

Polynator: a tool to identify and quantitatively evaluate polyhedra and other shapes in crystal structures

Lukas Link and Rainer Niewa*

Institute of Inorganic Chemistry, University of Stuttgart, Pfaffenwaldring 55, 70569 Stuttgart, Germany. *Correspondence e-mail: rainer.niewa@iac.uni-stuttgart.de

Received 8 August 2023

Accepted 26 September 2023

Edited by J. Ilavsky, Argonne National Laboratory, USA

Keywords: distortion; shape; polyhedra; continuous symmetry.**Supporting information:** this article has supporting information at journals.iucr.org/j

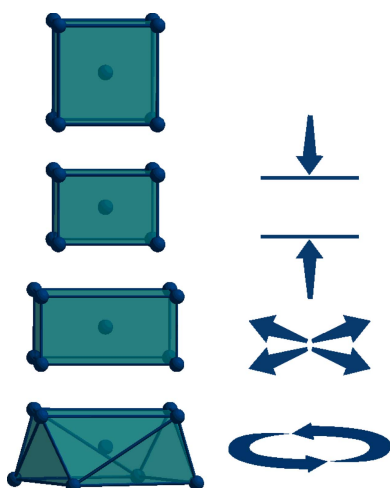
Polynator is a Python program capable of identifying coordination polyhedra, molecules and other shapes in crystal structures and evaluating their distortions. Distortions are quantified by fitting the vertices of a model to a selected set of atoms. In contrast to earlier programs, models can be deformable, which allows them to represent a point group or a range of shapes such as the set of all trigonal prisms, rather than a specific, rigid shape such as the equilateral trigonal prism. The program comes with a graphical user interface and is freely available. This paper discusses its working principle and illustrates a number of applications.

1. Introduction

In chemistry and crystallography, coordination environments are regularly described in terms of polyhedra. These polyhedra are crucial in understanding and analyzing crystal structures and their relations to material properties. Owing to electronic effects or size mismatches, they are frequently distorted to various degrees from their ideal configuration. With some coordination environments it can even be challenging to find the closest-matching well defined polyhedron in the first place. Moreover, the extent of the distortion is notoriously difficult to quantify. In many cases, it can be desirable to know the degree of distortion relative to the closest-matching shape which obeys a given set of constraints, *e.g.* a specific point-group symmetry. However, although various authors have proposed formulae, algorithms and programs, there is not yet a universal tool to quantify these distortions. The idea of quantifying deviations from models with an ideal reference shape was explored in an early publication by Dollase (1974). His purely geometrical method pairs the atoms in a given coordination environment with the vertices of a model polyhedron. After the model has been iteratively rotated and scaled to minimize the squares of the distances between atoms and the corresponding model vertices, the distortion can be quantified on the basis of the residual deviation. Using a similar approach, Zabrodsky *et al.* (1992) defined the continuous symmetry measure (CSM), which Pinsky & Avnir (1998) later followed up with the continuous shape measure (CShM). Both rely on the scale-independent metric S for the deviation of an arrangement of atoms from a reference model, defined as

$$S = 100 \left(\frac{\sum_i |\mathbf{a}_i - \mathbf{v}_i|^2}{\sum_i |\mathbf{a}_i - \mathbf{c}|^2} \right), \quad (1)$$

where \mathbf{a}_i is a vector with the Cartesian coordinates of an atom i , \mathbf{v}_i is the vector of its corresponding model vertex and \mathbf{c} is the



centroid of all atom vectors **a**. Provided an optimal orientation and size of the model and an optimal pairing of atom and model vertices, *S* may assume values from 0 to 100, where 0 signifies perfect agreement between atoms and model vertices. Although both CSM and CShM are based on the same definition of *S*, the difference lies in the way the reference model is obtained. CShM refers to scalable but otherwise rigid predefined models. CSM on the other hand refers to the closest-matching model that conforms to a given point-group symmetry. In the cases of the regular polygons, the Platonic solids, the Archimedean cuboctahedron and the Archimedean icosidodecahedron, CSM and CShM are equivalent. In any other case, CSM allows for more degrees of freedom and will thus generally produce lower values. Continuous chirality measures (CCMs) and symmetry operation measures (SOMs) belong to the same family (Alvarez *et al.*, 2005; Echeverría & Alvarez, 2008). They refer to the closest-matching models which are achiral (CCM) or conform to a single symmetry element (SOM).

The computer program *shape* was first released in 2003 to allow users to obtain continuous shape measures of coordination environments found in chemical compounds (Llunell *et al.*, 2013). *ChemEnv*, a program that can identify many polyhedra with fixed proportions by comparing their continuous shape measures, is also based on this approach (Waroquiers *et al.*, 2020). A great advantage of *ChemEnv* compared with *shape* is that it accepts *.cif* files as an input and finds coordination environments with minimal user intervention, based on the central atoms. The *CoSyM* (*Continuous Symmetry Measures*) program (website <http://csm.ouproj.org.il> maintained by Tuvi-Arad and Avnir) is capable of calculating SOMs and CCMs (as well as CShMs for a small set of models). By comparing various different SOMs, it enables the user to make approximate statements about the full point-group symmetry of a given atom arrangement (Echeverría & Alvarez, 2008).

In a different vein, a symmetry-independent method that quantifies the distortion according to the sphericity of an ellipsoid fitted to the coordination environment was proposed by Cumby & Attfield (2017). Yet another approach was taken with the *PolyDis* software, which allows the user to fit models of the most frequently encountered polyhedra to their data based on minimal average distances (Stoiber & Niewa, 2019).

Here we present *Polynator*, a Python program based on a similar method to that of the CSM family of metrics. However, in contrast to those, it overcomes the limitations of separate shape and symmetry measures by making use of dynamic models. It takes *.cif* files as input and, with minimal user input, finds coordination environments, molecules and other arrangements of atoms within the crystal structure. The program may be used independently or most comfortably via its own graphical user interface (GUI). In addition to detailed output files and tables, new *.cif* files with only the coordinates of real atoms and model vertices can be generated to allow for easy visualization in external programs. *Polynator* is equally well suited for the analysis of individual coordination environments, molecules or cages and for screening large

Table 1
Parameter types defined in *Polynator* with exemplary uses.

Parameter type	Exemplary models
Scale (<i>sc</i>)	Platonic and Archimedean solids, Johnson polyhedra, hexacapped cube
Height (<i>h</i>)	Prisms, pyramids, the majority of models
Width (<i>w</i>)	Prisms, pyramids, the majority of models
Torsion angle (φ)	Twisted prisms, biphenyl skeleton
Modulated height (\tilde{h})	Dynamic triangular dodecahedron, 18-crown-6 skeleton
Modulated width (\tilde{w})	Rhombus, 18-crown-6 skeleton
Modulated torsion angle ($\tilde{\varphi}$)	Rectangle, cuboid, anticuboid, 18-crown-6 skeleton
Composite parameters	Pyritohedron, elpasolite cuboctahedron

databases to map trends within structural families. It is freely available via download from <https://www.iac.uni-stuttgart.de/en/research/akniewa/> as a Python script or as a compiled version for simple use with Microsoft Windows.

2. Working principle

Most of the earlier programs are limited to polyhedra of fixed proportions, whereas *Polynator* is built to fit dynamic models with any number of free parameters. In contrast to the rigid models employed by *shape* and *ChemEnv*, dynamic models can not only be rotated and scaled but also deformed in a variety of ways specifically defined for each model (*e.g.* stretching, twisting, puckering *etc.*). Dynamic models can thus adequately represent the general definition of polyhedra such as prisms and pyramids with a variable height-to-width ratio, effectively providing point-group symmetry analysis for many atom arrangements of small and medium size.

For example, Fig. 1 shows a number of model polyhedra which are derived from the cube by adding one or more degrees of freedom. All of these are included in *Polynator*, along with a wide array of other predefined models of up to 60 vertices, currently 211 in total. This includes, among others, all Platonic and the nine Archimedean solids, 28 Johnson polyhedra, the canonical Frank–Kasper polyhedra and three of their dual fullerenes, various prisms, antiprisms, pyramids and bipyramids, and models for common organic building blocks, as well as dynamic models to account for distorted versions of the polyhedra most frequently encountered in inorganic crystal structures. The full list is included in the *Polynator* manual provided in the supporting information. To cover any unforeseen needs, the user can define additional custom models using the GUI.

Every model is granted three translational and two rotational degrees of freedom. In addition, various types of free parameters (see Table 1) determine its shape. A simple example is the scaling parameter, which preserves the proportions of the model and is utilized primarily for rigid models such as Platonic and Archimedean solids. There are also independent width and height parameters, which lend themselves to prisms, antiprisms, pyramids *etc.* Parameters for torsion angles help to fit twisted prisms and the like. Height, width and torsion angles can also be modulated via sinusoidal

functions. This allows, for example, a square to morph into a disphenoid, rhombus or rectangle. Lastly, functions binding multiple parameters to fewer variables can be defined in order to afford polyhedra such as the pyritohedron or the tetrahedrally distorted elpasolite-type cuboctahedron. For details on the fitting process, see the manual. Since quite a few models are neither rigid nor constrained by symmetry alone, it is not always clear how to classify the distortion values within the CSM framework. Some models contain rigid parts but are not altogether rigid, such as the biphenyl skeleton. Others have other extrasymmetrical constraints, *e.g.* preserving the planarity of their faces, such as the rhombohedron, rhombic prism, pyritohedron and others. Instead of adding a member to the already numerous CSM family (Pinsky & Avnir, 1998; Alvarez *et al.*, 2005; Echeverría & Alvarez, 2008), we propose

a new metric, which measures the distortion relative to a specified model. This is the distortion value δ , defined as

$$\delta = 100 \left(\frac{\sum_i |\mathbf{a}_i - \mathbf{v}_i|^2}{\sum_i |\mathbf{a}_i - \mathbf{c}|^2} \right)^{1/2}, \quad (2)$$

where \mathbf{a}_i is an atom vector, \mathbf{v}_i is the vector of its corresponding model vertex and \mathbf{c} is the centroid of all atom vectors. While this distortion metric is closely related to the S parameter of the CSM family via $S = 0.01\delta^2$ or $\delta = 10\sqrt{S}$, it scales approximately linearly instead of quadratically with small deviations from the reference model, as illustrated in Fig. 2. This leads to a more intuitive and arguably less skewed understanding of the extent of distortion. It also yields more reasonable values for small distortions: where CSM-type

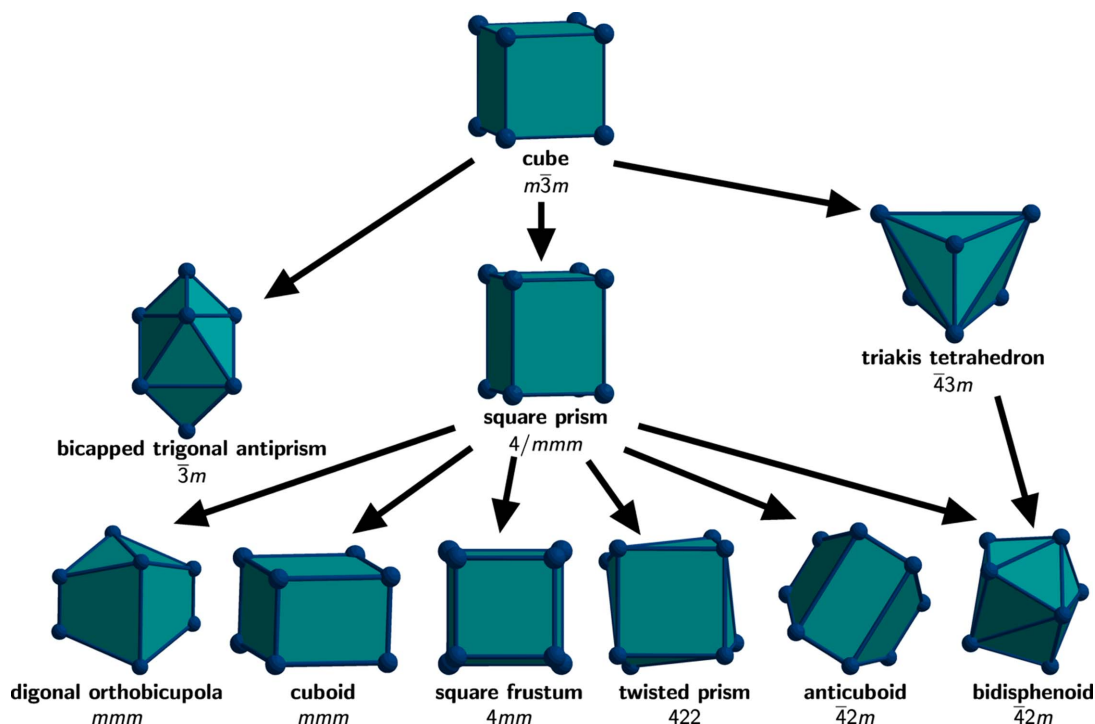


Figure 1 Partial tree diagram of model polyhedra derived from the cube by symmetry reduction. Higher positions in the diagram indicate higher symmetry. For simplicity, the bicapped trigonal antifrustum (3) and the bicapped trigonal twisted prism (32) have been omitted.

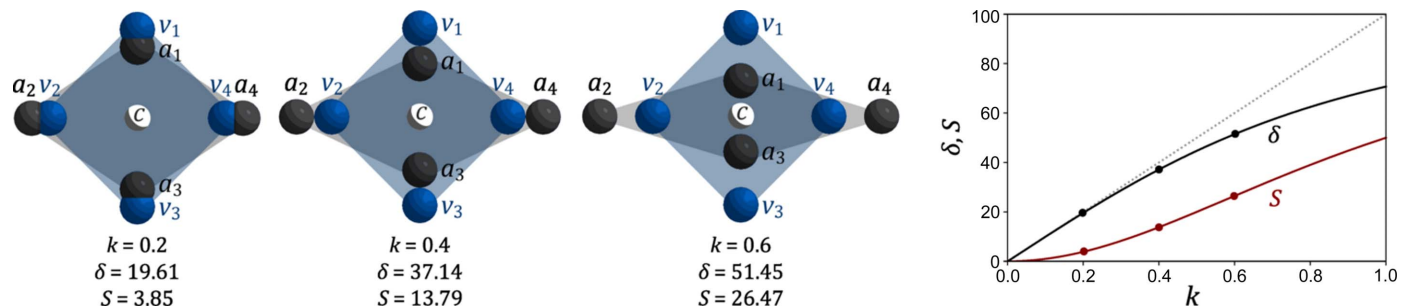


Figure 2 Progressive increasing rhombic distortion of a quadrilateral arrangement of atoms a (dark gray) measured against the vertices v of a square model (blue). The rhombus is contracted along the a_1a_3 diagonal by a factor k and stretched along the a_2a_4 diagonal by the same amount. Values of the distortion metrics δ and S (CSM) are plotted in the diagram on the right, with the examples given on the left marked with dots. Note: these are very strong distortions. Common use cases for a program such as *Polynator* involve distortion values $\delta < 30$. In this range, the δ curve is approximately linear, and the S curve is approximately quadratic.

values often require three or more decimal places to discern a small distortion from a perfect fit, two decimal places will basically always suffice for δ . Nonetheless, *Polynator* can be configured to yield CSM-type values instead of δ values and the detailed output files it generates always contain both values.

2.1. Process

For each *.cif* file provided as an input by the user, *Polynator* traverses the process outlined in Fig. 3. We will proceed by briefly explaining each step.

Step 1. The input *.cif* file is read and the relevant atom vectors are generated according to symmetry instructions.

Step 2. Atom arrangements to be fitted can be classified into coordination environments and molecules. A coordination environment may be generated around a selected central atom (or dummy atom coordinates) following one of several methods. The most basic method is to include all atoms whose distance from the central atom is within a user-specified range. Alternatively, the user may assign atomic radii to each atom. It is furthermore possible to specify a maximal coordination number; excess ligands with the highest distance from the central atom are excluded first. Coordination environments can also be constructed by a simple algorithm, finding gaps in consecutive center–ligand distances. Last but not least, an algorithm based on the Voronoi polyhedron around the central atom is available to the user. It considers an atom a ligand if the solid angle subtended by its associated Voronoi face from the perspective of the central atom is greater than a given threshold (20° by default). This last algorithm is loosely based on the definition by O’Keeffe (1979) for the coordination number and very similar to the method used by *ChemEnv* (Waroquiers *et al.*, 2020). Since there is no single unbiased method to determine which atoms should be counted as ligands, *Polynator* leaves the choice to the user.

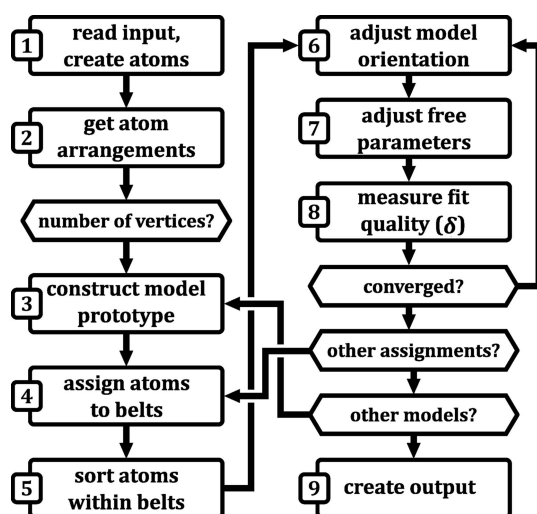


Figure 3
Simplified flow chart of the working principle. Multiple crystal structures and multiple atom arrangements per crystal structure can be processed in one run, not reflected here for simplicity.

Table 2

Widths, heights, torsion angles and free parameters for the construction of an anticuboctahedron.

Belt	<i>n</i> vertices	Width coefficient	Height coefficient	Torsion angle	Free parameters
1	3	$\sqrt{1/3}$	$\sqrt{2/3}$	0	sc_1
2	6	1	0	$\pi/6$	sc_1
3	3	$\sqrt{1/3}$	$-\sqrt{2/3}$	0	sc_1

Molecules are constructed on the basis of their connectivity according to user-specified or default atomic radii. Subsections (*e.g.* ethane units, icosahedra, porphyrin rings, fullerenes) can automatically be extracted from molecules or infinite frameworks by a number of predefined or user-specified topological strategies. This does not require any pre-defined bonds, just a regular *.cif* file and the name of the desired topological shape.

Step 3. A model prototype with the appropriate number of vertices for a given atom arrangement is constructed. Model vertices are stored not as Cartesian coordinates but as a compressed set of parameters directing their construction and manipulation. These parameters do not affect single vertices, but entire symmetry-aligned groups of them, referred to as belts. These belts are orthogonal to and are stacked along a model axis. For example, as illustrated in Fig. 4, an anticuboctahedron (or rigid triangular orthobicupola) is split into three belts. The information associated with each belt is listed in Table 2, namely the number of vertices it contains, the basic

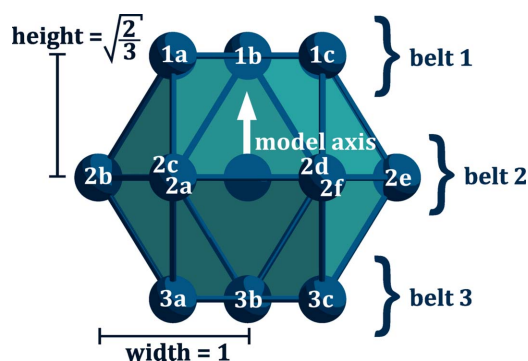


Figure 4
Anticuboctahedron (also known as a trigonal orthobicupola), split into belts along its 6 axis, which is used as the model axis. The individual model vertices within a belt are labeled a–f.

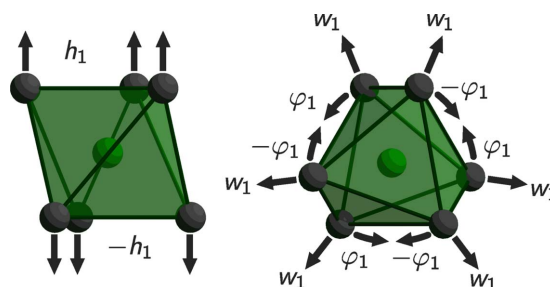


Figure 5
Twisted trigonal prism with three free parameters, namely h_1 for its height, w_1 for its width and φ_1 for the torsion angle between the bases. In general, a model may have any number of parameters.

cylindrical coordinates for the entire belt (width, height, torsion angle) and any free parameters (in this case only the same scaling parameter for each belt). The individual vertices are constructed by rotating them around the model axis by multiples of $2\pi/n$, where n is the number of vertices in that belt. Fig. 5 illustrates the deformations allowed by the free parameters of a dynamic twisted trigonal prism.

Step 4. Identifying the optimal pairing scheme of model vertices with their real counterparts found in the crystal structure is sometimes easy to do for humans but deceptively difficult for computer programs. The brute force approach of checking every possible permutation of model–real vertex pairs is the only way to guarantee the optimal result. However, the number of required evaluations grows factorially with the number of vertices, quickly rendering this approach infeasible for all but the lowest coordination numbers. As explained by Waroquiers *et al.* (2020), the number of evaluations can be reduced slightly by taking into account the symmetry of the model. Unfortunately, this does not ultimately solve the problem of unmanageable numbers of vertex permutations. The *CoSyM* program takes advantage of the topology of the vertex figure (Alon & Tuvi-Arad, 2018) to achieve a significant reduction in the number of permutations. Although this approach is very well suited for molecular structures due to their rather predictable bond lengths and comparatively low connectivities, it can run into problems when dealing with more irregular coordination polyhedra. Instead, depending on the model, *Polynator* chooses one out of several strategies to find the best fit without checking many combinations. When fitting a coordination polyhedron or cage, it is assumed that the real ligands are spread out more or less equally on a spherical shell around their centroid. The atom arrangement is sequentially scanned along the directions of 92 vectors pointing from the centroid to equally spread out locations on a unit sphere surface. It is then estimated how well the scanned atom arrangement agrees with the model for each of these directions. To do this, for each scanning step, the ligand vectors are first filled into the model belts according to their dot product with the scanning direction vector. The approximate model axis is then constructed as the eigenvector with the largest eigenvalue of a covariance matrix formed from the centroids of all belts. The fit quality is estimated according to the distance of each belt centroid from the model axis and the difference in dot product with the model axis between each vertex and the respective belt centroid. A small number of assignment schemes with the best estimated agreement qualify for full evaluation.

When handling strongly prolate models, such as the biphenyl skeleton, a very simple strategy of finding the belt assignment scheme is employed instead. The model axis, or the ‘longest’ axis of the atom arrangement, is constructed as the eigenvector with the largest eigenvalue of a covariance matrix formed from all atom vectors. The vectors are ranked according to their dot product with that axis and filled into the belts accordingly. For multi-belt oblate models, such as the porphyrin skeleton, the atom vectors are simply ranked by their distance from the centroid. The performance and accu-

racy of the assignments is further improved by the possibility for models to inherit their assignment from a less general version of themselves, provided that version is a good fit. For example, a trigonal antiprism may effectively copy the assignment of a previously fitted regular hexagon or octahedron. However, when copying from an octahedron, four different orientations must still be fitted. Though none of these strategies are technically guaranteed to find the best assignment scheme, they appear to be very reliable as long as the atom arrangement resembles the model to some degree.

Step 5. After the atom vectors have been assigned to belts, they have to be paired with the model vertices within their belt. This is done by rotating the model so that one of its vertices has the same torsion angle around the model axis as one of the atom vectors in the same belt. This torsion angle is set to 0° and the clockwise angles of the other vectors are measured with respect to it. The vectors within each belt are then sorted according to their angle and paired up according to this order. The agreement between model and real atoms is estimated according to the individual angular deviations between paired vectors, weighted by the distance from the model axis. The final pairing scheme is determined by repeating this process for each atom vector and selecting for the best agreement.

Step 6. Once the vectors are paired up, the model orientation can be adjusted to optimally fit the real coordination environment. This task is carried out by a Kabsch algorithm (Kabsch, 1976).

Step 7. Subsequently, the values of free shape parameters defined individually for each model are determined via least-squares fits. In the majority of cases, this can be done analytically by simply measuring averages of vector components (height, width, magnitude). Otherwise, the optimal value must be determined iteratively. Since the optimal solutions for shape and orientation are generally interdependent, the sequence of steps 6 and 7 has to be executed several times until a convergence criterion is reached (the δ value ceases to improve by more than 10^{-5} between cycles).

3. Examples and applications

3.1. Shape recognition and classification

Since *Polynator* is capable of automatically recognizing polyhedra and quantifying their similarity to a wide range of models, a very basic use case is the identification of ‘exotic’ polyhedra, such as the elongated gyrobifastigium. This is found as a coordination polyhedron of the Th site in the α -ThSi₂ structure type, but has not been identified as such in the publications we are aware of. This elongated gyrobifastigium is distinct from the regular gyrobifastigium, which can be found in the same structure type (Alvarez, 2017), as depicted in Fig. 6. The high-pressure CaFeTi₂O₆ phase, as described by Leinenweber & Parise (1995), contains both the regular gyrobifastigium and another somewhat uncommon Johnson polyhedron, the sphenocorona. *Polynator* also natively includes over 40 model polyhedra with more than 12 vertices,

many of which are represented in intermetallic phases (Zintl & Haucke, 1938, 1940; Kasper *et al.*, 1965; Ban & Sikirica, 1965), zeolites (Broussard & Shoemaker, 1960) or boron clusters (Popov *et al.*, 2015).

Another simple application is finding the best description of a given ‘irregular’ coordination polyhedron. For example, in the scheelite-type Bi[VO₄] compound (Mariathasan *et al.*, 1986), bismuth is coordinated by eight oxygen atoms in an arrangement that could be understood as a distorted version of either a triangular dodecahedron, a bicapped trigonal prism

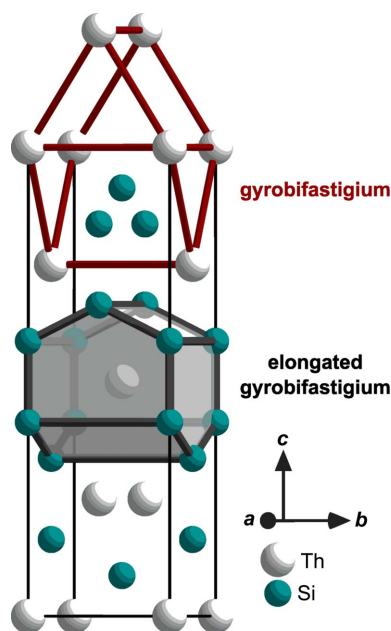


Figure 6 Both the regular gyrobifastigium and the elongated gyrobifastigium are found in the crystal structure of the ThSi₂ type.

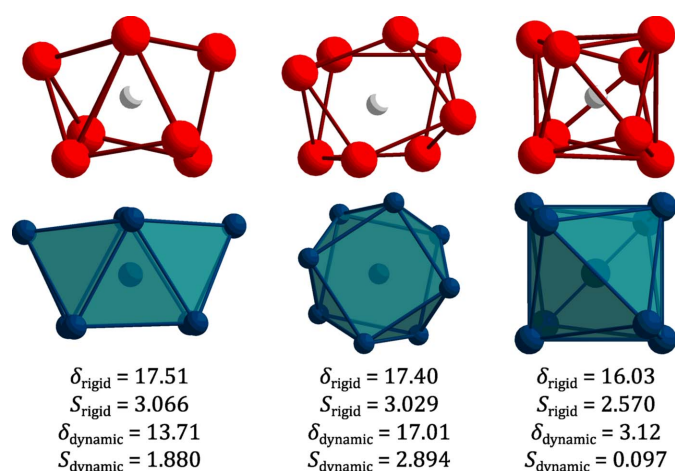


Figure 7 Coordination environment of bismuth in Bi[VO₄], viewed from three different perspectives, corresponding to the optimal orientation of an equilateral bicapped trigonal prism (left), an equilateral tetragonal antiprism (center) and an equilateral triangular dodecahedron (right). Distortion values quantify the fit of the aforementioned rigid equilateral models, as well as dynamic models which share the basic structure of the rigid models, but are otherwise only constrained by the respective point group.

or a tetragonal antiprism. As illustrated in Fig. 7, when comparing the fit for rigid versions of all three models, it becomes clear that the triangular dodecahedron is the best description, although not by a very large margin. These differences become more apparent when comparing dynamic versions of these models, as the distortion of the optimal triangular dodecahedron amounts to less than a quarter of the distortion of the optimal versions of the other candidates. This comparison between dynamic models is also arguably more in line with the intuitive understanding of a given polyhedron. For example, most coordination polyhedra commonly identified as triangular dodecahedra are not as prolate as the equilateral version, but more or less spherical.

3.2. Measuring characteristic parameters

Polynator can be used as a way to obtain exact values of parameters which can otherwise be somewhat challenging to

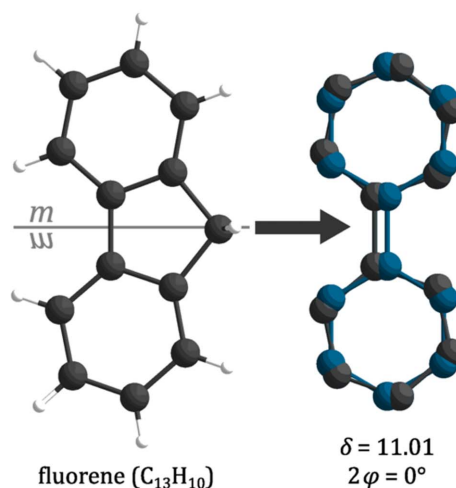
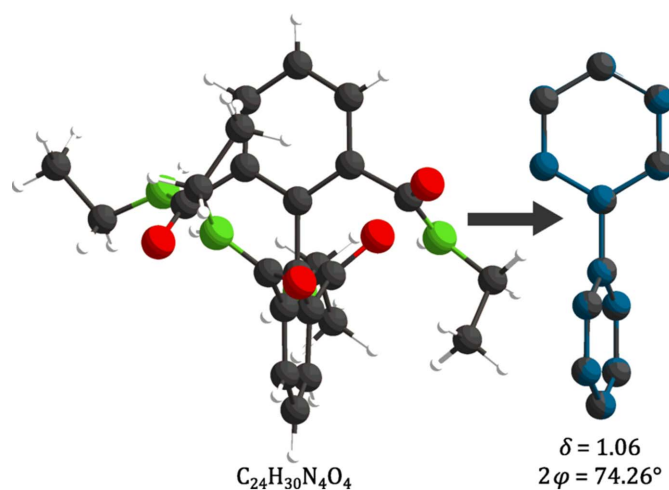


Figure 8 Overlay of real molecule (black) and fitted model (blue) for biphenyl skeletons cut from two different molecules. The C₂₄H₃₀N₄O₄ molecule (top) (Holý *et al.*, 2001) has somewhat bulky substituents on all *ortho* positions, hence the rather large dihedral angle 2φ . The biphenyl skeleton (bottom) is part of a fluorene molecule (Burns & Iball, 1955) with point group *m*. Though the latter molecule is not perfectly flat, its 2φ value is still exactly 0°. The considerable distortion is due to the skewed phenyl rings.

measure correctly. A good example is the dihedral angle between the phenyl rings in a biphenyl unit, as pictured in Fig. 8. Though it is easy to get an approximate value by

calculating the dihedral angle between an atom in the first ring and an atom in the second ring, using the bridging atoms as the hinge axis, this approach falls short if more precision is desired. *Polynator* provides a clean value for this angle as a by-product when finding the closest-matching ideal molecule. The model for a biphenyl skeleton has three free parameters: one for the radius of the phenyl rings, one for the length of the bridging bond and one for their dihedral angle. These values are accessible in the output files.

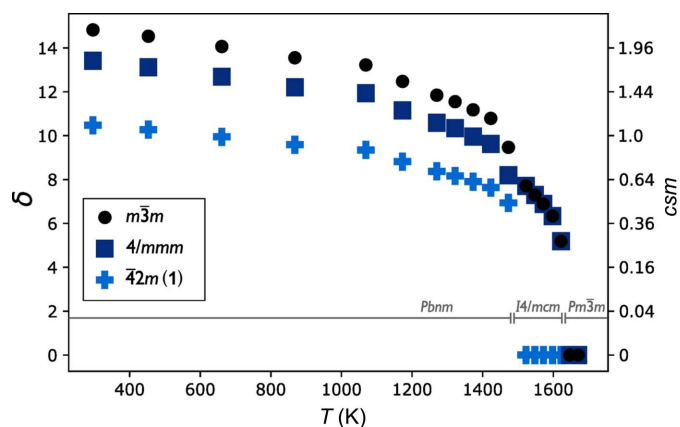


Figure 9 Temperature-dependent distortions of CaO_{12} cuboctahedra in $\text{Ca}(\text{TiO}_3)$ with respect to three model polyhedra which conform to different point groups. Experimental data according to a single sample powder X-ray diffraction measurement series (Yashima & Ali, 2009).

3.3. Quantification of distortions

Polynator is suited just as well for more intricate investigations, such as uncovering trends and patterns in the coordination behavior of a given structure type. Since any number of input files, atom arrangements and models can be processed at once and the process is quite fast (typically 10–20 models per second), large numbers of similar structures can be compared almost effortlessly. This is also facilitated by `.csv` output files listing important results in table form and the topological package allowing for the automatic extraction of specific structural units from large molecules or frameworks.

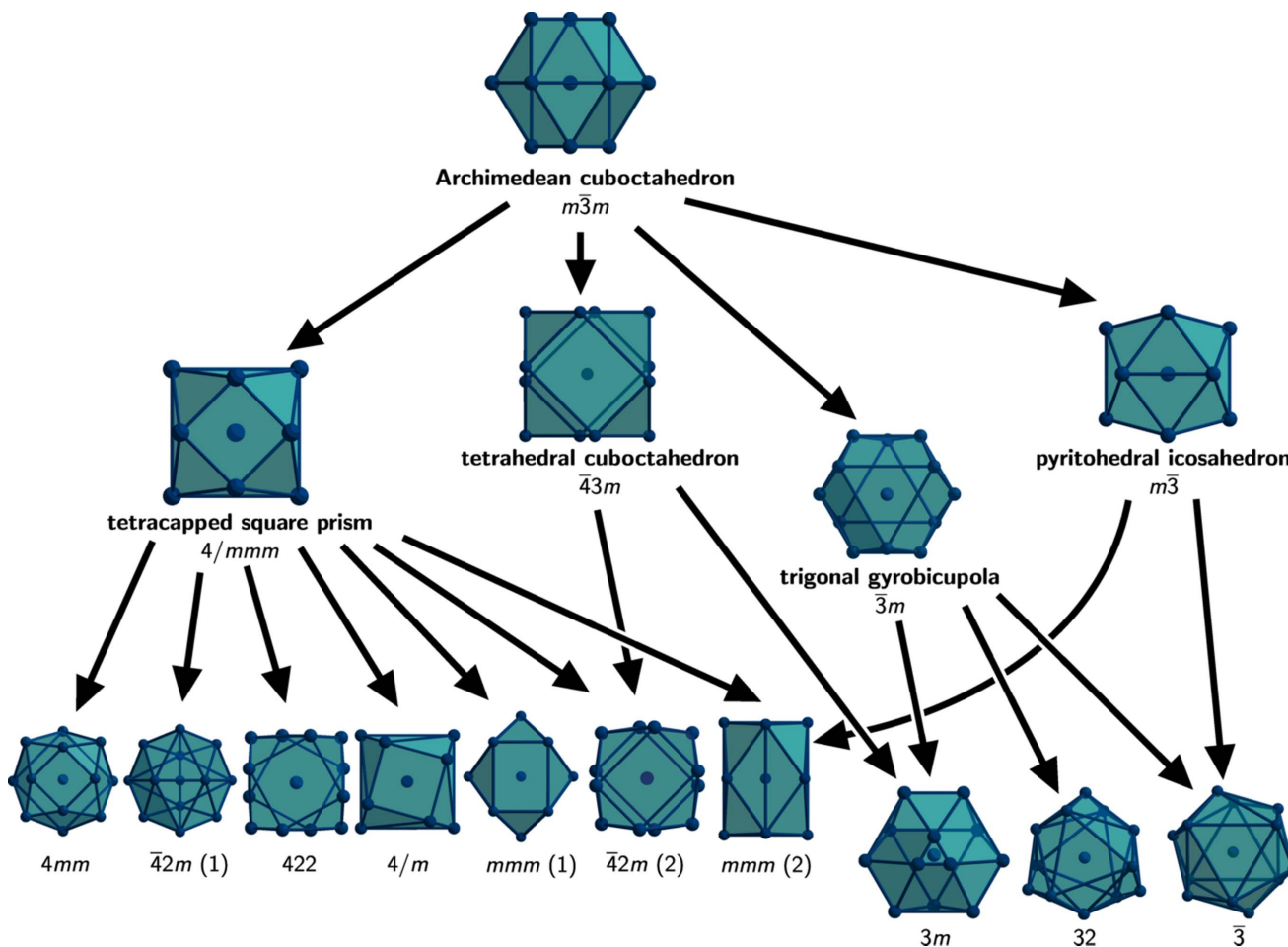


Figure 10 Tree diagram of distorted cuboctahedra. Higher positions in the diagram indicate higher symmetry. Note that there are two differently distorted cuboctahedra with the point group $\bar{4}2m$. Model $\bar{4}2m(1)$ is a combination of a bidisphenoid and a square, whereas $\bar{4}2m(2)$ can be described as a combination of an anticuboid and a disphenoid. Each is viewed along its $\bar{4}$ axis. Similarly, model $mmm(1)$ is a combination of a cuboid and a rhombus, whereas $mmm(2)$ is a combination of a diagonal orthobicupola and a rectangle.

Because of its rich structural diversity, the perovskite family lends itself well to the analysis of its coordination environments. A good example is provided by the eponymous $\text{Ca}(\text{TiO}_3)$ compound which, on heating to 1600 K, undergoes two phase transitions on its way from the $Pbnm$ room-temperature modification via $I4/mcm$ to the $Pm\bar{3}m$ aristotype structure. Fig. 9 shows that the distortion of the cuboctahedral coordination polyhedron of the Ca site decreases with increasing temperature. This is true with respect to the scaleable but otherwise rigid Archimedean model, but also with respect to dynamic models which represent lower-symmetry versions of the cuboctahedron (see Fig. 10). These are automatically deformed within the constraints of their point group in order to achieve the best fitting version. In the example at hand, the $4/mmm$ model starts out with a slightly better fit than the Archimedean model, but this difference all but disappears with the first-order phase transition to the $I4/mcm$ structure. This implies that the distortion of the cuboctahedron is at first in part due to an elongation or compression along one of its fourfold axes, which then vanishes following the phase transition. The $\bar{4}2m$ model fits best from the start and seems to improve roughly at the same rate as the higher-symmetry models. However, after the phase transition, its deviation from the real coordination environment falls to zero, indicating that it matches the point group of the crystallographic site. It follows that a model having all the degrees of freedom permitted by the $\bar{1}$ point group would not be a better fit for the cuboctahedra in the $I4/mcm$ structure than the $4/mmm$ model, as the remaining distortion of the latter is only due to the constraint of inversion symmetry imposed on it. The ratio of distortions relative to these point groups observed in $\text{Ca}(\text{TiO}_3)$ turns out to be overall typical

Table 3

Deviations (δ) from the Archimedean cuboctahedra, number of sampled crystal structures and average deviations from various models as fractions of $\delta(m\bar{3}m)$.

	$I4/mcm$	$I4/m$	$R\bar{3}c$	$R\bar{3}$	$Ibmm$	$I2/m$	$Pbnm$	$P2_1/n$
$\delta(m\bar{3}m)$	5.189	7.289	10.676	9.073	7.953	9.253	18.396	17.470
Sample size	14	30	17	18	9	13	46	32
$\bar{4}3m$	1	0.882	1	0.883	1	0.91136	1	0.979
$m\bar{3}$	1	1	1	1	1	1	0.943	0.945
$4/mmm$	0.999	0.999	1	1	1.000	1.000	0.911	0.913
$\bar{3}m$	1	1	0.991	0.999	1.000	1.000	0.999	0.999
$\bar{4}2m$ (1)	0	0.399	0.819	0.861	0.712	0.770	0.706	0.719
$\bar{4}2m$ (2)	0.999	0.877	1	0.883	1.000	0.909	0.911	0.889
$4mm$	0.999	0.999	1	0.997	0.997	0.990	0.910	0.912
422	0.999	0.997	1	1	1.000	1.000	0.911	0.913
$4/m$	0.999	0.999	1	1	1.000	1.000	0.911	0.913
mmm (1)	0.999	0.999	0.997	1.000	1.000	1.000	0.911	0.913
mmm (2)	0.999	0.999	1	1	1.000	1.000	0.817	0.823
32	0.817	0.858	0	0.424	0.585	0.675	0.753	0.764
$3m$	1	0.882	0.991	0.871	0.9954	0.905	0.997	0.976
$\bar{3}$	1	1	0.991	0.999	1.000	1.000	0.942	0.944

behavior for perovskites. To illustrate this, distorted cuboctahedron models with the point groups $\bar{4}3m$, $m\bar{3}$, $4/mmm$, $\bar{3}m$, 422, $4mm$, $\bar{4}2m$, $4/m$, mmm (two models), 32, $3m$ and $\bar{3}$ were fitted to the cuboctahedra found in a large selection of structures from four major tilted perovskite structure types and their elpasolite counterparts. The results are displayed in Fig. 11 and Table 3.

Several observations can be made. Firstly, while the $\bar{4}3m$ cuboctahedron model fits the cuboctahedra in the $Fm\bar{3}m$ elpasolite with untilted octahedra exactly, it fits only slightly better than the Archimedean $m\bar{3}m$ cuboctahedron model for each of the tilted elpasolite structure types investigated. Therefore, in most cases, the distortion caused by the elpasolite-type

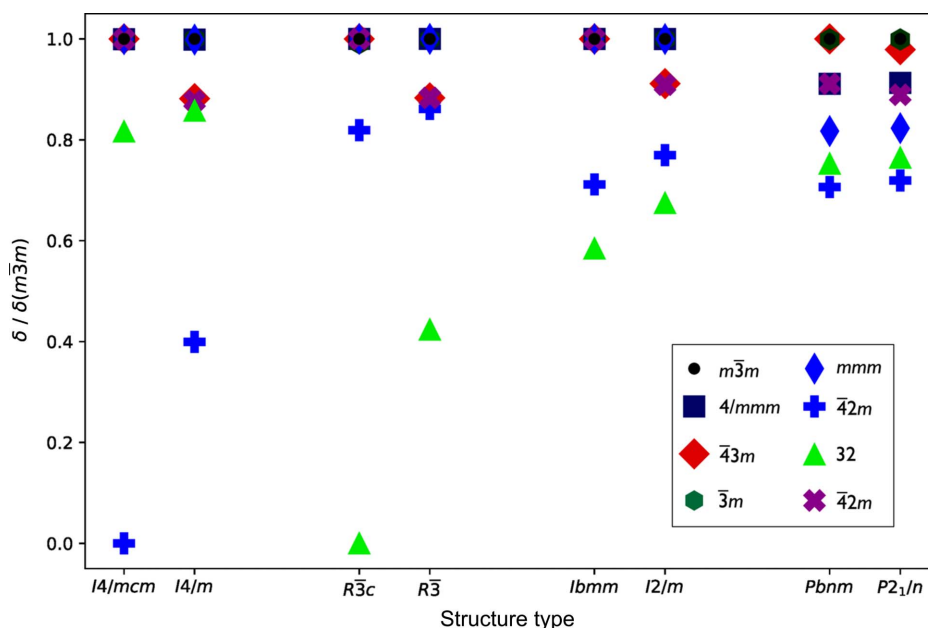


Figure 11

Various models fitted to the cuboctahedra found in four perovskite structure types and their respective elpasolite counterparts. Each structure type is represented by an array of crystal structures belonging to it. Each individual distortion δ was first divided by the distortion of the Archimedean cuboctahedron model fitted to the same coordination environment. Thus the arithmetic mean of all values generated was then taken for each combination of a model and a structure type in order to obtain a data point shown in the diagram. See the supporting information for a discussion of the methodology applied in this plot.

superstructure is much smaller than the distortion caused by the octahedral tilting. The difference is even more pronounced with lower space-group symmetries. This result is to be expected, as the distortion by tilting increases with decreasing space-group symmetry, while the difference in octahedron sizes should stay roughly the same on average. The maximal hettohedra with point groups $4/mmm$ and $\bar{3}m$ do not fit noticeably better than the Archimedean cuboctahedron, except in the case of the $Pbnm$ and $P2_1/n$ structure types, where the $4/mmm$ model shows some relative improvement. This suggests that there is generally no significant elongation or compression along the three- and fourfold (pseudo-)axes for the higher-symmetry structure types, as both would improve these fits. However, the $Pbnm$ and $P2_1/n$ structure types, which represent the $a^-a^-c^+$ tilting scheme (Glazer, 1972), adopt an elongation along one of the fourfold pseudo-axes, more precisely one of the axes along which the tilting has alternating rotational directions. Note that the elongation does not stem from an elongation of the octahedra, as compounds with octahedral distortions $\delta > 3.2$ were filtered out and the majority of octahedra in the admitted structures were better described as trigonal antiprisms than as tetragonal bipyramids.

The overall best fits are, in all cases, achieved by the 32 and $\bar{4}2m$ (1) models. These perfectly match the site symmetry for the $I4/mcm$ ($a^0a^0c^-$) and $R\bar{3}c$ ($a^-a^-a^-$) structure types, respectively. With the elpasolite counterparts $I4/m$ and $R\bar{3}$, the site symmetry is reduced to $\bar{4}$ and 3, respectively. This goes along with a significant deterioration of the fit for these models, which far outweighs the concomitant relative improvement of the $43m$ model fit.

Somewhat interestingly, the 32 model fits better than the $\bar{4}2m$ model for the $Ibmm$ and related $I2/m$ structure types. This is despite the fact that the cuboctahedral site in the $Ibmm$ structure type has $mm2$ symmetry, which is a maximal subgroup of $\bar{4}2m$, but not closely related to 32 .

4. Limitations

In some cases, especially for lower symmetries and larger numbers of vertices, more than one model may be required to cover all possible manifestations of a point group. This means that the strengths and weaknesses of *Polynator* when it comes to symmetry analysis are complementary with *CoSyM*. *CoSyM* especially shines during the measurement of symmetry deviations for low-symmetry point groups but is currently not equipped to cover, for example, cubic point-group symmetries, except in some trivial cases.

Since the atom-assignment process does not check all possible permutations, suboptimal fits can sometimes result if a given model does not resemble the evaluated atom arrangement at all. Although the reliability of the assignment algorithm depends on the number of atoms, the model and the particular type of distortion, fit results with δ values below 30 are generally highly trustworthy, which means this is not an issue in practice.

User-defined models may have a large variety of shapes, but not all possible shapes are guaranteed to work well with the existing strategies for assigning model vertices to atom vectors. Furthermore, in order to be properly fitted, the centroid of each model belt has to rest on the model axis. This should rarely be a problem in practice, since model shapes generally need to be somewhat symmetrical to be useful in the first place.

5. Conclusions

Polynator allows for the comparison of distortions by quantifying the dissimilarity between one or more rigid or dynamic models and a given arrangement of atoms found in a crystal structure. The GUI and the automatic pairing of model and real vertices make it very easy to use. *Polynator* comes with a large host of preset models, which can easily be expanded by the user according to specific needs. In contrast to earlier programs, models can feature multiple degrees of freedom. Hence, a model may represent either a rigid shape or a specific point-group symmetry (or subset thereof). Furthermore, there are models which are not rigid and neither is their shape determined by symmetry alone. This latter category contains models representing molecules with flexible joints and rigid components like the biphenyl unit, but also dynamic model polyhedra with specific constraints, such as the pyritohedron with its planar pentagonal faces. The program is a useful tool for identifying polyhedra and molecular shapes, quantifying their distortions, obtaining structural parameters such as polyhedral volumes or dihedral angles between molecular subunits, and making observations about structural trends.

Acknowledgements

The authors declare no conflicts of interest. Open access funding enabled and organized by Projekt DEAL.

References

- Alon, G. & Tuví-Arad, I. (2018). *J. Math. Chem.* **56**, 193–212.
- Alvarez, S. (2017). *Coord. Chem. Rev.* **350**, 3–13.
- Alvarez, S., Alemany, P. & Avnir, D. (2005). *Chem. Soc. Rev.* **34**, 313–326.
- Ban, Z. & Sikirica, M. (1965). *Acta Cryst.* **18**, 594–599.
- Broussard, L. & Shoemaker, D. P. (1960). *J. Am. Chem. Soc.* **82**, 1041–1051.
- Burns, D. M. & Iball, J. (1955). *Proc. R. Soc. London A*, **227**, 200–214.
- Cumby, J. & Attfield, J. P. (2017). *Nat. Commun.* **8**, 14235.
- Dollase, W. A. (1974). *Acta Cryst.* **A30**, 513–517.
- Echeverría, J. & Alvarez, S. (2008). *Inorg. Chem.* **47**, 10965–10970.
- Glazer, A. M. (1972). *Acta Cryst.* **B28**, 3384–3392.
- Hauke, W. (1940). *Z. Anorg. Allg. Chem.* **244**, 17–22.
- Holý, P., Podlaha, J., Císařová, I. & Závada, J. (2001). *Collect. Czech. Chem. Commun.* **66**, 947–958.
- Kabsch, W. (1976). *Acta Cryst.* **A32**, 922–923.
- Kasper, J. S., Hagenmuller, P., Pouchard, M. & Cros, C. (1965). *Science*, **150**, 1713–1714.
- Leinenweber, K. & Parise, K. (1995). *J. Solid State Chem.* **114**, 277–281.
- Llunell, M., Casanova, D., Cirera, J., Alemany, P., Alvarez, S. (2013). *SHAPE. Program for the Stereochemical Analysis of Molecular*

- Fragments by Means of Continuous Shape Measures and Associated Tools*. University of Barcelona, Spain.
- Mariathasan, J. W. E., Hazen, R. M. & Finger, L. W. (1986). *Phase Transit.* **6**, 165–173.
- O’Keeffe, M. (1979). *Acta Cryst.* **A35**, 772–775.
- Pinsky, M. & Avnir, D. (1998). *Inorg. Chem.* **37**, 5575–5582.
- Popov, I. A., Jian, T., Lopez, G. V., Boldyrev, A. I. & Wang, L.-S. (2015). *Nat. Commun.* **6**, 8654.
- Stoiber, D. & Niewa, R. (2019). *Z. Kristallogr.* **234**, 201–209.
- Waroquiers, D., George, J., Horton, M., Schenk, S., Persson, K. A., Rignanese, G.-M., Gonze, X. & Hautier, G. (2020). *Acta Cryst.* **B76**, 683–695.
- Yashima, M. & Ali, R. (2009). *Solid State Ionics*, **180**, 120–126.
- Zabrodsky, H., Peleg, S. & Avnir, D. (1992). *J. Am. Chem. Soc.* **114**, 7843–7851.
- Zintl, E. & Haucke, W. (1938). *Z. Electrochem.* **44**, 104–111.