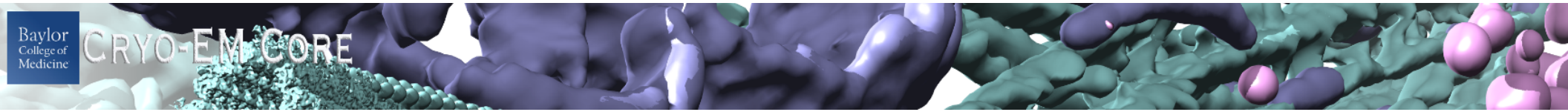


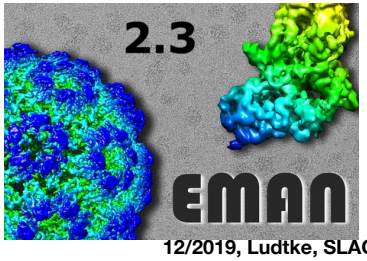
Python and EMAN2

Steve Ludtke

Charles C. Bell Professor
Biochemistry and Molecular Biology
Director, CryoEM/CryoET Core
Co-director CIBR Center
Baylor College of Medicine

Baylor College of Medicine

VERNA & MARRS McLEAN
DEPARTMENT OF
BIOCHEMISTRY AND
MOLECULAR BIOLOGY



12/2019, Ludtke, SLAC

Python ?

PYTHON OOL- developed by Guido van Rossum, and named after Monty Python. (No one Expects the Inquisition) a simple high-level interpreted language. Combines ideas from ABC, C, Modula-3, and ICON. It bridges the gap between C and shell programming, making it suitable for rapid prototyping or as an extension of C. Rossum wanted to correct some of the ABC problems and keep the best features. At the time, he was working on the AMOEBA distributed OS group, and was looking for a scripting language with a syntax like ABC but with the access to the AMOEBA system calls, so he decided to create a language that was extensible; it is OO and supports packages, modules, classes, user-defined exceptions, a good C interface, dynamic loading of C modules and has no arbitrary restrictions.

www.python.org

Python Numbers

- integers
  - effectively unlimited
- floating point
  - 64-bit (15 significant figs, $<10^{308}$)
- complex
  - 5.0+3.0j

Strings

’string’

”also a string”

”””This too

but this one can span lines”””

”A”+” test”

”A test”

Methods of Strings

- upper, lower, title, capitalize
- count, find, rfind, index
- replace
- split
- regular expressions later...

Lists

```
[item1,item2,item3,...]   # items can be anything
a=[0,1,2,3,4,5,6]         # A list of 7 numbers
a[n]                      # nth element in list
a[n:m]                    # sublist elements n to m-1
a[-n]                     # nth item from the end
a[3] -> 3
a[1:4] -> [1,2,3]
a[-2] -> 5
a[2:-2] -> [2,3,4]
a[2]=”x” -> [0,1,”x”,3,4,5,6]
tuples:  a=(0,1,2,3,4,5,6)     # tuples are immutable
a[3] -> 3
a[3]=5 -> ERROR!
```

List Methods

- append, extend
- del, remove
- count
- index
- reverse, sort

Dictionaries

- keys must be immutable, values are arbitrary
- { k1:v1, k2:v2, k3:v3, ... }

  Example:

  a={ 1:2,2:3,”a”:”b”,2.0:3.2,(1,2):”really?” }

  a[1] -> 2

  a[(1,2)] -> “really?”

  a[2] -> 3.2

Dictionary Methods

- has_key
- keys
- values
- items

Operators

- + - add (number), concatenate (list,str)
- - - subtract (number), difference (set)
- * - multiply (number), replicate copies (list,str)
- / - division (number)
- // - integer division (number)
- ** - raise to power (number)
- % - modulus (number), old-sytle string formatting (str)
- & - logical and (number), intersection (set)
- | - logical or (number), union (set)
- ^ - logical exclusive or (number), symmetric difference (set)

for Loops

- Execute 'code' for each item in list, assigning the element to 'var' in each cycle:

```
for var in list:
    code
```

Example:

```
a=[1,2,3,4,5]
for i in a:
    print(i,i*2)
```

Resources

- http://eman2.org/Library - Docs for the C++/Python library
- http://eman2.org/Eman2Metadata - header parameters
- use help()
- e2.py - interactive python interface for EMAN2
- examples/ - lots of small programs, some useful, some just simple code

Key Classes

- EMData - an image in 1-3D, this is the primary class in EMAN2
- Transform - 2/3D Transformation with translation, many conventions
- Symmetry - specifies symmetry transforms
- Processor - an image processing operation, often EMData.process
- Aligner - 2D or 3D registration, often EMData.align
- Averager - Various averaging-like operations
- Reconstructor - 3D reconstruction
- LSXFile - To safely manipulate .lst files

Key Functions

- display(object,[single]) - interactive in e2.py, blocks in regular python
- js_open_dict(filename) - Open a JSON file as a dictionary-like object
- to_numpy(EMData),img.numpy() - Convert an EMData object to a NumPy array
- from_numpy(array) - Convert a NumPy array to EMData
- test_image(size=<n>,type=<i>) - Generate various 2-D test images
- os.listdir(<path>) - useful standard python function
- good_size(boxsize) - next good box size larger than specified
- base_name(filename) - returns unique portion of image filename
- info_name(filename) - returns json file associated with filename in project
- EMUtil.get_image_count(filename) - number of images in file

EMData

- img=EMData(nx,ny,nz)
- img=EMData(filename,n,[hdr_only],[region])
- img.read_image(filename,n,[hdr_only],[region]) - replaces current image
- list=EMData.read_images(filename,[list of #s],[header only])
- img.write_image(fsp,image #,[filefmt],[hdr_only],[region],[numtype],[endian])
- img[x,y,z] - returns (or sets with assignment) value at coordinates
- img["name"] - returns header value (if exists), or sets with assignment
- img.get_attr_dict() - gets all header items as a dictionary
- img.to_zero() - all pixels to 0.0
- img.to_one() - all pixels to 1.0
- img2=img.get_clip(Region(x0,y0,[z0],nx,ny,[nz])) - extract subimage
- img.insert_clip(target,(x,y,z)) - insert subimage into another image

EMData

- fft=img.do_fft() - compute FFT
- img=fft.do_ift() - compute inverse FFT img2=img.process("processor",{parms})
- img.process_inplace("processor",{parms})
- img=img1.align("aligner",img2,{params})
- similarity=img1.cmp("cmp",img2,{params})
- img2=img+img1
- img.add(img2)
- img.mult(img2)
- img.calc_fourier_shell_correlation(img2) - see help()
- locs=img.calc_n_highest_locations(<n>)
- calc_radial_dist(<n>,<r0>,<dr>,[inten]) - radial curve, real or FFT

Transform

- xf=Transform() - identity matrix with no translation
- xf=Transform({"type":"eman","az":5,"alt":10,"phi":15})
- xf.inverse() - inverse of the Transformation including translation
- xf.get_rotation("eman") - e2help.py rotationtypes
- xf.printme() - print the actual matrix
- xf*xf2 - multiply matrices
- Transform.icos_5_to_2() - transform to go from icos 5-fold on Z to 2-2-2
- Transform.tet_3_to_2() - tetrahedral 3-fold on Z to 2-fold on Z

Symmetry

- sym=Symmetries.get("icos") - e2help.py symmetries
- n=sym.get_nsym() - number of asymmetric units
- list=sym.get_syms() - list of all Transforms for this symmetry
- list=sym.gen_orientations("name",{parms}) - make a set Transforms within an asymmetric unit