BMRB
BioMagResBank

BMRB is a member of the wwPDB

WORLDWIDE PDB PROTEIN DATA BANK

BMRB collaborates with

PDBj Protein Data Bank Japan

CCPN

RCSB PDB PROTEIN DATA BANK

PDBe PROTEIN DATA BANK EUROPE

CONDOR high throughput computing

National Magnetic Resonance Facility At Madison

Using Condor behind the scenes to provide a public CS-Rosetta server at the BioMagResBank

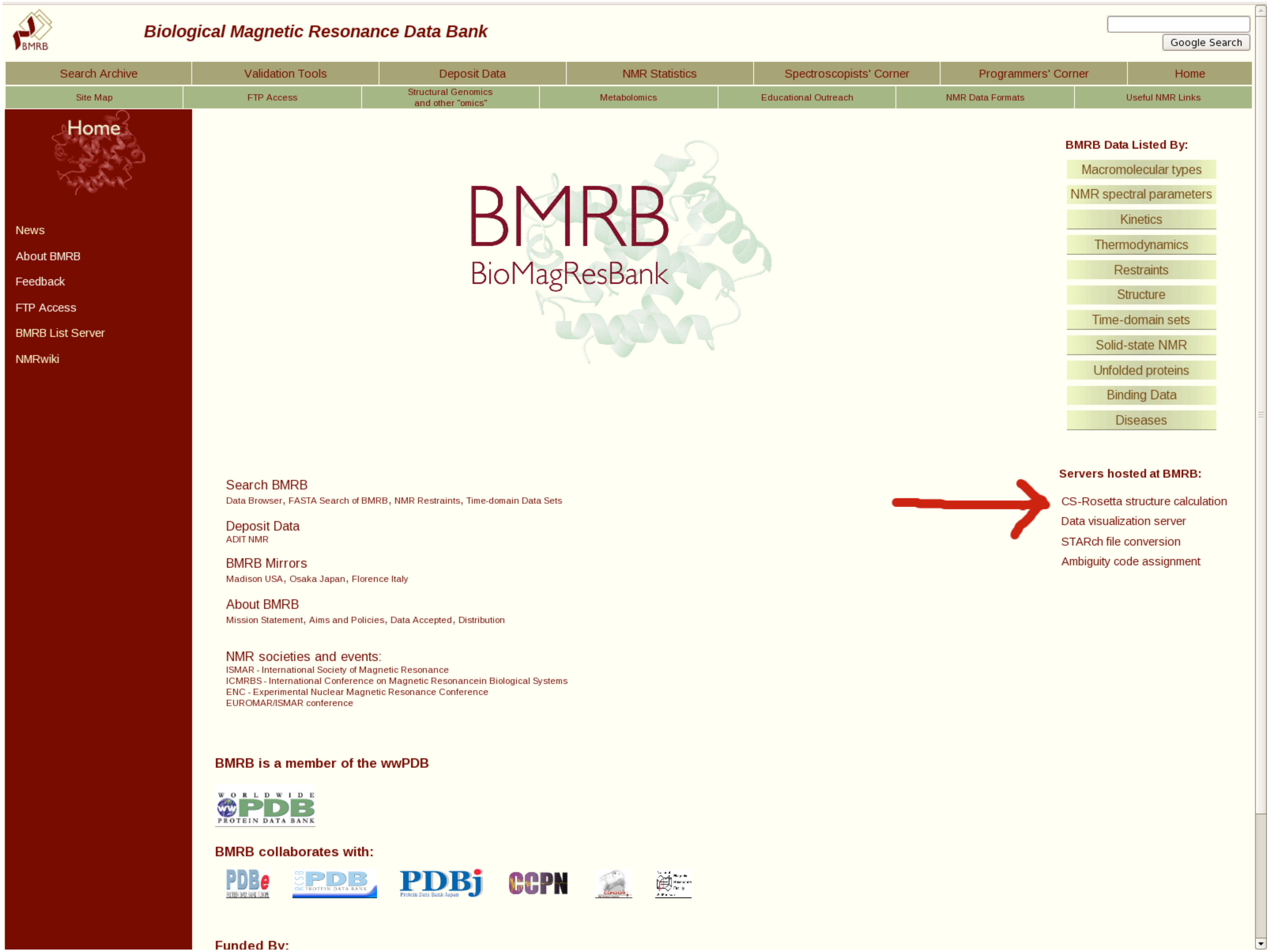
Jonathan Wedell

Funded By:

NATIONAL LIBRARY OF MEDICINE

What is the BMRB?

- Biological Magnetic Resonance Data Bank
  - NMR (nuclear magnetic resonance) data
    - Known as “chemical shifts”
  - Validation tools
  - Metabolomics
  - Visualizations
    - Kent Wenger
  - Lots of other stuff
  - CS-Rosetta Server



Biological Magnetic Resonance Data Bank

Google Search

Search Archive | Validation Tools | Deposit Data | NMR Statistics | Spectroscopists' Corner | Programmers' Corner | Home

Site Map | FTP Access | Structural Genomics and other "omics" | Metabolomics | Educational Outreach | NMR Data Formats | Useful NMR Links

Home

- News
- About BMRB
- Feedback
- FTP Access
- BMRB List Server
- NMRwiki

BMRB
BioMagResBank

BMRB Data Listed By:

- Macromolecular types
- NMR spectral parameters
- Kinetics
- Thermodynamics
- Restraints
- Structure
- Time-domain sets
- Solid-state NMR
- Unfolded proteins
- Binding Data
- Diseases

Search BMRB
Data Browser, FASTA Search of BMRB, NMR Restraints, Time-domain Data Sets

Deposit Data
ADIT NMR

BMRB Mirrors
Madison USA, Osaka Japan, Florence Italy

About BMRB
Mission Statement, Aims and Policies, Data Accepted, Distribution

NMR societies and events:
ISMAR - International Society of Magnetic Resonance
ICMRBS - International Conference on Magnetic Resonancein Biological Systems
ENC - Experimental Nuclear Magnetic Resonance Conference
EUROMAR/ISMAR conference

Servers hosted at BMRB:

- CS-Rosetta structure calculation
- Data visualization server
- STARch file conversion
- Ambiguity code assignment

BMRB is a member of the wwPDB

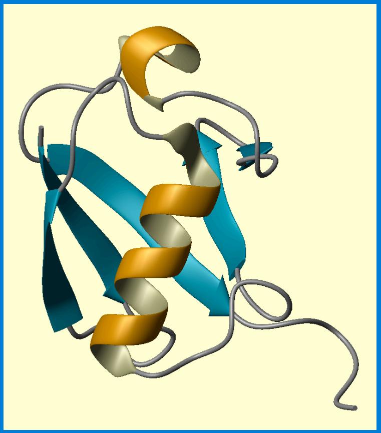
BMRB collaborates with:

Funded By:

BMRB
BioMagResBank

What is CS-Rosetta?

- It is an existing software package
  - "System for chemical shifts based protein structure prediction using ROSETTA."
- Input (chemical shifts)
  - Multiple formats allowed
- Output
  - Structure file
    - 3d model of a protein
  - Structure energy level and RMSD
    - RMSD = root mean square deviation
      - Measurement of how similar a given structure is to another

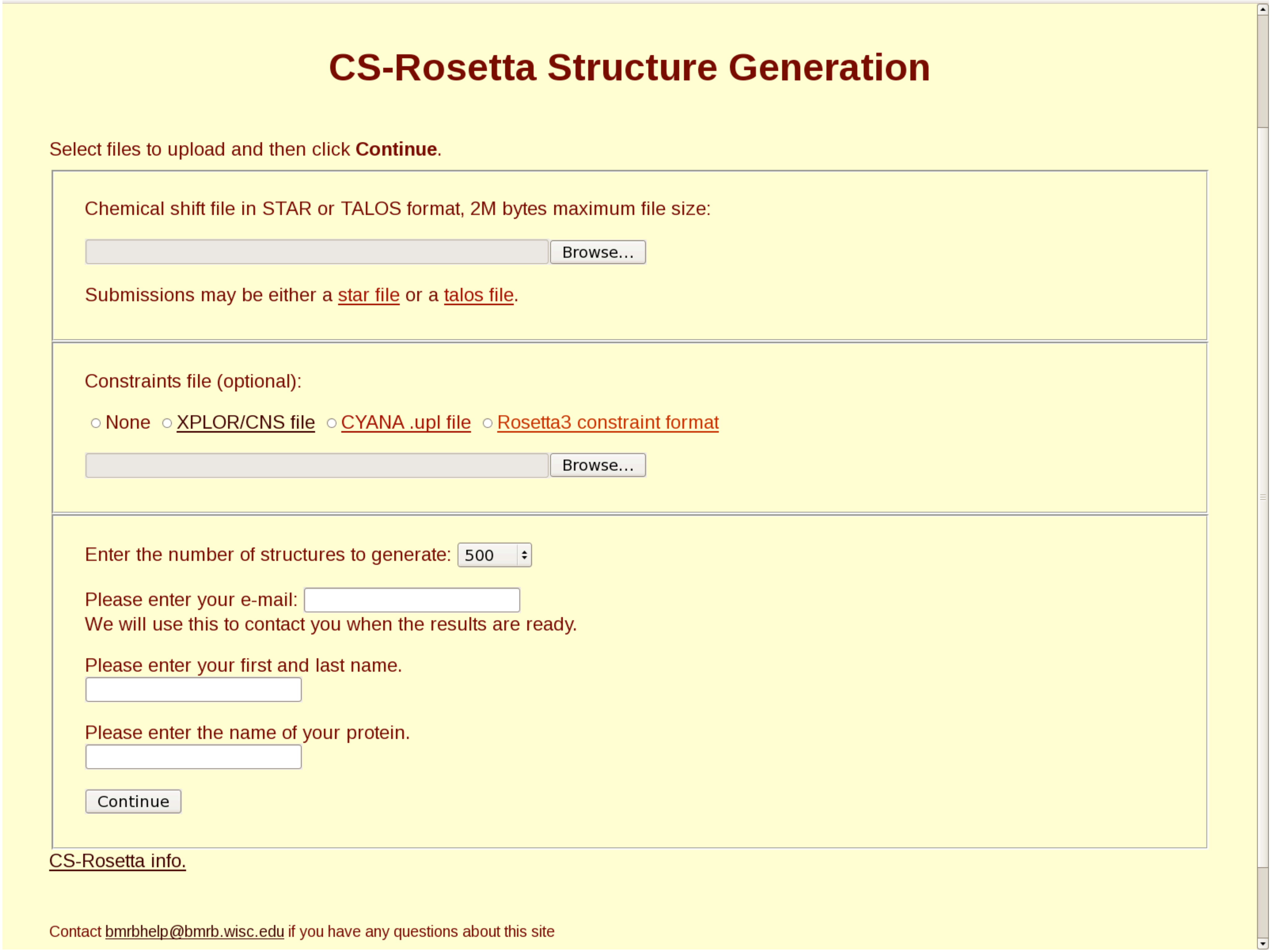


How does it work?

- Searches a database of known chemical shifts from existing proteins
  - Finds 3-segment and 9-segment matches
- Generates structures from matches
- Performs simulated annealing
  - Simulated heating and cooling of protein – results in a protein shape with the lowest net energy
- Annealed structures are scored and compared

Submission

- PHP interface
  - Chemical Shift File
  - Constraint File (optional)
    - Additional data that helps refine the generated structures
  - User name
  - Protein Name
  - Number of structures
  - E-mail address
  - *Bribe amount*



CS-Rosetta Structure Generation

Select files to upload and then click **Continue**.

Chemical shift file in STAR or TALOS format, 2M bytes maximum file size:

Submissions may be either a star file or a talos file.

Constraints file (optional):

○ None ○ XPLOR/CNS file ○ CYANA .upl file ○ Rosetta3 constraint format

Enter the number of structures to generate: 500

Please enter your e-mail:
We will use this to contact you when the results are ready.

Please enter your first and last name.

Please enter the name of your protein.

CS-Rosetta info.

Contact bmrbhelp@bmrb.wisc.edu if you have any questions about this site

Result Access

- PHP interface
  - Accessible using randomly assigned key
  - Graph of structures
    - Relative energy
    - RMSD
  - Raw data download

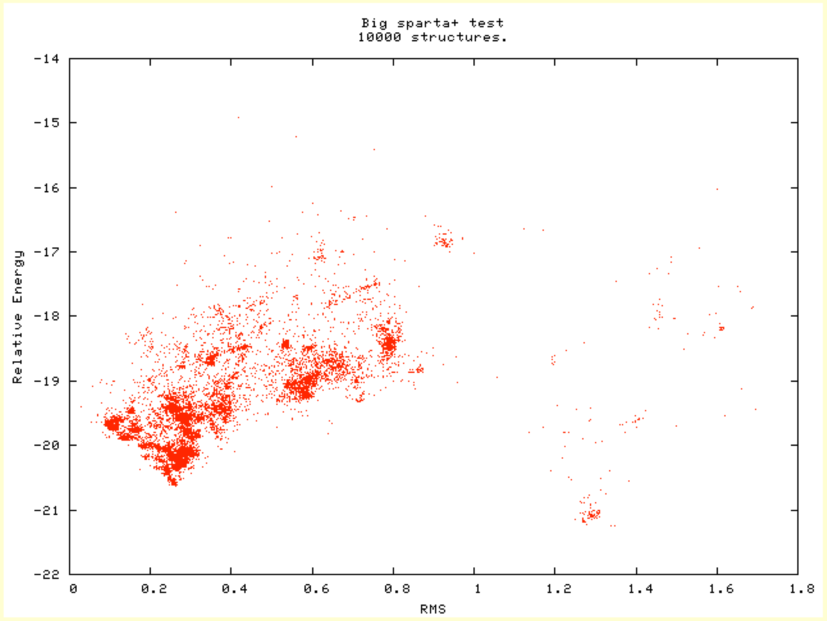
Biological Magnetic Resonance Data Bank

A Repository for Data from NMR Spectroscopy on Proteins, Peptides, Nucleic Acids, and other Biomolecules

Google Search

CS-Rosetta Results Access

For entry 'Big sparta+ test'.

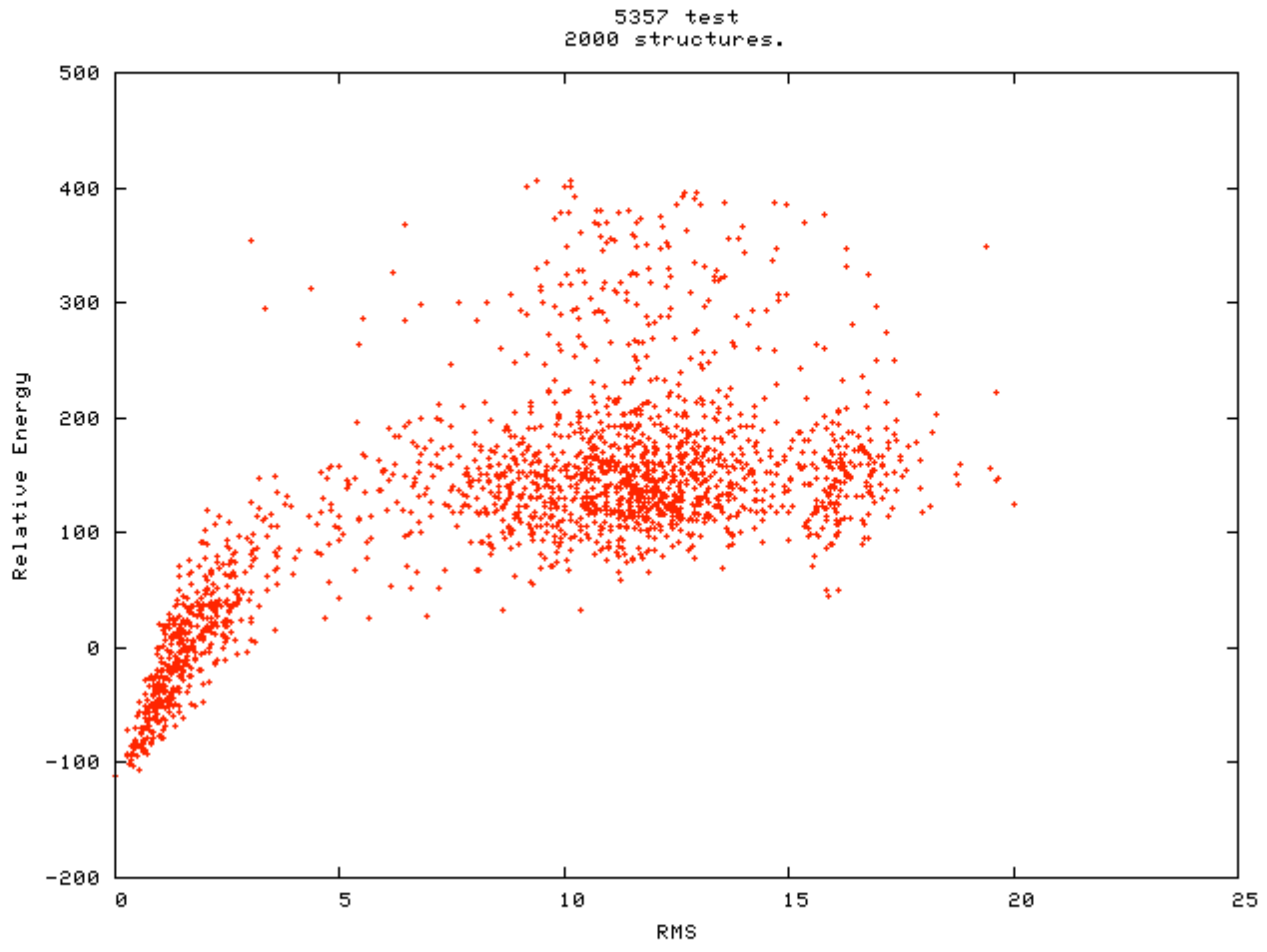


Clustering Image | Download

Please see the CS-Rosetta homepage for an explanation of the output files.

Contact bmrbhelp@bmrb.wisc.edu if you have any questions about this site

Copyright © The Board of Regents of the University of Wisconsin System.
Last Modified: Wednesday, 09-Feb-2011 20:04:07 CST


5357 test
2000 structures.
Relative Energy
RMS
500
400
300
200
100
0
-100
-200
0
5
10
15
20
25

Behind the scenes

- PHP web interface
  - Writes files to submission folder
- PERL daemon
  - Scans for new entries at configurable interval
    - Web server uses SIGHUP when entries are deposited
  - Performs initial verification of data
  - Converts submitted data to appropriate format
  - Creates DAG file for entry
  - Submits DAG file to Condor and monitors progress
    - Sends e-mails to author to inform them of progress

Condor Jobs

- Entry processing involves 4 stages
  - Preparation (1 job)
  - *Ab initio* (1 job per 25 structures)
    - 25 structures/job is arbitrary but selected for reasonable balance between job times and number of jobs
      - Many small jobs
        - Lots of network overhead
      - Few large jobs
        - Fail to utilize all available machines
        - More to re-do in case of job failure or eviction
  - Re-scoring (1 job)
  - Cleanup (1 job)

BMRB
BioMagResBank

Preparation

- Vanilla universe job on local cluster
- Large database needed (>9 gigabytes)
- Searches for fragment matches
- Usually several hours
- Very difficult to parallelize
- Failures occur here due to lack of matching fragments

Preparation Submit File

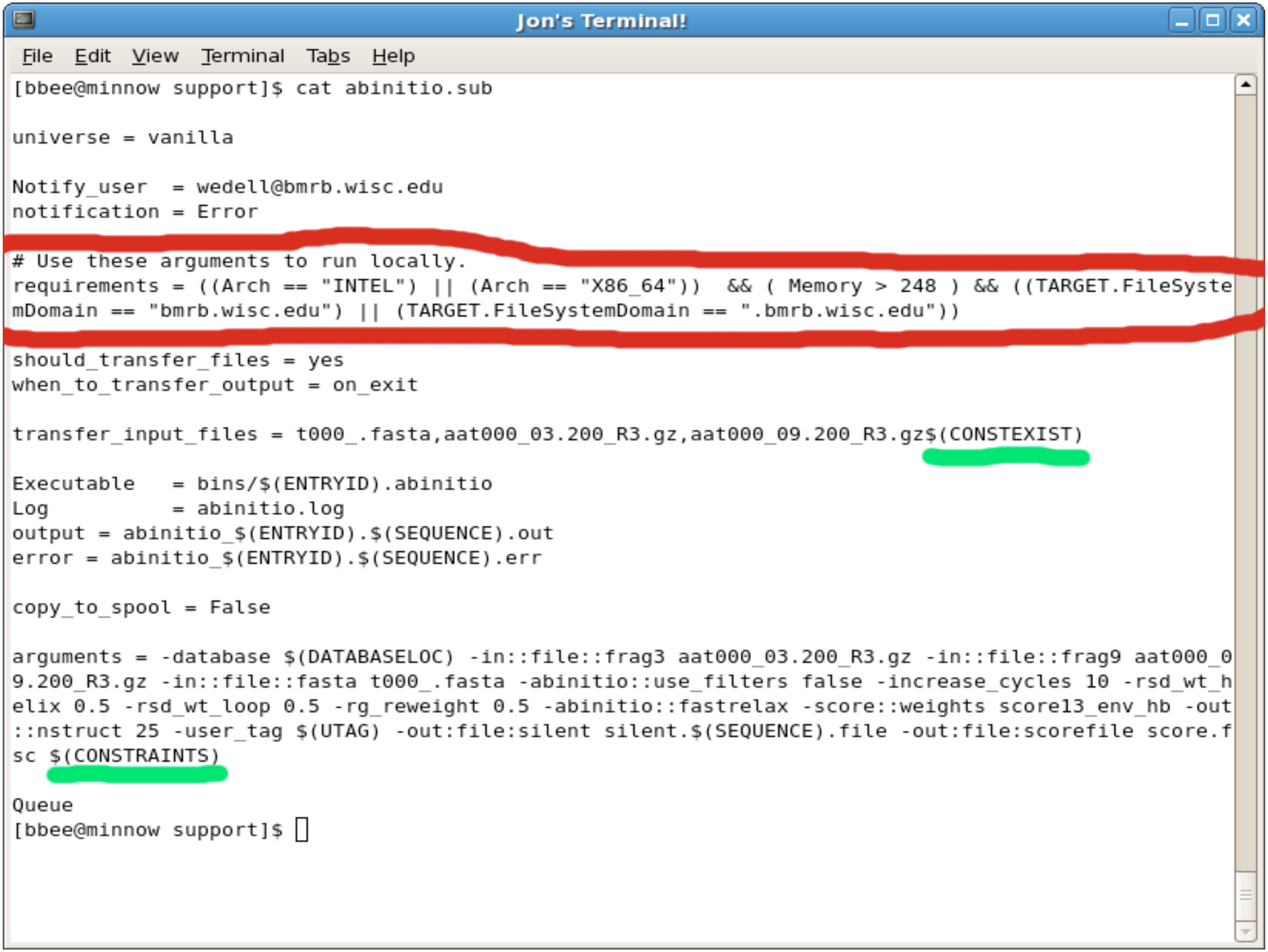
```
universe = vanilla

Executable = bins/$(ENTRYID).prepare

log = prepare.log
output = prepare_$(ENTRYID).out
error = prepare_$(ENTRYID).err

queue
```

Ab initio
(From the beginning)

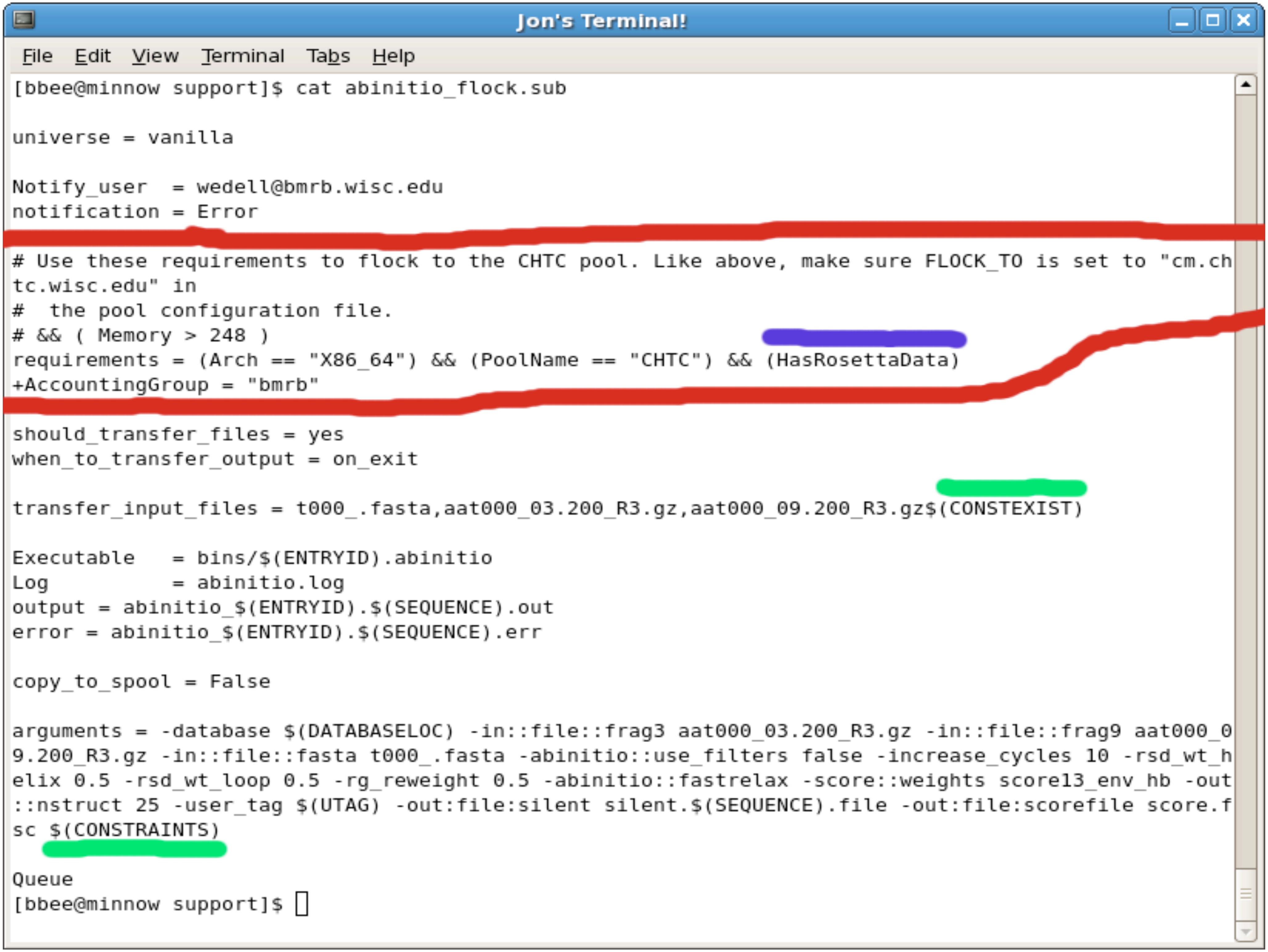
- Structures generated and annealed
- 25 structures generated in each Condor job
  - 5 minutes – 16 hours per job
- Requires 50 megabyte database
  - Store on machines or send with each job?
    - Store on machines
- Vanilla universe
- Flock to CHTC or run locally?
  - CHTC = 1823 x86 cores
  - Local = 32 x86 cores

Ab initio

- Run at CHTC
    - Able to fall back to running locally



```
[bbee@minnow support]$ cat abinitio.sub

universe = vanilla

Notify_user  = wedell@bmrb.wisc.edu
notification = Error

# Use these arguments to run locally.
requirements = ((Arch == "INTEL") || (Arch == "X86_64"))  && ( Memory > 248 ) && ((TARGET.FileSyste
mDomain == "bmrb.wisc.edu") || (TARGET.FileSystemDomain == ".bmrb.wisc.edu"))

should_transfer_files = yes
when_to_transfer_output = on_exit

transfer_input_files = t000_.fasta,aat000_03.200_R3.gz,aat000_09.200_R3.gz$(CONSTEXIST)

Executable   = bins/$(ENTRYID).abinitio
Log          = abinitio.log
output = abinitio_$(ENTRYID).$(SEQUENCE).out
error = abinitio_$(ENTRYID).$(SEQUENCE).err

copy_to_spool = False

arguments = -database $(DATABASELOC) -in::file::frag3 aat000_03.200_R3.gz -in::file::frag9 aat000_0
9.200_R3.gz -in::file::fasta t000_.fasta -abinitio::use_filters false -increase_cycles 10 -rsd_wt_h
elix 0.5 -rsd_wt_loop 0.5 -rg_reweight 0.5 -abinitio::fastrelax -score::weights score13_env_hb -out
::nstruct 25 -user_tag $(UTAG) -out:file:silent silent.$(SEQUENCE).file -out:file:scorefile score.f
sc $(CONSTRAINTS)

Queue
[bbee@minnow support]$
```



Jon's Terminal!

File Edit View Terminal Tabs Help

```
[bbee@minnow support]$ cat abinitio_flock.sub

universe = vanilla

Notify_user  = wedell@bmrb.wisc.edu
notification = Error

# Use these requirements to flock to the CHTC pool. Like above, make sure FLOCK_TO is set to "cm.ch
tc.wisc.edu" in
#  the pool configuration file.
# && ( Memory > 248 )
requirements = (Arch == "X86_64") && (PoolName == "CHTC") && (HasRosettaData)
+AccountingGroup = "bmrb"

should_transfer_files = yes
when_to_transfer_output = on_exit

transfer_input_files = t000_.fasta,aat000_03.200_R3.gz,aat000_09.200_R3.gz$(CONSTEXIST)

Executable   = bins/$(ENTRYID).abinitio
Log          = abinitio.log
output = abinitio_$(ENTRYID).$(SEQUENCE).out
error = abinitio_$(ENTRYID).$(SEQUENCE).err

copy_to_spool = False

arguments = -database $(DATABASELOC) -in::file::frag3 aat000_03.200_R3.gz -in::file::frag9 aat000_0
9.200_R3.gz -in::file::fasta t000_.fasta -abinitio::use_filters false -increase_cycles 10 -rsd_wt_h
elix 0.5 -rsd_wt_loop 0.5 -rg_reweight 0.5 -abinitio::fastrelax -score::weights score13_env_hb -out
::nstruct 25 -user_tag $(UTAG) -out:file:silent silent.$(SEQUENCE).file -out:file:scorefile score.f
sc $(CONSTRAINTS)

Queue
[bbee@minnow support]$ 
```

BMRB
BioMagResBank

HasRosettaData requirement

- Job requirement for *ab initio* jobs
- Only machines with our 50 mb database should run *ab initio* jobs.
  - Hawkeye script monitors which machines have the necessary data
  - Those machines match the "HasRosettaData" requirement

Hawkeye Script

```
#! /bin/sh

dir="/data2/bmrb"

if [ ! -d $dir ]; then
    echo "HasRosettaData = false"
    exit
else
    cd $dir
    ./check_database.pl database_reference database
    if [ $? != 0 ]; then
        echo "HasRosettaData = false"
    else
        echo "HasRosettaData = true"
    fi
fi
```

Re-scoring

- Energy score of generated entries is easily calculated but not very accurate
  - Re-scoring provides a more accurate energy score for entries
    - Goal is a low-energy structure
- Runs locally
  - 4 hours – 3 days
- Severe bottleneck
  - Possible to parallelize
    - Future goal
- Requires locally installed software and libraries

Re-scoring submit file

```
universe = local

Notify_user  = wedell@bmrb.wisc.edu
notification = Error

# Use these arguments to run locally.
requirements = ((Arch == "INTEL") || (Arch == "X86_64"))  && ( Memory > 248 )

Executable   = bins/$(ENTRYID).rescore
Log          = rescore.log
output = rescore_$(ENTRYID).out
error = rescore_$(ENTRYID).err
copy_to_spool = False


arguments = silent_file inCS.tab
queue
```

BMRB
BioMagResBank

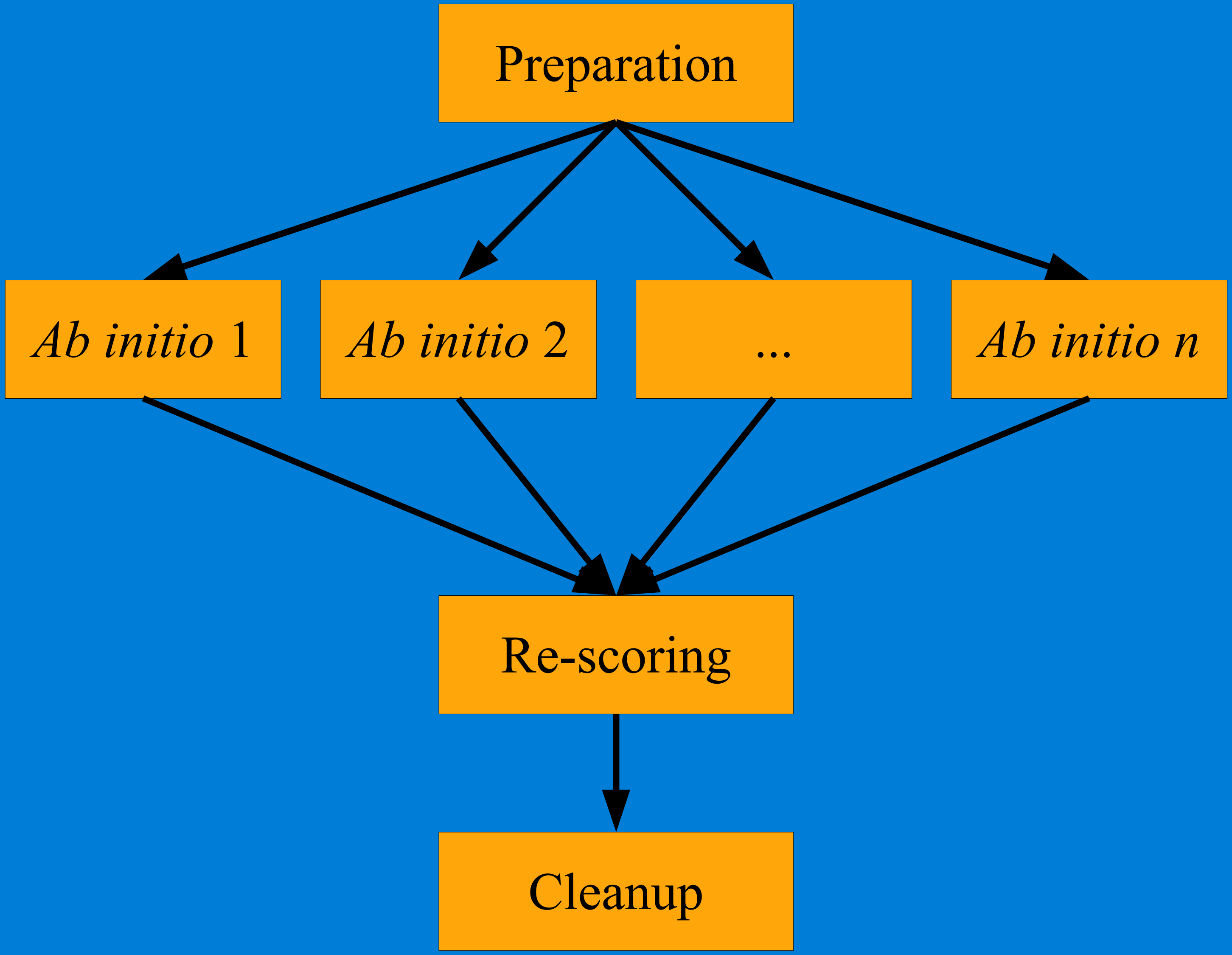
Cleanup

- Bash script
- Cleans up run directory
  - Deletes unneccesary files
  - Tarballs Condor run files
  - Zips data for user download
- Generates graph using gnuplot

Cleanup submit file

```
universe = vanilla

Notify_user  = wedell@bmrb.wisc.edu
notification = Error

Executable   = bins/$(ENTRYID).cleanup

Log          = clean.log
output = clean_$(ENTRYID).out
error = clean_$(ENTRYID).err
copy_to_spool = False

arguments = $(ENTRYID)

queue
```

DAG Layout


Preparation
Ab initio 1
Ab initio 2
...
Ab initio n
Re-scoring
Cleanup

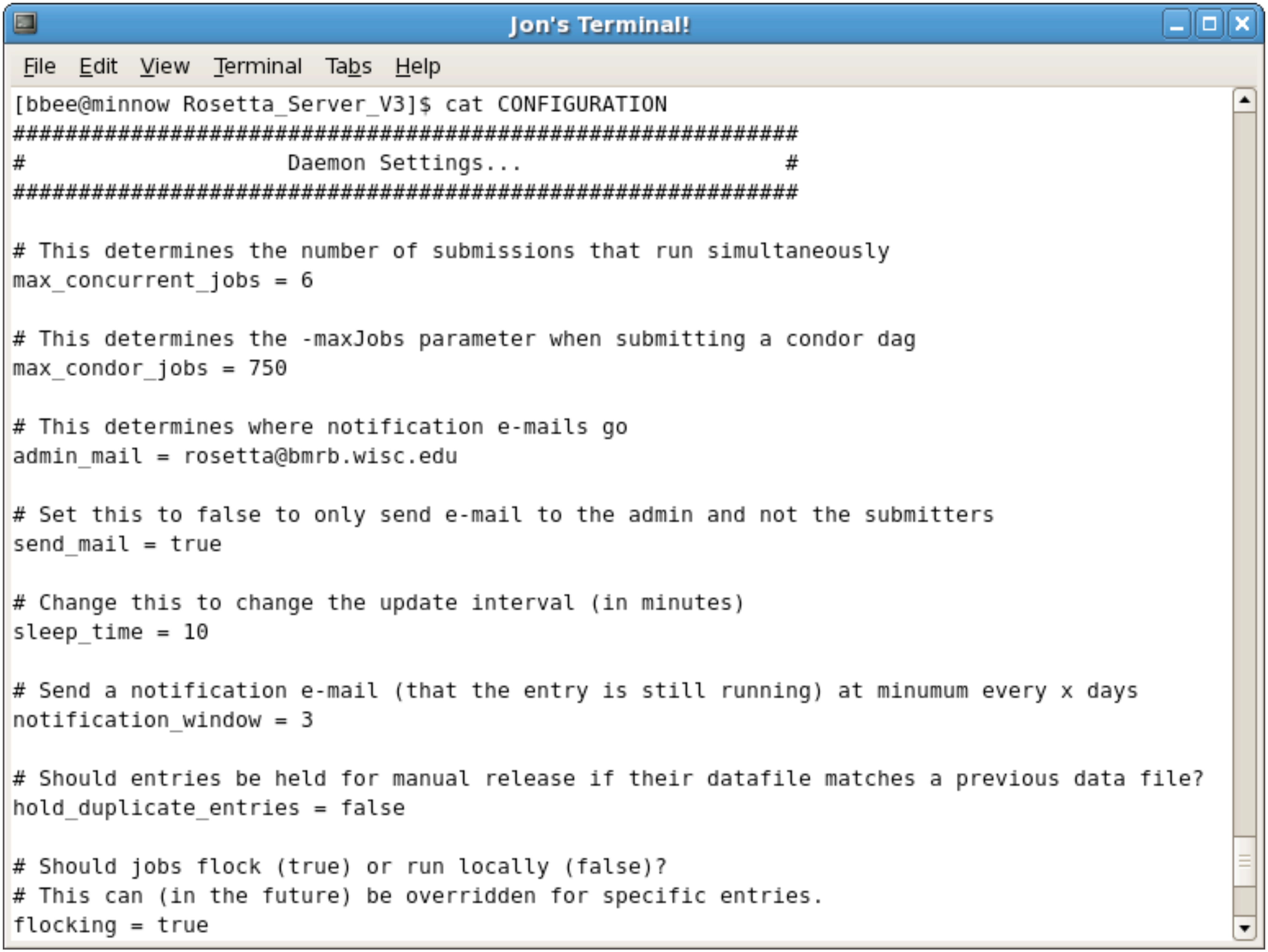
The DAG

```
JOB 7yh2z9.prep prepare.sub DIR .
VARS 7yh2z9.prep ENTRYID="7yh2z9"
SCRIPT POST 7yh2z9.prep /minnow/bbee/Rosetta_Server_V3/scripts/support/prep_post 7yh2z9

JOB 7yh2z9.abinitio.1 abinitio_flock.sub DIR .
VARS 7yh2z9.abinitio.1 ENTRYID="7yh2z9" SEQUENCE="1" UTAG="j0001" CONSTRAINTS=""
DATABASELOC="/data2/bmrb/database/" CONSTEXIST=""
RETRY 7yh2z9.abinitio.1 1
PARENT 7yh2z9.prep CHILD 7yh2z9.abinitio.1

...

JOB 7yh2z9.abinitio.400 abinitio_flock.sub DIR .
VARS 7yh2z9.abinitio.400 ENTRYID="7yh2z9" SEQUENCE="400" UTAG="j0400"
CONSTRAINTS="" DATABASELOC="/data2/bmrb/database/" CONSTEXIST=""
RETRY 7yh2z9.abinitio.400 1
PARENT 7yh2z9.prep CHILD 7yh2z9.abinitio.400

JOB 7yh2z9.rescore rescore.sub DIR .
VARS 7yh2z9.rescore ENTRYID="7yh2z9"
PARENT 7yh2z9.abinitio.1 CHILD 7yh2z9.rescore
...
PARENT 7yh2z9.abinitio.400 CHILD 7yh2z9.rescore

JOB 7yh2z9.cleanup cleanup.sub DIR .
VARS 7yh2z9.cleanup ENTRYID="7yh2z9"
PARENT 7yh2z9.rescore CHILD 7yh2z9.cleanup
```

BMRB
BioMagResBank

The DAG

- Constraint data only present in some entries
    - Use DAG variables to add it in
- Use DAG variables to run locally or flock
- DAG ensures correct job order



```
[bbee@minnow Rosetta_Server_V3]$ cat CONFIGURATION
############################################################
#                    Daemon Settings...                    #
############################################################

# This determines the number of submissions that run simultaneously
max_concurrent_jobs = 6

# This determines the -maxJobs parameter when submitting a condor dag
max_condor_jobs = 750

# This determines where notification e-mails go
admin_mail = rosetta@bmrb.wisc.edu

# Set this to false to only send e-mail to the admin and not the submitters
send_mail = true

# Change this to change the update interval (in minutes)
sleep_time = 10

# Send a notification e-mail (that the entry is still running) at minumum every x days
notification_window = 3

# Should entries be held for manual release if their datafile matches a previous data file?
hold_duplicate_entries = false

# Should jobs flock (true) or run locally (false)?
# This can (in the future) be overridden for specific entries.
flocking = true
```

Some statistics

- 262 submissions on current version of server
  - 31 users
  - Average of ~6500 structures per submission
- ~1,700,000 structures
  - Slightly fewer structures in actuality due to fragment matching failures
  - 370,000 CPU hours / 42 CPU years on CHTC machines
    - 4.6 structures per hour / 13 minutes per structure

Plans for the future

- Add option to use more refinement data such as RDC's
- Parallelize re-scoring job
    - Included in upcoming version of CS-Rosetta

Thanks to:

- Eldon Ulrich
- Dimitri Maziuk
- Kent Wenger
- Condor Team

BMRB
BioMagResBank

Questions?