PSDI
PHYSICAL SCIENCES
DATA INFRASTRUCTURE

Case Study 2: Exploring CSD-Theory as a tool for assisting materials discovery

Christopher R. Taylor [1], *Joseph C.R. Thacker* [2], *Andrew I. Cooper* [2], *Simon J. Coles* [1], *Graeme M. Day* [1]

[1]*Department of Chemistry, University of Southampton, Southampton, SO17 1BJ*
[2]*Materials Innovation Factory & Department of Chemistry, University of Liverpool, Liverpool, L69 7ZD*

1 Background and Motivation

1.1 Crystal structure prediction and materials discovery

Computational organic crystal structure prediction (CSP) combined with high-throughput materials synthesis and characterisation offers the possibility of vastly accelerated discovery of functional materials. One bottleneck in this approach is the sharing and analysis of computational results in a manner conducive to experimental exploitation. The comparatively large volume of data produced by a typical CSP workflow consists (for example) of sets of hypothetical crystal structures, computed descriptors and properties, and metadata such as computational parameters. This data must be summarised and analysed in a fashion that allows experimental colleagues to make rapid, reliable comparisons with their own results, and to determine if experiments can be usefully modified in light of any insight gained from these calculations.

As a concrete example, recent work[1] by several of the current authors demonstrated the success of CSP in predicting the structure of a polymorph of trimesic acid (TMA), for which only limited experimental data could initially be obtained, in the form of a powder X-ray diffraction (PXRD) pattern. Because the computational results can be obtained independently of any experimental workflows, there is clear scope here for simulation to "pull ahead" and create libraries of e.g. PXRD patterns of hypothetical crystal structures for molecules under investigation, against which the experimental workflow could compare samples. Ideally, this experimental setup is itself automated, with autonomous robotic apparatus systematically characterising samples and comparing to computed predictions, flagging observed matches for further study. In order to do this, however, experimentalists must have appropriate tools to explore and understand the computational results, whether before or after they have been prioritised by such an automated workflow.

Thus, our case study focusses on our requirements for the ideal form of the PSDI to support such workflows. In particular, we desire straightforward import/ingestion of existing datasets, tools to more easily facility collaborative work with experimentalists such as visualisation and GUI-based/automated analysis methods, amenability to fully-automated workflows (e.g. the use of APIs and standard formats for accessing and contributing to stored datasets). We present trimesic acid, and the case of matching an experimental, unsolved PXRD pattern to CSP-derived predictions, as an example of a situation where the PSDI would ideally make the existing workflow simpler, more automatable, and more readily available to experimental collaborators.

1.2 The CSD-Theory software

It is in this context that the Cambridge Crystallographic Data Centre (CCDC) offered for this case study to explore their CSD-Theory system, which is currently under active development as part of the Crystal Structure Database (CSD). CSD-Theory as it stands consists of extensions to the CSD's Python API and the WebCSD platform that allows the creation of databases of crystal structures with associated properties and metadata that are compatible with both the Python API and the WebCSD. Through the Python API, this offers a mature implementation (with documentation and support) of software for carrying out basic analysis of these structures, such as simulating PXRD patterns, calculating void volumes, and describing hydrogen bond networks. Meanwhile, the WebCSD allow for GUI-based searching of these databases using the CSD's existing search tools (encompassing both simple text-based search queries and more advanced forms, such as searching by chemical structure) and the automatic production of useful images and figures (such as the relative energy vs density plot, a ubiquitous means of summarising CSP results in the literature).

2 Activity and Outputs

2.1 A note on CSD-Theory: under active development

Before discussing our activity and experience in exploring the CSD-Theory system, we feel it appropriate to emphasise that this software remains in ongoing development by the CCDC and does not necessarily represent a complete product. We therefore preface our analysis by noting that any criticisms, recommendations, or feature requests may already be active topics of discussion or implementation at the CCDC.

2.2 Installation

This study began with the installation of the CSD-Theory software and related tools from the CCDC on a shared virtual machine, assisted by technical support from the CCDC. (The virtual machine consisted of a 2 virtual CPU, 8 GiB CentOS system hosted on the Microsoft Azure service.) This process was relatively straightforward, demonstrating an attractive ease-of-setup for collaborative projects.

2.3 Bringing CSP datasets into CSD-Theory

Next, we explored how to ingest CSP data – computed crystal structures and their properties – into the CSD-Theory system. Unfortunately, at present the database schema used internally by the CSD-Theory software is closed – instead, the CCDC provides utilities for importing (and deleting) annotated CIF files into the database. Therefore, it is not currently possible to directly prepare databases programmatically for use with CSD-Theory.

Instead, we wrote a short Python script (around 50 lines of code) to annotate the CIF files from existing CSP datasets to be then imported into CSD-Theory using the supplied utility. The CCDC has been involved in developing a standard vocabulary of CIF annotations for CSP-derived data, including data fields for relative lattice energies, optimisation model (e.g. force-field, DFT, etc.), and other computational data and metadata, which are then used to add the relevant information to the resulting database. The dictionary defining these annotations is extensive and allows for a wealth of CSP information to be imported, and we acknowledge the CCDC in driving the development of this.

The importing of structures into CSP-Theory-compatible databases was relatively straightforward using the provided utilities. However, we make several minor observations with relevance to the PSDI. Firstly, as discussed above, the lack of an open database schema means structures must be first be written to intermediate (annotated CIF) files, before being added to the database. Ideally, such an intermediate step would be unnecessary, and CSP researchers could directly prepare databases in the appropriate format, either as part of their workflows or in a post-hoc processing step. This would obviate the need to write intermediate CIF files (even temporarily) and would expedite the addition of data to such databases. (We do acknowledge that ideally the data addition step only takes place once, but given the volume of potential CSP data, including historical datasets, that would be of interest in adding to such databases, we feel it is worth highlighting nonetheless.)

Secondly, related to the above, we considered the speed and scaling of the import process with respect to the number of structures being considered. In Figure 1, we present timings for a small series of import runs, with numbers of included structures in each run ranging over approximately three orders of magnitude. We see linear scaling with the number of structures to be included, which is expected in principle but reassuring as it demonstrates that larger datasets are not an undue hindrance. It should be said however that the speed of import might cause issues with very large datasets – importing approximately 17,000 structures into one database took just under 25 minutes on this virtual machine. (Again, we recognise that this is in principle a one-time cost per dataset, but datasets of these size are routinely generated in CSP research.) If the timings presented here are not due to a hardware bottleneck, then for very large datasets it could prove beneficial to have a means of straightforwardly combining databases, to allow separate parallel preparation of smaller subset databases which can then be "concatenated" into one master database. It could also prove beneficial for collaborative projects, where separately-produced databases could be straightforwardly combined.

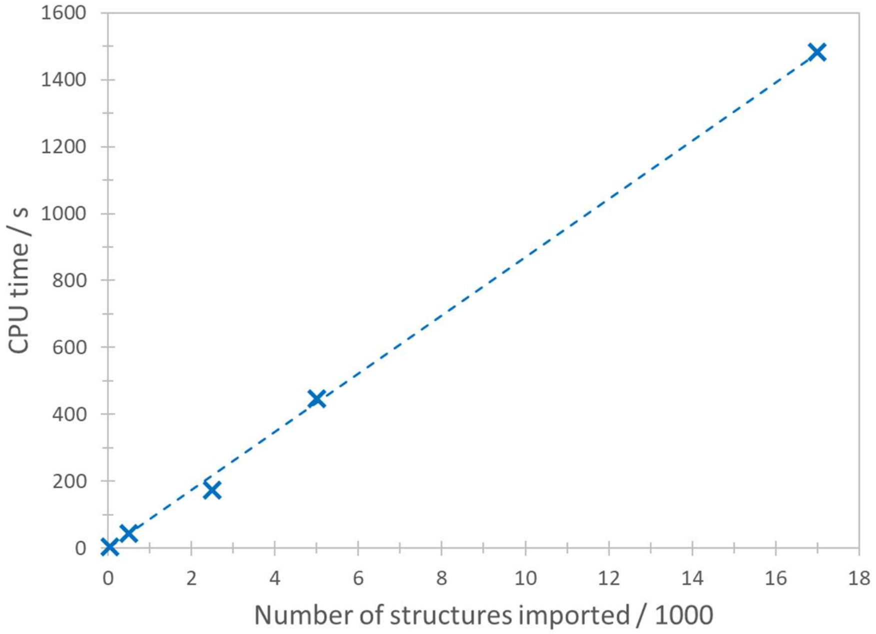

Figure 1: Time for import into a CSD-Theory database using the supplied script for ingesting annotated CIF files. The expected linear scaling is apparent over a range of dataset sizes (line-of-best-fit from linear regression shown for clarity).

2.4 Visualising imported CSP data

Ideally, any solution for storing, sharing, and standardising data in the physical sciences would include straightforward means to visualise said data where appropriate. This is particularly attractive in the case of collaboration between simulation and experiment – graphical summaries of datasets can provide much more rapid, intuitive, and lasting insight. One of the key features of the CSD-Theory system is its compatibility with the WebCSD, the CCDC's existing browser-based tool for searching its databases and visualising structures. Through the WebCSD, CSD-Theory is capable of automatically producing the conventional "CSP landscape plots" common to this field, providing an overview of a CSP landscape as a graph of structure ranking (very frequently either the relative cohesive energy or the relative free energy of a structure) against some physical descriptor, typically the density.

We tested these visualisation tools on the aforementioned imported datasets using the WebCSD in a typical web browser (Google Chrome). One significant technical limitation we encountered, which is known to the CCDC, is that a CSP landscape is assumed to have a maximum of (approximately) 1000 structures in it and thus all points in a landscape can be plotted at once; in contrast, our initial test dataset of trimesic acid contained nearly 27,000 unique structures. As a result, the visualisation procedure timed out trying to retrieve so many structures from the database and generated no landscape plot. When we instead tested searching a much smaller database (around 100 structures of acetic acid), we could successfully visualise both the individual structures and the landscape plot as expected.

While the visualisation and plotting features of CSD-Theory are particularly attractive for collaboration, this testing highlighted a discrepancy in expectations between the current implementation and the needs of researchers working on CSP datasets. It is increasingly typical for experimentally-relevant CSP datasets to contain tens of thousands of unique structures – while these do not necessarily all need to be plotted at one time, it is crucial that the internal representation of a CSP landscape in software to be used as part of the PSDI is capable of handling such numbers of entries. One solution suggested in conversation with the CCDC developers, at least for the problem of graph-plotting, is implementing a filter to admit only certain structures (e.g. below a certain relative energy) to be plotted. While this would be useful in this specific context, the greater issue is whether the CSD-Theory tools in general are designed around smaller datasets than we would be interested in sharing with experimentalists, particularly in future, larger-scale projects.

2.5 Example application: matching predicted structures via PXRD

As mentioned in Section 1.1, our test system for the capabilities of CSD-Theory to streamline collaboration is the combined computational-experimental screening of trimesic acid (TMA) carried out previously by Cui *et al.*[1] In this previous work, a new, porous polymorph of TMA, subsequently named the delta form, was predicted by CSP and experimentally obtained via a high-throughput robotic crystallisation screening procedure. The crucial connection between the two processes in this work was the determination of PXRD patterns for simulations and samples, and the comparison of the two.

In such experimental screens, hundreds of potentially unique samples may be obtained and must be rapidly characterised and differentiated, and PXRD is an excellent tool for this. Equally, PXRD patterns can be straightforwardly simulated for every structure in a CSP dataset, and this allows comparison of the experimental characterisations and the predicted set. Thus, experimental matches (or similarities) to predicted forms can be rapidly identified and selected for further analysis. (In the case of TMA, similarity of several solvated samples to a predicted low-density form – and their dissimilarity to known forms – motivated desolvation experiments that led to a single, desolvated porous form, δ-TMA.)

Clearly, it is crucial in such an application that the simulation and comparison of PXRD patterns is automatable just as the experimental characterisation is. In the published work, this was done using our own in-house software. However, an attractive feature of the CSD-Theory system is its close integration with the CSD's Python API, through which PXRD patterns can also be computed and compared. In principle, a database for a system of interest could be produced via CSP and then searched at a later date by an experimentalist for PXRD matches to an experimental pattern, without requiring active intervention on the part of the computational researcher – only initial preparation of appropriate scripts for interacting with the database of predictions. (Such scripts could also more easily be prepared by the experimentalist themselves than in the case of our custom software, given the extensive documentation of the CSD's Python API.) This could "decouple" the CSP researcher from the experimental timetable and allow more flexible, dynamic collaboration, as well as expedite the identification of experimental matches to predicted structures.

To assess the ease with which this kind of analysis could be automated using the CSD-Theory system, we used our previously-imported database of 27,000 unique TMA structures – the same structures predicted in the published work – and wrote another short Python script invoking the CSD Python API's PXRD simulation and comparison methods. Initially we encountered some technical issues that prevented us from accessing any experimental CSD data through the WebCSD (command line scripted queries were functional). However, a later build of the software provided by the CCDC rectified this and allowed comparison to the structures held in the CSD.

Ranking the structures by PXRD similarity to the experimental determinations of δ-TMA obtained in the published work, CSD refcodes BTCOAC03 and BTCOAC04 (obtained at 350 K and 200 K respectively), our highest-ranked structures correspond to those observed in the low-density "spike" in favourable lattice energies described therein. The CSD-Theory system and Python API therefore combine to allow straightforward, automatable – and scriptable – identification of matches in the database to experimental data, as was originally carried out in the published work using custom software. Such ease of automation and portability is an extremely attractive feature of the CSD-Theory system, and it is highly desirable for both this sort of analysis and the simplicity with which it is accomplished to be available to the PSDI.

3 Conclusions and recommendations

3.1 Summary of outputs

We explored the use of the CSD-Theory system as a potential component of, or example of desirable functionality for, the PSDI for enhancing collaboration between CSP simulation and experimental screening for materials design applications. There are a number of attractive features of the CSD-Theory system as it stands that motivate our recommendation that it continue to be explored and tested for this purpose. – the ease of setup, the integration with the CSD's Python API, and the availability of visualisation tools through the WebCSD. Any alternative system for the storage, manipulation, and analysis of crystal structures in the PSDI would ideally share similar features to encourage adoption.

From a technical standpoint, the CSD-Theory system is satisfactory, particularly for an unreleased product that remains under active development. The import process for annotated CIF data is reasonably straightforward, though ideally it would be possible to prepare compatible databases directly, without intermediates. We also note that the current limitation on assumed landscape size (on the order of 1000 crystal structures) is considerably smaller than a typical modern CSP dataset, and this limit particularly impacted the visualisation and plotting components of the system that would be among the most immediately useful to experimental collaborators. If it is to be employed for this use in the PSDI then this limit must be raised or circumvented somehow (e.g. through filters as proposed by the CCDC during informal conversation); any alternative implementation in the PSDI must similarly deal with the reality that current state-of-the-art CSP research is likely to result in considerably larger datasets than has been assumed (at least 10^4 crystal structures and associated metadata, if not 10^5 or more). While it is not necessarily the case that these all need to e.g. be plotted on the same set of axes, it is crucial that any system for this kind of collaboration not be limited thus in its other functionality (e.g. analysis functions, import and export, etc.).

However, in light of the above issues, we must acknowledge and commend the CCDC for the quality and responsiveness of their support and feedback during this case study. We recognise this is an unreleased product and there were likely to be technical issues during this study; just as important to the PSDI as functionality is long-term support and responsiveness from any providers of software or services (e.g. training or consultation), and we are pleased to report that the CCDC was generally excellent in this regard.

The main test for assessing the CSD-Theory system's suitability for our purposes was the automated matching of predicted structures' PXRD patterns to the experimentally-known patterns, which proved to be very successful. With relatively small and easily portable Python scripts (totalling less than 100 lines of code), we were able to create a compatible database of over 26,000 predicted structures and search for PXRD similarities to those from experiment in much the same way as was done in the published work. The speed and relative ease with which this was possible using CSD-Theory and the Python API considerably lowers the "barrier to entry" for performing or automating this sort of analysis. The PSDI would greatly benefit from this or similar functionality; straightforward automation and portability of analysis methods would drive more rapid adoption of the infrastructure.

3.2 Recommendations

Our recommendations for the PSDI from this case study are likely evident from the above discussion, but we explicitly enumerate them below for clarity.

1. We strongly recommend continued exploration and testing of the CSD-Theory system as a potential component, or at least as a model of desired functionality, for the PSDI. Close cooperation between simulation and experiment in solid form screening has already demonstrated significant successes, and tools like those in the CSD-Theory system will only streamline such cooperation, and are thus crucial to facilitate in the PSDI as we see it.
2. We recommend that the CCDC and research community collaborate (cf. the CIF annotation dictionary) to produce an agreed-upon, open database schema for the storage of CSP data, to allow researchers to directly produce databases compatible with the CSD-Theory system. In principle, such a schema could itself form part of the PSDI.

3. We recommend that any implementation of CSP-relevant analysis or visualisation tools in the PSDI consider that the size of modern CSP datasets can run into hundreds of thousands of structures. If the PSDI is to prove useful for collaboration between experiment and simulation, its tools must be scalable to this degree.
4. We also recommend, related to this case study but not a direct output thereof, that whatever system is implemented in the PSDI for the curation and sharing of CSP data ensure that a unique, citable identifier is created for every dataset that is stored, whether this is in the context of the CSD-Theory system or an alternative implementation. Correct, easy attribution of datasets to their originators (as opposed to only the repository) will ensure that researchers' efforts are appropriately recognised and thus encourage their adoption of the PSDI.

4 Acknowledgements

C.R. Taylor and J.C.R. Thacker acknowledge the STFC PSDI Pilot programme for funding while carrying out this case study. We are grateful to the CCDC for making the CSD-Theory system available to us for this study, as well as specifically David Bardwell and Pete Wood at the CCDC for technical support and guidance in the use of the software.

5 References

1 P. Cui, D. P. McMahon, P. R. Spackman, B. M. Alston, M. A. Little, G. M. Day and A. I. Cooper, *Chem. Sci.*, 2019, **10**, 9988–9997.