Connjur

An Open Source Solution for NMR Software Integration

2012 CT NMR Workshop Day 1
Michael R. Gryk
University of Connecticut Health Center
Farmington, Connecticut 06030-3305
June 18, 2012
Acknowledgements

Matthew Fenwick
Jeffrey C. Hoch
Mark W. Maciejewski
Mehdi Mobli
RJ Nowling
Colbert Sesanker
Jay Vyas

Heidi J.C. Ellis
Gerard Weatherby
Rensselaer at Hartford:
Timothy O. Martyn
NIDDK: Frank Delaglio
Rowland Institute: Alan Stern
NMR Computational Pipeline

1. Spectrometer Acquisition
2. Spectral Reconstruction
3. Spectral Analysis
4. Biophysical Characterization

< 5Å
CONNJUR Data Flow

- Sample Preparation
- Acquisition
- Spectral Reconstruction
- Analysis
- Characterization
CONNJUR Software

• Connjur will provide the interface between existing software tools and a relational database

• Existing software tools have extensive history and are time-tested

• User interface is designed to aid in navigation and management
3-Tier Architecture

DATA ACCESS LAYER
- Data Management
- Spectral Reconstruction
- Spectral Analysis
- Biophysical Characterization
- Project Management

BUSINESS LOGIC

USER INTERFACE

CONNJUR DB

External Data Access
- PDB
- Swis-Prot
- BMRB

f1
f2
f3
f4
Free and Open Source

- **Price**: Free to download and install
- **Extensibility**: No restrictions or royalties to modify or resell
- **Transparency**: Source code readily inspected and verified
- **Active Development**: the scientific community is encouraged to aid in the development of CONNJUR
- **Perpetual**: If we are unable to continue the project, anyone is free to continue developing
Free/Open Source Interoperability

• **Operating System**: LINUX
• **Framework Language**: JAVA
• **Relational Database**: MySQL
How much of the above information is NMR related?

How much is related to nmrPipe or the computer we are using?

bash vs. csh – the shell wars

which nmrPipe tool? nmrPipe, var2pipe, xyz2pipe, etc.

watch for trailing spaces!

Syntax?

filenames and filesystems

# = comment

#! /bin/csh

# My Processing Script

nmrPipe -in mydata.pipe

| nmrPipe -fn SOL -mode 1 -f1 16 -fs 1 -poly |
| nmrPipe -fn CBF -last 12 |
| nmrPipe -fn GMB -lb 7 -gb 0.1 -size 512 -c 0.5 |
| nmrPipe -fn ZF -size 2048 |
| nmrPipe -fn FT -verb |
| nmrPipe -fn PS -p0 68.6 -p1 -34.8 -di |
| nmrPipe -fn EXT -left -sw -verb |
| nmrPipe -fn TP |
| nmrPipe -fn LE -fb -ord 30 -x1 2 -xn 128 -pred 64 -fix -fixMode 1 -after |
| nmrPipe -fn SP -off 0.39 -end 0.98 -pow 2 -size 192 -c 0.5 |
| nmrPipe -fn ZF -size 256 |
| nmrPipe -fn FT -verb |
| nmrPipe -fn PS -p0 -0.0 -p1 20.0 -di |

-out mydata.ft2 -ov
# Functionality and Function Order!

```
#!/bin/csh

# My Processing Script
nmrPipe -in mydata.pipe
| nmrPipe -fn SOL -mode 1 -fl 16 -fs 1 -poly
| nmrPipe -fn CBF -last 12
| nmrPipe -fn GMB -lb -7 -gb 0.1 -size 512 -c 0.5
| nmrPipe -fn ZF -size 2048
| nmrPipe -fn FT -verb
| nmrPipe -fn PS -p0 68.6 -p1 -34.8 -di
| nmrPipe -fn EXT -left -sw -verb
| nmrPipe -fn TP
| nmrPipe -fn LP -fb -ord 30 -x1 2 -xn 128 -pred 64 -fix -fixMode 1 -after
| nmrPipe -fn SP -off 0.39 -end 0.98 -pow 2 -size 192 -c 0.5
| nmrPipe -fn ZF -size 256
| nmrPipe -fn FT -verb
| nmrPipe -fn PS -p0 -9.0 -p1 20.0 -di
-out mydata.ft2 -ov
```

What procedures do we use to massage our data?

What procedures do we use to transform our data?