NEF-Pipelines 0.1.20beta

How To

Export Data to MARS for Auto assignment using NEF-Pipelines

Introduction

This *How To* will show you how to use NEF-Pipelines, a set of tools for manipulating NEF (NMR Exchange Format) files, and how to read files from programs that aren't supported by NEF. The tutorial will show you how to output shifts and a sequence to formats used by the automated assignment program MARS. MARS is a program designed to make backbone assignments of proteins using chemical shifts from triple resonance experiments including HNCA, HNcoCA, HNCACB, HNcoCACB, HNcaCO and HNCO and RDC data. It makes conservative assignments (it prefers not to assign rather than getting a wrong assignment) and is from the Laboratory of Marcus Zweckstetter (http://www3.mpibpc.mpg.de/groups/zweckstetter/_links/software_mars.htm). In the tutorial we will explain what a pipeline is and demonstrate how a series of command can be coupled together to work with a single NEF file including a simple data translation on the file.
The example is only a small part of what NEF-Pipelines can do as it is a flexible set of tools which can be composed together.

NEF-Pipelines is a program being written by Gary Thompson (University of Kent), a member of the CCPN Working group. It is still unpublished.

You will need to use the the file **sec5.nef** which is provided in the NEF tutorial data which can be downloaded from the CCPN tutorials website https://ccpn.ac.uk/support/tutorials/.

Start A Terminal

- Apple users should start the terminal which is in the Utilities folder in Applications. It can be started by typing ⌘ **spacebar** and typing **terminal**
- Unix users type **Ctrl+Alt+T**
- Windows users open the start menu and type **powershell** in Search and open the Powershell app

Disclaimer
Datasets used for this tutorial are from Helen Mott and are part of the standard CCPN tutorial datasets. Please note that the images shown are only representative and you may encounter minor differences in your setup.

Introduction

Getting started, basic operations

NEF-Pipelines is a set of command line tools

Each tool is started by typing the command **`nef`** followed by a series of sub-commands. Typing **`nef`** on the command line on its own shows help which lists the available tools (this and other help displays may be summoned by using the **`--help`** option).

```
nef
Usage: main.py [OPTIONS] COMMAND [ARGS]...

Options:
  --install-completion [bash|zsh|fish|powershell|pwsh]
                                  Install completion for the specified shell.
  --show-completion [bash|zsh|fish|powershell|pwsh]
                                  Show completion for the specified shell, to
                                  copy it or customize the installation.
  --help                          Show this message and exit.

Commands:
  chains    - carry out operations on chains
  entry     - carry out operations on the nef file entry
  fasta     - read and write fasta sequences
  frames    - carry out operations on frames in nef frames
  header    - add a header to the stream
  mars      - export mars [shifts and sequences]
  nmrpipe   - read nmrpipe [peaks shifts & sequencess]
  nmrview   - read and write nmrview [peaks, sequences & shifts]
  pales     - read and write pales/dc [rdcs]
  pdb       - read pdb [sequences]
  stream    - stream a nef file
```

Commands may have sub commands

Most tools have sub commands, for example the **`mars`** command has an **`export`** sub command which then has the further sub commands **`sequence`**, **`shifts`** and **`input`**. So, for example, typing **`nef mars export`** or **`nef mars export --help`** shows:

```
Usage: main.py mars export [OPTIONS] COMMAND [ARGS]...

  - export mars [shifts and sequences]

Options:
  --help  Show this message and exit.

Commands:
  input     - convert nef file to mars input
  sequence
  shifts    - convert nef chemical shifts to mars
```

Introduction – II

Typing **`nef mars export sequence`** or

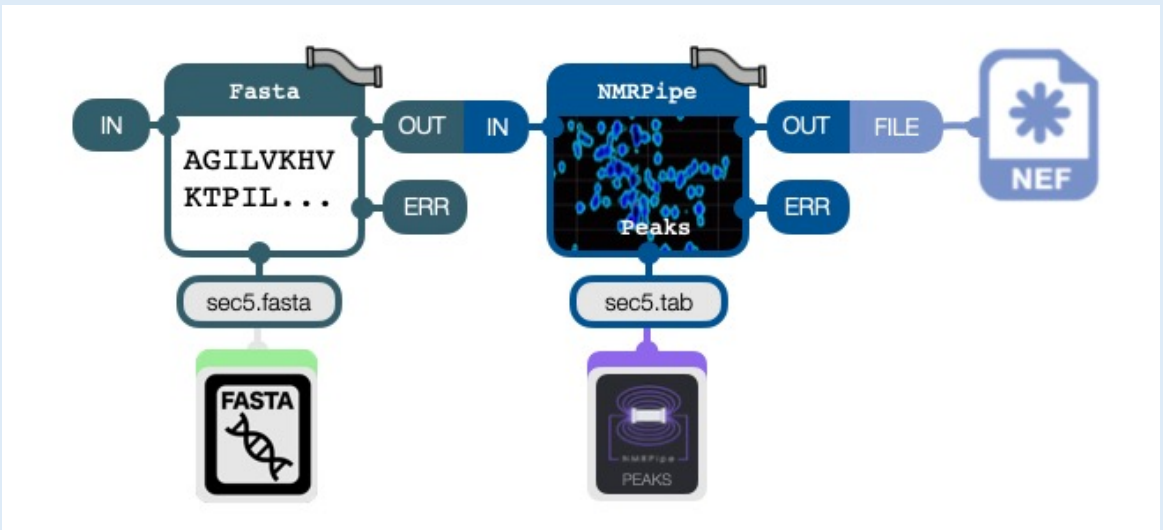
`nef mars export sequence --help` shows

```
nef mars export sequence --help
Usage: main.py mars export sequence [OPTIONS] <FASTA-SEQUENCE-FILE>

Arguments:
  <FASTA-SEQUENCE-FILE>  file name to output to [default <ENTRY-ID.fasta>] for
                         stdout use -

Options:
  -c, --chain_code <CHAIN-CODE>  single chain to export  [default: 'A']
                                 [default: A]
  -i, --in <NEF-FILE>            file to read nef data from
  --help                         Show this message and exit.
```

This shows input arguments to feed the command with data and options that maybe used to change how the command works. For example in this case **MARS** only works with a single molecular chain and we can use **`–c`** or **`–chain`** to select which chain to export from a NEF file if there is more than one. We won't use arguments or options today but they are an important part of NEF-pipelines.

Commands may be combined using the pipe symbol

Multiple commands maybe combined using the pipe symbol: | under Windows, Linux and OSX. This feeds the output of one command to the next command after the pipe symbol. The > character maybe used to write to a file. Below is shown symbolically a pipeline that reads a fasta file produces a stream which contains a NEF file that is fed to the nmrpipe command that adds nmrpipe peaks to the stream and then writes out to a NEF file containing both sets of data. This is followed by the command line you would use to run these commands.




```
nef fasta import sequence sec5.fasta | nef nmrpipe import peaks sec5.tab > sec5.nef
```

Installation

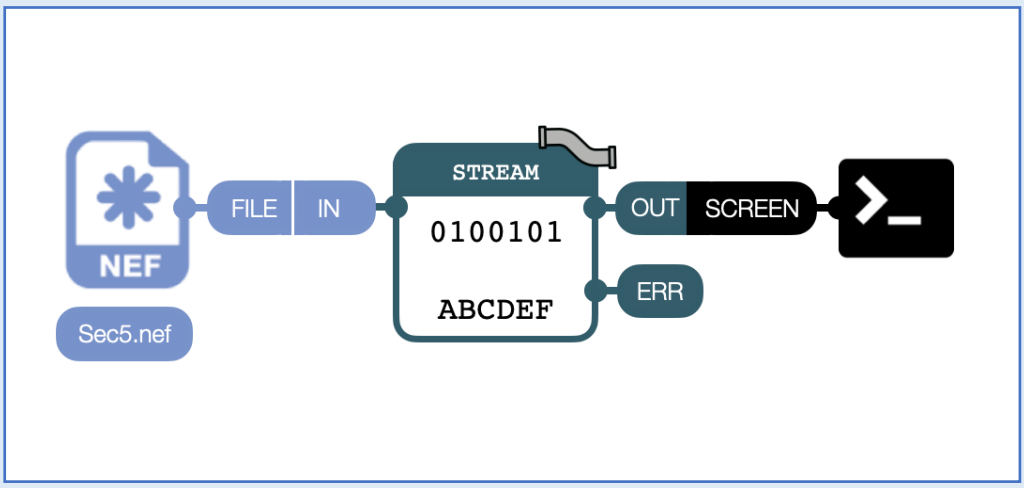
Installing NEF-Pipelines

NEF-Pipelines may be installed directly from the NEF-Pipelines github repo **https://github.com/varioustoxins/NEF-Pipelines** or with the command **pip install NEF-Pipelines.** However, it can be installed with a script provided with CCPN Analysis. Just type **install_nef_pipelines,** this will produce a lot of text output. Once NEF-Pipelines has installed you can type **nef** to get

```
nef
Usage: main.py [OPTIONS] COMMAND [ARGS]...

Options:
  --install-completion [bash|zsh|fish|powershell|pwsh]
                                  Install completion for the specified shell.
  --show-completion [bash|zsh|fish|powershell|pwsh]
                                  Show completion for the specified shell, to
                                  copy it or customize the installation.
  --help                          Show this message and exit.

Commands:
  chains    - carry out operations on chains
  entry     - carry out operations on the nef file entry
  fasta     - read and write fasta sequences
  frames    - carry out operations on frames in nef frames
  header    - add a header to the stream
  mars      - export mars [shifts and sequences]
  nmrpipe   - read nmrpipe [peaks shifts & sequencess]
  nmrview   - read and write nmrview [peaks, sequences & shifts]
  pales     - read and write pales/dc [rdcs]
  pdb       - read pdb [sequences]
  stream    - stream a nef file
```

Streaming a NEF File to screen

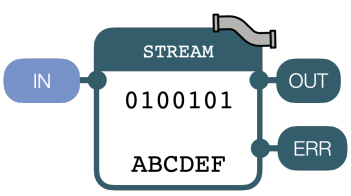
First of all we are going to stream a NEF file **sec5.nef** to screen, the diagram below show conceptually what we are doing



Note how much data this NEF file contains, its is the information about a complete NMR assignment project with peaks, shifts, a molecular system and other data its quite complex. NEF-Pipelines is designed to manage this complexity…

A NEF pipeline tool is represented by this symbol
It has a name an input and an output.

We shall use these symbols throughout the tutorial



1A Streaming the file to screen

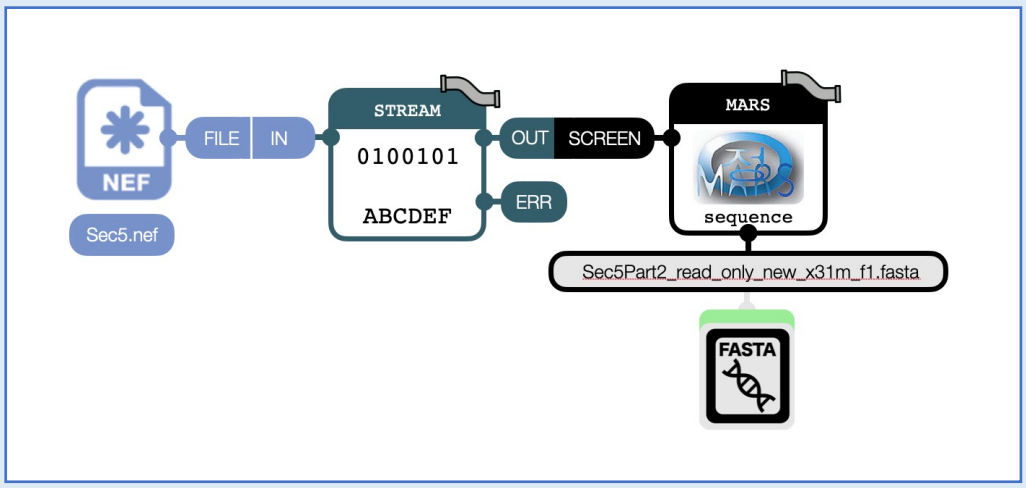
To do this type the following in the terminal

```
nef stream ./tests/test_data/sec5.nef
```

Note that our sec5 data is stored in **tests/test_data/sec5.nef** you will need to correct the path to your **sec5.nef** file from the tutorial data. You will see the file contents go by.

```
data_Sec5Part2_read_only_new_x31m_f1


   save_nef_nmr_meta_data

      _nef_nmr_meta_data.sf_category      nef_nmr_meta_data
      _nef_nmr_meta_data.sf_framecode     nef_nmr_meta_data
      _nef_nmr_meta_data.format_name      nmr_exchange_format
      _nef_nmr_meta_data.format_version   1.1                  …
      _nef_nmr_meta_data.program_name     AnalysisAssign
```

Now we will not connect the stream to the screen but to a tool that will write out a sequence for MARS. Conceptually we have



1B Outputting the stream to a MARS sequence file

To do this type the following in the terminal

```
nef stream ./tests/test_data/sec5.nef | nef mars export sequence
```

Note you will see text like the following, this is a bug from one of libraries NEF-Pipelines uses, it will disappear in a future version and and can be ignored!

```
2022-11-29 23:37:13,469,469 WARNING  [parser.py:161] Loop with no data on line: 2993
2022-11-29 23:37:13,469,469 WARNING  [parser.py:161] Loop with no data on line: 3020
```

If you type **`ls`** in your terminal you will now find a new file called Sec5Part2_read_only_new_x31m_f1.fasta this is a rather ugly name that comes from the name of the contents of the NEF file, we will make it a nicer name a bit later.

If you open the file Sec5Part2_read_only_new_x31m_f1.fasta with a text editor you will find it contains the text

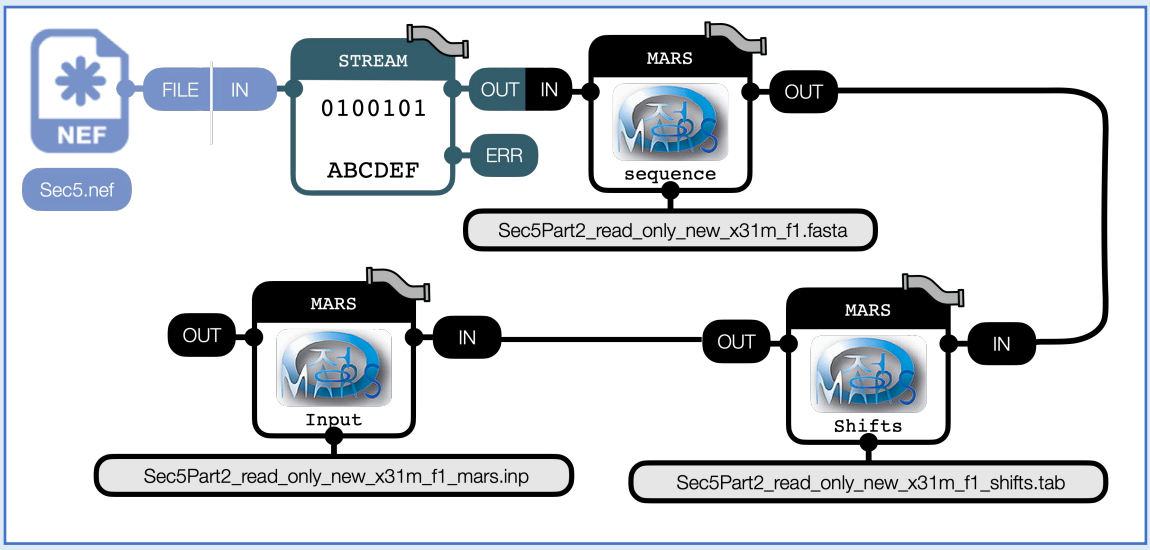
```
>CHAIN: A | START RESIDUE: 3
HMRQPPLVTG ISPNEGIPWT KVTIRGENLG TGPTDLIGLT ICGHNCLLTA EWMSASKIVC RVGQAKNDKG DIIVTTKSGG
```

This is the sequence of the protein sec5 in the format used by MARS (fasta). This is quite different to how the sequence is stored in the NEF file. If there was more than one chain in the NEF file NEF-Pipelines would ask you select which chain to use as MARS can only run on a single chain…

Note how you no longer see lots of NEF text written to screen. The pipeline assumes you don't want lots of screen output and so absorbs it at the end of the pipeline to make your experience easier. There is a way to display it again using the **cat** command on OSX and Linux and type / Get-Content under Windows PowerShell but that is beyond the scope of this tutorial.

Writing the rest of the files

MARS takes several files which it needs to run we will now add the extra tools that write these files (shifts and most of a project input file)



1c

Outputting the stream to the rest of the MARS files

To do this, type the following in the terminal. Please write all the text on one line (here it is spread out over more than one line to make it fit on the page!). Note that you can use the up and down arrows to show previous commands you ran which can be edited and rerun by typing return again

```
nef stream ./tests/test_data/sec5.nef | nef mars export sequence
| nef mars export shifts| nef mars export input
```

If you type **`ls Sec5*`** in your terminal you will now find you have quite a lot more files

```
Sec5Part2_read_only_new_x31m_f1.fasta
Sec5Part2_read_only_new_x31m_f1_mars.inp
Sec5Part2_read_only_new_x31m_f1_mars_ass.tab
Sec5Part2_read_only_new_x31m_f1_mars_pred.tab
Sec5Part2_read_only_new_x31m_f1_shifts.tab
```

...

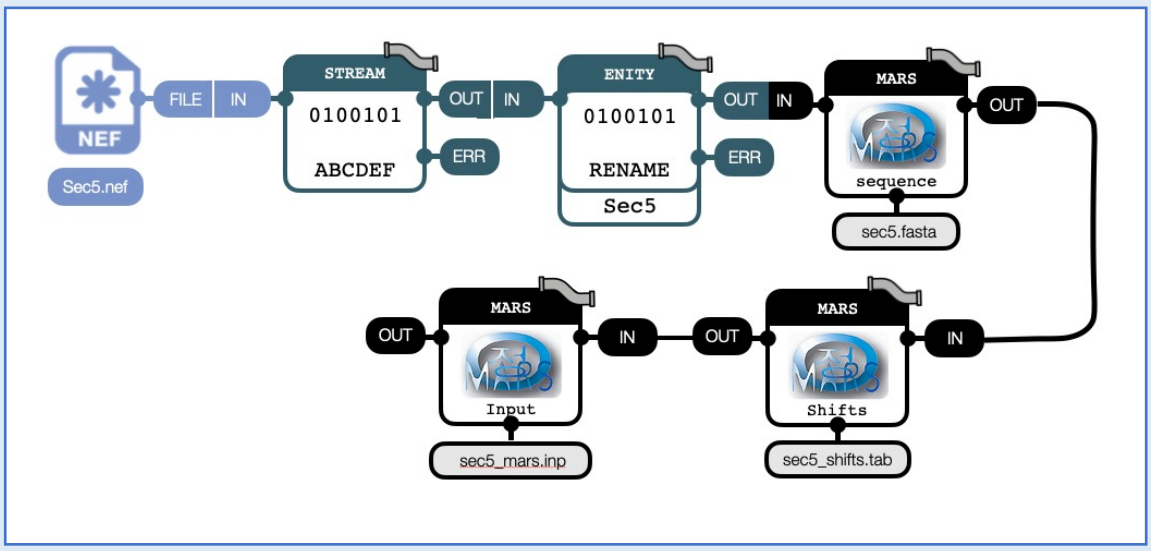
These are the rest of the files you need to run MARS: a sequence file, a list of chemical shifts and an input file. If you investigate the input tool you will find options to write in the input file if it is data for a folded protein and if it is deuterated protein.

The files **…_ass.tab** and …**_pred.tab** are for telling MARS about residue types and strings of residues that have already been combined together by previous assignments. These are currently empty as the MARS tool is not sophisticated enough to write them yet, you would have to write them yourself!

Please also note that you would need to obtain a psipred formatted prediction of the protein's secondary structure to run MARS, though we may be able to derive this from alphafold more quickly in the future….
Psipred can be found at http://bioinf.cs.ucl.ac.uk/psipred/.

Making it less ugly

The names of the files the MARS tools output are taken from the name of NEF file Entry. If we change this at the start of the stream it will give better file names but won't affect the original file on disk (unless you write it first)



1D Renaming the NEF file Entry

If we rename the entry inside the NEF file (the structure that contains all the data) we will change the names of all the output files when they are exported as their names are based on the entry name…

```
nef stream ./tests/test_data/sec5.nef | nef entry rename sec5
| nef mars export sequence| nef mars export shifts
| nef mars export input
```

If you type **`ls sec5*`** in your terminal you will now find you have files with much nicer names

```
sec5.fasta
sec5_mars.inp
sec5_mars_ass.tab
sec5_mars_pred.tab
sec5_shifts.tab
```