

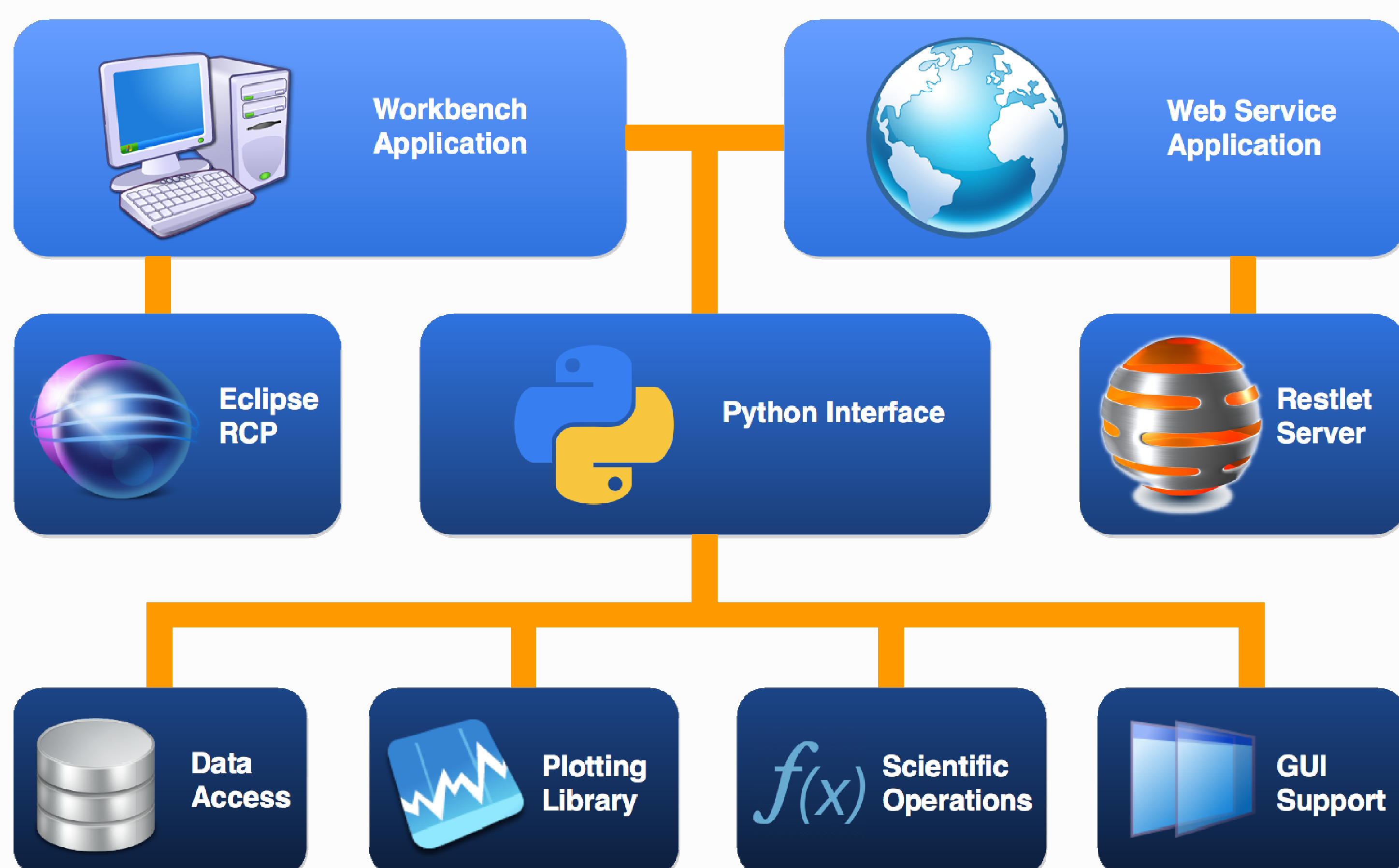


N. Xiong, D. Mannicke, N. Hauser, The Bragg Institute, ANSTO
Norman.Xiong@ansto.gov.au; Locked Bag 2001, Kirrawee DC NSW 2232 AUSTRALIA

N. Xiong, D. Mannicke, N. Hauser, The Bragg Institute, ANSTO
Norman.Xiong@ansto.gov.au; Locked Bag 2001, Kirrawee DC NSW 2232 AUSTRALIA



The GUMtree scripting interface has a *numpy*-like Python library that makes data treatment easier. It also has a GUI library that automatically generates control panels for scripts. Same scripts can be loaded in both the desktop application and the web service application for online data treatment. The scripting interface benefits both the users and the developers. Users can easily make scripts to run experiments or treat the data, with a graphical interface automatically created for these scripts. Developers save a lot of time in compiling and deploying products when using this feature.



- ◆ Scientific Computing Library: More than *numpy*
 - multi-dimensional array object
 - mathematical operations
 - error propagation
 - carrying axes information
- ◆ Library to Read Data from Different Sources
- ◆ Plotting Library
 - 1-D and 2-D plotting
 - real-time data plotting
 - fitting support
- ◆ Instrument Control Module
- ◆ Auto-GUI Support
 - generic GUI interface
 - desktop implementation
 - web service implementation

- ◆ Treat Experiment Data
 - preview data and send feedback to experiment
 - treat data on a local computer or on the server
- ◆ Align Instrument
- ◆ Create Quick Experiment Interfaces
- ◆ Make Highly Customisable Features

The screenshot displays the Origin 8.5 software interface, showing a multi-peak fitting project. The 'Data' window on the left lists various data files. The 'Fitting' window shows the 'Gaussian Fit' model. The 'Plot' window displays the 'bm2_counts vs s2' plot with a multi-peak fit. The 'Scripting Console' window shows the 'Fit' command being executed. The 'Command' window shows the 'Add peak position to Plot3' command.

Plot Window: bm2_counts vs s2

The plot shows 'bm2_counts' on the y-axis (ranging from 2,500 to 15,000) versus 's2' on the x-axis (ranging from -38.50 to -31.50). A multi-peak fit is shown, with individual peaks highlighted in different colors (blue, green, red, yellow, purple). The fit is labeled 'bm2'.

Scripting Console

```

[2015-07-14 14:48:19] F1RM001="14130115509 +/- 0.538638272277
[2015-07-14 14:48:19] CH12 = 0.00285247860058
[2015-07-14 14:48:19] POS_OF_PEAK=-89.067808193 +/- 0.15794682975
[2015-07-14 14:48:19] F1RM001="04989469333 +/- 0.720664466196
[2015-07-14 14:48:19] CH12 = 0.00053568809244
[2015-07-14 14:48:19] POS_OF_PEAK=-57.0293164778 +/- 0.155755193315
[2015-07-14 14:48:19] F1RM001="444591516075 +/- 0.792460232678
[2015-07-14 14:48:19] CH12 = 1.38858131642e-05
[2015-07-14 14:48:19] POS_OF_PEAK=-39.44894915748 +/- 0.14009412454
[2015-07-14 14:48:19] F1RM001="78324752335 +/- 0.440362624205
[2015-07-14 14:48:19] CH12 = 5.78408890224e-05
[2015-07-14 14:48:19] POS_OF_PEAK=-33.97859602179 +/- 0.309973464409
[2015-07-14 14:48:19] F1RM001="783838909281 +/- 0.480839890366
[2015-07-14 14:48:19] CH12 = 4.03424348077e-05
  
```

Command Window

```

Command >
Add peak position to Plot3
  
```

```
>>> from gumpy.nexus import *
// create a 2D array, with dimension: 3x4
>>> arr = arange(0, 90.0, 90.0/12, [3, 4])
// multiply a random array
>>> arr *= rand([3, 4])
// print the array, with squares of errors and axes
>>> print arr
Dataset(Array([[0.000000, 3.578325, 13.678842, 1.89
                [17.238988, 2.261792, 13.606324, 40.
                [27.324902, 6.201477, 3.092464, 71.3
var=Array([[0.000000, 28.544689, 217.656676
            [527.075736, 84.953608, 616.3986
            [1651.938281, 419.169446, 232.06
axes=[SimpleData(Array([0, 1, 2])),
       SimpleData(Array([0, 1, 2, 3]))])
// do an integration along the 2nd dimension
>>> res = arr.intg(1)
>>> print res
Dataset(Array([19.151623, 73.382659, 108.005331]),
          var=Array([288.986143, 3373.792127, 8254.32
          axes=[SimpleData(Array([0, 1, 2]))])
// plot the result
>>> Plot1.set dataset(res)
```

```
// set UI label of the control panel
>>> __script__.title = 'Kowari Reduction'
// set the number of columns of the control panel
>>> __script__.numColumns = 2
// create a new group: a collection of parameters
>>> g_jump = Group('Jump to Scan Index')
>>> g_reg = Group('Region Selection')
4 // set the height of the group
2
2 >>> g_reg.rowspan = 3
8 >>> g_eff = Group('Efficiency Correction')
// set the number of columns in the group
2
2 >>> g_eff.numColumns = 2
// create a new parameter, giving type, initial
2 // value, and command to run when changing the value
>>> ind_jump = Par('int', -1, options = [],
                  command = 'jump_to_index()')
// set the UI label of the parameter
>>> ind_jump.title = 'select data index'
>>> var_jump = Par('float', float('NaN'), [],
                  command = 'jump_to_var()')
5 >>> var_jump.title = 'select scan variable'
// add parameters to a group
>>> g_jump.add(ind_jump, var_jump)
```

The screenshot displays the Kowari - Jython Runner web application. The interface includes a file browser on the left, a central control panel with tabs for 'Script Editor' and 'Auto GUI', and a right-hand display area with three plots. The 'Auto GUI' tab is active, showing settings for 'Kowari Reduction 1.0'. The 'Fitting' section displays a 'Gaussian Fit Plot' with parameters: peak position (42.530354), FWHM (0.5259631), and amplitude (1186.7094). The right-hand display area shows three plots: a 2D detector image, a 1D integration plot, and a 1D intensity plot. At the bottom, there is a 'Script Runner Status' bar showing 'IDLE' and buttons for 'Reload GUI' and 'Process Data'. A command window at the very bottom shows Jython script output and input commands.

Raw Data

- KWR0044002.nx.hdf
- KWR0044033.nx.hdf
- KWR0047821.nx.hdf
- KWR0047822.nx.hdf
- KWR0047823.nx.hdf
- KWR0047824.nx.hdf
- KWR0047825.nx.hdf
- KWR0049477.nx.hdf
- KWR0056560.nx.hdf
- KWR0056561.nx.hdf
- KWR0056562.nx.hdf
- KWR0056563.nx.hdf
- KWR0056564.nx.hdf
- KWR0056565.nx.hdf
- KWR0056566.nx.hdf
- KWR0056567.nx.hdf
- KWR0056568.nx.hdf

Script Editor **Auto GUI**

Kowari Reduction 1.0

Jump to Scan Index

select data index: 62

select scan variable: 309.99988

Region Selection

region enabled: ☐

region min X (deg): 0.0

region max X (deg): 180.0

region min Y (mm): -138.0

region max Y (mm): 138.0

Efficiency Correction

enabled: ☐ efficiency map: KWR0049477.nx.hdf

Geometry Correction

enabled: ☒

Fitting

min X: NaN max X: NaN

auto fit: ☒

Gaussian Fit Plot

peak position: 42.530354 fix: ☐

FWHM: 0.5259631 fix: ☐

amplitude: 1186.7094 fix: ☒

Script Runner Status **IDLE** **Reload GUI** **Process Data**

56560_62

Detector (mm)

Two Theta (degree)

56560_integration_62

Counts

Two Theta (degree)

56560_intensity

Intensity (a.u.)

phi (degrees)

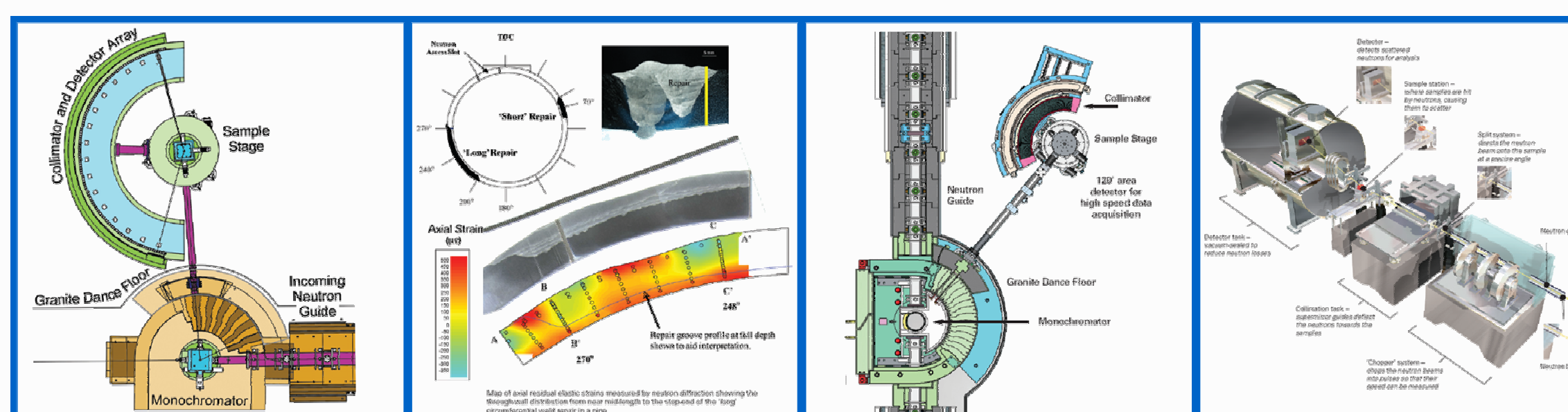
Command >>

```
[2015-07-14 15:45:14] POS_OF_PEAK = 42.6012812201 +/- 0.04999917221706
[2015-07-14 15:45:14] FWHM = 0.512249469757 +/- 0.100844858022
[2015-07-14 15:45:14] Quality = 0.0013976799245

>> move_amp(-1)
>> amp_fix.value=True;fix_amp()
>> fit_curve()

[2015-07-14 15:45:26] POS_OF_PEAK = 42.6167197509 +/- 0.0431822196166
[2015-07-14 15:45:26] FWHM = 0.525963127613 +/- 0.0834699023272
[2015-07-14 15:45:26] Quality = 0.0013957634236

>> move_mean(-1)
>> move_mean(-1)
>> move_mean(-1)
>> move_mean(-1)
```



Bragg Institute