# cctbx PDB handling tools

Ralf W. Grosse-Kunstlevea and Paul D. Adamsa,b

*aLawrence Berkeley National Laboratory, Berkeley, CA 94720*

*bDepartment of Bioengineering, University of California at Berkeley, Berkeley, CA 94720*

Correspondence email: [RWGrosse-Kunstleve@LBL.Gov](mailto:RWGrosse-Kunstleve@LBL.Gov)

## Introduction

The PDB format is the predominant working format for atomic parameters (coordinates, occupancies, displacement parameters, etc.) in macromolecular crystallography. Many small-molecule programs also support this format. The PDB format specifications are available at <http://www.pdb.org/>. Technically, the format is very simple, therefore a vast number of parsers exist in scientific packages. This article is about the PDB handling tools included in the Computational Crystallography Toolbox (cctbx, <http://cctbx.sourceforge.net/>), the open-source component of the *PHENIX* project (<http://www.phenix-online.org/>).

|  |  |  |
| --- | --- | --- |
|  |  |  |

This article presents an overview of the main types in the iotbx.pdb module, considerations that lead to the design, and related important nomenclature. It is not a tutorial for using the iotbx.pdb facilities. For this, refer to <http://cctbx.sourceforge.net/sbgrid2008/tutorial.html>. See also <http://cci.lbl.gov/hybrid_36/> which describes iotbx.pdb facilities for handling very large models.

## Real-world PDB files

The simplicity of the PDB format is only superficial and, in the general case, stops after the initial parsing level. The structure of the PDB file implies a *hierarchy* of objects. A first approximation is this hierarchical view is:

model

chain

residue

atom

This is only an approximation because of a feature that is easily overlooked at first: the "altloc" (official PDB nomenclature) column 17 of PDB ATOM records, specifying "alternative location" identifiers for atoms in alternative conformations. As it turned out, about 90% of the development time invested into iotbx.pdb was in some form related to alternative conformations. Our goal was to provide robust tools that work even for the most unusual (but valid) cases, since this is a vital characteristic of any automated system. The main difficulties encountered while pursuing this goal were:

* Chains with conformers that have different sequences
* Chains with duplicate resseq+icode (residue sequence number + insertion code)
* Conformers interleaved or separated