Automating NMR workflows: Introduction to python programming

Table of contents

Introduction

This beginners course in python programming for structural bioinformatics will cover a number of basic concepts that can be widely used for quick processing and analyses of data in text formats. The concepts will be illustrated and practiced in the context of a common format conversion in biomolecular NMR structure determination:

> ***The conversion of inter atomic distance restraints from CYANA format to the XPLOR format.***

This conversion is often needed to refine molecular structure models calculated with CYANA in other softwares that use the XPLOR/CNS format. A detailed explanation of this course project is given below.

It is very important before any programming or scripting project, whether small or large, that you understand in detail what is the goal of automation and the logics behind the program that you will develop. It is very helpful, in particular for automation that involves multiple steps, to write down the individual steps and understand what information is needed and how it needs to be processed. For this course you need to understand some of the background of NMR data formats, which is explained below. Please go through it carefully, as it will speed up the learning process during the course.

The course project

Introduction

Biomolecular NMR structure determination makes use of distances between atoms in molecules to calculate structure models. The way these distances are obtained is usually through the Nuclear Overhauser Effect (NOE), which allows to derive upper and lower limits of the distance between 2 or more atoms in a molecule.

To calculate a structure based on these derived upper and lower limits of an inter atomic distance, structure calculation programs require that this information is defined in a readable format for the specific program. These formats typically differ from program to program, and sometimes conversions between formats are required to make the best use of the available information, for example from CYANA to XPLOR.

To convert distance information in CYANA format to XPLOR format we first need to analyse the 2 different formats and the information that is contained in their data files. Also, we need to know some things about what distance information is used.

Distance restraints

Distance information is stored in so called distance restraints. A distance restraint in its simplest form is a way of defining a distance between 2 atoms in a molecule which a calculated structure model should fulfill.
For example, in text:

- *Atom Hα of residue Alanine 35 is within a distance of 2.5-4.5Å from Atom H of residue Valine 36.*

A bit more advanced form is a distance restraint involving a group of atoms:

- *The methyl hydrogen atoms Hβ* of residue Alanine 35 are within a distance of 2.5-4.5Å from atom H of residue Valine 36.*
  Note: The * is a wildcard for the 3 atoms Hβ1, Hβ2, Hβ3, which cannot be distinguished individually in NMR due to fast rotation of the methyl group.

Another common form is distance restraints that contain ambiguity:

- *Either atom Hβ2 or atom Hβ3 of residue Arginine 37 is within a distance of 2.5-4.5Å from atom H of Residue Valine 36.*
  Here it is not clear which of the two Hβ atoms was observed and should be included in the distance restraints.

There are more examples that could be mentioned, but are not required for the moment. What is important to know is that each restraint consists of atom selections and a distance definition.

CYANA and XPLOR distance restraint formats

To convert CYANA to XPLOR distance restraints definitions we need to analyze the information and representations in both formats:

CYANA distance restraint format

The CYANA upper limit distance restraints format looks like in the table below:

Atom selection 1	Atom selection 2	Upper distance limit	Comment
`45 ILE HA`	`46 VAL HB`	`4.78`	`#SUP 3.60`

```
28 ALA QB          29 PHE HA          4.93          #SUP 4.23
61 GLU HA          64 ARG HB2         3.38          #SUP 6.27
61 GLU HA          64 ARG HB3         0.00          #SUP 4.83
32 CYS HA          48 VAL QG1         3.93          #SUP 5.07
32 CYS HA          46 VAL QG1         0.00          #SUP 4.08
32 CYS HA          48 VAL QG2         0.00          #SUP 5.20
```

It consists of:

- 2 atom selections, each containing the residue number, residue type and atom
- an upper distance limit
  - If > 0.00, this is the upper distance between the 2 atom selections
  - Ambiguous upper distance restraints are defined using the first line with a distance > 0.00, followed by lines with upper distance set to 0.00. In the table above, we have 2 ambiguous restraint definitions.
- a comment field, preceded by "#".

CYANA has the option to read lower limit distance restraints, but often these are not used and no lower limit is imposed on a distance explicitly. The lower limit is usually implicitly 1.8Å, which is the distance between the centers of 2 Hydrogen atoms that are in direct contact. In structure calculations, the minimum distance between 2 hydrogen nuclei should not be significantly smaller than 1.8Å. For example, this means that the allowed distance range for the first restraint in the table above is effectively from 1.8-4.78Å.

XPLOR distance restraint format

The XPLOR distance restraint format looks rather different from CYANA:

Assign	Atom selection 1	Atom selection 2	Distance range	Comment
assign	((resid 45 and name HA))	((resid 46 and name HB))	4.78 2.98 0.00	! #SUP 3.60
assign	((resid 28 and name HB*))	((resid 29 and name HA))	4.93 3.13 0.00	! #SUP 4.23
assign	((resid 61 and name HA))	((resid 64 and name HB2))	3.38 1.58 0.00	! #SUP 6.27
or	((resid 61 and name HA))	((resid 64 and name HB1))		! #SUP 4.83
assign	((resid 32 and name HA))	((resid 48 and name HG1*))	3.93 2.13 0.00	! #SUP 5.07
or	((resid 32 and name HA))	((resid 46 and name HG1*))		! #SUP 4.08
or	((resid 32 and name HA))	((resid 48 and name HG2*))		! #SUP 5.20

It consists of:

- An assign statement: "assign" or "or"
  - "assign" is used for starting a distance restraint definition
  - "or" is used for treating ambiguous restraints, and is equivalent to the 0.00 for the upper distance in ambiguous restraints in CYANA.
- 2 atom selections, each containing the residue id or number (resid) and atom name (s). Note that some atoms have a different name in the two formats, and that the amino acid name is not present in both formats!!!
- a distance range definition consisting of 3 numbers d, d_{minus}, d_{plus}
  - d_{minus}, d_{plus} are used to calculate the upper and lower distances used in the restraint:
    - Upper limit = $d+d_{plus}$
    - Lower limit = $d-d_{minus}$
  - is empty when the assign statement is "or"
- a comment field, preceded by "!".

The distance range definition allows for some flexibility, but in this example the numbers are chosen that all distance ranges have an effective lower limit of 1.8Å, and the upper

limit equal to d (d_{plus}=0). In principle, one could also choose d_{minus} equal to d, which would lead to the same lower limit of 0.00 as in the CYANA example above.

Conversion requirements

To convert one format to the other, you thus need to be able to distinguish the different types of information in the definitions of both formats:

- Syntax: this means how is the restraint written
- Atom selection information: How are atoms and residues represented
- Distance limits definitions: How are distance restraint limits defined
- Comments: How are comments represented

A good exercise at this point is to manually convert one restraint from CYANA to XPLOR format using a text editor, and see if you understand how to do this:

Exercise

Convert the following CYANA distance restraint into a XPLOR format distance restraint using a text editor:

```
25 GLN  HA     28 ALA  H        4.16 #SUP  2.53
61 GLU  HA     65 ASN  H        0.00 #SUP  4.56
```

Identify

- Atom selections (residue numbers, residues and atom names)
- Distance limit information
- Ambiguity (try to understand what the meaning of the restraint is)
- Comments

Atom selections

Atom nomenclature

As mentioned above, atom names (nomenclature) can be different in different software packages, even though there is the IUPAC standard for naming atoms in chemical compounds. The latest versions of CYANA for example use this IUPAC standard definition by default. In XPLOR or CNS different nomenclatures are still in use and it is important to distinguish these.
For example:

- The IUPAC name HB3 of Asparagine is called HB1 in XPLOR format.

Pseudo atoms

Pseudo atoms form a special class, as these indicate groups of atoms. Pseudo atoms are used in cases where no distinction in NMR can be made between the atoms a group of atoms: so the group of atoms is used as a single “pseudo atom”. The simplest example of this is the Alanine methyl group, for which the 3 hydrogens always are observed as a single signal. Instead of using the three HB1, HB2, HB3 atoms, a pseudo atom is used.

- For these pseudo atoms also different nomenclatures are used, for example the Alanine methyl group is called QB in CYANA, whereas it can be called HB*, HB#, HB % or HB+ in XPLOR.

The atom nomenclature for amino acids are defined in the file **‘Atom nomenclature.tbl’** in your **‘Additional Material’** directory on the Desktop, and looks like:

```
ALA        H     HN
ALA        HA    HA
ALA        HB1   HB1
ALA        HB2   HB2
ALA        HB3   HB3
ALA        C     C
ALA        CA    CA
ALA        CB    CB
ALA        N     N
ALA        O     O
ALA        QB    HB*
```

The first column is the amino acid type, the second column is the atom name in CYANA format, and the third column is the atom name in XPLOR format.

IMPORTANT NOTE: The atom nomenclature table has been simplified for educational purposes and may not be suitable for use in all situations!!

Exercise

Using the atom nomenclature file, convert the following 2 (!) CYANA distance restraints into XPLOR formatted distance restraints using a text editor:

```
32 CYS  HA     33 GLU  QB      4.63 #SUP  3.09
32 CYS  HA     48 VAL  HB      0.00 #SUP  2.89
25 GLN  HB3    28 ALA  QB      5.04 #SUP  1.23
```

Pay attention to the atom nomenclature.

Automation steps

Now that you are a bit more familiar with the course project's goals, it is important to think about the steps that need to be taken to build a script that automatically does the CYANA to XPLOR conversions for you. These steps depend on what you exactly want the script to do:

- how often you want to use it
- whether it has to deal with exceptions
- whether it should have flexible input
- etcetera

A script that you will only need to use one time doesn't require that it can deal with situations that you do not encounter in your particular situation, and thus it can be just a quick and dirty piece of code. For a script that is intended for general and repeated use it is worth it to invest a bit more time in ease of use and in dealing with exceptions.

The goal of the course project script is that it can do the following basic things (limited):

- Read a CYANA distance restraints file
- Read a file that contains atom nomenclatures of CYANA and XPLOR
- Convert a distance restraint from CYANA to XPLOR format
- Write an XPLOR distance restraint file with all distance restraints

The script will also include:

- Variable input and output
- It should run with only providing an input filename and an output filename

To do this we need the following global steps:

- Input file reading and parsing (parsing is interpreting the content of the file)
- Atom nomenclature file reading and parsing
- Conversion of syntax from CYANA to XPLOR
- Conversion of atom nomenclature from CYANA to XPLOR
- Creation of distance restraint in XPLOR format
- Writing of converted distance restraints to a new XPLOR distance restraints file
- Creating functionality for using variable input and output

In the course you will be guided through these steps using lectures and exercises.

Format of the course

The course consists of a series of lectures, demonstrations and exercises in which you learn to apply the methods presented in lectures yourself. All the exercises are designed to let you build your own script(s) step by step, explaining all the concepts in detail and providing ample time to get familiar with the basics of programming. In short, the program will be as follows:

- First you will be introduced to the working environment, in which you will learn how to efficiently set up and use a programming workspace. Following the initial introductions into programming you will learn some important basic concepts of python programming and apply them in the exercises.
- After initial introductions, we will focus on combining the learned concepts and pieces of code into building blocks (functions) that allow you to program more efficiently. In the exercises you will combine different concepts into functions and create and test a fully functional script. The detailed overview of the course is given in Table 1.

Table 1. Program details

Lecture	*Concept*	*Exercise*	*Application*
Course introduction	• *Scope*	• *Demo - Introduction to the working environment*	
Python Introduction	• *Applications* • *Resources* • *First touch*	• *Exercise 1 - Getting started*	• Introduction
Running python	• *Interpreter* • *Running modes* • *versions*	• *Exercise 2 - Running scripts*	• Introduction
Variables	• *Types* • *Definition* • *Comparison* • *Manipulation*	• *Exercise 3 - Variables*	• Introduction
Strings	• *Testing* • *Searching* • *Manipulation* • *Formatting*	• *Exercise 4 - Strings*	• Convert a distance to XPLOR distance limits. • Create a XPLOR type distance restraint.
Conditionals	• *if* • *else* • *elif*	• *Exercise 5 - Conditionals*	• Create a XPLOR type distance restraint depending on ambiguity.
Functions	• *Purpose* • *Definition* • *Results*	• *Exercise 6 - Functions*	• Combine the results from Exercises 4 and 5 into functions that can create XPLOR distance restraints with variable input.
Intermezzo: Modules	• *Purpose* • *Resources* • *Import* • *Creation* • *PYTHONPATH*	• *Exercise 7 - Modules*	• Importing existing functions for string manipulation. • Reusing your code in different programs

Lecture	Concept	Exercise	Application
Interactive input	• *Purpose* • *Command line arguments* • *Input during program runs*	• *Exercise 8 - Interactive input*	• Create a function for using interactive input for variable filenames.
Files	• *Reading* • *Writing* • *Appending*	• *Exercise 9 - Files*	• Create a function that automatically reads a file and stores the content.
Lists	• *Applications* • *Definition* • *Accessing* • *Manipulation* • *Sorting*	• *Exercise 10 - Lists*	• Create function that can read a line from a CYANA distance restraint file and split the different types of information for easy processing
Dictionaries	• *Applications* • *Definition* • *Accessing* • *Manipulation* • *Sorting* • *Nesting*	• *Exercise 11 - Dictionaries*	• Create initial dictionaries that can be used to translate CYANA format to XPLOR format for a single atom: conversionDictionary
Loops	• *while loops* • *for loops*	• *Exercise 12 - Loops*	• Expand the conversionDictionary
Intermezzo: Errors and Exceptions	• *Purpose* • *Avoiding error messages* • *Dealing with exceptions in runtime*	• *Exercise 13 - Exceptions*	• Finalize the conversionDictionary
Notes and tips	• *Best practices*	• *Exercise 14 - Final*	• Combine all the previous functions into a conversion script • Create a function that converts a CYANA distance restraint file into a XPLOR distance restraints file, using all the previous functions • Add functionality for flexible input

Resources

Additional python resources may be found on the python project page www.python.org. In particular the following pages you may find useful:

- http://docs.python.org/py3k/
- http://docs.python.org/py3k/tutorial/index.html
- http://docs.python.org/py3k/library/index.html
- http://docs.python.org/py3k/py-modindex.html
- and last but not least: http://openbookproject.net/thinkcs/python/english3e/

Copyright notice