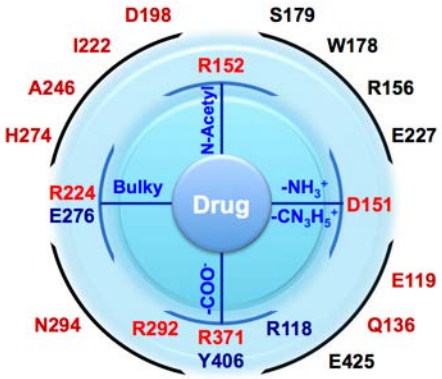
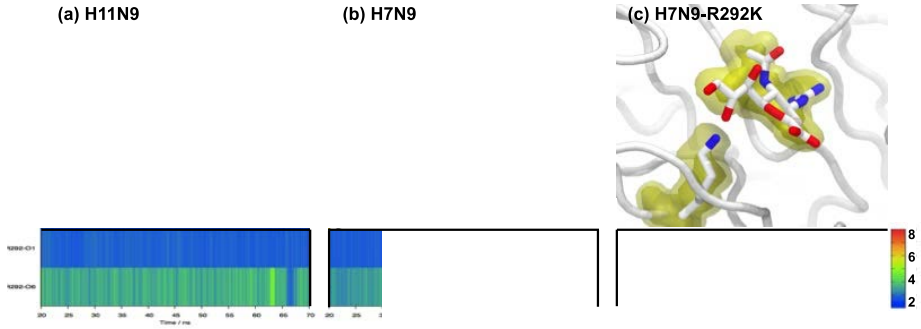
Christopher Woods – EPSRC Research Software Engineering Fellow
Data Intensive Research and Me

School of Chemistry
Software for rational inhibitor design



How does mutation affect binding?
Can we design inhibitors that are one step ahead of evolution?



Terabytes of simulation data…

Advanced Computing Research Centre (ACRC)
Research Software Engineering Group

```
if (q0[i][ii] != 0)
{
    const MultiFloat x(x0[i][ii]);
    const MultiFloat y(y0[i][ii]);
    const MultiFloat z(z0[i][ii]);
    const MultiFloat q(q0[i][ii]);

    if (eps0[i][ii] == 0)
    {
        //coulomb energy only
        for (int j=0; j<n1; ++j)
        {
            //calculate the distance between atoms
            tmp = x1[j] - x;
            r2 = tmp * tmp;
            tmp = y1[j] - y;
            r2.multiplyAdd(tmp, tmp);
            tmp = z1[j] - z;
            r2.multiplyAdd(tmp, tmp);

            soft_r = r2 + alfa;
            soft_r = soft_r.sqrt();

            one_over_soft_r = soft_r.reciprocal();

            //calculate the coulomb energy using shift-electrostatics
            // energy = q0q1 * { 1/r - 1/Rc + 1/Rc^2 [r - Rc] }
            tmp = soft_r - soft_Rc;
```

```
from multiprocessing import Pool, current_process
import contextlib
import time

def slow_sum( nsecs, x, y ):
    """Function that sleeps for 'nsecs' seconds, and
       then returns the sum of x and y"""
    print("Process %s going to sleep for %d second(s)" \
          % (current_process().pid,nsecs))

    time.sleep(nsecs)

    print("Process %s waking up" % current_process().pid)

    return x+y

if __name__ == "__main__":
    print("Master process is PID %d" % current_process().pid)

    with contextlib.closing( Pool() ) as pool:
        r1 = pool.apply_async( slow_sum, [1,6,7] )
        r2 = pool.apply_async( slow_sum, [1,2,3] )

        r1.wait()
        print("Result one is %s" % r1.get())

        r2.wait()
        print("Result two is %s" % r2.get())
```

Developing efficient software

- Reusable
- Reliable
- Flexible

How to manage data flow in parallel programs?
How to create useful abstractions for complex data?