Nextflow for chemistry - crossing the divide

Tim Dudgeon - Informatics Matters
tdudgeon@informaticsmatters.com

Topics

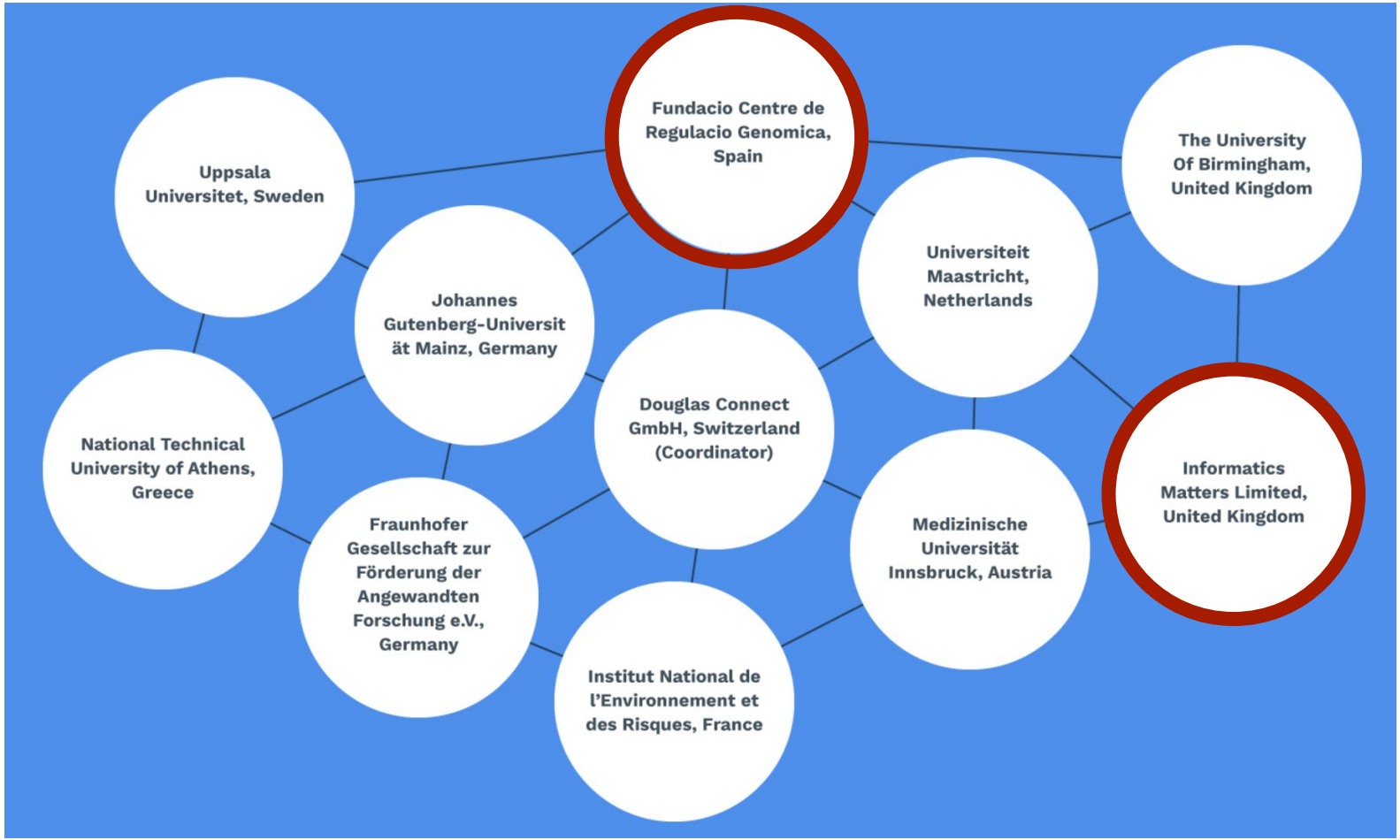
- About the OpenRiskNet project
- About Squonk
- Nextflow in Squonk

***OpenRiskNet**: Open e-Infrastructure to Support Data Sharing, Knowledge Integration and in silico Analysis and Modelling in Risk Assessment*

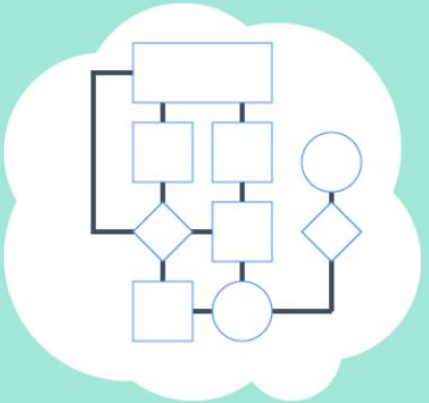
- Funded under Horizon 2020 EINFRA-22-2016 programme
- 3 year project, 11 partners across Europe
- Provides e-infrastructure for handling various aspects of chemical safety assessment
- Aims to standardise ways to access data and run modelling workflows
- Target communities: chemicals, cosmetic ingredients, therapeutic agents, nanomaterials

OpenRiskNet is a 3 year project funded by the European Commission within Horizon2020 EINFRA-22-2016 Programme (Grant Agreement 731075; start date 1 December 2016)

This project is funded by the European Union


Uppsala Universitet, Sweden
Fundacio Centre de Regulacio Genomica, Spain
The University Of Birmingham, United Kingdom
Johannes Gutenberg-Universität Mainz, Germany
Universiteit Maastricht, Netherlands
National Technical University of Athens, Greece
Douglas Connect GmbH, Switzerland (Coordinator)
Informatics Matters Limited, United Kingdom
Fraunhofer Gesellschaft zur Förderung der Angewandten Forschung e.V., Germany
Medizinische Universität Innsbruck, Austria
Institut National de l'Environnement et des Risques, France

How?



Easily accessible
Standardised
Harmonised
Scalable
Robust
Infrastructure

For whom?

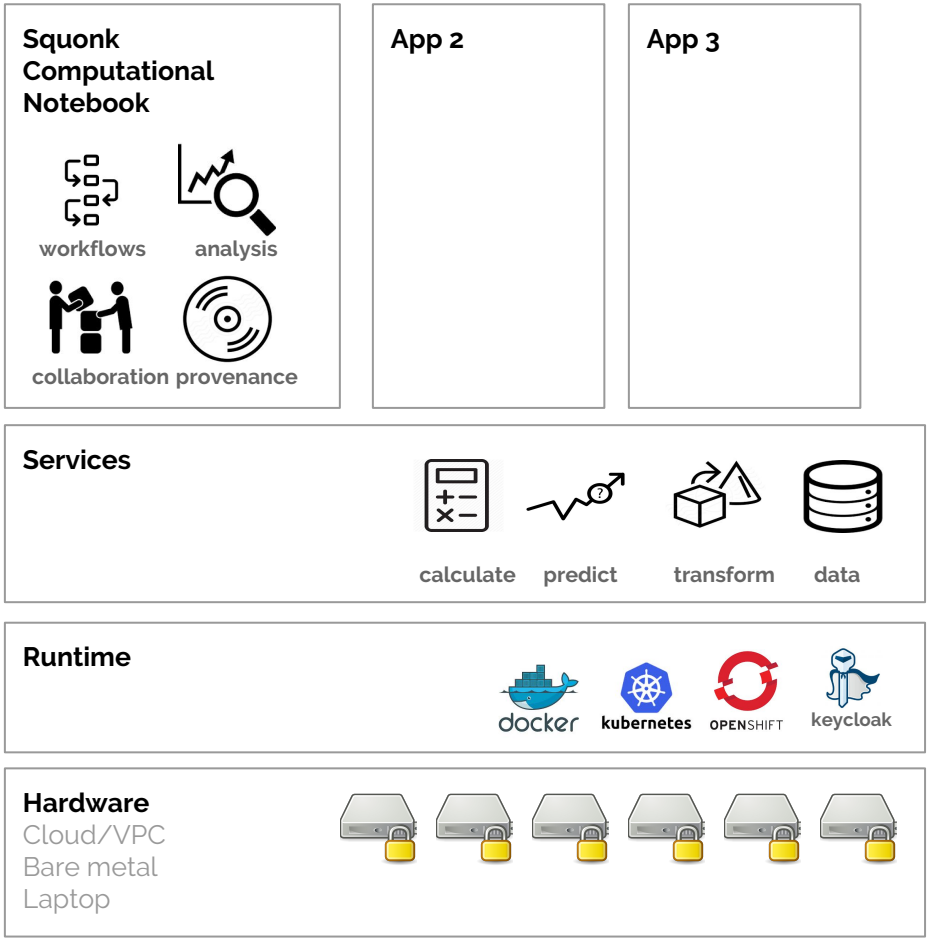
Researchers
Risk assessors
Regulators
Informed public

To what end?

Improve industrial risk assessments
Prototyping new services and apps
Enabled access to integrated resources
Complete and qualified system
Support inovative product development

Develop + Deploy Integrated, Secure, and Sustainable e-Infrastructure

Virtual Research Environments



Vendor agnostic
Commercial + Open Source
Interoperable
Pluggable
Containerised

Performant
Scaleable
Resilient
Flexible
Secure
Cost effective
Cloud enabled

Types of OpenRiskNet Application

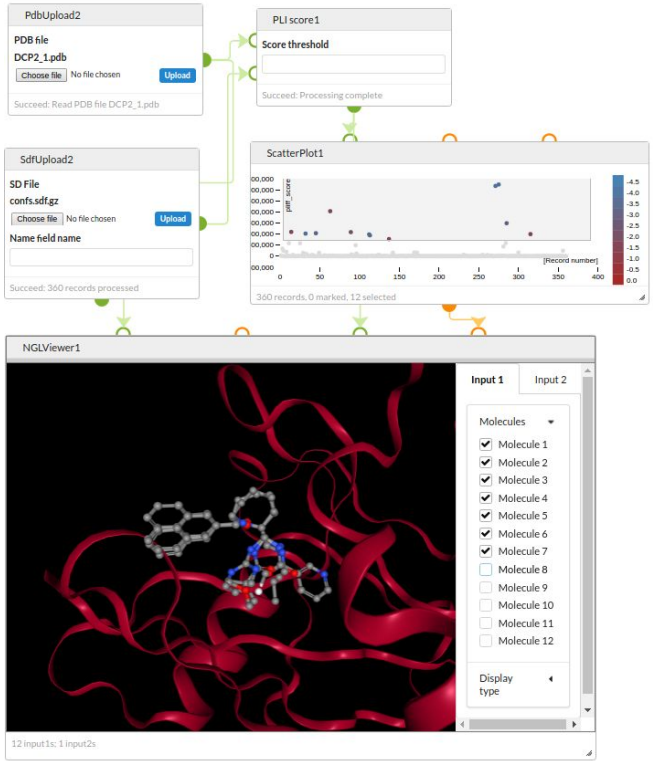
- QSAR models for tox & metabolism
- Analysis of ‘omics data
- PK/PD predictions
- Dataset curation & annotation
- Search & Discovery

=> Combination of Biology and Chemistry
=> Frequently involve multi-step workflows
=> Some aspects require HPC

=> Enter Nextflow

Squonk Computational Notebook

- Browser based & simple to use
- Targeted at scientists not geeks
- Allows to create and execute workflows
- Allows to analyse and visualise
- Provides reproducibility and traceability
- Facilitates collaboration



https://github.com/InformaticsMatters/squonk

Squonk Workflows

- Currently focussed on cheminformatics and comp chem ...
- … but want to incorporate biology and ‘omics
- Partly driven by Fragment Based Lead Discovery activities at Diamond Light Source
- Examples:
  - Virtual screening (ligand and target based)
  - Library enumeration
  - Chemical database search
  - Compound profiling

Squonk Demo

Squonk Services

- Extensible mechanism for plugging in computational services
  - Execute Docker container
  - Execute Nextflow workflow
- Step 1: implement the algorithm/workflow
- Step 2: write a deployment descriptor
- Step 3: deploy to Squonk

Example upstream projects that provide these services

- Pipelines: https://github.com/InformaticsMatters/pipelines
- Docking validation: https://github.com/InformaticsMatters/docking-validation

Example: rDock

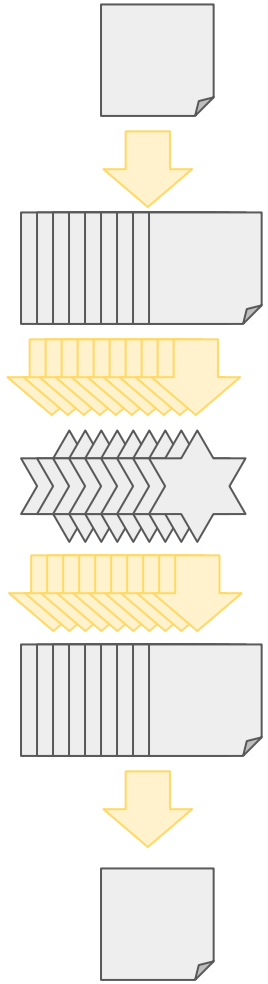
- Performs molecular docking (http://rdock.sourceforge.net/)
- Typical use is for virtual screening e.g. screening 10,000 candidate compounds for binding to protein active site
- Computationally demanding
- Algorithm is single threaded but screening can easily be parallelised

=> Enter Nextflow

Virtual screening using rDock

rDock setup

Protein structure
Cavity definition
Docking parameters



SD file with 10,000 compounds

Split into chunks of 200 compounds

Execute rDock

Docking results

Collect results

Params

```
params.ligands = "$baseDir/ligands.data.gz"
params.receptor = "$baseDir/config.zip"
params.chunk = 25
params.num_dockings = 100
params.top = 1
params.score = null
params.nscore = null
params.limit = 0
params.digits = 4


ligands = file(params.ligands)
receptorzip = file(params.receptor)
```

Prepare config

```
process unzip_config {

    container 'informaticsmatters/rdkit_pipelines'

    input:
    file receptorzip

    output:
    file 'receptor.prm' into prmfile
    file 'receptor.mol2' into protein
    file 'receptor.as' into asfile

    """
    unzip $receptorzip
    """
}
```

Split into parts

```
process splitter {

    container 'informaticsmatters/rdkit_pipelines'

    input:
    file ligands

    output:
    file 'ligands_part*.sdf' into ligands_parts mode flatten
    file 'ligands_part_metrics.txt' into splitter_metrics

    """
    python -m pipelines.rdkit.filter -i $ligands -c $params.chunk -l $params.limit -d
$params.digits -o ligands_part -of sdf --no-gzip --meta
    """
}
```

Run rDock

```
process rdock {

    container 'informaticsmatters/rdock'

    input:
    file part from ligands_parts
    file prmfile
                file protein
                file asfile

    output:
    file 'docked_part*.sd' into docked_parts

    """
    rbdock -r $prmfile -p dock.prm -n $params.num_dockings -i $part -o
${part.name.replace('ligands', 'docked')[0..-5]} > docked_out.log
    """
}
```

Process & collect results

```
process results {

                container 'informaticsmatters/rdock'

                input:
                file ligands
                file part from docked_parts.collect()

                output:
                file 'results.sdf' into results

                """
                sdsort -n -s -fSCORE docked_part*.sd |${params.score == null ? ''
: " sdfilter -f'\$SCORE <= $params.score' |"}${params.nscore == null ? '' : "
sdfilter -f'\$SCORE.norm <= $params.nscore' |"} sdfilter -f'\$_COUNT <=
${params.top}' > results.sdf
                """
}
```

Generate metrics

```
process metrics {

    container 'informaticsmatters/rdkit_pipelines'

    publishDir baseDir

    input:
    file 'results.sdf' from results
    file 'splitter_metrics.txt' from splitter_metrics

    output:
    file 'output.data'
    file 'output.metadata'
    file 'output_metrics.txt'

    """
    python -m pipelines.rdkit.filter -i results.sdf -of json -o output --meta
    mv output_metrics.txt old_metrics.txt
    grep '__InputCount__' splitter_metrics.txt | sed
s/__InputCount__/DockingRDock/ > output_metrics.txt
    grep '__InputCount__' splitter_metrics.txt >> output_metrics.txt
    grep '__OutputCount__' old_metrics.txt >> output_metrics.txt
    """
}
```

https://github.com/InformaticsMatters/pipelines/blob/master/src/nextflow/docking/rdock.nsd.nf

See also workflow for ligand based virtual screening: https://github.com/InformaticsMatters/pipelines/blob/master/src/nextflow/rdkit/screen-multi-dataset.nsd.nf

Why Nextflow?

- CRG and IM are OpenRiskNet partners
- Simple to define multi-step workflows
- Makes parallelisation trivial
  - Currently just on a single server according to number of available cores
  - But plan to parallelise across multiple servers
- We already spoke fluent Java and Groovy
- Good support for Docker
- Transparency of implementation
  - Should be simple to switch from Docker to Singularity
  - Should be simple to run on Kubernetes or HPC environments
- Should be able to leverage existing bioinformatics and ‘omics workflows
- Really nice implementation (Thank you Paolo!)

What Next for Nextflow in Squonk

- We are still novices - eager to learn!
- Investigate security implications
  - Important in a multi-tenant environment where you can’t trust what a user might deploy
  - Want to learn more about Singularity
- Implement more chemistry related workflows
- Investigate application areas that cross the chemistry-biology divide
- Want to investigate how best to parallelise across multiple servers
  - Kubernetes seems the obvious choice, but how to do this and is this the right approach?
  - Hackathon session on this - please join if interested

Acknowledgements

- OpenRiskNet partners
- EU for funding
- XChem project at Diamond Light Source (Anthony Bradley & Frank van Delft)
- Paolo for such a great tool and for answering all my stupid questions