Welcome to the 7th Biennial NCMI Workshop on Single Particle Reconstruction, Structural Variability and Modeling
CryoEM

- Movie mode alignment/averaging
- Particle Picking
- CTF
- 2-D Class-averaging
- 3-D Reconstruction
- Conformational and Compositional Variability
- Tomography & Subtomogram Averaging
- Helical Reconstruction
- 2-D Crystals
- Phase Plates
EMAN2.1
320 CPU-hr

Relion 1.3
2200 CPU-hr

S. Cerevisiae 80S Ribosome (EMD-2275)
Data taken from PDBe 3DEM test data
no movie alignment performed
Dataset 10002 (Bai XC, Fernandez IS, McMullen G, Scheres SH)
- Single Frames
- Poor DQE past 1/2 Nyquist

- Movies
- Good DQE to Nyquist
- Counting Mode
- Supersampling
- (4 TB/day!)
Fatty Acid Synthase
~30 Å motion

IP3 Receptor
~15 Å motion

TriC (open)
~12 Å motion
CryoEM

- Movie-mode alignment/averaging
- Particle Picking
- CTF
- 2-D Class-averaging
- 3-D Reconstruction
- Conformational and Compositional Variability
- Tomography & Subtomogram Averaging
- Helical Reconstruction
- 2-D Crystals
- Phase Plates
The Plan

- Day 1 - Intro Lectures & EMAN2.12 tutorial
  - Dinner @ BCM-> Vendor Talks
- Day 2 - Structural Variability, Relion Demo
  - Early Release for Halloween
- Day 3 - Structural Variability, SPARX, EMAN Tutorials
  - Dinner @ BCM -> Participant Short talks
- Day 4 - Map Assessment & Model Building
  - Banquet at Trevisio
432/30,000 particles
TEM Produces Projections
# High Resolution SPA

<table>
<thead>
<tr>
<th>Stage</th>
<th>Data</th>
<th>Typ Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Collection</td>
<td>~4 TB/day</td>
<td>Typ ~2 days</td>
</tr>
<tr>
<td>Preprocessing</td>
<td>6 TB -&gt; ~150 GB</td>
<td>~30 hr data I/O</td>
</tr>
<tr>
<td>Particle Selection</td>
<td>150 GB -&gt; ~20 GB</td>
<td>varies</td>
</tr>
<tr>
<td>CTF Correction, etc</td>
<td>20 GB -&gt; 50 GB</td>
<td>minimal, ~1 CPU-hr</td>
</tr>
<tr>
<td>2-D Class-Averages</td>
<td>~500 MB</td>
<td>~10 CPU-hr</td>
</tr>
<tr>
<td>Initial Model</td>
<td>~100 MB</td>
<td>~10 CPU-hr</td>
</tr>
<tr>
<td>Refinement</td>
<td>~10 GB tmp</td>
<td>10-10,000 CPU-hr</td>
</tr>
<tr>
<td></td>
<td>~100 MB final</td>
<td></td>
</tr>
</tbody>
</table>
EMAN2

EMAN2 Wiki:
http://blake.bcm.edu/emanwiki/EMAN2

Software Download page:
http://ncmi.bcm.tmc.edu/ncmi/software/software_details?selected_software=counter_222

Discussion Mailing List/Google Group:
https://groups.google.com/forum/?fromgroups#!forum/eman2
EMAN2 Architecture

Ease of Use

Project Manager Interface

High-Level Programs

Command-Line Programs

Python Core

C++ Core

Flexibility
Programs

Command-Line Programs (EMAN2)

syntax:

e2<name>.py --help

e2<name>.py <file> [--option=value] [--option] [-O]

<> - required parameter

[] - optional parameter

e2help.py <category>
<table>
<thead>
<tr>
<th>File format</th>
<th>Read/Write</th>
<th>Other format</th>
<th>Read/Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRC</td>
<td>R/W</td>
<td>IMAGIC</td>
<td>R/W</td>
</tr>
<tr>
<td>SPIDER</td>
<td>R/W</td>
<td>HDF5</td>
<td>R/W</td>
</tr>
<tr>
<td>PIF</td>
<td>R/W</td>
<td>ICOS</td>
<td>R/W</td>
</tr>
<tr>
<td>VTK</td>
<td>R/W</td>
<td>PGM</td>
<td>R/W</td>
</tr>
<tr>
<td>Amira</td>
<td>R/W</td>
<td>Xplor</td>
<td>W</td>
</tr>
<tr>
<td>Gatan DM2</td>
<td>R</td>
<td>Gatan DM3</td>
<td>R</td>
</tr>
<tr>
<td>Gatan DM4</td>
<td>R</td>
<td>FEI SER</td>
<td>R</td>
</tr>
<tr>
<td>TIFF</td>
<td>R/W</td>
<td>Scans-a-lot</td>
<td>R</td>
</tr>
<tr>
<td>LST</td>
<td>R/W</td>
<td>PNG</td>
<td>R/W</td>
</tr>
<tr>
<td>Video-4-Linux</td>
<td>R</td>
<td>JPEG</td>
<td>W</td>
</tr>
</tbody>
</table>
# Extensible Core

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>Generic image processing algorithms, filters, masks, thresholds, etc.</td>
<td>180</td>
</tr>
<tr>
<td>Aligner</td>
<td>Algorithms used to align 2 images or volumes to each other</td>
<td>22</td>
</tr>
<tr>
<td>Projector</td>
<td>Routines to generate 2-D projections of 3-D objects</td>
<td>6</td>
</tr>
<tr>
<td>Reconstructor</td>
<td>Routines to reconstruct 3-D objects from 2-D projections</td>
<td>11</td>
</tr>
<tr>
<td>Cmp</td>
<td>Similarity metrics used to compare two images or volumes</td>
<td>10</td>
</tr>
<tr>
<td>Averager</td>
<td>Average together stacks of images in various ways</td>
<td>7</td>
</tr>
<tr>
<td>Analyzer</td>
<td>Perform various operations on sets of images, such as classification or PCA</td>
<td>6</td>
</tr>
<tr>
<td>Orientgen</td>
<td>Routines describing how projections cover the asymmetric triangle</td>
<td>6</td>
</tr>
</tbody>
</table>
Utility Programs

- e2version.py – Display version info
- e2speedtest.py – Test machine performance
- e2help.py – Documentation for modular functions
- e2display.py – GUI for general visualization
- e2proc2d.py – 2d image processing of stacks and single images
- e2proc3d.py – 3d image processing of 3-D stacks and single volumes
- e2iminfo.py – general image information tool
- e2.py – Python command-line for EMAN2
GUI

- `e2projectmanager.py` - Project management and workflow
- `e2display.py` - General image/volume display
- `e2filtertool.py` - Interactive 2/3D image processing
- `e2evalimage.py` - Whole micrograph quality/CTF evaluation
- `e2boxer.py` - Interactive particle picker
- `e2helixboxer.py` - Filament picker
- `e2tomoboxer.py` - Interactive tomogram picker
- `e2ctf.py` - Particle-based CTF analysis
- `e2evalparticles.py` - Class-based particle separation/evaluation
- `e2eulerxplor.py` - Evaluate particle orientation distributions
- `e2simmxxplor.py` - Evaluate how well orientations can be determined
High Level Programs

- `e2refine2d.py` - reference free class-averages
- `e2initialmodel.py` - Make initial models from a few class-averages
- `e2refine_easy.py` - Standard single particle 3-D refinement
- `e2eotest.py` - even/odd test for resolution assessment
- `e2refinemulti.py` - multiple map simultaneous refinement
- `e2classifyligand.py` - Split data into 2 groups based on 2 models
Interoperability

- `e2refinetorelion2d.py` - Seed a 2D Relion refinement
- `e2refinetorelion3d.py` - Convert project for 3D Relion refinement
- `e2refinetofrealign.py` - Convert project for Frealign refinement
- `e2runfrealign.py` - Execute Frealign
- `e2refinefromfrealign.py` - Process the results of a Frealign run
- `e2runresmap.py` - Run ResMap on an EMAN2 refinement
Ask These People For Help!

Boxue Ma  Vernon Williams  Zhaoming Su  Jason Kaelber  Muyuan Chen
Wei Dai, Ph.D.  Jesus Galaz, Ph.D.  Brandon Smith  Kaiming Zhang  Bo Chen
Hun Roh, Ph.D.  Rossi Irobalieva  Corey Hryc  Zhao Wang  Michael Bell