Identifying Duplicate Crystal Structures: XtalComp, an Open-Source Solution

David C. Lonie[a], Eva Zurek[a,*]

[a]*Department of Chemistry, State University of New York at Buffalo, Buffalo, New York, 14260-3000*

Abstract

We describe the implementation of XtalComp, an efficient, reliable, and open-source library that tests if two crystal descriptions describe the same underlying structure. The algorithm has been tested and found to correctly identify duplicate structures in spite of the "real-world" difficulties that arise from working with numeric crystal representations: degenerate unit cell lattices, numerical noise, periodic boundaries, and the lack of a canonical coordinate origin. The library is portable, open, and not dependent on any external packages. A web interface to the algorithm is publicly accessible at `http://xtalopt.openmolecules.net/xtalcomp/xtalcomp.html`.

*PACS:*61., 61.50.Ah, 61.50.-f

PROGRAM SUMMARY

Program Title: XtalComp
Journal Reference:
Catalogue identifier:
Licensing provisions: "New" (3-clause) BSD [1]
Programming language: C++
Computer: No restrictions
Operating system: All operating systems with a compliant C++ compiler
Keywords: Duplicate, Structure, Crystal, Crystalline, Computational Crystallography, Matching, Similarity
Classification: 7.8 Structure and Lattice Dynamics
External routines/libraries:

*Corresponding author.
E-mail address: ezurek@buffalo.edu (E. Zurek)

Subprograms used:
Nature of problem: Computationally identifying duplicate crystal structures taken from the output of modern solid state calculations is a non-trivial exercise for many reasons. The translation vectors in the description are not unique – they may be transformed into linear combinations of themselves and continue to describe the same extended structure. The coordinates and cell parameters contain numerical noise. The periodic boundary conditions at the unit cell faces, edges, and corners can cause very small displacements of atomic coordinates to result in very different representations. The positions of all atoms may be uniformly translated by an arbitrary vector without modifying the underlying structure. Additionally, certain applications may consider enantiomorphic structures to be identical.
Solution method: The XtalComp algorithm overcomes these issues to detect duplicate structures regardless of differences in representation. It begins by performing a Niggli reduction on the inputs, standardizing the translation vectors and orientations. A transform search is performed to identify candidate sets of rotations, reflections, and translations that potentially map the description of one crystal onto the other, solving the problems of enantiomorphs and rotationally degenerate lattices. The atomic positions resulting from each candidate transform are then compared, using a cell-expansion technique to remove periodic boundary issues. Computational noise is treated by comparing non-integer quantities using a specified tolerance.
References:
[1] `http://opensource.org/licenses/BSD-3-Clause`

1. Introduction

The computational data objects used to store crystalline structure information typically describe a single unit cell: the cell's translation vectors and the identities and positions of the atoms in a single translation unit. This storage paradigm is simple, convenient, and intuitive to users familiar with periodic solid-state systems. Such a structural description fully describes the unbounded physical system, but is not unique; due to periodicity and the many degrees of freedom, there are infinite unit cell descriptions corresponding to a given structure. As a result, we will make a strong distinction between a **description** or **representation** of a crystal (i.e. a single unit cell), and the underlying **structure** (the infinite system).

Such finite descriptions are sufficient for most purposes, but the lack of a canonical descriptive standard introduces problems, particularly when it comes to determining the equivalence of two descriptions. Comparing two finite representations is not a straightforward task – Figure 1 shows four distinct, valid unit cell descriptions that share the same underlying structure. From visual inspection alone, it is clear that comparing the geometric properties of the infinite systems underpinning these finite unit cells will not be a straightforward task. The problem is not any easier when examining the numeric values from a unit cell data object: Table 1 shows the raw numeric data for two descriptions of the same structure.

Characteristic	**Description 1**	**Description 2**
$\vec{a}$	(3.16, 0.00, 0.00)	(6.00, 0.00, 0.00)
$\vec{b}$	(-.95, 4.14, 0.00)	(1.00, 3.00, 0.00)
$\vec{c}$	(-.95, -.22, 4.13)	(2.00,-3.00, 3.00)
Atom 1 (Type A)	(0.44, 0.40, 0.30)	(0.00, 0.00, 0.00)
Atom 2 (Type A)	(0.94, 0.40, 0.79)	(0.00, 0.00, 0.50)
Atom 3 (Type B)	(0.45, 0.90, 0.79)	(0.50, 0.00, 0.00)
Atom 4 (Type C)	(0.94, 0.40, 0.29)	(0.00, 0.50, 0.00)

Table 1: Two numeric descriptions of the same structure. The translation vectors, atom types, and fractional coordinates are given. It is not possible to determine that these two representations are equivalent from a simple examination of the descriptions.

Our interest in performing such comparisons is to improve the duplicate matching performance of the crystallographic evolutionary algorithm XtalOpt[1–3].

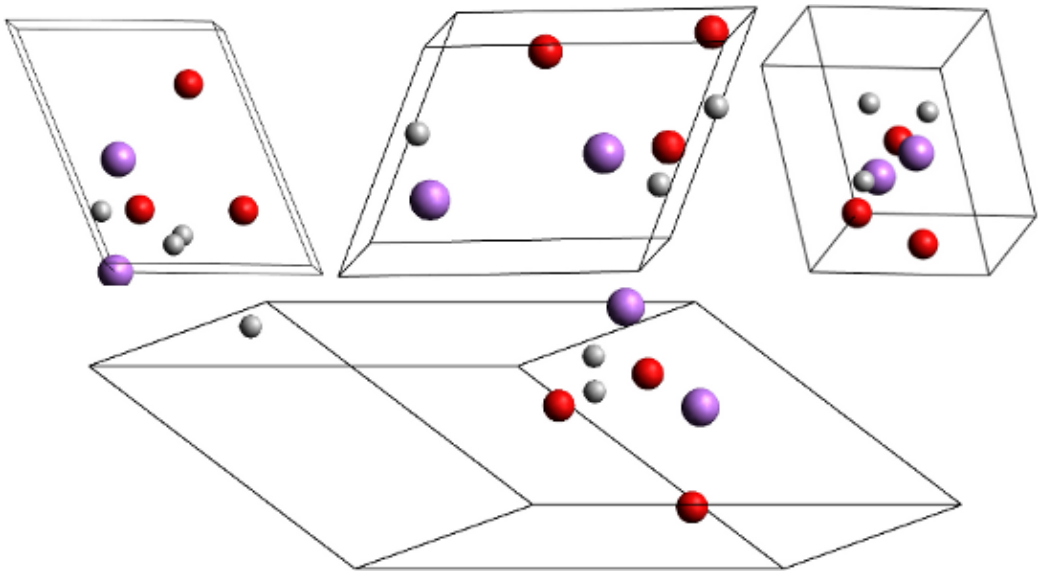

Figure 1: Four distinct descriptions of the same structure. Red, white, and purple spheres represent different atom types, black lines denote the edges of the unit cell. A color version of this figure is available online.

This software (coupled with an external code used for structure relaxation and enthalpy evaluation) is used to predict thermodynamically stable crystal structures by mutating and mixing individuals from a pool of locally minimized configurations. The "parent" structures are chosen using an enthalpy-weighted probability distribution, and the presence of duplicate crystals in the pool unfairly increases a structure's odds of being selected. Thus, the identification and removal of redundant structures from the pool is essential for maintaining an accurate probability distribution. Without this crucial step the pool would stagnate and become fixed on a small number of structures, destroying the diversity necessary for a successful evolutionary search.

To carry out the duplicate removal we need an equality operator for crystal descriptions that meets the following requirements: (1) It must be freely available under an open-source license. XtalOpt is released under the GNU Public License, and in order to integrate a comparison algorithm it must be released under a compatible license. (2) We make the distinction between **exact** comparison techniques that compare atomic positions directly and **approximate** methods that reduce the descriptions into intermediate forms, which are then compared. The desired algorithm must perform an exact comparison. The original duplicate matching tech-

nique in XtalOpt is an example of an approximate method – it compared each structure's "fingerprint", consisting of the structure's space group, volume, and enthalpy. This provides a reasonable guess as to the equality of the descriptions, but is unreliable under certain conditions (see below).

In the field of automated computational structure prediction, many such approximate methods are used with reasonable success. There are some that compare enthalpy alone[4–7]. Such enthalpy-based equality operators are prone to false positives, especially when searching near a phase transition or other situation where many unique structures have similar enthalpies. This can be improved by supplementing the enthalpy comparison with additional criteria[8].

The previous duplicate matching algorithm used by XtalOpt augmented the enthalpy-matching method by adding two other description-independent metrics: the unit cell volume and space group. While this helped to eliminate some of the false positives described above, it introduced its own set of difficulties. Low symmetry structures (say, *P1*) are very common towards the beginning of a randomly-seeded structure search, and certain systems (again, such as those near a phase transition) may have many low symmetry structures that are closely clustered energetically. In these low symmetry situations, the space groups of most structures match, leading to the same problems seen when only comparing the enthalpies. Additionally, accurately determining a crystal structure's space group requires a bit of intuition and experimentation to correctly set the tolerances. This human element is difficult to reproduce algorithmically.

This is not to say that approximate methods are without merit. Recent work by Valle, Lyakhov, and Oganov[9–12] shows that their crystal-fingerprinting technique, an approximate method based on interatomic distances, provides a useful measure of similarity that may be used to guide a structure search and offer insight into the chemical system[11, 12]. The use of radial distribution functions and atomic separation metrics is a common approach in identifying duplicate crystal structures[13–15].

Early work in computational crystal structure comparison by Gelato and Parthé[16, 17] culminated in the STRUCTURE TIDY routine[17]. Their approach identifies duplicate structures by comparing the Wyckoff positions of the atoms. While such an algorithm does provide an exact comparison of two crystal structures' atomic content, the dependence on algorithmic space group determination introduces difficulties for automated use.

The CRYCOM program from Dzyabchenko[18] combines concepts from Gelato and Parthé[16, 17] with a modified method of treating differences in lattice choice between the input structures. Applying ideas from Burzlaff and Rothammel's

paper[19] describing the use of transformative matrices to map one structure onto the other, CRYCOM generates many possible descriptions of the input structures and then searches for a match. However, CRYCOM also depends on the structures' space groups as input parameters.

The CMPZ algorithm by Hundt, Schön, and Jansen[20] is a robust exact comparison technique. By searching for an affine transformation that will map one structural description onto another, it provides a reliable and effective equality operator. However, CMPZ is not suitable for inclusion in XtalOpt, as the source code for the algorithm is not published and the licensing is not clear. CMPZ is implemented in the KPLOT application[21].

We were unable to locate an algorithm that satisfied both of the requirements stated earlier, and have consequently written and released XtalComp. This algorithm performs an exact test on the real-space atomic coordinates to determine the equality of two crystal descriptions, and is released under the "New" three-clause BSD license. XtalComp borrows the idea of a transform search from CMPZ, and is optimized for cases where both structures are known to have identical composition.

Before writing XtalComp, we identified five obstacles that must be overcome in order to perform a reliable comparison between two descriptions:

1. **Lattice choice** The most striking difficulty is that of translation vector choice. As shown in Figure 1 and Table 1, the description can change drastically from one triple of translation vectors to another. Complicating the situation, atomic positions are often reported and stored in fractional units, meaning that the coordinates use the translation vectors as a basis. Thus, the coordinate information is often dependent on the choice of translation vectors. Clearly, a solution to this problem is essential to the success of a comparison algorithm.

2. **Ambiguous origin** The origin of the unit cell with respect to the atomic structure is arbitrarily chosen. This is equivalent to stating that the atomic positions in a description may be uniformly translated by an arbitrary vector without upsetting the underlying structure.

3. **Numerical noise** The small error that results from floating point round-off, as well as the uncertainty that arises from the iterative methods used in periodic calculations, make exact comparison of coordinates and lattice parameters unfeasible.

4. **Boundary errors** The problem of numerical noise is exacerbated by the periodicity of the system. Even a small displacement of an atom near a unit cell boundary can cause the atom to appear to "move" from one side of the cell to another.

5. **Symmetry considerations** For our purposes, we consider enantiomorphic structures (those that are mirror images of each other) to be duplicates.

The solution to each of the above obstacles is detailed below. An overview of the algorithm is provided as a flowchart in Figure 2.

2. Algorithm Details

The XtalComp algorithm can be conceptually divided into four distinct operations: **preparation**, **screening**, **transform searching**, and **comparison**. Each is fully described below.

2.1. Preparation

The preparation stage initializes the library's internal data structures with the information passed in through the `XtalComp::compare` function. The two crystal structure descriptions are standardized as much as possible by using the following procedure:

1. **Niggli reduction** Simply put, the Niggli reduction procedure transforms the lattice vectors of an unit cell into their "most cubic" form. An important characteristic of Niggli-reduced cells in this application is that they are unique; no matter which representation (i.e. linear combination) of the translation vectors is used, the same Niggli-reduced cell will result from the reduction algorithm. The algorithm has been described fully in the literature[22–25].

2. **Standardize Orientation** Although the Niggli reduction algorithm described in Ref. [25] produces a unique translation unit, the reduced cell's orientation is still dependent on the input translation vectors. To remove this dependency, a simple rotation is performed on the cell matrix and atomic positions to place the structure in the conventional orientation (i.e. constrain $\vec{a}$ to lie along the x-axis and $\vec{b}$ in the xy-plane).

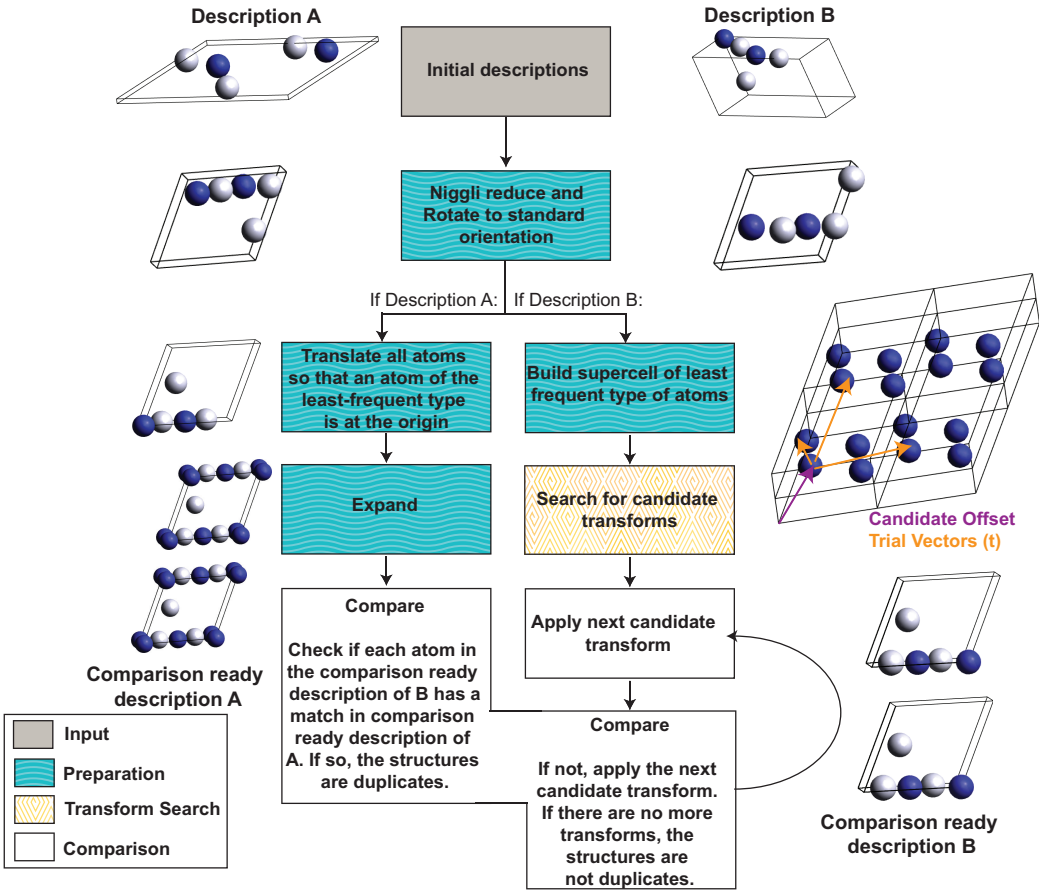

Figure 2: A visual overview of the XtalComp algorithm using a random M_2X_3 unit cell as an example. Both of the initial descriptions undergo a Niggli reduction and are rotated to standard orientation in the Preparation stage. The atoms in description $\mathbb{A}$ are translated such that an atom of type $\hat{Z}$ is at the origin (in this visualization, the origin is set at the near-bottom-left corner). All atoms near a unit cell boundary are replicated at the appropriate translationally degenerate positions, as shown in the "Expand" step. Description $\mathbb{A}$ is now ready for comparison.

A super-cell is constructed using atoms of type $\hat{Z}$ from description $\mathbb{B}$ during the Transform Search. The vectors separating the atoms in the super-cell are searched, attempting to locate the lattice vectors from $\mathbb{A}$. The basis formed by the trial vectors may be rotated or reflected relative to the basis of $\mathbb{A}$'s lattice vectors. If a satisfactory set of trial vectors is found, the transformation that maps the trial vectors to the reference vectors is calculated and stored.

During the comparison, each of these transformations is applied to description $\mathbb{B}$. If each atom in a transformed $\mathbb{B}$ has a matching atom in the translated and expanded $\mathbb{A}$, the underlying structures match. If none of the transformations produce a reasonable match, the descriptions represent unique structures.

3. Wrap atoms to cell As the Niggli reduction algorithm may change the dimensions of the lattice, some atoms may now lie outside of the unit cell boundaries. Translating the atoms by the appropriate linear combination of the new (Niggli-reduced and rotated) translation vectors corrects this problem.

At this point, the translation vectors will match (within numerical noise) if the two crystal descriptions have degenerate lattices. This concludes the preparative treatment of the lattice vectors – the remainder of the algorithm attempts to map the atoms of one lattice onto the other.

An arbitrary decision is made at this point as to which description will be the reference description ($\mathbb{A}$) and which will be the tested description ($\mathbb{B}$). Assuming the standardized lattices match (see Section 2.2), the lattice vectors of description $\mathbb{A}$ are cached for use during the transform search (Section 2.3).

There is little that can be done to standardize the atomic positions; the preparation procedure does, however, perform analysis and cache some results that are needed in later stages of the algorithm. The least frequently occurring atom type $\hat{Z}$ is identified and cached, along with the number of atoms of type $\hat{Z}$ in the descriptions. The atoms of $\mathbb{A}$ are uniformly translated so that an atom of type $\hat{Z}$ is at the coordinate origin.

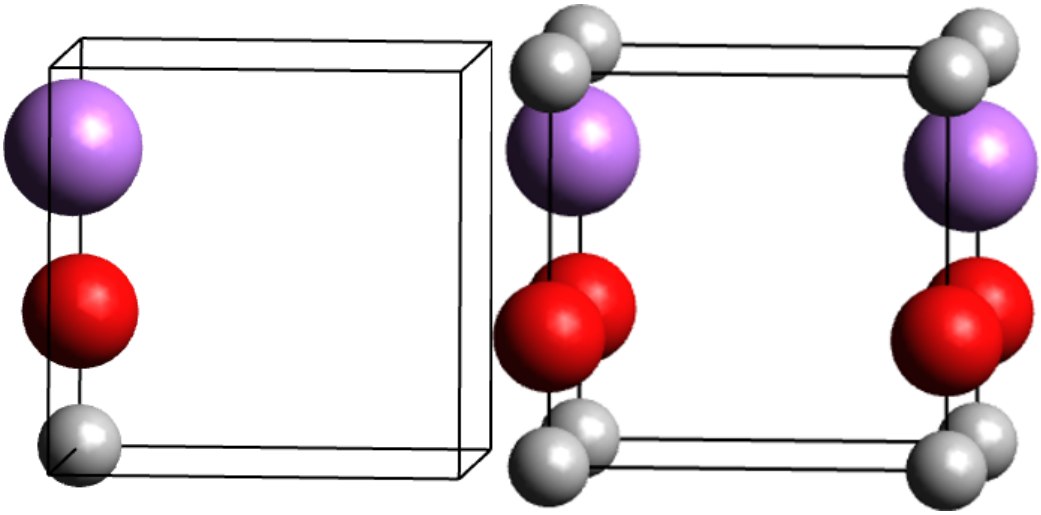

Figure 3: A unit cell before and after cell expansion. This figure shows the three possible expansion cases: the gray atom is near a corner, and is replicated to each of its eight translation-degenerate positions; the red atom is near an edge and is replicated to four positions; the purple atom is near a single face and is expanded to two positions.

A "cell-expansion" is performed on structure $\mathbb{A}$ to treat the issue of numeric noise at the unit cell boundaries. This is accomplished by checking the distance from each atom to the unit cell faces, and if it is close enough to the boundary (i.e. within the Cartesian tolerance specified in the `XtalComp::Compare` function), an image of the atom is placed at the translationally-degenerate position across the boundary. See Figure 3 for an example of the three expansion cases: face, edge, and corner. Such an expansion ensures that any atoms which have crossed the unit cell boundaries due to noise will be properly detected during comparison.

2.2. Screening

Before starting the costly transform search and comparison routines, several simple and fast comparisons are performed to quickly "weed-out" descriptions that have easy-to-detect differences. The following quantities are compared during screening:

- Total number of atoms
- Composition
- Unit cell volume
- Niggli-reduced lattice vector lengths
- Niggli-reduced lattice angles[1]

If description $\mathbb{B}$ differs significantly from description $\mathbb{A}$ in any of the above characteristics, the algorithm returns false to indicate that the descriptions do not represent the same crystal structure. Non-integer quantities are compared using a specified tolerance.

For efficiency, these screening checks are performed as soon as possible, rather than waiting until the preparation stage is complete. For example, the total number of atoms in each description is compared before the comparatively costly Niggli-reduction.

[1]Note that enantiomorphic structures may differ in their lattice angles. This problem is treated by replacing all angles γ greater than 90° with $\gamma' = \gamma - 2(\gamma - 90)$

2.3. Transform Search

The transform search identifies candidate affine transforms (combinations of translations, rotations, and reflections) that could potentially map the positions of $\mathbb{B}$'s atoms onto the atoms in $\mathbb{A}$. Two key pieces of information are sought during the search: a translation vector that maps the origin of $\mathbb{B}$'s lattice onto that of $\mathbb{A}$, and a rotation-reflection matrix that maps the translated atomic positions of $\mathbb{B}$ onto those of $\mathbb{A}$.

The search begins by constructing a super-cell of $\mathbb{B}$ that contains only atoms of type $\hat{Z}$. This restriction improves efficiency by minimizing the number of atoms considered during the transform search. If the length of the unit cell diagonal $|\vec{a} + \vec{b} + \vec{c}|$ is the same as the length of any of $\vec{a}$, $\vec{b}$, or $\vec{c}$, a $3 \times 3 \times 3$ super-cell is constructed, otherwise a $2 \times 2 \times 2$ super-cell is sufficient to contain all relevant atoms needed in the search.

Each atom in the super-cell is tested as a candidate origin by using the negative of its Cartesian position vector as the candidate translation (recall that $\mathbb{A}$ has been pre-translated to place an atom of type $\hat{Z}$ at the origin). The vectors between this atom and all others in the super-cell (referred to as "trial vectors", $\vec{t}$) are compared to the cached reference vectors of $\mathbb{A}$: $\vec{a}_{ref}$, $\vec{b}_{ref}$, and $\vec{c}_{ref}$. If the length of a trial vector matches the length of any of the reference vectors, the trial vector $\vec{t}$ is recorded. Thus, if $|\vec{t}| = |\vec{a}_{ref}|$, $\vec{t}$ is added to a list of candidate $\vec{a}$, and the same for $\vec{b}$ and $\vec{c}$.

The lists of candidate $\vec{a}$, $\vec{b}$, or $\vec{c}$ are iteratively tested to see if the angle (see footnote in Section 2.2) between two trial vectors matches the angle between the two corresponding reference vectors. If a set of trial vectors is found that match the reference vectors in terms of both lengths and angles, the candidate rotation/reflection matrix $[R]$ can be calculated as

$$[R] = [V][T]^{-1} \tag{1}$$

where $[R]$ is the 3×3 rotation/reflection matrix, $[V]$ is the column matrix formed by the reference vectors, and $[T]$ is the column matrix formed by the trial vectors. The above equation is simple to derive by considering that the transform operation can be described as a transformation of $[T]$ into $[V]$, that is, $[V] = [R][T]$.

The translation of the candidate origin and any corresponding rotation/reflection matrices are combined and stored as standard 4×4 transform matrices for later use in the comparison stage. The search continues until all atoms in the super-cell have been tested as an origin and used to search for trial vectors.

2.4. Comparison

The comparisons are performed by iteratively testing each candidate transform found during the transform search and checking the atomic positions of the transformed $\mathbb{B}$ (denoted here as $\mathbb{B}'$) against those of $\mathbb{A}$.

First, the current 4×4 transform $[X]$ is applied to $\mathbb{B}$ to create a working description $\mathbb{B}'$ that will be compared to $\mathbb{A}$. The linear portion of $[X]$ ($[X]_L$, the upper left 3×3 portion of the transform matrix) is used to generate the new cell matrix from $\mathbb{B}$'s cell matrix, $[\mathbb{B}'] = [X]_L[\mathbb{B}]$. The atomic positions are converted to homogeneous coordinates (x, y, z, w=1) and multiplied by $[X]$, then converted back into Cartesian vectors[2] by the standard conversion, $\left(\frac{x}{w}, \frac{y}{w}, \frac{z}{w}\right)$.

As a result of the transformation, $\mathbb{B}'$ will have the same unit cell dimensions as $\mathbb{A}$, but may lie in a different octant depending on the sign of the vectors produced during the transformation. To allow direct comparison between the atomic coordinates, all atoms are wrapped into one of the structures' unit cells. Our implementation iterates through each atom in $\mathbb{B}'$, translates it by some combination of lattice vectors into $\mathbb{A}$'s unit cell, and then searches for a matching atom in $\mathbb{A}$. If all atoms in $\mathbb{B}'$ find a match in $\mathbb{A}$, the structures are considered to be duplicates and the algorithm returns true. If not, the next transform is applied and the comparison repeats. Once all transforms have been exhausted, the algorithm returns false, indicating that the input descriptions do not describe the same structure.

3. Computational Details

XtalComp is written in native C++ and uses only STL containers and algorithms, ensuring that the algorithm will work on all platforms that provide a compliant C++ compilation environment. The library has no external dependencies and can be easily interfaced to work with existing code. The library has been built and tested using Linux/GCC 4.6.1 and Windows/MSVC 2008.

The library is intended to be statically linked into an application and contains a single entry point, `XtalComp::compare`. This function's arguments are the cell matrix, atom types, and atom positions of each structure, and optionally the Cartesian and angular tolerances. If the tolerances are not specified, the default tolerances of 0.05 Å and 0.25° are used. Equality is reported through a boolean return value. It will optionally return the 4×4 transformation matrix used to map

[2]Due to the nature of these transforms, the w coordinate is actually ignored in our implementation, as it will always be unity in these calculations.

$\mathbb{B}$ into the Niggli-reduced $\mathbb{A}$. This matrix may be analyzed to determine whether the input descriptions are enantiomorphs by checking for reflections in the linear portion. Non-default tolerances may be specified at run-time and all management of working memory is handled internally by the library.

Note that the default tolerances were obtained by inspecting the output of several structural relaxations using empirical potentials, and may need to be adjusted based on the precision used to generate the input descriptions (e.g. larger tolerances should be used if a calculation with loose convergence criteria generated the descriptions).

The library contains a C++ testing program that demonstrates how to use and call the comparison algorithm. The setup to use XtalComp is simple:

```
#include "xtalcomp.h"

...

// Declare input variables
XcMatrix cell1, cell2;
std::vector<XcVector> pos1, pos2;
std::vector<unsigned int> types1, types2;
double transform[16];

// Fill cell[1|2] with the column matrix of
//   the unit cell vectors
// Fill pos[1|2] with the fractional
//   coordinates of the atoms
// Fill types[1|2] with the atomic numbers
//   of the atoms

// Compare the descriptions
bool match = XtalComp::compare(
  cell1, types1, pos1, cell2, types2, pos2,
  transform);
```

The `XcMatrix` and `XcVector` objects are defined in the library to provide simple linear algebra functionality. `XcMatrix` is a 3×3 matrix, and `XcVector` is a column 3-vector. The comparison is made in the call to `XtalComp::compare`, and the result is stored in the `match` variable for later use; if `match==true`, the descriptions represent the same crystal structures. The `transform` argument will be

overwritten with a row-major array containing the successful 4×4 transformation matrix if the descriptions match.

A web interface to XtalComp is publicly accessible at `http://xtalopt.openmolecules.net/xtalcomp/xtalcomp.html`. The interface takes two structural descriptions as input, compares them using XtalComp, and displays the result. A sample set of inputs is provided online.

4. Results

4.1. Accuracy

Comparison Test

To test the reliability of the algorithm, the performance of XtalComp was compared to XtalOpt's existing "fingerprinting" equality test described earlier. In particular, the reliability of the fingerprint's space group component was of interest. A set of 512 known duplicate descriptions was created using a Niggli-reduced 16 formula unit super-cell of titanium dioxide in the rutile phase. The first set of 256 test descriptions were created by applying rotations and reflections to the original description and changing the lattice vector choice by applying a change-of-basis matrix to the translation vector matrix and atomic positions. An additional 256 "noisy" descriptions were created by applying atomic translations to each of the descriptions in the first set. Each atom in every new "noisy" description was translated by a uniform displacement vector and a random noise vector. The displacement vector was chosen at random once per description and identically applied to every atom, while the noise vector is of length 0.005 Å and randomly oriented. The random noise vector simulates numeric noise and is of reasonable length in the context of a computational geometry optimization. Thus, a test set of 512 known duplicate test descriptions is created, half identical to the reference descriptions save lattice choice, and the other half identical to the first half with the addition of atomic noise.

These test descriptions were compared to the reference using both XtalComp and automated spacegroup detection. XtalComp was performed using the default tolerances of 0.05 Å for spatial measurements and 0.25° for angles. Spacegroup matching is performed using the open-source C SPGLIB library[26] with a Cartesian tolerance of 0.05 Å.

The XtalComp algorithm correctly identified all 512 test descriptions as duplicates of the original, in spite of the simulated coordinate noise and lattice randomizations. Space-group detection performed admirably, correctly matching the reference with 470 of the 512 test descriptions. The 42 false negatives occurred in

the test subset that contained coordinate noise, and highlight the sensitivity to the tolerance setting in space group detection algorithms.

Stress Testing

An additional accuracy test randomly generates two unique descriptions of a random structure. The descriptions will vary in lattice shape and origin, and may be reflected or rotated. The two distinct descriptions are first compared using XtalComp to ensure the algorithm performs correctly and identifies the input representations as degenerate. Following this, a random atom from one description is randomly displaced by a distance much greater than the comparison tolerance such that the descriptions no longer share the same structure. The XtalComp comparison is carried out again to ensure that the algorithm correctly identifies that the descriptions refer to distinct structures. The initial random structure is generated in a manner ensuring that each of the 14 Bravais lattices will be sampled with a reasonable probability. This test has been run without failure several hundreds of thousands of times on random structures using up to 500 atoms per unit cell and up to five types of atoms. Atom types are randomly chosen from a uniform distribution containing 5 choices.

4.2. Performance and Scaling

The performance of the XtalComp algorithm has been benchmarked using a single core of a 2.00 GHz Intel T4200. Using the stress-test method described above on a unit cell with 100 atoms and 5 atom types, 2500 tests were performed. The comparison times for both the positive (i.e. same structure) and negative (i.e. distinct structures) comparisons were averaged, and the mean time of a positive test was measured to be 0.69 ± 0.03 milliseconds per comparison, while the negative comparisons took an average of 1.0 ± 0.4 milliseconds per comparison. The longer time for mismatched descriptions is expected, as the algorithm stops the comparison once a successful transform is found.

Additionally, the scaling performance of XtalComp was investigated by performing the same test as described above while varying the number of atoms. Unit cells with between 5-500 atoms were tested, using a maximum of 5 atomic species. The results of these tests are shown in Figure 4, and the timings are the average of 2500 unique comparisons.

For unit cells with fewer than 40 atoms, the difference between a positive and negative match is negligible. Above this threshold, the time needed to exhaust the candidate transforms becomes significant and we see the timings diverge.

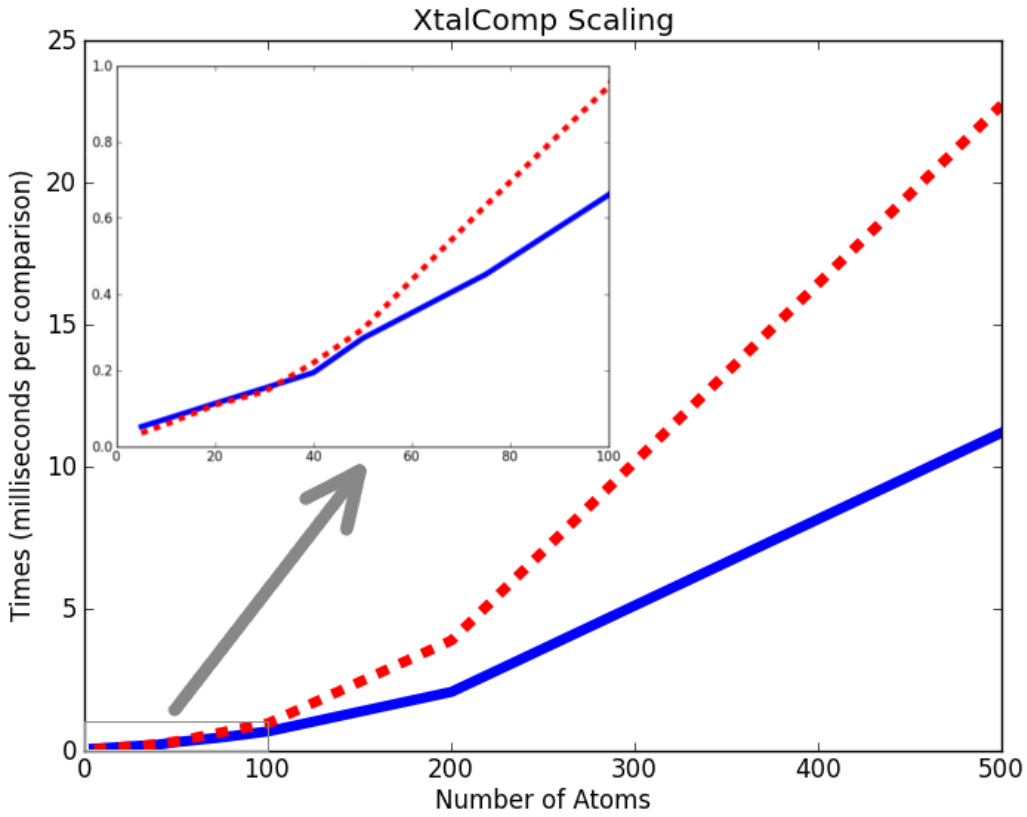

Figure 4: Change in comparison time as the unit cell size is increased. Average time of a positive match is shown in solid blue, and average time of an negative match is shown in dotted red. The inset shows a magnified view of the 0-100 atom range.

4.3. Evolutionary Algorithm Performance

The evolutionary algorithm benchmarking test described in Ref. [1] was carried out using XtalOpt to investigate how a search performed using the XtalComp duplicate matching scheme compares to one which employs a fingerprint composed of the structure's enthalpy, volume, and space group. The test system was a 16 formula unit super-cell of titatium dioxide, with the rutile phase as the target structure. The parameters used to guide the XtalOpt search are the defaults provided in Ref. [1], which also gives the pairwise potential used for energy calculations and structure relaxations. The fingerprinting test used an enthalpy tolerance of 0.002 eV, a volume tolerance of 0.5 $Å^3$, and a space group tolerance of 0.05 Å. The XtalComp test used the XtalComp algorithm for duplicate matching with the

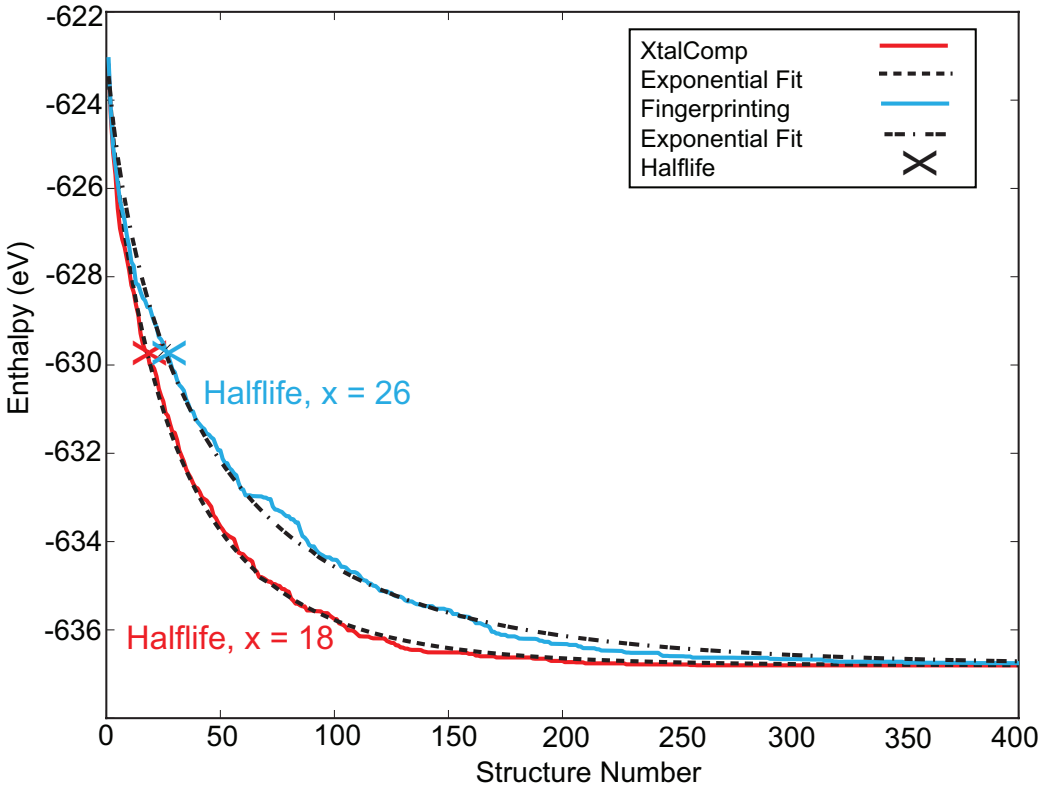

Figure 5: A "Hartke Plot"[1] showing the average enthalpy information plotted against structure index for benchmarks using the XtalComp and XtalOpt "fingerprinting[1]" duplicate matching methods. The dotted lines are fitted to the average data, and "×" denotes the half-life point of the exponential decay. A color version of this figure is available online.

default tolerances of 0.05 Å for lengths and 0.25° for angles.

Figure 5 shows the results of the benchmarks using the fingerprinting and XtalComp duplicate matching methods. Ref. [1] provides a full description of these plots, we will only summarize them here. The plots are of enthalpy vs. structure index, and show how quickly a given search approaches the target structure. The solid lines are is the average lowest-enthalpy structure for a given structure index, taken from a sample of 100 searches. The dotted lines are an exponential decay function fitted to the average structure data. The rate at which this fitted function converges towards its minimum indicates how well the search performs under a given set of conditions. Thus, the half-life of the exponential decay function is a useful quantification of search performance – a shorter half-life cor-

responds to better performance.

The fingerprinting method resulted in a half-life value of 26 structures, with 96 of the 100 searches finding the target rutile phase. The XtalComp method yielded a half-life of 18 structures, with all 100 searches successfully locating the target structure in less than 280 iterations. The plotted difference is most noticeable at the tail of the decay; the XtalComp test appears reasonably well converged around 200 structures, while the fingerprint test takes over 300 structures to reach a similar convergence. XtalComp identified 1.7% of all structures produced in the searches as duplicates, while fingerprinting identified 3.0% as duplicates.

Interestingly, a third benchmark (not shown) using no duplicate matching whatsoever also outperformed fingerprinting, scoring a half-life of 18 structures with a 100% success rate. All searches located the global minimum by the 325^{th} structure, with the decay function well-converged by structure 250. There is a lesson in this result: an unreliable duplicate matching scheme is worse than none at all, and that benchmarking and testing these techniques is essential to providing the best performance possible.

While testing the effect on the evolutionary algorithm, occasional false negatives were seen. Inspection of these representations show that noise in the unit cell vectors caused the descriptions to reduce to different Niggli cells. Using a stricter convergence criterion during the structure relaxations reduces the occurrence of these unidentified duplicates.

5. Conclusion

The reliability of the XtalComp algorithm has been demonstrated by testing a set of known duplicate descriptions, as well by stress-testing using randomly generated structures. The algorithm has excelled in both cases, correctly identifying the 512 descriptions in the test sample as duplicates and by performing $> 10^5$ random tests without error. The random tests include both a positive and negative test to ensure that the algorithm not only correctly detects duplicates, but also correctly identifies distinct structures. The random test is biased to generate structures from all 14 Bravais groups with reasonable probability, to ensure sufficient sampling of common lattice types. All tests have been designed to produce structures that include rotations and reflections of the crystal structure relative to the coordinate origin.

The performance is suitable for most applications, with an average run time of approximately one millisecond for a 100 atom unit cell comparison on a 2.00 GHz processor. We point out that this timescale is orders of magnitude smaller

than that needed to perform even a classical structure relaxation on a similarly sized system, and thus XtalComp can be used in structure prediction software without fear of a performance bottleneck.

The algorithm has been tested in the XtalOpt evolutionary crystal structure prediction software, serving as the duplicate structure removal technique. It outperformed the previous XtalOpt fingerprinting method in both efficiency and reliability. We show that a badly designed duplicate matching scheme (i.e. one that is prone to false positives or negatives) can actually decrease a search's performance compared to an implementation with no duplicate matching.

Acknowledgments

The authors thank Dr. Richard Hennig and Will Tipton at Cornell University for several fruitful discussions relating to duplicate structure identification. Dr. James Hooper's comments during the writing of this article are greatly appreciated. We acknowledge the NSF (DMR-1005413) for financial support and the Center for Computational Research (CCR) at SUNY Buffalo for computational support.

[1] D. C. Lonie, E. Zurek, Comput. Phys. Commun. 182 (2011) 372.

[2] D. C. Lonie, E. Zurek, Comput. Phys. Commun. 182 (2011) 2305.

[3] XtalOpt website, http://xtalopt.openmolecules.net.

[4] D. Deaven, K. Ho, Phys. Rev. Lett. 75 (1995) 288.

[5] R. L. Johnston, Dalton T. (2003) 4193.

[6] G. Trimarchi, A. Zunger, J. Phys: Condens. Mat. 20 (2008) 295212.

[7] S. Woodley, C. Catlow, Comp. Mater. Sci. 45 (2009) 84.

[8] Z. H. Li, A. W. Jasper, D. G. Truhlar, J. Am. Chem. Soc. 129 (2007) 14899.

[9] M. Valle, A. R. Oganov, 2008 IEEE Symp. on Vis. Analy. Sci. Tech. 11 2008.

[10] A. R. Oganov, M. Valle, J. Chem. Phys. 130 (2009) 104504.

[11] M. Valle, A. R. Oganov, Acta Crystallogr. A 66 (2010) 507.

[12] A. O. Lyakhov, A. R. Oganov, M. Valle, Comput. Phys. Commun. 181 (2010) 1623.

[13] E. L. Willighagen, R. Wehrens, P. Verwer, R. de Gelder, L. M. C. Buydens, Acta Crystallogr. B 61 (2005) 29.

[14] J. A. Chisholm, S. Motherwell, J. Appl. Crystallogr. 38 (2005) 228.

[15] N. Abraham, M. Probert, Phys. Rev. B 77 (2008) 134117.

[16] E. Parthé, L. M. Gelato, Acta Crystallogr. A 40 (1984) 169.

[17] L. M. Gelato, E. Parthé, J. Appl. Crystallogr. 20 (1987) 139.

[18] A. V. Dzyabchenko, Acta Crystallogr. B 50 (1994) 414.

[19] H. Burzlaff, W. Rothammel, Acta Crystallogr. A 48 (1992) 483.

[20] R. Hundt, J. C. Schön, M. Jansen, J. Appl. Crystallogr. 39 (2006) 6.

[21] KPLOT website, http://www.crystalimpact.de/download/kplot.htm.

[22] P. Niggli, Krystallographische und strukturtheoretische Grundbegriffe: Handbuch der Experimentalphysik, Vol. 7, Part 1,, Akademische Verlagsgesellschaft, Leipzig, 1928.

[23] B. Gruber, Acta Crystallogr. A 29 (1973) 433.

[24] I. Křivý, B. Gruber, Acta Crystallogr. A 32 (1976) 297.

[25] R. W. Grosse-Kunstleve, N. K. Sauter, P. D. Adams, Acta Crystallogr. A 60 (2003) 1.

[26] Spglib website, http://spglib.sourceforge.net/.