MMMx

Gunnar Jeschke

Jul 23, 2020

ARCHITECTURE

Warning: MMMx is at an early design and prototyping stage. Version 0.0.0 is not functional.

- MMMx is the successor program of MMM (Multiscale Modeling of Macromolecules).
- Graphical user interface is separated from computation modules.
- Methods can be implemented in any language. They communicate by a few defined data exchange formats.

CHAPTER

ONE

CONCEPT

MMMx relies on well designed and well maintained software for standard tasks.

It is intended for modelling structure of proteins and their complexes and for supporting label-based spectroscopic techniques, such as EPR, FRET and NMR techniques that use spin labels.

- **Imports** Imports are functionalities that MMMx sources from external apps or packages:
  - Protein and nucleic acid structure loading, saving, manipulation and visualization by *ChimeraX*
  - Small-angle scattering curve simulation and fitting by the *ATSAS* package
  - Amino acid side chain generation or optimization by *SCWRL4*
  - Sequence alignment by *MUSCLE*
  - Structure refinement by *Yasara*
- **Exports** Exports are functionalities that MMMx provides by building or processing structure data:
  - Spin label and chromophore modelling via rotamer libraries
  - Simulation of distance distributions between labels
  - Localization of labels at structurally unresolved sites with espect to a structure
  - Conformational transitions by distance-restraint based deformation of elastic network models
  - Ensemble modelling of flexible peptide and RNA chains or chain segments
  - RigiFlex ensemble models based on distributed rigid-body arrangement and flexible segments

1.1 Open source concept

All MMMx functions and definitions are open source. However, MMMx may rely for some of its functionality on external apps that are not open source or not even free (Yasara for structure refinement is an example).

1.2 Separation of user interface from computation

The graphical user interface (GUI) of MMMx coordinates the work of MMMx methods and functionalities of external apps and visualizes numerical data.

MMMx methods can be called from scripts without running the GUI and without reliance on ChimeraX. For that, MMMx has basic PDB file read and write facility.

1.3 Visibility to users and programmers

Although MMMx is open source, users and programmers need to be aware that implementation details can change without notice.

Only part of MMMx functionality can reliably be used in scripts, contributed methods, or by external requests:

- **Data** The following types of data have a stable, though extensible definition:
  - *entity*: internal representation of macromolecular (ensemble) structure
  - *rotlib*: rotamer library
  - *ddr*: distance distribution restraint
- **Functionality** The following elements of MMMx functionality will be stable, though extensible:
  - *Commands*: allow for control and scripting in the MMMx GUI
  - *Methods*: build, modify or analyze an entity
  - *Restraint file*: specify experimental restraints and methods that operate on them

CHAPTER TWO

DATA

2.1 Entity

An *entity* holds the structure information on a protein, a nucleic acid molecule, or complexes thereof, possibly including small-molecule ligands and water molecules.

The entity consists of several *conformers* for an ensemble structure. Conformers have the same primary structure (amino acid or nucleotide sequence), but can differ in secondary, tertiary, and quaternary structure.

Each atom has a unique address and, in general, Cartesian coordinates (in Angstroem). Cartesian coordinates can be unspecified ("not a number").

Addresses

The atom address for specifying the CA atom of residue 131 in conformer 1 of chain A in structure 2LZM is **``[2LZM](A){1}131.CA``** in MMM. In ChimeraX, it is **``#1/A:131@CA``**. The user must know that structure 2LZM is model #1.

Methods can operate on the coordinates, which are hold in one common array for each conformer of a chain. Methods can also change residues and thus sequence in a chain (mutation). For any changes beyond that, a new entity should be derived.

Matlab realization

In Matlab, an entity is described by data type `struct` with dynamic field names. Field names that begin with a capital letter describe primary structure (chain identifiers, residue numbers, conformer numbers, atom identifiers). Field names that begin with a lower-case letter provide additional information, such as coordinates, residue names, the element, etc.

The MMMx entity file format is based on this extensible Matlab structure variable. Extension is by *attributes*. Each level (entity, chain, residue, conformer, atom) can have additional attributes, whose lower-case names are dynamical field names in Matlab.

2.2 Rotamer library

A rotamer library contains sidechain conformers for a spin label or chromophore label, together with their populations in the absence of interactions with a macromolecule.

Further *attributes* specify the attachment frame (by three atoms), the label position and possibly the label orientation. They may also provide information on how the library was generated.

Representation in Matlab and as MMMx file format is analogous to an *entity*.

2.3 Distance distribution restraints

Distance distribution restraints are specified at least by a distance axis (in Angstroem) and probabilities that a distance falls into a given bin on the diatance axis. The sum of all probabilities is unity.

Further, optional attributes are lower and upper probability bounds, a handle to a function that generates a parametric distance distribution, and parameter values and uncertainties.

CHAPTER THREE

CHIMERAX

MMMx communicates with ChimeraX for visualization. MMMx can use ChimeraX for PDB and mmCIF file loading and saving, for retrieving information on an entity, for entity building, and entity modification.

The graphical user interfaces (GUIs) of ChimeraX and MMMx can be used simultaneously. As ChimeraX is the more powerful GUI, it holds the master copy of the entity. MMMx retrieves the entity from ChimeraX before applying a method. For MMMx methods that potentially change the entity, MMMx then deletes the entity in ChimeraX, and resubmits the entity to ChimeraX after the method has completed.

- **Communication protocol** MMMx uses REST (representational state transfer) remote control both for setting states of ChimeraX and for polling ChimeraX for information:
  - ChimeraX needs to call the command `remotecontrol rest start port 51051` upon startup
  - communication depends on log response of ChimeraX commands
  - MMMx claims the name of entities (models) that it loads into ChimeraX
- **Using MMMx without ChimeraX** MMMx can be used standalone, including its own GUI, but the following restrictions apply:
  - no 3D structure visualization
  - only structure information is read from and written to PDB files
  - no mmCIF support
  - selection is only by commands and supports only the selection modes of MMM
  - for methods that require secondary structure analysis, DSSP must be installed and registered

3.1 Remarks

- The startup commands can be set in ChimeraX via the menu item `Favorites\Settings...` after navigating to tab `Startup`.
- The log response of ChimeraX is very well structured, but we are not guarded against future format changes
- MMMx models in ChimeraX have unique names MMMx_*mmmid*, where *mmmid* is the internal identifier in MMMx This guards against accidental replacement of an MMMx model by another model by the user. The other model could then have the same number, but would not have the MMMx name.
- If ChimeraX is running and connected, MMMx accesses DSSP secondary structure analysis through ChimeraX
- Methods can specify what information they need on an entity. Only this information is polled from ChimeraX.

CHAPTER FOUR

PYTHON

MMMx could be implemented in Matlab, in Python, or as a mixture of both.

Here are some advantages and disadvantages of using Python:

- Advantages
  - larger potential programmer base
  - larger base of existing bioinformatics modules
  - fully open source
- Disadvantages
  - data types are not as good as in Matlab
  - documentation is very heterogeneous across packages
  - hard to keep consistent, while packages evolve
  - lower programming productivity
  - much overhead bt discussions, which packages should or should not be used
  - porting existing MMM functions is not generally trivial (data typing)
  - once, everybody programmed Java, now everybody programs Python, will it be Julia tomorrow?

4.1 Remarks

- After looking more closely into the language, I am astonished what people can do with it despite its design.
- The syntax is nice and reads well.
- With `numpy`, Python is well suited to numerical data processing.
- String processing is much better supported in Matlab since 2018
- Compilers have gone out of business, a non-backwards compatible change of the language has been done, there are plans for further changes
- quality and generality of existing modules is very heterogeneous
- many larger projects tend to use Python only as a layer above a C++ core

4.2 Current conclusion

MMMx should be able to integrate methods programmed in Python. These methods can be supplied with *entities*, *rotamer libraries*, *distance distribution restraints*, and *restraint files* by MMMx.

A full port of MMM to Python would delay the first version of MMMx by years.