OpenRiskNet

RISK ASSESSMENT E-INFRASTRUCTURE

Deliverable Report D2.5

Compute and data federation

This project is funded by
the European Union

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

Project Number 731075

www.openrisknet.org

Project identification

Grant Agreement	731075
Project Name	OpenRiskNet: Open e-Infrastructure to Support Data Sharing, Knowledge Integration and *in silico* Analysis and Modelling in Risk Assessment
Project Acronym	OpenRiskNet
Project Coordinator	Edelweiss Connect GmbH (previously Douglas Connect GmbH)
Start date	1 December 2016
End date	30 November 2019
Duration	36 Months
Project Partners	P1 Edelweiss Connect GmbH Switzerland (EWC) P2 Johannes Gutenberg-Universität Mainz, Germany (JGU) P3 Fundacio Centre De Regulacio Genomica, Spain (CRG) P4 Universiteit Maastricht, Netherlands (UM) P5 The University Of Birmingham, United Kingdom (UoB) P6 National Technical University Of Athens, Greece (NTUA) P7 Fraunhofer Gesellschaft Zur Foerderung Der Angewandten Forschung E.V., Germany (Fraunhofer) P8 Uppsala Universitet, Sweden (UU) P9 Medizinische Universität Innsbruck, Austria (MUI) P10 Informatics Matters Limited, United Kingdom (IM) P11 Institut National De L’environnement Et Des Risques, France (INERIS) P12 Vrije Universiteit Amsterdam, Netherlands (VU)

Deliverable Report identification

Document ID and title	Deliverable 2.5 Compute and data federation
Deliverable Type	Demonstrator
Dissemination Level	Public (PU)
Work Package	WP2
Task(s)	Task 2.7, 2.8
Deliverable lead partner	CRG
Author(s)	Evan Floden, Audald Lloret-Villas, Paolo Di Tommaso (CRG), Ola Spjuth (UU), Lucian Farcal (EwC), Tim Dudgeon (IM), Danyel Jennen (UM)
Status	Final
Version	V1.0
Document history	2019-04-26 First draft 2019-05-17 Consolidated draft 2019-06-11 Final version

Table of Contents

SUMMARY

This report details the work involved in the federation of compute and data resources between the OpenRiskNet e-infrastructure and external resources. The reference environment has been designed to be capable of handling the majority of requirements for users' wishes to deploy and run services. However specific situations demand solutions where either the computation, the data or both reside outside the OpenRiskNet e-infrastructure. This deliverable is related to Tasks 2.7 (Interconnecting virtual environment with external infrastructures) and Tasks 2.8 (Federation between virtual environments).

Resource intensive analyses, such as those performed in toxicogenomics, can have CPU, memory or disk requirements that cannot be assumed to be available across all deployment scenarios. Human sequencing data may have restrictions on where it can be processed and the vast quantity of this data often predicates that it is more efficient to "bring the computation to the data". In achieving Tasks 2.7 and 2.8, we can demonstrate how the virtual environment can utilise external infrastructure including commercial cloud providers and data stores.

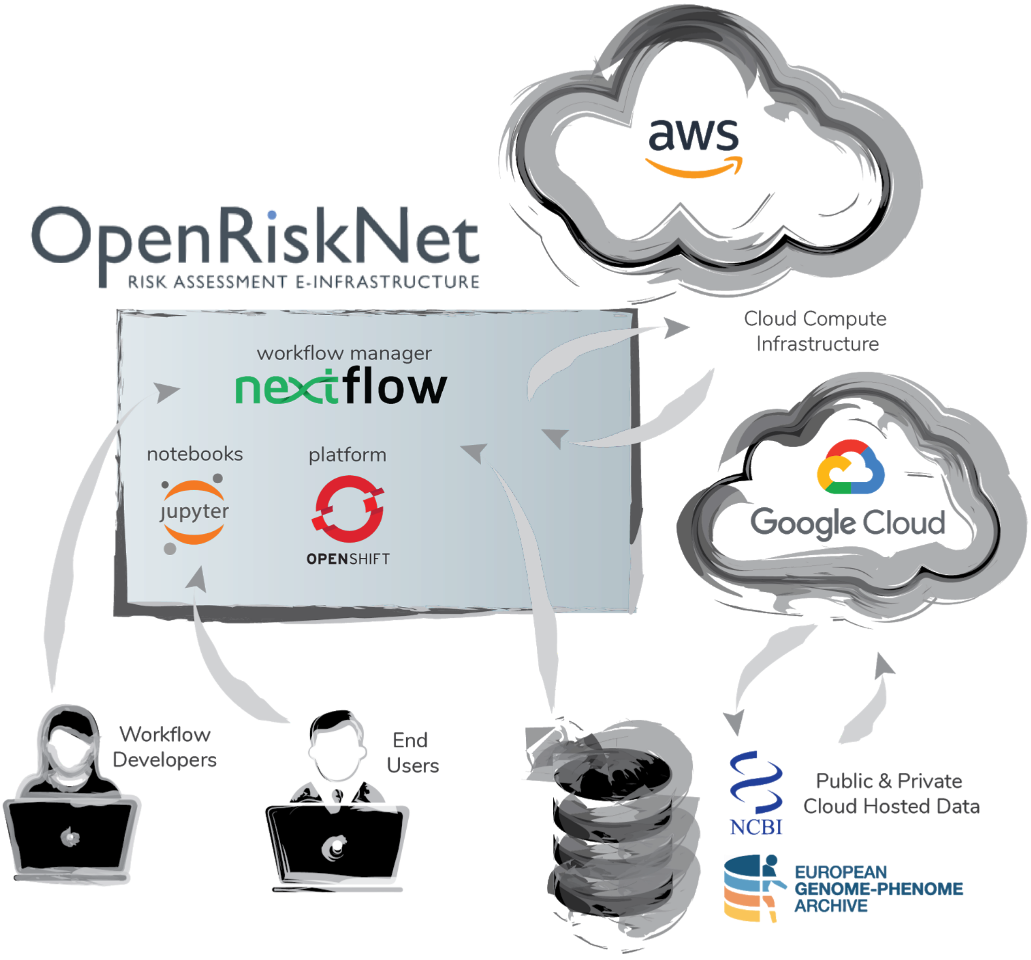

Figure 1. Schematic outlining the federation of computing and data resources between the OpenRiskNet virtual e-infrastructure and public computational and data resources.

INTRODUCTION

The OpenRiskNet Consortium develops the OpenRiskNet e-infrastructure for the harmonisation and improved interoperability of data and software tools in the area of predictive toxicology and risk assessment. It aims to combine interoperable web services providing data or analysis, processing and modelling tools communicating over well-defined and harmonised application programming interfaces (APIs), supplemented by a semantic interoperability layer added to every service to describe the functionality whilst guaranteeing the technical and semantic interoperability. The federation of compute and data resources refers to the ability of distinct, formally disconnected environments to be able to be interoperable via communication and the sharing of resources. With respect to the OpenRiskNet e-infrastructure, this aims to enable federation between the OpenRiskNet e-infrastructure and external resources, as well as with other instances of the e-infrastructure.

Available Computational Resources

Within predictive toxicology and risk assessment, techniques are increasingly being applied which rely upon datasets of extreme size, by what is often termed “Big Data”. This data originates from the aggregation of existing datasets and the systematic robotised testing of materials in a high-throughput manner. Yet it is the introduction of “omics” analysis that presents the biggest opportunity and challenge to safety science. Omics is the umbrella term for the biological fields of study with the -omics suffix, such as genomics, metabolomics and proteomics. In predictive toxicology and risk assessment, a single transcriptomic sample may consist of one hundred million reads, equating to 30 GB of raw data.

Of all the applications on these large datasets, read-across has been most widely adopted. In a read-across analysis, the data from one or more well-characterised substances (the source) are used to infer the characteristics of a relatively unknown substance (the target). When the properties of the substances are considered similar enough, the data can form the basis of a safety assessment. Combining this approach with machine-learning techniques holds great promise for the future of toxicology testing. In 2018, a technique was developed using the largest machine-readable chemical database (RASAR database) to create a model of read-across structure relationship activity. Using this model as a definition of chemical similarity, the derived feature vectors could be used for supervised learning predictions which were on average 87% accurate across nine common toxicity tests compared to 81% accuracy in the animal tests themselves [1]. Studies like this highlight the potential for Big Data approaches to eliminate the need for costly and ethically questionable animal testing.

The RASAR database described above contains over 80 thousand chemicals which alone requires 3.2 x 10^9 pairwise comparisons. The millions of structures in PubChem would require approximately 10^{14} comparisons. At this scale, the use of single computers for analysis becomes untenable. Traditionally, such analysis has relied upon expensive, on-premise distributed computing systems using batch queuing schedulers. Cloud computing has democratised distributed computing. Cloud provides unlimited resource elasticity with zero capital investment and an effective cost of zero when not in use.

Cloud-bursting offers a hybrid approach where specific tasks are computed on cloud resources when the required resources are unavailable locally. In achieving Task 2.7, we demonstrate the use of external cloud resources directed from applications and services running on the OpenRiskNet e-infrastructure. This enables users and developers of OpenRiskNet e-infrastructure to gain the benefits of cloud resource elasticity.

Data Localisation - Moving Computation to the Data

The requirement for portable computation is not only driven by the availability of CPU or memory resources. The increasing quantities of data have resulted in data localisation becoming a real concern. If we consider two infrastructures having identical resources, with a large dataset on one infrastructure and application code on the other, it is self-evident that the application code should move to the infrastructure where the data resides. There is a real cost associated with moving big data, in terms of network congestion, the time required to move files and economic cost. Data transfer charges for the most popular cloud storage service AWS Simple Storage Service (S3) equate to approximately 0.1€ per GB. A 40 terabyte analysis could result in 8,000€ in ingress and egress charges alone.

More recently, large public datasets are being hosted on public cloud infrastructure. In the case of Google Cloud Genomics Public Datasets and AWS iGenomes, the intention is to attract customers to perform computation on the cloud. In 2018 the National Institute of Health (NIH) created the STRIDES initiatives to tackle the main challenges of large datasets being disconnected, incompatible, difficult to find and expensive to generate, store, move and compute on. STRIDES establishes partnerships with commercial cloud service providers to harness the power of the commercial cloud in support of research and echoes the FAIR data principles of findability, accessibility, interoperability, and reusability. In this new world where computation is performed at the location of the data, portability of computation becomes a key requirement.

Location Sensitive Data and Hybrid Deployment

Not all workloads and services are amenable to public cloud computation. Sensitive data may be subject to restrictions. The impact of general data protection regulation (GDPR) regulation is starting to be seen in the specific health data definitions and requirements. In the United States, the Health Insurance Portability and Accountability Act of 1996 (HIPAA) legislation ensures data privacy and security provisions for medical information and places restrictions with the onus on service providers to prove their infrastructure is HIPAA compliant. Other jurisdictions restrict the export of personal health information from one country to another. In these instances, some tasks may be computed locally, whilst others can be computed on external infrastructure. This hybrid-cloud approach is increasingly common and e-infrastructure systems must therefore provide a flexible manner for allowing such computation.

Data Security, Data Control and Authentication

With the use of external resources comes additional concerns for data security, data control and authentication. If data (or metadata) leaves one environment to be computed on another concerns are raised. This is particularly true for human genomic information. Whilst we did not use any identifiable genomic data within the OpenRiskNet case studies, it was important to factor the possibility of such data in future risk assessment use cases. For this we engaged with three organisations who are tackling these open challenges.

The Global Alliance for Genomics and Health (GA4GH) is a policy-framing and technical standards-setting organization, seeking to enable responsible genomic data sharing within a human rights framework [2]. Sensitive data protection is, among others, a cornerstone for this collective effort. Different Work Streams of the GA4GH are developing standards that combine the optimisation of data transference and analysis while keeping the security of all the actions. Just to mention an example, the “Data Use & Researcher Identities” Work Stream is leading the implementation of researcher identities and data use restrictions for a personalised and appropriate access to controlled data.

In a similar direction, ELIXIR, an intergovernmental organisation that brings together life science resources from across Europe, is coordinating databases, software tools, training materials, cloud storage and supercomputers to generate a single infrastructure. A joint effort from ELIXIR and GA4GH is the ELIXIR Authentication and Authorisation Infrastructure (AAI). In a nutshell, log-in systems implementing ELIXIR AAI guarantee a secure, atomic and appropriate access to sensitive datasets.

The European Genome-phenome Archive (EGA) is a resource jointly coordinated from the Centre for Genomics Regulation (CRG) and the European Bioinformatics Institute (EBI) that offers long-term and secure storage for biomedical data that requires controlled access. Human-derived datasets, e.g. toxicogenomics data, can be securely archived to the EGA with no charges. Leveraging technologies and standards from GA4GH and ELIXIR, where EGA is actively participating and works on implementing their recommendations, these datasets are totally encrypted and properly tagged with Data Use Ontologies (DUO). Data encryption strengthens data protection but adds additional challenges for processing the data. DUO terms, on the other hand, are controlled vocabulary that indicate data access conditions and restrictions for an easy identification of suitable datasets. Nevertheless, these datasets can only be accessed upon explicit authorisation from the relevant Data Access Committee (DAC). The DACs govern the data access through standard process of data access application. Work is ongoing to ensure that as these standards become adopted that become integrated with the risk assessment VE.

Workflows as a Means of Portable Computation

Workflow managers allow scientists, engineers and developers to write code, manage application dependencies, and then run the resulting data analysis workflows. Scientific analyses typically consist of a “mash-up” of different tools and custom scripts which are wrapped up together with the steps lending themselves to embarrassing parallelization, where the input data can be split and the same task can be repeated as many times as the data as split.

Historically, scientific applications are difficult to install, configure and deploy. The problem is exacerbated when considering the different dependencies and libraries that make up any given risk assessment service. Owing to this, an application developed in one environment often has little chance of running correctly in another environment without significant time and effort in adaptations. Previously, virtualization technology known as virtual machines (VMs) have been proposed, however this approach has some significant disadvantages. VM images are large (gigabytes) as they contain a full copy of the operating system (OS) and they can have a significant startup time which limits their suitability for short-lived web-services or big data applications running hundreds of thousands of tasks. Containerisation provides an alternative to VMs and was popularised with Docker beginning in 2013 as a way to run short running services on the web. In scientific computing, the most obvious advantage of containers is the ability to download a single pre-built ready-to-run image containing all the required software. A container often requires as little as a few hundreds of milliseconds to startup and many instances of the same container can be executed on the same hosting environment with a minimal memory and performance footprint. Another major advantage of containers is that each task runs an isolated container which prevents conflicts and enables tasks to be orchestrated across distributed infrastructure using workflow managers.

When analyses are split across ten, hundreds or thousands of tasks, it is the job of the workflow manager to manage the orchestration of the tasks. A given workflow can be represented as a directed acyclic graph where the dependencies of a downstream task are defined by the outputs of upstream tasks. When represented in this manner, the computation can be distributed across infrastructure with each task submitted as an isolated individual task with an appropriate container. Common scientific workflow managers include Galaxy [3], Nextflow [4], Snakemake [5] and Cromwell. In OpenRiskNet, we chose Nextflow as the primary workflow system for the infrastructure due to native support for OpenShift via the Kubernetes executor and the ability to use external public cloud resources via integration with Amazon Web Services and Google Cloud Platform.

The Nextflow Workflow Manager

Nextflow is popular workflow manager for scientific data analysis workflows. Most importantly, Nextflow provides an abstraction layer between the workflow logic and the underlying execution platform that allows workflows to be completely portable. Scientists and developers write code in any scripting language, wrap the dependencies in containers or other popular package managers and have Nextflow orchestrate the tasks across different execution environments. A key difference of Nextflow is the dataflow programming model which uses a functional, data-driven and reactive approach which is capable of scaling large data applications consisting of millions of tasks.

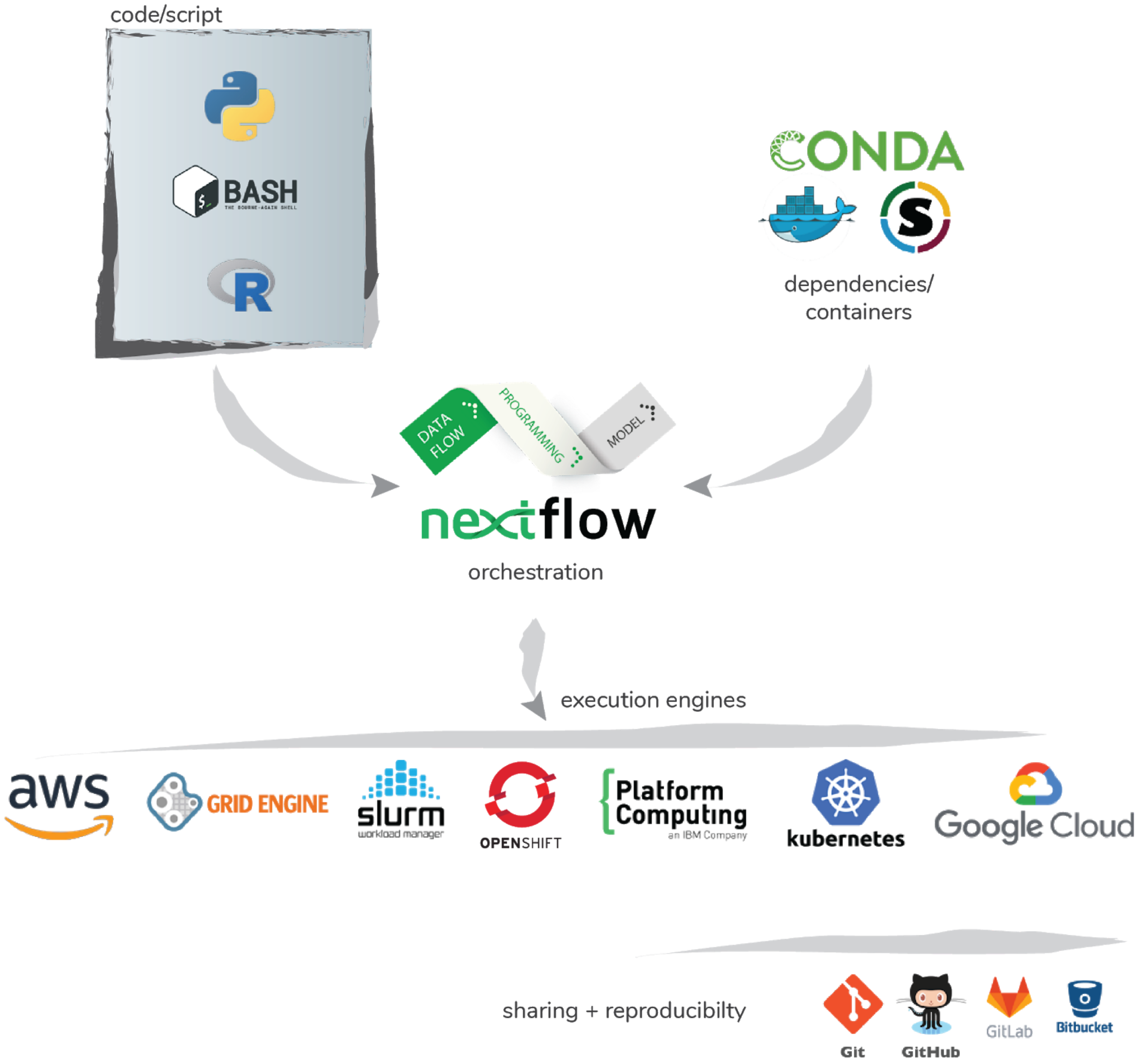

Figure 2. The Nextflow workflow manager was chosen as the primary workflow system for the OpenRiskNext infrastructure due to native support for OpenShift on the OpenRiskNet VRE and the ability to use external cloud resources via integration with Amazon Web Services and Google Cloud Platform

COMPUTE AND DATA FEDERATION

In this section we present the OpenRiskNet community efforts to enable compute and data federation and establish links from OpenRiskNet to use external resources for risk assessment workflows.

External Compute Infrastructure

Federating external compute infrastructure refers to the use of 3rd-party computational resources. After investigating different options, we chose public cloud providers as the most widely applicable use case for external compute federation. The main factors for deciding this was their close integration with workflow engines, the advent of managed HPC services such as AWS Batch and the lack of user restrictions for using public cloud infrastructure. Finally, and importantly, the "pay-as-go" cost model has zero overhead costs which is essential for analyses which need access to the federated resources only when such resources are not available locally.

Kubernetes-based execution (K8S)

The core technology that the OpenRiskNet virtual infrastructure is based on is OpenShift. OpenShift is a container platform based on Kubernetes. All OpenRiskNet services ultimately run on top of the OpenShift and it was therefore important to have scientific workflows that are able to interact with OpenShift and Kubernetes at a native level. The technology provides an excellent infrastructure for the execution and the management of containers at scale, however it does not solve the challenges related to the orchestration and the deployment of scientific workloads such as tasks parallelisation and graph of dependencies, input and output data management, execution provenance, etc.

Nextflow covers many of these requirements, moreover its ability encapsulate the workflow tasks into container executions that make a perfect match for the Kubernetes computing model. We developed the Kubernetes executor for Nextflow to aid in this deployment scenario. Nextflow deploys the workflow execution as a pod in the Kubernetes cluster. This pod behaves as a driver application, which orchestrates the job executions and parallelisation spinning the execution of a separate pod for each task that needs to be computed by the workflow application. Effectively, Nextflow deploys the computation mapping the workflow tasks to Kubernetes and orchestrates their execution.

Kubernetes is at the core of Cloud Native Computing which aims to use open source software stacks to deploy applications as microservices, packaging each part into its own container, and dynamically orchestrating those containers to optimize resource utilization. This integration of Kubernetes with scientific computing workflows is the first step in federating external compute resources for risk assessment pipelines.

OpenShift Configuration

To connect Nextflow into the reference VE, we created a project (equivalent of a Kubernetes namespace) called "nextflow". Within this project, a cluster was provisioned where Nextflow pipelines can be executed. Nodes are dedicated to executing Nextflow by means of labels and a default node selector for the nextflow project. Additionally, consolidated logging, metrics and prometheus were installed to allow monitoring. Five persistent volumes (PVs) named `nf-pv-000{1-5}` and corresponding persistent volumes claims (PVCs) `nf-pvc-000{1-5}` were created to enable individual tasks (pods) to share data via an NFS export from the nf-infra node. Input/output for workflow data can be sent over ssh to the nf-infra node in the `/exports-nf/pv-000{1-5}` directories. This `/exports-nf` directory is backed by a 300GB cinder volume, with each PVC being limited to 100GB. The complete OpenShift recipe can be found on the OpenRiskNet repository[1].

User Steps

From the user perspective, from within the OpenRiskNet VE, any Nextflow workflow can be executed using the following commands:

```
$ oc project nextflow
$ nextflow kuberun <workflow_name> -v <pvc_name>
```

Amazon Web Services (AWS)

Amazon Web Services is widely considered to be the market leader in public cloud architecture. AWS has hundreds of different offerings for compute, storage and networking solutions. One recent development is the introduction of products that offer what can be termed HPC as a service. AWS Batch is a fully managed service that allows scientists and developers to scale thousands of jobs on AWS infrastructure. Based on the specification of submitted jobs, AWS Batch dynamically allocates the optimal quantity and type of compute resource (e.g., CPU or memory optimized instances) using EC2 and Spot Instance services. The integration of Nextflow with AWS Batch allows users to submit a workflow from the VE and have the task executed on the public cloud.

To configure workflows to run jobs on AWS from the OpenRiskNet VE, the following values are required for the nextflow.config file.

```
profile {
   batch {
      workDir            = 's3://cbcrg-eu/work'  // Any AWS S3 bucket
      process.executor  = 'awsbatch'
      process.queue     = 'demo'                 // AWS Batch Queue
      executor.awscli    = '/home/ec2-user/miniconda/bin/aws'
      aws.region         = 'eu-west-1'            // AWS Region
      aws.accessKey      = ABCDEF                // AWS Access Key
      aws.secretKey      = 0123456789            // AWS Secret Key
   }
```

[1] https://github.com/OpenRiskNet/home/tree/master/openshift/recipes/nextflow-cluster

```
}
```

With the above configuration, from within the OpenRiskNet VE, users can execute any Nextflow workflow with tasks running on AWS Batch using the following command:

```
$ nextflow kuberun <workflow_name> -profile batch -v <pvc_name>
```

Google Cloud Platform (GCP)

We wanted to provide users with alternative public cloud providers to AWS whilst also ensuring that workflows would be portable across all solutions. The Google Cloud Platform (GCP) is a widely used collection of cloud computing services that runs on the same infrastructure that Google uses internally. The Google Genomics Pipelines API is similar to AWS Batch in that it offers a managed service for the execution of tasks handled through the API. One key difference between the two is the need to specify an instance type as opposed to a queue. Each task subsequently is assigned to an instance with no optimisation of CPU/memory resources.

To configure workflows to run jobs on with the Google Genomics Pipelines API from the OpenRiskNet VE, the following values are required for the nextflow.config file.

```
profile {
    google
        process.executor = 'google-pipelines'
        cloud.instanceType = 'n1-standard-1'     // GCP instance type
        google.project = 'your-project-id'       // GCP project ID
        google.zone = 'europe-west1-b'           // GCP zone
    }
}
```

Subsequently, from within the OpenRiskNet VRE, users can execute any Nextflow workflow with task running on GCP using the following command:

```
$ nextflow kuberun <workflow_name> -profile google -v <pvc_name>
```

Hybrid, Cloud-Bursting and Multi-Cloud

The above solutions also provide the possibility to perform hybrid deployments where specific tasks are executed across different environments. Some risk assessment analyses may require specific resources such as a large amount of memory or accelerators (GPU or FPGA). In these situations, tasks can be dynamically assigned to a cloud service. To demonstrate this, we wished to apply this in the OpenRiskNet VE, using the specification that if any given task required more than 32GB of RAM, then these tasks should burst into the AWS cloud using AWS Batch and a queue associated to the appropriate sized instance.

To configure workflows to run specific tasks on AWS from the OpenRiskNet VE, we can add the following to the nextflow.config file.

```
process {
    queue     = { task.mem < 32.GB ? null : 'aws-batch-queue'  }
    executor = { task.mem < 32.GB ? 'k8s' :  'aws-batch'  }
}
```

The same logic can be applied to any task. This could include tasks where the input data was already in a cloud bucket (data localisation) or the opposite where data security restrictions mean that some data can not be moved from one environment to another. This flexible approach to hybrid cloud solves the majority of issues for users of risk assessment workflows and is made possible with the use of managed services provided by the public cloud infrastructure companies.

External Data Federation

This section describes approaches that were taken to integrate external data sources and allow OpenRiskNet users to source both data and application code.

The NCBI Sequence Read Archive (SRA)

The Sequence Read Archive is a public database for sequencing data, in particular, "short reads" generated by high throughput sequencing technologies. The Sequence Read archive is a prototypical database whereby having a local copy within a VE is unfeasible. As of April 2019, the database contains 70.6 trillion sequences and the database is doubling in size approximately every two years.

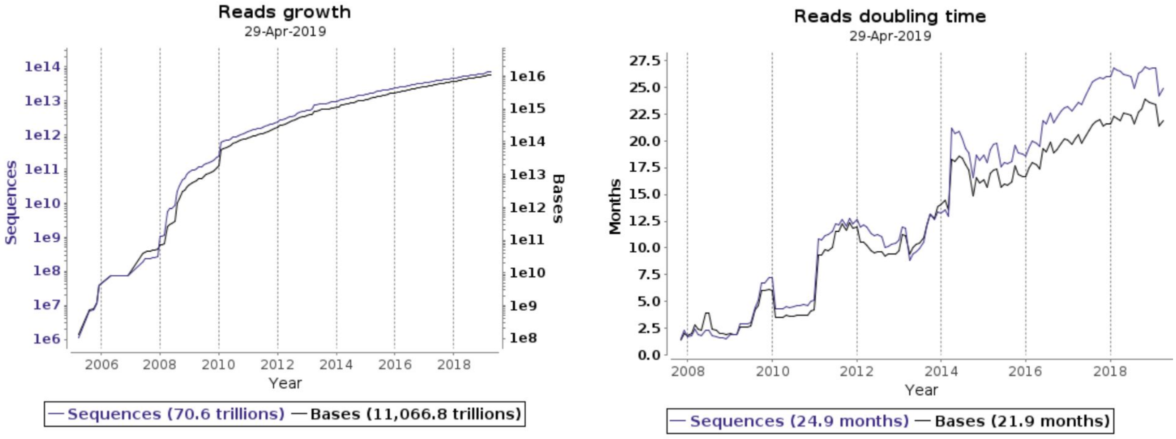

Figure 3. The growth of publically available genomic read data in the Sequence Read Archive Database

OpenRiskNet is providing its users with programmatic access to this resource from within their workflows. In order to do this, we implemented a method in Nextflow to retrieve samples from the SRA in the most flexible way possible.

```
// Create a channel from an experiment ID
Channel
    .fromSRA('SRX3859232')
    .println()

// returns the sample ID & FTP address
[SRR6911292, /vol1/fastq/SRR691/002/SRR6911292/SRR6911292.fastq.gz]
[SRR6911293, /vol1/fastq/SRR691/003/SRR6911293/SRR6911293.fastq.gz]
[SRR6911294, /vol1/fastq/SRR691/004/SRR6911294/SRR6911294.fastq.gz]
...

// Create a channel from a text search
Channel
   .fromSRA('liver[All Fields] AND toxicity[All Fields] AND "Homo sapiens"[Organism]')
   .println()
```

This returns an FTP address of the specific read files. In theory this could be applied to any addressable data source including in S3 buckets and tasks could then be run in the same cloud location as the S3 bucket. In practice we illustrate the use of this feature in the preprocessing steps of the toxicogenomics cases study described in more detail below.

The European Genome-Phenome Archive (EGA)

The European Genome–phenome Archive (EGA) is a permanent repository for all types of potentially identifiable bio-molecular and phenotypic data from biomedical research projects. The resource accepts raw data from sequencing, genotyping, transcriptome or epigenetics experiments using next-generation sequencing platforms or array-based technologies. The EGA can also be used to archive any processed data, such as the locations of individual variations (e.g. SNPs) from the raw data or summary statistics from a particular project. The samples can be associated with phenotype data that have been consented for use in research.

Following the previous example of the SRA, the EGA has developed a series of APIs and services that allow for programmatically retrieval of metadata and downloading the corresponding data files:

- The EGA Beacon enables researchers to interrogate variants without compromising data sensitiveness by providing boolean answers to simple queries (e.g. would I find a specific single nucleotide polymorphism (SNP) in the dataset?). Depending on the answer, the researcher would be interested in applying for access to the dataset.
- The metadata API permits any researchers to identify datasets of interest so these can be requested to the relevant Data Access Committee (DAC). Information about

the samples, technologies, health conditions or toxic components could be programmatically retrieved and exposed in other services.

- The download API can be used by authorised users in order to physically download the files and datasets this user has been granted permission for.

Therefore, projects of interest could be identified either via the EGA Beacon or the metadata API and the files transferred to authorised users. The EGA API is beyond the scope of the specific OpenRiskNet case studies however it provides an important building block for working with sensitive genomic data in future risk assessments within virtual environments.

Git-based Sharing

Git is an open source version control system with many hosting options. Sharing of applications, tracking changes and distributing scientific code is best done using Git-based sharing platforms. In cloud-native architecture like the OpenRiskNet VE, it is not given that the application code is in the same location as where the workflow will be executed. In these circumstances, application code must be accessible at the point of orchestration. For this task, we used Git. It allows users to manage their code in a consistent manner and also share and use other people's risk assessment workflows that are published through Git hosted services such as BitBucket, GitHub or GitLab. In the same way, with Nextflow it is possible to run code using any Git branch name, tag or commit ID as defined in the project repository. This allows for fully transparent and reproducible executions of a given analysis. For example, to run the development (dev) branch of the toxicogenomics use case from the OpenRiskNet GitHub we be specify the branch with "-r".

```
$ nextflow kuberun https://github.com/OpenRiskNet/nf-toxomix -r dev -v <pvc_name>
```

When executing the above command, Nextflow downloads the workflow project using the specified revision and stores it in the project execution file system. This allows the precise tracking of all workflow assets (i.e. application script, config files, containers, etc) with a single command. Moreover, it simplifies the staging of the application files in the OpenRiskNet storage location, which may be separated and not directly accessible from the hosting environment.

FEDERATION BETWEEN OPENRISKNET VIRTUAL ENVIRONMENTS

Cluster federation

Today, each OpenShift and Kubernetes cluster is a relatively self-contained unit, which typically runs in a single "on-premise" data centre or single availability zone of a cloud provider (Google's GCE, Amazon's AWS, etc).

We have investigated the options for federating multiple OpenRiskNet and Kubernetes clusters. The objectives includes the future notion of having multiple VEs running on different infrastructures, and with data and workload spread across these. Such solutions would improve the high availability capabilities of the system, and minimize the impact of cluster failure. Further it could contribute to reduce the latency; having clusters in multiple regions minimizes latency by serving users from the cluster that is closest to them. Federated clusters could also improve the scalability and offer better means to fault isolation. The main challenges for OpenShift and Kubernetes in this are[2]:

- location affinity (pods relative to each other, and to other stateful services like persistent storage)
- cross-cluster scheduling (given location affinity constraints and other scheduling policy, assigning resources to which clusters)
- cross-cluster service discovery (how do pods in one cluster discover and communicate with pods in another cluster)
- cross-cluster migration (how do compute and storage resources move from one cluster to another)

A potential hybrid solution is for a VRE to make computational services available through traditional HPC cluster solutions such as SGE, Slurm or HTCondor running within the Kubernetes cluster and to have these HPC clusters accessible from other VREs for executing workflows.

Cluster federation at the OpenShift and Kubernetes level is still very young and immature, and the architectures and implementations proposed by the community (including Ubernetes[1] and Kubernetes Federation, currently V2[3]) are under heavy development and the pilot projects are subject for public comments. Due to this fact and the unavailability of multiple VRE instances in the OpenRiskNet project, focus in compute federation was shifted more towards External Compute Infrastructure in the form of HPC systems (see above) and data federation.

[2] https://github.com/kubernetes/kubernetes/blob/8813c955182e3c9daae68a8257365e02cd871c65/release-0.19.0/docs/proposals/federation.md

[3] https://github.com/kubernetes-sigs/kubefed

Data federation

Data federation is a form of data virtualization where the data stored in a heterogeneous set of autonomous data stores is made accessible to data consumers as one integrated data store by using on-demand data integration. For OpenRiskNet, the driving case study is to make data available to VREs between different data sources, such as between two separate VREs, or for federating large public repositories.

When exploring the open-source solutions that have been gaining momentum within the cloud-computing marketplace, the following projects are the most adopted: iRODS, ONEDATA, Owncloud and its most recent fork Nextcloud. We have evaluated these projects for use in OpenRiskNet, with the goal to obtain a user-friendly, lightweight and reliable tool to be easily deployed within the context of a kubernetes-based application.

Onedata (https://onedata.org/) has large momentum and is one of the technologies proposed by e.g. INDIGO Datacloud project for data federation. It is a very good candidate for data federation in OpenRiskNet, and has a very nice user interface and integration with EOSC AAI services. On the disadvantages, we find that OneData requires substantial resources to run as a service, and the deployment is not straightforward and requires specialists to deploy and investigate/fix errors.

iRODS (https://irods.org/) is a production-ready open-source tool that has been demonstrated in several international projects. One disadvantage that we discovered is that iRODS is not equipped with ease-of-deployment in Kubernetes/OpenShift, and also does not have as easy-to-use user interface as the other tools evaluated.

Nextcloud (https://nextcloud.com/) offers the essential functionalities when it comes down to provide a suite of client-server tools for file sharing with the plus option of enabling full federation. It is a more light-weight tool than e.g. Onedata, and is equipped with an official, stable kubernetes helm chart.

In the OpenRiskNet project we decided to focus most of the work of data federation towards Nextcloud, providing OpenRiskNet with an easy installable, stable, secure, encrypted out-of-the-box file-sharing solution with the option of full federation between VREs. The solution was successfully demonstrated between two instances of OpenRiskNet. At present, data federation between VRES has not been highly requested within the active use cases, but the service has been implemented and can easily be switched on when the project requires it.

A TOXICOGENOMICS USE CASE

We illustrate the use of external resources with the workflow developed to support the toxicogenomics case study TGX[4]. The TGX case study, “toxicogenomics-based prediction and mechanism identification”, features the top-down approach for creating prediction models based on differentially regulated genes and the corresponding workflow termed “nf-toxomix” is a pipeline for toxicology predictions based on transcriptomic profiles. The workflow is built using Nextflow with an accompanying container.

Nf-toxomix is adapted from research titled "A transcriptomics-based *in vitro* assay for predicting chemical genotoxicity *in vivo*" [6]. The method focuses on training the genotoxicity model with microarray based gene expression data from treated/non-treated samples and then assesses the prediction of genotoxicity using a training test validation approach. The steps followed in the workflow are presented in Figure 4. The transcriptomics data were obtained from NCBI’s Gene Expression Omnibus[5] (GEO) and the genotoxicity information from the paper. The metadata were manually curated in order to align the transcriptomics data with the genotoxicity information. All other steps were run automatically.

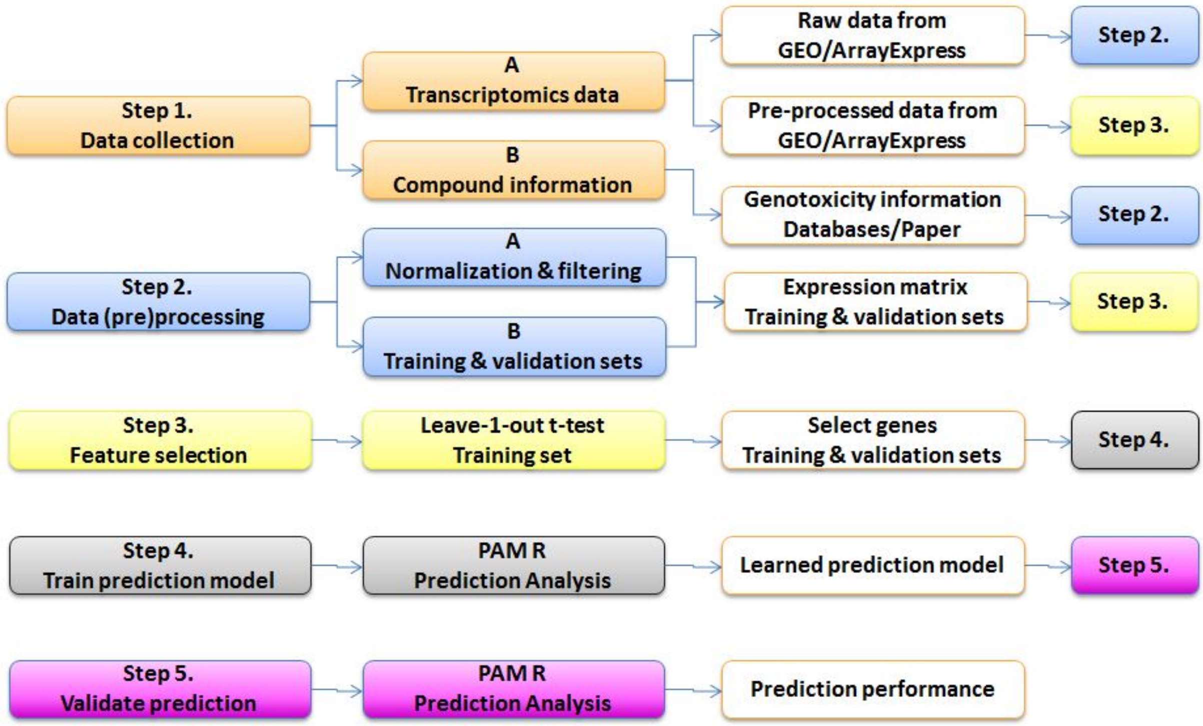
Step 1.
Data collection
A
Transcriptomics data
B
Compound information
Raw data from GEO/ArrayExpress
Step 2.
Pre-processed data from GEO/ArrayExpress
Step 3.
Genotoxicity information Databases/Paper
Step 2.
Step 2.
Data (pre)processing
A
Normalization & filtering
B
Training & validation sets
Expression matrix Training & validation sets
Step 3.
Step 3.
Feature selection
Leave-1-out t-test Training set
Select genes Training & validation sets
Step 4.
Step 4.
Train prediction model
PAM R Prediction Analysis
Learned prediction model
Step 5.
Step 5.
Validate prediction
PAM R Prediction Analysis
Prediction performance

Figure 4. 5-step workflow of top-down approach from TGX case study based on prediction method 1 described in Magkoufopoulou et al. 2012 [4]

We expanded the workflow to include preprocessing steps where raw transcriptomic data is searched for and downloaded from the Sequence Read Archive, then mapped against

[4] https://openrisknet.org/e-infrastructure/development/case-studies/case-study-tgx/
[5] https://www.ncbi.nlm.nih.gov/geo/

the human reference genome and read counts are generated. These steps are computationally demanding and therefore were configured to be deployed on public cloud resources (AWS Batch) from the OpenRiskNet VE. Due to the large amounts of data, the expanded study benefited from distributed computing architecture highlighted in this report. Additionally, this approach can be expanded to other case studies in the OpenRiskNet project In summary, this workflow allows easy integration of transcriptomics and genomics data sets and fast identification and localization of enriched genes on reference genomes using distributed cloud computing.

A **webinar** event was held including demonstration and is available from the event page on the OpenRiskNet site[6].

All documentation of the workflows can be found on GitHub[7].

[6] https://openrisknet.org/events/65/

[7] https://github.com/OpenRiskNet/nf-toxomix

CONCLUSION

In summarising the current and future challenges of risk assessment, it is evident that solutions must be flexible to handle the multitude of possible application requirements and deployment scenarios. This is especially true with large datasets. Currently, the most flexible approach to enable portable computation is not at the infrastructure level, which is too fragmented. Nor is it at the individual application or service level which requires low-level re-implementation. For scientific computing at least, workflow managers offer a proven solution to provide risk assessment analyses across the plethora of computing environments.

Within OpenRiskNet, we were able to use start-of-the-art workflow management techniques to solve several challenges including limitations of available resources via public cloud services, data localization via portable computation and data sensitivity via hybrid. We implemented querying and retrieval of large genomic datasets and examined how highly sensitive genomic data can be applied from within the virtual environment. Through the investigation of cluster federation we detailed projects that aim to unite multiple virtual environments together as well as data federation for common data stores.

In 2014, the Cloud Native Computing Foundation (CNCF) launched with a promise of less infrastructure fragmentation. It is still early in this transition to "cloud native" technology and in scientific computing at least, many challenges still remain. We would argue that currently the heterogeneity of infrastructure, platforms and services is only increasing. Through projects such as OpenRiskNet, we as scientists and system architects are able to explore the latest technologies and see how they can be applied to our domains. The case studies and insights generated from the compute and data federation of OpenRiskNet provide an excellent example of taking this new knowledge and extending what is possible within the fields of predictive toxicology and risk assessment.

GLOSSARY

The list of terms or abbreviations with the definitions, used in the context of OpenRiskNet project and the e-infrastructure development is available:

https://github.com/OpenRiskNet/home/wiki/Glossary

REFERENCES

1. Luechtefeld T, Marsh D, Rowlands C, Hartung T. Machine Learning of Toxicological Big Data Enables Read-Across Structure Activity Relationships (RASAR) Outperforming Animal Test Reproducibility. Toxicol Sci. 2018;165: 198–212.

2. Terry SF. The Global Alliance for Genomics & Health [Internet]. Genetic Testing and Molecular Biomarkers. 2014. pp. 375–376. doi:10.1089/gtmb.2014.1555

3. Afgan E, Baker D, Batut B, van den Beek M, Bouvier D, Cech M, et al. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. Nucleic Acids Res. 2018;46: W537–W544.

4. Di Tommaso P, Chatzou M, Floden EW, Barja PP, Palumbo E, Notredame C. Nextflow enables reproducible computational workflows. Nat Biotechnol. 2017;35: 316–319.

5. Köster J, Rahmann S. Snakemake—a scalable bioinformatics workflow engine. Bioinformatics. Narnia; 2012;28: 2520–2522.

6. Magkoufopoulou C, Claessen SMH, Tsamou M, Jennen DGJ, Kleinjans JCS, van Delft JHM. A transcriptomics-based in vitro assay for predicting chemical genotoxicity in vivo. Carcinogenesis. 2012;33: 1421–1429.