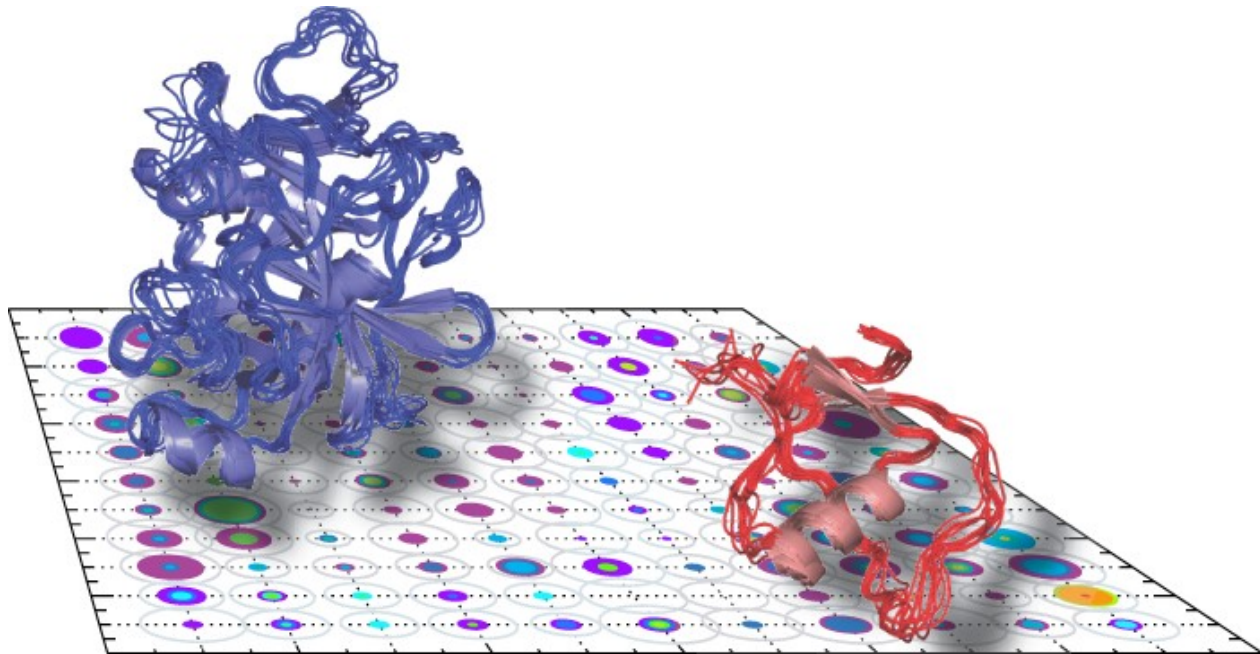


Python in Science

Type "help", "copyright", "credits" or "license" for more information.

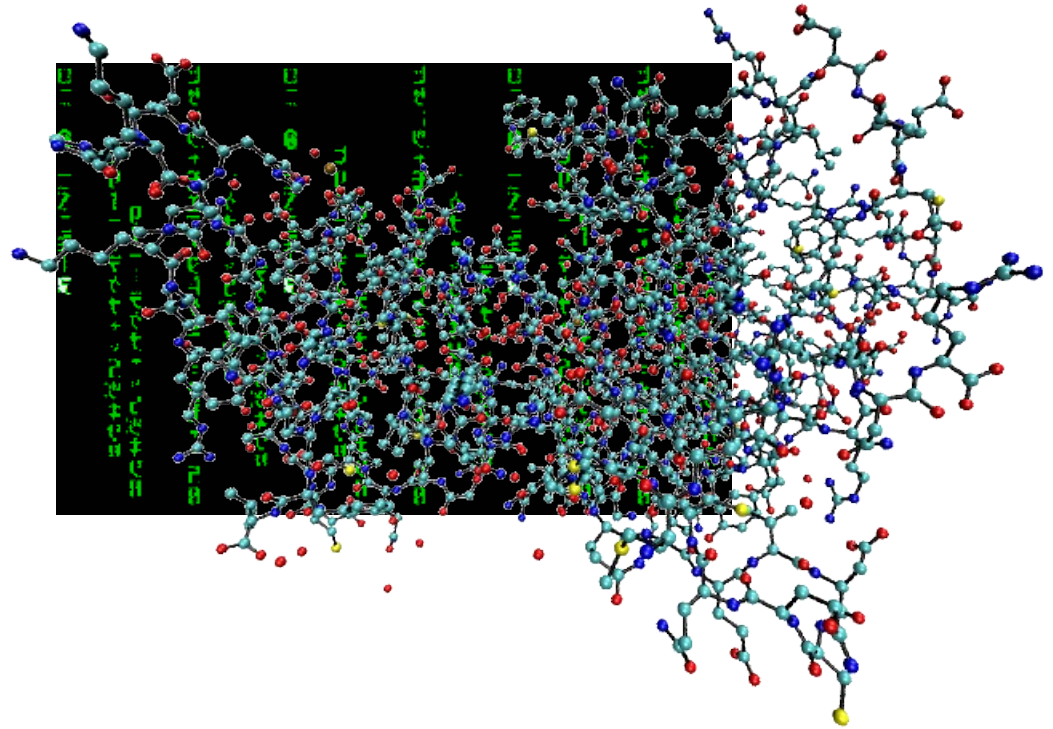
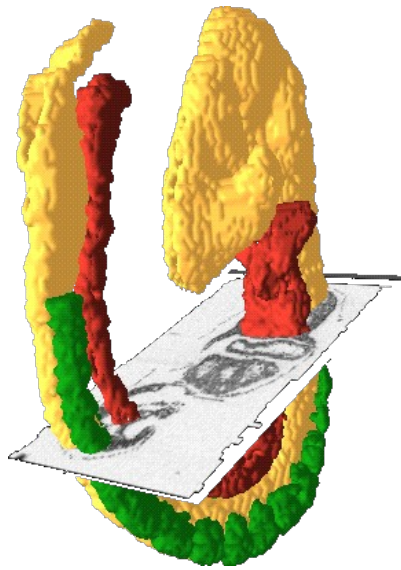
```
>>>
```



Content

- **Scientific Programming Challenges**
- **Proteins on the edge**
- **Scientific Libraries**
- **Interactive programming**
- **From scripts to large projects**
- **Conclusion**

Scientific Programming Challenges



Scientific Programming Challenges

- **Large amounts of diverse and messy data**
- **Unclear project outcome / specifications**
- **Everyday new algorithms and tricks**

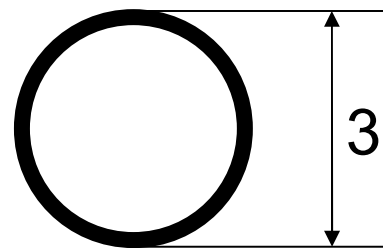
- **Scientists are no programmers!**

- **Culture of exchange and collaboration**
- **(and competition)**

Proteins on the edge

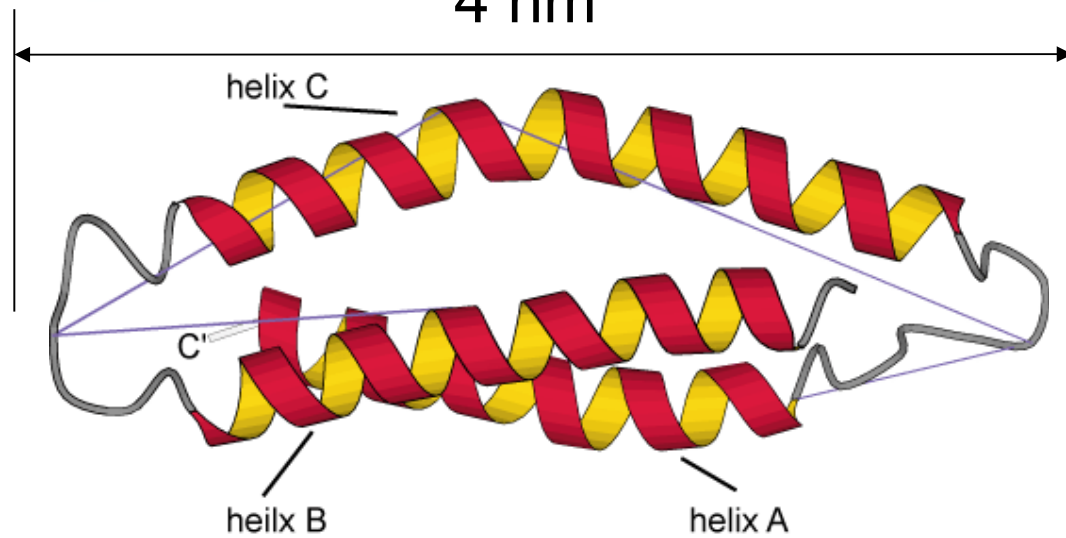


7-8 μm



3-5 μm

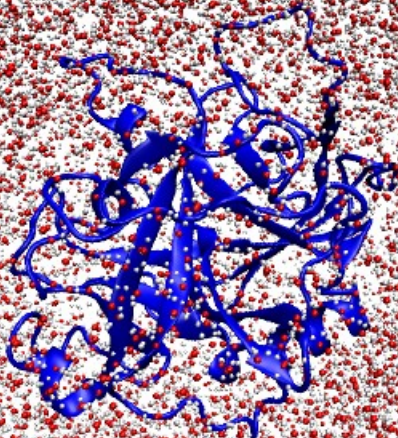
4 nm





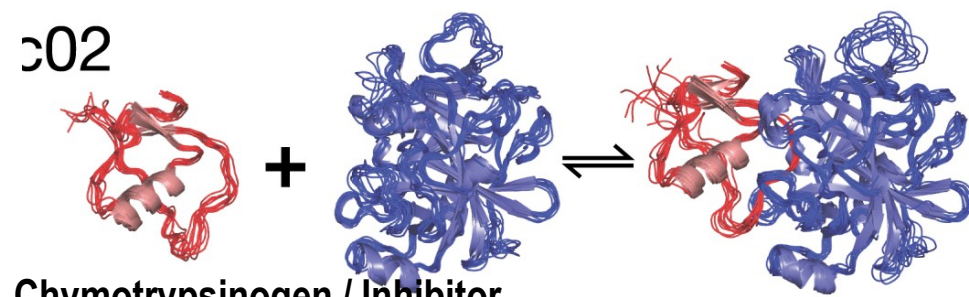
Molecular Dynamics Simulations

10 x 1 ns

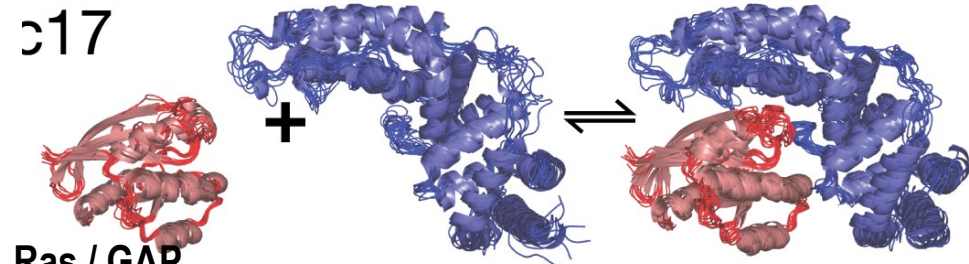


②

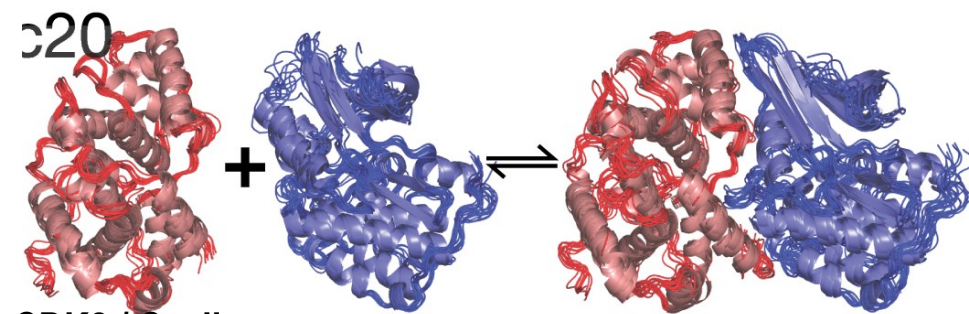
Free and bound ensembles



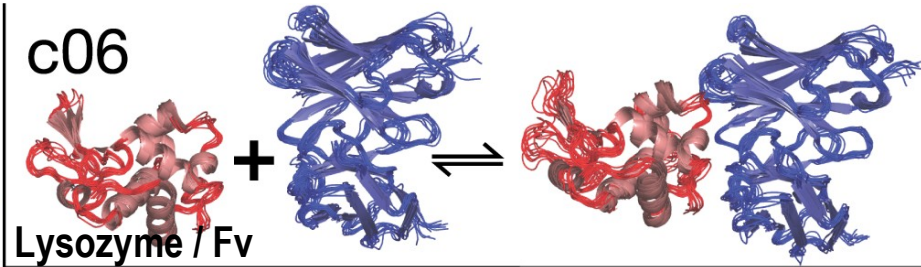
Chymotrypsinogen / Inhibitor



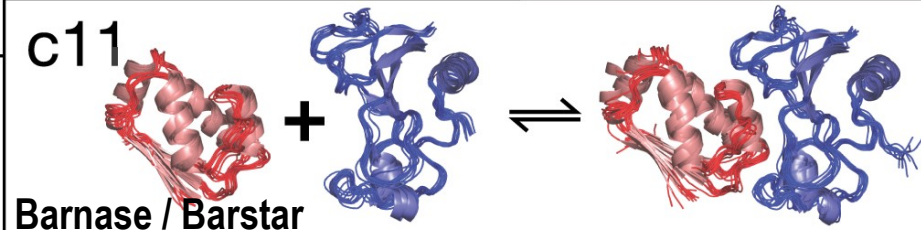
Ras / GAP



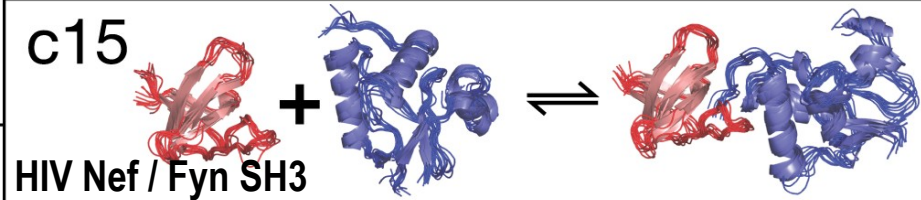
CDK2 / Cyclin



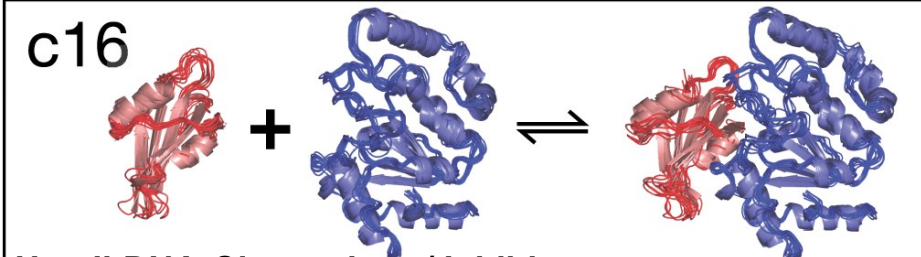
Lysozyme / Fv



Barnase / Barstar



HIV Nef / Fyn SH3



Uracil-DNA-Glycosylase / Inhibitor

Scientific Python Libraries

Wiki

[Documentation](#)
[Mailing Lists](#)
[Download](#)
[Installing SciPy](#)

[Topical Software](#)

[Cookbook](#)
[Developer Zone](#)
[Blogs](#)

Page

[Immutable Page](#)
[Info](#)
[Attachments](#)

More Actions: ▼

Topical Software

This page indexes add-on software and other resources relevant to SciPy, categorized exhaustively. If you know of an unlisted resource, see [About This Page](#), below.

Contents

1. [About This Page](#)
2. [General Python resources](#)
3. [Tutorials and texts](#)
 1. [Some generic Python/programming tutorials:](#)
 2. [And some more specifically geared towards scientific computing:](#)
4. [Working environments](#)
5. [Science: basic tools](#)
6. [Plotting, data visualization, 3-D programming](#)
 1. [Tools with a \(mostly\) 2-D focus](#)
 2. [Data visualization \(mostly 3-D, surfaces and volumetric rendering\)](#)
 3. [LaTeX, PostScript, diagram generation](#)
 4. [Other 3-D programming tools](#)
7. [Wrapping C, C++ and FORTRAN codes](#)
8. [Optimization](#)
9. [Data Storage / Database](#)
10. [Parallel and distributed programming](#)
11. [Topic guides, organized by scientific field](#)
 1. [Astronomy](#)
 2. [Artificial intelligence & machine learning](#)
 3. [Bayesian Statistics](#)
 4. [Biology](#)
 5. [Dynamical systems](#)
 6. [Economics and Econometrics](#)
 7. [Electromagnetics](#)
 8. [Geosciences](#)
 9. [Molecular modeling](#)
 10. [Signal processing](#)
 11. [Symbolic math, number theory, etc.](#)
12. [Miscellaneous](#)

• http://scipy.org/Topical_Software

Scientific Python Libraries

- **SciPy, numpy, ScientificPython**
- **gnuplot, biggles**
- **pyPvm**

SciPy

- <http://scipy.org>

Tools

- [Cluster](#) : vector quantization / kmeans
- [Fftpack](#) : discrete fourier transform algorithms
- [Integrate](#) : integration routines
- [Interpolate](#) : interpolation tools
- [Linalg](#) : linear algebra routines
- [Misc](#) : various utilities (including Python Imaging Library)
- [Ndimimage](#) : n-dimensional image tools
- [Optimize](#) : optimization tools
- [Signal](#) : signal processing tools
- [Sparse](#) : sparse matrices
- [Stats](#) : statistical functions

Other packages

- [Io](#) : data input and output
- [Lib](#) : wrappers to external libraries (BLAS, LAPACK)
- [Sandbox](#) : incomplete, poorly-tested, or experimental code
- [Special](#) : definitions of many usual math functions
- [Weave](#) : C/C++ integration

You might also want to check out the [SciPy API documentation](#)

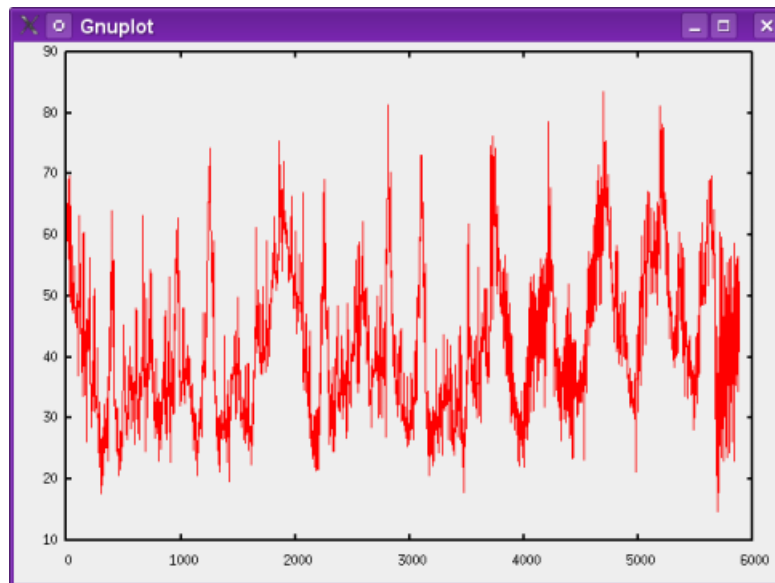
NumPy

- <http://scipy.org>
 - multi-dimensional arrays
 - C-coded – fast!
 - all kind of number crunching and analysis

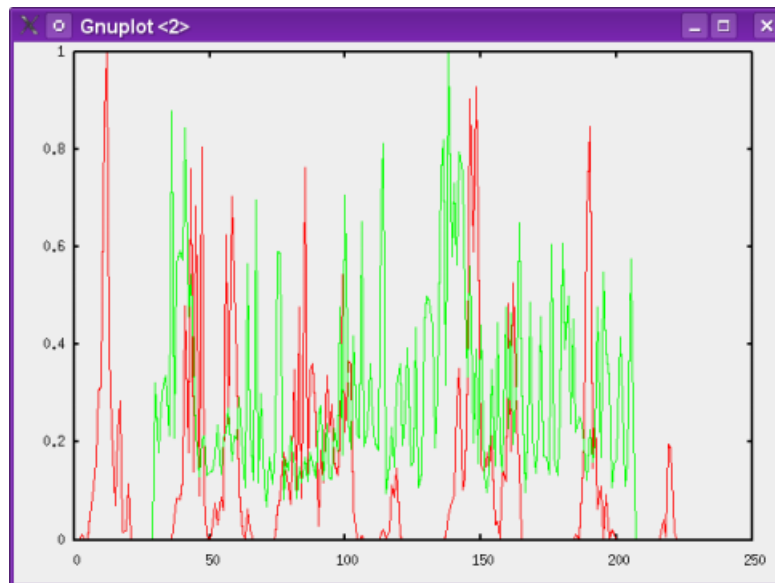
Plotting Libraries

gnuplot.py

```
>>> import gnuplot as G  
>>> G.plot( temperature )
```



```
>>> import gnuplot as G  
>>> G.plot( temperature, press )
```



Plotting Libraries

biggles – <http://biggles.sf.net>

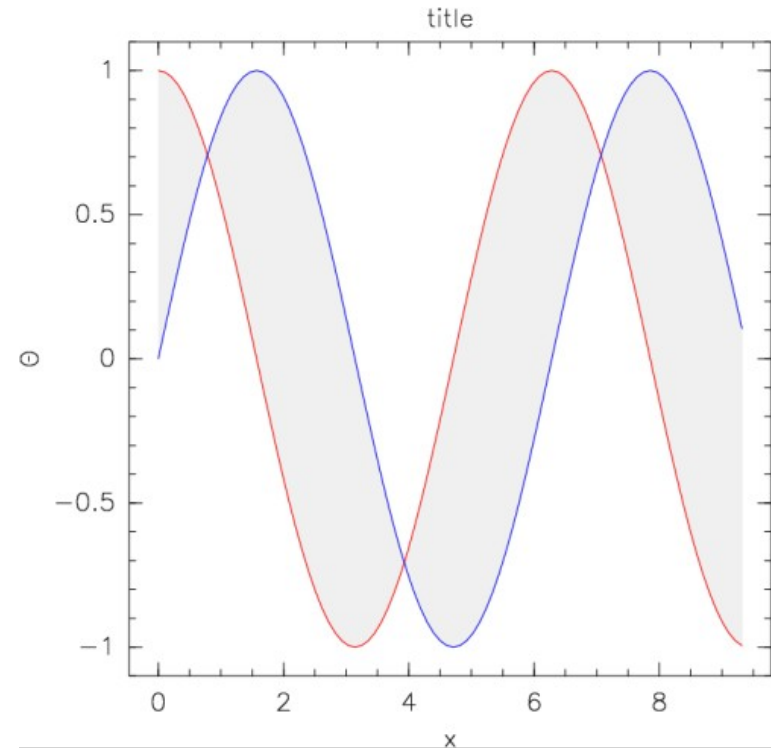
```
import biggles
import numpy, math

x = numpy.arange( 0, 3*math.pi, math.pi/30 )
c = numpy.cos(x)
s = numpy.sin(x)

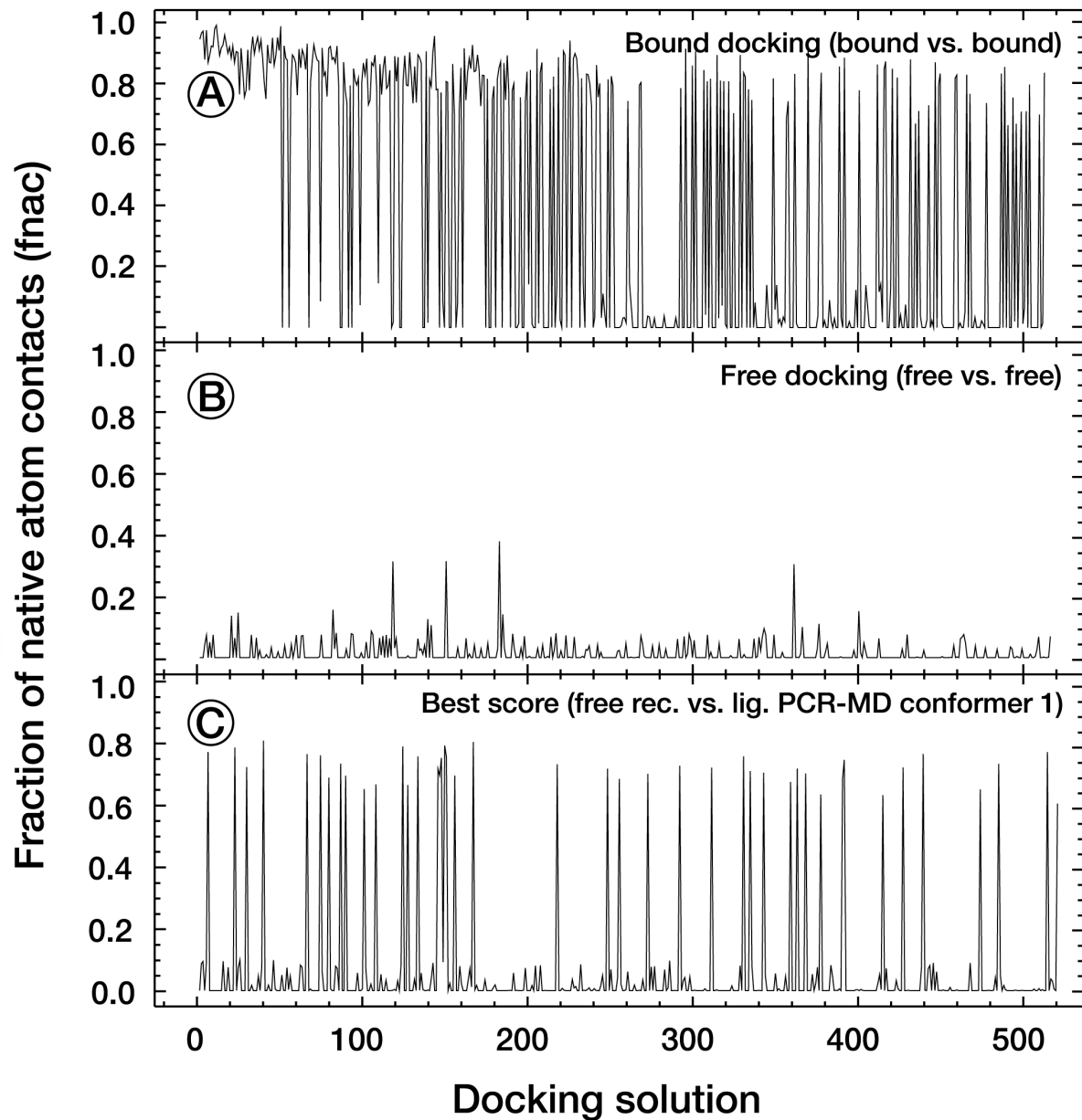
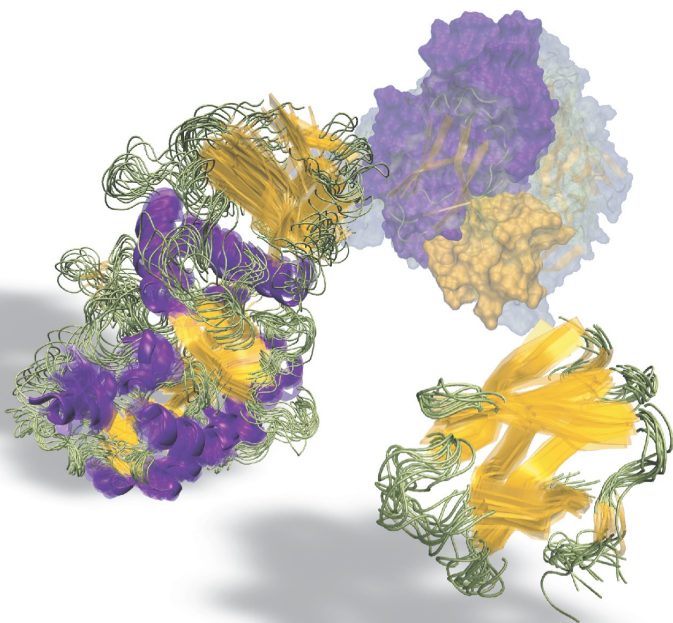
p = biggles.FramedPlot()
p.title = "title"
p.xlabel = r"$x$"
p.ylabel = r"$\Theta$"

p.add( biggles.FillBetween(x, c, x, s) )
p.add( biggles.Curve(x, c, color="red") )
p.add( biggles.Curve(x, s, color="blue") )

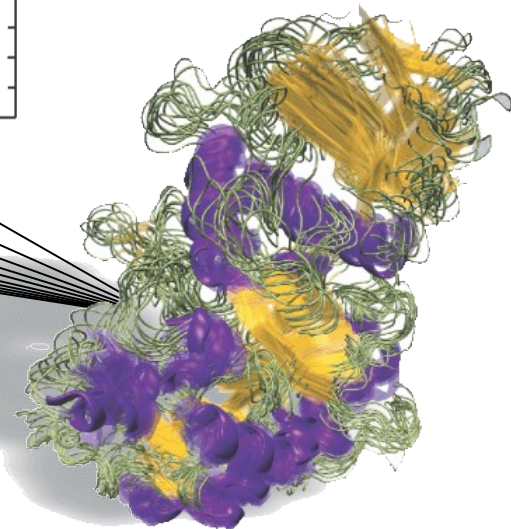
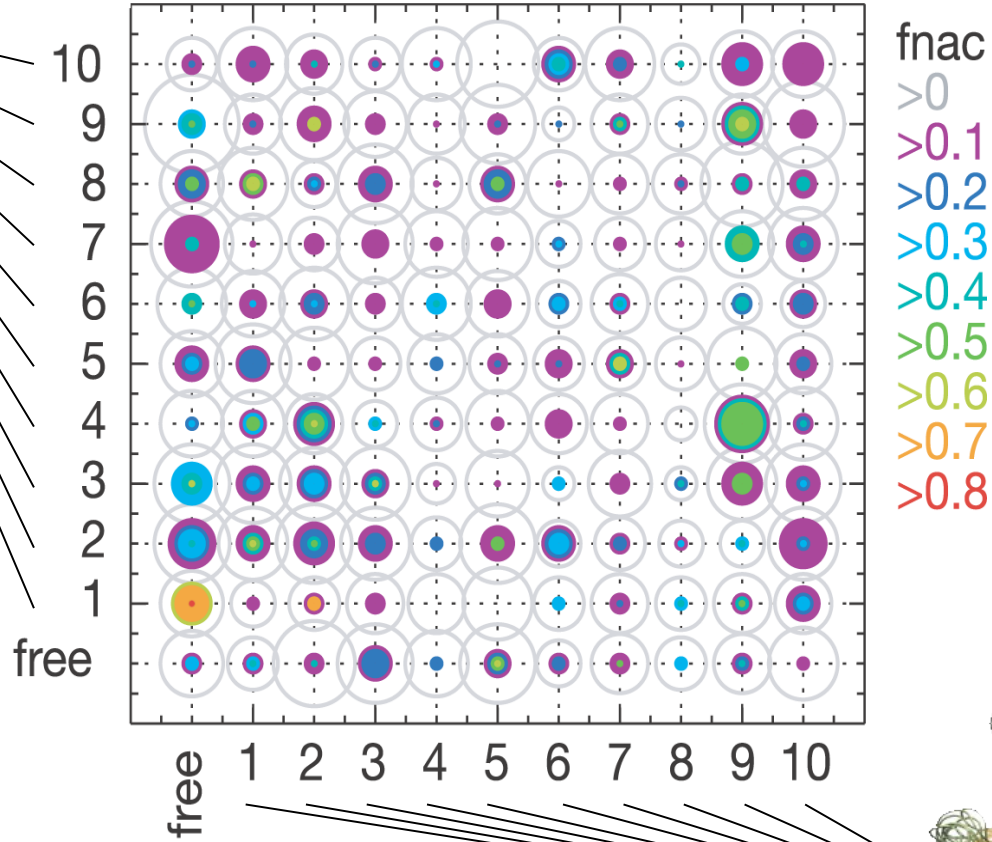
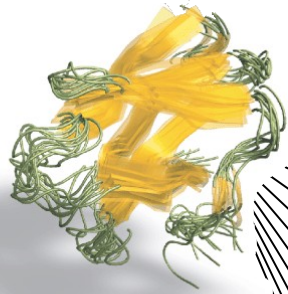
p.show()          # X11
p.write_eps( "example1.eps" )
p.write_img( 400, 400, "example1.png" )
```



more advanced plotting examples

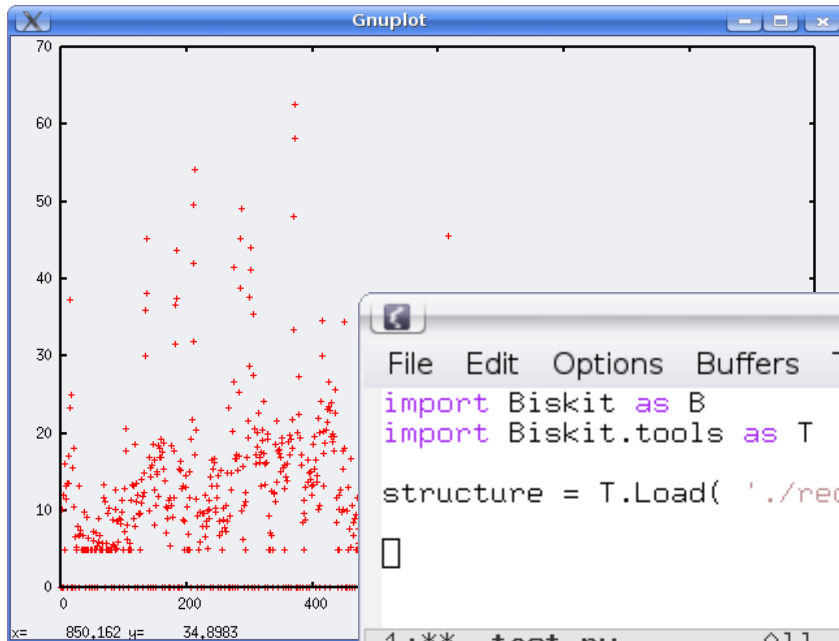


... and more plotting



[Grünberg, Leckner, Nilges (2004) Structure 12, 2125-36]

interactive programming



```
emacs@kaa
File Edit Options Buffers Tools Complete In/Out Signals Help

import Biskit as B
import Biskit.tools as T

structure = T.Load( './rec.model' )

□

-1:** test.py All (6,0) (Python)-----
Python 2.5.2 (r252:60911, Apr 21 2008, 11:17:30)
[GCC 4.2.3 (Ubuntu 4.2.3-2ubuntu7)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> ## working on region in file /tmp/python-30479DD1.py...
>>> structure.report()
[PDBModel 1BGS 1093 atoms, 110 residues, 1 chains]
    * chain 0: AQVINTFDGVADYLQTYHKLPDNYITKSEAQALGWVASKGNLADVAPGKSIGGDIFSN..
  source: LocalPath[ {/data/raik/py/biskit|$projectRoot}/test/com/rec.model ]
 15 atom profiles:  ['name', 'insertion_code', 'residue_number', 'alternate..
 0 residue profiles: []
 1 info records:  ['date']
>>> from Biskit.gnuplot import *
>>> scatter( structure['temperature_factor'] )
>>> □

-1:** *Python* All (14,4) (Comint:run)-----
```

1. edit file

2. execute into buffer

3. interrogate result

from scripts to large projects

pyPvm – Parellization

<http://pypvm.sf.net> ... lower-level PVM interface

<http://biskit.sf.net> – Biskit/PVM

... high-level pypvm wrapper /
M. Habeck & W.

```
from Biskit.PVM import TrackingJobMaster

class Master(TrackingJobMaster):

    ## Slave script that goes with this master
    slave_script = '~/py/ExampleSlave.py'

    def __init__(self, *args, **kw):
        TrackingJobMaster.__init__(self, *args, **kw)

    def getInitParameters(self, slave_tid):
        return {'progress_str': 'slave calculating..'}

    def cleanup(self):
        pass

    def done(self):
        pass
```

```
from Biskit.PVM import JobSlave

class Slave(JobSlave):

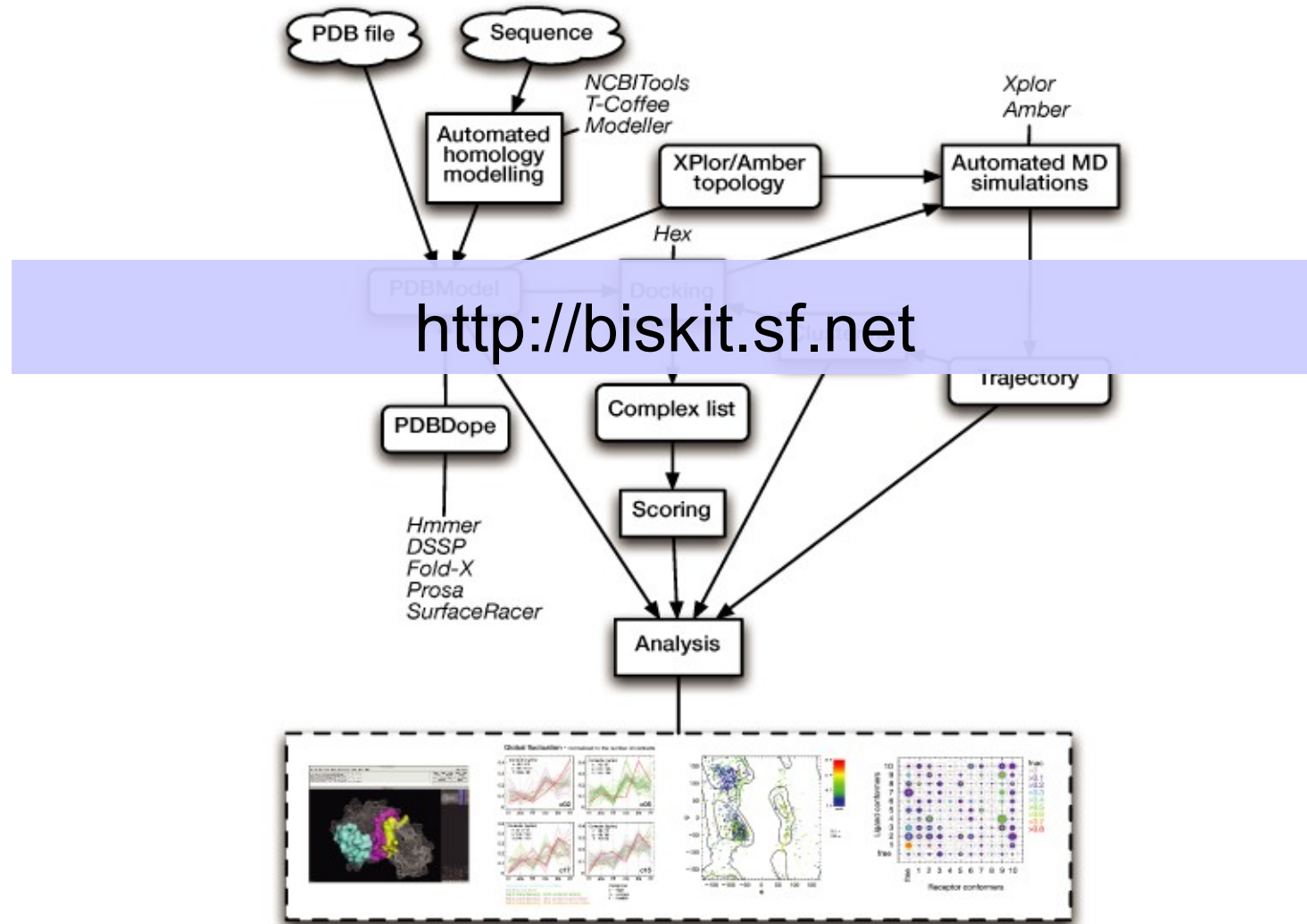
    def initialize(self, params):
        self.params = params

    def go(self, dict):
        d = {}

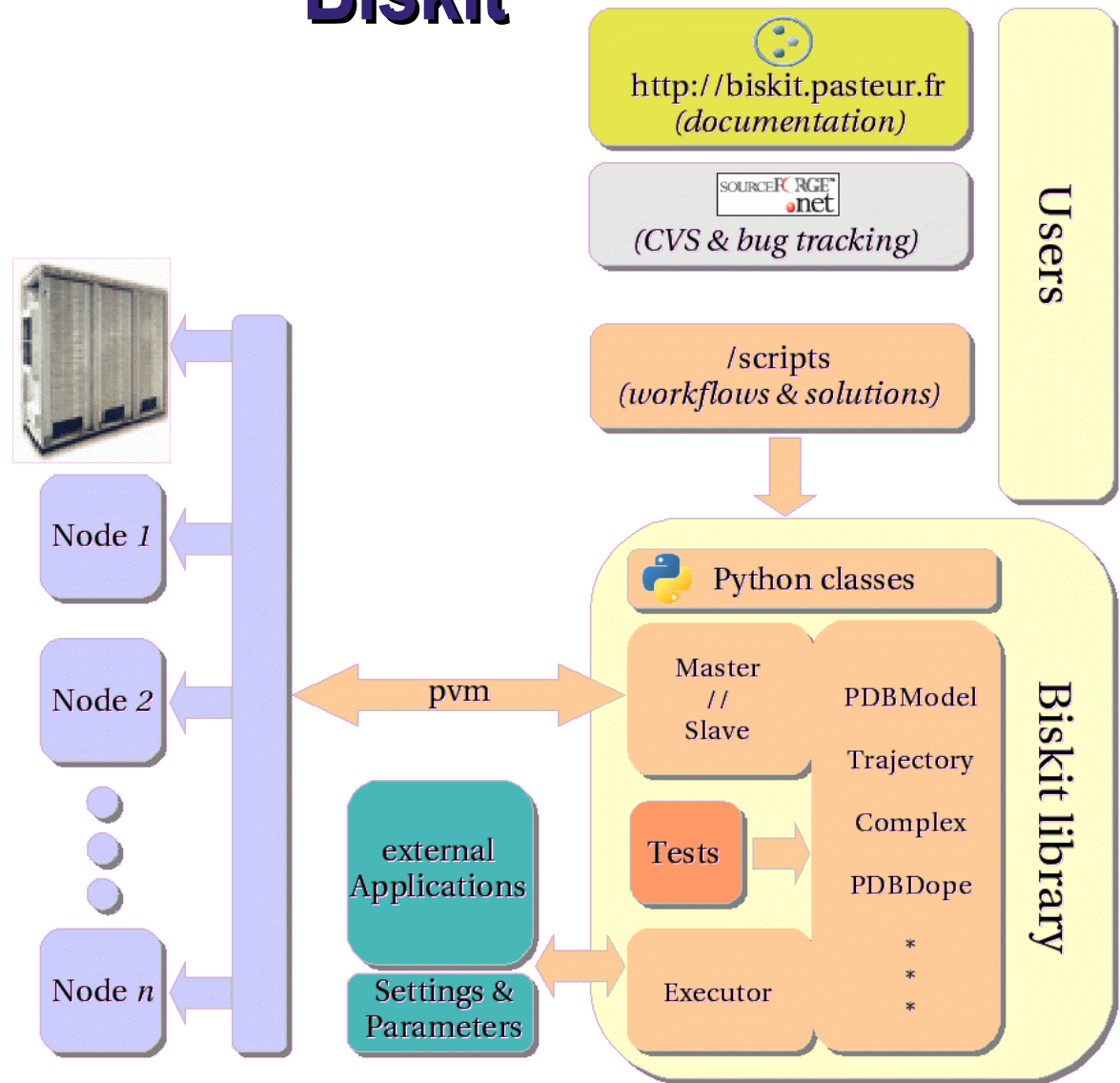
        for id, val in dict.items():
            d[id] = val+1 ## do your calculation

        return d
```

Biskit



Biskit



New Frontier: Synthetic Biology

What are Biobricks?



Part Types - Registry - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://parts.mit.edu/registry/index.php/Part_Types

openSUSE Getting Started Latest Headlines

Help:BioBrick Prefix and... Synthetic Biology: BioBr... StdAsssmall.png (PNG I... Protein Biobricks — SFil... Part Types - Registry

Create an account or log in

article discussion edit history

Part Types

Browse parts by projects:

[iGEM 2007](#) | [iGEM 2006](#) | [iGEM 2005](#) | [Labs](#) | [Courses](#) | [Featured Parts](#)

Or browse parts by [part categories](#):

Systems

- [Measurement](#)
- [Measurement \(Under Development\)](#)
- [Projects\(empty\)](#)

Devices

- [Reporters](#)
- [Protein Generator](#)
- [Inverters](#)
- [Composite Devices](#)
- [Signalling](#)
- [Measurement](#)

Parts

- [Ribosome Binding Sites](#)
- [Protein Coding](#)
- [Regulatory](#)
- [Terminators](#)
- [RNA](#)
- [DNA](#)
- [Conjugation](#)

Chassis

- [E.coli Strains](#)

Mammalian

Vectors

- [Plasmids](#)

Other

- [Yeast Parts](#)
- A.B** [Construction Intermediate](#)
- [PCR Primer](#)
- [Tags](#)
- [Other](#)
- [Deleted](#)
- [Bacteriophage T7](#)

- For more information on each part type, click the [?](#) for [help documentation](#) next to the category name.
- To discuss how to measure the functions of these parts visit [Characterization of Parts](#)



jump to part

- navigation
- Main Page
 - Browse Part Types
 - iGEM 2007 Wiki
 - Community portal
 - Recent changes
 - Recent part changes

- resources
- User Accounts
 - Add a Part
 - Part Searches
 - DNA Repositories
 - Sequence Analysis
 - Assembly Tool
 - Help

search

Go Search

- toolbox
- What links here
 - Related changes
 - Upload file
 - Special pages
 - Printable version

Part:BBa_J06702 - Registry - Mozilla Firefox

File Edit View History Bookmarks Tools Help


http://parts.mit.edu/registry/index.php/Part:BBa_J06702

openSUSE Getting Started Latest Headlines

Help:BioBrick Prefix and... Synthetic Biology: BioBr... StdAsssmall.png (PNG I... Protein Biobricks — SFil... Part:BBa_J06702 - Regis...

Create an account or log in

article discussion edit history

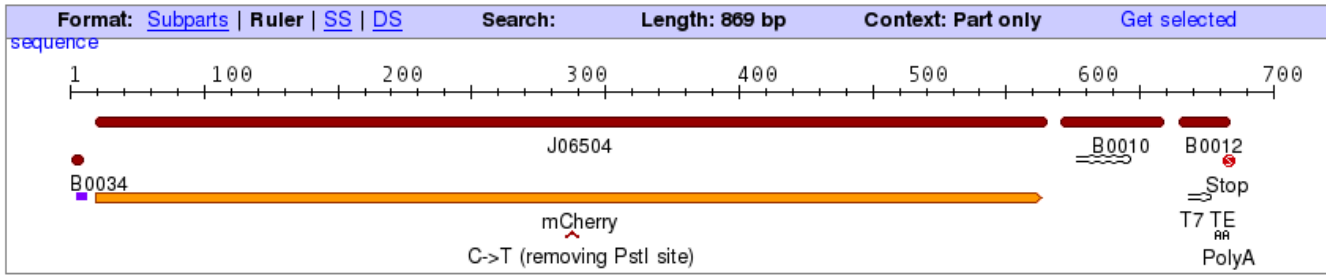
Part:BBa_J06702  **DNA Available**
Experience:

Designed by Yves Wang Entered: 2005-07-25

mCherry, bacterial with RBS and forward terminator
Combines BBa_B0015 forward terminator with BBa_J06602 mCherry, bacterial with RBS

Sequence and Features

Format: [Subparts](#) | [Ruler](#) | [SS](#) | [DS](#) Search: Length: 869 bp Context: Part only [Get selected](#)



sequence

1 100 200 300 400 500 600 700

B0034 J06504 mCherry B0010 B0012 Stop T7 TE PolyA

C->T (removing PstI site)

jump to part

BBa_

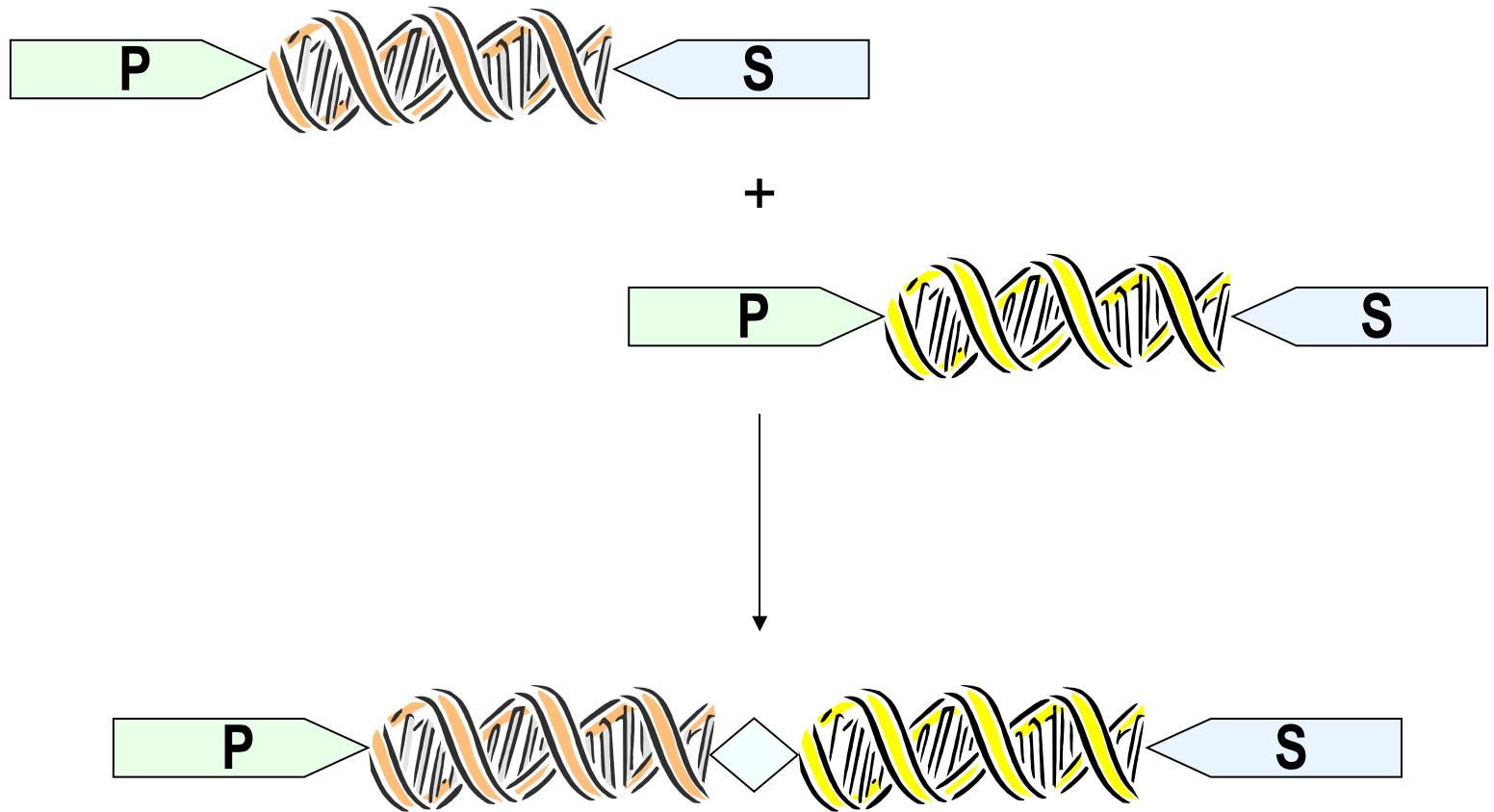
BBa_J06702

- Part Main Page
- Part Design
- Experience
- Hard Information
- Physical DNA

navigation

- Main Page
- Browse Part Types
- iGEM 2007 Wiki
- Community portal
- Recent changes
- Recent part changes

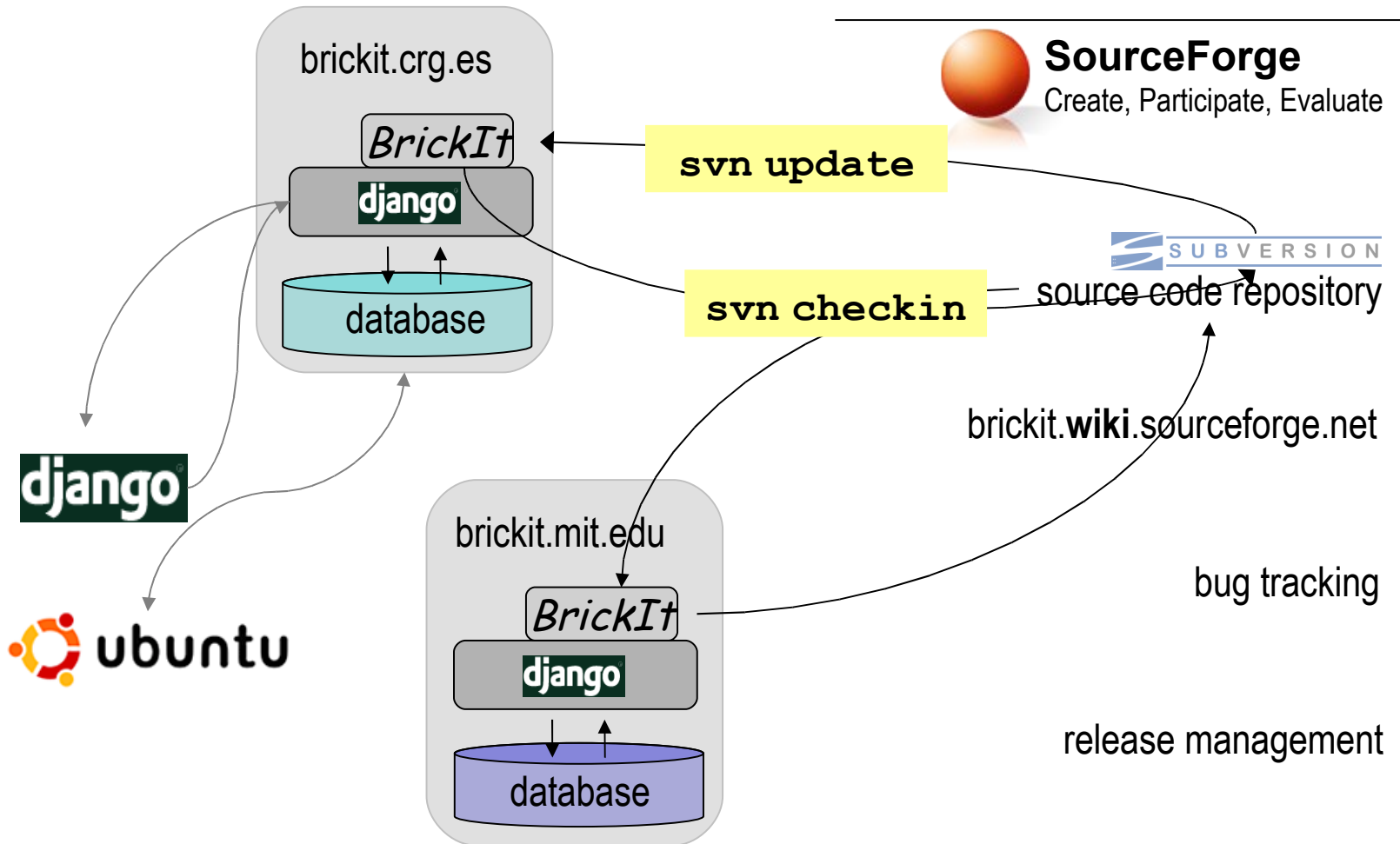
What are Biobricks?



BrickIt – Open Source BioBrick management

- <http://brickit.sf.net>

shared development



local data but *shared* infrastructure & development

Python + Science = perfect match

- **Many libraries and tools – vibrant communities**
- **rapid prototyping – optimal for “discovery mode”**
- **interactive programming – optimal for data analysis**
- **smooth growth into well structured, large projects**
- **and ...**
- **... because it doesn't suck.**