Deliverable

D9.5 Full technical specifications and publication submission

SELECTA full technical specifications and SELECTA publication to ensure long-term sustainability.

Version: 1.0

Due: 60

Completed: 60

Authors: Peter Harrison[1], Blaise Alako[1], Nadim Rahman[1], and Guy Cochrane[1]
[1]EMBL-EBI, UK.

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 643476.

Contents

D9.5 Full technical specifications and publication submission

In this report, we provide an overview of the COMPARE SELECTA technical specification documents, and the submitted publication that outlines the SELECTA framework. These together contribute to the long-term sustainability of SELECTA, with the plan for future grants and consortia to take up easily the maintenance and further development of the system.

In this report we document the safeguarding of SELECTA future sustainability by making available full technical documentation that covers system architecture, file format descriptions, API protocols, installation and data coordination workflows. This technical documentation is released alongside the openly licensed SELECTA software (released under Apache 2.0; https://github.com/EBI-COMMUNITY/ebi-selecta/blob/master/LICENSE). This open access will ensure that future developers can easily deploy the system to a new location and continue to improve the SELECTA Workflow Engine with further developments to its architecture. The utilisation of containerisation and cloud virtualisation technologies is key to this aim to ensure supported simple installation and reinstatement of the SELECTA framework.

We have also prepared an outline of the SELECTA system for submission to the OUP Bioinformatics journal for publication (https://academic.oup.com/bioinformatics), with a preprint to be made available through bioRxiv shortly. This manuscript acts as the entry point for the technical specifications for the SELECTA system, that is then further supported by the full detail provided in the documents folder of the SELECTA repository in GitHub (https://github.com/EBI-COMMUNITY/ebi-selecta/tree/master/docs).

The SELECTA installation at EMBL-EBI will remain active beyond the end of COMPARE and will be utilised in such ongoing research projects ZIKALLIANCE, RECODID and VEO as well as continuing to support the communities established through COMPARE.

Full technical specifications

The full technical specifications of SELECTA are provided within the SELECTA GitHub code repository itself (https://github.com/EBI-COMMUNITY/ebi-selecta). This includes an overview README file that provides instructions on the installation and basic operation of SELECTA. SELECTA itself has been containerised to improve the ease of installation so that the SELECTA framework can be launched without the requirement of installing every SELECTA component individually. As SELECTA is likely to be deployed in a cloud HPC environment, we have also provided installation scripts to initialize SELECTA in a docker swarm cluster. The readme also provides basic operational instructions meant as a minimal requirement to launch SELECTA processing. The most up to date version of the overview readme documentation is available in the GitHub repository at the following link https://github.com/EBI-COMMUNITY/ebi-selecta/blob/master/README.md, but the current version has been included in Annex 1 for clarity.

A more detailed full technical specification including more detailed installation and operational instructions are included in a PDF document, that is also held within the SELECTA GitHub repository in the documents folder (https://github.com/EBI-COMMUNITY/ebi-selecta/tree/master/docs). This covers the system architecture, file format descriptions, API protocols and data coordination workflows. Due to this document running at over 50 pages, these instructions have not been placed in this deliverable as an Annex, but can be accessed from the GitHub repository directly https://github.com/EBI-COMMUNITY/ebi-selecta/tree/master/docs.

Publication of SELECTA overview

To act as an entry point to the SELECTA framework for future user groups, we have prepared a manuscript that provides an overview of the SELECTA framework features and methods. It includes a clear overview schematic (Figure 1) that summarises the full workflow operations and individual component steps. SELECTA has been developed with the ability to be of benefit to the wider pathogen and scientific communities as other user-provided pipelines can be adapted for use with the framework. As the SELECTA framework is openly developed on GitHub, it has the potential to benefit from community-suggested improvements to the codebase in coming years. In order to be of benefit to the COMPARE community a preprint of the manuscript will be placed on the bioRxiv archive, but for clarity of this deliverable the current manuscript has also been placed in Annex 2. The manuscript will be submitted to the bioinformatics journal (https://academic.oup.com/bioinformatics).

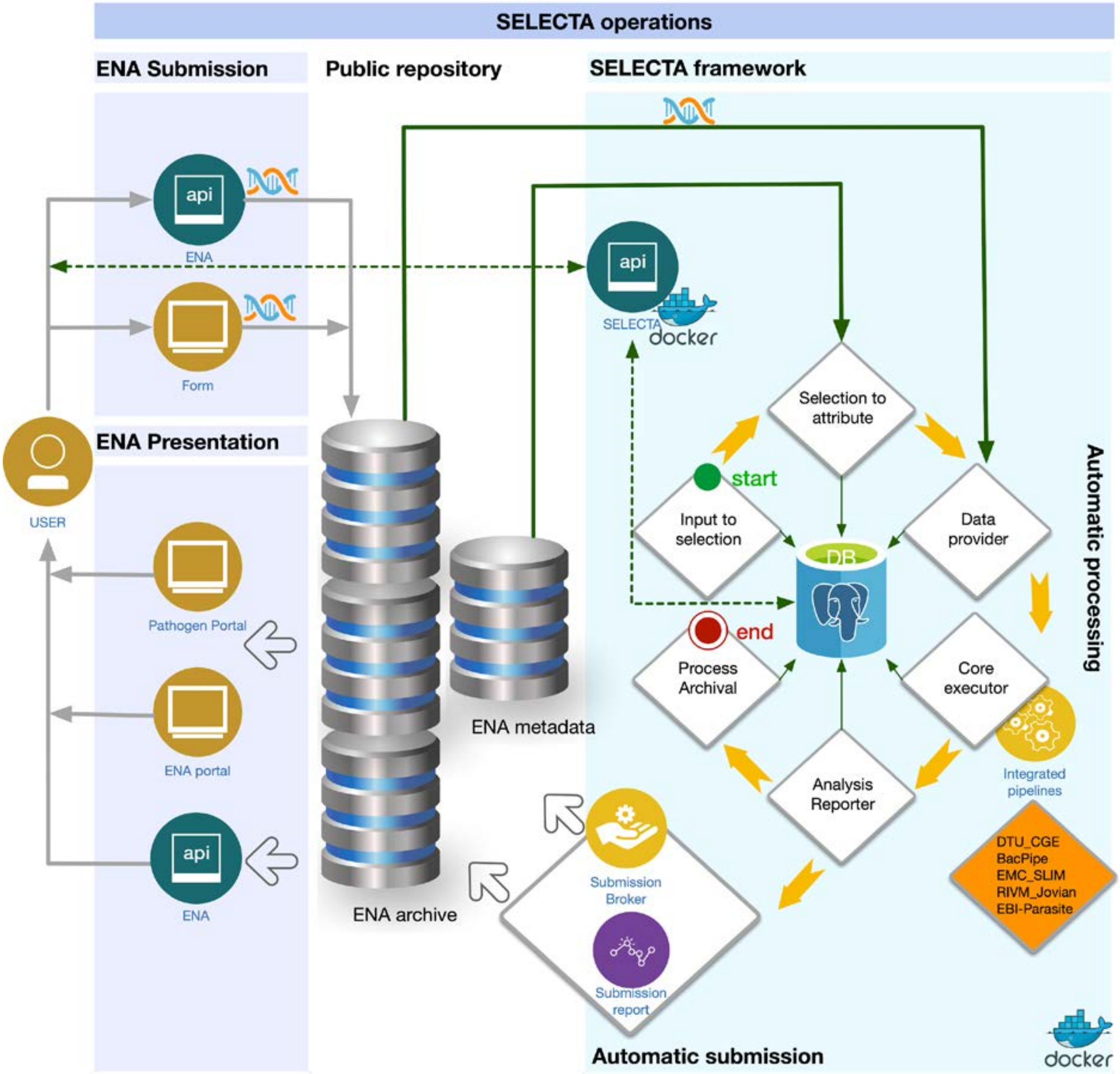


Figure 1. The SELECTA systems leveraging of ENA submission, archiving and presentation infrastructures. The submitted data are subjected to various internal processing stages with an audit trail recorded in backend PostgreSQL database. The analysis results are automatically resubmitted back to the ENA for discovery and retrieval via the ENA web browser (https://www.ebi.ac.uk/ena/browser/home) and the pathogen portal (https://www.ebi.ac.uk/ena/pathogens/home). Interaction with the SELECTA operations is possible through direct command line interaction or through the SELECTA API.

Annexes

Annex 1. SELECTA technical specification installation and basic operations readme

ebi-selecta

Contents:

What is SELECTA?

SELECTA is a rule-based workflow engine that runs analytical pipelines, first developed to manage pipelines specifically for the COMPARE community. However, SELECTA can also be setup to manage other user provided analytical pipelines. SELECTA automatically submits the analysis results output by the analytical pipelines back to the European Nucleotide Archive (ENA) mitigating the burden for a submitter when handling bulk submissions and ensures a consistent and accurate record of the analysis results.

SELECTA framework Docker-Compose version.

This repository provides SELECTA toolkits as a set of Dockerfiles, configuration docker-compose files to launch the SELECTA framework without the requirement of installing every SELECTA component individually. This repository also provides a script (init_swarm.sh) to initialize a docker swarm environment and a stack file to launch the framework in a docker swarm cluster. There is also a script to remove entirely (rm_swarm.sh) the created swarm cluster environment.

Installation

The open-source software Docker and Docker-compose must be pre-installed for the proper functioning of the SELECTA docker-compose version:

On Ubuntu:

Add the GPG key for the official Docker repository to the operating system.

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add
-
```

Add the Docker repository to the APT (Advanced Package Tool) sources.

```
sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

Next, update the Ubuntu package database with the newly added docker packages repository.

```
sudo apt-get update
```

Install the Docker community edition:

```
sudo apt-get install -y docker-ce
```

Verify that Docker daemon is running with the following command:

```
sudo systemctl status docker
```

Next, install the docker-compose and make it executable.

```
sudo curl -L
https://github.com/docker/compose/releases/download/1.18.0/docker-compose-
`uname -s`-`uname -m` -o /usr/local/bin/docker-compose && sudo chmod +x
/usr/local/bin/docker-compose
```

Finally, verify that docker-compose installation is successful by checking its version.

```
docker-compose --version
```

On Windows and Mac:

Install the executable from the docker-compose page:

```
https://docs.docker.com/compose/install/
```

Retrieving the SELECTA docker-compose version

Retrieve the docker-compose version from the Github to launch the docker-compose version of the SELECTA framework.

Clone this repository:

```
git clone <repo>
```

If utilising for COMPARE you will also want to obtain the following codebases for the integrated COMPARE pipelines:

- DTU_CGE
- EMC_SLIM
- UAntwerp_Bacpipe
- RIVM_Jovian (private, request by email)

Alternatively, obtain the codebases of the analysis pipelines that you wish to implement in SELECTA.

Usage on a single computer

To launch the SELECTA framework on a single machine. Access the SELECTA docker repository:

```
cd </path/to/ebi-selecta>
```

Launch SELECTA

```
Docker-compose up --build -d
```

The last command in detach mode will pull the SELECTA required images dependencies from Docker hub and build where necessary SELECTA-tools docker images and finally launch all the SELECTA containers. The docker-compose also creates a docker network that allows communication between the SELECTA docker containers.

SELECTA API

We provide in this repository SELECTA API documentation detailing SELECTA endpoints. SELECTA API implements both the GET and POST methods.

Minimum requirement for a SELECTA processing:

The SELECTA framework is database-centric. Two main tables, the Account and the Process_Selection tables, contain mandatory requirements to enact SELECTA processing. SELECTA API facilitates populating these tables with the required attributes. The API listens to port 5002 on the localhost, and the accompanying docker-compose.yml file defines this port.

- **Create a SELECTA user account**

The following command will create an account for dcc_xxxx

```
curl -d '{"account_id":"dcc_test","account_type":"datahub","email":
"selecta@ebi.ac.uk","password": "letmein"}' -H "Content-Type:
application/json" -X POST http://localhost:5002/account
```

- **Create SELECTA rule**

The PROCESS_SELECTION table sports the processing rules for each datahub. Here we define the analytical workflow name that is responsible for analyzing the data from a specific datahub or study or run. The following command creates a rule in the PROCESS_SELECTION table for processing data from datahub dcc_dvorak with analysis workflow UAntwerp_Bacpipe; the rule ensures that the analysis of the run is an ongoing process by setting the CONTINUITY flag to YES.

This specific rule continually analyzes a specific run data in the dcc_dvorak datahub.

```
curl -d '{"datahub": "dcc_dvorak","run_accession":
"ERR1102130","pipeline_name": "UAntwerp_Bacpipe","public": "NO","webin":
"Webin-45433","continuity": "YES","process_type": "run"}' -H "Content-Type:
application/json" -X POST http://localhost:5002/input2selection
```

Retrieve SELECTA database data

Via API

Fetch all SELECTA account

```
curl  -H "Content-Type: application/json" -X GET
http://localhost:5002/accounts
```

Fetch all SELECTA rules

```
curl  -H "Content-Type: application/json" -X GET
http://localhost:5002/selections
```

Via Postgres PgAdmin client

```
Http:localhost:5050 with the following: username: selecta@ebi.ac.uk/letmein
```

Run SELECTA Workflow manually

When manually running each SELECTA stage, run each of the following scripts sequentially: selection_to_attribute.py, data_provider.py, core_executor.py, analysis_reporter.py, and process_archival.py. Each script require SELECTA configuration file (properties.txt) as an argument:

Selection_to_attribute Stage

```
docker run -ti --rm --network=ebi-selecta_postgres embl-
ebi/selecta_selection_to_attribute:1.0
/usr/scr/app/scripts/selection_to_attribute.py -p
/usr/scr/app/resources/properties.txt
```

Data_provider Stage

```
docker run -ti --rm --network=ebi-selecta_postgres embl-
ebi/selecta_data_provider:1.0 /usr/scr/app/scripts/data_provider.py -p
/usr/scr/app/resources/properties.txt
```

Core_executor Stage

```
docker run -ti --rm --network=ebi-selecta_postgres embl-
ebi/selecta_core_executor:1.0 /usr/scr/app/scripts/core_executor.py -p
/usr/scr/app/resources/properties.txt
```

Analysis_reporter Stage

```
docker run -ti --rm --network=ebi-selecta_postgres  embl-
ebi/selecta_analysis_reporter:1.0
/usr/scr/app/submission/analysis_reporter.py -p
/usr/scr/app/resources/properties.txt
```

Process_archival Stage

```
docker run -ti --rm --network=ebi-selecta_postgres embl-
ebi/selecta_process_archival:1.0 /usr/scr/app/scripts/process_archival.py -
p /usr/scr/app/resources/properties.txt
```

Run SELECTA Workflow automatically

To automate the steps described above, create individual cronjobs for each stage to run periodically.

Usage on a SWARM cluster

Access the SELECTA docker repository:

```
Cd </path/to/ebi-selecta>
```

Initialize SELECTA swarm cluster

```
./init_swarm.sh
```

Confirm cluster creation

```
docker node ls
docker node inspect node-id
```

launch SELECTA in the swarm cluster

To launch SELECTA in the swarm cluster, issue the following command:

```
Docker stack deploy -c docker-stack-compose.yml selecta
```

The last command will pull the SELECTA required images from the docker hub and launch all the SELECTA services as configured in the docker-stack-compose YAML file.

The SELECTA framework integrates a swarm visualizer that is accessible at

```
http://localhost:8080
```

To scale up core_executor service, issue the following command:

```
docker service scale selecta_core_executor=10
```

Confirm that the service has scaled up via swarm visualizer at http://localhost:8080

To scale down core_executor service, issue the following command:

```
docker service scale selecta_core_executor=1
```

Confirm that the service has scaled up via swarm visualizer at http://localhost:8080

remove the swarm cluster

To remove the swarm cluster, issue the following command:

```
rm_swarm.sh
```

Useful docker commands

The following are some useful docker commands to interacts with the containers in the SELECTA framework.

Show active containers:

```
Docker-compose ps
```

Request an interactive shell in a running container.

```
Docker exec -it container_name csh
```

List virtual volumes:

```
docker volume ls
```

List Images.

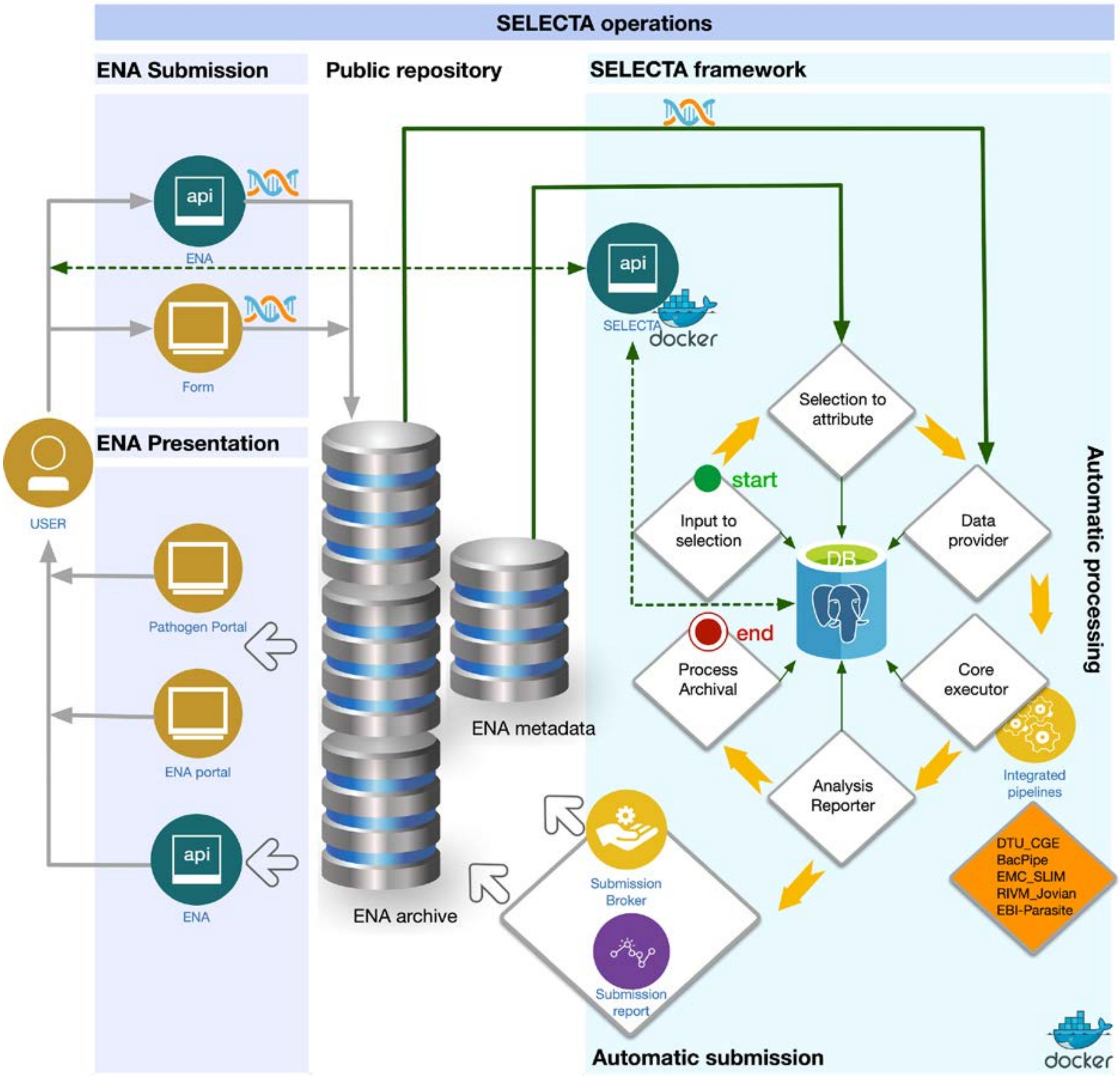
```
Docker images
```

Remove images

```
Docker rmi image_name
```

Background

Workflow



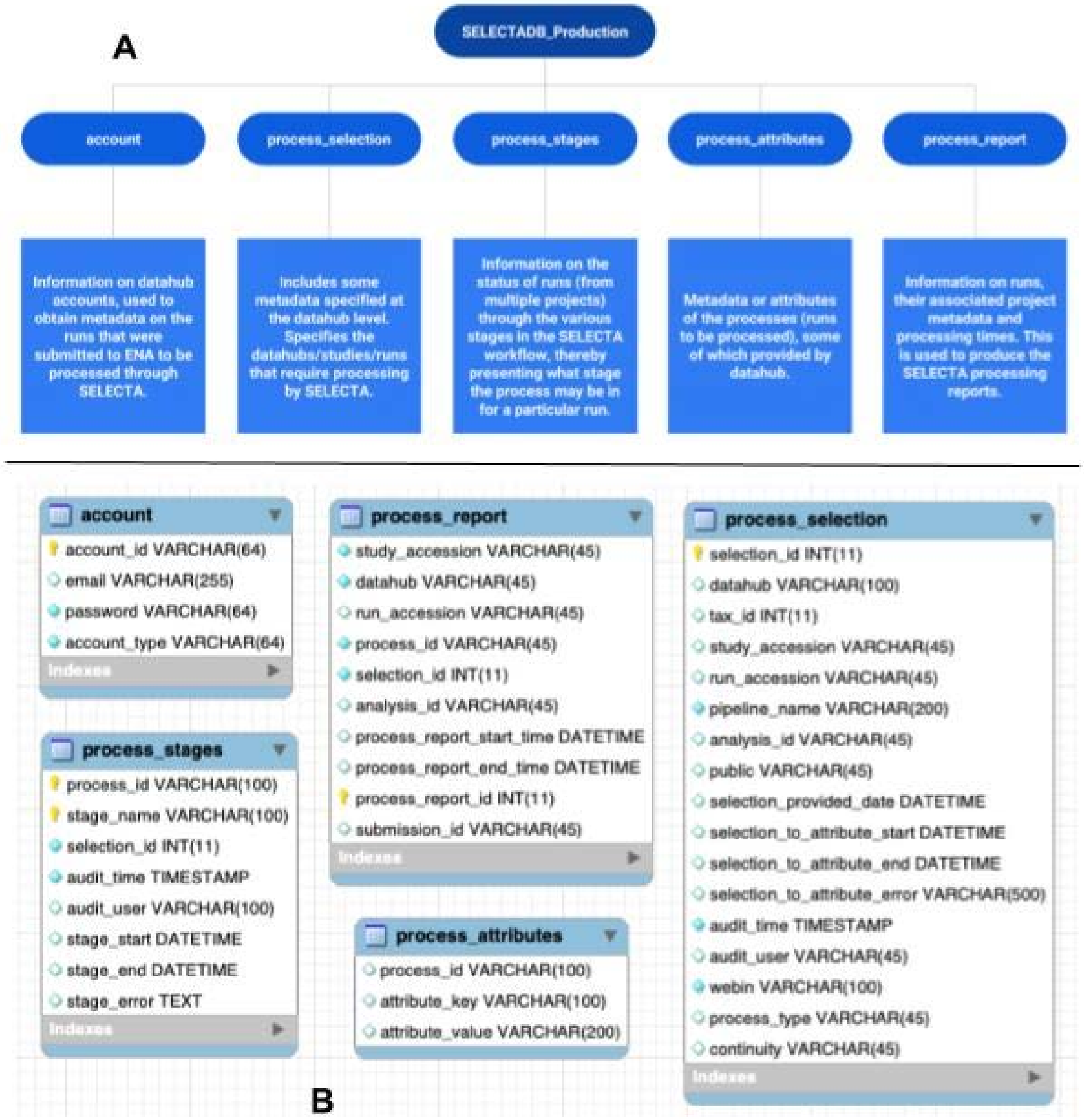
The SELECTA workflow includes the stages shown in the image above. Each step runs an associated python script(s). The workflow can be scheduled through cronjobs which run these scripts, note the direction of the flow in the SELECTA framework section of the diagram.

Database

SELECTA utilises a backend PostGres database for storing, accessing and updating key information required for tracking the various processes. The database consists of 5 tables:

- account
- process_selection
- process_stages
- process_attributes
- process_report

The image below presents the fields for each table:



Information on datahub accounts, used to obtain metadata on the runs that were submitted to ENA to be processed through SELECTA.

Includes some metadata specified at the datahub level. Specifies the datahubs/studies/runs that require processing by SELECTA.

Information on the status of runs (from multiple projects) through the various stages in the SELECTA workflow, thereby presenting what stage the process may be in for a particular run.

Metadata or attributes of the processes (runs to be processed), some of which provided by datahub.

Information on runs, their associated project metadata and processing times. This is used to produce the SELECTA processing reports.

account
- account_id VARCHAR(64)
- email VARCHAR(255)
- password VARCHAR(64)
- account_type VARCHAR(64)

process_stages
- process_id VARCHAR(100)
- stage_name VARCHAR(100)
- selection_id INT(11)
- audit_time TIMESTAMP
- audit_user VARCHAR(100)
- stage_start DATETIME
- stage_end DATETIME
- stage_error TEXT

process_report
- study_accession VARCHAR(45)
- datahub VARCHAR(45)
- run_accession VARCHAR(45)
- process_id VARCHAR(45)
- selection_id INT(11)
- analysis_id VARCHAR(45)
- process_report_start_time DATETIME
- process_report_end_time DATETIME
- process_report_id INT(11)
- submission_id VARCHAR(45)

process_attributes
- process_id VARCHAR(100)
- attribute_key VARCHAR(100)
- attribute_value VARCHAR(200)

process_selection
- selection_id INT(11)
- datahub VARCHAR(100)
- tax_id INT(11)
- study_accession VARCHAR(45)
- run_accession VARCHAR(45)
- pipeline_name VARCHAR(200)
- analysis_id VARCHAR(45)
- public VARCHAR(45)
- selection_provided_date DATETIME
- selection_to_attribute_start DATETIME
- selection_to_attribute_end DATETIME
- selection_to_attribute_error VARCHAR(500)
- audit_time TIMESTAMP
- audit_user VARCHAR(45)
- webin VARCHAR(100)
- process_type VARCHAR(45)
- continuity VARCHAR(45)

Annex 2 SELECTA manuscript

Manuscript version at time of submission of deliverable is shown.

SELECTA: A toolkit for the systematic orchestration of analytical workflows and the automation of analysis data sharing through the European Nucleotide Archive

Authors

Blaise T.F Alako[1], Peter Harrison, Nima Pakseresht[1], Nadim Rahman[1] and Guy Cochrane[1]

Affiliations

[1]European Molecular Biology Laboratory, European Bioinformatics Institute, Wellcome Genome Campus, Hinxton, Cambridge CB10 1SD, UK

Abstract

Motivation: Whole-genome sequencing approaches for real-time pathogen surveillance are seeing increasing adoption across public health operations. In order to enable rapid turnaround from raw sequencing data to interpretable analyses, systematic and autonomous computational processing of data is required as data emerge from surveillance programmes and outbreak response initiatives. In order to maximise reusability of the data, such systems must also be closely linked to public data sharing infrastructure.

Result: This article introduces SELECTA, a rule-based computational workflow engine and scheduler, developed within the COMPARE initiative, with applications across health and life sciences. SELECTA is available as a set of command-line tools and RESTful API. SELECTA fully automates the analysis of sequence data and autonomously submits the generated results to the “Data Hub” system, itself built upon the foundations of the European Nucleotide Archive (ENA), for subsequent discovery and retrieval, ensuring analysis results are recorded consistently and accurately. Moreover, the analysis id obtained can immediately be referenced from a research manuscript in preparation for publication and allows the effective use of the analysis outputs in real-time pathogen surveillance.

Availability: SELECTA was written in python3. The source code and a comprehensive user manual are freely available under the Apache 2.0 License at https://github.com/EBI-COMMUNITY/ebi-selecta

Contact: blaise@ebi.ac.uk

Supplementary information: Supplementary data (presentation results screenshot via ENA and pathogen portal, User manual) will be made available online.

1 Introduction

Those involved in pathogen surveillance and outbreak investigation are turning increasingly to whole-genome sequencing-based methods. The comparable and interoperable data that these methods produce support not only identification and tracking, but also provide a foundation for many further analyses, including deep epidemiological study, research into mechanisms behind transmission and infection and exploration of potential therapeutic targets. Such broad applicability of these sequence data makes a compelling case for the broadest sharing of data, where possible through fully open database systems. The EU COMPARE project (https://www.compare-europe.eu/) seeks to enable open whole-genome sequencing-based data sharing through its “Data Hub” system (Amid *et al.*, 2019), itself based on the foundation provided by the European Nucleotide Archive (ENA; Amid *et al.*, 2020) which, with its partner databases in the International Nucleotide Sequence Database Collaboration (INSDC; Karsch-Mizrachi *et al.*, 2018), provides the established database of public record for sequence data. In this paper we introduce open source technical infrastructure to supports the application of multi-step computational workflows to pathogen sequence data that are public, or destined to be public, in the Data Hub system, returning the outputs of the computational workflows back into the system.

Our system, SELECTA, is a toolkit for the systematic orchestration of computational analytical workflows and the automation of analysis data sharing. The system can be configured autonomously to consume raw sequence data from ENA or Data Hubs, operate a given workflow and return the outputs of analysis - in the form of data files and data summaries - to ENA or Data Hubs. The current EMBL European Bioinformatics Institute instance of SELECTA in operation for COMPARE and its related EU projects (RECODID and VEO) currently supports five installed pathogen computational analysis workflows spanning taxonomies (bacteria, viruses and parasites) and sequencing methods (isolate genomics and metagenomics).

We provide SELECTA as open software at https://github.com/EBI-COMMUNITY/ebi-selecta under the Apache 2.0 License. We encourage those responsible for pathogen data analysis systems operating upon public data and providing public analytical results to use, adapt and contribute to the codebase. While the system has been built with pathogen analysis in mind, it is workflows-agnostic and will support workflows beyond the pathogens; we encourage those with these broader interest working on ENA data and wishing to publish their results to participate.

2 Features and Methods

2.1 Common data submission scenario

The ENA has two main, well-documented, routes for researchers to share their genomic sequencing data with the community: a web-based and a command-line programmatic approach, both leveraging the Webin submission toolkit (https://ena-docs.readthedocs.io/en/latest/submit/general-guide.html). However, within the COMPARE initiative, there is a need to analyze the raw genomic data and ultimately share both the raw data and analytical findings with the scientific community. Hence was born the requirement for SELECTA to include functionality for automated data submission brokering that ensures consistent, accurate and rapid release of analysis data to the scientific communities that it supports.

2.2 SELECTA algorithmic approach

The raw genomic sequences data submission follows the conventional data submission routes above (**figure 1**). Once submitted, the data are automatically identified and captured by the SELECTA framework for further processing. SELECTA relies on a rule-based approach, defined in its backend PostgreSQL database, to identify newly submitted raw data for processing. Data grouped by either datahub, project accession are annotated with bioinformatic tools that should operate on them. The system also monitors and controls, based on user defined parameters, whether processing in a group is an ongoing or a one-time operation. This implies that the same dataset can continually be processed by multiple different analysis pipelines for example for the purpose of algorithm comparison or join-merge analysis. There are also circumstances where re-analysis is necessary. For example due to an error in sequencing or subsequent re-sequencing with new and better chemistry reagent. SELECTA rule-based selection makes it possible to define a rule that enforces re-analysis of the new sequence data by setting it’s processing type and the continuity parameter, accordingly.

2.3 Operations

SELECTA processing goes through six sequential and dependent stages. The input to the selection, the selection to the attribute, the data provider, the core executor, the analysis submission and the data archival. All stages make use of the same configuration file. This defines the paths to the integrated analysis pipelines, its dependent databases, and parameters for load balancing if required. A full list of the configuration file options is detailed in the supplementary documentation. The Input to attribute stage is the first step in enacting the SELECTA workflow. It is done by amending new records into the SELECTA backend PostgreSQL database, such as data hub, study or run, data center, analysis pipeline name and other metadata such as continuity and process type. The SELECTA documentation in supplementary data provides example codes to achieve this input loading. Figure 1 illustrates the data and processing flow of the SELECTA system. **Table 1.** illustrates SELECTA sequential and hierarchical modes of action.

Table 1 SELECTA operation stages.

Operation stages	Description
Selection_to_attribute.py -p configuration_file	This stage fetches all known metadata for a run from the ENA repository to populate various database tables for the purpose of annotating the analysis where needed.
Data_provider.py -p configuration_file	The data provider stage, downloads the appropriate sequence reads deposited at the ENA. These are the primary data consumed by the integrated analysis workflows.
Core_executor.py -p configuration_file	The core executor stage, calls on the appropriate pipeline as defined in the configuration rules and constructs the necessary command in the backend to process the genomic sequence data.
Analysis_reporter.py -p configuration_file	This operation automates the submission of the pipeline analysis results back to the ENA and provides the SELECTA database with the submission and analysis identifiers for subsequent discovery of the data at the ENA or pathogen portals.
Process_archival.py -p configuration_file	This operation performs some sanity checks and archives the submitted analysis results into a dedicated ENA archive.

The supplementary SELECTA document illustrates the sequential calls of SELECTA operation in a crontab file.

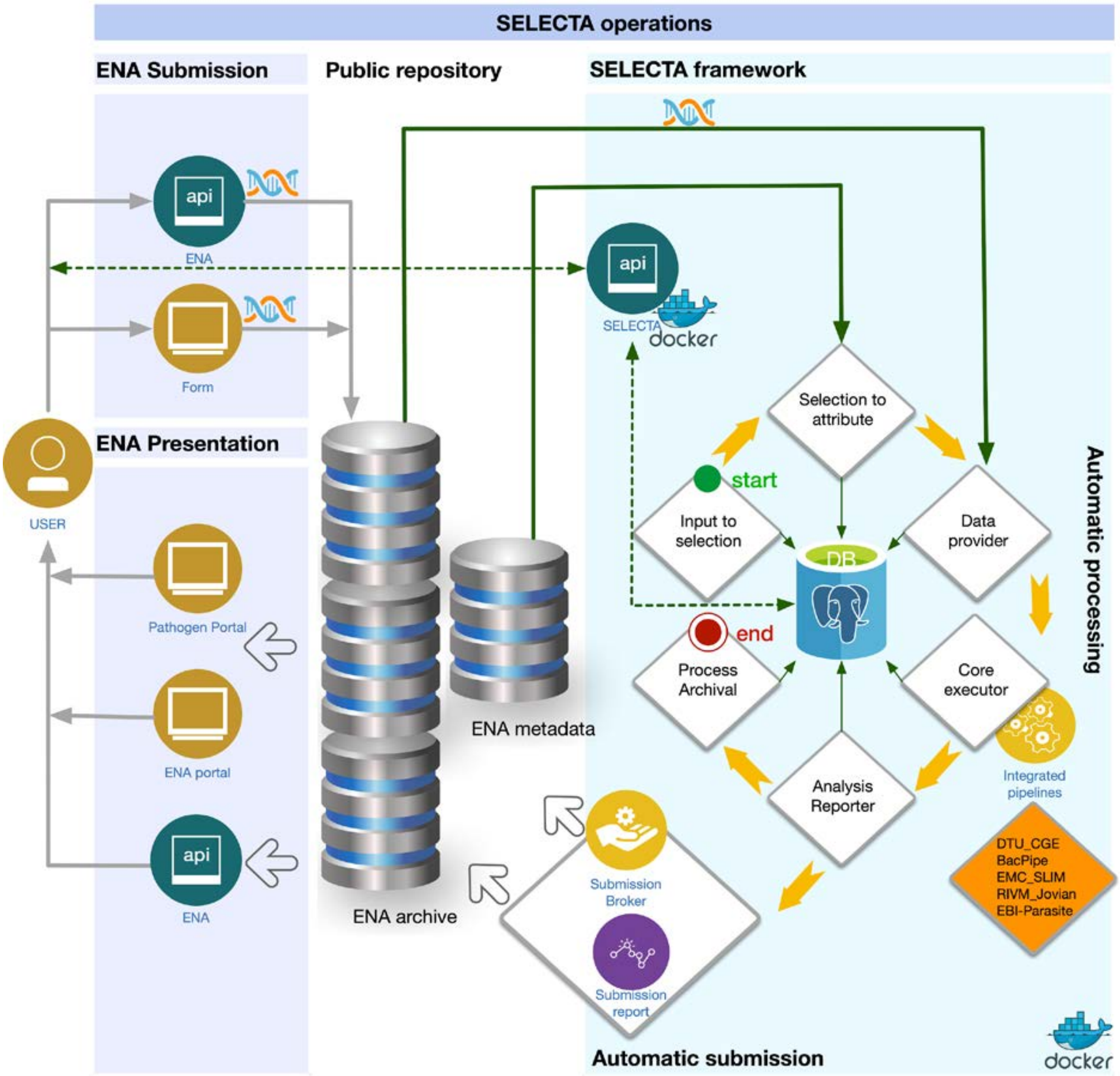


Figure 1. The SELECTA systems leveraging of ENA submission, archiving and presentation infrastructures. The submitted data are subjected to various internal processing stages with an audit trail recorded in backend PostgreSQL database. The analysis results are automatically resubmitted back to the ENA for discovery and retrieval via the ENA web browser (www.ebi.ac.uk/ena) and the pathogen portal (www.ebi.ac.uk/ena/pathogens). Interaction with the SELECTA operations is possible through direct command line interaction or through the SELECTA API. A list of the COMPARE integrated analytical workflows is detailed in the supplementary SELECTA documentation.

2.4 Analytical workflows integration

An important feature of the SELECTA framework is the capacity to integrate any analytical tool whose processing results can be presented and archived at the European Nucleotide Archive. The supplementary documentation illustrates the required steps to integrate an analytical tool into the SELECTA framework. One mandatory prerequisite for tool integration is the provision of one or many summary results in ASCII format, other additional relevant output formats can be submitted. The benefit of the ASCII format is that it is readily indexed by the ENA infrastructure and immediately discoverable in the ENA (www.ebi.ac.uk/ena) web browser and the pathogen (www.ebi.ac.uk/ena/pathogens) portal. The utilisation of the SELECTA framework within the COMPARE consortium integrated five different analytical workflows for automated processing and submission of its genomics analysis data.hese are briefly detailed in a recent manuscript by Amid *et al.* (The COMPARE data hubs:bioRxiv 10.1101/555938) and latest results available from the pathogens portal (https://www.ebi.ac.uk/ena/pathogens/home).

2.5 Troubleshooting

The SELECTA system is database-centric. In other words, each sample has an audit trail for each stage in the processing life cycle. Resetting the timespan of a sample analysis at any stage of the processing ensures that re-processing will process the analysis from that stage up to its completion and automatically submit the result to the ENA public archive. The supplementary documentation illustrates various troubleshooting scenarios at every key stage of the SELECTA framework.

2.6 RESTful API

We provide a RESTful application programming interface as a docker image for interrogating the status of various processing stages. The API can also retrieve analysis ids of successful submission from the SELECTA backend database. The API is able to amend status in the database to enforce that a specific stage is reprocessed. The supplementary document illustrates the API mode of operation and the various output expectation from the various APIs endpoints.

2.7 Advantages

There are various bioinformatic computing frameworks for orchestrating bioinformatics tools such as Nextflow (Di Tommaso *et al.*, 2017), Galaxy (Afgan *et al.*, 2018), Toil (Vivian *et al.*, 2017), Bpipe (Sadedin *et al.*, 2012). However, one of SELECTA's strengths is the packaging of analysis results with all necessary biosample metadata for a systematic and effective automatic submission to the ENA, therefore acting as a submission broker. Moreover, for research laboratories that lack computing facilities for analysing a large amount of genomic data, SELECTA leverages the Infrastructure as a service (IaaS) of the EMBL-EBI and benefits from physical proximity to genomic archive data held within the ENA to perform computationally expensive operations and systematically submit the analysis and appropriate metadata for public discovery and dissemination. Additionally SELECTA API facilitates the transparency on the processing stages, as well as the retrieval of metadata pertaining to a single or multiple analyzed biological samples. Furthermore, the source code is freely available, for processing sensitive data in a controlled environment. The SELECTA framework is flexibly deployable to run on a desktop, server or even deploy to the cloud, thanks to its availability as docker images.

An essential feature of the SELECTA framework is its portability across various operating systems. Two open-source tools, namely docker and docker-compose, enable SELECTA system-agnostic feature. To use the toolkit, simply fork a copy from its Github repository and run "docker-compose up" in the code directory. Furthermore, the toolkit can be deployed to a cloud environment thanks to the Docker Swarm, docker-engine build-in orchestration tool. We provide a stack definition file, as well as scripts to initialize and remove the swarm cluster (https://github.com/alexei-led/swarm-mac). The Kompose tool (https://github.com/kubernetes/kompose) can port the SELECTA stack definition file into a configuration file usable by the Kubernetes engine, should the user require a different orchestration engine. SELECTA services images are readily available in Docker Hub for effective Swarm or Kubernetes cloud deployment.

Acknowledgments

NP conceives and designed the initial version of the SELECTA in python2.

BA ported SELECTA into python3 and further enhanced its features and containerisation. The manuscript was drafted by BA. The SELECTA documentation was drafted by NR. The manuscript and documentation were reviewed by all authors.

Funding: The COMPARE consortium, with funding from the European Union’s Horizon 2020 research and innovation program under grant agreement No. 643476. This article reflects only the authors views. Neither the European Commission nor its Agency REA are responsible for any use that may be made of the information it contains..

Conflict of interest
None declared.

References

Afgan,E. *et al.* (2018) The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. *Nucleic Acids Res.*, **46**, W537–W544.

Amid C, Alako BTF, Balavenkataraman Kadhirvelu V, Burdett T, Burgin J, Fan J, Harrison PW, Holt S, Hussein A, Ivanov E, Jayathilaka S, Kay S, Keane T, Leinonen R, Liu X, Martinez-Villacorta J, Milano A, Pakseresht A, Rahman N, Rajan J, Reddy K, Richards E, Smirnov D, Sokolov A, Vijayaraja S, Cochrane G. The European Nucleotide Archive in 2019. Nucleic Acids Res. 2020 Jan;48(D1) D70-D76. doi:10.1093/nar/gkz1063. PMID: 31722421.

Amid C, Pakseresht N, Silvester N, Jayathilaka S, Lund O, Dynovski LD, Pataki BÁ, Visontai D, Xavier BB, Alako BTF, Belka A, Cisneros JLB, Cotten M, Haringhuizen GB, Harrison PW, Höper D, Holt S, Hundahl C, Hussein A, Kaas RS, Liu X, Leinonen R, Malhotra-Kumar S, Nieuwenhuijse DF, Rahman N, Dos S Ribeiro C, Skiby JE, Schmitz D, Stéger J, Szalai-Gindl JM, Thomsen MCF, Cacciò SM, Csabai I, Kroneman A, Koopmans M, Aarestrup F, Cochrane G. The COMPARE Data Hubs. Database (Oxford). 2019 Jan;2019 . doi:10.1093/database/baz136. PMID: 31868882; PMCID: PMC6927095.

Karsch-Mizrachi I, Takagi T, Cochrane G, International Nucleotide Sequence Database Collaboration. The international nucleotide sequence database collaboration. Nucleic Acids Res. 2018 Jan;46(D1) D48-D51. doi:10.1093/nar/gkx1097. PMID: 29190397; PMCID: PMC5753279.

Sadedin,S.P. *et al.* (2012) Bpipe: A tool for running and managing bioinformatics pipelines. *Bioinformatics*, **28**, 1525–1526.

Di Tommaso,P. *et al.* (2017) Nextflow enables reproducible computational workflows. *Nat. Biotechnol.*, **35**, 316–319.

Vivian,J. *et al.* (2017) Toil enables reproducible, open source, big biomedical data analyses. *Nat. Biotechnol.*, **35**, 314–316.