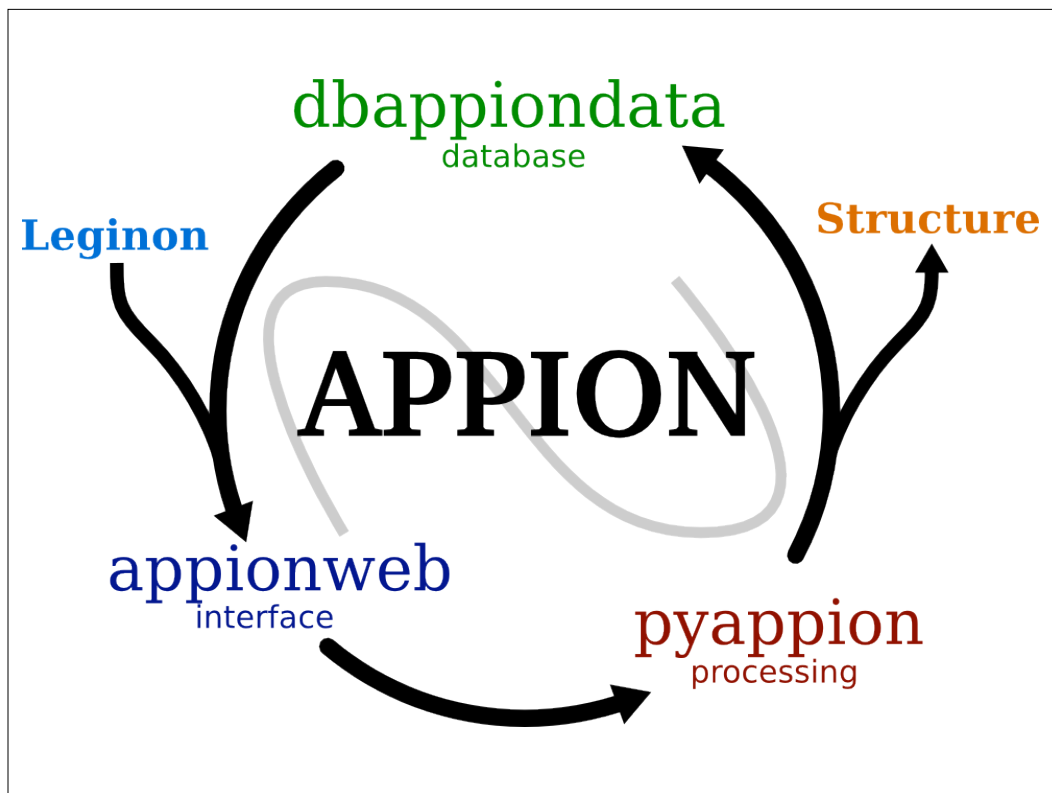


Workshop on Advanced Topics in EM Structure Determination
November 14, 2007

APPION: an automated pipeline for the processing of images

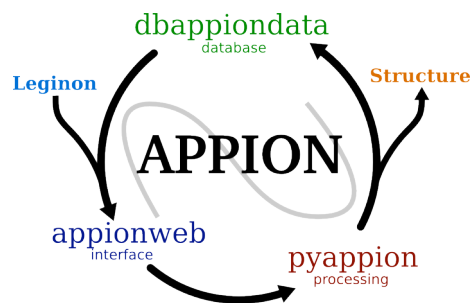
Neil Voss

*AMI: Carragher & Potter lab
The Scripps Research Institute*



Appion: simple, but transparent

- Simple user interface
 - PHP scripts that form HTML form
 - HTML form generate python commands
- Simple user interaction
 - User fills out form and clicks button
 - Copy and paste command into terminal
- Transparent python scripts
 - Easy to develop
 - Easy to follow the process
 - Easy to modify
 - Uses SPIDER, EMAN, ...



Appion: a loop over all images

Most Appion python scripts need to loop over all images and store some result to the database

- Steps in the loop:
 - Global initialization
 - Read parameters
 - For each image:
 - Process image
 - Insert data into DB
- Common parameters:
 - Shuffle images
 - Commit or Test
 - Continue or reprocess
 - Check for more images
 - Only process a few images

Common tasks

- Check for memory leaks
- Common error messages
- Log file
- Parameter parsing
- Statistics
- Write done table
- Query images from DB

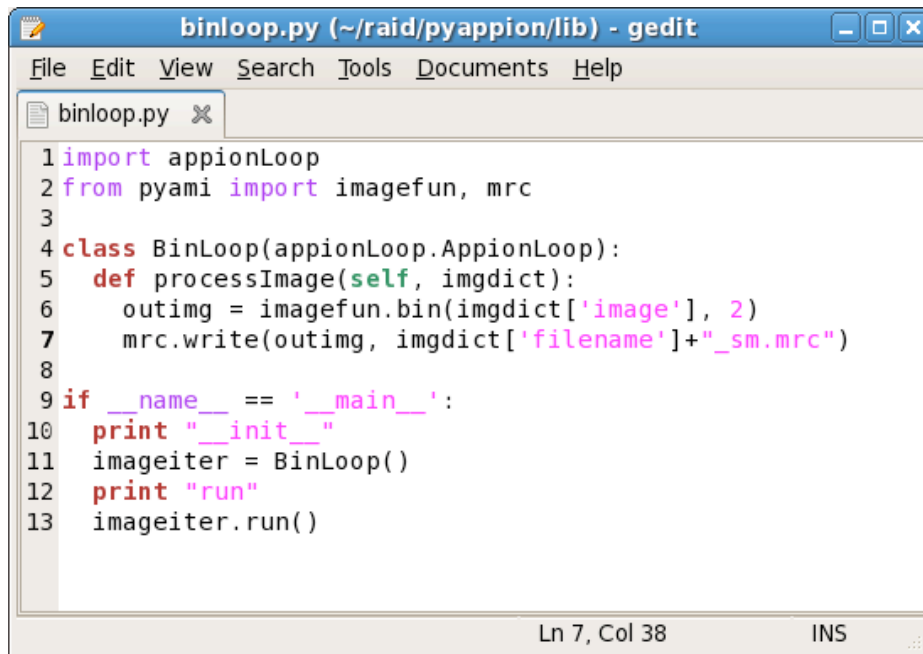
```

vossman@amiv123:~/pyappion
File Edit View Terminal Tabs Help
... Found 22 particles for 07nov01a_045gr_050sq
!!! WARNING: not committing results to database, all data will be lost
... to preserve data start script over and add 'commit' flag

SUMMARY: using updated method for manualpicker
-----
PEAKS:      22 peaks
AVG PEAKS:  23.0 +/- 3.6 peaks
(- TOTAL: 69 peaks for 3 images -)
-----
TIME:       34.64 sec
AVG TIME:   1.36 +/- 1.12 min
(- REMAINING TIME: 2 hrs 6 min for 51 images -)

Starting new image 4 ( skip: 3  remain: 50 ) 07nov01a_045gr_049sq
... processing 07nov01a_045gr_049sq
... filtered image in 0.37 sec
... writing peak JPEG: /ams1/data06/appion/07nov01a/extract/testrun1/jpgs/07nov01a_00045gr_00050sq_
prtl.jpg
... Found 10 particles for 07nov01a_045gr_049sq
!!! WARNING: not committing results to database, all data will be lost
... to preserve data start script over and add 'commit' flag
    
```

Appion: an example



```
binloop.py (~/raid/pyappion/lib) - gedit
File Edit View Search Tools Documents Help
binloop.py x
1 import appionLoop
2 from pyami import imagefun, mrc
3
4 class BinLoop(appionLoop.AppionLoop):
5     def processImage(self, imgdict):
6         outimg = imagefun.bin(imgdict['image'], 2)
7         mrc.write(outimg, imgdict['filename']+"_sm.mrc")
8
9 if __name__ == '__main__':
10     print "__init__"
11     imageiter = BinLoop()
12     print "run"
13     imageiter.run()
Ln 7, Col 38 INS
```

Appion: a pipeline in action

Functions that inherit the looping structure

- Particle pickers
 - Template correlation
 - Difference of Gaussians (DoG)
 - Manual picking & editing
 - Tilt pair alignment
- ACE (CTF estimation)
- Defocal pair alignment
- Stack making
- CTF correction
- Mask maker (crud removal)

Hey Bridget, why don't you just use multivariate statistical analysis with your big database and tell us what conditions give the best data?



Vinzenz Unger, 11/12/07

- Just started uploading refinement parameters last June.
- We do have some early results Euler analysis...

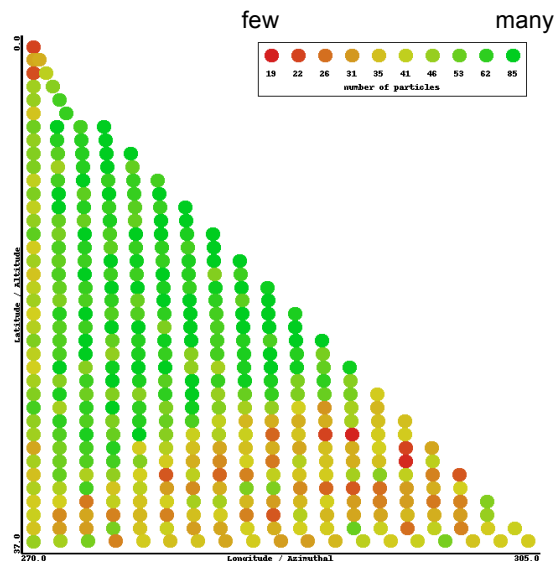
Database in 2007

Image records: **594,506**
 Experiment records: **2891**
 Experiment Size: **15,082,558,053 kB**

	Total Cell	Tables	Data	Indexes	Total
dbemdata	162,355,637	183	905.3 MB	651.4 MB	1.5 GB
processing	663,741,994	50	3.2 GB	2.9 GB	6.2 GB

A Practical Application: *Euler tracking*

- Euler angles for every particle are recorded at each step of refinement.
- Particles with Eulers that change each round are called “jumpers”
- For one dataset: 42% were jumpers
- When jumpers were removed, resolution was unchanged, but detail increased
- In general, “good” datasets have less jumpers, independent on # of particles



S. Stagg *et al.* (2008) JSB, *submitted*

Appion: under the hood

Required for all *appion* tools:

- Images collected with [Leginon](#) (*manual, simple*)
- Python, [Leginon](#)
- PHP
- MySQL

Required for individual modules:

- SPIDER
- EMAN
- FindEM for template picking
- R-measure
- Chimera for volume imaging
- Matlab[†] and ACE for CTF estimation

[†]commercial product

Action	Results	New run
✗ Particle Selection	none	Upload template for picking DoG Picking Manual Picking
✗ CTF Estimation	none	ACE Estimation
✗ Micrograph Assessment	none	Manual Image Assessment
✗ Region Mask Creation	none	Crud Finding Manual Masking
✗ Stacks	none	Pick some particles first
✗ Reference-free Classification	none	Create a stack first
✗ Reference-based Alignment	none	Create a stack first
✗ Reconstructions	none	Create a stack first
Pipeline tools:		
✗ Templates	none	Upload template
✗ Initial Models	none	Upload model

Conclusions

- Even beginning users to perform advanced tasks.
- Common structure for processing images in a loop.
- Transparent python scripts allow developers to easily modify code
- Results are stored in a relational database and linked to the [Leginon](#) database.

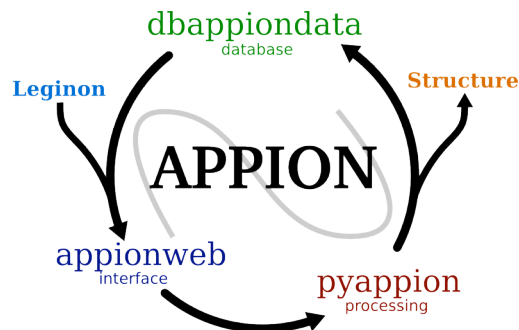
Appion Contributors

- Scott Stagg
- Gabe Lander
- Anchi Cheng
- Neil Voss
- Sunita Nayak
- Pik-wei Lau
- Anke Mulder
- Craig Yoshioka
- Denis Fellman
- Jim Pulokas
- Bridget Carragher
- Clint Potter



Appion Available Now

- Prototype only
- Rapidly changing
- Minimal support
- Requires Legimon



- **pyappion**, is available free at Google Code:
 - <http://appion.org>
- **dbemtools**, including **appionweb** is part of **Legimon**:
 - <http://legimon.org>

Appion: under the hood

Required for all *appion* tools:

- Python, [Leginon](#)
- PHP
- MySQL
- Images collected with [Leginon](#) (*manual, simple*)

Required for individual modules:

- SPIDER
- EMAN
- FindEM for template particle picking
- R-measure
- Chimera for volume imaging
- Matlab[†] and ACE for CTF estimation

[†]*commercial product*

END TALK

Appion: custom loop for particle picking

- Custom Particle Loop
 - Particle specific looping program
 - Workhorse for all four particle pickers
- Common parameters:
 - Image filtering: *low pass, high pass, median, pixel range, binning*
 - Threshold value, min and max
 - Defocal pairs
 - Particle diameter
- Common output:
 - Summary image with particles highlighted
 - Insert particle coordinates into database

Appion: stack processing

Stack-based tasks – *start with a stack of particles*

Implemented so far:

- SPIDER:
 - Reference-free alignment
 - Multi-reference alignment
- EMAN:
 - 3D Reconstruction