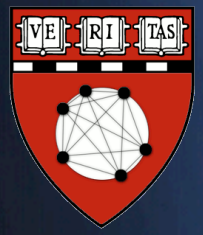
Large Scale Data Management with GridSite

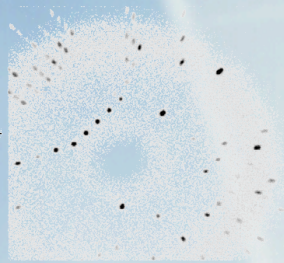
Web-centric data access and visualization



Ian Stokes-Rees
SBGrid/Sliz Lab
Harvard Medical School

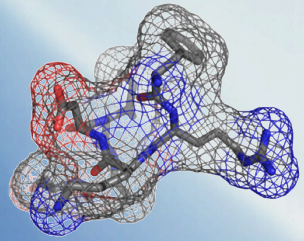
crystal

x-rays



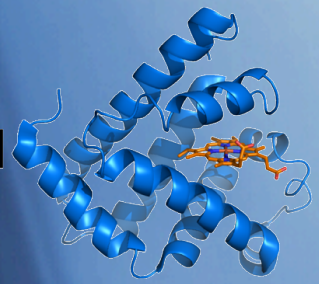
diffraction pattern

phases



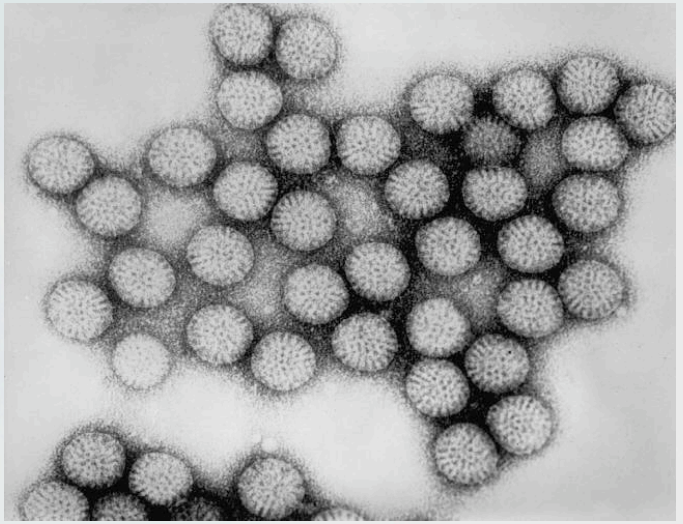
electron density map

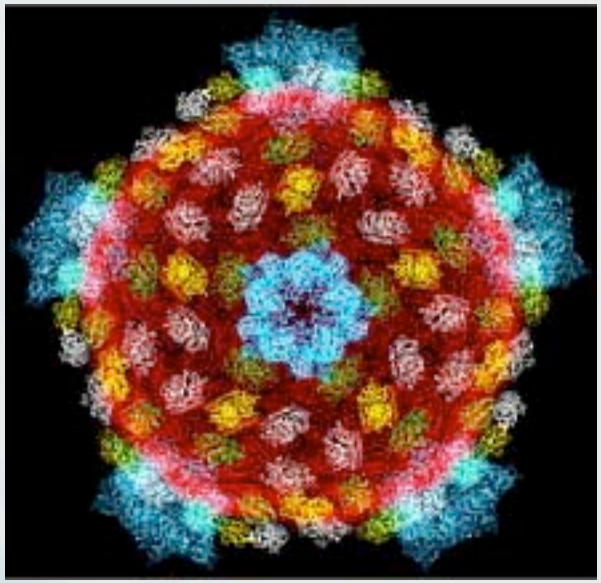
fitting



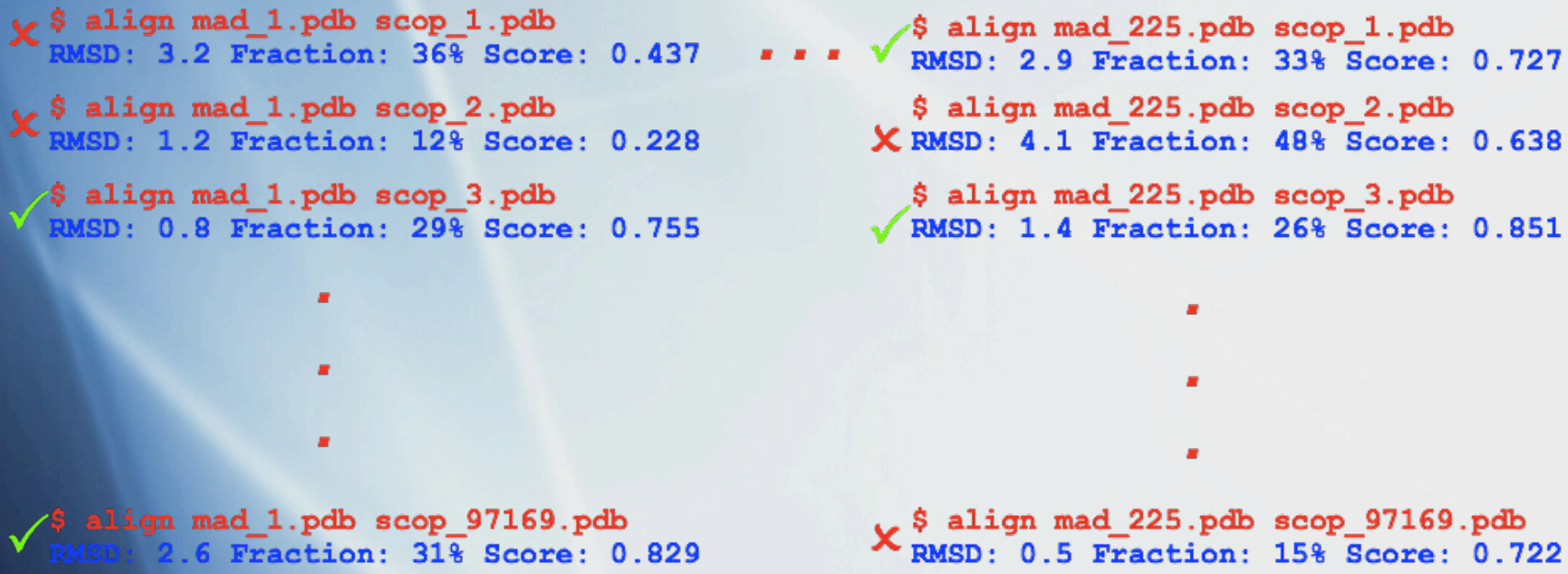
atomic model

refinement





Workflow Overview



- Stage 1: Protein sequence alignment
  - 100,000 x 300 protein pair comparisons
  - 1.5 days wall clock compute time
- Stage 2: Protein model construction
  - 50 x 120 alignment of models to proteins
  - 10-20 days wall clock compute time
- Stage 3: Cluster solutions
  - 50 x 120 rotation alignments

Challenges

- Lots of files and data
  - > 1 million files, 10 GB data per workflow iteration
- Workflow staging
  - 3-5 stages, each dependant upon completion of previous stage and analysis of results
- DB not practical
  - but need to put meta data into DB
- Combining security and sharing
- Collating results into tables and graphs

Approach

- Use GridSite to serve files via http(s)
  - mod_gridsite plugin to Apache httpd
  - Serve "site" and "user" files
    - http://abitibi.sbgrid.org/se/data/site/jobs
    - http://abitibi.sbgrid.org/~ijstokes/jobs
- Job input and output (tarballs) carefully constructed
  - file names and directories
- Each atomic job self-summarizes
  - collated results via
    - cat */summary.row > summary.dat

Key Features of GridSite

- GACL
  - Simple security policies, based on X.509 DN or DN group

```
<gacl>
  <entry>
    <person>
       <dn>/DC=org/DC=doegrids/OU=People/CN=Ian Stokes-Rees 411174</dn>
    </person>
    <allow><list/><read/><write/><admin/></allow>
  </entry>
</gacl>
```

- Shared header and footer
  - allows construction of simple HTML
- gsexec
  - precursor to glexec
  - allows user to use web i/f to run CGI commands as local user
- htcp
  - Make use of HTTP PUT and DELETE
- SlashGrid
  - FUSE module that allows file system mounting of mod_gridsite enabled directories, based on GACL permissions.

Content Delivery

- Static content
  - Accessible via well defined URLs
    - RESTful principle
    - Conceptually easy to think of data organized identically to file system
- "Dynamic" content
  - Generate summary tables and graphs
  - Provide hyperlinks to details
  - Image map hyperlinking is nice
  - Slowly adding in AJAX features (jQuery)
- Link between portal (Django) and GridSite is a challenge

Sage Math

- Python-based scientific/mathematical programming and data exploration environment
- Packages many scientific extensions to Python
- Web-based "notebook" for data sharing and exploration
- For most people, can replace 100% of Matlab
  - and benefit of very similar syntax
- We use this for data analysis and generation of graphics

Take away points

- GridSite provides some great features
  - Can secure web content using simple file -based ACLs tied to existing X.509 PKI
- Combining web-centric data access with file-system features gives best of "both worlds" for large data sets
  - Missing piece is DB-based search and dynamic content generation
  - coming soon with Django portal
- Sage Math is an easy way to integrate powerful data analysis and graphics

Summary

- Acknowledgements:
  - OSG Task Force: Abishek Rana, Greg Thain, Terrence Martin, Jeff Porter, Steve Timm
  - Andrew McNab (GridSite author)
  - Piotr Sliz (PI for SBGrid)
  - Ruth Pordes (continued encouragement with OSG)
  - Members of osg-* mailing lists
- Any questions?
  - http://sbgrid.org
  - ijstokes@crystal.harvard.edu

Ian Stokes-Rees