VILNIUS UNIVERSITY

ANDRIUS MERKYS

EXTRACTION AND USAGE OF CRYSTALLOGRAPHIC KNOWLEDGE FOR REFINEMENT AND VALIDATION OF MOLECULAR MODELS

Doctoral dissertation
Technological sciences, chemical engineering (05T)

Vilnius, 2018

The dissertation work was carried out at Vilnius University from 2013 to 2017

Scientific supervisor – prof. dr. Saulius Gražulis (Vilnius University, technical sciences, chemical engineering – 05T).

VILNIAUS UNIVERSITETAS

ANDRIUS MERKYS

KRISTALOGRAFINĖS INFORMACIJOS IŠGAVIMAS BEI PANAUDOJIMAS MOLEKULIŲ MODELIŲ TIKSLINIMUI IR TIKRINIMUI

Daktaro disertacija
Technologiniai mokslai, chemijos inžinerija (05T)

Vilnius, 2018

Disertacija rengta 2013–2017 metais Vilniaus universitete

Mokslinis vadovas – prof. dr. Saulius Gražulis (Vilniaus universitetas, technologiniai mokslai, chemijos inžinerija – 05T).

Acknowledgments

I am very grateful to my supervisor Saulius Gražulis for guidance and invaluable insights in bioinformatics, cheminformatics and science in general.

I would like to thank the head of Vilnius University Institute of Biotechnology Department of Protein–DNA Interactions Virginijus Šikšnys and my present and former colleagues from the Department, especially Elena Manakova, Justas Butkus, Antanas Vaitkus and Algirdas Grybauskas. My thanks also go to Nicola Marzari, Giovanni Pizzi, Nicolas Mounet, Ivano E. Castelli, Andrea Cepellotti, Marco Gibertini, Philippe Schwaller, Miguel Quirós Olozábal, Aleksandras Konovalovas, Garib N. Murshudov, Peter Murray-Rust, Fei Long and Robert A. Nicholls. I am also grateful to the reviewers of the dissertation, Mindaugas Bloznelis and Kliment Olechnovič, for their insight and comments.

Special thanks go to my wife Miglė, my parents, sisters and grandparents for their endless patience and support. Many thanks to my friends who always were there for me.

This research was partially funded by the Research Council of Lithuania grant agreement No. MIP-025/2013, SCIEX Scientific Exchange Programme fellowship No. 13.169, MARVEL National Centre of Competence in Research of the Swiss National Science Foundation and European Union's Horizon 2020 research and innovation program under grant agreement No. 689868.

Table of Contents

Chapter 1

Introduction

Problem

Knowledge of atomic arrangement in crystal structures has led to unprecedented achievements since the beginning of the 20th century. Determination of the structures of the first organic compounds (1930s), myoglobin (Kendrew, 1958, awarded Nobel Prize in 1962), DNA (Franklin, Wilkins, Watson & Crick, 1952-1954, awarded Nobel Prize in 1962) and ribosome (Ramakrishnan, Steitz & Yonath, early 2000s, awarded Nobel Prize in 2009), to name a few, have resulted in novel insights into both the structure and function of the main driving components of life. All of these breakthrough studies were carried out thanks to crystallography, which provides mathematically sound methodology to relate diffraction patterns from the crystals of chemical compounds to their atomic models [1].

X-ray crystallography, a prominent method of crystal structure determination, consists of non-trivial steps leading to the 3D coordinates of atomic structure of a material in question. Usually data from X-ray diffraction alone is not enough to determine all parameters of a macromolecule independently. Small molecules, which require orders of magnitude less parameters to define, are used as a reference for the geometry of macromolecules. In this way additional observations are introduced in model building of macromolecules as “restraints”, or model parameters are eliminated by defining “constraints”. Restraints and constraints are usually applied to interatomic bond lengths, bond angles and dihedral angle sizes. Additionally, a group of atoms may be forced to stay on a same plane or move together as a rigid body (retaining internal distances and angles) during the refinement. It is obvious that both the supplemental observations and the strict geometric relations should be derived from top quality structures [2, 3, 4] and reflect the geometry of highly similar compounds [5, 6, 7, 3, 8, p. 221–250]. Both requirements are not trivial to achieve.

The number of small molecule structures solved each year is increasing with time [9]. Number of entries in the Cambridge Structural Database (CSD), the largest archive of small molecule structures, has doubled over the last decade, reporting growth rate of over 50 000 new structures each year[1]. However, this number of novel structures is much larger than the number of

[1] `https://www.ccdc.cam.ac.uk/solutions/csd-system/components/csd/`, accessed on 2017-07-12

experienced crystallographers and journal referees [10], thus low quality or incorrect structures sometimes get published [11]. Software tools are often employed to detect such structures by spotting unusual features in them, comparing the structures in question to the libraries of geometric knowledge of crystal structures. Commonly, these libraries are compiled manually by the experts in the field. Recently, attempts of automatic construction of geometric libraries have taken place [7, 12], using the CSD as a source [13]. However, data derived from the CSD is subject to the restrictive CSD license and not suitable to be used and disseminated freely[2] [14].

Objectives

- Develop methods and software for the extraction of geometric parameters from small molecule crystal structures. Employ the developed software to collect geometric parameters of crystal structures from the Crystallography Open Database (COD[3] [15]).
- Develop methods and software to construct a library of geometric knowledge of small molecule crystal structures. Use the developed software to organise and describe the parameters collected from the COD.
- Develop methods and software for validation of small molecule crystal structures against the constructed library of geometric knowledge.

Scientific novelty, results and their value

Usually data and software from the CSD is used for the construction of geometry libraries. As the CSD is a proprietary database, its data, programs and the derived libraries are subject to the CSD license, therefore not suitable for public domain. We have developed open-source software (released under GNU GPL2 or compatible licenses) and used it to extract data from the COD, an open-access collection of small molecule crystal structures. Using the COD instead of the CSD allows unrestricted dissemination of the results.

Most of the libraries postulate that each geometric parameter follows the normal distribution. In practice, however, asymmetric, multimodal or otherwise non-normal distributions are observed by almost all researchers of the field. In our study we have replaced the normal distribution with mixtures of location-scale family distributions. This substitution allows flexible description of all aforementioned cases.

The most common method for outlier detection (validation) assuming normal distribution is Z score, that is a measure of deviation from the mean of a normal distribution [16]. Having distributions of geometric parameters described as mixtures, we have employed Bayesian framework for outlier detection.

Developed software for geometry extraction, distribution description and validation is fully automated and is prepared for unsupervised data processing. Software to extract the geometric parameters and describe their distributions is prepared to be used for automated periodic updates of the COD geometry library. The validation interface is open for general public,

[2] PURY licensing policy. `http://pury.ijs.si/beta_servers.html`, accessed on 2017-07-12

[3] `http://www.crystallography.net/cod`

and it is as well ready to be integrated into the COD data deposition pipeline to check the crystal structures prior to their deposition in the database.

Propositions to be defended

- Crystallography Open Database can be used as a source of structural small-molecule information to build a knowledge library of crystal geometry.
- The developed method for unsupervised extraction and organisation of small-molecule geometry information is suitable to describe the variety and features of small-molecule crystal geometry.
- The developed library is suitable for outlier detection using unsupervised Bayesian methods.

Chapter 2

Literature overview

2.1 Organisation of atoms in crystals

Crystals are generally regarded as formed by stable (under arbitrary conditions) arrangements of atoms that are periodic in three dimensions. Although highly ordered aperiodic structures (quasicrystals) are known to exist [17], only periodic ones are considered in this research. Organisation of atoms and molecules in crystals are mostly governed by intra- and inter-molecular, and crystal packing interactions. First two can be described (in the most simple way) using theories of valence shell electron pair repulsion (VSEPR) and Lennard–Jones potential, correspondingly. Intermolecular and crystal packing interactions are extensively modelled using the concept of van der Waals surfaces [18]. Despite the fact that the fundamental rules of atom organisation in crystals are well-known to the point of making predictions of crystal structures possible [19], incontrovertible results are achieved only by experimental means.

2.1.1 Connectivity

Classification of atom contacts as either intra- or inter-molecular plays a pivotal role in crystal structure understanding and determination, since modelling of these interactions differ vastly: intra-molecular forces are shown to be stronger than their inter-molecular counterparts [20, p. 6]. This classification depends on the concept of “connectivity”, that is, perception of the network of molecular bonds between the atoms of a compound. A bond is in most cases understood as a relation between strictly two atoms. For macromolecules and other organic compounds which are described well by the valence bond theory (VBT), connectivity is well-defined and usually known *a priori* to the structure determination. However, the concept of connectivity is not defined absolutely and unambiguously for all types of compounds, thus its extension outside the organic subset (for example, purely ionic and metallic compounds, boranes, metallocenes) depends on conventions [21].

Given a set of atoms in space it is possible to detect their connectivity algorithmically: two atoms are deemed to be connected if their distance is shorter than a sum of their bonding radii, typically covalent radii, which is the experimentally observed contribution of an atom to covalent bond distances in different compounds and crystals [20, p. 221–222]. This method is known since as early as the publications by William Lawrence Bragg [22]. It has been shown that the variance of atom's covalent radii is small in different environments, therefore, covalent radii are usually treated as constant values [20, p. 221–222]. However, there is no single univocal table of covalent radii, since methodologies for defining them differ strongly. It has been noted that clear gaps exist in the atom pair distance distributions, allowing to differentiate between bonded (shorter distances) and non-bonded (longer distances) interactions. Unpopulated ranges in between them correspond to the so-called van der Waals gaps, regions of energetically unfavourable interatomic distances, which may in some cases be "contaminated" due to the presence of non-covalent interactions. Furthermore, distance distributions of some elements, such as alkaline elements, copper, silver, mercury, iron, tin, do not have such clear van der Waals gaps and pose a difficulty in fitting them to the aforementioned approach [18]. Nevertheless, distance criterion for connectivity is widely used, and covalent radii for problematic elements are approximated using, for example, quantum chemical calculations [23].

2.2 Crystal structure determination

Crystal structure determination, like the most of the scientific experiments, is a workflow consisting of sample preparation, instrumentation, measurement, calculation and interpretation. Crystals of the material under study are prepared during the process of crystallisation. The resulting samples are then characterised by recording diffracted X-ray or neutron reflections in a few different orientations. Measured diffraction intensities are converted into electron density, which is in the last stage used to construct an atomic structure [24], usually by fitting ("refining") spherical atoms into the densest regions of electron density map. Each step from the acquisition of the experimental data to the refinement is complex and could be potentially insolvable [8, p. 251–269]. Clarity of the electron density map depends on quality of diffraction data, which in turn relies on intrinsic order of the crystal. Successful structure determination results in the coordinates, occupancy and displacement parameters for all atoms of the structure. Structure determination by crystallography follows a clear mathematical procedure, producing electron density directly from the experimental data [3]. Therefore, careful and well-documented refinement of sufficiently high quality data leaves no doubt about the atomic structure of a crystal. However, the lower the data quality, the more assumptions have to be made about the underlying structure.

2.2.1 X-ray diffraction

X-ray diffraction is prominent in the crystallography, as the wavelength of X-rays is comparable to the molecular dimensions [8, p. 333–342]. X-rays are reflected from electrons of the crystal structures, therefore, in fact, positions of electrons are determined using this method [25]. This property results in systematically displaced electron cloud positions with respect to nuclei of hydrogen atoms, that are determined shifted towards the atom to which the hydrogen atom is covalently bonded. Consequently, resulting electron density maps show C–H and C–N bonds 0.1 Å shorter as compared to spectroscopic measurements and neutron diffraction [8, p. 205–219]. This may as well lead to incorrect interpretations of chemical atom types at low resolution: C (6 electrons) might be detected as N (7 electrons), and N in turn might be detected as O (8 electrons) due to inability to tell the attached hydrogen atoms from their covalent neighbours.

2.2.2 Neutron diffraction

Capabilities of X-ray crystallography are partially covered by neutron diffraction. In neutron diffraction experiment, an incident beam of neutrons is used instead of X-rays. As neutrons are scattered by nuclei, atomic positions are determined without the biasses caused by bonding, lone pairs and other valence-electron density features. This is particularly relevant for H atoms, whose electron density is usually shifted towards the adjacent atom causing systematic bond shortening in X-ray studies. The same holds for other light atoms. In addition to that, neutron diffraction is better than X-ray in distinguishing between atoms of neighbouring elements in the periodic table, as for such atoms X-ray gives very similar scattering, rendering telling them apart difficult. As neutron scattering can be very different for even the neighbouring elements of the periodic table, this method may be crucial for reliable structure determination, especially for metal alloys and mixed-metal complexes [8, p. 333–342].

2.2.3 Powder diffraction

In cases when single crystals of material under study are not available, diffraction from multitude of randomly oriented tiny crystals ("powder") is analysed using either X-ray, neutron or electron diffraction. Rietveld method [26, 27] is then used for refinement against collected data, although starting coordinates are needed for the refinement, as the data alone is not enough to construct the initial crystal density map. However, reports on ab initio crystal structure determination from powder data have started to appear recently [8, p. 251–269].

2.2.4 Refinement

Interpretation of electron (or, in the case of neutron diffraction, neutron) density leads to the construction of the initial rough model by placing its atoms inside the density map. The starting model usually contains a lot of small errors in its geometry. These errors are then removed in the refinement process, in which model's parameters are iteratively improved until best fit to the experimentally derived density is reached [3]. The refinement is driven by measures of agreement between the observed data and the constructed model. The most commonly used measures are the R factor, goodness-of-fit and shift/standard uncertainty ratio. The first one takes into the consideration the average differences between observed (F_{obs}) and calculated (F_{calc}) structure factors:

$$R = \frac{\sum\limits_i |F_{obs,i} - F_{calc,i}|}{\sum\limits_i |F_{obs,i}|}, \tag{2.1}$$

In goodness-of-fit measure the differences are squared and the denominator is replaced by the difference between the numbers of used individual reflections and model parameters. Shift/standard uncertainty ratio measures the maximum or average difference between the parameter estimates in two consecutive refinement iterations. Weighted, or generalised R factor, denoted R_w, is sometimes used instead by introducing weights w_i for each member of sums in both the numerator and denominator of Equation 2.1. Both R and R_w are usually criticised, as they can be adapted ("massaged") to prove a better fit to the data [8, p. 221–250]. In addition to that, anomalous structure features, such as heavy atoms, may dominate the rest of the features by influencing the R factor, thus unwittingly hiding misfits [28]. For example, the contribution of an oxygen atom (8 electrons) is extremely low in the vicinity of a bismuth atom (83 electrons) [29]. "Free" R factor (R_{free}), introduced as a means of cross-validation, is calculated using observed data not used in the model construction. Causing a lot of discussions in the last decades of the 20th century, R_{free} has become a common practice, and was reported in 92% of all protein crystal structure studies of the year 2000 [30].

Refinement is an optimisation of either empirical energy function, least-squares residual [1] or likelihood [31, 32, 33]. For the refinement without additional prior information, the number of observations per refined parameter has to be more than 10, what is only possible at atomic resolution of at least 1.2 Å. However, such resolutions are generally limited to the structures of small molecules (<900 Da), allowing these structures to be determined with great accuracy and precision [3]. And even structure determination with data of atomic resolution in some cases requires the use of prior knowledge to resolve disordered regions [34]. Crystallography of macromolecules usually ranging from tens to thousands kDa (proteins) generally uses lower resolution data; for this reason typical observation to parameter ratio for macromolecules ranges from 0.5 to 5 [5] and requires either restraints or constraints for refinement [3]. Introduction of restraints increases the number of observations, whereas application of constraints reduces the number of model parameters. Thus either way the ratio of observations per parameter is modified in favour of better model convergence [5]. Although prior knowledge is a powerful tool to drive poor initial model towards correct stereochemistry, it should be used only if other initial models are not available, as better outcome is always achieved by starting from a high quality initial coordinate set [4]. Furthermore, it should be noted that the less experimental data is available, the more the model is based on prior geometrical knowledge [3]. Both a restraint and a constraint is held as an expression of prior chemical or physical knowledge of the system and is usually expressed as a target value for a single geometric parameter: bond length, bond angle or dihedral angle (expressed either directly or by distances between constituent atoms [35, 36]), nonbonded contacts, planarity [1] and chiral volume [37], plus an indication of deviation allowed from the associated value [33, 5]. Usually restraints and constraints are collectively referred to as restraints, as a constraint can be viewed as a specific kind of restraint having zero allowed variance. In the case of a value following normal distribution, the allowed deviation (in other words, the confidence in it) is indicated by the standard deviation (σ) [33]. For least-squares refinement the following penalty function could be used [35]:

$$
\begin{aligned}
P = w_{\text{reflections}} \sum_{i} & w_i(|F_{\text{obs},i}| - |F_{\text{calc},i}|)^2 + \\
+ w_{\text{bonds}} \sum_{b \in \text{bonds}} & w_b(d_{\text{obs},b} - d_{\text{ideal},b})^2 + \\
+ w_{\text{angles}} \sum_{a \in \text{angles}} & w_a(\alpha_{\text{obs},a} - \alpha_{\text{ideal},a})^2 + \\
+ w_{\text{dihedrals}} \sum_{d \in \text{dihedrals}} & w_d(\alpha_{\text{obs},d} - \alpha_{\text{ideal},d})^2 + \\
+ P_{\text{nonbonded}}, &
\end{aligned}
\tag{2.2}
$$

where w_b, w_a and w_d are weights which are used to control the significance of geometric parameters and might be defined as $w_i = \sigma_i^{-2}$. Improper usage of weights can result in completely deformed models. For parameters with too few or none observations, quantum mechanics calculations may be used to derive the standard deviation values [4]. Typical weights both in protein and ligand refinement are 0.02 Å for bonds and 2° for angles [5].

In the beginning of protein crystallography the protein itself was the main target of the research [3]. For the refinement of their structures geometrical knowledge was collected from X-ray and neutron diffraction structures of small-molecule structures of individual amino acids or oligopeptides. Later on this knowledge was improved based on analysis of large databases [38], resulting in comprehensive restraint libraries. Recently the highlight shifted to small molecules that are bound to macromolecules and acting as ligands, cofactors, inhibitors, metal clusters, ions, solvent molecules or drugs [35, 3]. A recent study concluded that more than 75% of the protein structures in the Protein Data Bank (PDB) contained one or more small molecules alongside their protein content [4]. Modern methods allow "freezing" of such complexes during reactions or molecular binding processes providing insight into intermediate states of their mechanisms [25]. However, the restraints for chemically synthesised ligands bound to proteins are much more difficult to generate reliably since the structures of most of these ligands are not observed at high resolution, and even if they are, observations are often scarce. This is chiefly due to the diversity of chemical composition and conformation of ligands [3]. As a consequence ligand structures in protein-ligand complexes suffer from poor model quality more often than it would be desirable [5]. Lack of usable stereochemical knowledge of inorganic compounds results in poorly defined parts of structures and is perceived as a serious bottleneck in high-throughput crystallography. Furthermore, restraints used for such refinements are hardly ever mentioned in structure reports preventing their reproduction [39].

Taylor et al. have classified methods to obtain the prior knowledge to two groups: theory- and database-derived, the former consisting of calculations of force fields and quantum mechanics, and the latter relying on crystal structure data from structural databases [13]. Although results of application of theory-based approach can be promising [39], Taylor et al. argue in favour of database-derived stereochemical information over theoretical as the former better represents the *in vivo* environment. Theoretical energy calculations almost always are performed in vacuum, effectively limiting their use for modelling of aqueous solutions, what protein crystals usually are [40, 13]. For example, carboxyl group would always be protonated in vacuum, although the protonation state and the geometric details of ligands depend on their environment. Moreover, high-level theoretical calculations usually require large computational resources [25]. Another argument in favour of database-derived stereochemical information is the similarity of assumptions made during the determination of the structures in the source database to the "recipients" of the derived information. A common example of incorrect usage is the application of carbon–hydrogen bond length of 1.08 Å, a value determined spectroscopically from simple hydrocarbons, for X-ray determined structures. As carbon–hydrogen lengths are systematically shorter in X-ray analyses, spectroscopically derived restraint will try to push one or both connected atoms out of their electron density maxima [8, p. 221–250].

An assumption is held that the geometry of ligands in small molecule crystals quite well reflects the geometry attained by the ligand in protein-ligand complexes [41, 3, 42]. However, this is not always true, as crystal packing effects in small molecule crystals cause non-bonded contacts to be comparatively shorter and hydrophobic contacts less common than in protein-ligand complexes [40, 43]. Structures of isolated ligand should not be exclusively relied upon to model ligands in protein-ligand binding sites [41], as studies of ligands at protein binding sites in the PDB report ligand conformations out of their energetic minima. It is reported that average strain energy per torsional angle is 0.6 kcal mol^{-1} with a maximum of 3 kcal mol^{-1} total strain energy per ligand. Others suggest strain energies greater than 9 kcal mol^{-1} in as much as 10% of analysed ligands [2].

2.3 Molecular databases as a source of the knowledge

Since the beginning of the 21st century, public Web databases have become valuable resources among researchers, who trust and rely upon them for their data for the use in cheminformatics, bioinformatics, systems biology, medicine and drug research [44]. Virtual high-throughput screening is performed on large amounts of crystallographically derived molecule models for feature mining, protein-ligand docking and molecular superposition [40, 43]. Distributions of geometric parameters in small molecule crystal structures are used for the construction of probability spaces. For example, if a particular molecular fragment with a rotatable bond is found in a set of crystal structures, it is likely that lower energy conformations will occur more often than higher energy conformations. Therefore, the potential energy function for this bond could be replaced by the observed distribution of dihedral angles [40]. Derived probability spaces can serve as input in Bayesian methods for chemical structure assignment, which attempts to answer the question "what is the most likely chemical structure of a compound given its geometry?" [45]. A much promising feature of online structural databases is their constant growth. Taylor et al. (2014) envisaged and developed a robust and future-proof automated system for the derivation of knowledge database of molecular geometry (reviewed in detail in Section 2.8.8), capable of performing unsupervised periodic updates. Authors also implemented manually populated set of "problem fragments" which would be deemed as intractable automatically [13]. The distinctive property of small molecule structure databases is that the structures in such databases should be determined without prior knowledge of molecular geometry, with a possible exception of parameters of hydrogen atoms, solvent and disordered regions. For structures determined in single crystal studies, this means using full matrix least squares refinement without any assumptions on molecular geometry [15]. Therefore, knowledge derived from such resources does not carry prior assumptions, effectively preventing from fallacy of circular reasoning. However, there is still some controversy as to how far the abstractions of crystallographically derived data could be employed in other fields of study. For example, it is noted that molecules in crystals are exposed to conditions that usually do not exist in the environments of interest [46]. Nevertheless, Cruz-Cabeza et al. (2012) concluded that geometry in crystallographically derived stereochemistry of protein–ligand binding sites are closer to reality than coming from theoretical calculations of gas phase or *in vacuo* [47]. In conclusion, the knowledge extracted from crystallographically derived data should be applicable to other crystal structures, provided that the same or similar assumptions were made.

The largest to date resource of small molecule crystal structures is the Cambridge Structural Database[1] (CSD), containing around 900 000 organic and organometallic crystal structures [12]. Inorganic Crystal Structure Database[2] (ICSD) [48] and Crystal Data for Metals Database[3] (CRYSTMET) [49] are complementary to the CSD by collecting structures of inorganic compounds, metals and their alloys [8, p. 327–331]. The CSD, developed by Cambridge Crystallographic Data Centre (CCDC), has been a prominent source for stereochemical knowledge derivation since the study by Engh & Huber in 1991 (reviewed in Section 2.8.2) [4]. However, usage and distribution of data derived from the CSD is limited by its license to the subscribers of the database. While traditionally it has been perceived that requiring readers to pay for monographs or database access should support the human effort to compile them from the scientific literature, largely increased amount of data, advent of computer networks and automated systems not needing human supervision has led to social and political debates about the ownership and intellectual property associated with the scientific data [50]. Furthermore, results abstracted from the CSD are of limited use for the studies of inorganic materials, as the CSD does not contain crystal structures of this kind. Circumventing this limitation, CSD software could be used to derive results from complementary inorganic databases, such as ICSD, PDF[4] and Pauling file[5] [29]. Another alternative source is the Crystallography Open Database[6] (COD). Founded in 2003 and totalling more than 390 000 entries, the COD aims at collecting all available organic nonpolymeric, inorganic, metal-organic compounds and minerals into a single public domain database [51]. Having the whole spectrum of small molecules simplifies multidisciplinary research [15], while open-access nature of the COD, allowing immediate access to the data and putting no bounds on sharing it, encourages data cross-linking and derivation of knowledge, as evident in the recent studies [52, 53]. Openness of scientific data and knowledge without any artificial barriers is being recognised by a growing number of institutions as pivotal in global development. United Nations Educational, Scientific and Cultural Organisation (UNESCO) has expressed a commitment for support and promotion of open access to the scientific information [54].

It is a well known fact that the utility of data crucially depends on its quality. Williams

[1] `http://www.ccdc.cam.ac.uk/products/csd/`
[2] `http://www2.fiz-karlsruhe.de/icsd_home.html`
[3] `http://www.tothcanada.com/databases.htm`
[4] `http://www.icdd.com/products/pdf4.htm`
[5] `http://paulingfile.com`
[6] `http://www.crystallography.net/`

and Ekins (2011) point out that the quality of the chemical structure-based data in the public domain is poor [44]. Spek (2002) notes that even peer-reviewed publications tend to describe interesting features of molecular structures that turn out to be based on overlooked artefacts [10]. Thus, individual entries in database pools should not be trusted ultimately, despite the fact that careful manual analysis of every entry is usually not feasible. Surely only high-quality structures provide enough confidence in the unique features reported [11]. Another limiting factor for usage in macromolecular refinement is underrepresentation of ligands in the small molecule databases [4]. As discussed before in Section 2.2.4, the diversity of chemical composition and conformational freedom of small molecules is overwhelming. Andrejašič et al. (2008) noticed that only 12% ligands found in complexes in the PDB had an exact match in the CSD [7]. Therefore, heuristics for fuzzy matching may be necessary.

2.3.1 Knowledge extraction from small molecules

As an initial step, crystal structures are usually filtered in order to remove low quality, inappropriate or intractable entries. The most of the studies impose a cutoff for crystallographic R factor as a means for quality control. Then application-specific checks usually commence. For example, virtual screening studies of the COD tend to exclude crystal structures with partial occupancies of atom sites [53, 55].

Crystallographic descriptions then have to be converted to chemical in a process sometimes called "structure assignment": they have to get chemical bonds identified, polymers detected and limited to representative units (if needed), disorder recognised, bond types and formal charges assigned as well as missing hydrogen atoms located [45]. This process involves many parametrised heuristics, heavily sensitive both to the input and the selection of parameters. Algorithmic structure assignment was deemed to have a success rate of 85% in 2005 [43], leaving for manual correction the rest of the features, that can be easily overlooked. Inaccuracies and errors may lead to completely incorrect results [56]. As crystal structures exhibit symmetry, they are almost always described reduced to an asymmetric unit, accompanied by either identifier of the symmetry group or explicit symmetry operators [8, p. 20]. To measure the geometric parameters of molecules in crystal, its contents have to be reconstructed given the set of atoms in asymmetric unit and the group of crystal symmetry operators. In short, every symmetry operator is applied to every atom, all resulting atoms are translated to the unit cell and coinciding atoms are merged together. Since crystal structure determination rarely provides connectivity information, chemical bonds between atoms are "discovered" using distance heuristics: two atoms are considered bonded if distance between them is less than the sum of their covalent radii [57]. However, there are many covalent radii sets, reflecting different opinions about covalent bonding [45]. Blake (2009) argues that this well-known method is generally valid for organic compounds, but special care is required for other types of compounds where covalent radii are not so well defined [8, p. 299–317]. Automated topology determination is frequently deemed unreliable [4], however, its usage is inevitable unless author-provided topology description is present, what is rarely the case. Bruno et al. (2011) report a number of chemically annotated structures whose bonding and non-bonding distance distributions overlap substantially, concluding that opinions of the authors are sometimes often contradictory regarding bonding interactions. Authors also report many complications in algorithmic chemistry detection originating from the metallo-organic crystal structures, particularly in asserting the oxidation states, coordination numbers and aromaticity. Disorder is also noted as difficult to tackle [45]. Small molecule crystals usually have high symmetry, which has to be taken into account when locating independently observed geometric parameters. For example, four-coordinate metal atom on an inversion center participates in only two independent bond lengths and one independent bond angle [8, p. 299–317]. Therefore, symmetrically equivalent measurements have to be programmatically filtered out in order to prevent unwanted overrepresentation.

2.3.2 Crystallographic data formats

The need of machine-readable crystallographic data was first addressed in 1976 by Protein Data Bank (PDB) by defining a data format for macromolecular structures, named PDB format [58, 59]. International Union of Crystallography (IUCr) followed the suit in 1990 by defining the Crystallographic Information Framework/Format (CIF) [60]. In 1999 yet another data format, Chemical Markup Language (CML) [61] was developed. Based on XML, the CML allowed the direct inclusion of chemical and crystallographic data in XML documents as well as the analysis of this data using many XML-oriented tools.

The PDB format files are human-readable and consist of fixed-width lines, each identified by prefixes of up to six characters long. These features made the data straightforward to browse and read. However, with the increase of protein size the PDB format became too restrictive and was replaced by PDBx/mmCIF, which aimed to retain the best features of the PDB and CIF formats.

During a quarter of a century of its existence, CIF has been widely adopted and used as a standard by most of crystallographic journals as well as structural databases (ICSD, CSD, CRYSTMET and COD) [50]. New CIF dictionaries have been developed with the aim of unambiguously defining ontologies in order to uniformly present data in various fields of crystallography, with notable examples including macromolecular crystallography [64], powder diffraction [65], and electron density studies [66]. It is assumed that main reasons for the popularity of CIF format are the use of human-readable text, a relatively simple syntax, extensibility, continued support by the IUCr and an increasing availability of software for CIF processing [67]. As probably all other formats, CIF is sometimes criticised for strictness of its syntax, as parsing failures caused by minor mistakes are usually difficult to trace by consulting parser error messages only. Blake (2009) lists missed termination marks of text strings and text blocks among the most common syntax mistakes [8, p. 319–326]. A wide variety of software tools have been developed for reading, writing, validating, manipulating and visualising CIF files [68]. The unprecedented development in the field of in silico materials simulation initiated the emergence of high-level software suites for materials analysis, such as `AiiDA` [69], `ASE` [70] and `pymatgen` [71], which support structural data input/output in the CIF format. In 2016 a second version of CIF format, named CIF 2.0, was announced [72]. As version 2.0 of the format is currently in its early adoption stage, this study concentrates exclusively on CIF 1.1 format, except in passages with explicit indications.

CIF format provides a well-defined framework for reporting details of all the processes during the crystal structure determination from the crystallisation and crystal preparation to the refinement. At some point it was suggested that CIF files containing text of the reporting article in IUCr-defined data items should become a standard means for article submissions, at least to the IUCr journals. However, current usage of CIF is primarily to report the conditions of the experiment, coordinates and in some cases the diffraction data. It must be noted, however, that crystallographic data is the primary aim of the CIF format: CIF files are produced by data collection and refinement software providing inputs and outputs of these processes, on top of whom the inferences concerning the chemistry of a compound under study are built. Therefore, reporting of neither precise connectivity nor systematic chemical name is enforced by the publishers, thus these data are usually not included in CIF files albeit it is obvious that both further chemical studies and independent validation would benefit if these data were required in CIF files [45, 50]. Geometric parameters in CIF reports are also optional. There is some sense in omitting these as they can always be derived from the coordinates. Furthermore, restrained parameters should not be mixed with parameters determined from just diffraction data alone [8, p. 319–326].

The CML was designed as an ontologically neutral markup language chiefly for the usage on the Web. Differently from PDB and CIF, CML was developed using XML – an already existing data carrier format, allowing to use variety of XML tools to query, transform and validate the crystal structure descriptions [61]. Recently an extension of CML for computational chemistry was developed [73].

2.4 Errors

Errors can happen in virtually any step of crystal structure determination and, if undetected, may subsequently cause incorrect conclusions in studies based on them [74]. An editorial of *Drug Discovery Today* in 2011 has stressed the need of government-funded data curation programs to improve public chemical resources on the Web to stop error proliferation, which happens as the data is cited and reused. Apart from government-funded programs, crowd-sourced efforts to validate public data with limited resources were also acknowledged. It was reported that as much as 10% of datasets used for quantitative structure activity relationship (QSAR) studies published in *Journal of Medicinal Chemistry and in QSAR and Combinatorial Science* have errors either in their chemical structures or biological activities (or both). Analysis of NIH Chemical Genomics Center's NPC browser, containing curated molecular structures of clinically approved drugs, identified fundamental errors in stereochemistry, valence and charge balance of some of the entries. A "screening data set" was found to have 5-10% flawed molecules. Furthermore, these errors can easily be spread [44] and, in case of automated analyses, lead to incorrect conclusions owing to the principle of "garbage in, garbage out", known since the works of Charles Babbage [75, p. 67]. It is of no surprise that more experienced researchers tend to produce crystal structures of higher quality [76]. However, independent quality control is usually harnessed to find and possibly correct the errors. Nevertheless, a significant number of crystal structures published in peer-review journals contain errors, meaning that neither the authors, nor reviewers or editors have spotted at least some symptoms of errors [11].

Significant portion of errors in crystal structures is due to incorrect assumptions made during the construction of an initial model. Incorrect identification of atom chemical types during the model building is credited as a common error [11]. A study concentrated on tautomers, compounds that are readily interconvertible by a movement of an atom (usually hydrogen) or a group of atoms between two sites of the molecular structure [77], has concluded that around 10% of the structures from the CSD contain incorrect tautomeric forms of molecules [78].

Choice of lower than true symmetry is quite common in crystal structure determination. In the most cases such misassignment makes structural refinement more difficult by leaving the parameters of symmetrically equivalent fragments to be refined independently. On the other hand, in some cases refinement in a false space group may lead to the assignment of an incorrect chemistry [79]. Baur and colleagues (1986, 1992) predict that around 3% of all published small molecule crystal studies may have been refined using lower than true symmetry. Authors notice that an inversion center is most often overlooked in space groups *C2/c* and *Pnma*. *Cc* is named the most often incorrectly ascribed symmetry space group with as much as 10% *Cc* crystal structures possibly belonging to higher true symmetry [80, 79].

Increasing automation of crystal structure determination process is in need of even more critical assessment of quality and reliability of the determined models. Although the automation is supposed to reduce the introduction of human errors in the process of model building, it may lead to an increase of errors should the automated means be used as “black boxes” with an ultimate faith in their outcome [30, 81]. Leaving out the human reasoning and intuition may be detrimental. It has been noticed that automated analyses of macromolecules also rarely pay any attention to the interpretation details of input models, expecting all of them to be equally correct [76]. Deller et al. (2015) concludes that as much as 12% of protein–ligand models in the PDB are only partially based on evidence (electron density) and should only be used after careful investigation [3]. Usage of such data without at least automatic filtering of problematic structures easily proliferates the errors.

Despite the fact that the most of numerical data in crystallography nowadays is generated and stored by computers (therefore called “born-digital”), non-negligible compendium of crystal structure descriptions have been produced in pre-CIF, moreover, pre-digital era, surviving to day mostly in printed form. Usage of such material is subject to possibility of typographical errors that may distort the meaning of the data. Allen and Taylor (2005) conclude that around 10% of typeset structures from pre-CIF era contain at least one numerical error [43]. In addition, transferring these descriptions to a digital form is complicated due to the amount of effort required and possibility of introducing more errors, be it human or optical character recognition introduced errors. Therefore, digitalisation must be coupled with a means of error detection and, if possible, correction.

Noteworthy source of incorrect structure reports is the push to report results, as summarised very well by the phrase “publish or perish”. Brown and Ramaswamy (2007) have noticed the trend for the most prestigious general science journals to publish crystal structures of much lower quality than they would be expected [76]. Significant increase of publications retracted due to fraud, error, plagiarism or duplicate publication during the 2000s is noted, especially in high impact factor journals [82], resulting in as much as 500-600 retractions each year [83]. This finding may signal either the greater scrutiny of their peer-review process, or researchers giving in to the incentive to desperately get their publications in prestigious journals for higher payoffs [82]. Recently fraudulent reports were detected both in the field of protein crystallography [84, 85] and small molecule structural studies [86]. The large portion of retractions due to fraud or suspected fraud (67% [82]) is troublesome, however, carefully crafted hoaxes can only be detected by replication attempts.

Terwilliger and Bricogne (2014) conclude that albeit locating model errors or inadequacies in protein crystal structures in the PDB is relatively easy as crystallographers usually stumble upon them, correcting issues that could not be remedied automatically is difficult as sociological factors come into play. As contributions to the database are done personally, deposited structures are usually regarded as one's own due to the efforts leading to the determination of the structure. Furthermore, successive studies of the same scientist/group might be based on the current interpretation of the structure. Shortage of motivation and funding to revisit previously published structures plays a substantial role in this process too. Correction of other researchers work might be easily perceived as criticism, invoking defensive behaviour of original contributor(s), possibly even unwillingly [87].

2.5 Validation

Validation of the determined models should be the final and crucial step of the crystal structure determination, and it should firstly be carried by the authors, then by the reviewers during the peer-review process prior to publication [10]. General public of readers may also be counted on for the post publication peer-review, although current trend to publish less and less raw data (coordinates, displacement and molecular geometry parameters) makes such validation difficult [8, p. 299–317]. From the viewpoint of the scientific method, validation is a continuous process of comparing the data against the ever evolving model, therefore, crystal structures should be analysed with the newest validation tools even years after their determination [87]. In fact, replication of results or the comparison of redundant albeit independently achieved observations could be employed as a method of validation. For example, dihedral angles are in practice rarely restrained during the refinement, therefore, they may be used for geometry checks [35]. Availability of electron density maps makes easier to distinguish genuine unusual features from model building errors or artefacts [30]. Availability of structure factor files contributed greatly in the confirmation and subsequent retraction of over 70 fraudulent crystal structures, unwittingly published in the IUCr journals [86]. There has been a lot of discussions about the need to store and share raw data of macromolecular structure determinations for the subsequent redeterminations once methods and software improve (see for example Terwilliger & Bricogne, 2014 [87] and Helliwell et al., 2017 [81]). However, Kleywegt and Jones (2002) conclude that low quality of structures (particularly signaled by high R_{free} values) discourage the authors to publish the experimental data which the model is expected to explain [30]. A very reasonable means of validation is the analysis of chemistry of determined crystal structures. Authors usually provide their chemical interpretation of structure models in a form of formulae and diagrams. Day et al. have analysed several thousand structures from *Acta Crystallographica Section E* by comparing systematic chemical names and author provided structural diagrams with the coordinates. Almost in no cases mismatches between the crystallographic and chemical data were detected [50]. We have performed a comparison between manually constructed SMILES descriptors [88] with the ones derived from author provided systematic chemical names [21]. Over 60% of entries were found to have identical descriptors whereas almost 14% contained mismatches that could not be explained by different conventions of SMILES generation used by manual construction and *OPSIN* [89], a software tool to convert chemical names to SMILES descriptors.

Programmatic tools, which are much more abundant for the validation of proteins than for other types of molecules [6], can be harnessed to inspect large amounts of crystal structure reports automatically. Some of these tools are overviewed in Section 2.8. Validation software usually follows the Bayesian method to assess the quality of a crystal structure model: structure's properties are compared with "known" properties of similar structures. This method requires the reviewer to correctly identify the known properties and their values. Evidently, such method highlights "wrong" structures, but is unable to answer whether a structure is "correct" [28]. Theoretical calculations based on experimentally determined crystal structures are regarded as a means to tell novel features from erroneous aberrations even in the absence of reports of similar findings in the scientific literature. The downside of theoretical approach is the amount of resources required for calculation of even small molecules (approximately 800 hours on a single 1 GHz Opteron processor for an average organic crystal structure from *Acta Crystallographica Section E*) [90]. Therefore, fully automated analysis of the correctness of a crystal structure is not possible to achieve. On the other hand, computer programs could be used to detect and report every unusual feature of the structure under study, leaving throughout investigation to the author or a referee [10]. An example of such program is `PLATON`, reviewed in Section 2.8.1.

Many parameters of a crystal structure determination report can be consulted during the quality assessment, both manually and programmatically, the most obvious being the R factor values. However, there are many ways to manipulate this and similar criteria [9]. A lower limit for the ratio of observation count to parameter number is postulated by the IUCr as a guideline, suggesting that the ratio of ten or more observations per parameter (in some cases lowered to eight due to smaller number of independent reflections) reduces the possibility of publishing fundamentally wrong structure significantly [36]. Unusual atomic displacement parameters (ADPs) are usually regarded as warning signs of various systematic errors in data, atom misassignment, inappropriate model building and refinement, as ADPs are easier affected by these deficiencies than the coordinates [11, 8, p. 205–219].

2.5.1 Geometric checks

Various errors in crystal structures, besides already reviewed quality criteria, often manifest in unusual geometry and interactions [79, 11]. Investigation of "suspicious" geometry has led to the discovery of aforementioned fraudulent crystal structures in the PDB as well as published in the IUCr journals [84, 86]. One of the first knowledge-based geometric check for the correctness of macromolecular models was the Ramachandran plot [91], developed in the early 1960s. The plot was an attempt to organise theoretical knowledge about likely and unlikely conformations of adjacent amino acids in the protein chain [92]. The Ramachandran plot has been used widely ever since in both protein crystallography and structure prediction. However, an alternative to the Ramachandran plot for small molecule geometry is far much more difficult to devise, mainly due to the many possible chemical environments and their influence on the molecular geometry [29]. Nevertheless, there have been many attempts to construct knowledge databases and tools for detection of unusual geometric features in small molecule crystals (reviewed in Section 2.8).

It is a well known fact that incorrect local geometry is rarely plausible in small molecule crystals [28]. In their study of ligands in protein structures Liebeschuetz et al. (2012) conclude that portion of unusual dihedral angles of ligands deemed to be poorly refined are over 20% [2]. Besides, unusual bond lengths may signal incorrect cell dimensions of small molecule crystals, low quality of diffraction data, inappropriate refinement or unresolved disorder. “Bumps”, unusually short contacts between non-interacting atoms (shorter than the sum of their van der Waals radii), are also very informative, suggesting either missing interactions or errors in atom positions [10].

A simple and widely used visual aid for the detection of outliers – a histogram – is sometimes also used by software. Liebeschuetz et al. (2012) demonstrates the usage of histogram-derived dihedral angle frequency ratio in Gibbs free energy (ΔG) calculation:

$$\Delta G = -RT \log(F_{\text{max}}/F_{\text{query}}), \tag{2.3}$$

where F_{max} is the size of the most populous histogram bin and F_{query} is the size of the histogram bin in which the observed value falls. The smaller the ΔG, the less favourable the dihedral angle is [2]. Nevertheless, histogram-based methods for outlier detection are known to be highly sensitive to the choice of bin width and end points, all of them arbitrary. Kernel density estimation is often used to remedy these disadvantages, however the problem of parametrisation of the kernel functions persists [93]. Much less sensitive to the initial assumptions is the Z score, inspired by the so-called three-sigma rule in normally distributed populations. Z score provides the number of multiples of the standard deviation for each data point x_j:

$$Z_j = \frac{x_j - x}{\sigma}, \tag{2.4}$$

where x is the target value and σ is the standard deviation of the distribution. For multiple values, root mean squared Z score (RMSZ) is used:

$$RMSZ = \sqrt{\frac{1}{N}\sum_{j=1}^{N} Z_j^2}. \tag{2.5}$$

For high resolution (<1 Å) small molecule crystal determinations, Deller and Rupp give RMSZ values of 0.02 Å for bond lengths and 2.0° for bond angles. Individual outliers of $RMSZ > 5$ are included in PDB validation reports as highly unusual [3].

Studies report abundance of unusual ligand stereochemistry in protein–ligand complexes in the PDB, mostly attributing them to incorrect usage of restraints. Liebeschuetz et al. (2012) estimated that 70% then recently determined structures of complexes contained geometric errors, and that these errors could have been averted by using better restraint libraries [6, 2]. Particular care should be taken to distinguish unusual stereochemistry arising due to poor restraint libraries and due to the fact that ligands in protein binding pockets do exhibit slightly unfavoured conformations, as discussed before [2].

2.5.2 Voids

Due to the laws of physics, voids or empty spaces are very unfavourable in crystal structures, with a notable exception of fullerenes, which contain vacuum bubble of 4 Å diameter, which is not accessible from the outside of the molecule. Therefore, in most cases voids in crystal structures suggest either errors or omissions. However, highly disordered solvent, especially in macromolecular crystals, is impossible to model by the standard approach of assigning discrete positions. Thus the molecules of the solvent are either left out or, as in the case of `PLATON` SQUEEZE method [94], marked as displaced within defined regions. Nevertheless, detection of voids in small molecule crystal structures is used in validation protocols. Most of current methods for void detection employ sampling of discreet grid points across the unit cell, trying to fit a probe of 1.2 Å radius (approximate solvent accessible void space for a water molecule) between van der Waals surfaces of molecules in the crystal [95, 96]. Whereas discreet grid approach might overlook some voids, small grid steps (approximately 0.2 Å, as given by [95]) reduce such possibility significantly at the price of more intense computations.

2.6 Building geometry libraries

Usually, the generation of geometry library and organisation of its knowledge proceeds as follows: at first, so-called "atom types" are defined, observations of geometry involving atoms of these types are extracted, then pooled together into "classes" (for example, a bond class is defined by the pair of types of atoms which are involved in the bond) and analysed [35, 7, 4]. This technique is present since *CHARMM* library for peptide geometry, which used a set of about 30 distinct atom types [97]. In this set, for example, there are six distinct atom types for oxygen: `O` for carbonyl, `OC` for carboxy oxygen, `OH1` and `OH2` for oxygen with one and two covalently bound hydrogens respectively, and `OH1E` and `OH2E` for one and two "attached" hydrogens respectively, meaning that hydrogens are not treated as separate atoms, but as part of "augmented" oxygen atom. Concerning the huge variety of small molecules, atom typing techniques are applied in fuzzy matching manner to treat atoms with similar chemical environments as the same in all compounds they occur in. Usually matching is performed with the regard to the chemical element type, hybridisation state, charge and the types of attached atoms [5]. Fuzzy matching was applied by Engh and Huber to the data in the CSD two decades after the publication of the *CHARMM* library. The authors have partitioned *CHARMM* set of atom types into an even finer set, having recognised that large errors were caused by the use of too few atom types and that partitioning was necessary to reduce the standard deviation of bond length and angle samples [98]. In 2008 Andrejašič et al. attempted to supplement Engh and Huber's library with atom types from the so-called "hetero" compounds – small molecules, found in complexes with biomacromolecules – and have extended the number of atom types to nearly 2000. To achieve that the authors have developed automatic tools for the detection of atom types and collection of geometric parameters. However, increase of the number of atom types may result in underrepresentation of certain classes of observations. Andrejašič et al. noted that only a small portion of the parameters, namely 2.2% of bonds, 0.7% of bond and 0.4% of dihedral angles were really accurately described. Authors conclude that at least 30 observations are required for a geometric parameter to be statistically reliable [7]. In their research, Liebeschuetz et al. lowered this threshold to 5 observations for bonds lengths and bond angles, and 15 for torsional angles [2]. For the underrepresented classes, theoretical simulations could be used to calculate the missing parameters [4].

An alternative to atom typing method is the so-called monomer approach, particularly aimed at biomacromolecules. This method exploits the fact that these molecules consist of repeating units (amino acids in proteins, nucleotides in DNA and RNA). Vagin et al. (2004) reported construction of monomer library consisting of 2000 distinct monomers that can in turn be linked and/or modified in a number of described ways. This library is available for use with maximum likelihood refinement program `REFMAC5`, also equipped with atom type-based library with around 200 atom types [34]. Application of the monomer approach for ligands is difficult due to their wide variety, although it is employed for the most common molecules. Descriptors for the ligand molecules are usually generated using a graph-based approach: all atoms of a molecule are enumerated in a defined order ("linearised"), retaining information about cyclic edges and other chemically or geometrically important features. Some of the most widely used such methods are SMILES [88] or SYBYL line notation (also known as SLN) [99]. SMILES notation is widely used despite being based on VBT, which is difficult to extend outside the organic domain of chemistry [21]. SYBYL line notation is an extension of SMILES devised to overcome the most of SMILES deficiencies [99]. SMILES notation was first employed for automatic description generation in `PRODRG` [56] in 1996.

It is a common practice to use the sum of least squares for the minimisation in the refinement [97, 98, 7]. However, such approach requires all parameters to actually have a single optimal value and be distributed according to the normal distribution. Andrejašič et al. have acknowledged the existence of both multimodal and asymmetric distributions of geometric parameters that clearly would not be suitable for least squares minimisation and have attributed such distributions either to short-sightedness of the atom type assignment or unreliable data [7]. At the same time the single optimal value approach was challenged in the field of protein refinement [100]. Subsequent studies have confirmed the presence of distributions with heavy tails and gross outliers [45]. Another problem is posed by distributions of dihedral angles that can contain a number of distinct peaks [2], often located periodically due to the symmetry (for example, immobilised methyl groups) or almost randomly due to possible free rotation around bonds [1]. Andrejašič et al. have described distributions of dihedral angles using histograms. The authors defined all freely rotatable angles as periodic with a single ideal value, although concluding that such representation was suboptimal [7]. It is true, however, that peaks of most of parameter distributions could be approximated by Gaussian distributions in the vicinity of a peak maxima, but such approximation might result in great loss of information concerning the shape of the original distribution.

Constructed geometry libraries should be subjected to scrutiny before further applications.

Possible checks are based on manual inspection [4]. Kleywegt (2007) advises refinement of randomised set of coordinates against the library without the use of experimental data. Should the refinement arrive at chemically infeasible geometry, the library would be deemed incomplete, erroneous or inconsistent [5]. Andrejašič et al. have reported that distributions of bond lengths involving hydrogens usually display a few sharp, well separated peaks, reasoning that they originate from the usage of restraints during their refinement. The authors then have removed non-neutron derived observations to discover single-peaked and narrower distribution with values 0.1 Å larger than averaged through the initial sample [7]. This observation comes to show that despite the fact that neutron-derived bond lengths may reflect reality better, they should be applied for refinement or validation of neutron experiments.

2.7 Statistical methods

2.7.1 Distributions

As described in Section 2.6, geometric parameters are usually described as representing a single ideal value with errors of unknown origin and are assumed to follow the normal distribution. However, deviations from normal law are observed, suggesting for the search of more suitable statistical distributions. It is clear that histograms should not be used instead, as they are very sensitive to the selection of range and number of bins. Moreover, they are discrete, while continuous models are usually preferred. Although similar density estimation methods, such as the smoothing spline, might seem to be a good alternative, they are also sensitive to initial assumptions which should both have convincing theoretical properties and perform well in practice [93]. Instead, mixture models could be applied for the description of other than normal distributions. Generally, a mixture model is a sum of any number of statistical distributions (called components of the mixture) scaled so as to maintain the integral of whole mixture equal to 1:

$$F(x) = \sum_i a_i f_i(x), \tag{2.6}$$

where $F(x)$ is density of the mixture, $f_i(x)$ – density of its ith component and a_i is the mixture proportion of ith component. If for each component i

$$\int f_i(x)dx = 1, \tag{2.7}$$

what is the case with the densities of all statistical distributions, it is enough to require that the sum of mixture proportions is 1:

$$\sum_i a_i = 1. \tag{2.8}$$

A class of algorithms, called expectation maximisation (EM) algorithms [101], iteratively selects the parameters maximising the likelihood of the population. Resulting parameters are hence called “maximum likelihood estimates”. EM algorithms exist for mixture models, most importantly, for normal mixture models, often used in cluster analysis. Normal mixture models could also be employed to describe multimodal distributions of bond lengths and bond angles. Distributions of dihedral angles, nevertheless, are poorly modelled by the normal distribution due to the requirement to choose a cutting point in an otherwise circular range. A counterpart of normal distribution for circular data is the circular normal distribution, better known as von Mises distribution (see Section 3.7.2 for probability density function). This distribution is rarely used in chemical literature despite the existence of an EM algorithm for its parameter selection [93]. A question on how many normal components should be used in such mixtures per sample can be answered by constructing a set of models with $1...M$ components via EM and choosing the best of them using some criterion. Model selection criterion should prevent overfitting: allowing the data to select a model will almost always result in the conclusion that a mixture of $n+1$ components fits the data better than n components. Overcoming this deficiency are Akaike information criterion (AIC [102]) and Bayesian information criterion (BIC [103]), both acting as Occam's razor and achieving perceivably the best fit with the least model parameters [104]. It is held that in general both AIC and BIC tend to favour models with more parameters as the sample size increases, arriving at overparametrised models for larger samples [105, 104, 106]. On the other hand, BIC has a tendency to oversimplify models of smaller sample sizes [104]. In the field of cluster analysis, every mixture component is usually perceived as originating from a separate cluster, what is not always the case [106]. In essence, EM for mixture models prefers flexible models, capable to accommodate the non-normality of the data indifferent to its interpretation. Therefore, the presence of n components in the mixture model perceived as the best does not necessarily mean that each of them stands for a separate chemical of physical property [104].

Mixture models, derived from the data using EM, constitute an elegant Bayesian framework to treat accumulating observations: at some point number of observations not represented by the model (outliers) reaches critical point and is "recognised" by the algorithm as deserving a separate mixture component, by so becoming a part of the "theory". Therefore, putative outliers in the EM input have to be checked per-case prior to the removal to scrutinise their authenticity. Otherwise the exclusion of genuine unrepresented observations might distort the resulting model [104].

2.7.2 Outlier detection

There are many theoretical methods of outlier treatment, however, none of them is unanimously accepted. The most common of them is based on Z score (described in Section 2.5.1) and requires the data to follow a normal distribution. A more flexible is Bayesian theory-backed method to compute the odds that an observation was sampled from the population rather than from the distribution of outliers [16]. Along these lines Sain et al. (1999) have demonstrated a successful outlier detection technique based on ratio of likelihoods that an observation under study is sampled from either the population or outlier distribution. The authors used EM for mixture parametrisation and AIC for the selection of number of components. However, they conclude that EM tends to dedicate a component for numerous identical or close outliers in the training sample, consequently ceasing to consider this part of population as outliers [104]. On the other hand, given no additional information it is hard to tell the observations belonging to the population from outliers. Nevertheless, such eventual "transformation" of a group of outliers to a part of the "rule" fits very well with the scientific method. In the course of evidence collection, a certain group of events arises, which is not explained well by the current hypothesis. Therefore, if allowed by Occam's razor, the hypothesis is extended to accommodate the newly observed feature. Thus the initial hypotheses about stereochemistry should be modified by the introduction of new observations, otherwise there is no need for further collection of the evidence [28].

2.8 Existing libraries and tools

2.8.1 PLATON

`PLATON` was developed in 1980 as a program for the automatic calculation of stereochemical parameters for structures refined with `SHELX76`, including bond lengths, bond and dihedral angles. Eventually the program was improved to include and evaluate even more stereochemical parameters, such as unusual or forbidden contacts and voids [10, 107, 108]. `PLATON` attempts to deduce hybridisations from the connectivity information reporting every failure as it may help locating missing atoms in the model. Many other heuristics are applied to identify errors, most of them based on features that rarely occur naturally: isolated oxygen atoms (most likely water molecules missing hydrogens), isolated hydrogen atoms (possibly incorrect coordinates) and many more [10]. Checks of `PLATON` are incorporated into `checkCIF`[7] Web service by the IUCr, which is employed in validation of crystal structures prior to their publication. Source of `PLATON` is open and the program is free for academic usage.

2.8.2 Engh & Huber, 1991

Library of stereochemical parameters for the refinement of protein structures by Engh and Huber (1991) was derived from small molecule crystal structures in the CSD (thus it is also referred to as CSD-X) and is still widely used today [38, 4]. The authors analysed the stereochemistry of molecule parts equivalent to protein backbone and amino acids. At that time, the CSD contained around 100 000 entries [12]. In their study, the authors have supplemented the set of *CHARMM* atom types with 14 novel atom types, arguing that such additions were necessary when a distinction between parameters of two different chemical fragments was apparent [98]. Parameter values were given for 59 distinct types of bonds and 108 types of angles. Over 15 years later Jaskolski et al. (2007) analysed the contents of the CSD and near-atomic resolution structures from the PDB, concluding that ideal stereochemical values, as reported by Engh & Huber, required only minor adjustments [38]. It was, however, noticed that protein backbone bond length between carbon and nitrogen depends on the type of amino acid residue that is attached to the carbon [109]. Subsequent discussions have arrived at a conclusion that the angle of protein backbone (N–C$_\alpha$–C, so-called τ angle) is influenced by more factors than it was thought initially and that all of them should be accounted for [100, 109]. Cole et al. (2017)

[7] `http://checkcif.iucr.org`

proposed a constantly updated library of protein-relevant restraints, automatically generated from small molecule structures containing small peptides, as opposed to amino acids only. Such inclusion of ligated amino acids should improve the quality of protein backbone-related parameters [12].

2.8.3 MIMUMBA

MIMUMBA, a generator of biologically relevant conformations of ligands, was introduced in 1994 by Klebe and Mietzner [110]. The authors have employed atom typing technique similar to SYBYL line notation to query the CSD. Structures with $R < 0.1$ were analysed, ring and open-chain fragments were treated separately. Fused rings (rings that share bonds) were split into smallest possible rings of up to seven members. 216 distinct dihedral angle parameters that cover the most important fragments in typical organic molecules were identified and analysed. Probability distributions of dihedral angles were smoothed using fifth order polynomial spline and converted into empirical potentials by the approach of Murray-Rust [111]. The authors argue that the generated conformations have some probability to resemble geometries that ligands adopt at the protein binding sites [110].

2.8.4 PRODRG

PRODRG is a program developed to recognise ligands from their 3D coordinates using SMILES-like descriptor strings, called "MOLDES". Descriptors include topology information of whole ligands. PRODRG is also capable of generating 3D coordinates for input descriptors. MOLDES descriptors encode atom chemical types by arbitrary integer codes, therefore, they are less human-readable than SMILES and include only chemical types most usually found in ligands [56].

2.8.5 Mogul

Mogul is a library of small molecule geometry, derived from observations from over 800 000 crystal structures in the CSD [12]. The library describes most often observed bond lengths, bond angles and acyclic dihedral angles as well as preferred conformations of ring systems. Atom typing approach is used to facilitate exact substructure search inside the library. The developers of Mogul have put the upper limit for large samples: prior to the analysis, bond lengths, bond and dihedral angles are reduced to 10 000 observations by random selection, whereas ring samples are reduced to 500. Such a limit could be useful, since storage and interpretation of large samples may be problematic [112]. Furthermore, Mogul does not include fragments with hydrogen atoms, as their positions may not be reliable (discussed in Section 3.4.4). Parameters of these fragments are summarised by a mean and standard deviation. Bond and dihedral angles containing metal atoms are also excluded as well as rings of less than five atoms [113]. Angle direction (chirality) of dihedral angles is not retained as all distributions are assumed to be symmetric around 0° [13]. Mogul is used to evaluate the geometry of input structures: Z scores are calculated for bond lengths and bond angles, distances from the nearest observations in the database are returned for dihedral angles. Functions for the detection of unusual geometry have been included in crystal structure solution program CRYSTALS, allowing also to use CSD-derived values as targets in its least-squares refinement [43]. Alongside Mogul, a complementary library IsoStar is developed for non-bonded interaction between parts of molecules. Program Isogen is used to produce geometry distributions for these observations [114]. Usage of Mogul, IsoStar and Isogen is limited to CSD/CCDC license holders, as is the case with all the other products of the CCDC.

2.8.6 VaLigURL

Kleywegt and Harris (2007) have presented VaLigURL, a Web server for comparison of user-supplied crystal structures with all analogous entries in the PDB. The server also enables studies of ligand conformational diversity and quality across the PDB. The authors suggest using VaLigURL as a means for the evaluation of sets of candidate models in searches for the best model, as models are judged by how common their geometries are. However, VaLigURL uses neither atom typing nor the monomer-based approach. Instead it relies upon atom labels, that are quite standard in macromolecular PDB files. The authors acknowledge that inconsistent or erroneous atom names in the input result in unrecognised atoms or high deviation from “regular” bond lengths and angle sizes [6].

2.8.7 PURY

In 2008 Andrejašič et al. reported the creation of PURY, an online database of geometric parameters of chemical compounds, derived from the CSD. PURY is comprised of lists of bond lengths, bond and dihedral angles, chirality, planarity and conformation parameters. A total of 1978 atom types were identified, participating in 32 702 bond, 237 068 bond angle and 201 860 dihedral angle classes. The authors emphasised that a vast part of these classes consisted of just a few observations. Nevertheless, for classes with a single observation that could not result in the derivation of standard deviations (important for the description of conformation spaces), they have assigned meaningful values of σ, taken to be compatible with the rest of parameter classes. However, due to the licensing policy of the CCDC being also applicable to the products derived from the CSD, PURY is only available to CSD/CCDC license holders [7].

2.8.8 CV

Taylor et al. (2014) have developed the CV system, which was used to construct geometry libraries for proteins, consisting of parameters for bond lengths, bond and dihedral angles. A geometry optimiser and a generator for conformational isomers (conformers) were also presented. The authors have stressed the importance of dihedral as well as bond angles to the overall molecular form. Bond lengths were judged to be less influential to it, albeit of widespread interest. Mogul was used as a starting point and was supplemented by taking into consideration chirality, three- and four-membered rings, fused rings and symmetry. Novelties (compared to Mogul) were introduced in dihedral angle analysis by including all single, double and aromatic bonds, both acyclic and cyclic. Distributions of dihedral angles span full 360° range and are not by design symmetric around 0°. Only structures with $R \leq 5\%$ were considered, solvate molecules were excluded. Lower and upper quartiles were used for the outlier detection. Large samples were reduced to 250-1000 observations by random sampling and the construction of the library was carried out automatically. Several libraries were constructed using atom types of decreasing degree of precision for searching in cascading fashion. The authors have concluded that correct handling of chirality by `CV` improved the results, as compared to `Mogul` [13].

2.8.9 CSD-KBF

`CSD-KBF`, a knowledge-based optimisator of organic molecules, was reported by Cole et al. (2016). Again, customised version of `Mogul` was used to analyse the data in the CSD to construct empirical force fields, in a manner similar to that of `CV`. Geometric distributions were smoothed and plugged in the objective function as terms concerning bond lengths, bond and dihedral angles, planarity and bumps. Terms for previously unseen fragments were based on input 3D coordinates, allowing a small degree of variation around values. In the conclusion the authors state that empirical force fields result in similar outcome as density functional theory (DFT) based conformational scan [113].

2.8.10 AceDRG

`AceDRG` is a program for automated derivation of geometrical information from the crystal structures of small molecules. Long et al., the authors of the program, have applied `AceDRG` to extract chemical knowledge from the COD and concluded that their results were in close agreement to those derived using `Mogul`, albeit both programs implement different algorithms. The authors have devised SMILES-like atom types and many methods for a posteriori cleaning up of the observations. Extreme outliers with Z score over 5 were removed, as well as whole molecules having either exactly the same or too different lengths for the same class of bonds. Bond classes having particularly small standard deviations were given special treatment due to the possible bias, caused by constrained refinement. Classes of less than 100 observations were ignored as not significant. Afterwards, tests for skewness, kurtosis and multimodality were performed to identify departures from the normal distribution. The authors have concluded that bond lengths are affected by even their third covalent neighbours [115].

Chapter 3

Methods and algorithms

3.1 Extraction of crystallographic data

The first step of the analysis of crystal structures (diagram of the whole workflow for building the geometric library is given in Figure 3.1) is the extraction of data from the CIF format files. It was noticed that minor deviations from both the CIF 1.1 syntax [67] and semantics [15] appear to be relatively common in the supplementary CIF files of the published articles. Some constraints of the format could be relaxed without any harm (allowing, for example, inclusion of Unicode code points past the single byte limit). Moreover, introduction of additional rules to the formal grammar could account for error-correcting heuristics, for example, detection of runaway closing quotes. The absence of such error-detecting and correcting features in the existing CIF parsers motivated us to develop our own parser, to which we refer to as `COD::CIF::Parser` [67]. In general, in the development of our parser we have followed the principle of robustness as formulated by Postel [116]: "be conservative in what you do, be liberal in what you accept from others". Therefore, `COD::CIF::Parser` is able to automatically fix the most common and the most obvious syntactic errors. Both tractable and intractable deficiencies are accurately reported, including their precise location and nature. Based on `COD::CIF::Parser` we have developed `cod-tools`[1] – a set of tools for manipulating the CIF files in the COD. The `cod-tools` package is successfully used in the automated COD data deposition pipeline and the validation of the COD data against the IUCr data validation guidelines[2]. The package has been intensively used and developed during the current research.

Conversion of the CIF files into internal data representations (parsing) is obviously of great importance to all CIF handling tools. It is important to stress that grammars of both CIF 1.1 and CIF 2.0 are context-free (type of CIF 1.1 grammar could be disputed due to the existence of rules showing the character of context-sensitive grammar), therefore regular expressions are not enough to parse CIF format.

A special kind of CIF handling software is general purpose parsers that are developed to serve

[1] Available under the GPL2 free software license at `svn://www.crystallography.net/cod-tools/tags/v2.0`, this study refers to version 2.0 (source revision 5425), which can be also obtained from `http://www.crystallography.net/archives/2017/software/cod-tools/cod-tools-2.0.tbz2`

[2] `ftp://ftp.iucr.org/pub/dvntests`

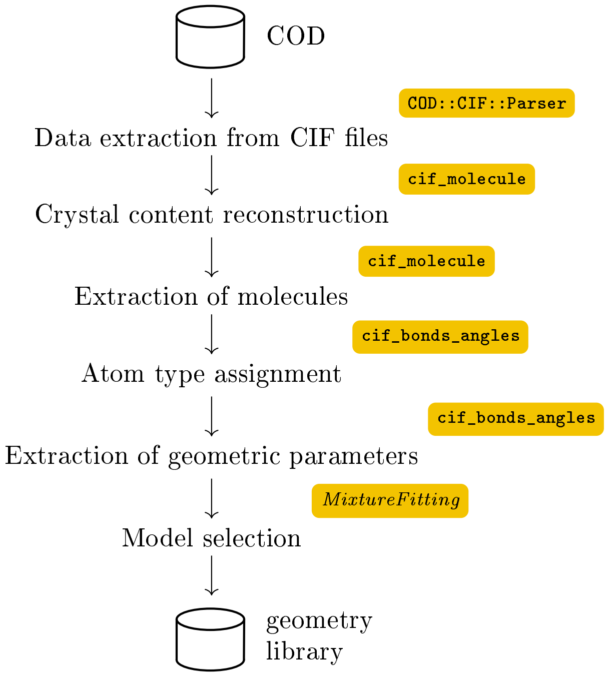


Figure 3.1: Diagram of the construction of geometry library from the data in the COD. Software used for each of the processes is shown in orange background.

as CIF reading libraries for other software tools. Examples of CIF 1.1 parsers include `vcif`[3] [117], `vcif2` (also known by the name of the executable file `cif2cbf`) [118] and `cif_api` [119] in `C` language, `gemmi` [120] and `ucif` [121] in `C++` language, `cif2cif` [122] in `Fortran` language, `ASE` [123] and `PyCIFRW` [124] in `Python` [125] language. Another noteworthy tool is the `ZINC` package [126], which provides a set of converters from CIF to its own ZINC format and allows convenient manipulation of data in the command line environment. Finally, since the syntax of CIF 1.1 is a subset of a more general STAR 1 [127] format, STAR parsers like `STAR::Parser` [128] in `Perl` [129] and `StarTools` [130] in `Java` could also be used to read CIF files.

Most of the parsers are well-suited for reading syntactically correct CIF 1.1 files, however, departures from the CIF standard occasionally occur in the supplementary material of the published articles. Such files trigger problems in processing, for example when screening or viewing. In the process of depositing supplementary CIF files to the COD, minor departures from the standard (missing quotes or data block headers, duplicated data names or forbidden characters, etc.) appeared to be common and too numerous to be remedied manually by the human editors. It was deemed too inefficient to require the CIF editors (usually volunteers contributing in their spare time) to fix these departures from the CIF syntax.

[3] `http://www.iucr.org/resources/cif/software/archived/vcif-1.2`

3.1.1 Programming tools

The COD::CIF::Parser was implemented in both Perl and C in parallel due to convenience reasons. High-level Perl language, permitting concise formulation of algorithms and rich in text processing features such as native support for enhanced regular expressions, has been chosen for COD::CIF::Parser for CIF 1.1 initially. Long history of consistent Perl development and a wide community of users and developers helped us gain experience in using this programming language. Large number of community-developed Perl libraries in the Comprehensive Perl Archive Network (CPAN)[4] is at hand to supplement our own developments. COD::CIF::Parser, developed in Perl, is robust and fast with a performance comparable to those of other interpreted languages (see Section 4.1.3 for the comparison). Therefore, Perl was deemed suitable for our first CIF 1.1 parser prototype and continues to serve as a helper for future developments.

However, relatively low speed (in comparison to the compiled languages) of COD::CIF::Parser in Perl is the only drawback. To counter this we have reimplemented the CIF 1.1 parser in lower-level C language. Developing and maintaining COD::CIF::Parser in C requires considerably more effort than Perl code, nevertheless, COD::CIF::Parser in C retains its portability and could be linked by a wide range of high-level computer languages by the use of bindings. We have developed binding for Perl, allowing for drop-in replacement of Perl parser with its C counterpart. In order to reduce the otherwise doubled efforts in developing and maintaining parser code in two languages we have implemented CIF 2.0 parser in C only.

CIF format parsers were implemented using generators for bottom-up syntactic analysis parsers instead of writing them “by hand” (e.g., using recursive descent method). As the grammar rules for parser generator could be put concisely, explicitly and in a readable form in a single input file, fixing, updating and extending the language is easier. These features are especially important as the IUCr introduces further developments of the CIF format as well as we devise special error-correcting extensions. For Perl parsers the Yapp tool [131] is employed, while for C parsers the Bison parser generator is used [132]. These tools accept nearly identical input syntax based on well-known Yacc parser generator [133, 134], which is in turn based on somewhat simplified Backus-Naur Form (BNF) syntax. Since the CIF grammar is published in BNF-like notation [135, 136], correspondence between it and Yacc input is often straightforward.

Historically, CIF 1.1 parser in C was implemented by porting the Yapp grammar to the Bison input file and replacing Perl code by the corresponding C code. Binding for Perl and, in turn, Python, were generated using automatic binding generator SWIG [137]. Resulting parsers for CIF 1.1 adhere to the same CIF syntax and produce identical internal representation of parsed input files. However, due to the differences in the parser generators the error reporting slightly differs among the parsers in the different languages. Nevertheless, the strict syntax of error messages (see Section 3.1.4) and identical internal representations of the CIF files make the seamless substitution of the CIF parser in Perl with the binding of parser in C possible without disrupting the dependent software. The availability of C compilers, Perl ports (available in more than 100 computing platforms [138]) and target programming languages of SWIG (over 20 at the moment of writing) allow for relatively easy porting and linking of the COD::CIF::Parser.

[4] https://www.cpan.org/

Key	Value
name	Scalar. String denoting the name of a CIF data block.
tags	Array. Lower-cased data names present in the CIF data block.
values	Hash. Keys are equal to the values of the **tags** array. Values are arrays containing values for each data item.
types	Hash. Keys are equal to the values of the **tags** array. Values are arrays containing lexically derived data types for each data item.
precisions	Hash. Keys are equal to the values of the **tags** array. Values are arrays containing standard uncertainties for each data item.
loops	Array of arrays. Each inner array corresponds to a loop from the CIF data block and contains a list of data items present in the loop.
inloop	Hash. Keys are equal to the values of the **tags** array. Values correspond to indices of the outer **loops** array. It is used as an index to optimise data item-in-loop related searches.
save_blocks	Array of hashes. Contains the list of CIF save frames, where every frame is represented using a data structure, identical to a CIF data block.
cifversion	Hash. Has keys "major" and "minor", corresponding to the major and minor versions of CIF format, currently 1.1 or 2.0.

Table 3.1: Key-value pairs of a hash that represents a single CIF data block as constructed by `COD::CIF::Parser`

```
data_global
_journal_year                       1998
data_example
_cell_measurement_temperature       200.0(5)
_symmetry_space_group_name_Hall     '-P 1'
loop_
_space_group_symop_id
_symmetry_equiv_pos_as_xyz
1   x,y,z
2   -x,-y,-z
```

Figure 3.2: An example of a CIF input for parsing

3.1.2 Data structures

In the `Perl` language parser/binding a CIF file is internally represented by an array of `Perl` hashes, each of them representing a single CIF data block. Key-value pairs of the data block hash are shown in Table 3.1. For example, upon parsing of the CIF file from Fig. 3.2, a `Perl` data structure shown in Fig. 3.3 is constructed. The same data structure is retained in the `Python` binding (it should be noted that `Perl` arrays and hashes are straightforwardly represented by `Python` lists and dictionaries, accordingly).

Both textual and numeric CIF values are stored as strings. If present, standard uncertainties are preserved for numbers, just as found in the original CIF file. Therefore, numeric precision is not lost during the parsing. To ease the application of the uncertainties, they are provided in the `precisions` subhash. CIF comments are ignored by the parser, as they should not contain important machine-readable data and since the order in which they appear in input files might be difficult to reproduce if an application chooses to reorder data items in the parsed CIF. Out of order comments, in turn, may introduce false interpretation of (meta)data if used for inference. The same convention is found to be followed by `PyCIFRW`. Additional justification for

```
[
   #   0:
   {
      "cifversion" => {
         "major" => "1",
         "minor" => "1",
      },
      "name" => "global",
      "precisions" => {
         "_journal_year" => [ undef ],
      },
      "tags" => [ "_journal_year" ],
      "types" => {
         "_journal_year" => [ "INT" ],
      },
      "values" => {
         "_journal_year" => [ "1998" ],
      },
   },
   #   1:
   {
      "cifversion" => {
         "major" => "1",
         "minor" => "1",
      },
      "inloop" => {
         "_space_group_symop_id" => "0",
         "_symmetry_equiv_pos_as_xyz" => "0",
      },
      "loops" => [
         [
            "_space_group_symop_id",
            "_symmetry_equiv_pos_as_xyz",
         ],
      ],
      "name" => "example",
      "precisions" => {
         "_cell_measurement_temperature" => [ "0.5" ],
         "_space_group_symop_id" => [ undef, undef ],
      },
      "tags" => [
         "_cell_measurement_temperature",
         "_symmetry_space_group_name_hall",
         "_space_group_symop_id",
         "_symmetry_equiv_pos_as_xyz",
      ],
      "types" => {
         "_cell_measurement_temperature" => [ "FLOAT" ],
         "_space_group_symop_id" => [ "INT", "INT" ],
         "_symmetry_equiv_pos_as_xyz" => [ "UQSTRING", "UQSTRING" ],
         "_symmetry_space_group_name_hall" => [ "SQSTRING" ],
      },
      "values" => {
         "_cell_measurement_temperature" => [ "200.0(5)" ],
         "_space_group_symop_id" => [ "1", "2" ],
         "_symmetry_equiv_pos_as_xyz" => [ "x,y,z", "-x,-y,-z" ],
         "_symmetry_space_group_name_hall" => [ "-P 1" ],
      },
   },
],
```

Figure 3.3: An internal CIF data structure created by the `COD::CIF::Parser` after processing the CIF file from Figure 3.2

our decision is provided by the absence of language features to accommodate comments in such widely used data formats as JSON.

3.1.3 Error detection and correction

`COD::CIF::Parser` is developed with the abilities to find and report CIF syntax errors as well as apply heuristics to remedy the most common ones: insert missing `data_` headers and quotes, resolve multiple occurrences of the same data item and so on. The error detection is facilitated by the extended CIF grammar that recognises syntactically incorrect constructions. For example, all CIF values appearing before the header of the first CIF data block are detected as CIF data values and are ignored when parsed in the error correcting mode as malformed initial comment lines. The full list of detectable errors and their solutions is presented below. Each of the heuristics can be enabled or disabled via providing corresponding parser options (given in **bold**):

- stray CIF values before the first data block – ignored (**fix_data_header**);
- no `data_` header – ignored (**fix_data_header**);
- stray CIF values after the data block name – appended to the data block name (**fix_datablock_names**);
- duplicate data items – if all data items report the same value, duplicate items are skipped (**fix_duplicate_tags_with_same_values**). The error is not corrected if two data items with the same name have different values – this is not done in order to prevent incorrect interpretation of the input;
- items with duplicate data names, where only one data item contains a known value (i.e. a value that is not equal to a single question mark or a single period) – only the data item with the known value is retained (**fix_duplicate_tags_with_empty_values**);
- more than one value for a single non-loop data item – all values are taken as quoted (**fix_string_quotes**);
- unquoted strings starting with opening square bracket ([) – treated as single-quoted strings (**allow_uqstring_brackets**);
- ^Z symbols – removed (**fix_ctrl_z**);
- other non-ASCII symbols – these are encoded as `XHTML` character references [139] (**fix_non_ascii_symbols**);
- missing
  single or double closing quote – an appropriate quote is inserted at the end of the line (**fix_missing_closing_single_quote** and **fix_missing_closing_double_quote**).

All of the aforementioned heuristics can be enabled with the **fix_all** parser option. Reports concerning the errors and the performed changes are collected and either printed to the standard error channel (default behaviour) or collected and returned as an array (when **no_print** option is present). Message format, described in detail in Section 3.1.4, is designed to be both human- and machine-readable. In addition, the total number of errors is returned from the parser.

3.1.4 Error reporting

Most of the programs typically issue informational messages concerning the encountered problems during the processing of the input. Commonly, the addressee of such messages is the human user and the purpose of their content is to inform about the nature of an emerged problem and how to fix it. Therefore, the messages are usually written in informal language. As `cod-tools` package (and the most of our other software) is intended to be integrated into larger systems such as the COD data deposition server, we have strived to make diagnostic messages both human- and machine-readable. We have found that:

1. in the absence of error codes, text in error messages becomes the public API and their changes should be strongly discouraged (unless between major software versions);
2. strict, formal specification of the error message format is advantageous.

As the composition of program systems is common in Unix-type operating systems, our solution might be applicable outside both the `cod-tools` and the current study.

We have composed a formal grammar describing our error message format in the Extended BNF (`EBNF`) form [140]. The format can be readily adopted and used by the software authors for further development of formatters and parsers of the error messages. In order to ensure the completeness, unambiguity and correctness of the devised grammar, we have employed `EBNF` parser and analyser as a means of computer-aided verification of desired properties. To do so we have implemented simple `BNF` and `EBNF` parser *grammatiker*[5] by using `Grammatica` [141] parser generator. `Grammatica` is one of the parser generators that are capable of keeping the grammar and the processing code in the separate files in order to ease both the readability and reuse. The output files are generated in `Java`, thus the `EBNF` processors were developed in this language.

The grammar of error messages has to be initially converted into an input file for `Grammatica`, from which a parser in `Java` is generated. The initial step ensures that all grammar rules are properly defined, and further processing with `Grammatica` ensures its unambiguity and correctness. The generated parser may in turn be used to check the error message syntax against the initial error message format grammar.

Top-level rules of the error message syntax are given in Figure 3.4. The *progname* is the name of the program, which issued the message, the *filename* corresponds to the name of a file where the error was detected. In order to better localise the error, optional line and column numbers (*lineno* and *linepos*, accordingly) may be provided in parentheses immediately after the file name, as well as additional information: the CIF processing programs of `cod-tools` package output CIF data block name (*additional_position*). The message text (*message*) corresponds to human-readable problem description, which is optionally preceded by the level of severity (*status*):

[5] The `BNF` and `EBNF` parsers are available as a *grammatiker* package at `svn://saulius-grazulis.lt/grammatiker` and `https://github.com/sauliusg/grammatiker`

```
error_report = progname, ':', [spaces], [location], ':', [spaces],
               (status, ',', [spaces] | lowercase_word), [message],
               [ ':', [spaces], newline, { space, code_line } ],
               {newline};

location = file_position, [spaces], [ additional_position ] ;

file_position = filename, [spaces], [ file_line_column ] ;

file_line_column = '(', lineno, [ ',', linepos ], ')';
```

Figure 3.4: The top-level grammar rules defining error message syntax for `COD::CIF::Parser`

- ERROR – indicates unrecoverable situation, rendering the output (if any) of the program unusable;
- WARNING – indicates that the output of the program could in principle be processed further, although it may contain results that were not intended in the current situation and thus should be treated with additional care;
- NOTE – an informative message; such message may be dismissed and the processing should proceed as usual.

If appropriate, messages concerning syntax errors are followed by an excerpt of the original file (one or more *code_line*). The origin of the error is signaled by a caret symbol ("^"). A few examples conforming to our message grammar are provided in Figure 3.5.

As some symbols are used by the grammar as the delimiters of syntactic components, they must not appear in file names and the message texts. As is evident from the `EBNF` grammar, to be parsed correctly, file names must not contain colons (":") and parentheses, while message texts must not contain colons. Since these "forbidden" characters may be found in any of the aforementioned parts of the message, they have to be replaced by arbitrary escaping sequences. Despite the fact that the provided grammar does not define any particular escaping scheme leaving it to be defined by the application-level agreement, programs of `cod-tools` package use the `XHTML` character entity references, a compromise between the simplicity of the escaping/unescaping algorithms and the readability for the human user (an example of escaped file name is given in the last line of Figure 3.5). Therefore, any text can be encoded and placed in an error message without any loss of information, with the benefits of being both human- and machine-readable.

3.1.5 Writing CIF files

The construction of an initial file from the data structure of a parsed CIF file is straightforward, since the data structure contains all the required information. The algorithm to determine the appropriate CIF data type from a value is relatively easy: values with spaces are enclosed by quotes, and values for which quotes are not enough are put in CIF text fields. We have developed

```
cif_parse: stray.cif(2): ERROR, stray CIF values at the beginning of the input file.
cif_parse: noquote.cif(2,25) data_I: ERROR, incorrect CIF syntax:
 _journal_name_full Acta Crystallographica
                                         ^

cif_parse: loops.cif(14,1) data_I: ERROR, wrong number of elements in the loop starting at line 7:
 loop_
 ^

cif_parse: d.cif(2,6) data_I: ERROR, dollar symbol ('$') must not start an unquoted string:
 _tag $value
      ^

cif_parse: non&colon;existent.cif: ERROR, could not open file -- no such file or directory.
```

Figure 3.5: Examples of `COD::CIF::Parser` diagnostic messages

`COD::CIF::Tags::Print` Perl module of the `cod-tools` for writing CIF files. The parser and writer pair was shown to perform successful round-trips for its own written CIF files [67].

CIF format has a potential to be used as a universal data carrier for any data available as key-value pairs, in the manner similar to JSON, XML or YAML. This feature is already used by `SHELXL2014` [142] to embed its input and output files in CIF in order to store the providence information next to the data itself. However, embedding of any data in CIF text fields is limited by the restrictions on the character set, line length and their non-nestable nature (lines inside text field must not start with a semicolon, as such construction is used to mark the termination of a text field). The last two limitations were lifted by the introduction of the text field folding and prefixing protocols in CIF 2.0 [143, 72]. Although, strictly speaking, the line prefixing protocol is standardised only in CIF 2.0, we have implemented both folding and prefixing in `COD::CIF::Parser` and `COD::CIF::Tags::Print` as methods to circumvent the limitations of CIF 1.1, treating line prefixing as an application-level agreement. Indeed, such agreements could be used to bypass all limitations of the format as long as the reading applications are aware of these methods. On the other hand, prefix-unaware software should process such files correctly unless programmatic interpretation of prefixed text field contents is attempted. In such case failure is likely inevitable. Should the unprefixing or unfolding of read CIF values be undesired in `COD::CIF::Parser`, either or both functions can be disabled by using parser options **do_not_unprefix_text** and **do_not_unfold_text**, respectively.

Apart from the folding and prefixing, which are defined by the Committee for the Maintenance of the CIF Standard (COMCIFS), we have devised and implemented a couple more methods for effective evasion of the restrictions of the character set, present in both CIF 1.1 and CIF 2.0 versions [144]:

- **Numeric Character Reference** (NCR): used in `cod-tools` package (explained in Section 3.1.4) to escape non-ASCII and other context-dependent forbidden symbols. Sparse usage of NCRs in texts with preserves the readability;

- **Quoted-Printable** [145]: has the same properties as NCRs plus the line folding ability;

- **Base64** [145]: overcomes all the limitations of CIF by encoding the content in base 64 system using printable ASCII symbols; used only when the content is purely binary;

- **gzip+Base64** [145, 146]: same as Base64 with the compression.

The choice of the encoding can be made according to the requirements of readability and file size. It is evident that the gzip+Base64 encoding defines a stack of two layers: Base64-encoding of gzipped contents. In order to implement such stacks we have defined a set of `_tcod_content_encoding_*` CIF data items to describe encoding stacks of any complexity in a dedicated human- and machine-readable CIF loop.

3.2 Usage of crystallographic data

3.2.1 Data source

We have chosen the COD as a source for small molecule structures due to its open access nature and inclusion of the whole spectrum of small molecule structures. The COD provides many means to obtain its data. One of them is using `Subversion`, a version control system, allowing to access any version ("revision" in `Subversion` language) of the COD data at any time. This method was chosen as preferred, since it allows to pinpoint a specific immutable version of the data in the database. For the current study, revision 199925 was chosen, having 382 807 entries.

3.2.2 Data curation

Responding to the calls from the publishers and the community for the higher quality standards of publicly available crystallographic data, developers of the COD (including me) have implemented stricter checks for incoming data and automatic fixes/regularisations. The following automatic checks were implemented:

- Semantic validation of powder diffraction reports using powder diffraction CIF documents, as described by Toby et al. [65];
- Detection of incomplete symmetry operator lists. In addition to these data, CIF files usually contain symmetry space group symbols (Hermann–Mauguin and/or Hall) that must correspond. Failure to match these would signal a possible error.
- Semantic validation of supersymmetric structures. The IUCr has defined methods to describe symmetry in more than three dimensions and these descriptions could be checked.

3.2.3 Measurements and connectivity

CIF format defines data items for storing geometric measurements, namely `_geom_bond_*`, `_geom_angle_*` and `_geom_torsion_*`. Although these items are not mandatory, it was found that 321 762 of 382 807 COD entries (~85%) contain them. Nevertheless, there is no guarantee that these lists of measurements are complete and free of typographical mistakes. Therefore it is a common practice to independently calculate the geometric parameters from the coordinates and connectivity. CIF format also defines data items for the former, `_chemical_conn_bond_*`, albeit they are virtually never present in the published CIF files (only three occurrences were detected in the COD so far). Algorithmic detection of connectivity is widely used, our approach is detailed in Section 3.3.2. As CIF format uses fractional coordinate system, where the base is formed from the cell vectors, we convert atom coordinates to the orthogonal system as described in Parsons & Clegg (2009) [8, p. 205–219].

3.2.4 Detection of chemical species

Despite the fact that the core CIF dictionary defines a means to specify the chemical species for the observed atoms, it is often ignored or misused. The IUCr recommends to use the `_atom_site_type_symbol` data item that is designated just for this purpose. Alternatively, one or two letter chemical symbol can be prepended to atom labels (values of `_atom_site_label` data item). For example, `C11`, `Au` and `Pb*` labels would be used to specify carbon, gold and lead atoms accordingly, following this naming scheme. The latter approach seems to be preferred in practice, despite that it introduces a lot of ambiguities. First of all, it is unclear whether the author meant to use the labels for this purpose or that one simply forgot to include the `_atom_site_type_symbol` data item. Furthermore, detection of chemical species from a label is also ambiguous. Usually, it is sufficient to perceive the first one or two letters from the atom label as its chemical symbol (`/^([A-Za-z]{1,2})/` in regular expression form). However, this approach may fail with cases when labels contain some additional information, for example, `HO` and `HOH`, which are often used to indicate hydroxide and water molecules accordingly, would both be perceived as holmium (Ho). Labels often used for water molecules (`Wat`, `W` and `Ow`) showcase the other flaws of such simplistic approach. The maintainers of the COD have adopted a practice of manually adding the chemical species as values of `_atom_site_type_symbol` data items (if none given) thus removing any ambiguity. However, this is not yet done automatically for reasons mentioned earlier.

For this study the simplistic approach for the detection of chemical species is employed. If present, values of `_atom_site_type_symbol` are preferred, hoping that in the long run all ambiguous atom labels will be resolved by the curators of the COD. As usual, structures having synthetic chemical elements are excluded from the consideration [147] starting with darmstadtium. Deuterium is also deemed intractable in this study.

3.2.5 Multiple occupation of sites

Atom sites that are modelled as mixtures of two or more different chemical types are commonly represented in CIF by multiple entries of `_atom_site_*` loop, identified by (almost) identical coordinates. For example, grunerite structure in COD entry 9000000 (as of revision 176465) contains four iron-magnesium sites. Eight `_atom_site_*` entries are used to describe this property. Collated by their coordinates these entries result in occupancy values summing up to 1 at each of the sites. Four Br/Cl sites are as well found in COD entry 2218544 (Figure 3.8, **e** and **f**). To present downstream applications with semantically connected `_atom_site_*` entries of such sites, we have adopted a practice to mark such sites as alternative using CIF format's `_atom_site_disorder_*` data items if the sum of their occupancies $\sum o$ maintains the inequality $|1 - \sum o| \leq 0.1$. Instead of modifying all COD CIF files, however, we apply this convention on-the-fly via command line tool `cif_mark_disorder` from `cod-tools` package. The approach proved itself handy since as much as 6% of the COD CIF files had been shown to contain unmarked multiply occupied atom sites. Nevertheless, vast majority (around 90%) of such

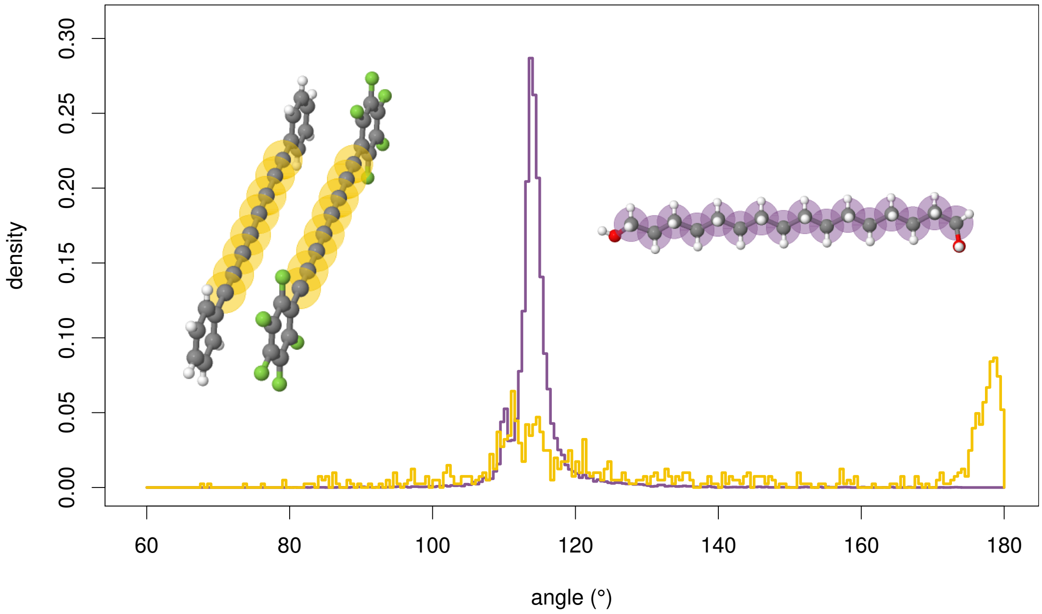


Figure 3.6: Angle density histogram of polyyne carbon (atom type `C(CC)2`, yellow) and alkane carbon (atom type `C(CCHH)2(H)2`, violet) fragments from the COD. While normally polyynes are linear compounds (C–C–C angle is 180°), this distribution of angles is distorted by the contamination with observations from saturated alkane chains (peak at around 107°) without both explicitly and implicitly modelled hydrogens. 3D visualisations of molecular structures for this and the rest of illustrations are produced with *Jmol* [148].

structures did not produce geometric measurements, largely (∼60%) due to disorder-dependent connectivity (described in detail in Section 3.3.5).

It is possible to go even further and employ an algorithm to check all the atoms of incomplete occupancy for possible assemblies, as it was done by Bruno et al. [45]. However, the number of potential variants tends to grow exponentially. Moreover, it is easy to misassign atoms to disorder groups, thus we argue that automatic detection of disorder should better be coupled with human supervision.

3.2.6 Implicit hydrogen atoms

Low resolution crystallography extracts very little to no information about the locations of hydrogen atoms in crystal structures, mostly due to tiny contributions of these atoms to the diffraction patterns. In studies concerning positions of heavy atoms only, hydrogen atoms are usually completely eliminated from the determined crystal structures. Nevertheless, there are many methods for hydrogen position treatment even when little structural information is available about them, for example restraints and geometric prediction, just to name two of them. However, employed hydrogen treatment methods and possible locations are of great importance for chemical interpretation of crystal structures and subsequent structure-based studies. Usually, simple heuristics are applied to detect counts and attachment sites of hydrogen atoms. For example, isolated oxygen atoms in crystal structures are generally treated as water molecules missing attached hydrogen atoms [10]. However, they also may stand for hydronium (H_3O^+) or hydroxide (OH^-) ions, as CSD 5.32 contained 954 and 317 entries with such compounds, accordingly [45]. To eliminate ambiguities CIF standard defines data item `_atom_site_attached_hydrogens` to indicate a known number of hydrogen atoms attached to a known site, albeit not modelled. On the other hand, no recommended notation exists for a known number of hydrogen atoms whose attachment sites are unknown. A widespread tendency is to introduce dummy atoms (marked as such via `_atom_site_calc_flag` data item) with coordinates $(-1., -1., -1.)$ and either occupancies or attached hydrogens numbers giving the quantity. Such behaviour might disrupt programs that are not prepared for interpretation of `_atom_site_calc_flag` or special treatment of $(-1., -1., -1.)$ coordinates. We have decided to put the total number of implicit hydrogen atoms without known attachment sites as a value of `_atom_site_attached_hydrogens` of a dummy atom with unknown coordinates (all equal to "."; see COD entry 2000135 for example) manually during the curation of the COD as a means to unify their representation throughout the COD.

Although observations from fragments with missing hydrogens were found to seriously contaminate the otherwise scientifically sound distributions (see Figure 3.6 for example), we have made no attempt to automatically detect and exclude them judging by geometry only. Although this would have resulted in construction of "correct" models, the aim of the current research is to describe the current geometry in the COD thus being suited to "accept" data with features similar to those already existing in the database. Therefore, further structures with missing hydrogens would not be treated as unusual, since in a sense they do not contain anything previously unseen.

3.2.7 Missing atoms

A wider class of crystal structure features is related to missing parts of the structure, that range from disjoint atoms to whole moieties. Usually, small solvent molecules are excluded from crystal structure descriptions, as their electron density is too widely distributed to allow localisation of individual atoms, even disordered. Missing atoms seriously affect crystal property calculations, such as electronic band structures. Therefore, such studies tend to exclude crystal structures, whose declared summary chemical formulae do not correspond to ones calculated from their unit cell [53]. However, internal solvent geometry is rarely of interest for the studies of stereochemistry. Thus we also have not employed any heuristics to exclude potentially incomplete structures from this study, but we have performed scans for voids in the COD as a part of the database curation effort.

For the detection of voids in crystal structures in the COD we have developed a program `cif_voids`[6], which reads in CIF files, reconstructs atoms in *P 1* (see Section 3.3.1) and tries to fit spheres between van der Waals surfaces of the crystal. The empty sphere lookup is carried out using `voronota` [149], a tool intended for Voronoi tessellation construction for molecular models. `cif_voids` reports all empty spheres ("voids") with radii larger than some arbitrary R_{min}, either modelled as regions of disordered solvent, or originating from errors.

[6] Available under the GPL2 free software license at `svn://saulius-grazulis.lt/crystalvoids/trunk`, this study refers to source revision 64.

Since 1988 PLATON's SQUEEZE method is used to model disordered solvent during molecular refinement. Program's reports about modelled voids are usually appended to the resulting CIF files [94], thus they can be consulted in order to tell "natural" voids from unintentional. In revision 199925 the COD contained around 12 000 structures with PLATON's SQUEEZE reports appended in `_platon_squeeze_*` CIF data items.

3.3 Crystal reconstruction

3.3.1 *P 1* reconstruction

We have investigated two approaches to the crystal reconstruction. In the first approach the asymmetric unit of the crystal is expanded to the *P 1* unit cell, then excess symmetrically equivalent moieties are removed preserving however correct ratios of them. This results in the generation of all moieties in the unit cell. In the second approach unique moieties are reconstructed by using symmetrically equivalent atoms (if necessary), then those symmetry operators are applied which were not yet applied to them, but are required to generate their counterparts in the crystal. The first approach is simpler, however, superfluous moieties are generated, only to be discarded afterwards. The second approach uses some algebra, but is less computationally intense. For further comparison we have implemented both algorithms in `Perl` programming language in the `cod-tools` package. Below is the informal description of the first approach:

1. Every symmetry operator of the crystal symmetry space group is applied to every atom, and the resultant image is reduced modulo 1, i.e. moved to a representative unit cell (unit cell closest to the origin in the first octant, spanning fractional coordinates [0..1), [0..1), [0..1)). Each generated atom image gets a unique identifier "cell_label", a list is initiated for all "cell_label" identifiers which are already used in moieties, originally empty.

2. For neighbour search the representative unit cell is used to generate 3×3×3 mesh (so-called "supercell"). Supercell is necessary to capture connections crossing unit cell boundaries. In order to speed up the neighbour search, all atoms of all 27 unit cells of the supercell are moved into an array of cubic "boxes", with each box having an edge equal to the longest possible covalent bond, that is twice the largest covalent radius of an atom in the analysed crystal with the addition of configurable margin. This way the detection of covalent neighbours for an atom is carried out in only 27 adjacent cubic boxes. This method is significantly less computationally intensive compared to the search in all 27 unit cells of the supercell (see for example Levinthal (1966) [150]), making the algorithm's complexity linear if the density of atoms remains constant. An implicit assumption is made that all lengths of the unit cell are longer than the longest bond in the crystal.

3. An atom with yet unused "cell_label" is taken as a starting point of a new molecule. Starting from this atom a connected graph (a molecule) is built by recursively searching for connected atoms in the surrounding "boxes". When neighbours are searched for an

atom outside the representative cell, its coordinates are reduced modulo 1. The translation (former integer part of the coordinates) is later added to the coordinates of the neighbours. By doing so we ensure that atoms outside the supercell are also found. When no more covalently connected atoms can be added to the graph, its construction is complete and the step is repeated for the next atom with yet unused "cell_label". The search is stopped when all "cell_label" identifiers of the representative unit cell are used.

4. Symmetrically equivalent molecules are generated during the previous step if at least one of their atoms is present in the representative unit cell. Such molecules fall into groups, where each molecule of the group is a symmetrically equivalent image of the other molecule in the group. For the minimal stoichiometrically correct representation of the substance not all equivalent images are required. Each group of equivalents originate from the same set of atoms in the original CIF file, thus they have the same atom site labels (same values of `_atom_site_label` CIF data items) in each of the equivalent molecules. We identify each of the generated molecules by key K, formed from sorted and concatenated atom site labels. Grouping molecules by K we acquire counts of molecules under each keys and find the greatest common divisor D of these counts. We then construct the stoichiometrically correct description of the substance by outputting only N_i/D molecules from i-th group with the total count of N_i molecules.

5. Constructed description is not yet minimal as a crystal may contain more than one chemically identical, albeit symmetrically nonequivalent molecules in an asymmetric unit, and all such molecules would be present in the output of the previous step. To eliminate duplicates a chemical fingerprint, for example, Morgan fingerprint [151] could be used for key K generation instead of site labels in step 4. Morgan algorithm establishes canonical numbering of the molecular graph thus interpreting molecules with identical connectivity as equal and chemically different molecules as different. Therefore, chemically identical molecules of the asymmetric unit are grouped together yielding even more reduced stoichiometrically correct representation. As fingerprint usage introduces additional assumptions about the molecular identity from different chemical properties, key generation using Morgan method has been implemented as user-selectable alternative in the algorithm.

The second approach is similar to the algorithm described above; instead of using all unit cell atoms as starting points of molecules, in step 3 we use only atoms of the asymmetric unit, as described in the original CIF file. By doing so we determine a minimal set of molecules with each molecule having at least one atom in the asymmetric unit. As some molecules may contain symmetrically equivalent atoms while others not, this set of molecules is not stoichiometrically correct. For example, for COD entry 2231955 such algorithm would produce one naphthalene-1,5-disulfonate and one dimethyl(4-methylphenyl)ammonium moiety as the most of the commonly used algorithms. Since an inversion centre operator is applied to the atoms of naphthalene disulfonate moiety to restore full moiety from the atoms in the asymmetric unit, to preserve stoichiometry, the same operator (more than one operator, in general) has to be applied to all other moieties of the crystal as well, only if this operator was not used to generate these molecules.

We have developed a program, `cif_molecule` [152], to reconstruct crystal descriptions from asymmetric units, described in CIF files. The output of this program is a CIF file with crystal description in *P 1* with additional information given in `_cod_molecule_*` data items:

- `_cod_molecule_atom_*`: a CIF loop listing details of symmetrically-restored atom sites:
  - `_cod_molecule_atom_label`: a generated unique atom label;
  - `_cod_molecule_atom_orig_label`: original atom label from the input CIF file;
  - `_cod_molecule_atom_symmetry`: symmetry operator, applied to the original atom to produce this symmetric equivalent, expressed as `S_ABC`, where `S` is a numeric identifier of crystal symmetry operator and `A`, `B` and `C` denote translation (in unit cells) along x, y and z axes, correspondingly, augmented by 5. Therefore, `1_555` means identity operator with no translation. Some of the atoms might be in a "special position", that is, there is more than one symmetry operator placing images of an original atom at the same point in space. For such images the symmetry operator with the smallest number `S` is stored, the rest of the operators are placed in `_cod_molecule_transform_*` CIF loop.
  - `_cod_molecule_atom_symop_id`: numeric identifier of crystal symmetry operator used to produce this symmetric equivalent;
  - `_cod_molecule_atom_symop_xyz`: string representation of symmetry operator's transformation matrix;
  - `_cod_molecule_atom_transl_id`: translational part (`ABC`) of `_cod_molecule_atom_symmetry` value;
  - `_cod_molecule_atom_transl_{x,y,z}`: translation along x, y and z axes, correspondingly (integer);
  - `_cod_molecule_atom_mult_ratio` ("multiplicity ratio"): number of crystal symmetry operators that map this atom site to itself;
  - `_cod_molecule_atom_mult`: number of total crystal symmetry operators divided by site's multiplicity ratio. This value is usually provided in CIF files as `_atom_site_site_symmetry_multiplicity`, however, we have noticed that in as many as 200 000 CIF files multiplicity ratios were provided instead of multiplicity values.
  - `_cod_molecule_atom_assembly`: identifier of disorder assembly (if any), as described in Section 3.3.4;
  - `_cod_molecule_atom_group`: identifier of disorder group (if any), as described in Section 3.3.4.
- `_cod_molecule_transform_*`: a CIF loop listing symmetry operators that map the original atom to its symmetric equivalents:
  - `_cod_molecule_transform_label`: "cell_label" of an image site;

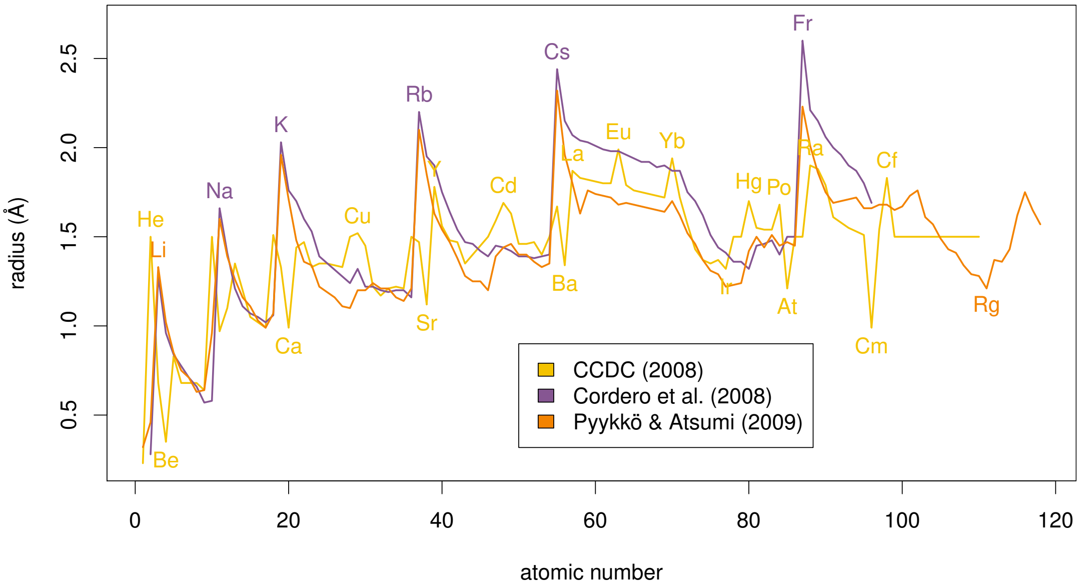


Figure 3.7: Comparison of three sets of covalent radii: CCDC [153], which is used in the current study; Cordero et al. (2008) [22] (missing data points are interpolated); Pyykkö & Atsumi (2009) [154].

  - `_cod_molecule_transform_symop`: string representation of symmetry operator's transformation matrix;

- `_cod_molecule_is_polymer`: `yes` indicates that the crystal structure contains at least one moiety that spans infinity unit cells at least in one dimension (for more information refer to Section 3.3.6). Absence of the data item or value `no` indicates otherwise.

Human- and machine-readable descriptions of these data items and their values are provided in CIF dictionary `cif_cod.dic`[7].

However, with our method, high symmetries, complicated connectivities and large unit cells usually require vast computational resources. To spare the resources we have limited the execution of `cif_molecule` to 600 seconds of processor time and 1 GB of virtual memory. In order to eliminate blocked processes that do not consume neither CPU time nor memory, each process is limited to an hour of wall clock time. limit them to 600 seconds of processor time and 1 GB of virtual memory. Processes exceeding these limits are killed off.

3.3.2 Connectivity

For the current research we have employed covalent radii table as reported in 2008 by the CCDC [153], based on tables published as early as 1979 [155, 156]. As an alternative, we have also investigated the study of Pyykkö & Atsumi (2009) [154], which provides a list of covalent radii for single bonds, and since single bonds usually are the longest, this table is sufficient for the purpose of connectivity determination. When compared with the CCDC table, radii of Pyykkö

[7] `http://www.crystallography.net/cif/dictionaries/cif_cod.dic`

& Atsumi are generally shorter for d- and f-block elements, with the greatest differences for copper and cadmium, and considerably longer for s-block elements (see Figure 3.7 for a detailed comparison). Although the results of Pyykkö & Atsumi rely on well-established theoretical background, radii of the CCDC are manually adjusted to better fit the connectivity as perceived by the researchers at the CCDC. Due to this reason we have chosen the table of CCDC covalent radii over the one of Pyykkö & Atsumi.

Difference of Cu–O distance cutoffs deserves more attention. Harding (1999) [157] has reported Cu–O distances in six-coordinate copper-water complexes in the CSD making up a bimodal distribution with peaks at 2.00 Å and 2.38 Å, demonstrating the Jahn–Teller effect [158]. As copper-water complexes have well-defined geometry, we have deemed the inclusion of such complexes in our library useful. Therefore, we had to treat these complexes as connected moieties. The sum of Pyykkö & Atsumi covalent radii for Cu–O is 1.83 Å and even with the addition of covalent sensitivity margin of 0.35 Å, as used in `cif_molecule` by default, the majority of long Cu–O "bonds" would not be treated as such. This observation, however, does not mean that radii of Pyykkö & Atsumi are flawed, since Cu–O relations in copper-water complexes are not proper covalent bonds. Nevertheless, Cordero et al. (2008) have observed increased plasticity of copper coordination sphere with respect to its neighbouring elements even with the exclusion of observations affected by the Jahn–Teller effect [22].

In order to determine correct connectivity for polymeric moieties (see Section 3.3.6 for discussion), `cif_molecule` constructs and outputs up to $9 \times 9 \times 9$ unit cells (supercell) to represent up to 9^3 repetitions of each atom. Such large supercells were deemed necessary for proper atom type assignment, as described in Section 3.5.4.

3.3.3 Bumps

The simplest method of bump detection is a threshold value, considered to be the shortest allowed contact distance: Su et al. (2017) used 0.6 Å whereas Long et al. (2017b) chosen even stricter cutoff of 0.1 Å [147, 115]. Two atoms are considered as a bump by `cif_molecule` if the distance between them is less than a certain fraction of the sum of their covalent radii (currently, a fraction of 0.75 is used by default) [152]. Bumps are found in as much as 10% crystal structures in the COD[8]. Reasons of bumps include incorrect symmetry, unmarked alternative atoms, presence of symmetrically equivalent atoms (several non-$P\ 1$ structures having all symmetric atoms listed have been identified and corrected during screening of the COD[9]) or the lack of numeric precision during the crystal reconstruction. Structures with bumps are often treated as containing errors, therefore removed from the consideration. Nevertheless, bumps usually occur in unmarked disordered solvent moieties (such as water, toluene, ClO_4^-, BF_4^- and PF_6^- anions [8, p. 221–250]), thus not affecting the perceived connectivity of the "main" moieties in crystals. Removal of moieties with bumps might seem as a solution, although genuine large moieties might also be excluded due to single unmarked variable groups, such as methyl. Therefore, we have decided to process structures with bumps alongside the others and inspect the influence of suspiciously short distances to the statistics afterwards.

[8] Starting from here, statistics are taken from revision 1339 of svn://www.crystallography.net/molecules-in-COD/trunk/statistics/statistics.csv

[9] See, for example, COD entries 2101709–2101727

3.3.4 Processing disorder

Disorder in crystal structures is usually resolved (if possible) by providing coordinates of all alternative positions for each of the disordered atoms. A cluster of atoms of a molecule, observed to occupy alternative positions simultaneously, is called "disorder assembly"[10]. Participation in an assembly (usually named with capital letters) is denoted in `_atom_site_disorder_assembly` CIF data item. Alternative positions of an assembly are called "disorder groups" (usually identified by integers)[11]. Therefore, an atom belonging to an assembly `A` that assumes three alternative locations `1`, `2` and `3` is represented by three CIF atom sites identified as { (`A`, `1`), (`A`, `2`), (`A`, `3`) }. Disordered regions are often difficult to accommodate in the frameworks of stereochemical analysis. Common practice is to ignore observations from minor groups taking only the most prominent group of an assembly [45]. We do not attempt to use occupancies as weights. Instead, all observations from disordered fragments are treated as equal in the current study. We argue that such observations would provide genuine knowledge about possible geometry in crystal structures, as long as all atom sites representing disordered fragments are measured independently. Thus we have attempted to include all disordered atoms. We treat atom as disordered if both the identifiers of assembly and group are known (not "`?`") and the value of the latter is not "`.`". If these requirements are not met, an atom is considered "stable". Therefore, all distances between stable and disordered atoms that match connectivity criterion are treated as connections. Connections between two disordered atoms are recognised unless both belong to different groups of the same disorder assembly. Thus connections between atoms of the same group and of different assemblies are treated as bonds.

Cases occur when disordered moieties are located around special positions, resulting in crystal structure models with fragments that are not affected by crystal symmetry. IUCr recommends placing symmetry-independent sites in disorder groups identified by negative numbers in CIF files. The COD was found to contain around 2% of such structures. We have adapted `cif_molecule` to skip symmetry reconstruction of such atoms as they are symmetry-independent.

3.3.5 Disorder-dependent connectivity

Alternative conformations, especially multiple occupations of the same site, cause changes in connectivity. In an imaginary situation, crystal contains 50%/50% mixture of benzene and pyrazine. Asymmetric unit contains two carbon sites and one carbon/nitrogen site with according occupations of 0.5/0.5. Such crystal has three independent observations of bond lengths, but some of them have weights of 0.5. Since the vast majority of the observations in the current research have weights of 1 and only 9% of structures were detected to have at least one connectivity-changing site, we have decided to exclude such structures from our research, arguing that the loss of observations from such structures would not influence the results significantly.

A real-life example is `C18` atom of COD entry 4317305 (as of revision 179253), whose neighbours are two carbon atoms in disorder group 1, or one carbon and one sulfur in disorder group 2. From the structure it is evident that the thiophene group obtains one of the two

[10] `https://www.iucr.org/__data/iucr/cifdic_html/1/cif_core.dic/Iatom_site_disorder_assembly.html`

[11] `https://www.iucr.org/__data/iucr/cifdic_html/1/cif_core.dic/Iatom_site_disorder_group.html`

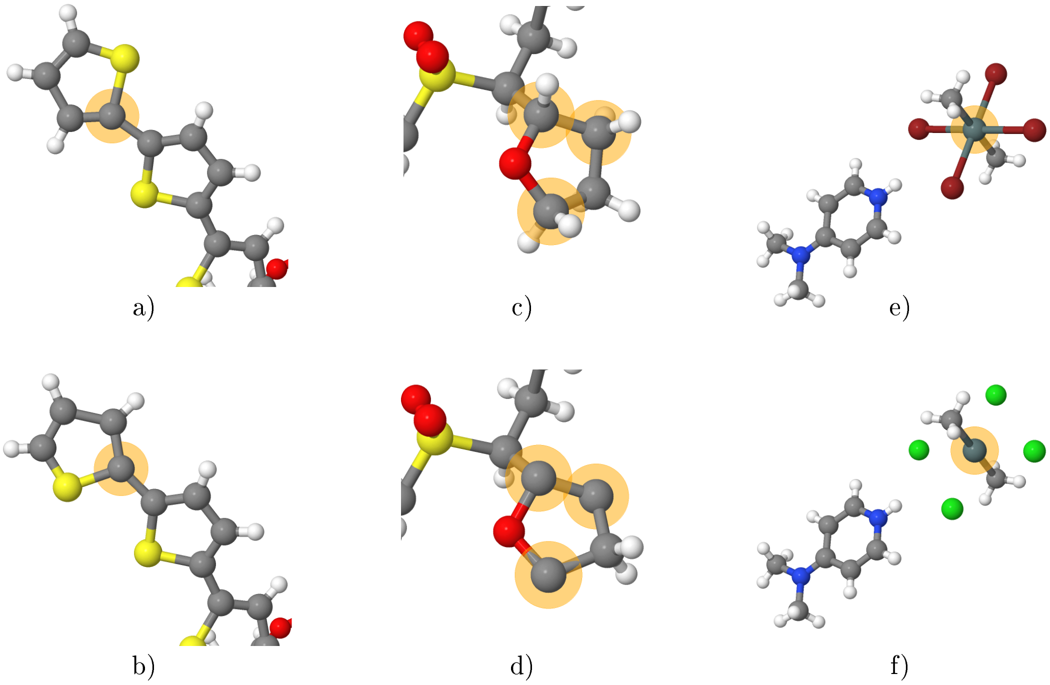


Figure 3.8: Examples of disorder-dependent connectivity in the COD. **a)** and **b)** (COD 1502099) display two possible conformations of thiophene group; although highlighted carbon atom has the same atom type in both conformations, current implementation is unable to handle it. Highlighted carbon atoms in **c)** and **d)** (COD 1502045) differ by the number of attached hydrogens that influence their types. Due to different equatorial atoms in **e)** and **f)** (COD 2218544) the type of highlighted tin atom is different between conformations.

positions, however, currently a single atom in a crystal structure is allowed to have a single atom type, thus such cases are not accommodated. More examples of this issue are shown in Figure 3.8.

3.3.6 Polymer spans

Our definition of connectivity allows a strict determination of boundaries of moieties. However, it is not uncommon for moieties to be virtually endless in a sense that they span infinitely many crystal cells in any direction. We refer to such moieties as "polymers" (a collision with a chemical meaning of polymer is avoided as the COD by definition excludes biopolymers). A moiety is polymer if and only if there is a vector $\vec{x}$ such that

$$\vec{x} = l\vec{a} + m\vec{b} + n\vec{c}, l, m, n \in \mathbb{Z}, \tag{3.1}$$

where $\vec{a}$, $\vec{b}$ and $\vec{c}$ are cell vectors, which maps each atom of a moiety onto its symmetry equivalent atom *in the same moiety.* As there may be more than one such vector $\vec{x}$, let us refer to them as $\vec{x_i}$:

$$\vec{x_i} = l_i\vec{a} + m_i\vec{b} + n_i\vec{c}, l_i, m_i, n_i \in \mathbb{Z}. \tag{3.2}$$

The rank of matrix of all components l_i, m_i, n_i would be equal to "polymer dimension" d [53], where $d = 1$ stands for linear, $d = 2$ – planar and $d = 3$ – three-dimensional polymer.

As processing of the polymeric structures requires substantial amount of resources, we have decided to limit the number of "polymeric atoms", that is, copies of the same initial atom in one moiety. When maximum number of repetitions is reached, `cif_molecule` terminates with an error message.

3.4 Structural biasses

3.4.1 Model bias

In some cases initial assumptions about the determined crystal structure can result in either missed genuine or emerging artificial features. For example, a group of atoms constrained to lie on the same plane would by no means reveal the out-of-plane placement, and supersymmetric structures would exhibit huge displacements for at least some atoms. Cruz-Cabeza et al. (2012) have investigated dihedral angles of biphenyl and cyclobutane to understand the reason of anomalous conformational behaviour. The authors have concluded that high-energy dihedral angles in these compounds correlated with a crystallographic symmetry operator, relating one part (usually a half) of the moiety to itself. Therefore, the chosen symmetry space groups forced systematic conformational bias. To prevent it, the authors suggested excluding moieties having parts related by symmetry operators from an analysed sample [47].

3.4.2 Observations related by symmetry

Crystals of small molecules usually display high-order symmetry. Thus it is of no surprise that a list of crystal's geometric parameters, such as bond lengths, contains a lot of values repeated due to the symmetry. In order not to bias statistics towards multiply repeated although once measured observations, measurements of symmetric fragments have to be excluded from consideration.

In the simplest case, repetitions arise from copies of whole moieties, for example, four copies of hexaphenylbenzene moiety in COD entry 1503391. It is important though to notice that symmetrically unrelated identical moieties are independent observations of the same moiety, as their positions in the model are not mathematically related. Therefore, they are measured independently. Symmetrically equivalent moieties can be easily discarded by keeping just a single copy of a moiety. Current algorithm in `cif_molecule` achieves this by selecting moieties of unique composition. However, current algorithm does not distinguish between isomers, thus only a single isomer is extracted from a cocrystal of a pair or more isomers. To avoid such deficiency, a more sensitive filter of duplicates could be employed, for example, comparison of moieties by their Morgan fingerprints [151]. Nevertheless, Morgan fingerprints do not distinguish stereoisomers.

Sometimes parts of the same moiety are related by symmetry, for example, two halves of benzene in COD entry 4501703. It is important to note that these two halves, related by an inversion operator, contain three independent carbon-carbon bond lengths: two of them are between the atoms in the asymmetric unit of the crystal and the third is between the first and the third atoms of the fragment and its symmetric copy, respectively. Other three carbon-carbon lengths are duplicates of aforementioned observations. In some cases an atom is projected onto itself by a symmetry operator (such atom is said to lie on a special position).

In order to exclude observations from symmetrically related fragments, we have implemented the following algorithm:

1. All symmetry operators $s_1, s_2, ..., s_m$ are applied to each atom x_i of the asymmetric unif of a crystal, resulting in an image $x_{i,j}$ of S_i. A set of site symmetry operators $S_{i,j} = \{s_j\}$ is then defined for $x_{i,j}$.

2. Copies x_{i,j_1} and x_{i,j_2} of an atom x_i that appear at the same point in space are merged and their sets of symmetry operators are merged to contain symmetry operators that appear in either s_{i,j_1} or s_{i,j_2}: $S_{i,\{j_1,j_2\}} = S_{i,j_1} \cup S_{i,j_2}$. Thus, symmetry operator sets for atoms on a special position collect all the symmetry operators that map an atom onto itself.

3. An empty list B_{i_1,i_2} of "accepted" bonds between each pair of parent atoms x_{i_1} and x_{i_2} is defined.

4. After the detection of a bond between atoms x_{i_1,j_1} and x_{i_2,j_2}, S_{i_1,j_1} and S_{i_2,j_2} are searched for a common symmetry operator that could map atoms of any bond from B_{i_1,i_2} onto x_{i_1,j_1} and x_{i_2,j_2}. If such symmetry operator is found, the bond between x_{i_1,j_1} and x_{i_2,j_2} is rejected as symmetrically equivalent. Otherwise the bond is accepted and included in B_{i_1,i_2}.

The extension of this algorithm for bond (three-atom) and dihedral (four-atom group) angles is straightforward.

3.4.3 Physical biasses

Small differences in geometry can be caused by tautomerism, an ability of a compound to readily interconvert to an isomer. The most common type of tautomerism is caused by relocation of a hydrogen atom or a proton. Such tautomers might not be told one from another by means of X-ray crystallography [159]. Interestingly, it has been noted that pairs of tautomers often tend to crystallise together in the same crystal structure [78].

It must be noted, however, that interatomic distances, among other parameters, depend also on the temperature [160]. Nevertheless, a large portion of the data in the COD is collected at or close to the room temperature ($\sim \frac{1}{3}$ of entries in the COD) with only around 3000 entries exceeding 400 K. Therefore we made no attempt to specifically treat the descriptions of the crystals measured in outstanding temperatures.

3.4.4 Hydrogen positions

In stereochemical studies hydrogen atoms usually require special treatment due to many issues related to their placement and refinement. However, positions of hydrogen atoms are of high importance from both structural and chemical points of view. General omission of hydrogen atoms during the refinement introduces some shift in the positions of the atoms that they are attached to, as omissions of model features are compensated for by other parameters. Therefore, hydrogen atoms should not be left out [11, 28]. It is argued that a widely used practice to use "riding model" for hydrogen positions during refinement (atoms are attached to some heavier atom and are moved together) may have been used for the most of small molecule structure determinations so far [161, 115]. Although neutron diffraction largely averts these problems, such structures are rather scarce (the number of such structures in the COD is around 1100, thus less than 0.3%). Furthermore, geometry of X-ray and neutron diffraction determinations should not be analysed together, as discussed in Section 2.3. Despite the fact that X-ray diffraction should obtain correct hydrogen positions at low temperatures [11], stereochemistry of hydrogen-containing fragments is rarely of interest in database-based knowledge extraction due to the aforementioned reasons [113]. In this study no special treatment is used for hydrogen atoms, taking their positions from crystal structure descriptions as they are. Thus the resulting distributions should reflect the geometry of fragments containing hydrogen atoms as they are found in the database. Therefore, we make no claim to precisely reflect the "real" geometry of hydrogen atom positions.

3.4.5 Results of theoretical simulations

In 2013 results of theoretical simulations being deposited to the COD were noticed for the first time. This caused the policy of accepting only experimentally detected structures to the COD to be reiterated. A sister database, the TCOD[12], was opened to collect all the kinds of theoretically defined structures. Instead of removing those theoretical structures that were already in the COD, it was decided to mark them, as is being done with the rest of the crystal structures not fitting either the scope or criteria of the COD. A special value "`theoretical`" for CIF data

[12] `http://www.crystallography.net/tcod`

item _cod_struct_determination_method was introduced to distinguish bespoke structures. Since the introduction, more than 450 structures were manually marked as theoretical in the COD. Detection and marking of such structures remain mostly manual tasks, as it is difficult to automatically identify theoretical structures from the data given in CIF files. However, properties like high numeric precisions of cell constants and coordinates, missing standard uncertainties and experimental details may be used to guide this task.

3.4.6 Duplicated data

The COD aims at collecting all small-molecule crystal structures, published in peer-reviewed journals with an addition of direct personal communications to the database. The policy of the COD is to avoid duplicates. This policy is also important for the aims of the current study. Uniqueness of each structure in the COD is enforced by comparison of the incoming structures with the rest of the database in an attempt to identify possible duplicates. Currently, two structures are considered as duplicates if they originate from the same publication, have the same unit cell parameters and contents, are measured under the same conditions and are not enantiomers of each other or deliberately deposited results of different refinement runs for the same diffraction data. It should be emphasised that the COD should not be treated as duplicate-free at any given moment in time. However, methods for duplicate identification are devised and deployed in the COD from time to time, in most of the cases requiring supervision. Identified duplicated entries are always marked with a special flag instead of being removed from the database. The flag is also used as a pointer to the original, more complete entry [15].

Thus observations which originate from different publications are treated as genuine despite their similarity. However, automatic detection of duplicates is far from being optimal. A single difference, either an alternative spelling or a typographical error, corrupts the matching of the bibliographic information. We have resorted to using Digital Object Identifiers (DOIs), although this piece of information is not known for at least 15% of the COD entries, and at least several dozen of known DOIs are possibly incorrect. Nevertheless, a DOI is not a mandatory item for incoming entries. Other methods of semiautomatic a posteriori duplicate detection, such as search for identical coordinates, are under investigation at the COD. During the course of this study over 800 duplicate structures in the COD were detected either manually or semi-manually and marked as such.

A somewhat overlooked problem in data analysis is overrepresentation. Often one or more research groups produce a vast amount of crystal studies of the same or very similar structures under identical or slightly varied experimental conditions. While undoubtedly useful on their own, results of such analyses bias quantitative studies by dominating samples of concern [115]. Current study does no attempt to downplay overrepresented fragments, as developing such means would probably be worth a separate analysis of its own.

3.4.7 Incomplete models

Reconstruction of chemically correct crystal structures requires the input models to be complete, that is, not missing any covalently bound atoms. In the most cases, incompleteness of crystal structures arises from missing hydrogen atoms, as described in Section 3.4.4. However, this is not the only case, as highly disordered parts of crystals are sometimes excluded from their models, especially solvent and alternative conformations. As solvent molecules are usually interacting with other molecules via non-covalent interactions they can be excluded from crystals in this study without introducing any bias. On the other hand, missing covalently bound parts of molecules may distort the statistics. In some cases incompleteness is caused by errors, for example, incorrect setting of symmetry space group in crystal structure description. Such errors may lead to the disappearance of symmetrically equivalent molecules[13] or their parts. Loss of whole molecules is again not as detrimental as their parts. To identify possibly missing structure parts, we have employed `voronota`. A 3 Å filter for void's diameter was applied to ignore voids so small that they could not contain a non-hydrogen atom. A dozen of incomplete structures were detected and corrected[14].

3.5 Atom types

We have developed an atom typing scheme, similar to already existing schemes which take into account chemical connectivity, planarity and participation in rings [25, 115, 162, 163]. We have decided against using atom aromaticity and bond orders, as assignment of these properties of chemical structure relies strongly on coordinate-based heuristics and is therefore unreliable [43]. Our scheme partitions the space of possible chemical environments into non-overlapping atom types, whose names are case-sensitive strings that provide prefix expression of recursive enumeration of atom's neighbours and their neighbours. The key concept in our scheme is "classification depth" which denotes the farthest neighbour ("terminal atom") of the atom in consideration ("core atom") which is taken into account during the type assignment. Here the distance is measured by the number of chemical bonds between two atoms or, in the language of graph theory, number of joining edges. In this work we use a variant of our classification scheme with depth of 2, that is, only core atom's neighbours and their neighbours are considered. For example, type of amino acid's C_α will include information about its attached hydrogens, C_β and its hydrogens, C_γ, amino and carboxyl groups without oxygen-attached hydrogens.

The type identifier for every atom starts with its chemical type. It is the only piece of information for the terminal atoms. If an atom is non-terminal and is deemed to be located in a planar environment (described in Section 3.5.1), the first letter of its chemical type is lowercased; the similar convention is used for atoms in aromatic environments in SMILES [164, 88]. Chemical type for every non-terminal atom is followed by ring participation information (described in Section 3.5.3) enclosed in square brackets. If an atom does not belong to any rings, the brackets are omitted altogether. Type identifiers of direct neighbours follow, always of decremented classification depths. All atoms, except the core and terminal, are enclosed in parentheses in order to reduce ambiguity. Identical types in the neighbour lists are contracted to shorten the type string. Thus the type of ammonia nitrogen is `N(H)3` (classification depth = 2) or `NH3` (classification depth = 1). An example of atom type construction for organic molecule is presented in Figure 3.9. The EBNF grammar for atom types is given in Figure 3.10. For the assignment of atom types for crystal structure descriptions prepared by `cif_molecule`, we have

[13] Compare, for example, revisions 91933 and 118838 of COD entry 2001917.

[14] See for example COD entries 2001521 and 7204155.

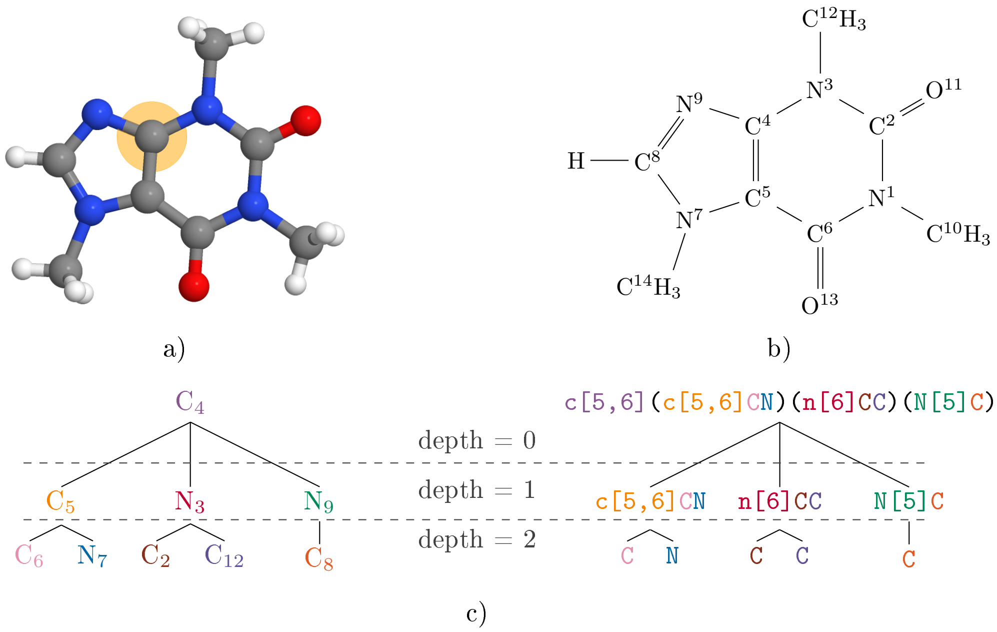


Figure 3.9: Construction of atom type for C_4 atom of caffeine molecule. **a)** caffeine molecule (COD entry 2100202) with C_4 atom (core atom) marked with orange halo. **b)** the structural formula. **c)** *left* – molecular tree for C_4, *right* – atom types for each corresponding atom (colour coded). Note that parent atom C_4 is excluded from neighbour lists in types of depth = 1.

developed a program `cif_bonds_angles` in *atomclasses* software package[15].

3.5.1 Planarity

Molecular geometries are usually dictated by orbital hybridisation of their constituents, thus accommodating the latter in atom classification is beneficial and is sometimes done [7, 25]. However, the determination of orbital hybridisations from connectivity information and geometry only is not straightforward and is highly sensitive to the stereochemistry, therefore we have decided to exclude this information from our scheme. The only geometry-derived information bit in our classification scheme apart from the connectivity is planarity. An atom is considered planar if it has three or more covalent neighbours and all of them are deemed to be lying on the same plane. An atom is deemed planar if the largest ratio of its absolute chiral volume [1, 37] to the product of three bond lengths is less than 0.1:

$$\frac{|\vec{v}_1(\vec{v}_2 \times \vec{v}_3)|}{||\vec{v}_1|| \times ||\vec{v}_2|| \times ||\vec{v}_3||} < 0.1, \tag{3.3}$$

where $\vec{v}_i$ is a vector of a bond. Interpreting $\vec{v}_i$ as vectors defining parallelepiped, Inequality 3.3 could be rewritten as

$$\sqrt{1 + 2\cos(\alpha)\cos(\beta)\cos(\gamma) - \cos^2(\alpha) - \cos^2(\beta) - \cos^2(\gamma)} < 0.1, \tag{3.4}$$

[15] Available under the GPL2 free software license at `svn://saulius-grazulis.lt/atomclasses/trunk`, this study refers to source revision 559

```
(* The top-level rule: *)

AtomClass = CoreAtom, [ TreeList | FlatList ] ;

(* Two types of neighbour lists are available: *)

TreeList = ’(’, AtomClass, ’)’, [ Integer ], { TreeList } ;

FlatList = NonplanarAtom, [ Integer ], { FlatList } ;

(* CoreAtom is the root of the tree: *)

CoreAtom = Atom, [ Rings ] ;

(* Rules for the ring list: *)

Rings = ’[’, RingSize, { ’,’, RingSize }, ’]’ ;

RingSize = Integer, [ ’x’, Integer ] ;

Integer = ( ’1’ | ’2’ | ’3’ | ’4’ | ’5’ | ’6’ | ’7’ | ’8’ | ’9’ ),
          { ’1’ | ’2’ | ’3’ | ’4’ | ’5’ | ’6’ | ’7’ | ’8’ | ’9’ | ’0’ } ;

(* Rules for the chemical type: *)

Atom = PlanarAtom | NonplanarAtom ;

PlanarAtom = LowercaseLetter, [ LowercaseLetter ] ;

NonplanarAtom = UppercaseLetter, [ LowercaseLetter ] ;

(* Basic character classes: *)

UppercaseLetter =
    ’A’ | ’B’ | ’C’ | ’D’ | ’E’ | ’F’ | ’G’ | ’H’ | ’I’ | ’J’ | ’K’ | ’L’ |
    ’M’ | ’N’ | ’O’ | ’P’ | ’Q’ | ’R’ | ’S’ | ’T’ | ’U’ | ’V’ | ’W’ | ’X’ |
    ’Y’ | ’Z’
;

LowercaseLetter =
    ’a’ | ’b’ | ’c’ | ’d’ | ’e’ | ’f’ | ’g’ | ’h’ | ’i’ | ’j’ | ’k’ | ’l’ |
    ’m’ | ’n’ | ’o’ | ’p’ | ’q’ | ’r’ | ’s’ | ’t’ | ’u’ | ’v’ | ’w’ | ’x’ |
    ’y’ | ’z’
;
```

Figure 3.10: The EBNF grammar for atom types

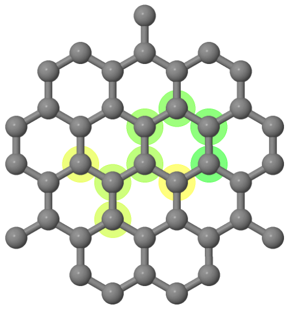


Figure 3.11: Ring perception in graphene sheet (cropped). Atoms are coloured according to the order of visiting: the most green atom is the original atom, and the most yellow is visited the last; in this case it is the ring closing atom.

where α, β and γ are the smallest angles between vectors. In case of equal angles, inequality holds for angles of less than $\sim 19.7°$ (chemically correct fragments should not fall into this range) or more than $\sim 119.9°$. Environments involving an atom bound to more than three neighbours are assumed planar if Inequality 3.3 holds for all possible permutations of its neighbours. Cutoff of 0.1 is deemed suitable to tell planar geometry from, for example, trigonal pyramidal (all internal angles $\sim 107°$), tetrahedral ($\sim 109.5°$), or flattened tetrahedral arrangements in highly strained fenestranes (angles up to 130°, COD entry 4127219).

3.5.2 Disorder-dependent planarity

As with disorder-dependent connectivity, described in Section 3.3.5, planarities of atoms can also be affected by the disorder. The current research has identified that 4% of structures in the COD have this issue. For example, `C18A` atom of COD entry 1502901 is located either on the same plane as its neighbours (disorder group 1) or not (disorder group 2). Theoretically, all piperidines, whose N–H bond takes either axial or equatorial conformation, fall to this category. Since atom typing includes information about planarity, different conformations of such compounds affect types of neighbouring atoms. Due to the difficulties of accommodating such observations, we have decided to exclude structures with disorder-dependent planarity from the further processing.

3.5.3 Rings

Information about participation in rings is included into atom types by listing numbers and sizes of all rings the atom of interest is in. Only rings of seven and less atoms are considered. We argue that participation in larger rings is of limited influence as local conformations in large rings become more similar to those of non-ringed chains, although rings up to 12 atoms are used by PURY classification scheme [7]. If deemed necessary, maximum ring size in our scheme can be increased by a command line option of `cif_bonds_angles`. However, consideration of larger rings would be more computationally intensive.

The list of rings for an atom is provided in square brackets. Inside the brackets, rings are ordered by their sizes, grouped and separated by commas. Thus, ring participation part of atom

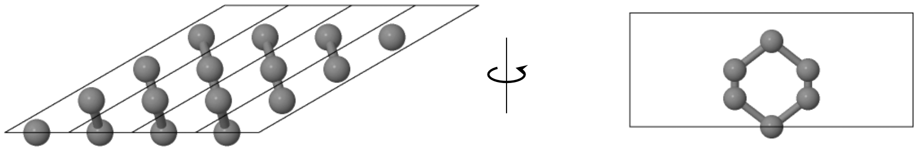


Figure 3.12: Hypothetical crystal structure with rings spanning four unit cells in one direction.

type for benzene carbon is `[6]`, cubane carbon – `[3x4]`, meaning "three rings of size four", carbon of the "bridge" in caffeine molecule – `[5,6]`. It should be noted, however, that grouping starts from three rights of the same size (see cubane for example). Brackets are omitted altogether if an atom does not belong to any rings.

The smallest set of smallest rings [165] for each atom is determined using a modified version of algorithm by Downs et al. (1989) [166] and others [167, 168, 169]. The algorithm uses depth-first search to find all chordless cycles containing the atom in question. The size of cycles is limited to the size of the largest allowed ring (seven atoms in the current study, with the possibility to be modified). As ring participation has to be determined for every atom, each of them is used as a starting point ("original atom") for the modified depth-first search. The algorithm is described below:

1. Upon visiting an atom it is set as the "current atom".
2. If current atom is not the original atom, is not preceding the current atom and it is in the list of already visited atoms, it is called a "Nachbarpunkt" ("neighbour point" in German, terminology is taken from Downs et al. (1989) [166]). The search is then terminated.
3. If no Nachbarpunkt is detected, neighbours of the current atom are searched for the original atom. If the original atom is found among the neighbours of the current atom, a ring is considered closed. Original atom is considered as a Nachbarpunkt, thus the search is terminated.
4. If Nachbarpunkt is not found in the previous steps, the depth is increased. If the maximum depth is reached, the depth-first search is resumed from step 1 without considering neighbours of the current atom. Otherwise, the current atom is put in the list of "seen atoms". Preceding atom is then set as the current atom and step 1 is performed for each its unvisited neighbour.

An example of visiting order is given in Figure 3.11.

3.5.4 Polymers

While individual moieties are always fully reconstructed for nonpolymeric crystal structures, correct representation of chemical environments in polymeric crystals requires additional heuristics. Our algorithm trims virtually infinite moieties of polymeric crystals along the sides of the unit cell, introducing so-called "polymer cuts". It is evident that chemical environment of atoms situated on these cuts is incorrect as it lacks connections across the cuts. This problem

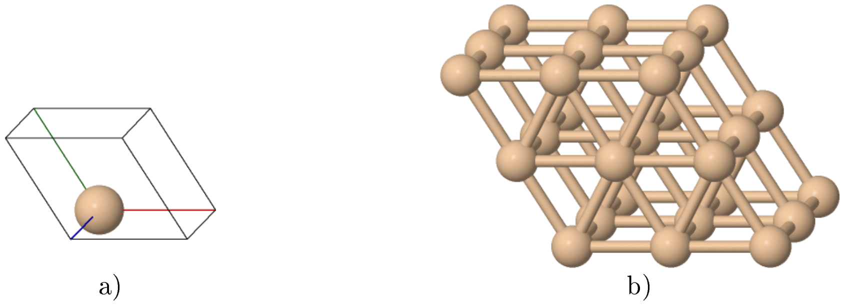


Figure 3.13: An example of 3D polymer with a tiny unit cell from a high-pressure silicon structure, COD entry 9012918. **a)** unit cell with a single Si atom, **b)** $2 \times 2 \times 2$ supercell, displaying connectivity of a single Si atom with analogous atoms in 8 neighbouring cells.

is avoided by using supercells sufficiently large to guarantee correct representation of chemical environments of atoms of the central unit cell. However, very small unit cells or cells with very acute angles need larger unit cells for correct detection of rings, atom types and fragments:

- An atom is connected to its translational equivalent along at least one of the crystal axes. Thus its first (closest) neighbours reside in adjacent cells, and second neighbours – in cells -2 and $+2$. An example of such crystal is high-pressure silicon (COD entry 9012918, Figure 3.13). Atom type detection for this crystal requires at most $5 \times 5 \times 5$ supercell for classification depth of 2.

- A four-atom fragment in previously described crystal can span at most 4 unit cells, requiring a $7 \times 7 \times 7$ supercell; otherwise, its dihedral angles would not be possible to measure.

- A seven-membered ring in crystal spanning 4 unit cells in one direction (see Fig. 3.12, more is difficult to imagine), therefore requiring at least $7 \times 7 \times 7$ supercell for its detection.

For polymer representation by `cif_molecule` we have chosen little larger, $9 \times 9 \times 9$ supercells. If during the construction of the supercell the limit of maximum number of polymer atoms (see Section 3.3.6) is reached, the size of the supercell is reduced by two (from $9 \times 9 \times 9$ to $7 \times 7 \times 7$) and the procedure is repeated. Such reduction is continued, if needed, until the supercell is reduced to the unit cell.

3.6 Geometric measurements

While the length of a bond is unambiguously defined as the distance between two points in space, the two other measurements – bond and dihedral angle – need to be addressed in more detail. It is important though to emphasise that bond length takes a value from range $[0, l_c]$, where l_c is maximum distance between two atoms that is considered as a bond. In this study, bond angle is measured as a smallest angle between two vectors expressed in degrees, and as such takes a value from inclusive range $[0, 180°]$. The following formula is used to measure angle

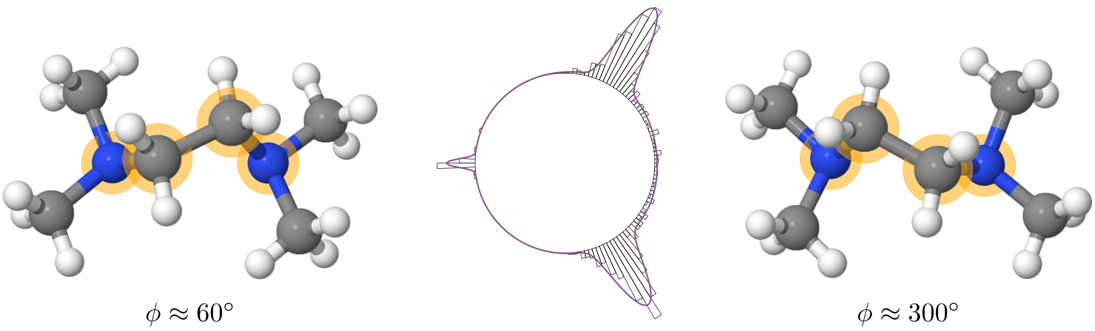


Figure 3.14: Circular histogram of N–C–C–N dihedral angles (atoms highlighted) in TEMED structures. Well-pronounced peaks at roughly 60° and 300° correspond to two enantiomeric forms of the compound, illustrated on both sides of the histogram.

α between bond vectors $\vec{a}$ and $\vec{b}$:

$$c = \cos\alpha = \frac{\vec{a} \cdot \vec{b}}{||\vec{a}||\ ||\vec{b}||} \tag{3.5}$$

$$\alpha = \mathrm{atan2}(\sqrt{1-c^2}, c), \tag{3.6}$$

where atan2 function is implemented as in `C` and `Perl` programming languages.

Dihedral angle of bonded atoms A–B–C–D is defined as interplanar angle in degrees between planes (A, B, C) and (B, C, D). As such, it takes a value from range $[0, 360°)$. The following formula is used to measure dihedral angle ϕ between planes defined by vectors $\vec{a}$, $\vec{b}$ and $\vec{c}$, $\vec{d}$:

$$c = \cos\phi = \frac{(\vec{a} \times \vec{b}) \cdot (\vec{c} \times \vec{d})}{||\vec{a} \times \vec{b}||\ ||\vec{c} \times \vec{d}||} \tag{3.7}$$

$$\phi = \mathrm{atan2}(\sqrt{1-c^2}, c). \tag{3.8}$$

If the vectors compose a left-handed system ($(\vec{c} \times \vec{a}) \cdot \vec{d} < 0$), the angle is corrected by subtracting it from 2π. It should be noted though that the dihedral angles A–B–C–D and D–C–B–A are always equal, and corresponding dihedral angles of a pair of enantiomers sum up to 360° [8, p. 205–219] (Figure 3.14). Geometry measurements from the crystal structures are extracted by `cif_bond_angles`.

Calculation of uncertainties, although possible as standard uncertainties of the atomic positions are most often provided in CIF files, are deemed out of scope of this study. It is generally held that the standard uncertainties, resulting from crystal structure refinement, are usually underestimated by a factor of 1.5-2. Special features of the refinement, such as applied constraints, usage of rigid groups (only centroid and three rotations are refined) or the riding model for hydrogen atoms, turn standard uncertainties of stereochemical parameters to zero. Free refinement of hydrogen atoms result in larger positional uncertainties than for other atoms [8, p. 205–219]. Therefore, accommodation of uncertainties is cumbersome.

3.7 Statistical model

It has been reported by the previous studies that observations of geometric parameters of molecules (bond lengths, bond and dihedral angles) are not always normally distributed, therefore, they are in need of better-fitting models. A well-known example is bond lengths influenced by Jahn–Teller effect, elongation of axial bonds in octahedral complexes of transitional metals.

Selection of best fitting model for a distribution of measurements can be formulated as follows. From the Bayesian point of view, we have data set $\vec{x}$, sampled from some unknown model, and a set of hypothetical models $\mathbb{M}$. The problem is to choose the model $\hat{M}$ from $\mathbb{M}$, which has the highest probability of being the "real" unknown model (model selection), and define a parameter vector $\hat{\vec{\theta}}$ which describes the observed data the best (parameter estimation). In other words,

$$\hat{M} = \arg\max_{M} Pr(M \in \mathbb{M}|\vec{x}), \tag{3.9}$$

$$\hat{\vec{\theta}} = \arg\max_{\vec{\theta}} Pr(\vec{\theta}|\vec{x}, \hat{M}). \tag{3.10}$$

3.7.1 Model selection

Assuming we have a prior distribution $Pr(\vec{\theta}|M)$ for each model M from $\mathbb{M}$, where $\vec{\theta}$ is the parameter vector for M,

$$\begin{aligned} Pr(M|\vec{x}) \propto Pr(M)Pr(\vec{x}|M) \propto \\ \propto Pr(M) \int Pr(\vec{x}|\vec{\theta}, M)Pr(\vec{\theta}|M)d\vec{\theta}. \end{aligned} \tag{3.11}$$

By assuming that $Pr(M) = \frac{1}{||\mathbb{M}||}$ for all M, Equation 3.11 is simplified to

$$Pr(M|\vec{x}) \propto Pr(\vec{x}|M). \tag{3.12}$$

Following the approximations and simplifications of [170, p. 234],

$$\log Pr(\vec{x}|M) = \log Pr(\vec{x}|\hat{\vec{\theta}}, M) - \frac{d(\hat{\vec{\theta}})}{2} \log ||\vec{x}|| + O(1), \tag{3.13}$$

where $\hat{\vec{\theta}}$ is maximum likelihood estimate of the parameter vector, $d(\hat{\vec{\theta}})$ is the number of independent parameters in $\hat{\vec{\theta}}$. Putting all together we get

$$\hat{M} = \arg\max_{M} Pr(M|\vec{x}) = \arg\max_{M} \log Pr(\vec{x}|\hat{\vec{\theta}}, M) - \frac{d(\hat{\vec{\theta}})}{2} \log ||\vec{x}||. \tag{3.14}$$

3.7.2 Distributions

We assume all observations of a geometric parameter to be independent and identically distributed variables. Thus the probability of a data vector $\vec{x}$ is

$$p(\vec{x}|\vec{\theta}, M) = \prod_{j=1}^{n} p(x_j|\vec{\theta}, M), \tag{3.15}$$

where M is a distribution model, parametrised by a vector $\vec{\theta}$ of latent parameters. n is the length of data vector $\vec{x}$. In the current research we have used the following distribution mixture models: Gaussian, von Mises, Cauchy and Student's t. Several methods were employed to obtain the best approximation $\hat{\vec{\theta}}$ for latent parameters. These methods are reviewed in Section 3.7.3.

Initially we have used Gaussian mixture model for bond lengths and bond angles assuming independence of measurement errors. However, existence of longer than normal tails in some of the distributions forced us to look for better statistical models like Cauchy. We have also employed Student's t model as a natural interpolation between Gaussian and Cauchy distributions [171], since its probability density function becomes Gaussian when its degrees of freedom parameter $\nu \to \infty$, and becomes Cauchy when $\nu \to 1$. Gaussian, Cauchy and Student's t distributions possess an inherent limitation on modelling dense areas at the edges of support domain, for example, frequent observations of angles scattered around 0 or 180°, what would possibly be better modelled by chi-like distributions. On the other hand, we have treated dihedral angle distributions as circular, using von Mises distribution, which is a circular counterpart of Gaussian model.

Given the Gaussian mixture model M with m components and parameter vector $\vec{\theta}$, probability of a single data point is

$$\begin{aligned} p_G(x|\vec{\theta}, M) &= p(x|A_1, ..., A_m, \mu_1, ..., \mu_m, \sigma_1, ..., \sigma_m, M) = \\ &= \sum_{i=1}^{m} \frac{A_i}{\sigma_i\sqrt{2\pi}} \exp(-\frac{(x-\mu_i)^2}{2\sigma_i^2}), \end{aligned} \tag{3.16}$$

where A_i is the mixing proportion of the i-th component, $\sum_{i=1}^{m} A_i = 1$, μ_i is the mean of the i-th component and σ_i is the standard deviation of the i-th component.

Given the von Mises mixture model M with m components and parameter vector $\vec{\theta}$, probability of a single data point is

$$\begin{aligned} p_{vM}(x|\vec{\theta}, M) &= p(x|A_1, ..., A_m, \mu_1, ..., \mu_m, \kappa_1, ..., \kappa_m, M) = \\ &= \sum_{i=1}^{m} \frac{A_i \exp\left(\kappa_i \cos(x-\mu_i)\right)}{2\pi I_0(\kappa_i)}, \end{aligned} \tag{3.17}$$

where parameters A_i and μ_i are equivalent to those of Gaussian mixture model and κ_i is concentration parameter ($1/\kappa$ is analogous to σ^2), $I_n(x)$ is the modified Bessel function of the first kind of order n.

Given the Cauchy mixture model M with m components and parameter vector $\vec{\theta}$, probability of a single data point is

$$\begin{aligned} p_C(x|\vec{\theta}, M) &= p(x|A_1, ..., A_m, c_1, ..., c_m, s_1, ..., s_m, M) = \\ &= \sum_{i=1}^{m} \frac{A_i}{\pi} \frac{s_i}{(x - c_i)^2 + s_i^2}, \end{aligned} \tag{3.18}$$

where A_i is component's proportion, c_i is location parameter and s_i is a scale parameter, equal to semi-interquartile range [172]:

$$s = \frac{1}{2}(F^{-1}(0.75) - F^{-1}(0.25)). \tag{3.19}$$

Student's t distribution

Given the univariate Student's t distribution mixture model M with m components and parameter vector $\vec{\theta}$, probability of a single data point is

$$\begin{aligned} p_{SMM}(x|\vec{\theta}, M) &= p_{SMM}(x|A_1, ..., A_m, \mu_1, ..., \mu_m, \sigma_1, ..., \sigma_m, \nu_1, ..., \nu_m, M) = \\ &= \sum_{i=1}^{m} \frac{A_i \Gamma(\frac{\nu_i+1}{2})}{\sqrt{\pi \sigma_i \nu_i} \Gamma(\frac{\nu_i}{2}) \big(1 + \nu_i^{-1} (\frac{x-\mu_i}{\sigma_i})^2\big)^{\frac{\nu_i+1}{2}}}, \end{aligned} \tag{3.20}$$

where parameters A_i, μ_i and σ_i are equivalent to those of Gaussian mixture model and ν_i is the degrees of freedom of the i-th component [173]. $\Gamma(x)$ is the Gamma function.

Student's t distribution is known to encompass both Gaussian and Cauchy distributions as well as interpolate between them by employing the parameter of degrees of freedom ν to regulate the heaviness of distribution's tails: with $\nu = 1$, $p_S(x)$ is the same as $p_C(x)$, and with $\nu = +\infty$, $p_S(x)$ is the same as $p_G(x)$ [174]. Latter property and the existence of EM algorithms [175, 173, 176, 174] make the distribution noteworthy for the robust modelling [174]. However, it was concluded that the convergence of ν parameter is very slow [176] and highly correlated with the starting value [177]. We have observed that while approximating a vector with random values from Gaussian distribution with $p_S(x)$, value of ν tends to approach infinity almost logarithmically: an approximation of a vector of 20 000 values and convergence criterion $\forall i : |\theta_i^{t+1} - \theta_i^t| < 0.01$, where θ_i^t is the ith component of parameter vector $\vec{\theta}$ at tth step, took over 500 000 EM iterations to pick a value $\nu = 459.1563$. This deemed unsatisfactory for our purpose. Apparently, the convergence of ν parameter does not fare well as one can not simply interpolate between 1 and $+\infty$. We have thus studied a couple of distributions that perform the interpolation between Gaussian and Cauchy distributions [178, 179] as well as symmetric α-stable [180] distribution, however, either their EM algorithms were not published yet or even their probability density functions were deemed too complex to compute for our task.

3.7.3 Parameter estimation

For the approximation of $\hat{\vec{\theta}}$ we have employed EM algorithms with variable numbers of mixture components m. It should be noted, however, that EM depends very heavily on its starting parameters and may mistake a local optimum for a global one [181]. Moreover, as m mixture components can have $m!$ permutations, the hypersurface of the optimised function has $m!$ identical global optima [182]. We assume that distribution of each geometric parameter should consist of no more than ten components, thus we obtain $\hat{\vec{\theta}}$ for each model with different number of components and select the model with the smallest BIC value:

$$BIC = -2 \log p(\vec{x}|M) + d_M \log n, \tag{3.21}$$

where $p(\vec{x}|M)$ is maximum likelihood of data given the model M, d_M is the number of free parameters in the model and $n = ||\vec{x}||$. For all mixture models having three parameters per component

$$BIC = -2 \log p(\vec{x}|\vec{\theta}) + (3m - 1) \log n. \tag{3.22}$$

As a special case, for the Student's t distribution, which has an extra "degrees of freedom" parameter,

$$BIC = -2 \log p(\vec{x}|\vec{\theta}) + (4m - 1) \log n. \tag{3.23}$$

Collapses of mixture components ($\sigma_i^t \rightarrow 0$ as $t \rightarrow \infty$) are sometimes observed while fitting mixtures with EM. As suggested by Archambeau et al. [183], components collapse due to overfitting (i) outliers or (ii) repeated observations. In our study models with collapsed components are excluded from the consideration. We argue that component collapse is a sign of a mixture already having more components than is necessary to describe a sample, therefore, exclusion of such models should not affect the choice of the best model for a sample.

Several different methods of selecting the initial estimates for $A_i(0)$, $\mu_i(0)$ and $\sigma_i(0)$ were examined, including random selection, selection of equidistant points as means with equal mixing proportions and standard deviations (shown to work well in Bohning et al. [181]), using the output of less costly iterative fitting and k-means clustering as suggested in Gupta & Chen and Biernacki et al. [184, 185]. In our research the second initialisation method is used:

$$A_j(0) = \frac{1}{m} \tag{3.24}$$

$$\mu_j(0) = \min(x) + j\frac{\max(x) - \min(x)}{m + 1} \tag{3.25}$$

$$\sigma_j(0) = \frac{\max(x) - \min(x)}{6(m + 1)}, \forall i \in [1, ..., m]. \tag{3.26}$$

The tendency of EM to overlook well-separated modes of a sample distribution [186] is troubling, but we hope that regular grid of initial location estimates should overcome this issue.

We have chosen stopping criteria based on absolute difference of mixture parameters between two consecutive EM steps. All EM algorithms except von Mises are stopped when $\forall i : |\theta_i^{t+1} - \theta_i^t| < \epsilon$, $\epsilon = 10^{-6}$. EM for von Mises is stopped when the difference of model log-likelihood becomes negligible, as suggested in Hornik & Grün (2014) [187]. Log-likelihood based approach was chosen due to the observed tendency of von Mises mixture parameters to get caught in endless periodic cycle.

Expectation maximisation for Cauchy distribution

We have applied expectation function of Gaussian mixture model in EM for Cauchy mixture models. For the maximisation of c_i and s_i, we have adapted iterative approach, as given in the Equations 10 and 18 of Nagy (2006) [188]:

$$g_{i,j,(t)} = \frac{x_i - c_{j,(t-1)}}{s_{j,(t-1)}} \tag{3.27}$$

$$e^{0k}_{j,(t)} = \Big(\sum_{i=1}^{n} h^i_{j,(t)}\Big)^{-1} \sum_{i=1}^{n} \frac{h^i_{j,(t)}}{1+g^2_{i,j,(t)}} \tag{3.28}$$

$$e^{1k}_{j,(t)} = \Big(\sum_{i=1}^{n} h^i_{j,(t)}\Big)^{-1} \sum_{i=1}^{n} h^i_{j,(t)} \frac{g_{i,j,(t)}}{1+g^2_{i,j,(t)}} \tag{3.29}$$

$$c_{j,(t)} = c_{j,(t-1)} + s_{j,(t-1)} \frac{e^{1k}_{j,(t)}}{e^{0k}_{j,(t)}} \tag{3.30}$$

$$s_{j,(t)} = s_{j,(t-1)} \sqrt{\frac{1}{e^{0k}_{j,(t)}} - 1} \tag{3.31}$$

We have chosen 20 for the number of iterations, as demonstrated to work by Nagy (2006) [188].

Expectation maximisation for Student's t distribution

We have implemented the EM for Student's t distribution according to the Equations 12-17 of Gerogiannis et al. (2009) [173]. We have solved the Equation 17 for ν_i^{t+1}

$$\log(\frac{\nu_i^{t+1}}{2}) - \psi(\frac{\nu_i^{t+1}}{2}) - \log(\frac{\nu_i^t + 1}{2}) + \psi(\frac{\nu_i^t + 1}{2}) + \frac{\sum\limits_{j=1}^{N} z^t_{ij}(\log u^t_{ij} - u^t_{ij})}{\sum\limits_{j=1}^{N} z^t_{ij}} + 1 = 0 \tag{3.32}$$

by using the asymptotic expansion series [189]

$$\psi(x) = \log(x) - \frac{1}{2x} + \sum_{n=1}^{\infty} \frac{\zeta(1-2n)}{x^{2n}}, \tag{3.33}$$

where $\zeta(x)$ is the Riemann zeta function. Thus we obtain

$$\sum_{n=1}^{\infty} \frac{\zeta(1-2n)}{x^{2n}} = \psi(\frac{\nu_i^t + 1}{2}) - \log(\frac{\nu_i^t + 1}{2}) + \frac{\sum\limits_{j=1}^{N} z^t_{ij}(\log u^t_{ij} - u^t_{ij})}{\sum\limits_{j=1}^{N} z^t_{ij}} + 1, \tag{3.34}$$

which can be limited to any number of members k with the absolute error of $O(x^{-2k})$ and solved via Jenkins–Taub or Newton–Raphson methods. We have chosen $k = 7$ and used Jenkins–Taub method to solve the polynomial. We have also tried to adapt multivariate EM algorithm reported by Aeschliman et al. (2010) [175] to univariate case, however, involvement of log(0) in summing was found inevitable for data point(s) equal to the median of the sample.

Greedy EM for Student's t distribution

We have implemented and tested greedy EM algorithm for Student's t distribution, which is based on regular EM although capable of automatically splitting components with maximum Kullback–Leibler divergence [190, 191]. However, this algorithm did not outperform regular EM with variable number of parameters.

3.7.4 Software

`R` programming language [192] was used for parameter estimation and model selection. Probability density functions, fitting and random sample generation algorithms were developed and bundled together into an `R` package named `MixtureFitting`[16]. For better performance, core numeric processing was written in the `C` programming language.

3.7.5 Hypothesis testing

The scientific method crucially relies upon the construction and evolution of hypotheses, or models. Given some observations, or data, the best candidate model is chosen ("accepted") to explain the data and to make predictions about new phenomena [105]. Observations that are not-so-well explained by the accepted model could be claimed to be outliers, until enough of them is collected to instigate the construction of a better model. In this study models constructed via maximum likelihood estimation (MLE) are used for the outlier detection.

There are two main approaches for Bayesian-based outlier detection. In the first one a null hypothesis is assumed to generate the data and the outliers are sought without any alternative model. The second approach maintains that a subset of the data sample is generated by an alternative model [193]. We have chosen the latter approach for this study, postulating that the outliers are independent and identically distributed random values. As is usual in the field, we represent both hypotheses using statistical models: a mixture model stands for the null hypothesis (H_0) and a uniform distribution for the alternative (H_1), assuming equal probability for every attainable value. Bayes factor, which obtains the form of a likelihood ratio, is then used to evaluate the fitness of each observation x to one or another hypothesis. Bayes factor is renowned for its usefulness for guiding evolutionary model building as well as serving as Occam's razor. While frequentist tests were designed to compare strictly two models and are prone to the rejection of null hypotheses for very large samples, Bayes factor does not have this deficiency, albeit it is sensitive to the assumptions of the parametric model and the choice of priors [105]. Given the equal probability of the hypotheses, the Bayes factor is defined as follows:

$$K = \frac{P(H_0|x)}{P(H_1|x)}. \tag{3.35}$$

Jeffreys gives the following guidelines for the values of K for the reference [194, p. 432]:

- Grade 0. $K > 1$. Null hypothesis supported.
- Grade 1. $1 > K > 10^{-0.5}$. Evidence against H_0, but not worth more than a bare mention.

[16] Available under GPL2 free software license at `https://github.com/merkys/MixtureFitting`, this study refers to version 0.1.0 (source revision 132)

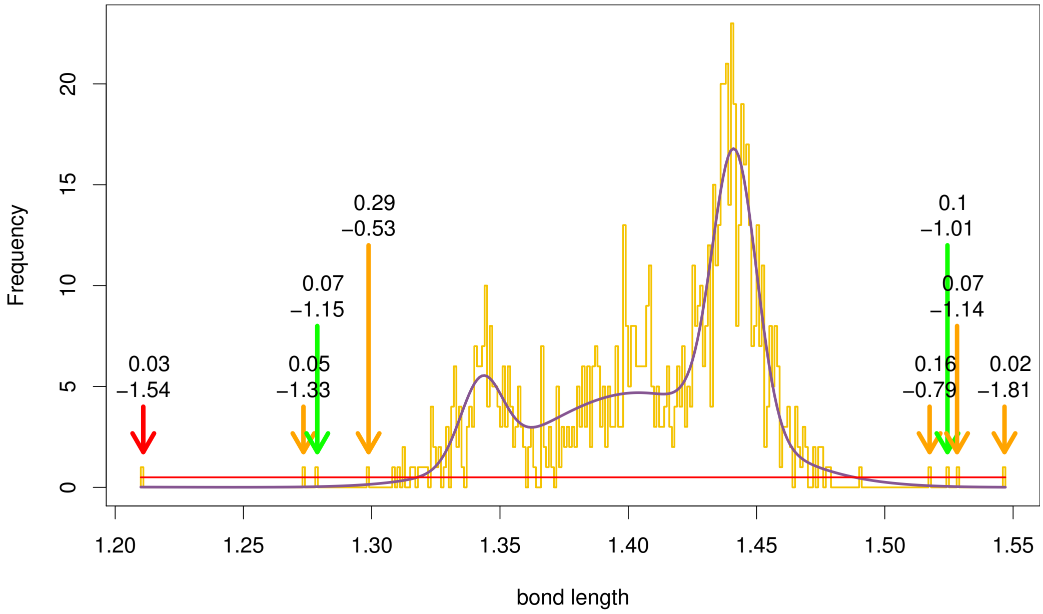


Figure 3.15: Bayes-based outlier detection in the class of polyacetylene C–C bond lengths. Yellow curve outlines the histogram, violet curve corresponds to the density of the best mixture model, whereas red line stands for the uniform distribution in the range. Arrows indicate observations with Bayes factor K lower than $10^{-0.5}$ (top numbers). Bottom numbers correspond to Schwarz criterion S, evaluated at possible outliers. Colours of arrows denote the results of manual outlier inspection: incorrect structure (red), possible symmetry-influenced biasses (orange), no defects (green).

- Grade 2. $10^{-0.5} > K > 10^{-1}$. Evidence against H_0 substantial.
- Grade 3. $10^{-1} > K > 10^{-1.5}$. Evidence against H_0 strong.
- Grade 4. $10^{-1.5} > K > 10^{-2}$. Evidence against H_0 very strong.
- Grade 5. $10^{-2} > K$. Evidence against H_0 decisive.

In the current study $K < 0.1$ is deemed enough to reject H_0 in favour of H_1, meaning that an observation is perceived as an outlier.

There are several techniques for the computation of Bayes factors, including easily computable asymptotic approximations, that can be derived from the MLE output, the simplest of them being the Schwarz criterion:

$$S = \log P(\text{data}|\hat{\theta}_0, H_0) - \log P(\text{data}|\hat{\theta}_1, H_1) - \frac{1}{2}(d_0 - d_1)\log n, \tag{3.36}$$

where $\hat{\theta}_k$ is the MLE under H_k, d_k is the dimension of θ_k, and n is the sample size. As $n \to \infty$, S may be viewed as a rough approximation to the logarithm of the Bayes factor [105].

An example of our approach could be demonstrated by applying it to the distribution of polyacetylene C–C bond lengths (Figure 3.15). Eight observations have Bayes factor $K < 10^{-0.5}$ (substantial evidence against an observation belonging to the proper distribution, or H_0). The shortest of the bonds (∼ 1.21 Å, COD entry 4020669) originates from a highly distorted benzene ring. Five of the observations (marked with orange arrows in Figure 3.15) correspond to bonds between symmetry related parts of a moiety, in most cases even between the equivalents of the same atom, therefore, they might be influenced by the summation of modelling errors.

3.8 Data organisation

3.8.1 Keyworded data format

As shown in Section 3.1, CIF format, albeit both human- and machine-readable, requires sophisticated parser and is in some occasions suboptimal for storage and exchange of data. Furthermore, handling of CIF data in most cases requires development of use case-specific software as opposed to using generic data handling programs, for example, `grep`, `cut` and `awk` from most of the GNU/Linux distributions. Therefore, to exchange and store the data we have used simpler linear format, to which we refer here as "keyworded data" format. This format is similar to PDB format in a way that each line describes an independent data record, always prefixed with a keyword and consisting of space-separated values that have meanings dependent on their position in the line. Unlike in PDB, fields are not fixed in their lengths and white space has no meaning, though white space characters are not allowed in values.

The keyworded data format is written by `cif_bonds_angles` (format `ATOM-CLASSES`), read and written by `cif_check_geometry` (format `CHECK-GEOMETRY`) and `calculate-models` (format `GEOMETRY-MODELS`). This format proved easy to read and write using `Perl`, `R` and `MySQL`, as well as spreadsheet software and standard tools from the most of GNU/Linux distributions.

3.8.2 Database versioning

The requirement for the reproducibility of searches in the evolving datasets at any time in the future is strongly felt. However, this is usually concurrent to the need to update the database to reflect the most recent state of the scientific knowledge. In 2016 the Research Data Alliance has published a recommendation package, suggesting scalable methodology for citation of dynamic data [195]. To accommodate these requirements we have introduced column `create_revision` in each of the raw data SQL tables as well as `*_history` counterparts ("historical tables") for each of them. In addition to `create_revision`, historical tables have `delete_revision` column to signify the moment of the accession of an entry into the historic table. To implement version control of the records we have redefined the following data management actions:

- **Insertion**. Each new entry in a database table with history tracking is assigned a revision number, which is stored in entry's column `create_revision`. More than one entry can share the same revision number.

- **Delete**. Each deleted entry is moved from the main table to `*_history`. During the transfer the current revision number is recorded in `delete_revision` of the `*_history` table.

- **Update**. Update operation is essentially separated into Delete and Insert actions, therefore, it is no different from the deletion of an old entry and insertion of a new, updated one.

Therefore, knowing a revision number (here assumed to be stored in @my_revision variable), a query executed before could be replayed by taking the union of the results of the following SELECT queries:

```
SELECT * FROM bonds WHERE
       create_revision <= @my_revision;
SELECT * FROM bonds_history WHERE
       create_revision <= @my_revision AND delete_revision > @my_revision;
```

The first query is used to select all entries that were inserted no later than at the given revision while the second query pulls deleted entries from the historic table that were in the main table at the revision in question. This method is similar to the MCVV database method of relational databases [196], the only difference is that our method is implemented using the database schema, not by the underlying database engine, therefore, easily portable between MySQL, SQLite and PostgreSQL database management systems.

3.9 Web interface

A Web-based user interface to the database of molecular geometry was developed based on Common Gateway Interface (CGI) scripts, developed using Perl programming language[17]. The user interface consists of search[18], parameter distribution preview and validation[19] interfaces. In order to make the Web interface compatible with most of the browsing software, basic functionality is provided via on-demand generated static HTML pages with some additional functionality presented via JavaScript.

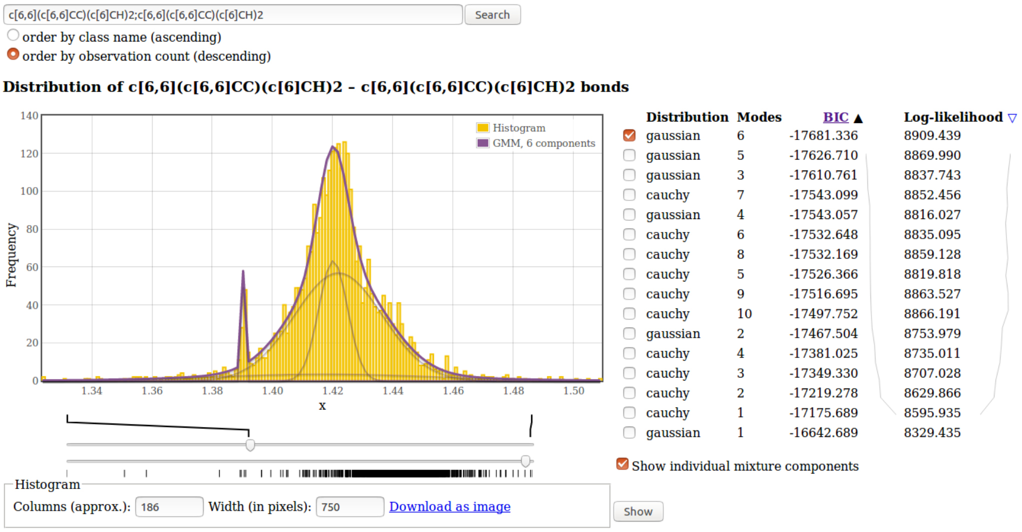
3.9.1 Search interface

Two ways to browse the parameter distributions were implemented. A graphical one uses JSME [197] JavaScript applet to provide a tool to draw a structural diagram, which is upon submission translated into a SMILES string. Open Babel package [198] is in turn used to convert supplied SMILES string into a molecular graph. Aromatic atoms (denoted by lowercase initial element characters in SMILES strings) are treated as being in planar environments only if they have three or more covalent neighbours, as the concepts of aromaticity in SMILES and planarity in the atom types of this study (see Section 3.5.1) are different. Atom types of produced molecular graph are then shown to the user next to the entered diagram allowing to select bonds, angles and dihedral angles of interest. When fragment of interest is selected, a preview of fragment's geometry is displayed (see Section 3.9.2). Another method to browse the distributions of bonds, angles and dihedral angles is to enter accordingly two, three or four atom type strings. As the construction of these strings "by hand" is tedious and error-prone, this method is intended to be used mostly by the other applications as an API.

[17] Available under the GPL2 free software license at svn://www.crystallography.net/molecules-in-COD/trunk, this study refers to source revision 1499

[18] Located at http://www.crystallography.net/geometry/

[19] Located at http://www.crystallography.net/geometry/cgi-bin/check_geometry.pl



c[6,6](c[6,6]CC)(c[6]CH)2;c[6,6](c[6,6]CC)(c[6]CH)2 Search

○ order by class name (ascending)

● order by observation count (descending)

Distribution of c[6,6](c[6,6]CC)(c[6]CH)2 - c[6,6](c[6,6]CC)(c[6]CH)2 bonds

	Distribution	Modes	BIC ▲	Log-likelihood ▽
☑	gaussian	6	-17681.336	8909.439
□	gaussian	5	-17626.710	8869.990
□	gaussian	3	-17610.761	8837.743
□	cauchy	7	-17543.099	8852.456
□	gaussian	4	-17543.057	8816.027
□	cauchy	6	-17532.648	8835.095
□	cauchy	8	-17532.169	8859.128
□	cauchy	5	-17526.366	8819.818
□	cauchy	9	-17516.695	8863.527
□	cauchy	10	-17497.752	8866.191
□	gaussian	2	-17467.504	8753.979
□	cauchy	4	-17381.025	8735.011
□	cauchy	3	-17349.330	8707.028
□	cauchy	2	-17219.278	8629.866
□	cauchy	1	-17175.689	8595.935
□	gaussian	1	-16642.689	8329.435

☑ Show individual mixture components

Histogram

Columns (approx.): 186 Width (in pixels): 750 Download as image

Figure 3.16: Distribution of naphthalene C_{4a}–C_{8a} bond lengths. Histogram shows the distribution of bond lengths in range 1.153–1.513 Å. Drawn on top of the histogram is the model with the smallest BIC. Four of six Gaussian mixture components are easily identifiable (thin violet line). Other models and their parameters are listed on the right-hand side of the histogram.

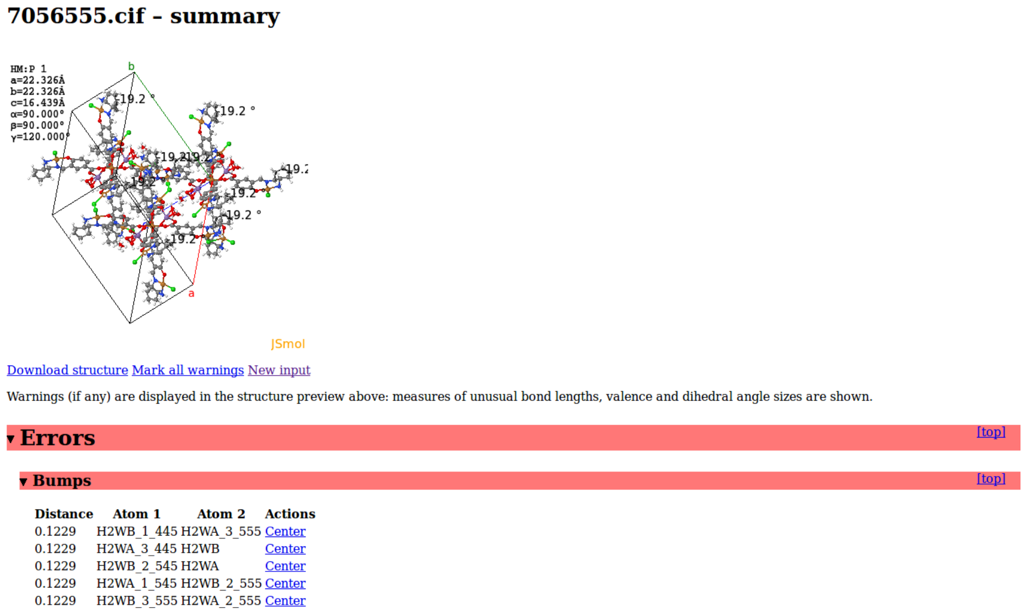
An option exists to view the distributions of all parameters of the entered structure. Upon submission of a drawn structural diagram, idealised coordinates are generated from the SMILES string using Open Babel. These coordinates are then forwarded to the validation interface with a flag set to display histograms for all the parameters.

3.9.2 Molecular geometry browser

A preview of the distribution for each of the parameter classes is facilitated in a form of Web page with an interactive histogram (Figure 3.16). All EM-generated statistical models can be visualised as fitted density function curves on top of the histogram; likelihood, BIC and other parameters of the models can be compared in a sortable table. As the user may need to distinguish between up to twenty different density curves, drawn on top of the histogram, we have used twenty colours of maximum contrast, as suggested by Kelly (1965) [199].

Regions of a histogram may be investigated by isolating observations that produce them. By clicking a bar of the histogram user can access a table listing all the constituent observations, providing atoms and COD entries of origin. Entries of the table could be selected: upon the selection of an entry a Jmol [148] preview of its crystal structure is shown with the observation marked in it.

To provide interactive display of the histogram while maintaining the same functionality for Web browsers without (or with disabled) support of JavaScript, we have implemented the histogram display using flot [200] (with JavaScript) and also using a selectable HTML image map (without/with disabled JavaScript). Both variants share the same functionality and can be used interchangeably.



7056555.cif – summary

Download structure Mark all warnings New input

Warnings (if any) are displayed in the structure preview above: measures of unusual bond lengths, valence and dihedral angle sizes are shown.

▾ Errors [top]

▾ Bumps [top]

Distance	Atom 1	Atom 2	Actions
0.1229	H2WB_1_445	H2WA_3_555	Center
0.1229	H2WA_3_445	H2WB	Center
0.1229	H2WB_2_545	H2WA	Center
0.1229	H2WA_1_545	H2WB_2_555	Center
0.1229	H2WB_3_555	H2WA_2_555	Center

Figure 3.17: Excerpt of validation report for COD entry 7056555. Distorted benzene rings are perceived as nonplanar and their dihedral angles, treated as unusual, are highlighted. Five pairs of bumping hydrogen atoms are reported as errors.

3.9.3 Validation interface

A Web interface was built for `cif_check_geometry` and `cif_voids`. Once uploaded, a CIF file is checked by these programs and the output messages signaling about unusual geometric features are shown. The structure is displayed using `Jmol` applet. Unusual bond lengths, angle sizes and voids are marked in the applet (see Figure 3.17 for example). Each unusual geometric parameter is displayed with a reference histogram of all the observations of the same class. Histograms for all "usual" geometric parameters can also be ordered for reference. A link to an interactive preview of the molecular geometry distribution is attached to every histogram in the results page.

Both known parameter classes without calculated models and completely unseen classes are included in the generated reports. Notification of the former kind means that a class is underrepresented in the database, whereas the latter kind signifies previously unseen parameter, arising from either genuine or erroneous connectivity.

Chapter 4

Results

4.1 CIF parser

4.1.1 Overview

The most straightforward use of our error-correcting parser is the maintenance of the COD. The capability of error detection and correction is now employed both in the automatic data deposition interface and by the maintainers of the COD to curate the data. Besides, the strict mode of the `COD::CIF::Parser` is used to ensure that all CIF files in the COD conform to the CIF description as provided by the IUCr.

Program `cifparse` was developed in `C` language to provide the command line interface for the parser. The program was in turn employed to check the syntactic correctness of every CIF file in the COD. In order to implement an additional check for syntax correctness of added or modified CIF files, we have developed a pre-commit hook for the COD `Subversion` [201] repository. By doing so we have effectively prevented changes that introduce syntax errors in CIF files from being accepted into the COD. An equivalent check is performed at the COD data deposition Web site before doing any further semantic checks.

Nevertheless, requirement for strict compliance to the CIF format while accepting structure reports from researchers or from published sources might do more harm than good. Deviations from the CIF standard are common in supplemental material, and most of them could be corrected in automatic manner, employing "common sense" heuristics. Here the error-correcting capability of `COD::CIF::Parser` proves handy, performing the most of the required changes without human supervision. It must be noted that when used on the COD server, output of our tools is directed to the server log files for further inspection. When used interactively the messages are presented to the depositor in the Web interface. Usage of `COD::CIF::Parser` and `cod-tools` allowed to reduce the human effort to maintain the COD. Furthermore, these tools support the ongoing data curation in the database. Apart from these, `COD::CIF::Parser` could be employed for:

- **Format conversion** – `COD::CIF::Parser` allows to convert the CIF format to other widely-used lossless data formats (i.e. JSON, as implemented in `cif2json` from `cod-tools`) or field-specific formats (i.e. input formats of DFT codes);

- **Crystallographic computations** – COD::CIF::Parser enables reading CIF data into C, Perl, Python, and potentially Fortran programs. Our parser serves as a base for AiiDA workflows for preparation of crystallographic data for DFT calculations [53];
- **Validation** – IUCr has defined protocols for the validation of CIF files against dictionaries, and dictionaries in turn against their Dictionary Definition Language (DDL) dictionaries, all of them use CIF as carrier format.

4.1.2 Behaviour

Agreement on common data formats and strict compliance to their specifications are of crucial importance for data exchange between different software pieces as well as different researchers. Deviations from the format precipitate unnecessary interruptions in data processing, a need of human participation and, in the worst case, corrupt data and erroneous results. Thus we do our best to implement the CIF parsers precisely to the specification of the IUCr. We have, though, decided to require less that the IUCr specification in one aspect. Our parser tolerates lines of arbitrary length, reporting them only if the check for line length is explicitly requested. We justify this decision with the fact that most of the modern programming languages (we use Perl and C, but the same features exist at least in Python, Java, Julia) are able to seamlessly process strings of virtually any length, therefore limiting them in this case would be an additional burden. Moreover, reading in long lines does in no way lose or corrupt the data. On the contrary, limiting line length and discarding symbols past the limit may cause the loss of data. Therefore, rejection of otherwise conforming files due to long lines is detrimental, unless there is a need to check the files before processing them with Fortran programs which use fixed-length buffers. We have decided to make COD::CIF::Parser permissive, at the same time writing out CIF files as closely adhering to CIF 1.1 specification as possible. In this way we achieve that our software is capable of reading the maximum number of inputs, including its own, and producing output suitable for the largest number of other programs.

To compare the parsing behaviour of different CIF parsers we have carried out syntactical analysis of two sets of synthetic CIF files:[1] test cases as published in Merkys et al. (2016) [67] and a new set of test cases, incorporating all features from the test suite of vcif. We have selected to compare a set of widely used open-source command-line compatible CIF parsers, namely ase (version 3.14.1), cif2cif (version 2.0.0), cif_api (version 0.4.2), gemmi (GIT commit 860d285), PyCIFRW (version 4.2), ucif (revision 23314), vcif (version 1.2), vcif2 (version 0.9.3.1), ZINC (version 1.12), and our CIF parsers (revision 5518). We have also checked the parser of pymatgen, but decided to exclude it as too specific: it employs regular expressions to determine symmetry and coordinate data from CIF files ignoring the remaining content. As CIF is a subset of STAR format and STAR parsers are able to parse CIF, two STAR parsers, STAR::Parser (version 0.59) and StarTools (version 0.2.0), were also added to the set of analysed parsers. As our intent was to investigate the default behaviour of bespoke parsers, we have not used any command line options or arguments, except for COD::CIF::Parser[2]. The results of the

[1] All test cases as well as the results are accessible on the Web at https://github.com/cod-developers/CIF-parsers

[2] We have explicitly set command line option --report-long-items for COD::CIF::Parser to enable line and

Test	CIF conforming?	ase	cif2cif	cif_linguist	COD::CIF::Parser	COD::CIF::Parser$_{\text{fix}}$	gemmi	PyCIFRW	STAR::Parser	StarTools	ucif	vcif	vcif2	zinc
ascii-127.cif	×			×	×	/	×				×	/	/	
byte-order-mark.cif	×			×			×	/			×	×	×	×
closing-bracket.cif	×			×	×			/		/	×			
comment-only.cif												/	×	
dos-ctrl-z.cif	×		×		×	/	×					×	×	×
duplicate-tags-different-cases.cif	×		/	×	×	×	×	×						
duplicate-tags-different-values.cif	×		/	×	×	×	×	×				×		
duplicate-tags-same-values.cif	×		/	×	×	/	×	×				×		
empty-datablock.cif												/	/	×
empty-datablock-name.cif	×			×	×	/	×	×		/	×	×	/	
empty-file.cif										×		/	×	
form-feed.cif	×		×	×			×	×					/	
global.cif	×		×	×	×	×	×	×			×			×
long-line.cif	×		/	×	/							/	/	
loop-without-tags.cif	×	/	×	×	×	×	×		–		×	×	×	×
loop-without-values.cif	×	/	×	×	×	×	×		–		×	×	×	×
missing-closing-quote.cif	×		/	×	×	/	×	×	/	/	×	×	/	
missing-data-header.cif	×		/	×	×	/	×	×			×	×	/	
non-ascii.cif	×			×	×	/	×				×	/	/	
non-ascii-in-comment.cif	×			×	/	/					×		/	
null-symbol.cif	×			×	×	×	×				×	/	×	
_refine_ls_extinction_expression.cif				×										
single-quote-in-value.cif														
stray-values-at-start.cif	×			×	×	/	×	×			×	×	×	×
tag-immediately-following-textfield.cif	×		×	×	×	×	×			/		×	×	
textfield-in-loop.cif		/												
textfield-no-closing-semicolon.cif	×			×	×	×	×	×	×	/	×	×	×	–
unquoted-loop-prefix.cif			×				×		×	/			×	×
value-immediately-following-textfield.cif	×			×	×	×	×			/			×	
value-starting-with-bracket.cif	×			×	×			/		/	×			
value-starting-with-closing-bracket.cif	×			×	×			/		/	×			
value-starting-with-dollar.cif	×			×	×	×	×	×			×			
vertical-tab.cif	×		×	×			×	×			×		/	
whitespace-placement.cif		/												
wrong-number-of-loop-values.cif	×	/	×	×	×	×	×	×	×		×	×	/	

Table 4.1: Comparison of CIF 1.1 parsers. Crosses ("×") denote detected parsing failures, slashes ("/") denote emitted warnings, and dashes ("–") mark cases when parsing programs hang for an indefinite amount of time and have to be terminated manually.

analysis are given in Table 4.1. We have identified four possible outcomes of file parsing: error (parser program failure), warning (parser completes the parsing and reports issues with syntax/semantics), failure to terminate (program hangs for unreasonably long time) and success. It is important to note that parsing failure could mean either that a parser has recognised an error and terminated or the program of a parser failed to proceed due to inability to cope with the state it arrived at. Moreover, checking the resulting CIF representations was deemed out of scope for this comparison of parsers, therefore successful parsing does not necessary mean that a parser correctly reads in and represents the input file in its internal representation.

It is interesting to compare the reaction of different CIF 1.1 parsers to various test suites of CIF files. Indeed, most parsers seem to be capable to read the conforming CIF files and identify the incorrect ones. `ase` and `ZINC` prove to be robust, low-level tools, being able to parse CIF files with forbidden symbols, closing quotes and missing headers. However, `ase` parser is unable to process two correct CIF files from the test suites and `ZINC` becomes trapped in

data item length checks, which are disabled by default. This was done in order to make `COD::CIF::Parser` behaviour as close to the other parsers as possible.

an infinite loop after running into unterminated text field and has to be stopped manually. cif2cif detects and reports some of the syntax and semantic errors (for example duplicated data items and overlong lines), but is also insensitive to some symbols forbidden in the CIF values. The parser does not accept CIF values starting with "loop_", which are allowed by the syntax definition. cif_linguist reports all nonconforming CIF constructions of the test suites except ^Z symbol and overlong data item names. In addition, a couple of false positives is reported: DOS line ending symbols and valid unquoted strings. gemmi CIF parser stands very close to cif_linguist. gemmi detects ^Z, correctly processes DOS line endings and does not impose restrictions on line and data item name lengths. Moreover, parser's requirements for character sets are lax. PyCIFRW parser also seems to relax the limitations on the character set and line lengths. Nevertheless, overlong data item names are reported. The parser does not warn about missing data item or value parts of CIF loops and missing white space separators following the text fields. ucif is capable of reporting most of the nonconforming tests with the exceptions of duplicated data items, missing mandatory white space following text fields and overlong lines and data item names. vcif proves to be more sensitive to borderline CIF cases than most of other parsers. Apart from most of the errors, vcif warns about empty files and data blocks, long lines and data item names. However, data item names which differ only in character cases are not reported, although mandated as case-insensitive. Missing white space between text fields and following values is not reported, as well as reserved symbols in unquoted data values. vcif2 relaxes some restrictions of vcif: duplicate data items are not reported at all, as well as unterminated text fields and data items immediately following text fields. However, warnings are emitted concerning non-ASCII symbols in comments, values starting with "loop_" and forbidden white space symbols. Both STAR format parsers, STAR::Parser and StarTools, are less prone to issue warnings, however this might be because some limitations of the CIF format are not imposed on STAR. Nevertheless, STAR::Parser seems to be the least robust as it gets caught in an endless loop upon encountering malformed loops.

On one hand, such diversity of the parser behaviour possibly reflects the differing requirements and purposes intended by their developers. On the other hand, it allows an insight into their engineering solutions and trade-offs. In the COD, for example, the task is to retrieve as much as possible reliable data from publication supplements and depositor-uploaded files, therefore, permissive CIF parser is required. As overlong lines and byte order marks (BOM) do not corrupt the data, we accept CIF files with these features, but try to adhere to the CIF standard when producing the output. It is evident that other uses might require different behaviour.

Our parsers operate in a manner similar to cif_linguist and gemmi. Strict mode of COD::CIF::Parser detects all syntax and semantic errors except UTF-8 BOM and a pair of white space symbols that are forbidden in CIF (vertical tabulation and form feed, decimal ASCII values 11 and 12, accordingly). While it is true that BOM constituent bytes as well as the other white space symbols fall into the restricted part of the character set of CIF 1.1, we argue that they might get inserted by some text editors from time to time and should be deemed artefacts. The same treatment of BOMs is suggested in the CIF 2.0 standard definition [72], therefore we feel that it could be also applied to its predecessor. On the other hand, COD::CIF::Parser$_{\text{fix}}$ is much laxer, processing the most of the inputs of the test suites, repairing them and issuing

Parser	Run time (min)
ase	90.69
cif2cif	31.54
cif_linguist	27.05
COD::CIF::Parser	25.53
COD::CIF::Parser$_{\text{fix}}$	16.07
gemmi	12.25
ucif	16.61
vcif	15.77
zinc	16.16

Table 4.2: Total parsing time (in minutes) of CIF files from the COD

warnings where appropriate. These features proved very useful in reading non-conforming CIF files, which occur even in the data from peer-reviewed publications. All in all, diverse behaviour of different parsers demonstrates the handiness of having several parsers available: firstly, comparison of different parsers lets spot bugs in our code; secondly, a parser with desired trade-offs (compatibility with used programming languages, performance, maintenance costs and dependencies) can be selected on demand.

4.1.3 Performance

The comparison of CIF parser performance was evaluated by parsing 382 807 CIF files from the COD (all entries from revision 199925 totalling in ~49 GB) on an unloaded computer with 31 GB of RAM and 16 × Intel(R) Xeon(R) CPU E5-2450 v2 @ 2.50GHz, running Debian GNU/Linux 8.6 (jessie), with gcc version 4.9.2, Perl version 5.20.2, Python version 2.7.9. Versions of the parsers were the same as listed in Section 4.1.2. Wall clock timings are presented in Table 4.2 for comparison (we have decided to exclude PyCIFRW and vcif2 from the performance benchmark due to the observation that the parsers's parsing time depends quadratically on the sizes of CIF text fields, possibly due to ineffective memory management). Our tests indicate that our C parser is one of the fastest in the field, while at the same time capable of recognising most of the CIF features defined by the IUCr CIF grammar.

4.1.4 Conclusion

A parser for CIF format was implemented in Perl programming language and later optimised as a C library with bindings for Perl and Python programming languages. According to our tests, resulting COD::CIF::Parser is one of the fastest and the most accurate existing CIF parsers. Comparing results of various parsers was essential in developing and testing our parsers, as some test cases elicit different behaviour from various parsers. It must be noted that emerging differences are not necessarily manifestations of errors as different behaviour might be intentional. In particular, error-correcting mode of COD::CIF::Parser proved necessary to import and repair non-conforming CIF files from external sources. Thus our parser turned out handy in managing large collections of crystallographic data. Furthermore, high speed parsing and compatibility with other programs in Perl, C and Python languages allow usage of COD::-CIF::Parser in various crystallographic software, and we expect it to facilitate easier data exchange between researchers.

We have developed CIF 1.1 and CIF 2.0 parsers in parallel. Thus we have implemented a separate lexer and a grammar for each CIF format version. As the development costs of software grow worse-than-linearly with the system size [202], the maintenance of two sets of parsers will result in increased costs and will require considerably more effort than a single parser. Nevertheless, as CIF 2.0 parser is backwards incompatible, the current situation is inevitable.

4.2 Geometry library

4.2.1 Overview

A total of 382 807 structures from COD revision 199925 were processed using the methods described above. Calculations took almost two full days on 140 cores of 240 × Intel(R) Xeon(R) CPU E5-4650 v2 @ 2.40GHz shared memory machine with 1.1 TB RAM, under CentOS 6.8 operating system. At least one pair of bonded atoms was extracted from around 320 000 structures, the rest being either empty, skipped or containing isolated atoms. To better understand possible causes and correlations of certain structure parameters, we have developed a program to automatically assign labels to entries in the COD by analysing the outputs of `cif_molecule` and `cif_bonds_angles`. The following labels are defined:

- `PROCESSED` – a structure is processed; this label is supposed to be assigned to every structure regardless its properties or processing outcome;
- `BUMPS` – a structure contains at least one bump;
- `CIF_EMPTY` – the output of `cif_molecule` for a structure is empty;
- `CIF_KILLED` – the process of `cif_molecule` was killed due to the overuse of either memory or time resource;
- `DISORDER_DEPEND_CLASSES` – a structure contains at least one atom with an atom type dependent on disorder;
- `DISORDERED` – a structure is disordered;
- `DISORDER_SPEC_POSN` – a structure contains at least one atom deemed disordered around special position;
- `DUPLICATE` – a structure is a duplicate;
- `MARKED_DISORDER` – a structure contains at least one site marked as disordered via `cif_mark_disorder`;
- `NO_ATOMS` – a structure does not contain any atoms;
- `POLYMER` – a structure is deemed to be a polymer;

- `TAB_EMPTY` – the output of `cif_bonds_angles` for a structure is empty and the reason is given;
- `TAB_NOT_READY` – the output of `cif_bonds_angles` for a structure is empty and the reason is not given;
- `UNK_CHEMTYPE` – a structure contains at least one atom of unrecognised chemical element.

Table 4.3 displays co-occurrence of flags, automatically assigned to the processed structures. According to the table, geometric observations are not extracted from half of disordered structures, mostly due to disorder-dependent atom types. Over a third of polymer structures as well as structures with bumps are also skipped.

4.2.2 Atom types

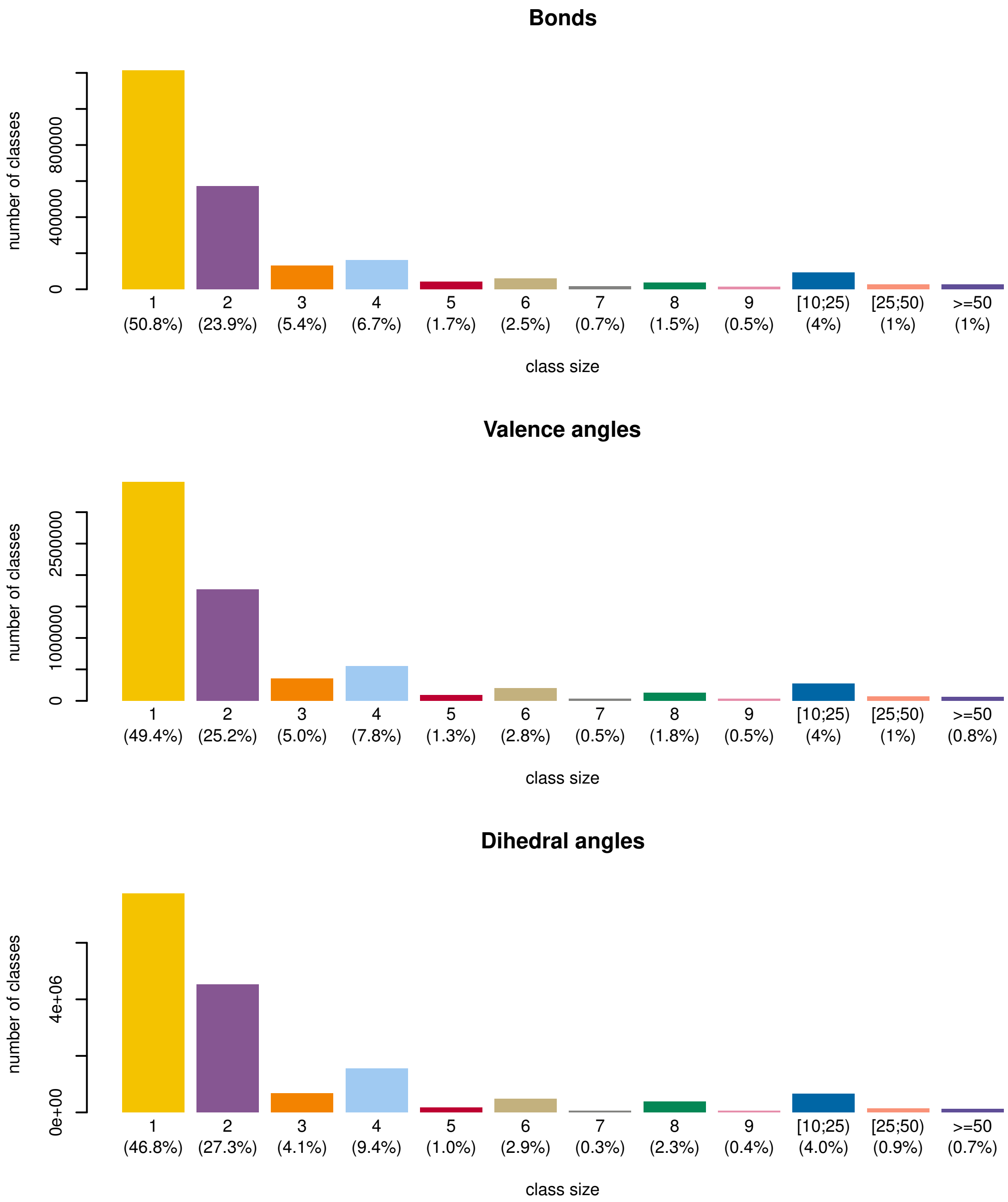
There is a total number of 1 073 426 level 2 atom types in the COD, around 4 times more than reported previously [115]. Most of the types, 822 860 belong to the "organic subset" as defined by SMILES (B, C, N, O, P, S, F, Cl, Br, I), 374 303 out of them are of carbon atoms alone. Carbon-to-all ratio of 35% is much larger than 14%, reported by PURY [7].

The average number of distinct atom types per crystal structure is 20. Structures with bumps on average have around 26 types. This leads to the conclusion that bumps introduce "impurities" in chemical environments thus leading to more singleton atom types.

	PROCESSED	BUMPS	CIF_EMPTY	CIF_KILLED	DISORDER_DEPEND_CLASSES	DISORDERED	DISORDER_SPEC_POSN	DUPLICATE	MARKED_DISORDER	NO_ATOMS	POLYMER	TAB_EMPTY	TAB_NOT_READY	UNK_CHEMTYPE
PROCESSED	■	11	2	1	9	21	2	1	6	0	23	16	4	1
BUMPS	*	■	0	0	26	44	5	1	10	0	34	34	7	0
CIF_EMPTY	*	0	■	44	0	0	0	0	26	15	0	*	0	37
CIF_KILLED	*	0	*	■	0	0	0	0	13	0	0	*	0	0
DISORDER_DEPEND_CLASSES	*	31	0	0	■	*	5	0	37	0	41	*	0	0
DISORDERED	*	24	0	0	45	■	9	1	24	0	30	51	5	0
DISORDER_SPEC_POSN	*	29	0	0	28	*	■	1	1	0	12	30	1	0
DUPLICATE	*	16	0	0	0	23	2	■	10	0	40	*	0	0
MARKED_DISORDER	*	21	9	2	62	91	0	1	■	0	83	91	19	7
NO_ATOMS	*	0	*	0	0	0	0	0	0	■	0	*	0	0
POLYMER	*	16	0	0	17	27	1	1	20	0	■	35	17	0
TAB_EMPTY	*	24	12	5	58	66	3	4	31	2	50	■	25	5
TAB_NOT_READY	*	19	0	0	0	26	0	0	25	0	97	*	■	0
UNK_CHEMTYPE	*	0	*	0	0	0	0	0	55	0	0	*	0	■

Table 4.3: Co-occurrence of flags. Intersections of rows and columns show the percentage of all structures with the row flag having also the column flag. Symbol * corresponds to 100%.

Automatically generated atom types can be subjected to validation based on the "common chemical sense". These validation criteria could be then expressed as a system of rules and



Bonds

Valence angles

Dihedral angles

Figure 4.1: Breakdown of class sizes.

applied to detect nascent improbable atom types. For example, around 20 000 atom types of hydrogen atoms bound to two other atoms were detected, most of them clearly being artefacts. 1500 atom types were detected for planar atoms with five or more covalent neighbours, what is also quite unusual. Participation in hundred or more rings might also seem odd, however, around 350 such atom types were found. It should be noted, though, that the most of these types will be singletons, mostly caused by the presence of bumps.

Most of bond, angle and dihedral angle classes are singletons (Figure 4.1 presents breakdown of classes by the number of observations), with only 1% or less of them having 50 or more observations. This can be attributed to the overwhelming diversity of chemical environments in small molecule crystal structures.

Apparent limitations of current atom typing methods are due to selected classification depth and maximum ring size. It is out of scope of the current study to determine whether/when significant differences are introduced by further covalent neighbours or by closedness (cyclicity) of covalent chains of more than seven atoms. However, it is clear that including more information into atom types would result in a finer partition of chemical environments causing underrepresentation of many of them. Therefore, current method is a compromise between precision and redundancy.

Decision to ignore atom aromaticity and bond orders is bound to increase number of prominent modes in the distributions of bond lengths. Admittedly, some of the required information could be taken from author-provided chemical names, thus avoiding the need to use coordinate-based heuristics. This could be achieved, for example, by using *OPSIN* to convert author-provided chemical names to SMILES strings and Morgan algorithm to overlay coordinates- and SMILES-derived molecular graphs. However, chemical names are present for one in three entries in the COD, moreover, for as much as 14% of chemical names there might be mismatches [21].

The applicability of VBT to the compounds outside the organic subset is also questionable. Metal coordination could be represented very well by the current approach, provided that all pairs of metal and coordinating atoms are considered as bonded. For example, ferrocene, a sandwich compound, has Fe–C contacts of 2.04 Å in length, that are recognised as bonds by the current software. However, the current algorithm is not able to tell coordinated atoms from their neighbours should they be located close enough to the coordinated metal, nor are the atom types able to distinguish different coordination types of the same coordination number [203]. There is no clear criterion to tell whether two metal atoms are bonded in crystal structure. Ionic bonds, that usually allow greater variations in geometry, should not be treated as bonds by the current algorithm.

4.2.3 τ angle

The value of τ angle [109] was chosen as a touchstone to compare the results of this study with the ones performed before. Atom types relevant to protein backbone were composed to select the observations of τ angle in the COD. Average τ angle sizes are reported in Table 4.4 alongside the results of other studies. It is evident that the outcome of the analysis from the COD is comparable to the earlier works, although τ angles are generally larger. Compared

Source	Gly	Other except Gly & Pro
EH 1991 [98]	112.5 ± 2.9	111.2 ± 2.8
LMT 1993 [205]	112.19 ± 3.64	110.77 ± 3.29
EH 2001 [206]	113.1 ± 2.5	111 ± 2.7
TV 2010 [109]	113.1 ± 3.4	111 ± 3
this study	113.5 ± 1.9	111.3 ± 2.7

Table 4.4: Comparison of protein backbone τ angles.

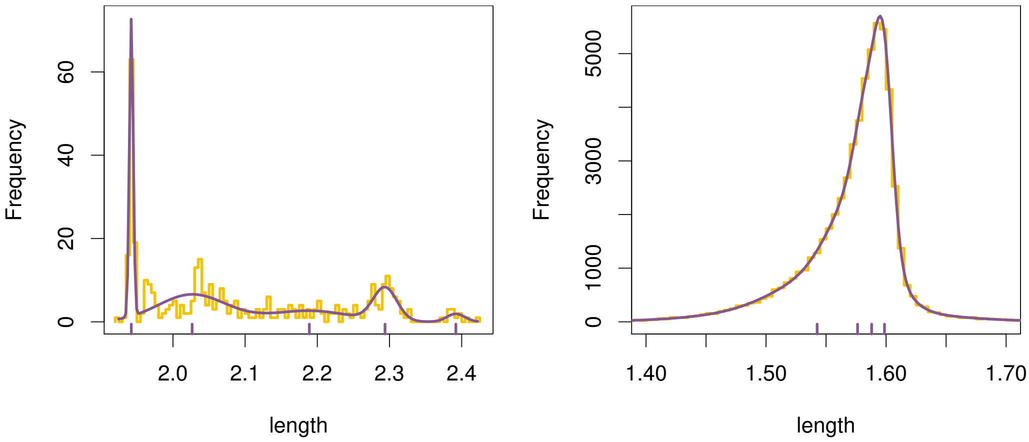


Figure 4.2: Bond lengths in coordinate complexes: **left)** Cu–O in six-coordinate copper-water complexes, **right)** P–F in hexafluorophosphate. Histogram is outlined in yellow, density of the best model in violet.

to the study of Balasco et al. (2017), who analysed τ angle (at residues except glycine and proline) and its dependence on protein secondary structure, τ value derived from the COD is very similar to one averaged over all observations ($\sim$ 111.3°) [204]. Very small number of occurrences in the COD (11 hits of glycine-like and 36 hits of the rest amino acids except proline) could be accounted for the discrepancies. Despite this fact, the standard deviation of glycine is smaller than observed before, hinting the absence of outliers. This study defines eight non-empty classes of τ angle environment depending on C_β: glycine (`NCH`, 11 observations), alanine (`CH3`, 15 observations), linear alkane chain (`CCHH`, 10 observations), threonine (`CCHO`, 3 observations), β-branched residues (valine and isoleucine, `CCCH`, 4 observations), serine (`CHHO`, two observations), cysteine (`CHHS`) and *tert*-leucine (`CC3`), with a single observation each.

4.2.4 Jahn–Teller effect

The evidence of Jahn–Teller effect, responsible for the elongation of axial bonds in six-coordinate copper-water complexes, in the COD is similar to one previously seen in the CSD by Harding (1999) [157] (Figure 4.2, left). However, short bonds in the COD are $\sim$ 0.2 Å shorter and a substantial part of observations of longer bonds are missing due to Cu–O covalent cutoff of roughly 2.5 Å. A mixture model of five normal components (Figure 4.2, right; density shown in violet) was chosen to approximate the distribution of bonds.

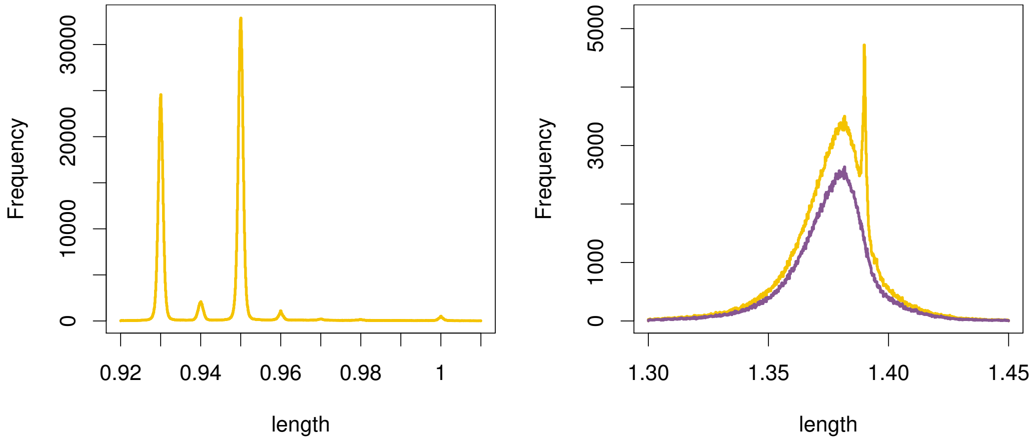


Figure 4.3: Refinement bias in benzene bond lengths: **left)** C–H and **right)** C–C. Yellow line outlines the histogram of all observations in the COD, whereas violet line represents the histogram of observations from the independently refined C–C bonds.

4.2.5 Refinement bias

Laskowski et al. (1993) noticed that geometric libraries and software used for refinement usually introduce bias so significant that a very crude ruleset detects the settings used with the accuracy of 95% [205]. Possible footprints of different refinement settings are also visible in data in the COD. For example, almost discrete distribution of benzene C–H bond lengths (Figure 4.3, left) contains five prominent peaks that are located at 0.93, 0.94, 0.95, 0.96 and 1 Å, indicating putative target values used during the refinement. Bias may be averted by considering only observations that are explicitly marked as refined without restraints (values of CIF data item `_atom_site_refinement_flags` or related items are "`.`"). Histogram of benzene C–C bonds from only non-restrained fragments lacks otherwise anomalous peak at ∼ 1.39 Å (Figure 4.3, right), highly likely caused by bonds being refined to this ideal value. We have investigated C–H bond lengths in order to identify the systematic bias introduced by refinement software. However, there seems to be no straightforward correlation between used values and programs in the COD, possibly due to different libraries used with the same software.

4.2.6 Validation of novel structures

To evaluate our method of structure validation we have investigated 100 novel structures that were deposited to the COD after we had derived the current geometry library. In parallel, we have used `PLATON` in order to obtain results for cross-validation of our method. Out of 100 structures, 9 were considered to contain bumps. 12 structures were unable to be processed by `cif_check_geometry` due to the need of extensive calculation resources. 23 structures of 100 had at least one unusual feature according to our method, while `PLATON` had warnings for 18 structures. 6 structures had warnings by both methods. The full list of structures with warnings in this subset is given in Table 4.5. Below we present manual analysis of ten structures with the most warnings by our method.

The most of the analysed structures have unusual parameters arising due to possibly poor

COD ID	d_C	α_C	ϕ_C	$\sum_C$	$\sum_P$	Bumps
1546823	11	17	0	28	1	
2020874	0	18	0	18	0	6
7229026	0	7	0	7	0	
4002839	0	6	0	6	0	
1546859	2	3	0	5	2	
4126340	5	0	0	5	0	
7228987	0	0	5	5	0	
1546887	0	4	0	4	4	
7056555	2	1	1	4	0	10
1546858	1	2	0	3	0	
4126332	1	2	0	3	0	
7229031	0	0	3	3	0	
1546862	0	2	0	2	6	
7044067	0	2	0	2	1	2
1546918	0	2	0	2	0	
7155853	0	2	0	2	0	
7228989	0	0	2	2	0	
7229032	0	2	0	2	0	
7120611	1	0	0	1	1	
1546820	0	1	0	1	0	
7044036	0	1	0	1	0	
7044045	0	1	0	1	0	
7228985	0	1	0	1	0	
1546884	0	0	0	0	4	
1546885	0	0	0	0	4	
1546889	0	0	0	0	4	
7229014	0	0	0	0	4	
7044095	0	0	0	0	3	3
7120612	0	0	0	0	3	
7056573	0	0	0	0	2	
1546848	0	0	0	0	1	
1546913	0	0	0	0	1	
7044089	0	0	0	0	1	
7229036	0	0	0	0	1	
7229042	0	0	0	0	1	

Table 4.5: Validation of 100 novel structures from the COD. Structures without warnings were omitted for brevity. d_C, α_C, ϕ_C are accordingly counts of bonds, bond angles and valence angles deemed unusual by our method; $\sum_C$ and $\sum_P$ are accordingly total counts of warnings by our method and `PLATON`.

COD ID	d_C	α_C	ϕ_C	$\sum_C$	$\sum_P$	Bumps
2214032	4	5	0	9	1	
2214266	7	0	0	7	1	
2214740	7	0	0	7	1	
2214731	6	0	0	6	1	
2215545	3	3	0	6	0	
2214494	4	1	0	5	0	
2215330	1	3	0	4	1	
2211060	1	3	0	4	0	
2215738	4	0	0	4	0	
2216003	2	2	0	4	0	
2216548	3	1	0	4	0	
2216555	3	1	0	4	0	
2216233	1	2	0	3	1	
2214035	3	0	0	3	0	
2214948	0	2	1	3	0	
2215165	1	1	1	3	0	
2215546	3	0	0	3	0	
2215724	3	0	0	3	0	
2215994	0	3	0	3	0	
2216004	3	0	0	3	0	
2215725	0	2	0	2	1	
2215733	1	1	0	2	1	
2214020	0	2	0	2	0	
2215167	2	0	0	2	0	
2215544	0	0	2	2	0	
2216567	1	1	0	2	0	
2214265	1	0	0	1	1	
2211463	0	1	0	1	0	
2211708	0	1	0	1	0	1
2211760	1	0	0	1	0	
2214728	0	1	0	1	0	
2214858	0	1	0	1	0	
2214947	1	0	0	1	0	
2214955	1	0	0	1	0	
2215999	0	1	0	1	0	
2216217	1	0	0	1	0	
2214262	0	0	0	0	2	
2215170	0	0	0	0	2	
2216576	0	0	0	0	2	
2214066	0	0	0	0	1	
2214261	0	0	0	0	1	
2214492	0	0	0	0	1	
2217640	0	0	0	0	1	285

Table 4.6: Validation of 70 structures, retracted in 2010 [86]. Structures without warnings were omitted for brevity. Notations are the same as in Table 4.5

hydrogen treatment. As much as 11 bond lengths and 17 angles involving hydrogens in methyl groups are deemed unusual in 1546823 by our method as well as `PLATON`, suggesting poor hydrogen atom treatment. 1546859 is deemed unusual by both methods: `cif_check_geometry` warns about unusual parameters involving hydrogen atoms (mostly restrained), whereas `PLATON` has spotted suspicious angles involving silver atoms. Hydrogen atoms refined using riding models were involved in unusual conformations in 7228987 (5 dihedral angles) and 7229031 (3 dihedral angles). Poor overall refinement of 7229026 (R_w factor of all reflections, 0.2966, is quite high, possibly signalling poorer than ordinary refinement) is possibly the reason why 7 parameters throughout the whole moiety are judged unusual, mostly in the disordered parts. Unmarked disorder around special positions resulted in bumps in two structures. Six pairs of bumping atoms were detected in 2020874, arising from a methyl group. 18 warnings concern fragments involving hydrogen atoms. Two unusual bonds in 7056555 are due to a water molecule placed on an axis of symmetry and not marked as disordered around special position. One bond angle in Cu coordination complex is also judged slightly distorted, as well as a dihedral angle

COD ID	d_C	α_C	ϕ_C	$\sum_C$	$\sum_P$	Bumps
1519776	5	27	0	32	0	66
4112502	22	9	0	31	0	
2005559	9	19	0	28	0	
4316914	7	9	8	24	0	
7152986	7	15	0	22	2	
1517225	9	10	0	19	41	4
7103775	9	8	0	17	1	
4083625	0	17	0	17	0	
2000089	11	3	0	14	0	2
4326131	3	9	2	14	0	
4028178	4	7	2	13	0	
4313003	6	4	3	13	0	4
2006345	6	6	0	12	0	
4308222	0	12	0	12	0	
7226719	0	3	9	12	0	
2101240	3	7	0	10	0	
4508401	0	8	2	10	0	
2100783	3	6	0	9	0	
4316029	0	9	0	9	0	879
4323192	4	5	0	9	0	1
4122124	6	2	0	8	1	6
1543698	2	6	0	8	0	
2001438	4	4	0	8	0	8
4061664	6	2	0	8	0	
4110701	4	3	0	7	4	
8102293	5	2	0	7	1	
1515403	7	0	0	7	0	
2008570	5	2	0	7	0	
4022241	0	7	0	7	0	
4068650	3	4	0	7	0	

Table 4.7: Validation of 1000 random structures from the COD. 30 structures with the most warnings by our method. Notations are the same as in Table 4.5

COD ID	d_C	α_C	ϕ_C	$\sum_C$	$\sum_P$	Bumps
4111438	0	0	0	0	15	1
4061731	0	0	0	0	12	
2019970	0	0	0	0	9	639
4077645	0	0	0	0	9	
4101695	0	0	0	0	6	
4309879	0	0	0	0	6	
7009707	0	0	0	0	6	
2203936	0	0	0	0	5	
4070531	0	0	0	0	5	
4074116	0	0	0	0	5	
4076457	0	0	0	0	4	
4104545	0	0	0	0	4	
4113595	0	0	0	0	4	
4317601	0	0	0	0	4	
4333210	0	0	0	0	4	
4502024	0	0	0	0	4	
7002930	0	0	0	0	4	
7041367	0	0	0	0	4	
7102241	0	0	0	0	4	1
1515649	0	0	0	0	3	4
2204549	0	0	0	0	3	
7003771	0	0	0	0	3	
7219216	0	0	0	0	3	
2201582	0	0	0	0	2	
2207064	0	0	0	0	2	
4001338	0	0	0	0	2	
4027467	0	0	0	0	2	
4077122	0	0	0	0	2	2
4077989	0	0	0	0	2	
4079811	0	0	0	0	2	

Table 4.8: Validation of 1000 random structures from the COD. 30 structures with the most warnings by PLATON and no warnings by our method. Notations are the same as in Table 4.5

involving a pair of hydrogen atoms in riding positions. Unusual angles in ferrocene group of 1546887 were spotted by cif_check_geometry. Ferrocene rings in this structure are very close to perfect staggered conformation, what seems to be rare in the COD. PLATON judges angles in B and Ge coordination spheres as suspicious. Five bonds between carbon and oxygen atoms of cyclodextrin moiety in 4126340 are ruled unusual. The structure is a compound of a large unit cell and these unusual bonds are not accounted for in the text. However, examination of thermal ellipsoids with olex2 [207] revealed that all these bonds contain at least one highly displaced atom. Structure 4002839 reveals a limitation of our method, specifically, inability to fit a model to degenerate sample: 11 observations of Ge–P–Ag angles, 2 of which are unique, are approximated by a Cauchy mixture of a single sharp component, which is placed on the most populated peak. Six measurements from the input structure fall between the two peaks, therefore resulting in a very low likelihood.

Interestingly, there is little correlation between warnings issued by our method and PLATON. Only six structures were deemed unusual by both methods. Analysis of five structures with the most PLATON warnings (3 or more unusual angles each) revealed that fragments having unusual parameters were poorly represented in the COD and had just a couple observations. It is evident that our method is more sensitive, however, it is by definition unable to validate previously unseen geometry. Therefore, better results would be achieved by combining advantages of both our method and PLATON.

4.2.7 Validation of retracted structures

We have also carried out unusual feature detection in 70 retracted structures from *Acta Crystallographica*, as reported in 2010 [86]. Since the retraction took place long before the beginning of the current study, parameters from the retracted structures have not entered the parameter extraction stage. Out of 70 structures, 2 were considered to contain bumps. 36 structures of 70 had at least one unusual feature according to our method, while `PLATON` had warnings for 16 structures. 9 structures had warnings by both methods. Again, we have reviewed ten of the entries with the most warnings.

The most of unusual parameters were spotted in coordination compounds described by Zhong et al. (2007) [208]. `cif_check_geometry` judged unusual observations of fragments involving carbon and nitrogen atoms in coordinated organic ligands in 2214032, 2214266, 2214494, 2214731, 2214740 and 2215545. `PLATON` deemed chelation angles suspicious in most of these structures. In structures 2216548 and 2216555, published by the same group, C–C bond lengths in acetic acid moieties as well as angles in metal coordination spheres are judged suspicious. Four suspicious parameters not involving hydrogen atoms were reported in carboxyphenyl group in 2216003. Four unusual C–N bond lengths were spotted in 2215738. These observations lead to a conclusion that unusual parameters involving hydrogen atoms are relatively abundant, as compared to the non-hydrogen fragments. Therefore, suspicious geometry of non-hydrogen fragments should be accounted for by the authors or supported by diffraction data.

Again, little correlation between our method and `PLATON` is observed. Two structures with the most `PLATON` warnings, 2214262 and 2216576, are indeed genuine regarding the reported aspects: `PLATON` assumes hydroxy groups to be bonded to copper atom and judges angles of such bonds unusual. Structure of 2215170 contains a fragment with unmarked disorder whose angles are reported as unusual by `PLATON`. Our method fails to do so as it treats the resulting connectivity in the fragment as genuine, acknowledging though six previously unseen atom types.

4.2.8 Validation of random structures from the COD

1000 COD structures that were used as input for the construction of the geometry library in this study were also subjected to validation using `cif_check_geometry` and `PLATON`. Out of 1000 structures, 125 were considered to contain bumps. 159 structures were unable to be processed by `cif_check_geometry` due to the need of extensive calculation resources. 206 structures of 1000 had at least one unusual feature according to our method, while `PLATON` had warnings for 109 structures. 28 structures had warnings by both methods. Review of ten structures having the most warnings is presented below.

Again, four of the structures have warnings concerning only measurements involving hydrogen atoms, whose positions are in most cases calculated or not refined. Bond lengths involving hydrogen atoms in 4112502 are generally longer than usual, suggesting usage of constraints that are derived from non-X-ray crystal structures. Same could be said about 2005559 and 7103775. Hydrogen positions of 2000089 were located using electron density map, however, all warnings concern bonds and angles involving hydrogen atoms in this structure. Unusual observations involving heavy atoms are found in four structures bearing signs of poorer than usual refinement. In 4316914, structure with R and R_w factors greater than usual, many phenyl rings seem distorted. Phenyl rings and saturated carbon chains are distorted also in 7152986, which has large R and R_w factors as well as shift of the last refinement step that signals possible premature termination of the refinement. 1517225, structure of a very large unit cell and R_w factor slightly larger than usual, has a dozen of bond lengths and bond angles involving heavy atoms that are deemed unusual both by `cif_check_geometry` and `PLATON`. A dozen of observations involving heavy atoms (coordination spheres of rhenium atoms, imidazole and pyridine rings) are distorted in 4326131. Again, an exceptionally large shift of the last refinement step signals possible premature termination of the refinement. Two other structures were detected to have warnings concerning non-hydrogen fragments. In 1519776, all warnings originate from disordered fragment of the molecule, which might be poorly refined. A toluene moiety in 4083625 is highly distorted, albeit marked as refined with restraints. There is a possibility that these restraints were applied inappropriately.

4.2.9 Detection of typographical errors

In order to check the ability of our method to detect structures with possible typographical errors in coordinates, we have conducted tests with deliberately changed digits. We took 22 structures not judged unusual during the validation of random structures from the COD previously. Additional requirement was that all geometric parameters of the structures had to have defined models (no `*NODATA` warnings). In each structure a single digit in a coordinate was randomised, at first in the fourth position of the mantissa, then in the third.

Deliberately introduced typographical errors in the fourth positions were not detected by our method. In one structure, 9003119, Cl atom was moved out of special position by the introduction of a mistake and detected by multiplicity/multiplicity ratio tripwire. Typographical errors introduced in the third positions were detected in six structures. In five cases errors manifested in bumps. Three of them had warnings about previously unobserved atom types. One of the structures with nascent bumps had a warning about a bond being too short. In a structure without nascent bumps an angle was judged unusual. `PLATON` issued no warnings.

4.3 Curation of the COD and TCOD

In the course of the present research the COD has grown from $\sim 217\,000$ to over 390 000 records. As the present research was directly dependent on the quality of the data in the COD, much effort was allocated for its curation. Over 100 structures with redundant atoms were located and corrected by inspecting COD entries having the most bumps. Most of these structures originated from publications which aimed at detecting missed symmetry elements. `cif_voids` was used to locate and fix 30 crystal structures with incorrect (too low) symmetry descriptions, resulting in incomplete models having large voids. The browser and the validator of molecular geometry helped to locate and fix over 25 crystal structures with missing implicit hydrogens. Other errors were corrected in 200 COD entries in the course of the present research, for example, structures with dubious chemical atom types. Around 450 theoretical structures were located in the COD and marked as such. Most of them were transferred to its theoretical counterpart, the TCOD. To date the TCOD has grown to more than 2600 entries.

Chapter 5

Conclusions

The conclusions are presented below.

- Crystallography Open Database is a constantly growing and improving resource of structural small molecule information. Devised tools proved to be useful for the extraction of geometric information and building a knowledge library.
- The developed method is sufficient to organise the geometry, observed in the COD, into a knowledge library in automated and unsupervised fashion.
- The constructed library is suitable for Bayesian framework-based detection of geometric outliers in molecule structures.

Publications by the author

Papers that this dissertation is based on

1. Saulius Gražulis, Andrius Merkys, Antanas Vaitkus, and Mykolas Okulič-Kazarinas. Computing stoichiometric molecular composition from crystal structures. *Journal of Applied Crystallography*, 48:85–91, 2015. URL: `http://scripts.iucr.org/cgi-bin/paper?S1600576714025904`

2. Andrius Merkys, Antanas Vaitkus, Justas Butkus, Mykolas Okulič-Kazarinas, Visvaldas Kairys, and Saulius Gražulis. COD::CIF::Parser: an error-correcting CIF parser for the Perl language. *Journal of Applied Crystallography*, 49(1):292–301, Feb 2016. `doi:10.1107/S1600576715022396`

3. Fei Long, Robert A. Nicholls, Paul Emsley, Saulius Gražulis, Andrius Merkys, Antanas Vaitkus, and Garib N. Murshudov. Validation and extraction of stereochemical information from small molecular databases. *Acta Crystallographica Section D*, 73(2):103–111, Feb 2017. `doi:10.1107/S2059798317000079`

4. Fei Long, Robert A. Nicholls, Paul Emsley, Saulius Gražulis, Andrius Merkys, Antanas Vaitkus, and Garib N. Murshudov. ACEDRG: A stereo-chemical description generator for ligands. *Acta Crystallographica Section D*, 73(2):112–122, Feb 2017. `doi:10.1107/S2059798317000067`

5. Andrius Merkys, Nicolas Mounet, Andrea Cepellotti, Nicola Marzari, Saulius Gražulis, and Giovanni Pizzi. A posteriori metadata from automated provenance tracking: Integration of AiiDA and TCOD. *Journal of Cheminformatics*, 9(1), 2017. URL: `https://jcheminf.springeropen.com/articles/10.1186/s13321-017-0242-y`, `arXiv:1706.08704v3`, `doi:10.1186/s13321-017-0242-y`.

6. Nicolas Mounet, Marco Gibertini, Philippe Schwaller, Davide Campi, Andrius Merkys, Antimo Marrazzo, Thibault Sohier, Ivano Eligio Castelli, Andrea Cepellotti, Giovanni Pizzi, and Nicola Marzari. Two-dimensional materials from high-throughput computational exfoliation of experimentally known compounds. *Nature Nanotechnology*, Feb 2018. `doi:10.1038/s41565-017-0035-5`

Other papers

1. Saulius Gražulis, Adriana Daškevič, Andrius Merkys, Daniel Chateigner, Luca Lutterotti, Miguel Quirós, Nadezhda R. Serebryanaya, Peter Moeck, Robert T. Downs, and Armel Le Bail. Crystallography Open Database (COD): an open-access collection of crystal structures and platform for world-wide collaboration *Nucleic Acids Research*, 40:D420–D427, 2012.

International conference presentations

1. CECAM/Psi-k Research Conference: Frontiers of first-principles simulations: materials design and discovery (February 1^{st}–5^{th}, 2015, Berlin, Germany). Poster presentation: *Theoretical Crystallography Open Database – open-access repository of theoretically computed crystal structures*

2. Platform for Advanced Scientific Computing Conference (June 1^{st}–3^{rd}, 2015, Zurich, Switzerland). Poster presentation: *Developing Experimental & Theoretical Crystallography Open Databases*

3. 29th European Crystallographic Meeting (August 23^{rd}–28^{th}, 2015, Rovinj, Croatia). Poster presentation: *Integration of TCOD (Theoretical Crystallography Open Database) and AiiDA (Automated Interactive Infrastructure and Database for Atomistic simulations)*

4. OpenReadings2016 (March 16^{th}, 2016, Vilnius, Lithuania). Oral presentation: *Spotting the geometric properties in the Crystallography Open Database*

5. OpenReadings2017 (March 14^{th}, 2017, Vilnius, Lithuania). Oral presentation: *Spotting the Unusual Geometry in Crystal Structures*

6. OpenReadings2018 (March 20^{th}–23^{rd}, 2018, Vilnius, Lithuania). Poster presentation: *Statistical Insights into the Chemical Bonding in Crystal Structures*

Curriculum Vitae

Personal information

Name:	Andrius Merkys
Birth date and place:	1988-01-11, Vilnius, Lithuania
Phone:	+370 613 12191
E-mail:	andrius.merkys@gmail.com

Education

2013 – 2017	Chemical engineering, PhD studies, Vilnius University
2011 – 2013	Computer science, MSc (Magna Cum Laude), Vilnius University
2007 – 2011	Bioinformatics, BSc, Vilnius University
2000 – 2007	Vilnius Karoliniškės gymnasium
1995 – 2000	Vilnius Tuskulėnai middle school

Work

2010 – 2013	Assistant at Department of Protein – DNA Interactions, Vilnius University
2013 – now	Research assistant at Department of Protein – DNA Interactions, Vilnius University
2017 – now	Lecturer at Faculty of Mathematics and Informatics, Vilnius University

Internships

2014 – 2015	École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland (13 months)
2012	Medical Research Council Laboratory of Molecular Biology, Cambridge, UK (2 months)

List of abbreviations

ADP atomic displacement parameter.

AIC Akaike Information Criterion.

BIC Bayesian Information Criterion.

BNF Backus–Naur Form.

BOM byte order mark.

CCDC Cambridge Crystallographic Data Centre.

CGI Common Gateway Interface.

CIF Crystallographic Information Framework/Format.

CML Chemical Markup Language.

COD Crystallography Open Database.

CSD Cambridge Structural Database.

DDL Dictionary Definition Language.

DFT density functional theory.

EBNF Extended Backus–Naur Form.

EM expectation maximisation.

ICSD Inorganic Crystal Structure Database.

IUCr International Union of Crystallography.

JSON JavaScript Object Notation.

MLE maximum likelihood estimate.

NCR Numeric Character Reference.

QSAR quantitative structure activity relationship.

SMILES Simplified Molecular-Input Line-Entry System.

STAR Self-defining Text Archive and Retrieval.

TCOD Theoretical Crystallography Open Database.

TEMED tetramethylethylenediamine.

VBT Valence Bond Theory.

VSEPR Valence Shell Electron Pair Repulsion.

Abstract in Lithuanian (Santrauka)

Disertacijoje aprašyti automatiniai metodai geometrinės informacijos – tarpatominių jungčių ilgių, jungčių bei dvisienių kampų dydžių – išgavimui iš mažų molekulių kristalų struktūrų bei šios informacijos panaudojimui kitų struktūrų tikrinimui. Duomenų šaltiniu pasirinkta Atviros prieigos mažų molekulių kristalografinė duomenų bazė COD yra aktualus ir nuolat atnaujinamas struktūrinis resursas. Sukurta programinė įranga atlieka įvesties įrašų filtravimą, jų pritaikymą geometrinei analizei, geometrinių parametrų surinkimą bei šios informacijos organizavimą. Sudaryti pagal cheminį panašumą sugrupuotų geometrinių stebinių statistiniai modeliai gali būti naudojami Bajesiniu metodu pagrįstam neįprastos geometrijos aptikimui kristalų struktūrose: retai stebima molekulių geometrija automatiškai pažymima kaip reikalaujanti papildomos analizės. Šiuo principu remiantis sukurta programinė įranga, jos prieigai įrengta tinklinė naudotojo sąsaja. Neįprastõs molekulių geometrijos paieškõs metodas įvertintas su naujomis, atšauktomis bei dirbtinai sugadintomis molekulių struktūromis. Disertacijos išvados pažymi, jog COD yra tinkama naudoti kaip geometrinės informacijos šaltinis, sukurta metodika bei programinė įranga yra pakankama šaltinio informacijos organizavimui į žinių biblioteką, kuri aprašytu Bajesiniu metodu geba atpažinti neįprastą geometriją mažų molekulių kristalų struktūrose.

Bibliography

[1] Wayne A. Hendrickson. Stereochemically restrained refinement of macromolecular structures. *Methods in enzymology*, 115:252–270, 1985. doi:10.1016/0076-6879(85)15021-4.

[2] John Liebeschuetz, Jana Hennemann, Tjelvar Olsson, and Colin R. Groom. The good, the bad and the twisted: a survey of ligand geometry in protein crystal structures. *Journal of Computer-Aided Molecular Design*, 26(2):169–183, Jan 2012. URL: http://dx.doi.org/10.1007/s10822-011-9538-6, doi:10.1007/s10822-011-9538-6.

[3] Marc C. Deller and Bernhard Rupp. Models of protein–ligand crystal structures: trust, but verify. *Journal of Computer-Aided Molecular Design*, 29(9):817–836, Feb 2015. URL: http://dx.doi.org/10.1007/s10822-015-9833-8, doi:10.1007/s10822-015-9833-8.

[4] Roberto A. Steiner and Julie A. Tucker. Keep it together: restraints in crystallographic refinement of macromolecule–ligand complexes. *Acta Crystallographica Section D Structural Biology*, 73(2):93–102, Feb 2017. URL: http://dx.doi.org/10.1107/S2059798316017964, doi:10.1107/s2059798316017964.

[5] Gerard J. Kleywegt. Crystallographic refinement of ligand complexes. *Acta crystallographica. Section D, Biological crystallography*, 63:94–100, 2007. doi:10.1107/S0907444906022657.

[6] Gerard J. Kleywegt and Mark R. Harris. ValLigURL: a server for ligand-structure comparison and validation. *Acta Crystallographica Section D*, 63:935–938, 2007. URL: http://scripts.iucr.org/cgi-bin/paper?S090744490703315X, doi:10.1107/S090744490703315X.

[7] Miha Andrejašič, Jure Pražnikar, and Dušan Turk. PURY: a database of geometric restraints of hetero compounds for refinement in complexes with macromolecular structures. *Acta Crystallographica Section D, Biological Crystallography*, 64:1093–109, 2008. doi:10.1107/S0907444908027388.

[8] Alexander J. Blake, William Clegg, Jacqueline M. Cole, John S. O. Evans, Peter Main, Simon Parsons, and David J. Watkin. *Crystal Structure Analysis: Principles and Practice.* Oxford University Press, 2nd edition, 2009.

[9] Scott Kirklin, James E Saal, Bryce Meredig, Alex Thompson, Jeff W Doak, Muratahan Aykol, Stephan Rühl, and Chris Wolverton. The Open Quantum Materials Database

(OQMD): Assessing the accuracy of DFT formation energies. *npj Computational Materials*, 1(1):15010, Dec 2015. URL: http://dx.doi.org/10.1038/npjcompumats.2015.10, doi:10.1038/npjcompumats.2015.10.

[10] A. L. Spek. Single-crystal structure validation with the program PLATON. *Journal of Applied Crystallography*, 36:7–13, 2003.

[11] R.L. Harlow. Troublesome crystal structures. Prevention, detection, and resolution. *Journal of Research of the National Institute of Standards and Technology*, 101(3):327, May 1996. URL: http://dx.doi.org/10.6028/jres.101.034, doi:10.6028/jres.101.034.

[12] Jason C. Cole, Ilenia Giangreco, and Colin R. Groom. Using more than 801296 small-molecule crystal structures to aid in protein structure refinement and analysis. *Acta Crystallographica Section D*, 73(3):234–239, Mar 2017. URL: https://doi.org/10.1107/S2059798316014352, doi:10.1107/S2059798316014352.

[13] Robin Taylor, Jason Cole, Oliver Korb, and Patrick McCabe. Knowledge-based libraries for predicting the geometric preferences of druglike molecules. *Journal of Chemical Information and Modeling*, 54(9):2500–2514, Sep 2014. URL: http://dx.doi.org/10.1021/ci500358p, doi:10.1021/ci500358p.

[14] Pierre Baldi. Data-driven high-throughput prediction of the 3-D structure of small molecules: review and progress. A response to the letter by the Cambridge Crystallographic Data Centre. *Journal of chemical information and modeling*, 51:3029, 2011. URL: http://pubs.acs.org/doi/abs/10.1021/ci200460z, doi:10.1021/ci200460z.

[15] Saulius Gražulis, Adriana Daškevič, Andrius Merkys, Daniel Chateigner, Luca Lutterotti, Miguel Quirós, Nadezhda R. Serebryanaya, Peter Moeck, Robert T. Downs, and Armel Le Bail. Crystallography Open Database (COD): an open-access collection of crystal structures and platform for world-wide collaboration. *Nucleic Acids Research*, 40(D1):D420–D427, Jan 2012. URL: http://nar.oxfordjournals.org/content/40/D1/D420.abstract, doi:10.1093/nar/gkr900.

[16] Denis Cousineau and Sylvain Chartier. Outliers detection and treatment: a review. *International Journal of Psychological Research*, 3(1):58–67, 2010. URL: http://revistas.usb.edu.co/index.php/IJPR/article/view/844.

[17] Walter Steurer. Twenty years of structure research on quasicrystals. part I. pentagonal, octagonal, decagonal and dodecagonal quasicrystals. *Zeitschrift für Kristallographie - Crystalline Materials*, 219(7), jan 2004. doi:10.1524/zkri.219.7.391.35643.

[18] Santiago Alvarez. A cartography of the van der Waals territories. *Dalton Transactions*, 42:8617–8636, 2013. URL: http://pubs.rsc.org/en/Content/ArticleLanding/2013/DT/c3dt50599e#!divAbstract, doi:10.1039/c3dt50599e.

[19] Armel Le Bail. Inorganic structure prediction with *GRINSP*. *Journal of Applied Crystallography*, 38:389–395, 2005. URL: http://dx.doi.org/10.1107/S0021889805002384, doi:10.1107/S0021889805002384.

[20] Linus Pauling. *The nature of the chemical bond, 3rd ed.*, volume 1. Cornell University Press, Feb 1960.

[21] Miguel Quirós Olozábal, Saulius Gražulis, Saulė Girdzijauskaitė, Andrius Merkys, and Antanas Vaitkus. Using SMILES strings for the description of chemical connectivity in the Crystallography Open Database. Accepted to Journal of Cheminformatics, 2018.

[22] Beatriz Cordero, Verónica Gómez, Ana E. Platero-Prats, Marc Revés, Jorge Echeverría, Eduard Cremades, Flavia Barragán, and Santiago Alvarez. Covalent radii revisited. *Dalton Transactions*, pages 2832–2838, 2008. URL: `http://pubs.rsc.org/en/Content/ArticleLanding/2008/DT/b801115j#!divAbstract`, `doi:10.1039/b801115j`.

[23] Pekka Pyykkö and Michiko Atsumi. Molecular double-bond covalent radii for elements Li–E112. *Chemistry – A European Journal*, 15:12770–12779, 2009. URL: `http://dx.doi.org/10.1002/chem.200901472`, `doi:10.1002/chem.200901472`.

[24] Nick Spadaccini and Sydney R. Hall. DDLm: A new dictionary definition language. *Journal of Chemical Information and Modeling*, 52:1907–1916, 2012. URL: `http://pubs.acs.org/doi/abs/10.1021/ci300075z`,
`arXiv:http://pubs.acs.org/doi/pdf/10.1021/ci300075z`, `doi:10.1021/ci300075z`.

[25] Fei Long, Robert A. Nicholls, Paul Emsley, Saulius Gražulis, Andrius Merkys, Antanas Vaitkus, and Garib N. Murshudov. ACEDRG: A stereo-chemical description generator for ligands. *Acta Crystallographica Section D*, 73(2):112–122, Feb 2017. `doi:10.1107/S2059798317000067`.

[26] H. M. Rietveld. A profile refinement method for nuclear and magnetic structures. *J. Appl. Cryst.*, 2:65–71, 1969. URL: `http://www.ccp14.ac.uk/ccp/web-mirrors/hugorietveld/xtal/paper2/paper2.html`.

[27] CCP14. Classic Hugo Rietveld source code online: Original Hugo M. Rietveld report - RCN-104 - Reactor Centrum Nederland [online]. 1969. URL: `http://www.ccp14.ac.uk/ccp/web-mirrors/hugorietveld/riet-report/index.html`.

[28] David J. Watkin. *Refinement of crystal structures*, pages 169–188. International Union of Crystallography (IUCr), 2009.

[29] John Evans. *Analysis of extended inorganic structures*, page 189. International Union of Crystallography (IUCr), 2009.

[30] Gerard J. Kleywegt and T. Alwyn Jones. Homo crystallographicus–quo vadis? *Structure (London, England : 1993)*, 10(4):465–472, 2002. `doi:10.1016/S0969-2126(02)00743-8`, `PMID:11937051`.

[31] Garib N. Murshudov, Alexei A. Vagin, and Eleanor J. Dodson. Refinement of macromolecular structures by the maximum-likelihood method. *Acta Cryst. D*, 53:240–255, 1997. `doi:10.1107/S0907444996012255`.

[32] A T Brunger, P D Adams, G M Clore, W L DeLano, P Gros, R W Grosse-Kunstleve, J S Jiang, J Kuszewski, M Nilges, N S Pannu, R J Read, L M Rice, T Simonson, and G L Warren. Crystallography & nmr system: A new software suite for macromolecular structure determination. *Acta Crystallogr D Biol Crystallogr*, 54(Pt 5):905–21, Sep 1998. URL: `http://www.ncbi.nih.gov/entrez/query.fcgi?cmd=retrieve&db=pubmed&dopt=abstract&list_uids=9757107`, `doi:10.1107/S0907444998003254`.

[33] Dale E. Tronrud. Introduction to macromolecular refinement. *Acta Crystallographica Section D*, 60:2156–2168, 2004. URL: `http://dx.doi.org/10.1107/S090744490402356X`, `doi:10.1107/S090744490402356X`.

[34] Alexei A. Vagin, Roberto A. Steiner, Andrey A. Lebedev, Liz Potterton, Stuart McNicholas, Fei Long, and Garib N. Murshudov. *REFMAC5* dictionary: organization of prior chemical knowledge and guidelines for its use. *Acta Crystallographica Section D*, 60(12):2184–2195, Dec 2004. URL: `http://view.ncbi.nlm.nih.gov/pubmed/15572771`, `doi:10.1107/S0907444904023510`.

[35] Gerard J. Kleywegt, Kim Henrick, Eleanor J. Dodson, and Daan M.F. van Aalten. Pound-wise but penny-foolish: How well do micromolecules fare in macromolecular refinement? *Structure*, 11:1051–1059, 2003. `doi:10.1016/S0969-2126(03)00186-2`.

[36] David Watkin. Structure refinement: some background theory and practical strategies. *Journal of Applied Crystallography*, 41(3):491–522, Apr 2008. URL: `http://dx.doi.org/10.1107/S0021889808007279`, `doi:10.1107/s0021889808007279`.

[37] Philip R. Evans. An introduction to stereochemical restraints. *Acta Crystallographica Section D*, 63:58–61, 2007. URL: `http://dx.doi.org/10.1107/S090744490604604X`, `doi:10.1107/S090744490604604X`.

[38] Mariusz Jaskolski, Miroslaw Gilski, Zbigniew Dauter, and Alexander Wlodawer. Stereochemical restraints revisited: how accurate are refinement targets and how much should protein structures be allowed to deviate from them? *Acta crystallographica. Section D, Biological crystallography*, 63:611–620, 2007.

[39] Kristina Nilsson, David Lecerof, Emma Sigfridssona, and Ulf Rydea. An automatic method to generate force-field parameters for hetero-compounds. *Acta Crystallographica Section D*, 59:274–289, 2002. `doi:10.1107/S0907444902021431`.

[40] Robin Taylor. Life-science applications of the Cambridge Structural Database. *Acta crystallographica. Section D, Biological crystallography*, 58:879–88, 2002.

[41] Gerhard Klebe. *Structure Correlation and Ligand/Receptor Interactions*, volume 2, chapter 13, page 543–603. Wiley-Blackwell, 1994. URL: `http://dx.doi.org/10.1002/9783527616091.ch13`, `doi:10.1002/9783527616091.ch13`.

[42] Colin R. Groom and Jason C. Cole. The use of small-molecule structures to complement protein–ligand crystal structures in drug discovery. *Acta Crystallographica Section*

D, 73(3):240–245, Mar 2017. URL: `https://doi.org/10.1107/S2059798317000675`, `doi:10.1107/S2059798317000675`.

[43] Frank H. Allen and Robin Taylor. Librarians, crystal structures and drug design. *Chemical communications (Cambridge, England)*, (41):5135–40, Sep 2005. URL: `http://pubs.rsc.org/en/content/articlelanding/2005/cc/b511106b`, `doi:10.1039/B511106B`.

[44] Antony J. Williams and Sean Ekins. A quality alert and call for improved curation of public chemistry databases. *Drug Discovery Today*, 16(17-18):747–750, Sep 2011. URL: `http://dx.doi.org/10.1016/j.drudis.2011.07.007`, `doi:10.1016/j.drudis.2011.07.007`.

[45] Ian J. Bruno, Gregory P. Shields, and Robin Taylor. Deducing chemical structure from crystallographically determined atomic coordinates. *Acta Crystallographica Section B Structural Science*, 67(4):333–349, Jul 2011. URL: `http://dx.doi.org/10.1107/S0108768111024608`, `doi:10.1107/s0108768111024608`.

[46] Matthew Clark, Richard D. Cramer, and Nicole Van Opdenbosch. Validation of the general purpose Tripos 5.2 force field. *Journal of Computational Chemistry*, 10(8):982–1012, Dec 1989. URL: `http://dx.doi.org/10.1002/jcc.540100804`, `doi:10.1002/jcc.540100804`.

[47] Aurora J. Cruz-Cabeza, John W. Liebeschuetz, and Frank H. Allen. Systematic conformational bias in small-molecule crystal structures is rare and explicable. *CrystEngComm*, 14(20):6797, 2012. URL: `http://dx.doi.org/10.1039/c2ce25585e`, `doi:10.1039/c2ce25585e`.

[48] Alec Belsky, Mariette Hellenbrandt, Vicky Lynn Karen, and Peter Luksch. New developments in the Inorganic Crystal Structure Database (ICSD): accessibility in support of materials research and design. *Acta Crystallographica Section B*, 58(3 Part 1):364–369, Jun 2002. URL: `http://dx.doi.org/10.1107/S0108768102006948`, `doi:10.1107/S0108768102006948`.

[49] Peter S White, John R Rodgers, and Yvon Le Page. CRYSTMET: a database of the structures and powder patterns of metals and intermetallics. *Acta Crystallogr B*, 58(Pt 3 Pt 1):343–8, Jun 2002.

[50] Nick Day, Jim Downing, Sam Adams, N. W. England, and Peter Murray-Rust. CrystalEye: automated aggregation, semantification and dissemination of the world's open crystallographic data. *Journal of Applied Crystallography*, 45:316–323, 2012. `doi:10.1107/S0021889812006462`.

[51] Saulius Gražulis, Daniel Chateigner, Robert T. Downs, A. F. T. Yokochi, Miguel Quirós, Luca Lutterotti, Elena Manakova, Justas Butkus, Peter Moeck, and Armel Le Bail. Crystallography Open Database – an open-access collection of crystal structures. *Journal of Applied Crystallography*, 42(4):726–729, Aug 2009. URL: `http://dx.doi.org/10.1107/S0021889809016690`, `doi:10.1107/S0021889809016690`.

[52] Joachim Breternitz and Duncan Gregory. The search for hydrogen stores on a large scale; a straightforward and automated open database analysis as a first sweep for candidate materials. *Crystals*, 5:617–633, 2015. doi:10.3390/cryst5040617.

[53] Nicolas Mounet, Marco Gibertini, Philippe Schwaller, Davide Campi, Andrius Merkys, Antimo Marrazzo, Thibault Sohier, Ivano Eligio Castelli, Andrea Cepellotti, Giovanni Pizzi, and Nicola Marzari. Two-dimensional materials from high-throughput computational exfoliation of experimentally known compounds. *Nature Nanotechnology*, Feb 2018. URL: http://dx.doi.org/10.1038/s41565-017-0035-5, doi:10.1038/s41565-017-0035-5.

[54] United Nations Educational, Scientific and Cultural Organisation (UNESCO). Open access to scientific information [online]. URL: http://www.unesco.org/new/en/communication-and-information/access-to-knowledge/open-access-to-scientific-information/.

[55] Stanislav S. Borysov, R. Matthias Geilhufe, and Alexander V. Balatsky. Organic materials database: An open-access online database for data mining. *PLOS ONE*, 12(2):e0171501, Feb 2017. URL: http://dx.doi.org/10.1371/journal.pone.0171501, doi:10.1371/journal.pone.0171501.

[56] Daan M. F. van Aalten, Robert P. Bywater, John B. C. Findlay, Manfred Hendlich, Rob W. W. Hooft, and Gert Vriend. PRODRG, a program for generating molecular topologies and unique molecular descriptors from coordinates of small molecules. *Journal of Computer-Aided Molecular Design*, 10:255–262, 1996. doi:10.1007/BF00355047.

[57] F. H. Allen, O. Kennard, W. D. S. Motherwell, W. G. Town, D. G. Watson, T. J. Scott, and A. C. Larson. The Cambridge Crystallographic Data Centre, part 3. the unique molecule program. *Journal of Applied Crystallography*, 7(1):73–78, Feb 1974. URL: http://dx.doi.org/10.1107/s0021889874008739, doi:10.1107/s0021889874008739.

[58] Helen M. Berman. The Protein Data Bank: a historical perspective. *Acta Crystallographica Section A*, 64(1):88–95, Jan 2008. URL: https://doi.org/10.1107/S0108767307035623, doi:10.1107/S0108767307035623.

[59] wwPDB. Atomic Coordinate Entry Format Description. Version 3.3. URL: http://www.wwpdb.org/documentation/file-format-content/format33/v3.3.html.

[60] S. R. Hall, F. H. Allen, and I. D. Brown. The crystallographic information file (CIF): a new standard archive file for crystallography. *Acta Crystallographica Section A*, 47(6):655–685, Nov 1991. URL: http://dx.doi.org/10.1107/S010876739101067X, doi:10.1107/S010876739101067X.

[61] Peter Murray-Rust and Henry S. Rzepa. Chemical markup, XML, and the Worldwide Web. 1. basic principles. *J. Chem. Inf. Comput. Sci.*, 39:928–942, 1999. doi:10.1021/ci990052b.

[62] Colin R. Groom and Frank H. Allen. The Cambridge Structural Database in retrospect and prospect. *Angewandte Chemie International Edition*, 53(3):662–671, 2014. URL: `http://dx.doi.org/10.1002/anie.201306438`, `doi:10.1002/anie.201306438`.

[63] Yvon Le Page and John R. Rodgers. Quantum software interfaced with crystal-structure databases: tools, results and perspectives. *Journal of Applied Crystallography*, 38:697–705, 2005. URL: `http://dx.doi.org/10.1107/S0021889805017358`, `doi:10.1107/S0021889805017358`.

[64] P. M. D. Fitzgerald, J. D. Westbrook, P. E. Bourne, B. McMahon, K. D. Watenpaugh, and H. M. Berman. *Macromolecular dictionary (mmCIF)*, volume G, chapter 4.5, pages 295–443. International Union of Crystallography, 2006. `doi:10.1107/97809553602060000745`.

[65] Brian H. Toby, Robert B. Von Dreele, and Allen C. Larson. CIF applications. XIV. reporting of rietveld results using pdCIF: GSAS2CIF. *Journal of Applied Crystallography*, 36:1290–1294, 2003. URL: `http://www.ncnr.nist.gov/xtal/software/cif/gsas2cif.pdf`, `doi:10.1107/S0021889803016819`.

[66] P. R. Mallinson and I. D. Brown. *Classification and use of electron density data*, volume G, chapter 3.5, pages 141–143. International Union of Crystallography, 2006.

[67] Andrius Merkys, Antanas Vaitkus, Justas Butkus, Mykolas Okulič-Kazarinas, Visvaldas Kairys, and Saulius Gražulis. *COD::CIF::Parser*: an error-correcting CIF parser for the Perl language. *Journal of Applied Crystallography*, 49(1):292–301, Feb 2016. URL: `http://dx.doi.org/10.1107/S1600576715022396`, `doi:10.1107/S1600576715022396`.

[68] Brian McMahon. *Syntactic utilities for CIF*, volume G, chapter 5.3, pages 499–525. International Union of Crystallography, 2006.

[69] Giovanni Pizzi, Andrea Cepellotti, Riccardo Sabatini, Nicola Marzari, and Boris Kozinsky. AiiDA: automated interactive infrastructure and database for computational science. *Computational Materials Science*, 111:218–230, 2016. `doi:10.1016/j.commatsci.2015.09.013`.

[70] S. R. Bahn and K. W. Jacobsen. An object-oriented scripting interface to a legacy electronic structure code. *Computing in Science Engineering*, 4(3):56–66, 2002. `doi:10.1109/5992.998641`.

[71] Shyue Ping Ong, William Davidson Richards, Anubhav Jain, Geoffroy Hautier, Michael Kocher, Shreyas Cholia, Dan Gunter, Vincent L. Chevrier, Kristin A. Persson, and Gerbrand Ceder. Python Materials Genomics (pymatgen): A robust, open-source python library for materials analysis. *Computational Materials Science*, 68:314–319, 2013. URL: `http://www.sciencedirect.com/science/article/pii/S0927025612006295`, `doi:http://dx.doi.org/10.1016/j.commatsci.2012.10.028`.

[72] Herbert J. Bernstein, John C. Bollinger, I. David Brown, Saulius Gražulis, James R. Hester, Brian McMahon, Nick Spadaccini, John D. Westbrook, and Simon P. Westrip. Specification of the Crystallographic Information File format, version 2.0. *Journal of*

Applied Crystallography, 49(1):277–284, Feb 2016. URL: http://dx.doi.org/10.1107/S1600576715021871, doi:10.1107/s1600576715021871.

[73] Weerapong Phadungsukanan, Markus Kraft, Joe A. Townsend, and Peter Murray-Rust. The semantics of Chemical Markup Language (CML) for computational chemistry : CompChem. *Journal of Cheminformatics*, 4(1):15, Aug 2012. URL: https://doi.org/10.1186/1758-2946-4-15, doi:10.1186/1758-2946-4-15.

[74] Gerard J. Kleywegt. Validation of protein crystal structures. *Acta Crystallographica Section D*, 56:249–265, 2000.

[75] Charles Babbage. *Passages from the Life of a Philosopher.* Longman and Co., 1864.

[76] Eric N. Brown and S. Ramaswamy. Quality of protein crystal structures. *Acta crystallographica. Section D, Biological crystallography*, 63:941–50, 2007. doi:10.1107/S0907444907033847.

[77] Alan R. Katritzky, C. Dennis Hall, Bahaa El-Dien M. El-Gendy, and Bogdan Draghici. Tautomerism in drug discovery. *Journal of Computer-Aided Molecular Design*, 24(6-7):475–484, May 2010. URL: http://dx.doi.org/10.1007/s10822-010-9359-z, doi:10.1007/s10822-010-9359-z.

[78] Aurora J. Cruz-Cabeza and Colin R. Groom. Identification, classification and relative stability of tautomers in the Cambridge Structural Database. *CrystEngComm*, 13(1):93–98, 2011. URL: http://dx.doi.org/10.1039/C0CE00123F, doi:10.1039/c0ce00123f.

[79] W. H. Baur and D. Kassner. The perils of Cc: comparing the frequencies of falsely assigned space groups with their general population. *Acta Crystallographica Section B Structural Science*, 48(4):356–369, Aug 1992. URL: http://dx.doi.org/10.1107/s0108768191014726, doi:10.1107/s0108768191014726.

[80] W. H. Baur and E. Tillmanns. How to avoid unnecessarily low symmetry in crystal structure determinations. *Acta Crystallographica Section B*, 42(1):95–111, Feb 1986. URL: https://doi.org/10.1107/S0108768186098518, doi:10.1107/S0108768186098518.

[81] John R. Helliwell, Brian McMahon, J. Mitchell Guss, and Loes M. J. Kroon-Batenburg. The science is in the data. *IUCrJ*, 4(6), oct 2017. doi:10.1107/s2052252517013690.

[82] F. C. Fang, R. G. Steen, and A. Casadevall. Misconduct accounts for the majority of retracted scientific publications. *Proceedings of the National Academy of Sciences*, 109(42):17028–17033, Oct 2012. URL: http://dx.doi.org/10.1073/pnas.1212247109, doi:10.1073/pnas.1212247109.

[83] Alison McCook. Help us: Here's some of what we're working on [online]. 2016. URL: http://retractionwatch.com/help-us-heres-some-of-what-were-working-on/.

[84] Bert J. C. Janssen, Randy J. Read, Axel T. Brünger, and Piet Gros. Crystallographic evidence for deviating c3b structure. *Nature*, 448(7154):E1–E2, aug 2007. doi:10.1038/nature06102.

[85] Brendan Borrell. Fraud rocks protein community. *Nature*, 462(7276):970–970, dec 2009. doi:10.1038/462970a.

[86] William T. A. Harrison, Jim Simpson, and Matthias Weil. Editorial. *Acta Crystallographica Section E*, 66:e1–e2, 2010. doi:10.1107/S1600536809051757.

[87] Thomas C. Terwilliger and Gerard Bricogne. Continuous mutual improvement of macromolecular structure models in the PDB and of X-ray crystallographic software: the dual role of deposited experimental data. *Acta Crystallographica Section D*, 70(10):2533–2543, Oct 2014. URL: https://doi.org/10.1107/S1399004714017040, doi:10.1107/S1399004714017040.

[88] David Weininger. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Modeling*, 28(1):31–36, Feb 1988. URL: http://dx.doi.org/10.1021/ci00057a005, doi:10.1021/ci00057a005.

[89] Daniel M. Lowe, Peter T. Corbett, Peter Murray-Rust, and Robert C. Glen. Chemical name to structure: OPSIN, an open source solution. *Journal of chemical information and modeling*, 51:739, 2011. doi:10.1021/ci100384d.

[90] Jacco van de Streek and Marcus A. Neumann. Validation of experimental molecular crystal structures with dispersion-corrected density functional theory calculations. *Acta Crystallographica Section B Structural Science*, 66(5):544–558, Sep 2010. URL: http://dx.doi.org/10.1107/S0108768110031873, doi:10.1107/s0108768110031873.

[91] G. N. Ramachandran, C Ramakrishnan, and V. Sasisekharan. Stereochemistry of polypeptide chain configurations. *Journal of Molecular Biology*, pages 95–99, 1963. doi:10.1016/S0022-2836(63)80023-6.

[92] Francisco Carrascoza, Snezana Zaric, and Radu Silaghi-Dumitrescu. Computational study of protein secondary structure elements: Ramachandran plots revisited. *Journal of molecular graphics & modelling*, 50:125–133, 2014. doi:10.1016/j.jmgm.2014.04.001, PMID:24793053.

[93] Patrick McCabe, Oliver Korb, and Jason Cole. Kernel density estimation applied to bond length, bond angle, and torsion angle distributions. *Journal of Chemical Information and Modeling*, 54(5):1284–1288, May 2014. URL: http://dx.doi.org/10.1021/ci500156d, doi:10.1021/ci500156d.

[94] Ton Spek. PLATON/SQUEEZE in the context of twinning and SHELXL2013, 2013. URL: https://www.platonsoft.nl/spek/ppp/Mulheim.pdf.

[95] Ton Spek. *VOID & SOLV Calculations*, Jan 2005. URL: http://www.cryst.chem.uu.nl/spek/platon/pl000302.html.

[96] Clare F. Macrae, Ian J. Bruno, James A. Chisholm, Paul R. Edgington, Patrick McCabe, Elna Pidcock, Lucia Rodriguez-Monge, Robin Taylor, Jacco van de Streek, and Peter A. Wood. *Mercury CSD 2.0* – new features for the visualization and investigation of crystal structures. *Journal of Applied Crystallography*, 41(2):466–470, Apr 2008. URL: https://doi.org/10.1107/S0021889807067908, doi:10.1107/S0021889807067908.

[97] Bernard R. Brooks, Robert E. Bruccoleri, Barry D. Olafson, David J. States, S. Swaminathan, and Martin Karplus. CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *Journal of Computational Chemistry*, 4(2):187–217, 1983. URL: http://dx.doi.org/10.1002/jcc.540040211, doi:10.1002/jcc.540040211.

[98] Richard A. Engh and Robert Huber. Accurate bond and angle parameters for X-ray protein structure refinement. *Acta Crystallographica Section A*, 47:392–400, 1991.

[99] Sheila Ash, Malcolm A. Cline, R. Webster Homer, Tad Hurst, and Gregory B. Smith. SYBYL line notation (SLN): A versatile language for chemical structure representation. *Journal of Chemical Information and Computer Sciences*, 37(1):71–79, Jan 1997. URL: http://dx.doi.org/10.1021/ci960109j, doi:10.1021/ci960109j.

[100] P. Andrew Karplus, Maxim V. Shapovalov, Roland L. Dunbrack, Jr, and Donald S. Berkholz. A forward-looking suggestion for resolving the stereochemical restraints debate: ideal geometry functions. *Acta Crystallographica Section D*, 64(3):335–336, Mar 2008. URL: https://doi.org/10.1107/S0907444908002333, doi:10.1107/S0907444908002333.

[101] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977. URL: http://links.jstor.org/sici?sici=0035-9246%281977%2939%3A1%3C1%3AMLFIDV%3E2.0.CO%3B2-Z.

[102] Hirotugu Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19:716–723, 1974. doi:10.1109/TAC.1974.1100705.

[103] Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, pages 461–464, 1978. The PDF is also available at: http://www.andrew.cmu.edu/user/kk3n/simplicity/schwarzbic.pdf. URL: http://projecteuclid.org/euclid.aos/1176344136, doi:10.1214/aos/1176344136.

[104] Stephan R. Sain, H. L. Gray, Wayne A. Woodward, and Mark D. Fisk. Outlier detection from a mixture distribution when training data are unlabeled. *Bulletin of the Seismological Society of America*, 89(1):294–304, February 1999. URL: https://pdfs.semanticscholar.org/eef1/bb217a8235643318e38122605a8ca5d1d07a.pdf.

[105] Robert E. Kass and Adrian E. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795, Jun 1995. URL: http://dx.doi.org/10.1080/01621459.1995.10476572, doi:10.1080/01621459.1995.10476572.

[106] Christian Hennig. Methods for merging Gaussian mixture components. *Advances in Data Analysis and Classification*, 4(1):3–34, jan 2010. `doi:10.1007/s11634-010-0058-3`.

[107] A. L. Spek. *CIF validation with the program PLATON (version 01-10-2010)*, 2010. URL: `http://www.cryst.chem.uu.nl/platon/CIF-VALIDATION.pdf`.

[108] A. L. Spek. What is PLATON and how to get started [online]. URL: `http://www.cryst.chem.uu.nl/spek/platon/pl002000.html`.

[109] Wouter G. Touw and Gert Vriend. On the complexity of Engh and Huber refinement restraints: The angle τ as example. *Acta Crystallographica Section D*, 66:1341–1350, 2010. `doi:10.1107/S0907444910040928`.

[110] Gerhard Klebe and Thomas Mietzner. A fast and efficient method to generate biologically relevant conformations. *Journal of Computer-Aided Molecular Design*, 8(5):583–606, Oct 1994. URL: `http://dx.doi.org/10.1007/bf00123667`, `doi:10.1007/bf00123667`.

[111] Peter Murray-Rust. *Molecular Structure and Biological Activity*, pages 117–133. Elsevier, 1982.

[112] Cambridge Crystallographic Data Centre. *Mogul User Guide and Tutorials. 2017 CSD Release*, 2016. URL: `https://www.ccdc.cam.ac.uk/support-and-resources/ccdcresources/f627c0ea4173486893f8782a62132858.pdf`.

[113] Jason C. Cole, Colin R. Groom, Oliver Korb, Patrick McCabe, and Gregory P. Shields. Knowledge-based optimization of molecular geometries using crystal structures. *Journal of Chemical Information and Modeling*, 56(4):652–661, Apr 2016. URL: `http://dx.doi.org/10.1021/acs.jcim.5b00712`, `doi:10.1021/acs.jcim.5b00712`.

[114] Helen M. Berman, Margaret J. Gabanyi, Colin R. Groom, John E. Johnson, Garib N. Murshudov, Robert A. Nicholls, Vijay Reddy, Torsten Schwede, Matthew D. Zimmerman, John Westbrook, and Wladek Minor. Data to knowledge: how to get meaning from your result. *IUCrJ*, 2(1):45–58, Jan 2015. URL: `https://doi.org/10.1107/S2052252514023306`, `doi:10.1107/S2052252514023306`.

[115] Fei Long, Robert A. Nicholls, Paul Emsley, Saulius Gražulis, Andrius Merkys, Antanas Vaitkus, and Garib N. Murshudov. Validation and extraction of stereochemical information from small molecular databases. *Acta Crystallographica Section D*, 73(2):103–111, Feb 2017. `doi:10.1107/S2059798317000079`.

[116] Jon Postel. Transmission control protocol. Technical report, Information Sciences Institute University of Southern California, Jan 1980. URL: `https://tools.ietf.org/html/rfc761`.

[117] Brian McMahon. *vcif*, volume G, chapter 5.3.2.1, pages 499–501. International Union of Crystallography, 2006.

[118] Georgi Todorov and Herbert J. Bernstein. *VCIF2*: extended CIF validation software. *Journal of Applied Crystallography*, 41(4):808–810, Aug 2008. URL: `http://dx.doi.org/10.1107/S002188980801385X`, `doi:10.1107/S002188980801385X`.

[119] John C. Bollinger. A portable general-purpose application programming interface for CIF 2.0. *J Appl Crystallogr*, 49(1):285–291, Feb 2016. URL: `http://dx.doi.org/10.1107/S1600576715021883`, `doi:10.1107/s1600576715021883`.

[120] Marcin Wojdyr. *Gemmi - General MacroMolecular I/O*. Global Phasing Ltd., GIT commit 860d28508767752288ae74c3737ab602f444a896 edition, 2017.

[121] Richard J. Gildea, Luc J. Bourhis, Oleg V. Dolomanov, Ralf W. Grosse-Kunstleve, Horst Puschmann, Paul D. Adams, and Judith A. K. Howard. iotbx.cif: a comprehensive CIF toolbox. *Journal of Applied Crystallography*, 44(6):1259–1263, December 2011. `doi:10.1107/S0021889811041161`.

[122] S. R. Hall and H. J. Bernstein. CIF Applications. V. *ciftbx2*: extended tool box for manipulating CIFs. *Journal of Applied Crystallography*, 29(5):598–603, Oct 1996. URL: `http://dx.doi.org/10.1107/S0021889896006371`, `doi:10.1107/S0021889896006371`.

[123] Ask Hjorth Larsen, Jens Jørgen Mortensen, Jakob Blomqvist, Ivano E. Castelli, Rune Christensen, Marcin Dułak, Jesper Friis, Michael N. Groves, Bjørk Hammer, Cory Hargus, Eric D. Hermes, Paul C. Jennings, Peter Bjerre Jensen, James Kermode, John R. Kitchin, Esben Leonhard Kolsbjerg, Joseph Kubal, Kristen Kaasbjerg, Steen Lysgaard, Jón Bergmann Maronsson, Tristan Maxson, Thomas Olsen, Lars Pastewka, Andrew Peterson, Carsten Rostgaard, Jakob Schiøtz, Ole Schütt, Mikkel Strange, Kristian S. Thygesen, Tejs Vegge, Lasse Vilhelmsen, Michael Walter, Zhenhua Zeng, and Karsten W. Jacobsen. The atomic simulation environment – a Python library for working with atoms. *Journal of Physics: Condensed Matter*, 29(27):273002, jun 2017. `doi:10.1088/1361-648x/aa680e`.

[124] J. R. Hester. A validating CIF parser: *PyCIFRW*. *Journal of Applied Crystallography*, 39(4):621–625, Aug 2006. URL: `http://dx.doi.org/10.1107/S0021889806015627`, `doi:10.1107/S0021889806015627`.

[125] Guido van Rossum. *An Introduction to Python*. Network Theory, 2003.

[126] David R. Stampf. ZINC – galvanizing CIF to work with UNIX, 2004. URL: `http://www.iucr.org/__data/iucr/cif/software/zinc/doc/zinc-paper.pdf`.

[127] Sydney R. Hall and Nick Spadaccini. The STAR file: Detailed specifications. *Journal of Chemical Information and Computer Sciences*, 34(3):505–508, 1994. URL: `http://dx.doi.org/10.1021/ci00019a005`, `doi:10.1021/ci00019a005`.

[128] W. Bluhm. STAR (CIF) parser, 2000. URL: `http://pdb.sdsc.edu/STAR/index.html`.

[129] Larry Wall, Tom Christiansen, and Jon Orwant. *Programming Perl*. O'Reilly Media, third edition, July 2000.

[130] Peter A. Keller. A lexical analyser for STAR/CIF/mmCIF data, Sep 2013. URL: http://www.globalphasing.com/startools/StarTools_article.pdf.

[131] Francois Desarmenien. Parse::Yapp – Perl extension for generating and using LALR parsers, 1998. URL: http://search.cpan.org/perldoc?Parse::Yapp.

[132] C. Donnely and R. Stallman. *GNU Bison - The Yacc-compatible Parser Generator*. Free Software Foundation, 2015. URL: http://www.gnu.org/software/bison/manual/.

[133] S. C. Johnson. YACC: Yet Another Compiler-Compiler. Technical report, AT&T Bell Laboratories, Murray Hill, New Jersey, 1975. Computing science technical report 32.

[134] John Levine. *flex & bison*. O'Reilly, 2009.

[135] S. R. Hall, N. Spadaccini, I. D. Brown, H. J. Bernstein, J. D. Westbrook, and B. McMahon. *Formal specification of the Crystallographic Information File. Version 1.1 specification*, volume G, chapter 2.2.7, pages 25–36. International Union of Crystallography, 2006. URL: http://xrpp.iucr.org/Ga/ch2o2v0001/sec2o2o7/.

[136] COMCIFS. CIF 1.1 specification. Appendix A, February 2003. URL: http://www.iucr.org/resources/cif/spec/version1.1/cifsyntax#bnf.

[137] Dave Beazley, Luigi Ballabio, William Fulton, Mark Gossage, Matthias Köppe, John Lenz, Marcelo Matus, Jason Stewart, Art Yerkes, Shibukawa Yoshiki, Surendra Singhi, Xavier Delacour, Olly Betts, and Ding Zhi Gang. Simplified Wrapper and Interface Generator, 2015. URL: http://swig.org.

[138] Jarkko Hietaniemi. Perl ports (binary distributions), 2010. URL: http://www.cpan.org/ports/.

[139] Steven Pemberton, Daniel Austin, Jonny Axelsson, Tantek Çelik, Doug Dominiak, Herman Elenbaas, Beth Epperson, Masayasu Ishikawa, Shin'ichi Matsui, Shane McCarron, Ann Navarro, Subramanian Peruvemba, Rob Relyea, Sebastian Schnitzenbaumer, and Peter Stark. XHTML^{TM} 1.0 the extensible hypertext markup language (second edition): A reformulation of HTML 4 in XML 1.0, August 2000. W3C Recommendation 26 January 2000, revised 1 August 2002. URL: http://www.w3.org/TR/xhtml1/.

[140] ISO. Information technology – Syntactic metalanguage – Extended BNF. International Standard ISO/IEC 14977:1996(E), International Organization for Standardization, Geneva, Switzerland, 1996.

[141] Per Cederberg. Grammatica :: Parser generator [online]. 2015. URL: http://grammatica.percederberg.net/.

[142] George M. Sheldrick. Crystal structure refinement with SHELXL. *Acta Crystallographica Section C*, 71(1):3–8, Jan 2015. doi:10.1107/S2053229614024218.

[143] John Bollinger. CIF - changes to the specification, Jul 2011. URL: http://www.iucr.org/__data/assets/pdf_file/0020/59420/cif2_syntax_changes-jcb20110728.pdf.

[144] Andrius Merkys, Nicolas Mounet, Andrea Cepellotti, Nicola Marzari, Saulius Gražulis, and Giovanni Pizzi. A posteriori metadata from automated provenance tracking: Integration of AiiDA and TCOD. *Journal of Cheminformatics*, 9(1), 2017. URL: `https://jcheminf.springeropen.com/articles/10.1186/s13321-017-0242-y`, `arXiv:1706.08704v3`, `doi:10.1186/s13321-017-0242-y`.

[145] N. Freed and N. Borenstein. Multipurpose internet mail extensions (MIME) part one: Format of internet message bodies. Technical report, 1996. URL: `https://tools.ietf.org/html/rfc2045`.

[146] P. Deutsch. GZIP file format specification version 4.3. Technical report, 1996. URL: `https://tools.ietf.org/html/rfc1952`.

[147] Chuanxun Su, Jian Lv, Quan Li, Hui Wang, Lijun Zhang, Yanchao Wang, and Yanming Ma. Construction of crystal structure prototype database: methods and applications. *Journal of Physics: Condensed Matter*, 29(16):165901, Mar 2017. URL: `http://dx.doi.org/10.1088/1361-648X/aa63cd`, `doi:10.1088/1361-648x/aa63cd`.

[148] Robert M. Hanson. *Jmol* – a paradigm shift in crystallographic visualization. *Journal of Applied Crystallography*, 43:1250–1260, 2010. URL: `http://dx.doi.org/10.1107/S0021889810030256`, `doi:10.1107/S0021889810030256`.

[149] Kliment Olechnovič and Česlovas Venclovas. Voronota: A fast and reliable tool for computing the vertices of the Voronoi diagram of atomic balls. *Journal of Computational Chemistry*, 35:672–681, 2014. `doi:10.1002/jcc.23538`.

[150] C. Levinthal. Molecular model-building by computer. *Sci Am.*, 214:42–52, 1966.

[151] H. L. Morgan. The generation of a unique machine description for chemical structures – a technique developed at chemical abstracts service. *J. Chem. Doc.*, 5:107–113, 1965. `doi:10.1021/c160017a018`.

[152] Saulius Gražulis, Andrius Merkys, Antanas Vaitkus, and Mykolas Okulič-Kazarinas. Computing stoichiometric molecular composition from crystal structures. *Journal of Applied Crystallography*, 48:85–91, 2015. URL: `http://scripts.iucr.org/cgi-bin/paper?S1600576714025904`.

[153] The Cambridge Crystallographic Data Centre. Element data and radii [online]. 2008. URL: `https://web.archive.org/web/20080701015237/http://www.ccdc.cam.ac.uk/products/csd/radii/table.php4`.

[154] Pekka Pyykkö and Michiko Atsumi. Molecular single-bond covalent radii for elements 1–118. *Chemistry – A European Journal*, 15:186–197, 2009. URL: `http://dx.doi.org/10.1002/chem.200800987`, `doi:10.1002/chem.200800987`.

[155] F. H. Allen, S. Bellard, M. D. Brice, B. A. Cartwright, A. Doubleday, H. Higgs, T. Hummelink, B. G. Hummelink-Peters, O. Kennard, W. D. S. Motherwell, J. R. Rodgers, and D. G. Watson. The Cambridge Crystallographic Data Centre: computer-based

search, retrieval, analysis and display of information. *Acta Crystallographica Section B*, 35(10):2331–2339, Oct 1979. URL: `http://dx.doi.org/10.1107/S0567740879009249`, `doi:10.1107/S0567740879009249`.

[156] Elaine C. Meng and Richard A. Lewis. Determination of molecular topology and atomic hybridization states from heavy atom coordinates. *Journal of Computational Chemistry*, 12(7):891–898, sep 1991. `doi:10.1002/jcc.540120716`.

[157] Majorie M. Harding. The geometry of metal-ligand interactions relevant to proteins. *Acta Crystallographica Section D*, 55:1432–1443, 1999. URL: `http://dx.doi.org/10.1107/S0907444999007374`, `doi:10.1107/S0907444999007374`.

[158] H. A. Jahn and E. Teller. Stability of polyatomic molecules in degenerate electronic states. I. Orbital degeneracy. *Proc. R. Soc. Lond.*, 161:220–235, 1937. `doi:10.1098/rspa.1937.0142`.

[159] Ben Bax, Chun-wa Chung, and Colin Edge. Getting the chemistry right: protonation, tautomers and the importance of H atoms in biological chemistry. *Acta Crystallographica Section D*, 73(2):131–140, Feb 2017. URL: `https://doi.org/10.1107/S2059798316020283`, `doi:10.1107/S2059798316020283`.

[160] Dulal C. Ghosh and Raka Biswas. Theoretical calculation of absolute radii of atoms and ions. Part 1. The atomic radii. *International Journal of Molecular Sciences*, 3(2):87–113, Feb 2002. URL: `http://dx.doi.org/10.3390/i3020087`, `doi:10.3390/i3020087`.

[161] George M. Sheldrick. A short history of *SHELX*. *Acta Crystallographica Section A*, 64(1):112–122, Jan 2008. URL: `https://doi.org/10.1107/S0108767307043930`, `doi:10.1107/S0108767307043930`.

[162] B. Dittrich, C. B. Hübschle, K. Pröpper, F. Dietrich, T. Stolper, and J. J. Holstein. The generalized invariom database (GID). *Acta Crystallogr Sect B Struct Sci*, 69(2):91–104, Mar 2013. URL: `http://dx.doi.org/10.1107/S2052519213002285`, `doi:10.1107/s2052519213002285`.

[163] Sławomir Domagała, Bertrand Fournier, Dorothee Liebschner, Benoît Guillot, and Christian Jelsch. An improved experimental databank of transferable multipolar atom models – ELMAM2. construction details and applications. *Acta Crystallogr Sect A*, 68(3):337–351, Mar 2012. URL: `http://dx.doi.org/10.1107/S0108767312008197`, `doi:10.1107/s0108767312008197`.

[164] E. Anderson, G. D. Veith, and D. Weininger. SMILES: A line notation and computerized interpreter for chemical structures. Technical report, Environmental Research Laboratory-Duluth, 1987.

[165] Morris Plotkin. Mathematical basis of ring-finding algorithms in CIDS. *Journal of Chemical Documentation*, 11(1):60–63, feb 1971. `doi:10.1021/c160040a013`.

[166] Geoffrey M. Downs, Valerie J. Gillet, John D. Holliday, and Michael F. Lynch. Review of ring perception algorithms for chemical graphs. *J. Chem. Inf. Comput. Sci.*, 29:172–187, 1989.

[167] John Figueras. Ring perception using breadth-first search. *J. Chem. Inf. Comput. Sci.*, 36:986–991, 1996.

[168] T. Hanser, P. Jauffret, and G. Kaufmann. A new algorithm for exhaustive ring perception in a molecular graph. *Journal of Chemical Information and Modeling*, 36:1146–1152, 1996. URL: `http://pubs.acs.org/cgi-bin/doilookup/?10.1021/ci960322f`, `doi:10.1021/ci960322f`.

[169] Andrew R. Leach, Daniel P. Dolata, and Keith Prout. Automated conformational analysis and structure generation: algorithms for molecular perception. *Journal of Chemical Information and Modeling*, 30(3):316–324, Aug 1990. URL: `http://dx.doi.org/10.1021/ci00067a017`, `doi:10.1021/ci00067a017`.

[170] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, 2nd edition, 2009.

[171] Kenneth L. Lange, Roderick J. A. Little, and Jeremy M. G. Taylor. Robust statistical modeling using the t distribution. *Journal of the American Statistical Association*, 84(408):881, Dec 1989. URL: `http://dx.doi.org/10.2307/2290063`, `doi:10.2307/2290063`.

[172] H. A. Howlader and G. Weiss. On Bayesian estimation of the Cauchy parameters. *Sankhyā: The Indian Journal of Statistics, Series B (1960-2002)*, 50(3):350–361, 1988. URL: `http://www.jstor.org/stable/25052554`.

[173] Demetrios Gerogiannis, Christophoros Nikou, and Aristidis Likas. The mixtures of Student's t-distributions as a robust framework for rigid registration. *Image and Vision Computing*, 27(9):1285–1294, Aug 2009. URL: `http://www.cs.uoi.gr/~arly/papers/imavis09.pdf`, `doi:10.1016/j.imavis.2008.11.013`.

[174] D. Peel and G. J. McLachlan. Robust mixture modelling using the t distribution. *Statistics and Computing*, 10(4):339–348, 2000. URL: `http://dx.doi.org/10.1023/A:1008981510081`, `doi:10.1023/a:1008981510081`.

[175] Chad Aeschliman, Johnny Park, and Avinash C. Kak. A novel parameter estimation algorithm for the multivariate t-distribution and its application to computer vision. In *European Conference on Computer Vision 2010.* Purdue University, 2010. URL: `https://engineering.purdue.edu/RVL/Publications/Aeschliman2010ANovel.pdf`.

[176] Chuanhai Liu and Donald B. Rubin. ML estimation of the t distribution using EM and its extensions, ECM and ECME. *Statistica Sinica*, 5:19–39, 1995. URL: `http://www3.stat.sinica.edu.tw/statistica/oldpdf/A5n12.pdf`.

[177] Demetris Gerogiannis. Personal communication.

[178] Tong Liu, Ping Zhang, Wu-Sheng Dai, and Mi Xie. An intermediate distribution between Gaussian and Cauchy distributions. *Physica A: Statistical Mechanics and its Applications*, 391(22):5411–5421, Nov 2012. URL: `https://arxiv.org/abs/1208.5109`, `doi:10.1016/j.physa.2012.06.035`.

[179] A. Swami. Non-Gaussian mixture models for detection and estimation in heavy-tailed noise. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*, volume 6, pages 3802–3805, 2000. `doi:10.1109/ICASSP.2000.860231`.

[180] Mahdi Teimouri, Saeid Rezakhah, and Adel Mohammdpour. EM algorithm for symmetric stable mixture model. *Communications in Statistics – Simulation and Computation*, Feb 2017. URL: `http://dx.doi.org/10.1080/03610918.2017.1288244`, `doi:10.1080/03610918.2017.1288244`.

[181] Dankmar Bohning, Peter Schlattmann, and Bruce Lindsay. Computer-assisted analysis of mixtures (c.a.man): Statistical algorithms. *Biometrics*, 48(1):283–303, Mar 1992. URL: `http://dx.doi.org/10.2307/2532756`, `doi:10.2307/2532756`.

[182] Gilles Celeux, Merrilee Hurn, and Christian P. Robert. Computational and inferential difficulties with mixture posterior distributions. *Journal of the American Statistical Association*, 95:957–970, 2000. `doi:10.1080/01621459.2000.10474285`.

[183] Cédric Archambeau, John A. Lee, and Michel Verleysen. On convergence problems of the EM algorithm for finite Gaussian mixtures. In *ESANN'2003 proceedings - European Symposium on Artificial Neural Networks*, pages 99–106, April 2003.

[184] Maya R. Gupta and Yihua Chen. Theory and use of the EM algorithm. *Foundations and Trends in Signal Processing*, 4:223–296, 2010. `doi:10.1561/2000000034`.

[185] Christophe Biernacki, Gilles Celeux, and Gérard Govaert. Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models. *Comput. Stat. Data Anal.*, 41:561–575, 2003. `doi:10.1016/S0167-9473(02)00163-9`.

[186] R. E. Turner and M. Sahani. *Two problems with variational expectation maximisation for time-series models*. Cambridge University Press, 2011. URL: `http://www.gatsby.ucl.ac.uk/~maneesh/papers/turner-sahani-2010-ildn.pdf`.

[187] Kurt Hornik and Bettina Grün. movMF: An R package for fitting mixtures of von Mises-Fisher distributions. *Journal of Statistical Software*, 58(10), July 2014. URL: `https://www.jstatsoft.org/article/view/v058i10/v58i10.pdf`.

[188] Ferenc Nagy. Parameter estimation of the Cauchy distribution in information theory approach. *Journal of Universal Computer Science*, 12:1332–1344, 2006. URL: `http://www.jucs.org/jucs_12_9/parameter_estimation_of_the/jucs_12_09_1332_1344_nagy.pdf`, `doi:10.3217/jucs-012-09-1332`.

[189] José M. Bernardo. Algorithm AS 103. Psi (digamma) function. *Applied Statistics*, 25:315–317, 1976. URL: `http://www.uv.es/~bernardo/1976AppStatist.pdf`.

[190] Sibao Chen, Haixian Wang, and Bin Luo. Greedy EM algorithm for robust t-mixture modeling. In *Third International Conference on Image and Graphics (ICIG'04)*, pages 548–551. Institute of Electrical & Electronics Engineers (IEEE), 2004. URL: `http://dx.doi.org/10.1109/ICIG.2004.76`, `doi:10.1109/icig.2004.76`.

[191] Si-Bao Chen and Bin Luo. Robust t-mixture modelling with SMEM algorithm. In *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826)*, volume 6, pages 3689–3694. Institute of Electrical & Electronics Engineers (IEEE), 2004. URL: `http://dx.doi.org/10.1109/ICMLC.2004.1380451`, `doi:10.1109/icmlc.2004.1380451`.

[192] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017. URL: `https://www.R-project.org`.

[193] Daniel Pena and Irwin Guttman. Comparing probabilistic methods for outlier detection in linear models. *Biometrika*, 80(3):603, Sep 1993. URL: `http://www.jstor.org/stable/2337181`, `doi:10.2307/2337181`.

[194] Harold Jeffreys. *The Theory of Probability*. Oxford, 3 edition, 1961.

[195] Andreas Rauber, Ari Asmi, Dieter Van Uytvanck, and Stefan Proell. Data citation of evolving data: Recommendations of the RDA Working Group on Data Citation (WGDC). 2016. URL: `https://www.rd-alliance.org/group/data-citation-wg/outcomes/data-citation-recommendation.html`, `doi:10.15497/rda00016`.

[196] Jeremy D. Zawodny and Derek J. Balling. *High Performance MySQL: Optimization, Backups, Replication, and Load Balancing*. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2004.

[197] Bruno Bienfait and Peter Ertl. JSME: a free molecule editor in JavaScript. *Journal of Cheminformatics*, 5:24, 2013. URL: `http://www.jcheminf.com/content/5/1/24`, `doi:10.1186/1758-2946-5-24`.

[198] Noel M. O'Boyle, Michael Banck, Craig A. James, Chris Morley, Tim Vandermeersch, and Geoffrey R. Hutchison. Open Babel: An open chemical toolbox. *Journal of Cheminformatics*, 3:33, 2011. `doi:10.1186/1758-2946-3-33`.

[199] Kenneth L. Kelly. Twenty-two colors of maximum contrast. *Color Engineering*, 3:26–27, 1965. URL: `http://www.iscc.org/pdf/PC54_1724_001.pdf`.

[200] Ole Laursen. Attractive JavaScript plotting for jQuery [online]. URL: `http://www.flotcharts.org`.

[201] Ben Collins-Sussman, Brian W Fitzpatrick, and C Michael Pilato. *Version Control with Subversion*. O'Reilly Media, 2008. URL: `http://svnbook.red-bean.com`.

[202] Barry W. Boehm. *Software Engineering Economics*. Prentice Hall, 1981. URL: `http://csse.usc.edu/csse/research/COCOMOII/cocomo81.htm`.

[203] Antanas Vaitkus. Personal communication.

[204] Nicole Balasco, Luciana Esposito, and Luigi Vitagliano. Factors affecting the amplitude of the τ angle in proteins: a revisitation. *Acta Crystallographica Section D*, 73(7):618–625, Jul 2017. URL: `https://doi.org/10.1107/S2059798317007793`, `doi:10.1107/S2059798317007793`.

[205] Roman A. Laskowski, David S. Moss, and Janet M. Thornton. Main-chain bond lengths and bond angles in protein structures. *J. Mol. Biol.*, 231:1049–1067, 1993.

[206] R. A. Engh and R. Huber. *International Tables for Crystallography, Vol. F*, pages 382–392. Kluwer Academic Publishers, 2001.

[207] Oleg V. Dolomanov, Luc J. Bourhis, Richard J. Gildea, Judith A. K. Howard, and Horst Puschmann. OLEX2: a complete structure solution, refinement and analysis program. *Journal of Applied Crystallography*, 42(2):339–341, jan 2009. `doi:10.1107/s0021889808042726`.

[208] H. Zhong, X.-M. Yang, Q.-Y. Luo, and Y.-P. Xu. (1,10-phenanthroline)tri(3-phenylpropanoato)lanthanum(III). *Acta Crystallographica Section E Structure Reports Online*, 63(7):m1909–m1909, jun 2007. `doi:10.1107/s1600536807028693`.