```
{+ file: anneal.inp +}
{+ description: Crystallographic simulated annealing refinement +}
{+ authors: Axel T. Brunger, Luke M. Rice and Paul D. Adams +}
{+ reference: A.T. Brunger, J. Kuriyan and M. Karplus, Crystallographic
              R factor Refinement by Molecular Dynamics, Science
              235, 458-460 (1987) +}
{+ reference: A.T. Brunger, A. Krukowski and J. Erickson, Slow-Cooling
              Protocols for Crystallographic Refinement by Simulated
              Annealing, Acta Cryst. A46, 585-593 (1990) +}
{- begin block parameter definition -} define(
{====================== crystallographic data ========================}
{* space group *}
{* use International Table conventions with subscripts substituted by
   parenthesis *}
{===>} sg="P2(1)2(1)2(1)";

{* unit cell *}
{===>} a=61.76;   {===>} b=40.73;  {===>} c=26.74;
{===>} alpha=90;  {===>} beta=90;  {===>} gamma=90;

{* anomalous f' f'' library file *}
{* should be used when refining against anomalous data -
   libraries: "CNS_XTALLIB:anom_cu.lib" and "CNS_XTALLIB:anom_mo.lib" or
              a user created file.
   If blank no anomalous contribution will be included in the refinement *}
{===>} anom_library="";

{* reflection file *}
{===>} ref="example.hkl";

{* reciprocal space array containing observed amplitudes: required *}
{===>} obs_f="f_native";

{* reciprocal space array containing sigma values for amplitudes: required *}
{===>} obs_sigf="s_native";

{* reciprocal space array containing test set for cross-validation: required *}
{===>} test_set="test";
{* refinement target *}
{* mlf: maximum likelihood target using amplitudes
   mli: maximum likelihood target using intensities
   mlhl: maximum likelihood target using amplitudes and phase probability
         distribution
   residual: standard crystallographic residual
   vector: vector residual
   mixed: (1-fom)*residual + fom*vector
   e2e2: correlation coefficient using normalized E^2
   e1e1: correlation coefficient using normalized E
   f2f2: correlation coefficient using F^2
   f1f1: correlation coefficient using F *}
{+ choice: "mlf" "mli" "mlhl" "residual" "vector" "mixed"
           "e2e2" "e1e1" "f2f2" "f1f1" +}
{===>} reftarget="mlf";
}  {- end block parameter definition -}
```

Fig. 25.2.3.8. Example of a typical CNS task file: a section of the top portion of the simulated-annealing refinement protocol which contains the definition of various parameters that are needed in the main body of the task file. Each parameter is indicated by a name, an equal sign and an arbitrary sequence of characters terminated by a semicolon (*e.g.* 'a = 61.76;'). The top portion of each task file also contains commands for the HTML interface embedded in comment fields (indicated by braces, {...}). The commands that can be modified by the user in the HTML form are in bold.

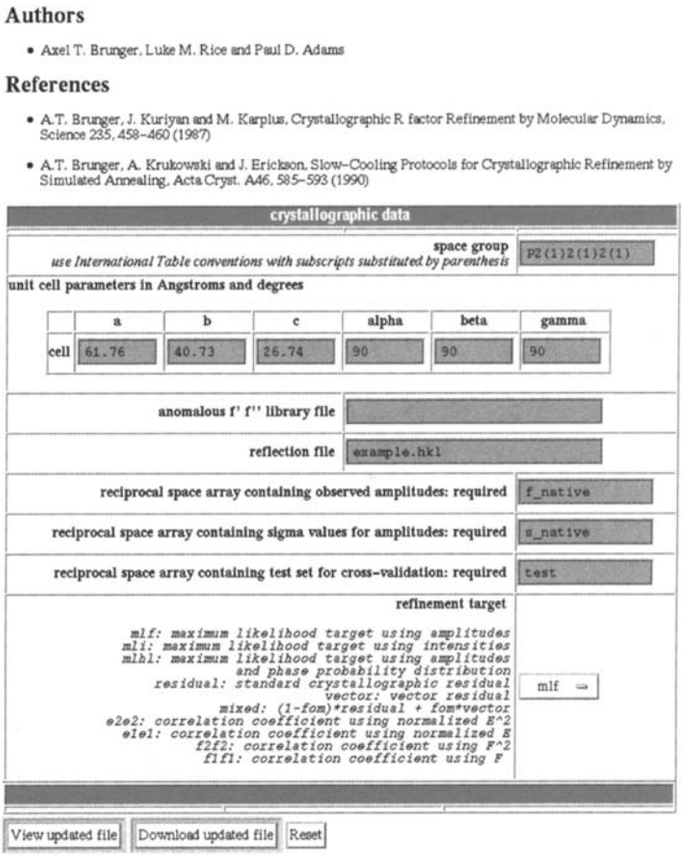

Fig. 25.2.3.9. Example of a CNS HTML form page. This particular example corresponds to the task file in Fig. 25.2.3.8.

25.2.3.7. *Task files*

Task files consist of CNS language statements and module invocations. The CNS language permits the design and execution of nearly any numerical task in X-ray crystallographic structure determination using a minimal set of 'hard-wired' functions and routines. A list of the currently available crystallographic procedures and features is shown in Fig. 25.2.3.7.

Each task file is divided into two main sections: the initial parameter definition and the main body of the task file. The definition section contains definitions of all CNS parameters that are used in the main body of the task file. Modification of the main body of the file is not required, but may be done by experienced users in order to experiment with new algorithms. The definition section also contains the directives that specify specific HTML features, *e.g.* text comments (indicated by {*...*}), user-modifiable fields (indicated by {===>}), and choice boxes (indicated by {+ choice: ...+}). Fig. 25.2.3.8 shows a portion of the 'define' section of a typical CNS refinement task file.

The task files produce a number of output files (*e.g.* coordinate, reflection, graphing and analysis files). Comprehensive information about input parameters and results of the task are provided in these output files. In this way, the majority of the information required to reproduce the structure determination is kept with the results. Analysis data are often given in simple columns and rows of numbers. These data files can be used for graphing, for example, by using commonly available spreadsheet programs. An HTML graphical output feature for CNS which makes use of these analysis files is planned. In addition, list files are often produced that contain a synopsis of the calculation.

25.2.3.8. *HTML interface*

The HTML graphical interface uses HTML to create a high-level menu-driven environment for CNS (Fig. 25.2.3.9). Compact and relatively simple Common Gateway Interface (CGI) conversion scripts are available that transform a task file into a form page and the edited form page back into a task file (Fig. 25.2.3.10). These conversion scripts are written in PERL.

A comprehensive collection of task files are available for crystallographic phasing and refinement (Fig. 25.2.3.7). New task files can be created or existing ones modified in order to address problems that are not currently met by the distributed collection of task files. The HTML graphical interface thus provides a common interface for distributed and 'personal' CNS task files (Fig. 25.2.3.10).

25.2.3.9. *Example: combined maximum-likelihood and simulated-annealing refinement*

CNS has a comprehensive task file for simulated-annealing refinement of crystal structures using Cartesian (Brünger *et al.*, 1987; Brünger, 1988) or torsion-angle molecular dynamics (Rice & Brünger, 1994). This task file automatically computes cross-

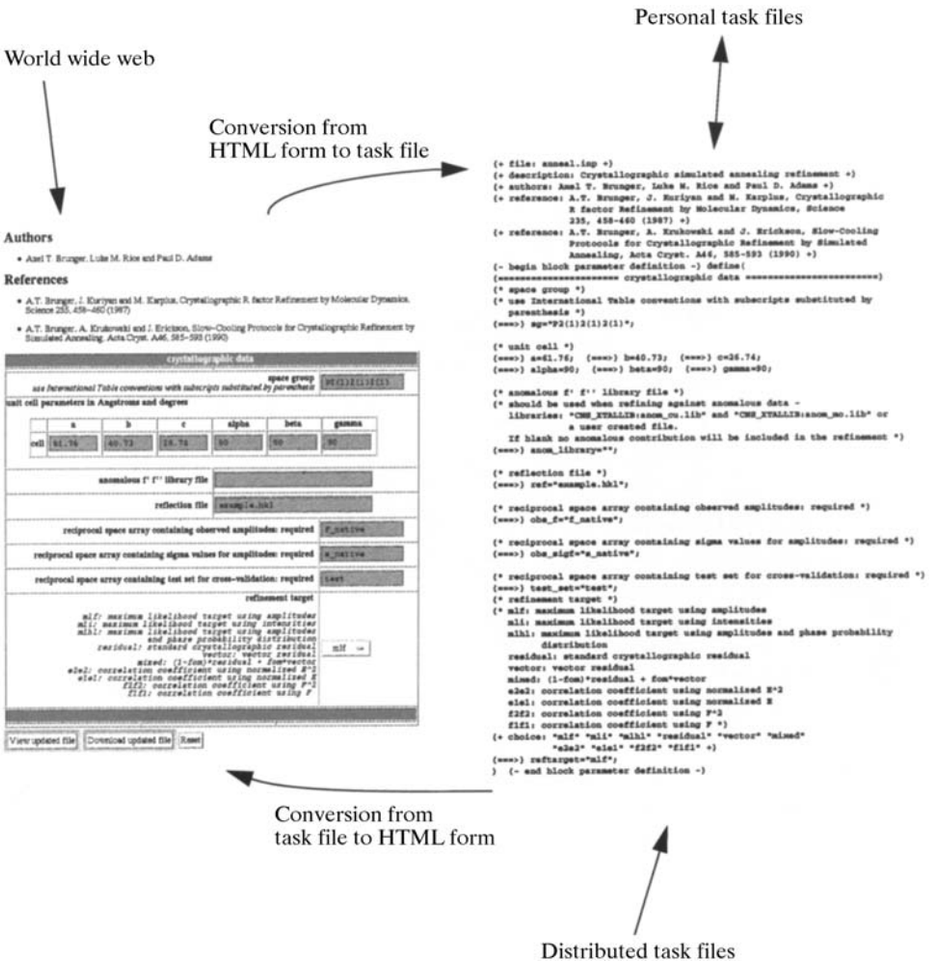

Fig. 25.2.3.10. Use of the CNS HTML form page interface, emphasizing the correspondence between input fields in the form page and parameters in the task file.

validated σ_A estimates, determines the weighting scheme between the X-ray refinement target function and the geometric energy function (Brünger *et al.*, 1989), refines a flat bulk solvent model (Jiang & Brünger, 1994) and an overall anisotropic B value for the model by least-squares minimization, and subsequently refines the atomic positions by simulated annealing. Options are available for specification of alternate conformations, multiple conformers (Burling & Brünger, 1994), noncrystallographic symmetry constraints and restraints (Weis *et al.*, 1990), and 'flat' solvent models (Jiang & Brünger, 1994). Available target functions include the maximum-likelihood functions MLF, MLI and MLHL (Pannu & Read, 1996*a*; Adams *et al.*, 1997; Pannu *et al.*, 1998). The user can choose between slow cooling (Brünger *et al.*, 1990) and constant-temperature simulated annealing, and the respective rate of cooling and length of the annealing scheme. For a review of simulated annealing in X-ray crystallography, see Brünger *et al.* (1997).

During simulated-annealing refinement, the model can be significantly improved. Therefore, it becomes important to recalculate the cross-validated σ_A error estimates (Kleywegt & Brunger, 1996; Read, 1997) and the weight between the X-ray diffraction target function and the geometric energy function in the course of the refinement (Adams *et al.*, 1997). This is important for the maximum-likelihood target functions that depend on the cross-validated σ_A error estimates. In the simulated-annealing task file, the recalculation of σ_A values and subsequently the weight for the crystallographic energy term are carried out after initial energy minimization, and also after molecular-dynamics simulated annealing.

25.2.3.10. *Conclusions*

CNS is a general system for structure determination by X-ray crystallography and solution NMR. It covers the whole spectrum of methods used to solve X-ray or solution NMR structures. The multi-layer architecture allows use of the system with different levels of expertise. The HTML interface allows the novice to perform standard tasks. The interface provides a convenient means of editing complicated task files, even for the expert (Fig. 25.2.3.10). This graphical interface makes it less likely that an important parameter will be overlooked when editing the file. In addition, the graphical interface can be used with any task file, not just the standard distributed ones. HTML-based documentation and graphical output is planned in the future.

Most operations within a crystallographic algorithm are defined through modules and task files. This allows for the development of new algorithms and for existing algorithms to be precisely defined and easily modified without the need for source-code modifications.

The hierarchical structure of CNS allows extensive testing at each level. For example, once the source code and CNS basic commands have been tested, testing of the modules and task files is performed. A test suite consisting of more than a hundred test cases is frequently evaluated during CNS development in order to detect and correct programming errors. Furthermore, this suite is run on several hardware platforms in order to detect any machine-specific errors. This testing scheme makes CNS highly reliable.

Algorithms can be readily understood by inspecting the modules or task files. This self-documenting feature of the modules provides a powerful teaching tool. Users can easily interpret an algorithm and compare it with published methods in the literature. To our knowledge, CNS is the only system that enables one to define symbolically any target function for a broad range of applications, from heavy-atom phasing or molecular-replacement searches to atomic resolution refinement.

25.2.4. The *TNT* refinement package

(D. E. Tronrud and L. F. Ten Eyck)

25.2.4.1. *Scope and function of the package*

TNT (Tronrud *et al.*, 1987) is a computer program package that optimizes the parameters of a molecular model given a set of observations and indicates the location of errors that it cannot correct. Its authors presume the principal set of observations to be the structure factors observed in a single-crystal diffraction experiment. To complement such a data set, which for most macromolecules has limitations, stereochemical restraints such as standard bond lengths and angles are also used as observations.

A molecule is parameterized as a set of atoms, each with a position in space, an isotropic B factor and an occupancy. The