Project JUNIOR: Drug Discovery using Azure
April 6-7, 2010
Redmond, Washington
The Team

• Computing Science
  – Paul Watson, Hugo Hiden, Simon Woodman, Jacek Cala

• Northern Institute for Cancer Research
  – David Leahy, Dominic Searson, Valdimir Sykora

• Microsoft
  – Christophe Poulain
The problem...

What are the properties of this molecule?

Toxicity

Biological Activity

Solubility

Perform experiments

Time consuming

Expensive

Ethical constraints
the alternative to experiments

Predict likely properties based on similar molecules

CHEMBL Database: data on 622,824 compounds, collected from 33,956 publications

WOMBAT Database: data on 251,560 structures, for over 1,966 targets

WOMBAT-PK Database: data on 1230 compounds, for over 13,000 clinical measurements

All these databases contain structure information and numerical activity data

What is the relationship between structure and activity?
Quantitative Structure Activity Relationship

Activity \approx f(\text{logP, number of atoms, shape, ...})

More accurately, Activity related to a quantifiable structural attribute

Currently > 3,000 recognised attributes

http://www.qsarworld.com/
JUNIOR Project

Aim:

To generate models for as much of the freely available data as possible and make it available on www.openqsar.com

... so that researchers can generate predictions for their own own molecules
Using QSAR

QSAR is a *Regression* exercise

Build Models ▶ Validate Models ▶ Predict activities
Using QSAR

QSAR is a *Regression* exercise

- **Build Models**
- **Validate Models**
- **Predict activities**

Select a “training set” of compounds

Which compounds?

Select descriptors most related to activity

Which descriptors?

Calculate descriptor values

Which model?

<table>
<thead>
<tr>
<th>Compounds</th>
<th>Descriptor Values – logP, shape...</th>
<th>Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Compound 1]</td>
<td>102</td>
<td>0.1</td>
</tr>
<tr>
<td>![Compound 2]</td>
<td>242</td>
<td>0.3</td>
</tr>
<tr>
<td>![Compound 3]</td>
<td>109</td>
<td>0.6</td>
</tr>
<tr>
<td>![Compound 4]</td>
<td>312</td>
<td>0.2</td>
</tr>
<tr>
<td>![Compound 5]</td>
<td>194</td>
<td>0.5</td>
</tr>
</tbody>
</table>

\[ y = mx + c \]
Using QSAR

Select a new set of compounds **not** used during model building

Calculate **same** set of descriptors

Use model to estimate activities

Compare the **estimated** activities with the **actual** activities

<table>
<thead>
<tr>
<th>Compounds</th>
<th>Descriptors</th>
<th>Estimate</th>
<th>Actual</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="https://via.placeholder.com/150" alt="Compound 1" /></td>
<td>102</td>
<td>10.4</td>
<td>4.7</td>
<td>4.6</td>
</tr>
<tr>
<td><img src="https://via.placeholder.com/150" alt="Compound 2" /></td>
<td>242</td>
<td>20.9</td>
<td>5.7</td>
<td>5.9</td>
</tr>
</tbody>
</table>

Keep model if error acceptable...

*what measure?*
Using QSAR

Build Models  Validate Models  Predict activities

Use the model to estimate activities for new compounds

\[ f(212, 0.9, 9.8) = 5.7 \]

Use  
Discard
QSAR choices

- QSAR Process requires many choices
  - Which descriptors?
  - Which modelling algorithm?
  - What model testing strategy?

- Quality of result depends on make correct choices
  - All runs are different

- Discovery Bus manages this process
  - Apply everything approach
The Discovery Bus

Manages the many model generation paths

1. Partition training & test data
2. Calculate descriptors
3. Select descriptors
4. Build model
5. Add to database

- Random split
- 80:20 split
- Java CDK descriptors
- C++ CDL descriptors
- Correlation analysis
- Genetic algorithms
- Random selection
- Linear regression
- Neural Network
- Partial Least Squares
- Classification Trees
The Discovery Bus

- Legacy system
  - Uses Oracle stored procedures and shell scripts
  - Models built using R
  - Designed to scale using agents on multiple machines
- Hard to use and maintain
  - Undocumented
  - Specific library and OS versions
- Runs on Amazon VMs
- AIM: Extend Discovery Bus using Azure
  - New agent types
  - Make use of more computing resources
- Move towards more maintainable system
Although unwieldy, Discovery Bus performs useful work.
Using Cloud to extend Discovery Bus

- Move computationally intensive tasks to Cloud
  - Descriptor calculation
  - Descriptor selection
  - Model building
- Keep Discovery Bus for management
- Apply this system to CHEMBL and WOMBAT databases
- MOAQ
  - Mother Of All QSAR
  - Model everything in the databases in one shot
Extending Discovery Bus

Bus Master

Planner

“Proxy” Agent

Calculation Agents

e-Science Central Workflows

Agent code executes on multiple Azure nodes

Windows Azure

Java

R

S
Initial results

Azure CPU Utilisation:

2 Workers ~70 Mins

10 Workers ~20 Mins

15 Workers ~15 Mins
An excellent result?

No 😞
Initial problems

Scales to \(~20\) worker nodes  
Want scalability to at least \(100\)

Improves calculation time, but still takes too long for a run:

Model validation and admin tasks form a long tail
Analysis of problems

Not sending enough work fast enough to Azure

Discovery Bus?

Agent code executes on multiple Azure nodes

Architecture?

Tops out at over 400 concurrent workflows

Bus Master

Calculation Agents

"Proxy" Agent

e-Science Central Workflows

Windows Azure

Add more admin capable agents

No improvement
Discovery Bus optimisation

Discovery Bus?

Bus Master

Optimise Amazon VM configuration

- Standard VM
- EBS database
- Custom file transfer
- High CPU VM
- Local disk database
- NFS file transfer

Significant gains

Planner optimisation

- Takes ~1 second to plan a request to Azure
- Feature / limitation of Discovery Bus – always “active”
- Multiple planning agents? Discovery Spec: Not allowed & doesn’t work
- Flatten / simplify plan structure

20 Nodes → 40 - 50 Nodes
Updated configuration

• Current setup scales to 50 nodes
  – Run two parallel Discovery Bus instances
  – Feeds 100 Azure nodes
• Moving more of the Admin tasks to Azure / e-Science Central
  – Move model validation out of Discovery Bus
  – More co-ordination outside Discovery Bus
• Work in progress
  – Azure utilisation increasing (average ~60% over entire run)
Lessons learned

• Moving to the Cloud exposes architectural weaknesses
  – Not always where expected
  – Stress test individual components in isolation

• Good utilisation needs the right kind of jobs
  – Computationally expensive
  – Modest data transfer sizes
  – Easy to run in parallel

• Be prepared to modify architecture

• However
  – We are dramatically shortening calculation time
  – We do have a system that can expand on demand
Future work

• Finish runs on 100 nodes using 2 Discovery Busses
  – Publish results on www.openqsar.com

• Move more code away from Discovery Bus
  – Many tasks already on Azure
  – Fixing planning issue will be complex
  – Move planning to e-Science Central / Azure

• Generic model building / management environment in the Cloud
Microsoft External Research

Project funders for 2009 / 2010

Roger Barga, Savas Parastatidis, Tony Hey
Paul Appleby, Mark Holmes
Christophe Poulain