BIOINFORMATICS APPLICATIONS NOTE
Vol. 27 no. 7 2011, pages 1021–1022
doi:10.1093/bioinformatics/btr055

Structural bioinformatics
Advance Access publication February 18, 2011

# fconv: format conversion, manipulation and feature computation of molecular data

Gerd Neudert and Gerhard Klebe*
Department of Pharmaceutical Chemistry, Philipps-Universität Marburg, Marbacher Weg 6, D-35032, Germany

Associate Editor: Anna Tramontano

**ABSTRACT**

**Summary:** fconv is a program intended for parsing and manipulating multiple aspects and properties of molecular data. Up to now, it has been developed and extensively tested for 3 years. It has become a very robust and comprehensive tool involved in a broad range of computational workflows that are currently applied in our drug design environment. Typical tasks are as follows: conversion and error correction of formats such as PDB(QT), MOL2, SDF, DLG and CIF; extracting ligands from PDB as MOL2; automatic or ligand-based cavity detection; rmsd calculation and clustering; substructure searches; alignment and structural superposition; building of crystal packings; adding hydrogens; calculation of various properties like the number of rotatable bonds; molecular weights or vdW volumes. The atom type classification is based on a consistent assignment of internal atom types, which are by far more differentiated compared with e.g. Sybyl atom types. Apart from the predefined mapping of these types onto Sybyl types, the user is able to assign own mappings by providing modified template files, thus allowing for tailor-made atom type sets.

**Availability:** fconv is free software available under GNU General Public License. C++ sources and precompiled executables for LINUX/UNIX, Mac OS and Windows, as well as tutorials are available on http://www.agklebe.de.

**Contact:** klebe@staff.uni-marburg.de

Received on August 19, 2010; revised on January 4, 2011; accepted on January 28, 2011

## 1 INTRODUCTION

Ongoing development of several programs in our group demanded for a library that enables a robust handling of molecular data and consistent atom type perception. Large-scale libraries like Open Babel (http://openbabel.sourcefourge.net) did not fulfill our needs with respect to a flexible novel definition of own atom types and robustness. Therefore, we developed a new framework which grew rapidly and thus we implemented fconv as a command line front end, featuring easy access to most functions even for users without programming skills.

The tasks performed by fconv cover a wide range in the field of handling molecular data. Thus, it is not possible to describe all features of this toolbox in detail. Rather we will give an overview with focus on the atom type perception and we provide the source code, not only to enable traceability of the applied methods but also to allow for user contributions and modifications. Additional features and details are explained in the available tutorials.

Several approaches concerning atom type perception have been reported in the last 20 years (Baber and Hodgkin, 1992; Froeyen and Herdewijn, 2005; Hendlich *et al*., 1997; Labute, 2005; Meng and Lewis, 1991; Sayle 2001; Zhao *et al*., 2007). In contrast to e.g. Hendlich's BALI which relies on CONECT entries in PDB files, fconv follows the philosophy of Zhao *et al*., thus only relying on essential information which are element types and coordinates. Apart from self-defined internal atom types, another difference compared with the above-mentioned methods is the isolation of atom and bond type perception.

## 2 DESCRIPTION

A typical workflow of fconv is structured as following: (i) Parse input files; (ii) perceive atom types; (iii) assign physicochemical properties and/or manipulate structures and/or apply geometric calculations; and (iv) produce output data/files.

Valid input formats are PDB(QT), MOL2, SDF, CIF and DLG. The accordant parser maps the data on a uniform object hierarchy, hence the internal representation is independent of the input type. Main focus is robustness and thus common format errors (especially in PDB files) are tolerated and corrected.

The internal atom types are basis for the major part of further processing and thus the heart of fconv. Currently, 157 internal types are differentiated and by default mapped onto the Sybyl types for output. fconv can write a definition file with this mapping and allows for modified definition files as input, thus enabling tailor-made atom type sets. A description of the internal types can be found in all definition files.

The atom typing is performed in several steps. First, atom connectivities are determined by means of given distance thresholds derived from the Cambridge structural database (CSD) (Allen, 2002). Next, atom hybridizations are derived from local geometries, using bond length again, and also bond angle thresholds according to standard values as derived from the CSD. This process does not fully rely on bonded hydrogens, as number and geometry are often not valid. At this point, there are atoms with well defined and atoms with ambiguous hybridization states. A carbon with three bound heavy atoms has either tetrahedral or planar geometry and is therefore well defined, whereas a carbon with only two adjacent heavy atoms remains ambiguous at first place, because bond lengths and angles (especially in PDB files) usually do not allow for a reliable assignment. However, in subsequent cycles it may become well defined due to the assigned states of its neighboring atoms. Thus, the assignment of hybridization states is performed iteratively.

*To whom correspondence should be addressed.

Subsequently, the smallest set of smallest rings is determined, and each ring is tested for planarity; with this information, additional hybridizations become well defined.

Up to this point, the workflow is very similar to what is described for Zhao's I-interpret. The next steps of I-interpret are recognition of predefined functional groups, bond type assignment and resolving conflicts between bond types and hybridizations. In contrast, fconv proceeds as following. Starting with a test for Hückel aromaticity for all planar rings, the next step is the assignment of internal atom types. Here, predefined patterns such as amides or acids are used.[1] As there are several functional groups with ambiguous protonation states, users can change the default state for distinct groups (e.g. change from charged to uncharged amidines). Finally, the internal types are mapped on Sybyl- or user-defined atom types. For the perception of bond types, first all definite bonds are assigned. For ambiguous bonds (as in conjugated systems), we construct a graph with nodes for each possible bond type and finally determine the clique with the highest total bond order.

As an alternative option, proteins may also be typesetted using a library for amino acids. On the one hand this is much faster, but on the other hand using the general atom type perception assures equal handling of proteins and small molecules. If there are multiple conformations of a ligand within a PDB file, also multiple ligand objects are generated. Peptidic parts of a ligand are joined with the HETATM parts before atom typing.

Various options of further processing were implemented based on well-established standard algorithms. For the alignment of small molecules and substructure searches, a graph matching using the Bron–Kerbosch algorithm (Bron and Kerbosch, 1973) was implemented. Here, the only conditions for the assignment of an atom–atom pair in the product graph are equal element types and connectivities. Demanding an exact match of atom types would not only result in higher speed, but also raise problems such as incomplete matchings of tautomeres. Superpositions are performed by a quaternion-based variant of the Kabsch algorithm (Horn, 1987).

Other options, like setting hydrogens or calculating the number of rotatable bonds, work on predefined attributes that are linked with the internal atom types. However, there are also options without using the atom type perception such as splitting and joining files, removing hydrogens or the construction of crystal packings.

All libraries used in fconv are part of the standard template library or self-implemented in ISO C++.

## 3 EVALUATION

In recent years, especially two successful methods for automatic atom type perception were presented (Labute, 2005; Zhao *et al.*, 2007). Thus, we have chosen the same dataset consisting of 179 entries from the PDB to give an estimation for the reliability of our program's atom type perception. For each PDB entry, the ligand was extracted as MOL2 (`fconv -l`) and the results were visually inspected and compared with the correct chemical structure of the given molecule. fconv failed in five cases, resulting in a success rate of 97.2 %. Labute's method was reported to fail in 11 cases (success rate 93.9 %[2]), 3 of them shared with fconv (1aaq, 1aqb, 2r04). Zhao's I-interpret was reported to fail in nine cases (success rate 95.0 %), but we found three of them (1etr, 1rne, 2xim) to be correct although stated incorrect, resulting in a success rate of 96.6 %. From the remaining six incorrect cases, fconv shares two (2r04, 8xia). Thus, there remains one case to explain, where only fconv is wrong. In flavin mononucleotide from 3fx2, one of the three ribose hydroxyl groups is perceived as carbonyl group due to the trigonal planar geometry of the corresponding carbon combined with a shorter C-O bond length.

Extracting all ligands from all 179 PDBs, atom type perception and writing the corresponding MOL2 files takes 5 s on an Intel Core2Duo E6600 (2.4 GHz). Searching for all PDB files containing at least one ligand with a phenolic substructure (`fconv -ssh`) takes 6 s. Converting an organic subset of 161000 CSD molecules from CIF to MOL2 takes 398 s.

## 4 CONCLUSION

For many tasks related with the handling of molecular data, fconv is an easy to use stand-alone tool. Its sources and precompiled executables are available under GNU GPL. The atom type perception routines are reliable and also allow for user-defined atom type sets. The scope of fconv is very broad and even more features will be added in the future. We appreciate any suggestions and contributions which may help to extend and improve fconv.

## ACKNOWLEDGEMENTS

We acknowledge the Cambridge Crystallographic Data Centre (CCDC) for the permission to use CSD-derived data for bond lengths and bond angles.

*Conflict of Interest*: none declared.

## REFERENCES

Allen,F.H. (2002) The Cambridge Structural Database: a quarter of a million crystal structures and rising. *Acta Crystallogr.*, **B58**, 380–388.

Baber,J.C. and Hodgkin,E.E. (1992) Automatic assignment of chemical connectivity to organic molecules in the Cambridge Structural Database. *J. Chem. Inf. Comput. Sci.*, **32**, 401–406.

Bron,C. and Kerbosch,J. (1973) Finding all cliques of an undirected graph. *Comm. ACM*, **16**, 575–577.

Froeyen,M. and Herdewijn,P. (2005) Correct bond order assignment in a molecular framework using integer linear programming with application to molecules where only non-hydrogen atom coordinates are available. *J. Chem. Inf. Model.*, **45**, 215–221.

Hendlich,M. *et al.* (1997) BALI: automatic assignment of bond and atom types for protein ligands in the Brookhaven Protein Databank. *J. Chem. Inf. Comput. Sci.*, **37**, 774–778.

Horn,B.K.P. (1987) Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am. A.*, **4**, 629–642.

Labute,P. (2005) On the perception of molecules from 3D atomic coordinates. *J. Chem. Inf. Model.*, **45**, 215–221.

Meng,E.C. and Lewis,R.A. (1991) Determination of molecular topology and atomic hybridization states from heavy atom coordinates. *J. Comput. Chem.*, **12**, 891–898.

Sayle (2001) Available at http://www.daylight.com/meetings/mug01/Sayle/m4xbondage .html (last accessed date August, 2010).

Zhao,Y. *et al.* (2007) Automatic perception of organic molecules based on essential structural information. *J. Chem. Inf. Model.*, **47**, 1379–1385.

[1] The functional groups we consider can be deduced from the internal types.

[2] Results cited from Zhao's study (Zhao *et al.*, 2007).