Databases in e-HTPX

Database Requirements for CCP4

17th October 2005

What is e-HTPX?

- “An e-Science resource for high throughput protein crystallography”
- Start at crystallisation, end at deposition
- Includes a lot of “project management” since operations are being performed at a number of remote sites
- Should be able to talk to PIMS, beamline, CCP4 &c.

Relevant Areas in e-HTPX

- Data collection & processing – XIA-DPA
- Structure solution *via* MR – BMP
- Structure solution *via* (M/S)AD – XIA-HA
- Deposition – Autodep

Components with DB Needs

- Data Collection (e.g. DNA/ISPyB)
- Automated data processing
  - Data exchange, internal data management
- BMP
  - External (EBI) databases, internal job management
- Experimental Phasing
  - Finding input, storing results

Particular Examples 1

- During crystal characterisation we decide that the crystal is *probably* tetragonal
- Collect 75 degrees of data & process when you get home
- Suddenly discover that the crystal is orthorhombic, and anyway the solvent content would have been 11%
- Kick yourself, apply for more beam time

Particular Examples 2

- During crystal characterisation we decide that the crystal is *probably* tetragonal
- However querying the database says that that would result in a solvent content of 11% - jolly unlikely
- Collect & process a little data
- Decide point group & store away some where – then compute strategy & collect new set

Particular Challenges

- People: they never fill things in!
- Software: needs to be able to find things out all by itself – so in the previous example program X needs to be able to find out about the molecule
- Consistency& robustness: we may find out later on that in fact we’ve only got about half the molecule – we need to be able handle this

Another Example

- I have just collected a bunch of data sets and I wish to process them automatically
- Three wavelengths, with a high resolution remote sweep which overloaded the detector at lower resolutions
- Want to be able to combine the two remote sweeps into a single data set, then scale the other two against this set

Data Processing Data

- Locations of files
- Derived “facts” for future reference
- Useful feedback to data collection
- Hooks to get downstream processes going
- Useful statistics for “Table 1” of your publication

MR Example

- Procedure
  - Generate a large number of search models
  - Starting with the best try each and then stop
- Record results – both for user interaction and future reference (e.g. learning what makes a “good model” or likelihood of success)
- Could allow jobs to be tracked more easily and also rerun manually if desired

MR Data

- Pointers to PDB files
- Sequences & identities
- Progress & job tracking

Yet Another Example

- I have a 3 wavelength data set, which is phasing badly in ${automated pipeline}
- The “system” says there may be radiation damage, so we need to be able to find out which set was collected last and try phasing from just that

Organization

- What makes a project?
  - Solve BRT1?
  - Collect 3 wavelength MAD set?
  - Process peak?
  - Figure out scaling parameters for peak?
- Probably all of the above…

Project brt1.peak.scale.refine_parameters?

So What?

- So we need to be able to express and record the *relationship* between different data sets – once these are properly expressed we can proceed
- This may require some kind of “import” mechanism where ${user} has an opportunity to provide a description of the data and the f’, f’’, correct beam & so on

What Else?

- Critical that things “discovered” at one stage are not lost thereafter
  - e.g. data processing step asserts that the space group is probably P43212 or P41212, so don’t bother with P4122 at the phasing stage
- Critical also that later “discoveries” can be fed back to earlier stages

Can Databases Solve This?

No!

But the are probably a part of the solution …