npj | Computational Materials

www.nature.com/npjcompumats

PERSPECTIVE OPEN

Towards efficient data exchange and sharing for big-data driven materials science: metadata and data formats

Luca M. Ghiringhelli[1], Christian Carbogno[1], Sergey Levchenko[1], Fawzi Mohamed[1], Georg Huhs[2,3], Martin Lüders[4], Micael Oliveira[5,6] and Matthias Scheffler[1,7]

With big-data driven materials research, the new paradigm of materials science, sharing and wide accessibility of data are becoming crucial aspects. Obviously, a prerequisite for data exchange and big-data analytics is standardization, which means using consistent and unique conventions for, e.g., units, zero base lines, and file formats. There are two main strategies to achieve this goal. One accepts the heterogeneous nature of the community, which comprises scientists from physics, chemistry, bio-physics, and materials science, by complying with the diverse ecosystem of computer codes and thus develops "converters" for the input and output files of all important codes. These converters then translate the data of each code into a standardized, code-independent format. The other strategy is to provide standardized open libraries that code developers can adopt for shaping their inputs, outputs, and restart files, directly into the same code-independent format. In this perspective paper, we present both strategies and argue that they can and should be regarded as complementary, if not even synergetic. The represented appropriate format and conventions were agreed upon by two teams, the Electronic Structure Library (ESL) of the European Center for Atomic and Molecular Computations (CECAM) and the NOvel MAterials Discovery (NOMAD) Laboratory, a European Centre of Excellence (CoE). A key element of this work is the definition of hierarchical metadata describing state-of-the-art electronic-structure calculations.

npj Computational Materials (2017)3:46 ; doi:10.1038/s41524-017-0048-5

INTRODUCTION

To aid and guide the search for new and improved materials, computational materials science is increasingly employing "high-throughput screening" calculations.[1–7] In practice, this means that computational material scientists produce a huge amount of data on their local workstations, computer clusters, and supercomputers using a variety of very different computer programs, in this domain usually called "codes". Though being extremely valuable, this information is hardly available to the community, since most of the data is stored locally. In publications, typically, only a small subset of the results is reported, namely that which is directly relevant for the specific topic addressed in the actual manuscript. Although several repositories have been created and maintained in the past for domain-specific applications, these typically do not store the full inputs and outputs of all calculations. For instance the Materials Project (www.materialsproject.org), the Open Quantum Materials Database (OQMD (http://oqmd.org/)), the Theoretical Crystallography Open Database (T-COD (www.crystallography.net/tcod)), the Electronic Structure Project (ESP (http://gurka.fysik.uu.se/ESP/)), and the Open Materials Database (http://openmaterialsdb.se/). And, when inputs and outputs are available (e.g., AFLOW (http://aflowlib.org)), they are restricted to results obtained by just one or very few codes.

Since 2014, this situation has been changing: the NOvel MAterials Discovery (NOMAD) Repository is designed to meet the increasing demand for storing scientific data and making them available to the community, http://nomad-coe.eu. The NOMAD Repository, https://repository.nomad-coe.eu, is part of the NOMAD Laboratory European Centre of Excellence (CoE). It is the only repository for materials science accepted by Nature Scientific Data. It is a unique facility, as it accepts (and in fact requests) the full input and output files of all important computer codes used in computational materials science, it guarantees to store these data for at least 10 years after the last upload, and a DOI is provided so that data is citable.

Currently, the NOMAD Repository holds the information of over 40 million total-energy calculations (corresponding to converged single point, atomic structure, calculations), which corresponds to several billion CPU hours. NOMAD offers their users to restrict (for up to 3 years) their upload to themselves and other selected users, or to make them "open access" right away. The time evolution of the open-access content of the Repository is shown in Fig. 1. The exact amount of restricted-access total-energy calculation is of course unknown (as restricted-access data are not parsed), but the vast majority of uploaded data is in fact open access, and we estimate the amount of restricted data as 2% (i.e., less than one million additional total-energy calculations). It has to be noted that the NOMAD Repository contains also *all* the original inputs and outputs of the data in the AFLOW, OQMD, and Materials Project databases. While this has been and is useful for its purpose of data sharing via a repository (enabling the confirmatory analysis of materials data, their reuse, and repurposing), the data is very heterogeneous, as they are provided by the different codes. Thus,

[1]Fritz Haber Institute of the Max Planck Society, Berlin, Germany; [2]Barcelona Supercomputing Center, Barcelona, Spain; [3]Humboldt-Universität zu Berlin, Berlin, Germany; [4]Daresbury Laboratory, Warrington, UK; [5]University of Liège, Liège, Belgium; [6]Max Planck Institute for the Structure and Dynamics of Matter, Hamburg, Germany and [7]Department of Chemistry and Biochemistry and Materials Department, University of California—Santa Barbara, Santa Barbara, CA 93106-5050, USA
Correspondence: Luca M. Ghiringhelli (ghiringhelli@fhi-berlin.mpg.de) or Micael Oliveira (mjt.oliveira@ulg.ac.be)

Received: 23 March 2017 Revised: 17 September 2017 Accepted: 19 September 2017
Published online: 06 November 2017

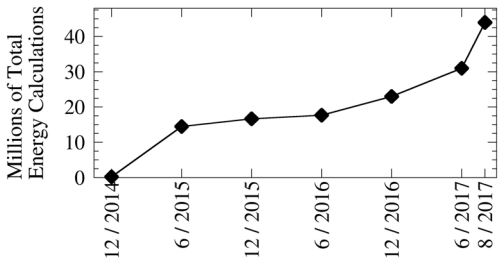

Fig. 1 Evolution of the content of the NOMAD Repository and Archive, in terms of open-access total-energy calculations

they are not useful for data analytics and extensive comparisons, because the input and output files of different codes typically use different units, representations, and file formats.

Advancing the sharing and comparison of such data is a pressing issue that needs to be addressed. To the purpose, the NOMAD Laboratory CoE has built a unified, code-independent database called the NOMAD Archive (https://metainfo.nomad-coe.eu/nomadmetainfo_public/archive.html), so that big-data analytics techniques can be applied to obtain unprecedented insight from the vast amount of already existing calculations. The NOMAD Archive also serves the NOMAD Encyclopedia (https://encyclopedia-gui.nomad-coe.eu/) and Advanced Graphics (https://www.nomad-coe.eu/index.php?page=graphics).

In a similar spirit, the E-CAM CoE (www.e-cam2020.eu), was recently established by CECAM, the European Center for Atomic and Molecular Computations. CECAM is devoted to the promotion of fundamental research on advanced computational methods and to their application to important problems in frontier areas of science and technology. E-CAM was established to build an e-infrastructure for software, training, and consultancy in simulation and modeling, is committed to actively support the development and adoption of software libraries and standards within the electronic structure community. One initiative aiming at this is CECAM's Electronic Structure Library (ESL) initiative http://esl.cecam.org, which includes in its plans the development of an Electronic Structure Common Data Format (ESCDF). The ESCDF provides a standardized data format and an application programming interface (API) that every code can use. Code interoperability is also strongly driven by communities, using the results for further analysis, such as spectroscopy calculations, based on converged and atomically relaxed ground state results.

As mentioned above, many codes are used by the computational materials science community, also referred to as the "Ψ_k community", after the name of the wide network (http://psi-k.net/) that links most if not all materials scientists who are dealing with or starting from the electronic structure. A list of the most important electronic-structure codes in the materials science community is given in Table 1. The codes are ranked by the number of citations over the last 5 years and their main features are summarized.

In this perspective paper, we will discuss the challenges of electronic-structure codes, but note that the NOMAD Laboratory CoE also includes force-field-based codes. The various electronic-structure codes differ by many conceptual and numerical aspects, first of all by the type of basis sets that are employed. This implies significantly different concepts and numerical methods for solving the many-body Schrödinger equation or the Kohn–Sham equations. Besides, some codes describe all electrons, others use a simplified description of the core electrons, e.g., the frozen core or even a pseudopotential approximation. On one side, this results in very different data representations; on the other side, it is not trivial to compare certain quantities like energies and wavefunctions.

Table 1. List of the electronic-structure codes with more than 100 citations in the 2012–2016 period

Code	Search name	Citations (2012–16)	Description	License
Gaussian	Frisch*	20,400	AE, PSP, GTO	C
VASP	Kresse	18,900	PSP, PW, PAW	C(O)[a]
GAMESS	Gordon	6050	AE, PSP, GTO	F
Quantum ESPRESSO	Giannozzi*	6020	PSP, PW	G
CASTEP	Payne	5380	PSP, PW, PAW	C(O)[b]
Molpro	Werner	4480	AE, PSP, GTO	C
WIEN2k	Blaha	4420	AE, (L)APW	C
SIESTA	Soler	4340	PSP, NAO	G
TURBOMOLE	Ahlrichs	3900	AE, PSP, GTO	C
DMol[3]	Delley*	3195	AE, PSP, NAO	C
ADF	Baerends	3160	AE, STO	C
ORCA	Neese	3120	AE, PSP, GTO	F
CRYSTAL	Dovesi*	2453	AE, PSP, GTO	C[b,c]
MOPAC	Stewart	2080	PSP, STO	F(O)[d]
Q-Chem	Shao*	2030	AE, PSP, GTO	C
ABINIT	Gonze	1970	PSP, PW, PAW, WLT	G
Dalton	Ågren	1820	PSP, GTO	F
Jaguar	*Schrödinger*	1690	AE, PSP, GTO, STO	C
NWChem	Valiev	1650	AE, PSP, GTO, PW, PAW	G
MOLCAS	Lindh	1300	AE, PSP, GTO	C[e]
CP2K	Hutter*	1510	AE, PSP, GTO, PW	G
ACES III	Bartlett*	1240	PSP, GTO	G
CPMD	Hutter	901	PW, PSP	F
octopus	Rubio	869	PSP, RS	G
TB-LMTO-ASA	Andersen*	843	AE, LMTO, *N*MTO	G
CFOUR	Stanton*	762	AE, PSP, GTO	F
GPAW	Mortensen	726	PSP, PW, PAW, RS, NAO	G
CASINO	Needs*	588	AE, PSP, PW, GTO, STO, NAO	F[d]
DIRAC	Saue*	580	AE, LAO	F(O)
FPLO	Koepernik	488	AE, NAO	C
FHI-aims	Blum	484	AE, NAO	C(O)
OpenMX	Ozaki	440	PSP, NAO	G
COLUMBUS	Lischka	383	AE, GTO	F
Smeagol	Lambert	325	PSP, NAO	F
Psi4	Sherrill	289	AE, GTO	F(O)
ONETEP	Skylaris	283	PSP, PW/RS, PAW, NAO	C(O)
Yambo	Marini	264	PSP, PW	G
exciting	Draxl*	230	AE, LAPW	G
TeraChem	Martinez*	227	AE, PSP, GTO	C
FLEUR	Blügel	198	AE, LAPW	F(O)
BerkeleyGW	Louie	192	AE, PSP, PW, NAO, RS	F(O)
PARSEC	Saad	171	PSP, RS	G
BigDFT	Genovese	148	PSP, WLT	G

Table 1 continued

Code	Search name	Citations (2012–16)	Description	License
ATOMPAW	Holzwarth	119	AE, PSP, PW, PAW	F(O)
CONQUEST O (N)	Bowler*	116	AE, PSP, NAO, PW	I
Elk	Dewhurst*	110	AE, LAPW	G
Qbox	Gygi	104	PW, PSP	G(O)

The number of citations is determined via Google Scholar, with a search performed in June 2017, by searching for the name of the code (as reported in the first column) and the name of one of the main developers (or *company* that develops and commercializes the product, as reported in the second column). The reason of the second search criterion is that the name of several codes have different meanings. We found that the combination of the two column gave very few, if any, false positive results. Some of the codes (marked by an asterisk in the "Search Name" field) require special search string, i.e., not just the code name in the first column and the search name in the second (this search criterion applies to all codes not in the following list). For these codes, we report within curl brackets the exact search strings. {Gaussian package Frisch}, {"Quantum ESPRESSO" Giannozzi}, {DMol3 Delley} + {"DMol3" Delley}—{"DMol3" DMol3 Delley} (i.e., the sum of the citations resulting from the first two strings minus the result of searching the third string), {"CRYSTAL14" Dovesi} + {"CRYSTAL09" Dovesi} + {"CRYSTAL06" Dovesi} + {"CRYSTAL code" Dovesi}, {"Q-Chem" Shao}, {"CP2K" Hutter}, {"ACES" Bartlett}, {"TB-LMTO" Andersen}, {"CFOUR" Stanton}, {DIRAC "Saue"}, {"CASINO" "Needs" Rios}, {elk.sourceforge.net Dewhurst}, {"CONQUEST" "Miyazaki" "Bowler"}, {"TeraChem" Martinez}, {"exciting" code Draxl}. When a word is in quotation marks, the result of the search without the marks gives unwanted results, so the quotation marks are necessary.
The significance of the citation numbers should not be overrated (as in any citation analysis). For example, young codes that were only developed in the last 7–8 years may have a high gradient of their employment in the community but still have a rather low citation index in the table. Nevertheless, we believe that the general impression provided by the table is correct. The fourth column lists the main features of each code and the fifth the type of license. A list including also force-field-based codes is maintained at: www.nomad-coe.eu/index.php?page=codes.
Abbreviations: *AE* all electron, *PW* plane wave, *GTO* Gaussian-type atomic orbitals, *NAO* numeric atomic orbitals, *STO* slater-type orbitals, *LAPW* linearized augmented plane wave, *PAW* projector augmented wave, *RS* real space, *PSP* pseudopotential (including also ECP, effective core potential), *LAO* London atomic orbitals, *LMTO/NMTO* (linear) muffin-tin orbitals/Nth-order MTO, *WLT* aavelet; for "License", *G* GPL or LGPL (includes also GPL compatible licenses, such as Educational Community License or Apache 2), *F* free (other licenses), *C* commercial/charged (usually, with a smaller cost for academics compared to non-academics), *O* open source, *I* individual-basis license (via contacting the authors)
[a] Free for academic use in Austria
[b] Free forcademic use in the UK
[c] Free demo serial version (max. 20 atoms/cell)
[d] Commercial for non-academic use
[e] Version 8.2 will have license G

In the remainder of this paper, we will define a common code-independent representation for all relevant quantities (e.g., structure, energy, electronic wave functions, trajectories of the atoms, etc.). The ideas and concepts presented here are also the result of discussions that took place at a CECAM/Psi-k workshop, held in Lausanne in January 2016, that was attended by experts and key developers of more than 20 of the most important electronic structure codes.

THE CONVERSION LAYER

In this section, we discuss the key issues for converting the information present in the inputs and outputs of electronic-structure codes into a common format. More specifically, this discussion addresses: (i) the metadata infrastructure for storage and retrieval of the code-independent quantities; (ii) the uniform file format as defined by the ESCDF team; (iii) the zero-level reference for energy-related quantities; (iv) the representation for the electronic and vibrational band structures and density of states (DOS); (v) the compact representation of scalar fields, such as wave functions, electron densities, and exchange-correlation potentials; (vi) the unified representation of quantities related to excited-state calculations (*GW*, Bethe–Salpeter). Furthermore, we discuss the general challenge, touching all the above points, of establishing error bars and confidence levels, with respect to the adopted numerical settings for each stored/retrieved calculation.

Metadata for the code-independent format

Metadata is the name or label that characterizes corresponding values. For example, `XC_method` is a metadata name and "LDA" may be the corresponding value. Thus, if one thinks of storing data as *key-value* pairs (as in a dictionary), the *key* is the metadata.

There are several examples of metadata approaches in computational materials science, the most prominent being ChemML, the Chemical Markup Language (www.xml-cml.org), CIF, the Crystallography Information File (www.iucr.org/resources/cif), code specific implementations such as the VASP (http://cms.mpi.univie.ac.at/vasp/vasp/vasp.html) or Molpro (www.molpro.net/info/2012.1/doc/manual/manual.html) XML outputs, or very simple and intuitive file formats like XYZ (http://openbabel.org/wiki/XYZ_(format)) for the storage of the configuration of a system (atomic species, coordinates, also allowing for comments).

In general, a metadata structure is built a priori, by starting from a list of names that identify the needed concepts and quantities. Code outputs and file formats are then designed to reflect the a priori metadata structure. In principle, a priori metadata approaches aim to be as exhaustive as possible; in practice, they are typically designed with a specific scientific field, application and/or code in mind, as the examples above show. Conversely, the "NOMAD Meta Info" (https://metainfo.nomad-coe.eu/nomadmetainfo_public/info.html), i.e., the metadata defined and used for the NOMAD code-independent data format, is defined a posteriori, by starting from the existing inputs and outputs of many different codes stemming from different scientific fields and thus with quite diverse feature sets. From a computer science point of view, the a posteriori approach is much more challenging than the a priori one, since it requires a more flexible, extensive, and generalized infrastructure (https://metainfo.nomad-coe.eu/nomadmetainfo_public/static/metainfo.html): On the one hand, the diverse quantities stored by each code need to find the proper place in the global hierarchy. On the other hand, not all codes' inputs and outputs contain all the defined metadata; therefore, having unassigned metadata when sorting a given input/output in the code-independent format must not create problems. From a physics, chemistry, and materials science point of view, the a posteriori approach has the critical advantage that essentially all the information contained in any input/output file of any considered code is recognized and stored into the properly identified *key-value* pair. Intrinsically, this guarantees a truly exhaustive coverage of all properties, even those that currently might not appear of interest. In turn, this lays the founding for authentic big data mining approaches that unveil hitherto unexpected correlation and relationships.

In NOMAD Meta Info (https://metainfo.nomad-coe.eu/nomadmetainfo_public/info.html), the *key* is not a simple string but a more structured object, with several attributes. The *name* is an attribute, a string that must be unique, well defined, intuitive and as short as possible. It is used to identify the metadata and therefore used to associate *values* with their metadata. In the following description, all metadata names are given in typeface font. The *description* is another attribute and contains an extensive human-readable text that clarifies the meaning of the metadata.

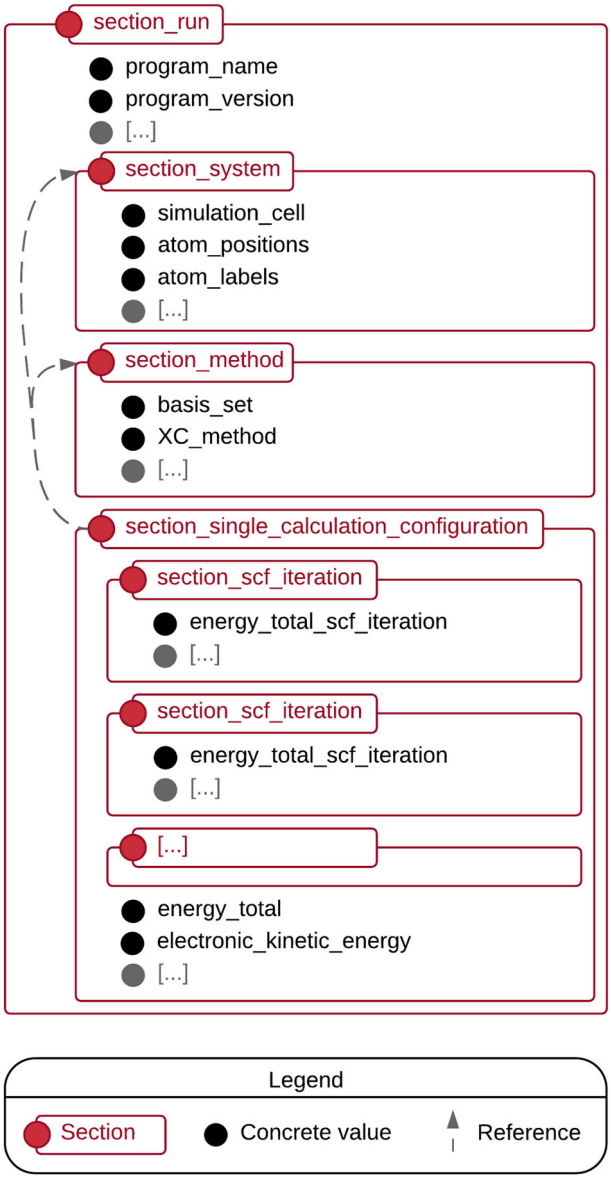

Fig. 2 A simplified metadata structure, according to NOMAD Meta Info (https://metainfo.nomad-coe.eu/nomadmetainfo_public/info.html), for a simple electronic-structure calculation is shown. The metadata indicated in black are of *concrete* type. The *names* are self explaining (the full *description* is given in (https://metainfo.nomad-coe.eu/nomadmetainfo_public/static/metainfo.html)). The metadata indicated in red are of type *section* and the content of the *section* defined by them is inside the red box. *Sections* can also have references, indicated by the dashed arrows, to other *sections* besides the parent *section*. The layout of the metadata in the figure is reminiscent of the JSON file, that is one of the file formats in which they are effectively stored

Another important attribute of a metadata is its *type*. Concrete *values*, scalars, strings, or (multidimensional) arrays, that are extracted by the parsers that read input and output files, have an associated *concrete-type* metadata. These *values* are organized in groups to which *section-type* metadata are associated. In computational informatics, these *sections* would correspond to tables of a relational database model, where the *values* would be the rows, and the *concrete* metadata the columns. In a relational database, data are organized in tables, where rows represent instances of some entity (e.g., a customer, a product) and the columns values attributed to that instance (e.g., the address, the price). Rows are identified by unique keys.

By considering the different steps that define an electronic-structure calculation run (i.e., from the invocation of the code to the completion of the task described in the input files, or the interruption due to several reasons), the following main *section-type* metadata are defined:

- `section_run`: represents a single run of the program. It contains, besides the sections listed in the following points, the metadata that are common to a whole calculation, such as `program_name` and `program_version`. The metadata *names* given in this summary should be self explaining. In any case, a full description can be found by searching the metadata *name* at https://metainfo.nomad-coe.eu/nomadmetainfo_public/index.html.
- `section_method`: contains the information defining the theory level and convergence parameters. For instance, `XC_functional`, where the allowed values are taken from the libxc library (https://gitlab.com/libxc/libxc/wikis/Functionals-list-unreleased).
- `section_system`: contains the specifics of the system configuration, such as the `simulation_cell` (a 3×3 matrix), the `atom_positions` (a $N_A \times 3$ matrix, where N_A is the number of atoms in the unit cell), and the `atom_labels` (an array of length N_A with the atomic species, corresponding to the `atom_positions`).
- `section_single_configuration_calculation`: contains the results for a physical system as defined in a single `section_method` and a single `section_system`. For instance, the converged `energy_total`, `atom_forces`, `stress_tensor`, (Kohn–Sham) `eigenvalues`, etc.
- `section_scf_iteration`: contains the results of a single self-consistency iteration.

The chosen hierarchy reflects that *sections* can be nested, meaning that each outer *section* can contain one or more inner *sections*, e.g., each `section_single_configuration_calculation` typically contains multiple iterations of `section_scf_iteration`. Also, the different *sections* depend on each other, e.g., the outcome in `section_single_configuration_calculation` depends on the system, method, and program defined in the higher layers of the structure.

The power and flexibility of this approach is schematically displayed in Fig. 2: We show the (simplified) representation following the metadata structure (according to NOMAD Meta Info (https://metainfo.nomad-coe.eu/nomadmetainfo_public/info.html)) of a simple calculation (a `section_run`) where one structure (defined in `section_system`) is evaluated with one electronic-structure method (defined in `section_method`). The results are reported in a single `section_single_configuration_calculation`, that in this example contains, besides the final results, also results from various scf iterations (`section_scf_iteration`). The metadata of *concrete* type (black font) have also *units* (in NOMAD Meta Info, we always use SI units) as attribute. In practice, `energy_total` could have *value* "$-1.344 \cdot 10^{-20}$", with *units* "J", for joule. *Concrete* metadata with associated values are contained in *sections* (red font) and *sections* can also be contained in parent *sections*, as graphically indicated by the boxes. *Sections* can also have references (dashed arrows) to other *sections*. In the present case, this is needed to relate the results contained in `section_single_configuration_calculation` with the physical model used to obtained them (`section_method`) and the geometrical structure used as input (`section_system`). In more complex cases, several `section_single_configuration_calculation` can have references to the same `section_method` (e.g., a geometry optimization) and/or to the same `section_system`" (e.g. the same system calculated with a self-consistent electronic-structure method, which is used as a starting point for a many-body perturbation method).

The standard definition of NOMAD Meta Info is maintained in a git repository (https://metainfo.nomad-coe.eu/nomadmetainfo_public/static/metainfo.html) and contributions are welcome. Access to the git repository can be granted by contacting the authors of this paper. The current metadata structure can be browsed at: https://metainfo.nomad-coe.eu/nomadmetainfo_public/info.html. In the current version of Meta Info, the metadata have been defined for virtually all information present in most of the electronic-structure codes. Work is in progress to define metadata for force-field (molecular mechanics) specific information. For instance, the topology (bond network) of the atomic system or, crucially, the detailed definition of the used atomistic force field. In this respect, integration with the API of the Knowledge of Interatomic Models (OpenKIM[8]) for the definition of the models is in progress. Besides the definition of new metadata, more work is needed to fully define the admissible *values* of many metadata. For instance, for the metadata `relativity_method`, we currently allow only a list of few options as *values* (https://gitlab.mpcdf.mpg.de/nomad-lab/nomad-meta-info/wikis/metainfo/relativity-method). Such list needs extension according to the actual methods implemented in parsed codes.

Other extensions, beyond the purely atomistic representation of systems (disordered alloys represented by fractional occupations of sites, coarse grained representation in soft matter, finite-element representation towards continuum, etc.) are possible. We note that the Meta Info infrastructure is fully extensible, thanks also to the enforced uniqueness of the metadata *names*, the unique identifier (a string) associated to each metadata, and the hierarchical structure that allows for opening a new section for any new group of needed information.

NOMAD Meta Info (https://metainfo.nomad-coe.eu/nomadmetainfo_public/info.html) is kept independent of the actual storage format as much as possible; it is a conceptual model, and in principle not bound to any specific storage method. We officially support three formats to store data using the NOMAD Meta Info: the human readable JSON, HDF5, and Apache Parquet. JavaScript Object Notation or JSON (www.json.org) is an open-standard file format that uses human-readable text to represent data objects consisting of key-value pairs and data of type array. Hierarchical Data Format (www.hdfgroup.org) is a set of file formats designed to store and organize large amounts of data. HDF5 is a binary, hierarchical, filesystem-like data format for the efficient storage of larger arrays and higher dimensional objects. Apache Parquet is a free and open source column-oriented data store of the Apache Hadoop ecosystem, designed for optimizing the queries over column values. This covers already quite different trade-offs with respect to human readability and storage efficiency, and has influenced and validated the choices done in the NOMAD Meta Info. The data could also be represented in other formats, like CIF or XML, and an official mapping to some of them could be added in the future. The Crystallographic Information File (www.iucr.org/resources/cif) is an archiving and information interchange standard for crystallographic and related structural science data promoted by the International Union for Crystallography. XML stands for eXtensible Markup Language (i.e., text plus annotations to mark its structure, such as sections, subsections, itemized lists, etc.) widely used across the internet.

The repository and archive infrastructures

For around 30 of the codes listed in Table 1, the NOMAD team have developed parsers, i.e., programs that convert the (open-access) raw data contained in the NOMAD Repository into the code-independent representation via NOMAD Meta Info. More parsers are under development, and a documentation is available for all code developers that would like to write a parser for their code. The parsers run routinely over the new additions to the Repository (or over the whole the Repository when parsers themselves are updated), and the converted data are stored in the NOMAD Archive. We distinguish between *standardized* and *normalized* data. *Standardized* data are mapped one to one from the data contained in the raw input and output to the corresponding metadata, with at most a unit conversion. *Normalization* involves additional processing of the data, e.g., a shift of the energy zero (see section "A common energy zero for total energies") or the representation of a band structure on a common *k*-point path (section "Electronic and vibrational properties of solids"). To a user, the Archive can be seen as a file system at http://data.nomad-coe.eu. The subfolders with the *standardized* and *normalized* data contain files organized according to NOMAD Meta Info, each subfolder containing one or more calculations (i.e. the content of one `section_run` after the parsing). This is a convenient way for providing access to all data in NOMAD, including bulk access, something that is critical for offering open access to big materials data. However, most users and web applications will need a more granular access to a subset of the parsed and normalized data. As this data is organized in a hierarchical way according to NOMAD Meta Info, data can be uniquely referred to using the newly developed NOMAD Resolve URI concept. The NOMAD Resolve URI is implemented as a REpresentational State Transfer (REST) API for the efficient browsing of the parsed and normalized data. In the Supplementary Information, we provide an example of accessing the Archive via NOMAD Resolve URI.

At https://analytics-toolkit.nomad-coe.eu/nomad-query-gui, we provide the first deployment of a graphical user interface for querying the content of the NOMAD Archive. It is possible to look for all calculations containing any given metadata (all the metadata listed in https://metainfo.nomad-coe.eu/nomadmetainfo_public/index.html), i.e., where the searched metadata has been assigned at least one value in a given calculation. For efficiency reasons, presently searches for specific values or intervals of values is allowed only for some selected metadata. Given the humongous size of the search space, technologies for supporting reasonably fast queries are currently being tested.

The result of a query is a list of NOMAD URIs. Via NOMAD Resolve URI, the user can access all values of metadata contained in the specific URI, further filter the data in order to prepare input and reference data for performing machine-learning and data-mining analyses. Tutorials to introduce the users to these functionalities are currently being prepared and will appear on https://analytics-toolkit.nomad-coe.eu/.

The practical usage of the NOMAD Repository and Archive infrastructure should be by now evident: the user can search over tens of millions of single point calculation for system compositions, calculation methods, and specific numerical approximations in order to start a new project and ascertain what has been already calculated or for performing new analyses for revealing yet unknown relationships among stored data. The hierarchical structure and the REST interface allow for the access to the value of each single metadata, yet maintaining the connection to the context to which each metadata belongs (e.g., for each result of a calculation, energy, forces, etc., what physical model and what atomic configuration determined it).

The electronic structure common data format

Whenever code developers are willing to design their inputs and outputs in a standardized format, the ESCDF library will provide the necessary common ground.

In particular, the ESCDF is focused on giving tools for developers to save the data produced by their code for the purpose of restarting a calculation, sharing the data with other codes, or further processing. This is achieved by standardizing the quantities to be put in an output file and via adopting self-describing formats like HDF5 or NetCDF, which are extensible and

allow the inclusion of metadata needed to interpret them. For a given code to be able to use the information produced by another code, the mere capability of reading it from a file is not sufficient. Indeed, as explained in the Introduction, the various electronic-structure codes use very different data representations, of which the type of basis set is the most paradigmatic example. This means that, after reading the data, it might be necessary to perform some conversions. Some are quite straightforward, like changing units, but other conversions can be quite involved, either requiring complex algorithms and/or be computationally demanding. For example, to be able to use wavefunctions that were written as a linear combination of Gaussian-type atomic orbitals, a plane wave code would have to perform a change of basis. Such transformations are beyond the scope of the ESCDF, but are an essential component of the NOMAD converters. From here the synergies between the two projects become clear: the ESCDF provides the tools for a standardized access to the data stored within the files, while the NOMAD conversion layer converts the data to a code-independent representation.

The first version of the ESCDF will include specifications to read/write the following type of data: geometry/structure of the system, basis sets, densities, potentials, and wavefunctions. At this point, the specifications do not aim at covering exhaustively all the quantities that an electronic code might need to read/write. Instead, the focus is on making sure the specifications are flexible and extensible. Because of its hierarchical structure and greater flexibility, HDF5 was chosen over NetCDF as the underlying file format. Each type of data is stored in an HDF5 group, which can be arranged in a way that is similar to a file system. This allows to store data for different use cases. For example, the most common use case is performing one calculation for one system, but the format should also be able deal with several systems that are calculated simultaneously. Using the NOMAD metadata and the experience accumulated by a previous standardization effort led by the European Theoretical Spectroscopy Facility (ETSF),[9] a set of specifications have already been agreed for the geometry/structure. In practice (see http://esl.cecam.org/ESCDF_-_System#NOMAD_Meta_Info), the ESCDF specifications for the geometry of the system (simulation cell, atom positions, etc.) follows the metadata contained in `section_system` of NOMAD Meta Info. Furthermore, work is currently underway regarding the representation basis sets and scalar fields. In a near future, *all* NOMAD metadata will be mapped into ESCDF attributes. In this way, any code that will adopt the ESCDF library for its output could be parsed into NOMAD Meta Info via a single "ESCDF parser".

The ESCDF software library and corresponding API will focus on flexibility, extensibility, and performance in order to maximize its usefulness and adoption by the community of code developers. In particular, the aim is at providing an API that does not force code developers to change the way how they store their data in memory, even in the case of parallel applications where the data is distributed among different processors. This is technically challenging, but essential if one wants to allow the code developers to focus on implementing new features and exploring new ideas instead of spending time porting and optimizing their codes to specific computer architectures. At present, no code is already making use of the ESCDF library. However, the ESCDF data format is essentially complete and a first implementation (for the code "octopus") is planned to be deployed soon.

A common energy zero for total energies

Many physical properties, e.g. forces, elastic constants, energy barriers, depend on total-energy differences. Therefore, they are well defined and the comparison of results from different codes is readily possible. To make also total energies stemming from different codes comparable, it is necessary to define a reference energy scale. To achieve this goal, a simple, pragmatic computational prescription viable for all codes is necessary.

Our idea is to define relative energies, where the energy of conveniently defined reference atoms is subtracted from the total energies calculated by each code. Ideally, the reference atoms would be isolated neutral atoms and we would then have formation/atomization energies as code-independent energies. However, it is well known that calculations for isolated atoms can be problematic for solid-state electronic-structure codes that use plane-wave basis sets (including augmented planewave ((L)APW) codes and alike). These codes are designed to study periodic systems, and the description of isolated atoms then requires large unit cells to ensure that the atoms do not interact. This makes the calculations expensive, and it may even prevent a systematic convergence study of all numerical settings, such as basis sets, grids, etc., for some atoms. To bridge the gap between periodic and non-periodic codes, both free atoms and simple bulk systems should be used as reference systems.

The coexistence of several reference-energy definitions is not a limit in the comparison as long as at least one code can encompass all definitions and evaluate all the reference values. Such code (or group of codes), would serve as a "Rosetta stone": This 2200 years old stele enabled the comparison and identification of ancient Egyptian hieroglyphs, demotic script, and ancient Greek. Here is a brief summary of the adopted strategy, which will be thoroughly discussed in a forthcoming publication.

When using free atoms as reference, the simplest choice is to define fully converged atoms (with respect to basis sets and integration grids). Their energies are calculated once for each physical model (exchange-correlation—xc—treatment) and, for pseudopotential-based codes, for each pseudopotential. In practice, free atoms are evaluated as spin unpolarized non-relativistic, in order to allow for a safe comparison among different codes where the implementation of the spin and relativistic treatments may be very different. However, other sets with spin-polarized atoms and selected relativistic treatments could be also stored, in order to obtain (atomization) energies that are closer to a physical meaning. Atomic energies evaluated at the same numerical (besides the physical) settings as the calculation of interest can also be considered. On one side this strategy allows for a well-known partial cancellation of numerical errors, on the other side the number of stored entries can potentially grow uncontrolled. To avoid this, a machine learning strategy that predicts the best atomic energy to be subtracted on the basis of a minimal, informative amount of stored information could be designed.

When using period bulk systems as reference, in a so-called "thermodynamic approach", the practical choice is to use the same crystal structures as used in K. Lejaeghere et al.[10,11] The few gaps in these publications (lanthanides and actinides) can be filled by using the structures reported in the CRC Handbook.[12] For species like O and N that form a molecular solid, where the combination of covalent and weak interactions may add numerical noise, one could also consider to use a binary compound where the other species of the binary compound forms a covalent or metallic crystal. Also in this case the practical choice is to use *fully converged* (with respect to basis sets, integration grids, and k grids) reference calculations, one for each physical model (including pseudopotential). Considering to subtract reference solids at the same numerical settings as the calculation of interest implies the same pros and cons as for the free atoms.

Electronic and vibrational properties of solids

Electronic band structures are typically represented along high-symmetry paths in the first Brillouin zone. In literature one can find a large variety of such paths, differing in directions and sequence

of the path segments. For an easy comparison between different calculations and codes, a practical choice is to represent all band structures along the paths defined in the paper by W. Setyawan and S. Curtarolo,[13] if these were calculated. Other properties of general interest are the density of electronic states and the effective masses. For electronic band structures, DOS, and related quantities, the energy zero can be conveniently set to the highest occupied Kohn–Sham level.

Calculations of harmonic vibrations in bulk materials (phonons) store all relevant information either in real space (force-constant matrix) or in reciprocal space (appropriate set of dynamical matrices), whereby it is important to note that, for non-polar materials, these two quantities are unambiguously related to each other via Fourier transformations. For polar materials, this data might be augmented by the dielectric tensor and the Born effective charges, which affect the long-wavelength vibrations in such materials. For calculations that contain them, force-constant matrices or dynamical matrices can be efficiently stored and they allow to compute other vibrational properties with negligible computational effort. For instance, vibrational band structures along the exact same paths used for the electronic band structures, but also densities of vibrational states and thermodynamic properties, such as specific heats can be easily derived. Obviously, an identifier of the original calculations used to obtain the vibrational properties should be stored along with the discussed vibrational properties. This ensures that the employed computational and physical settings can be retrieved, if needed.

Compact representation of scalar fields: density, wavefunction, xc potentials, etc.

The comparison of scalar fields across methodologies and codes requires to translate the internal, code and basis set-specific representation of these fields into a common format. For such a representation, an all-electron formalism is desirable, since it allows to evaluate additional properties, such as electric field gradients and NMR shifts.

Different codes use different basis functions which can be divided into two subclasses: localized functions and periodic functions. Popular choices for localized functions are Gaussian-type orbitals and numeric atomic orbitals (NAOs), while for periodic functions plane waves or augmented-plane waves are often used. Plane-wave basis sets are usually combined with pseudopotentials. Approximate all-electron wavefunctions can be straightforwardly restored from such pseudopotential calculations.[14]

To represent wavefunctions in a code-independent format, conversion to a universal basis set can be used. The conversion from the original basis ϕ_α to the universal basis η_β is performed by solving:

$$\sum_\gamma S_{\beta\gamma}\tilde{C}^i_\lambda = \sum_\alpha \langle \eta_\beta | \phi_\alpha \rangle C^i_\alpha, \quad (1)$$

where $\tilde{C}^i_\lambda$ and C^i_α are the coefficients of the expansion in the new and original basis, respectively, and S is the overlap matrix for the universal basis functions. Additional constraints can be taken into account when minimizing differences between the original and the universal representations, such as strict orthonormalization of the transformed wavefunctions.

The following two choices are most promising candidates for the universal basis:

- Gaussian basis functions. Online libraries of Gaussian basis sets are available (e.g., https://bse.pnl.gov/bse/portal).
- NAOs constructed once and for all for each species. A hierarchical construction (similar to Gaussian basis sets) of highly optimized NAO basis sets of various sizes is implemented in some codes, such as Dmol[3] and FHI-aims.

Which specific all-electron basis set is best suited for this purpose will be evaluated in detail in an upcoming publication.

In addition to the code-independent representation, a common storage format that allows for a quick restoration of scalar fields in the native, code-specific representation is useful, for example, for restarts from previous calculations. Such representation can be very compact because often only part of the information needs to be stored. The missing part can then be quickly obtained on the fly by running the code. For example, storing only plane-wave coefficients is sufficient for VASP to restore localized functions describing the wavefunctions near a nucleus, whereas an explicit storage of these localized functions would be very demanding. Since the concept of representing scalar fields in a basis-set expansion is common for all electronic-structure codes, defining a common, code-independent file format for storing the restart information for different codes is possible. Namely, one can store information identifying the functional form of a basis function (plane wave, contracted Gaussian, etc.), and the corresponding expansion coefficient. Obviously, such a representation cannot be used for code-independent data analysis, but it is beneficial from the practical point of view.

Quantities related to excited-state calculations

Advanced many-body perturbation theory (MBPT) calculations (*GW*, Bethe–Salpeter equation, etc.) currently output only few properties (spectra, self-energies, etc.) that need to be parsed and stored. To facilitate the analysis of this kind of calculations, it is essential to develop and store a detailed classification of all approximations used in the MBPT calculation in the metadata, given that many different numerical formalisms are implemented in different MBPT codes.

The *GW* approach is nowadays considered a routine approach for computing quasi-particle band structures. However, what is generically called *GW* comprises a lot of different approximations (besides those of the underlying ground state calculations). Presently, the situation is less transparent than for ground state. For this reason, it is necessary to store the following information:

- starting point (xc functional of the underlying density functional theory calculation),
- whether the calculation has been carried out in a perturbative manner or self-consistently (in fact, several types of self-consistency have been developed),
- further approximations, like plasmon-pole models,
- auxiliary basis sets used for non-local operators,
- numerical approximations, such as size of reciprocal space meshes, basis set size, frequency/time grid settings, etc.,
- whether the sum over unoccupied states is avoided/truncated/approximated,
- whether involved quantities are represented in real or reciprocal space,
- whether the Coulomb potential is truncated.

Obviously, all these approximations must be labeled by appropriate metadata (see also section "Metadata for the code-independent format"). *GW*-related quantities of interest to be stored are:

- matrix elements of the exchange and correlation contributions to the self-energies evaluated at the Kohn–Sham states in case of G_0W_0. For self-consistent *GW* calculations, the matrix elements are evaluated with quasi-particle states.
- matrix elements of the xc potential,
- quasi-particle energies,
- spectral functions (if calculated).

For optical spectra determined by time-dependent density functional theory or the solution of the Bethe–Salpeter equation the situation is similar to what was described above. The quantities of interest are:

- excitation spectra, energies, and oscillator strengths,
- exciton binding energies (if available),
- spectra obtained by the independent-particle approximation and/or Kohn-Sham response function.

Establishing error bars, uncertainties, and confidence levels

Quantifying the errors and uncertainties of the data included in computational materials' databases is an essential step to make this data useful. Challenges in this field arise, since the errors are code, material, and even property specific. Also, the dependence of different errors on each other needs to be taken into account. A first step in this direction is to establish unique identifiers for structures through "similarity recognition". In this way, group of calculations performed on atomic structures that are not identical but obtained by small distortions of the coordinates and/or the unit cell can be automatically recognized as similar and an error analysis can be performed. Furthermore, a systematic investigation of numerical errors is required across codes for both simple and complex properties, which also requires a clear definition of errors/deviances, e.g., for continuous functions. With respect to errors arising from the use of approximated xc-functionals (more) test sets are required as a reliable, high-level reference. The use of an experimental benchmark is avoided, for the moment, as temperature and pressure conditions, intrinsic defects and impurities, surfaces, or dislocations can make it difficult to obtain unambiguous (test) sets of reference values from experiments.

For a given atomic configuration, there are different error bars corresponding to the different approximations:

- basis set,
- choice of pseudopotential (if employed),
- grids and other numerical approximations,
- *k*-mesh,
- treatment of relativity,
- xc functional.

Thus, every calculated result stored in a code-independent format should be connected using the method-related metadata with six numbers that refer to the mentioned (mostly code-specific) approximations. Obviously, energies and energy differences will be associated with different error bars, just to mention one example. In general, the importance of these error bars depends on the material's property of interest. As a first step in the direction of assigning error bars to calculations, the NOMAD team has developed an interactive web-based tutorial (https://analytics-toolkit.nomad-coe.eu/tutorial-errorbars) for estimating the relative and absolute errors as function of the numerical settings, e.g., basis sets and *k*-grids. The estimate is based on a database of about 100,000 calculations performed with four different electronic-structure codes (exciting, FHI-aims, GPAW, VASP, encompassing very different choices for basis sets and numerical solvers) for 71 elemental solids.

The mentioned error-bar contributions may be evaluated from any dataset that contains results corresponding to the same material, but using different approximations and/or different codes. However, it will also be necessary to evaluate relevant quantities at different levels of approximations systematically. In fact, this will also yield a "test set for materials" which is a well-known and most useful concept in quantum chemistry but largely absent, so far, in materials science. The build-up of a "test set for materials science and engineering" has been initiated. Some such studies are being performed by the groups of G. Kresse in Vienna, and others by the NOMAD team.

In addition to the numerical errors discussed above, it is essential to develop means to assess the possible error coming from the actual implementation (coding) of the various atomic-scale calculations. This requires the establishment of high-quality benchmark calculations for various materials and properties. For ground-state calculations, the work by Cottenier and co-workers[10,11] for bulk materials represents a first and important step in this direction. The next step, which considers low symmetry situations, e.g. defects and surfaces is just initiated. For excited-state codes the GW100 paper[15] comparing TURBOMOL, FHI-aims, and BerkleyGW for quasi-particle energies of molecules, represents a first step towards this goal.

CONCLUSIONS

Motivated by the compelling need in the materials-science community for sharing and exchanging data, we have described challenges and practical strategies for achieving a common format for the representation of computational materials science data. In the context of the NOMAD Laboratory Centre of Excellence, we presented the definition and implementation of NOMAD Meta Info, a hierarchical and extensible metadata infrastructure for the efficient sharing of the data, the NOMAD Archive, a code-independent storage of the data by means of Meta Info, and NOMAD URI Resolve and Query, the tools for efficiently accessing any single metadata in the Archive. This opens the unprecedented possibility to guarantee the full reproducibility of materials science data and allow for new discoveries on stored data, thanks to the emerging big-data analytics methodologies. We also described a library for the direct output of present and future electronic structure codes into a standard format (the ESCDF initiative).

For several crucial topics, like the common energy zero-level, electronic and vibrational properties, the scalar-field representation, and excited states calculations, we present practical choices for achieving our goal of a code-independent representation. We also present the challenge of establishing reliable error bars and uncertainties for each calculation stored in a database, in terms of the adopted numerical settings.

At present, the metadata infrastructure deals with computational data only. The next step is to extend the infrastructure also to experimental data. This is a challenging task as a full characterization of an experimental measurement needs not only the definition of the probed system and the adopted measurement technique, but also the description of the preparation history of the specimen.

ACKNOWLEDGEMENTS

We thank James Kermode and Saulius Gražulis for their contribution to the discussion on the metadata, and Pasquale Pavone for precious suggestions on the metadata structure and names. We thank Patrick Rinke and Ghanshyam Pilania for carefully reading the manuscript. We thank Claudia Draxl and Kristian Thygesen for their contribution to the discussions on the necessary information to be stored for excited-state calculations and on the error bars and uncertainties. We gratefully acknowledge Damien Caliste, Fabiano Corsetti, Hubert Ebert, Jan Minar, Yann Pouillon, Thomas Ruh, David Strubbe, and Marc Torrent for their contributions to the ESCDF specifications. We acknowledge Benjamin Regler for the development of the graphical interface for the query on the NOMAD Archive. We acknowledge inspiring discussions with Georg Kresse, Peter Blaha, Xavier Gonze, Bernard Delley, and Jörg Hutter on the energy-zero definition and scalar-field representation. We thank Ole Andersen, Evert Jan Baerends, Peter Blaha, Lambert Colin, Bernard Delley, Thierry Deutsch, Claudia Draxl, John Kay Dewhurst, Roberto Dovesi, Paolo Giannozzi, Mike Gillan, Xavier Gonze, Michael Frisch, Martin Head-Gordon, Juerg Hutter, Klaus Koepernik, Georg Kresse, Roland Lindh, Hans Lischka, Andrea Marini, Todd Martinez, Jens Jørgen Mortensen, Frank Neese, Richard Needs, Taisuke Ozaki, Mike Payne, Angel Rubio, Trond Saue, Chris Skylaris, Jose Soler, John Stanton, James Stewart, Marat Valiev for checking the information provided in Table 1 and for useful suggestions. This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No. 676580, The

NOMAD Laboratory, a European Center of Excellence, and the BBDC (contract 01IS14013E).

AUTHOR CONTRIBUTIONS

L.M.G. and M.S. designed and lead the project. The various concepts discussed in the paper were worked out and implemented by all authors. All authors contributed to the writing of the paper.

ADDITIONAL INFORMATION

Supplementary information accompanies the paper on the *npj Computational Materials* website (https://doi.org/10.1038/s41524-017-0048-5).

Competing interests: The authors declare that they have no competing financial interests.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

REFERENCES

1. Kang, B. & Ceder, G. Battery materials for ultrafast charging and discharging. *Nature* **458**, 190–193 (2009).
2. Wood, B. C. & Marzari, N. Dynamics and thermodynamics of a novel phase of $NaAlH_4$. *Phys. Rev. Lett.* **103**, 185901 (2009).
3. Yang, J., Sudik, A., Wolverton, C. & Siegel, D.J. High capacity hydrogen storage materials: attributes for automotive applications and techniques for materials discovery. *Chem. Soc. Rev.* **39**, 656–675 (2010).
4. Jain, A. et al. A high-throughput infrastructure for density functional theory calculations. *Comput. Mater. Sci.* **50**, 2295–2310 (2011).
5. Potyrailo, R. et al. Combinatorial and high-throughput screening of materials libraries: review of state of the art. *ACS Comb. Sci.* **13**, 579–633 (2011).
6. Castelli, I. E. et al. Computational screening of perovskite metal oxides for optimal solar light capture. *Energy Environ. Sci.* **5**, 5814–5819 (2012).
7. Curtarolo, S. et al. The high-throughput highway to computational materials design. *Nat. Mater.* **12**, 191–201 (2013).
8. Tadmor, E. B., Elliott. R. S., Sethna, J. P., Miller, R. E. & Becker, C. A. *Knowledgebase of Interatomic Models* (*KIM*) https://openkim.org (2011).
9. Gonze, X. et al. Specification of an extensible and portable file format for electronic structure and crystallographic data. *Comput. Mater. Sci.* **43**, 1056–1065 (2008).
10. Lejaeghere, K., van Speybroeck, V., van Oost, G. & Cottenier, S. Error estimates for solid-state density-functional theory predictions: an overview by means of the ground-state elemental crystals. *Crit. Rev. Solid State* **39**, 1–24 (2014).
11. Lejaeghere, K. et al. Reproducibility in density functional theory calculations of solids. *Science* **351**, 1394–1395 (2016).
12. *Table "Standard Thermodynamic Quantities for Chemical Substances"*. In CRC Handbook of Chemistry and Physics, David R. Lide, ed., CRC Press, Boca Raton, FL, 2005.
13. Setyawan, W. & Curtarolo, S. High-throughput electronic band structure calculations: challenges and tools. *Comput. Mater. Sci.* **49**, 299 (2010).
14. Van de Walle, C. G. & Blöchl, P. E. First-principles calculations of hyperfine parameters. *Phys. Rev. B* **47**, 4244–4255 (1993).
15. Van Setten, M. J. et al. *GW*100: benchmarking G_0W_0 for molecular systems. *J. Chem. Theory Comput.* **11**, 5665–5687 (2015).