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.sourceforge.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 Heredewijn, 2005; Hendlich et al., 1997; Lubeke, 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.

© The Author 2011. Published by Oxford University Press. All rights reserved. For Permissions, please email: journals.permissions@oup.com
G. Neudert and G. Klebe

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 HEOTAM 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 require an exact superposition of both molecules. As an alternative option, proteins may also be typesetted using predefined patterns such as amides or acids.2

Conflict of Interest: none declared.

REFERENCES


1The functional groups we can consider are deduced from the internal types.

2Results cited from Zhao’s study (Zhao et al., 2007).