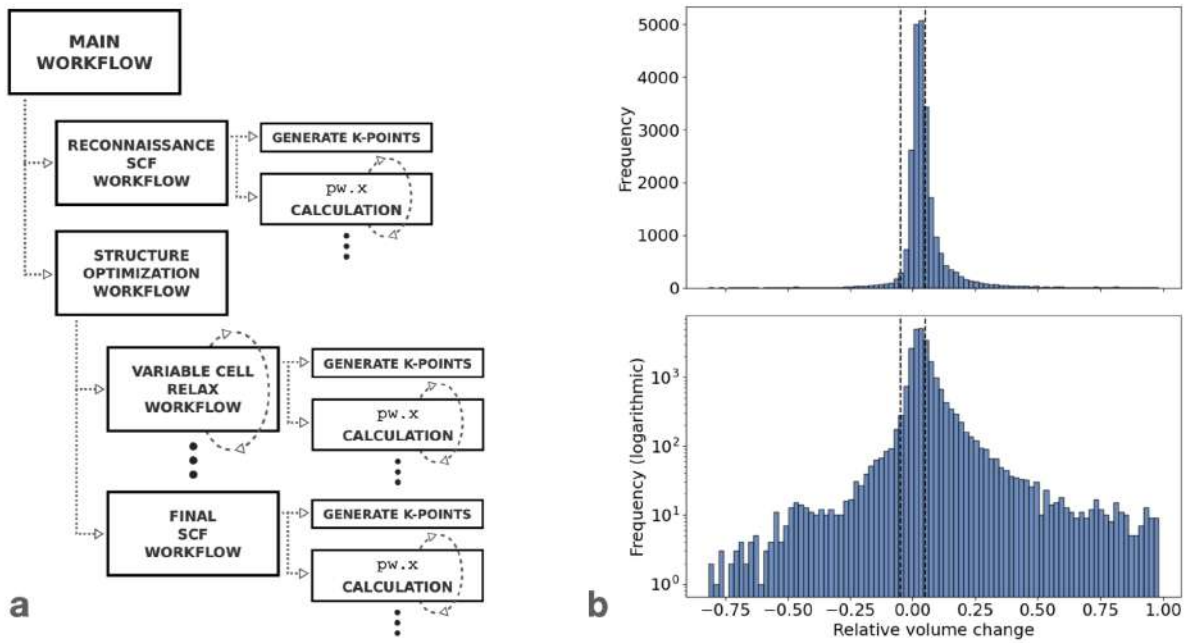

Figure 13: *(a) Diagram of the steps performed to optimize the geometry of the experimental structures. (b) Distribution of the relative volume change of the unit cell after optimization of the geometry.*

The next step in the procedure is to optimize the geometry of the structure using the PwRelaxWorkChain discussed in section 3.1 of the D5.3 report. This workflow will run several variable-cell relax calculations, once again wrapped inside a BaseRestartWorkChain to allow for automated error recovery. The structure is considered fully converged when the relative volume change between two such calculations is less than 1%. A final SCF is performed on the relaxed structure to determine the ground state charge density, which is subsequently stored for future calculations using the stashing functionality described in section 3.4 of the D5.4 report. Figure 13 (b) shows a histogram with the relative volume change of the optimized structures versus the experimental ones. It is clear that in most cases, the calculated volume is slightly larger compared to experiment, which is in accordance with the known effect that the PBE functional tends to overestimate the cell volume.

Based on this workflow, over 35,000 structures have been fully optimized and made available as the Materials Cloud three-dimensional (MC3D) discovery section (Fig. 14). Each structure has been assigned a unique ID based on the “parent” structure obtained from the analysis discussed in the

previous section, as well as an appended modifier that indicates the functional used in the calculations.

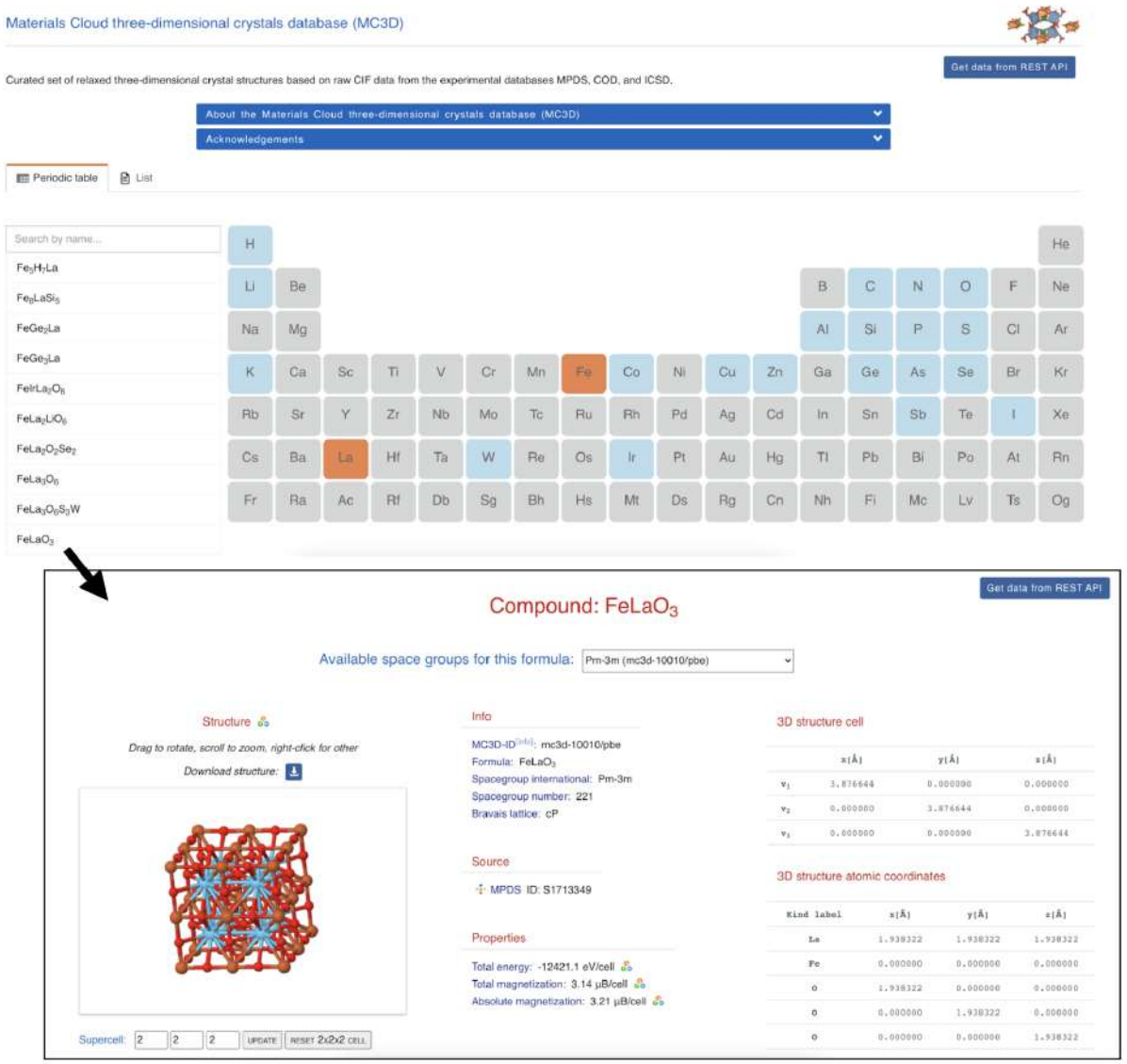

***Figure 14**: Materials cloud discover section for the MC3D. Users can search for the compound they are interested in by indicating the desired elements on the periodic table and subsequently selecting the compound from the list on the left. Currently the structural and magnetic details are provided, along with a visualization of the geometry, however in future releases we aim to include XRD simulations, band structures and (projected) density of states.*

Running AiiDA at scale: the LUMI pilot

As an extension of the MC3D project, we recently participated in the pilot phase of the CPU partition of the new LUMI supercomputer hosted in Finland (https://www.lumi-supercomputer.eu/). To demonstrate the capability of AiiDA at running calculations efficiently in high-throughput, we obtained a 12 hour time slot of exclusive access to all 1500 nodes of the partition. For this benchmark we decided to stick with the thoroughly tested PwRelaxWorkChain, rerunning the smallest structures in our database up to the highest possible system size with PBEsol.

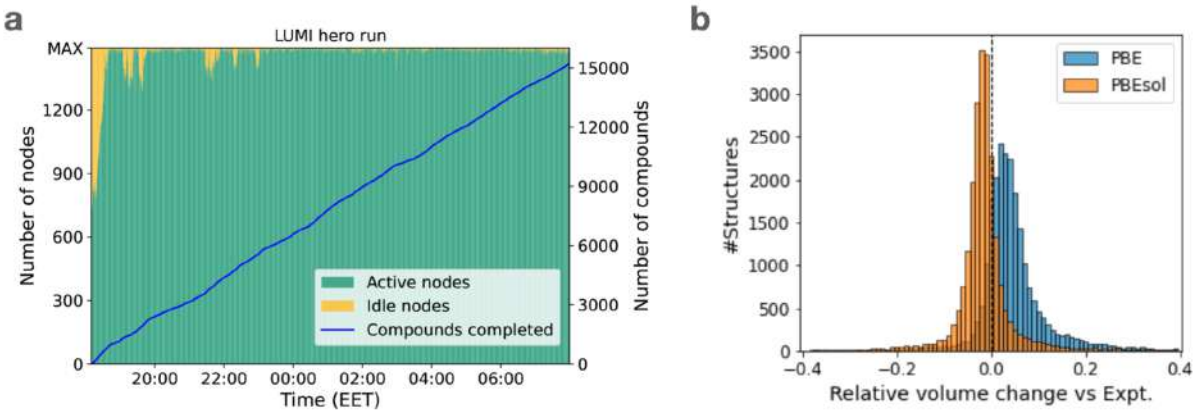

Figure 15: *(a) Results of the AiiDA "hero run" on the LUMI-C partition. The distribution of idle vs active nodes is shown throughout the time slot, along with the progression of the number of calculated compounds. (b) Comparison of the relative volume change of the unit cell of the PBE and PBEsol-optimized geometries versus that of the experimental structures.*

Figure 15 shows the results of this "hero run", during which 55,704 Quantum ESPRESSO calculations were orchestrated by AiiDA, fully optimizing the geometry of 15,324 geometries (all structures with up to 17 atoms in the unit cell in our database). Based on these results, as well as continued runs during the LUMI pilot and our previous results obtained with PBE, we are able to perform a thorough comparison of the accuracy of PBE and PBEsol when it comes to determining the crystal structure (Fig. 15). We can see that PBEsol does an overall better job at obtaining the correct unit cell volume, especially considering that we are performing these simulations at zero temperature and hence a slight underestimation of the crystal volume is to be expected.

3.4 Pilot 4: Edge computing

Connection with experimental beamlines of crystallography (Collaboration with ILL and Diamond Synchrotron).

Quantum mechanical investigation of the microscopic factors modulating the strength of biomolecule interactions

The wavelet-based electronic structure code, BigDFT, enables density-functional theory calculations which scale linearly with the number of atoms, in contrast to the traditional approaches which exhibit a cubic scaling behaviour. Thanks to its unique features, full quantum mechanical description of (solvated) proteins are accessible up to systems of tens of thousands of atoms. Combined with a molecular dynamics techniques, BigDFT has been successfully applied to macro-molecular systems in biology, notably in connection with the COVID pandemic [https://www.ox.ac.uk/news/features/unique-international-zoom-collaboration-develop-treatments-covid-19, published in Chemical Science, https://doi.org/10.1039/D1SC03628A], by making it possible to identify the active fragments of a protein in the presence of a target molecule and to provide the key parameters for their modelling. This is an example which gathers together results coming from X-ray and neutron crystallography and postprocess the data by employing Full DFT calculations on thousands of structures performed on supercomputers. We here describe the scientific context of the project and the main achievements.

Computational approaches that model molecule-molecule interactions have the potential to yield the necessary insights about interactions in biological systems. Such methods have been employed before for small-molecule (about a hundred atoms) drug discovery (Schneider, 2010; Gorgulla et al., 2020). Molecular docking uses geometrical constraints to assess how a small ligand interacts with a larger substrate (Brooijmans and Kuntz, 2003; Guedes, de Magalhães and Dardenne, 2014; Pagadala, Syed and Tuszynski, 2017). Geometry is the main factor in molecular docking, making the method relatively fast and thus of interest for surveying small-molecule candidates in drug discovery. Moving towards more mechanistic descriptions, hybrid quantum mechanics/molecular mechanics methods are commonly used for enzyme-substrate systems (Senn and Thiel, 2009). Such method uses highly accurate QM simulation for a small fraction of the system (identified a priori to be important for interaction), whereas the rest of the molecules is modeled with a less computationally demanding MM simulation. Determining the appropriate QM region of a hybrid model is not trivial, especially in situations where the sites of interactions on each molecule are unknown (Kulik et al., 2016). Another alternative is the use of force-fields (FFs) to calculate the forces between and within molecules. However, FFs are not based on first principles and require refined parameterization. A computational method that is able to perform a full quantum mechanical (QM) simulation by employing unparameterized ab-initio level of theory can effectively complement the drawbacks from previous approaches (Ratcliff et al., 2015, 2020; Mohr et al., 2017). Only recently, advances in computing enabled us to simulate molecular interactions large enough (tens of thousands of atoms, representing many hundreds of amino acids) to capture biological processes.

We have employed one of those recent large-scale ab-initio computational approaches, provided by the BigDFT computer program, to simulate large structures with a computational cost manageable on modern supercomputers. Starting from a population of fully atomistic 3D structural models, this approach opens up new possibilities for investigating large, biologically relevant molecules, and understanding the role of each of the system's constituents in such intra-molecular interactions. Our

approach has been used in conjunction with the techniques described above, in order to investigate the interaction patterns of one of the main enzymes of SARS-CoV-2 virus (the main protease) with natural peptidic substrates and designes peptide inhibitors which have been tested in vitro. This work has recently been published in Chemical Science [https://doi.org/10.1039/D1SC03628A]. The procedure employed is highlighted in the table below.

Electron Density	The distribution of electrons in a given molecular system, such as the SARS-CoV-2 Mpro-inhibitor complex. The electron density matrix determines the nature and strength of the chemical bonds of the system. Such an « electron cloud » is the main emerging property of the underlying atomic structure in defining the chemical characteristic of a molecule.
Fragments	The modular elements to which the electron cloud can be partitioned. The model partitions the electron cloud into physically-consistent regions; every such region, which can constitute one or more amino acids, is defined as a fragment.
Fragment Bond Order (FBO)	The descriptor of the inter-fragment interactions. FBO is the main quantity used in the model to represent the connection pattern of the fragments of interacting molecules.
The Final Output	At the end of the simulation, BigDFT provides an unbiased representation of the drug-enzyme interactions. It provides a simple representation of the strength of interaction between fragments of the two molecules. The model can ultimately describe the energy and nature of the acting chemical bonds. This enables a mechanistic prediction of how specific amino acid substitutions or deletions (in a spike or an antibody) impact the interactions with their hACE2 substrates or the viral spike, respectively.
Hardware Requirement	The model requires massively parallel calculations via high performance computing (HPC). Wwe need about 500k hours on average to quantumly post process 100 snapshots corresponding to a 600 residue assemblyHundreds of simulations of inhibitor- enzyme systems can be performed in a time frame of one hour.

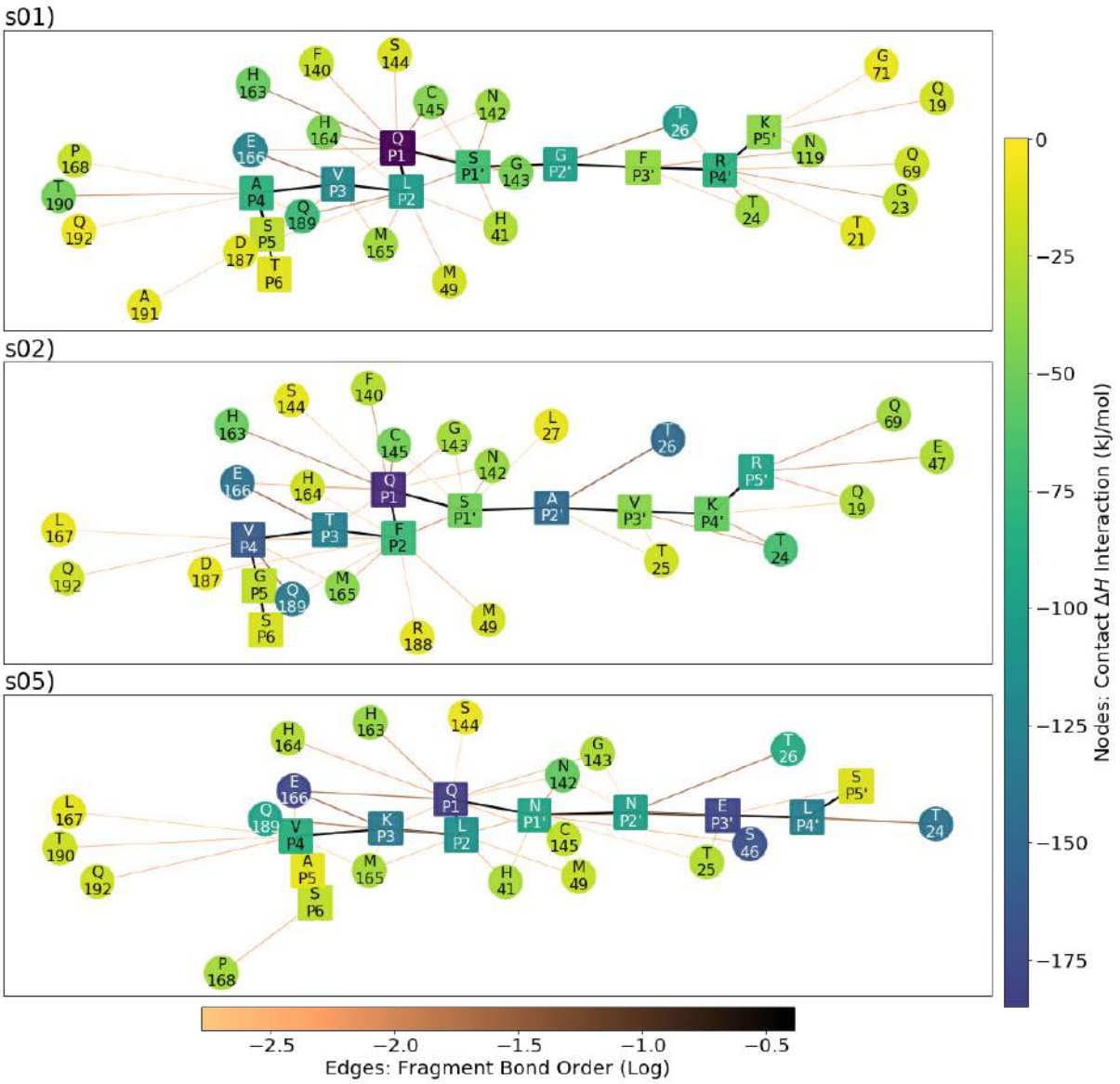

Figure 16: *QM interaction networks where node colour indicates interaction strength, from dark blue (strongest) through green to yellow (weakest). Square nodes denote substrate, while circular ones denote Mpro. The thickness and colour of the edges show the fragment bond order between residues, a unitless measure associated with bond strength and analogous to bond order; black is strongest, orange is weakest. Interaction energies and bond orders were computed using BigDFT and ensemble-averaged results of MD snapshots.*