A New Paradigm for Data Analysis Workflows

Emre Brookes[1], Joseph Curtis[2], Alexey Savelyev[1], David Wright[3], Hailiang Zhang[2], Paul Butler[4], Stephen Perkins[3], David Barlow[5], Jianhan Chen[6], Karen Edler[7], Thomas Irving[8], Susan Krueger[2], David Scott[9], Nicholas Terrill[10] & Stephen King[11]

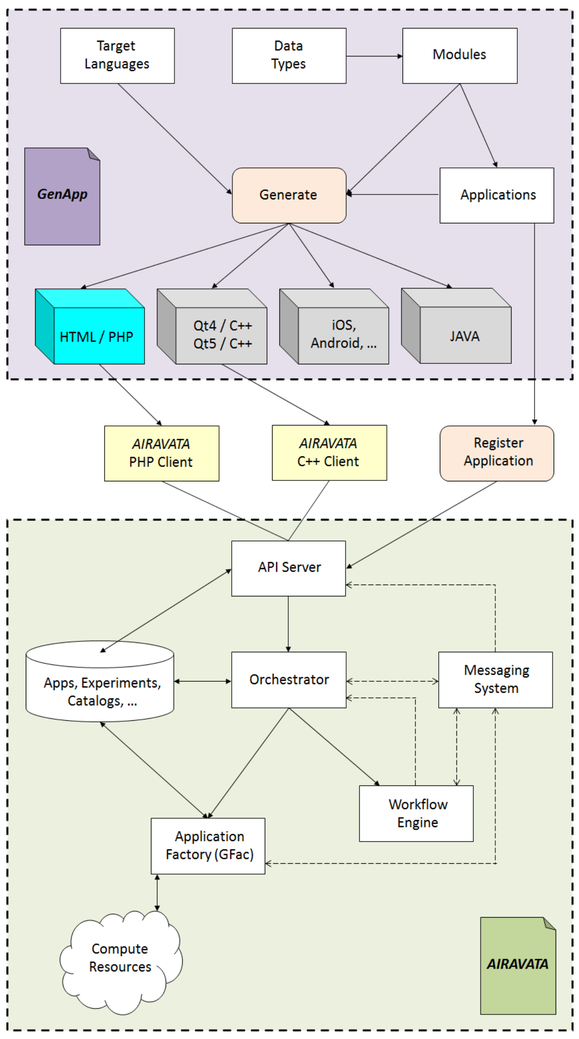
[1] Department of Biochemistry, University of Texas Health Science Center at San Antonio, San Antonio, TX 78229-3900, USA. [2] Centre for Neutron Research, National Institute of Standards and Technology, Gaithersburg, MD 20899-8562, USA. [3] Department of Structural & Molecular Biology, Darwin Building, University College London, Gower Street, London WC1E 6BT, UK. [4] Department of Chemistry, University of Tennessee, Knoxville, TN 37996-1600, USA. [5] Pharmacy Department, Franklin-Wilkins Building, King's College London, 150 Stamford Street, London SE1 9NH, UK. [6] Department of Biochemistry & Molecular Biophysics, Kansas State University, Manhattan, KS 66506, USA. [7] Department of Chemistry, University of Bath, Claverton Down, Bath, BA2 7AY, UK. [8] Department of Biology, Illinois Institute of Technology, 3101 S. Dearborn, Chicago, IL 60616, USA. [9] Research Complex at Harwell, STFC Rutherford Appleton Laboratory, Harwell Campus, Didcot, Oxfordshire, OX11 0FA, UK. [10] Diamond Light Source Ltd., Diamond House, Harwell Science & Innovation Campus, Chilton, Didcot, Oxfordshire, OX11 0DE, UK. [11] ISIS Pulsed Neutron & Muon Source, STFC Rutherford Appleton Laboratory, Harwell Campus, Didcot, Oxfordshire, OX11 0QX, UK. Email: stephen.king@stfc.ac.uk

The Problem

There are many data analysis packages for the different X-ray and neutron scattering techniques. These packages bring many challenges, most notably the challenges of: sustaining development *and support*, providing for *and maintaining* cross-platform deployment, utilising multi-processor compute resources as and when necessary, and all the time trying to keep things simple for the end user! How much better it would be if the end user could have the applications they need, in one place, running on a machine that is up to the job, within an environment that is both familiar and gives structure to their workflow?

A Solution?

Here we introduce ***GenApp***, a target-agnostic infrastructure for the creation and deployment of UIs for underlying executables which is also integrated with *Apache Airavata*, and illustrate its use to provide the ***SASSIE-Web*** framework for the constrained 'atomistic' modelling of macromolecular solution structures using SAXS/SANS and AUC data.



GenApp

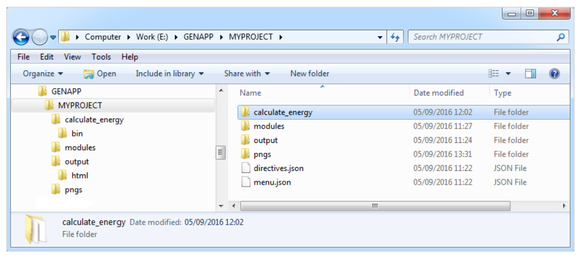
GenApp is an open extensible multi-target application generation tool for the simple and rapid deployment of multi-scale scientific codes.

An application is defined as a collection of executable modules which are then presented through a common user interface. This provides a powerful paradigm to combine both existing and new codes to perform novel workflows, or to develop new applications.

The addition of a module in *GenApp* is simple, and only requires the writing of a short JSON wrapper (a module) to detail the input and output, and the editing of two JSON files, one to specify where the module should appear in the applications menu system (*menu.json*), and the other to specify how the application itself is to be presented (*directives.json*). The modules themselves can be written in any supported language, independent of the choice of the target GUI implementation. Separating the scientific code from the GUI in this way not only facilitates the linking of component modules into larger workflows and applications, but also reduces the burden in supporting legacy codes.

Module executables either run locally (most GUI applications) or, if web-based, on a web server or other resource configured within *Apache Airavata*.

GenApp facilitates the creation of applications as web servers or gateways. This includes remote file management and the execution and management of lengthy non-interactive jobs. The latter capability, provided through integration with *Apache Airavata*, allows *GenApp* applications to harness a range of high-performance computing resources including local clusters, supercomputers, national grids, academic and commercial clouds. Instances of *GenApp* web applications have been tested on XSEDE and AWS.




```
# directives.json
# here generating instances of the application "calculate_energy" in
# three separate languages

{
  "title"           : "CALCULATE ENERGY",
  "application"     : "calculate_energy",
  "footer"          : "Powered by GenApp",
  "footersize"      : "50px",
  "version"         : "1.0",
  "languages"       : [ "html5", "qt4", "java" ],
  "executable_path" :
                      {
                       "html5" : "MYPROJECT/calculate_energy/bin",
                       "qt4"   : "MYPROJECT/calculate_energy/bin",
                       "java"  : "MYPROJECT/calculate_energy/bin"
                      }
# for more directives see
# http://gw105.iu.xsede.org/genapp/wiki/docs_directives
}
```

```
# menu.json
# providing each instance of the application "calculate_energy" with
# two menu options (to calculate the energy with Einstein's formula or
# with Planck's formula)

{
# "header" : "MY PROJECT",
   "menu" : [
       {
           "id"      : "calculate_energy",
           "label"   : "Calculate Energy",
           "icon"    : "MYPROJECT/pngs/myproject.png",
           "help"    : "calculate_energy help text",
           "modules" : [
               {
                   "id"    : "einstein",
                   "label" : "Einstein"
               },
               {
                   "id"    : "planck",
                   "label" : "Planck"
               }
           ]
       }
   ]
# for more menu options see
# http://gw105.iu.xsede.org/genapp/wiki/docs_menu
}
```

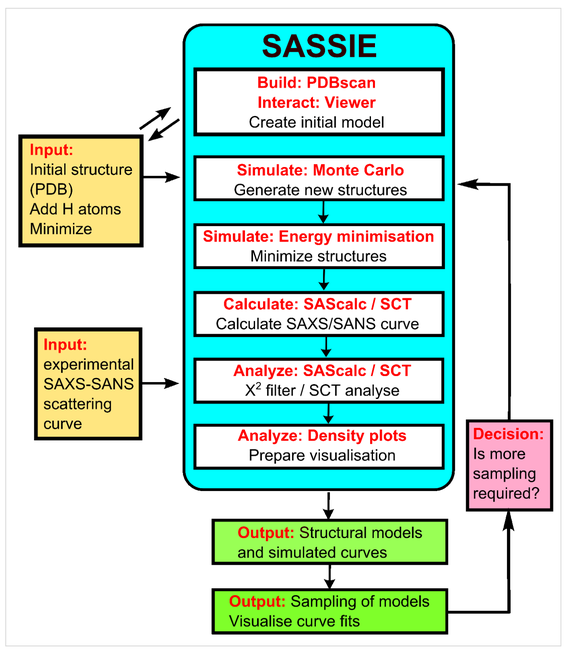
```
# einstein.json
# an example module

{
    "moduleid"    : "einstein"
    ,"label"      : "Einstein"
    ,"help"       : "help for Einstein"
    ,"executable" : "einstein"
    ,"uniquedir"  : "true"
    ,"fields"     : [
        {
            "role"        : "input"
            ,"id"         : "m"
            ,"label"      : "mass [kg]"
            ,"type"       : "float"
            ,"required"   : "true"
            ,"help"       : "Enter the mass in kg"
        }
        ,{
            "role"        : "input"
            ,"id"         : "c"
            ,"label"      : "Speed of light [m/s]"
            ,"type"       : "float"
            ,"default"    : 299792458
            ,"required"   : "true"
            ,"help"       : "Enter the speed of light in m/s"
        }
        ,{
            "role"        : "output"
            ,"id"         : "e"
            ,"label"      : "Energy [J]"
            ,"type"       : "text"
        }
    ]
}
# for more module options see
# http://gw105.iu.xsede.org/genapp/wiki/docs_module_fields
```

```
# einstein.py
#!/usr/bin/python

import json, sys, StringIO

if __name__=='__main__':

        argv_io_string = StringIO.StringIO(sys.argv[1])
        json_variables = json.load(argv_io_string)

        mass = float(json_variables['m'])
        speed_of_light = float(json_variables['c'])

        sys.path.append('./')
        import mass_energy

        output = {}
        output['e'] = mass_energy.einstein(mass,speed_of_light)

        print json.dumps(output)


def einstein(mass, speed_of_light):
    return mass*(speed_of_light**2.0)
```

We anticipate that *GenApp* will be useful to generate a wide-range of scientific applications beyond the scope covered by the CCP-SAS project. Interested parties should contact: genapp-devel@biochem.uthscsa.edu

Reference

Brookes, E.H., Anjum, N., Curtis, J.E., Marru, S., Singh, R. & Pierce, M. The GenApp framework integrated with Airavata for managed compute resource submissions. *Concurrency Comput. Pract. Exper.* (2015), **27**, 4292–4303.



The SASSIE-Web interface created by *GenApp*

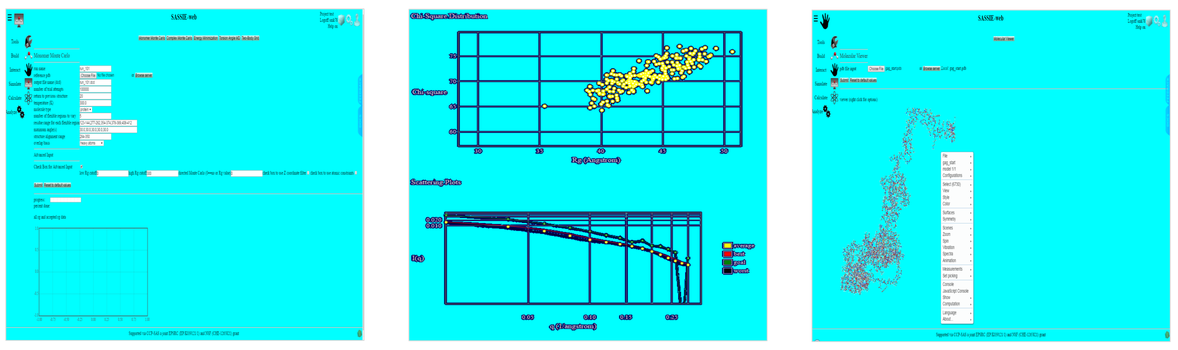
SASSIE-Web

The aim of *SASSIE-Web* is to allow experimentalists (especially novice end users) to construct modelling workflows from a set of sophisticated simulation and analysis modules and run them transparently on centrally-maintained HPC resources using nothing more than a web browser.

The provision of a web interface avoids the need for end users to install and maintain large complex software on their own machines, and remote execution accelerates the computationally expensive steps of the modelling process.

The *SASSIE-Web* menu organizes the analysis workflow in six sets of *GenApp* modules:

- **Tools**: which includes utilities to predict scattering length densities, interpolate experimental data files when required, and extract or merge macromolecular structures;
- **Build**: which includes utilities to check and correct PDB-formatted coordinate files;
- **Interact**: which provides the JSmol molecular viewer for interactive display of a specified structure;
- **Simulate**: which provides MC/MD programs to create representative ensembles of trial structures for testing against the data;
- **Calculate**: which provides a range of scattering curve calculators;
- **Analyze**: which identifies the simulated structures that best-fit the experimental scattering data and provides for their visualisation as envelopes.



(Above) *Screen shots from a SASSIE-Web session: (L-R) Setting up a molecular MC simulation to sample torsion angles; assessing the best-fit structures that are consistent with the experimental scattering data; visualising the structure of the HIV-1 Gag protein in JSmol.* ***(Below)*** *Results from SASSIE modelling: (L-R) Comparing the computed scattering from the best-fit structure with the experimental SAXS & SANS data; the resulting structure of IgA1 (Perkins et al., 2016)*

The modular design of the *SASSIE* framework not only gives the user the freedom to employ any combination of existing modules but also allows them to plug-in new modules and import coordinate models generated with other packages (eg, *AMBER*) at any stage of the workflow. This architecture makes *SASSIE-Web* an attractive option for other end users to contribute their codes. For example, the *Capriqorn* software - to calculate scattering curves from molecular simulations with explicit water models - is currently being integrated into the *SASSIE* framework (Köfinger & Hummer, 2013). And the *WillItFit* (Pedersen *et al*., 2013) and *QuaFit* (Spinozzi & Beltramini, 2012) packages have already been deployed for 'alpha' testing as web applications hosted on our CCP-SAS server. Anyone interested in contributing other relevant applications should contact: joseph.curtis@nist.gov

References

Curtis, J. E., Raghunandan, S., Nanda, H. & Krueger, S. SASSIE: A program to study intrinsically disordered biological molecules and macromolecular ensembles using experimental scattering restraints. *Comput. Phys. Commun.* (2012), **183**, 382-389.

Köfinger, J. & Hummer, G. Atomic-resolution structural information from scattering experiments on macromolecules in solution. *Phys. Rev. E* (2013), **87**, 052712.

Pedersen, M. C., Arleth, L. & Mortensen, K. WillItFit: a framework for fitting of constrained models to small-angle scattering data. *J. Appl. Crystallog.* (2013), **46**, 1894-1898.

Perkins, S. J., Wright, D. W., Zhang, H., Brookes, E. H., Chen, J., Irving, T. C., Krueger, S., Barlow, D. J., Edler, K. J., Scott, D. J., Terrill, N. J., King, S. M., Butler, P. D., Curtis, J. E. Atomistic modelling of scattering data in the Collaborative Computational Project for Small Angle Scattering (CCPSAS). (2016), *in review*.

Spinozzi, F. & Beltramini, M. QUAFIT: a novel method for the quaternary structure determination from small-angle scattering data. *Biophys. J.* (2012), **103**, 511-521.

www.ccpsas.org

This work has been supported by the Collaborative Computational Project for Small-Angle Scattering, a joint EPSRC (EP/K039121/1) and NSF (CHE-1265821) grant