D7.2 Interoperable data processing services for environmental RI projects: prototype

WORK PACKAGE 7 – Data processing and analysis

LEADING BENEFICIARY: National Research Council of Italy (CNR)

Author(s):	Beneficiary/Institution
Leonardo Candela, Roberto Cirillo, Gianpaolo Coro, Pasquale Pagano, Giancarlo Panichi	National Research Council of Italy (CNR)

Accepted by: Zhiming Zhao (Data for Science Theme Leader)

Deliverable type: DEMONSTRATOR

Dissemination level: PUBLIC

Deliverable due date: 31.10.2018/M42

Actual Date of Submission: 12.11.2018/M43

A document of ENVRI^{plus} project - www.envri.eu/envriplus

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 654182

ABSTRACT

This deliverable documents the implementation of the data processing solution defined in D7.1 "Interoperable data processing for environmental RIs projects: system design". The actual deliverable consists of the software realising the envisaged solution and the instances of it made available to the ENVRI community in the large. The distinguishing features of the proposed solution are (a) to be suitable for ***serving the needs of scientists involved in ENVRI RIs***, (b) to be ***open and extensible*** both with respect to the algorithms and methods it enables and the computing platforms it relies on to execute those algorithms and methods, (c) to be ***open-science-friendly***, i.e. it is capable of incorporating every algorithm and method integrated into the data processing framework as well as any computation resulting from the exploitation of integrated algorithms into a "research object" catering for citation, reproducibility, repeatability and provenance. The proposed solution is part of a larger software system named gCube and has been provisioned to ENVRI RIs via several Virtual Research Environments operated by D4Science.

PROJECT INTERNAL REVIEWER(S)

Project internal reviewer(s):	Beneficiary/Institution
Markus Stocker	University of Bremen

Document history:

Date	Version
01.10.2018	Release of the first version of the deliverable (this document)
15.10/2018	Comments and suggestions received
12.11.2018	Release of the revised version

DOCUMENT AMENDMENT PROCEDURE

Amendments, comments and suggestions should be sent to the editor (Leonardo Candela leonardo.candela@isti.cnr.it).

TERMINOLOGY

A complete project glossary is provided online here: envriplus.manageprojects.com/s/text-documents/LFCMXHHCwS5hh

PROJECT SUMMARY

ENVRIplus is a Horizon 2020 project bringing together Environmental and Earth System Research Infrastructures, projects and networks together with technical specialist partners to create a coherent, interdisciplinary and interoperable cluster of Environmental Research Infrastructures across Europe. It is driven by three overarching goals: 1) promoting cross-fertilization between RIs, 2) implementing innovative concepts and devices across RIs, and 3) facilitating research and innovation in the field of environmental understanding and decision-making for an increasing number of users outside the RIs.

ENVRIplus aligns its activities to a core strategic plan where sharing multi-disciplinary expertise will be most effective. The project aims to improve Earth observation monitoring systems and strategies, including actions to improve harmonization and innovation, and generate common solutions to many shared information technology and data related challenges. It also seeks to harmonize policies for access and provide strategies for knowledge transfer amongst RIs. ENVRIplus develops guidelines to enhance trans-disciplinary use of data and data-products supported by applied use-cases involving RIs from different domains. The project coordinates actions to improve communication and cooperation, addressing Environmental RIs at all levels, from management to end-users, implementing RI-staff exchange programs, generating material for RI personnel, and proposing common strategic developments and actions for enhancing services to users and evaluating the socio-economic impacts.

ENVRIplus is expected to facilitate structuration and improve quality of services offered both within single RIs and at the inter-RI (European and Global) level. It promotes efficient and multi-disciplinary research offering new opportunities to users, new tools to RI managers and new communication strategies for environmental RI communities. The resulting solutions, services and other project outcomes are made available to all environmental RI initiatives, thus contributing to the development of a coherent European RI ecosystem.

TABLE OF CONTENTS

TABLE OF FIGURES

1 Introduction

ENVRIplus WP7 'Data processing and analysis' is called to design and develop a technical solution for data analytics that is suitable for the needs and contexts arising in environmental Research Infrastructures. In particular, Task 7.1. 'Interoperable Data Processing, Monitoring and Diagnosis' is devised to design and develop a solution for data processing aiming at making it significantly easier for scientists to conduct a range of experiments and analyses upon a great variety of data. Expanding the common data processing workflow modelled in the ENVRI project, this task focuses on the engineering and technological aspects of managing the entire lifecycle of computing tasks and application workflows for the efficient utilisation of services and facilities offered by existing e-Infrastructures. Distinguishing features of the service include enabling scientists to enrich the data processing environment by easily injecting new algorithms and methods to be also reused by others. Algorithms and methods can be produced by using programming languages (e.g. Java) or scripting languages scientists are familiar with (e.g. R scripts). The objective of this task is to provide common and cost-effective data processing services for environmental RIs with consideration of existing technologies in e-Infrastructures, data infrastructures and other relevant RIs, building on recent advances in data-intensive computation.

Data Processing or Analytics is an extensive domain including any activity or process that performs a series of actions on dataset(s) to distil information [Bordawekar et al. 2014]. It may be applicable at any stage in the data life cycle, from QA to seismic event recognition, close to data acquisition to transformations and visualisations tailored for decision makers as results are presented. Data analytics methods draw on multiple disciplines including statistics, quantitative analysis, data mining, and machine learning. Very often these methods require compute-intensive infrastructures to produce their results in a suitable time, because of the data to be processed (e.g., huge in volume or heterogeneity) and/or because of the complexity of the algorithm/model to be elaborated/projected. Moreover, being devised to analyse dataset(s) and produce other "data"/information (than can be considered a dataset) these methods are strongly characterised by the "typologies" of their inputs and outputs. In some data-intensive cases, the data handling (access, transport, IO and preparation) can be a critical factor in achieving results within acceptable costs.

In ENVRIplus D7.1 [Candela et al. 2017] there was a long and detailed discussion aiming at setting the scene before proposing a design of the planned solution. In particular, D7.1 contains a description of (a) the existing technologies and solutions for data processing (including workflow management systems and data processing frameworks and platforms); (b) the envisaged data-processing-related patterns captured by the ENVRIplus reference model; and (c) the existing solutions seven environmental Research Infrastructures have currently in place for satisfying their data processing needs. The second part of the deliverable presented the platform for data processing that has been specifically conceived to complement the offering of existing solutions and meet some needs arising in environmental Research Infrastructure. This platform is characterised by the following features: (a) it is suitable for ***serving the needs of scientists involved in ENVRI RIs***, (b) it is ***open and extensible*** both with respect to the algorithms and methods it enables and the computing platforms it relies on to execute the algorithms and methods, (c) it is ***open-science-friendly***, i.e. it is capable to transform every algorithm and method integrated into the data processing framework as well as any computation resulting from the exploitation of integrated algorithms into a "research object" catering for citation, reproducibility, repeatability, provenance, etc. For the sake of completeness, a summary of the design decisions will be given in Section 2.

This Deliverable completes the characterisation of the ENVRIplus Data Analytics by reporting on the implementation of the envisaged solution and showcasing how the technology has been exploited to serve some use cases and scenarios emerged during the ENVRIplus activity [Chen et al. 2018]. In the reality it is a description accompanying the actual deliverable that being of type “Demonstrator” manifests in the software realizing the proposed solution and its instances made available to the ENVRIplus community by Virtual Research Environments (cf. Section 3).

The remainder of this deliverable is organised as follows. Section 2 reports a brief description of the data analytics platform by presenting the architecture, the main functionalities it offers, and the possible exploitation models. Section 3 describes how the proposed solution (software and its instances) has been made available to the ENVRIplus community. Finally, Section 4 concludes the report.

2 ENVRIplus Data Processing Platform

The ENVRIplus data processing platform was not conceived to be developed from scratch. The rationale behind this is manifold including (i) there are so many solutions for data processing that developing a completely new one is neither feasible (because of the resources and time) nor reasonable (existing solutions are mainly conceived to be open and extensible), (ii) there will never be a single solution suitable for any application context or community, (iii) there are a plethora of solutions and e-Infrastructures to leverage on.

The ENVRIplus data processing platform stemmed from the data processing solution developed during the ENVRI project. The new data processing platform, named DataMiner, is an ***open-source computational system*** part of the ***gCube system*** [Assante et al. 2018].

From the end user perspective, it offers a **collaborative-oriented working environment** where users:

- can easily **execute and monitor data analytics tasks** by relying on a rich and open set of available methods either by using a dynamically generated **web-based user-friendly GUI** or by using a RESTful protocol based on the **OGC WPS Standard**;
- can easily **share & publish their analytics methods** (e.g. implemented in R, Java, Python, etc.) to the workbench and make them exploitable by an automatically generated web-based GUI and by the OGC WPS protocol;
- are provided with a **"research object"[1] describing every analytics task** executed by the workbench enabling for **repeatability, computational reproducibility, reuse, citation** and **provenance**. These research objects are a set of files organized in folders and containing every input & output, an executable reference to the method as well as rich metadata including a PROV-O provenance record;

The data analytics framework is integrated with a **shared workspace** where the research objects resulting from the analytics tasks are automatically stored together with rich metadata. Objects in the workspace can be shared with coworkers as well as published by a catalogue with a license governing their uses. Moreover, the framework is conceived to operate in the context of one or more **Virtual Research Environments**, i.e. it is actually made available by a dedicated working environment offering (besides the framework and the workspace) additional services including those for managing users, creating communities, and supporting communication and collaboration among VRE members.

The data analytics framework is conceived to give access to two typologies of resource:

- a **distributed, open & heterogeneous computing infrastructure** for the real execution of the analytics tasks. This distributed computing infrastructure is capable to exploit resources from the EGI infrastructure.
- the pool of **methods** integrated in the platform, i.e. each method integrated in the framework is made available as-a-Service to other users according to the specific policy.

2.1 ENVRIplus Data Processing Architecture

The high-level architecture of the Data Processing is Platform is depicted in Figure 1 and Figure 2.

[1] These are packages of files worth being considered as a unit from the research activity perspective, e.g. they are expected to contain the entire set of data and metadata needed to capture a research activity and its results. From a conceptual point of view this is equal to the objects characterizing the http://www.researchobject.org/ initiative.

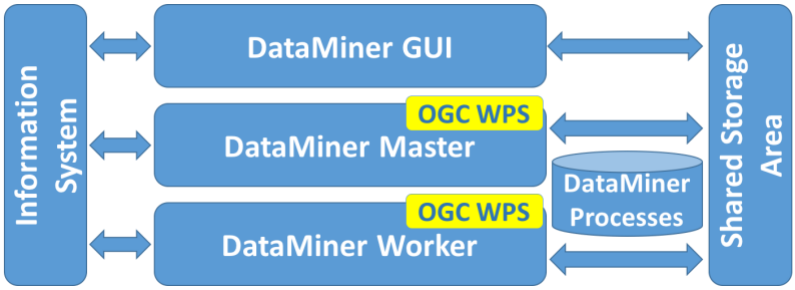

FIGURE 1. DATAMINER OVERALL ARCHITECTURE

The system consists of the following components:

- The ***DataMiner GUI***: a web-based user interface enacting users to select an existing process, execute it, monitor the execution and access to the results (cf. Sec. 2.2);
- The ***DataMiner Master***: this web service is in charge to accept requests for executing processes and executing them, either locally or by relying on the DataMiner Worker(s) depending from the specific process. The service is conceived to work in a cluster of replica services and is offered by a standard web-based protocol, i.e. OGC WPS;
- The ***DataMiner Worker***: this web service is in charge to execute the processes it is assigned to. The service is conceived to work in a cluster of replica services and is offered by a standard web-based protocol, i.e. OGC WPS;
- The ***DataMiner Processes***: this a repository of processes the platform is capable to execute. This repository if equipped with a set of off-the-shelf processes and it can be further populated with new processes either (a) developed from scratch in compliance with a specific API or (b) resulting from annotating existing processes (cf. Sec. 2.3).

These components are glued together thanks to (a) an information system that enable each service instance to dynamically discover other existing service instances and be informed on their capabilities and (b) a shared storage area the service instances use to exchange the data they are operating on, e.g. the datasets to be processed or the datasets resulting from a processing activity.

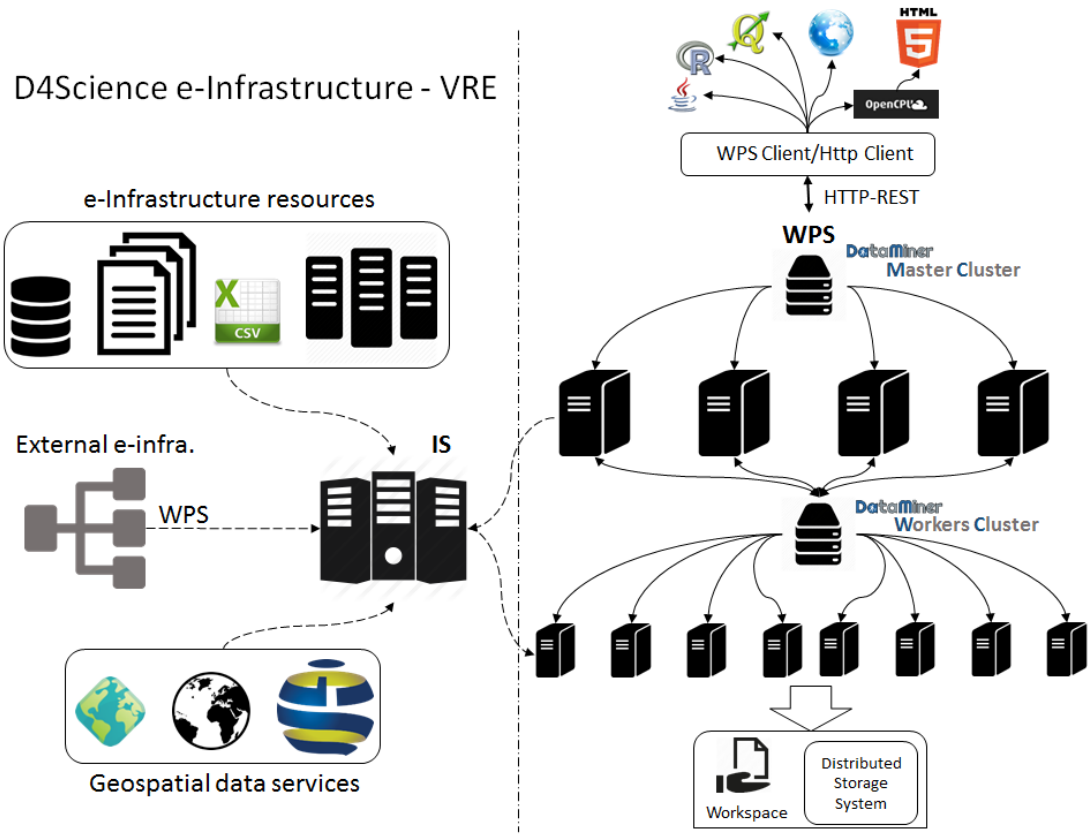

FIGURE 2. DATAMINER DATA PROCESSING SYSTEM

Figure 2 is oriented to describe how the system works by highlighting how the various instances interact. In a typical deployment scenario, the Master cluster is made up of a number of machines managed by a load balancer that distributes the requests uniformly to the machines hosting DataMiner Master instances. Each machine is endowed with a DM service that communicates with the Information System. The balancer is indexed on the IS and is the main access point to interact with the DMs. The machines of the Worker cluster have less local computational power and serve distributed computations. DataMiner is based on the 52North WPS implementation[2]. It is developed in Java and the Web service runs on an Apache Tomcat instance endowed with gCube system libraries.

When a WPS request comes to the Master cluster balancer, it is distributed to one of the DataMiner Master instances forming the cluster. Each DataMiner instance host the processes the service is capable to execute, these processes are likely to be provided by several developers and providers. In particular, two kinds of algorithms are hosted: “local” and “cloud” algorithms. Local algorithms are directly executed on the DataMiner Master instances and possibly use parallel processing on several cores and a large amount of memory. In contrast, cloud algorithms use distributed computing with a Map-Reduce approach and rely on the DataMiner Worker instances in the Worker cluster.

All the DataMiner instances (be them Masters or Workers) are exposed by the OGC WPS protocol. However, DataMiner WPS implementation adds a number of features to the standard 52North implementation it relies on. First, it returns a different list of processes according to the “application context” in which the service is invoked. In fact, the overall DataMiner framework is conceived to operate in multi-tenancy settings where the same instance can serve many “application contexts” (cf. Sec. 2.4).

[2] 52North. The 52north wps service 2016. http://52north.org/communities/geoprocessing/wps/

Whenever an algorithm is added to the DataMiner Processes repository, it is conceptually added to all the instances of the DataMiner relying on the specific repository. However, a specific DataMiner instance will expose and make executable only the processes it is configured to make available in a specific "application context". This approach is flexible and well suited with the Virtual Research Environment development model envisaged by gCube [Assante et al. 2018].

The exposition of DataMiner processes by the WPS standard allows a number of thin clients to use the processes and to transparently rely on a distributed computing infrastructure to execute them. Third party software (e.g. the well-known QGIS[3] and ArcMap[4] for geospatial data manipulation) can be able to retrieve the capabilities of a WPS service and run remote processes. Further, clients for R and Java have been developed[5] as well as the WPS service can manage HTTP-GET requests (thus, a process can also be invoked by using a common Web browser). Finally, by relying on OpenCPU[6] it is possible to transform WPS objects into Javascript objects and allows for fast building of HTML applications.

For authentication and authorization, DataMiner relies on a token-based mechanism[7] [Assante et al. 2018], i.e. the user is requested to acquire a "valid token" before being authorized to execute processes. This token is passed via basic HTTPS-access authentication, which is supported by most WPS and HTTP(S) clients. The token identifies both a user and an "application context" and this information is used by DM to query the IS about the capabilities to be offered in that context, i.e. the processes the user will be able to invoke with that authorization.

The DataMiner is conceived to rely on a shared workspace built on top of the storage area. In particular, it interfaces with the gCube-based workspace [Assante et al. 2018] for accessing input data. Inputs can also come from external repositories, because a file can be provided either as an HTTP link or embedded in a WPS execution request. The outputs of the computations are written onto the Distributed Storage System and are immediately returned to a client at the end of the computation. Afterwards, an independent thread also writes this information on the Workspace. Indeed, after a completed computation, a Workspace folder is created which contains the input, the output, the parameters of the computation, and a provenance document summarizing this information. This folder can be shared with other people and used to execute the process again. Thus, the complete information about the execution can be shared and reused. This is the main way by which DataMiner fosters collaborative experimentation. The DataMiner processes can access the resources available in an "application context" by querying the IS. For example, it is possible to discover geospatial services, maps, databases, and files. The DataMiner Java development framework simplifies the interaction with the IS. Since the IS interface is HTTP REST too, it could be managed by the processes directly. Further, the DataMiner development framework provides methods to transform heterogeneous GIS formats into a numeric matrix and thus simplifies the effort to process geospatial data.

DataMiner can also import processes from other WPS services. If a WPS service is indexed on the IS for a certain "application context", its processes descriptions are automatically harvested, imported, and published among the DataMiner capabilities for that "application

[3] QGIS. A free and open source geographic information system 2016. http://qgis.org/en/site/

[4] ArcMap. Arcgis for desktop 2016. http://desktop.arcgis.com/en/arcmap/

[5] National Research Council of Italy. gCube wps thin clients 2016. https://wiki.gcube-system.org/gcube/How_to_Interact_with_the_DataMiner_by_client

[6] OpenCPU. Producing and reproducing results 2016. https://www.opencpu.org

[7] National Research Council of Italy. gCube token-based authorization system 2016. https://wiki.gcube-system.org/gcube/Authorization_Client_Library

context". During a computation, DataMiner acts as a bridge towards the external WPS systems. Nevertheless, DataMiner adds provenance management, authorization, and collaborative experimentation to the remote services, that being standard WPS do not support any of them.

2.2 The DataMiner web-based GUI

DataMiner offers a web-based GUI to its users (Figure 3).

On the left panel (Figure 3 a), the GUI presents the list of capabilities available in the specific "application context", which are semantically categorised (the category is indicated by the process provider). For each capability, the interface calls the WPS *DescribeProcess* operation to get the descriptions of the inputs and outputs. When a user selects a process, in the right panel the GUI on-the-fly generates a form with different fields corresponding to the inputs. Input data can be selected from the Workspace (the button associated to the input opens the Workspace selection interface). The "Start Computation" button sends the request to the DM Master cluster, which is managed as explained in the previous section. The usage and the complexity of the Cloud computations are completely hidden from the user, but the type of the computation is reported as a metadata in the provenance file.

A view of the results produced by the computations is given in the "Check the Computations" area (Figure 3 b), where a summary sheet of the provenance of the experiment can be obtained ("Show" button, Figure 3 c). From the same panel, the computation can be also re-submitted. In this case, the Web interface reads the XML file containing the PROV-O information associated to a computation and rebuilds a computation request with the same parameters. The computation folders may also include computations executed and shared by other users.

Finally, the "Access to the Data Space" button allows obtaining a list of the overall input and output datasets involved in the executed computations (Figure 3 d), with provenance information attached that refers to the computation.

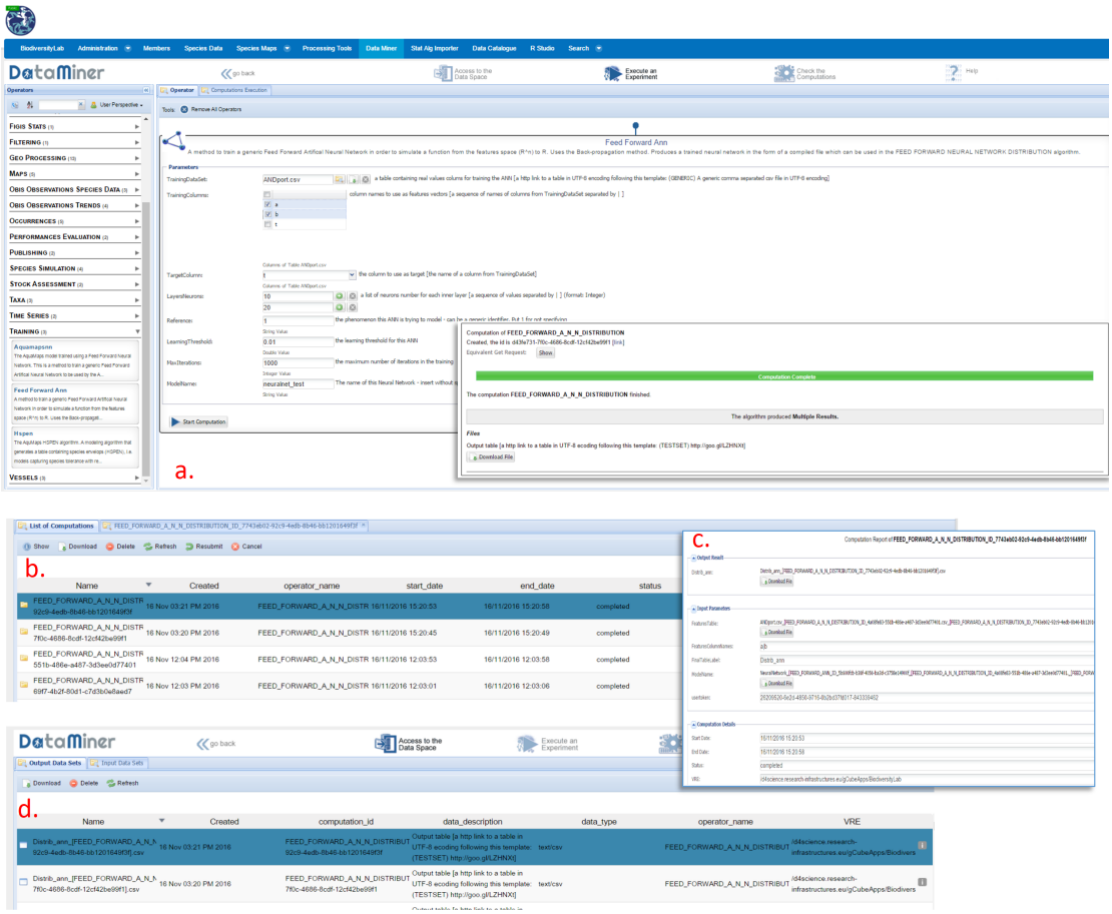

FIGURE 3. INTERFACE OF THE GCUBE DATAMINER SYSTEM

2.3 Importing new processes

Prototype scripting is the base of most models in environmental sciences. Scientists making prototype scripts (e.g. using R and Matlab) often need to share results and provide their models and methods for use by other scientists. They will encounter new data and may run in different contexts, which may require careful engineering to accommodate the wider scope. To help meet this aim, DataMiner lets them publish scripts as-a-Service, possibly under a recognized standard (e.g. WPS). The Statistical Algorithms Importer (SAI) is an interface that allows scientists to easily and quickly import R scripts onto DataMiner. DataMiner in turn publishes these scripts as-a-Service and manages multi-tenancy and concurrency. Additionally, it allows scientists to update their scripts without following long software re-deploying procedures each time. In summary, SAI produces processes that run on the DataMiner Cloud computing platform and are accessible via the WPS standard.

The SAI interface (Figure 4) resembles the R Studio environment, a popular IDE for R scripts, in order to make it friendly to script providers.

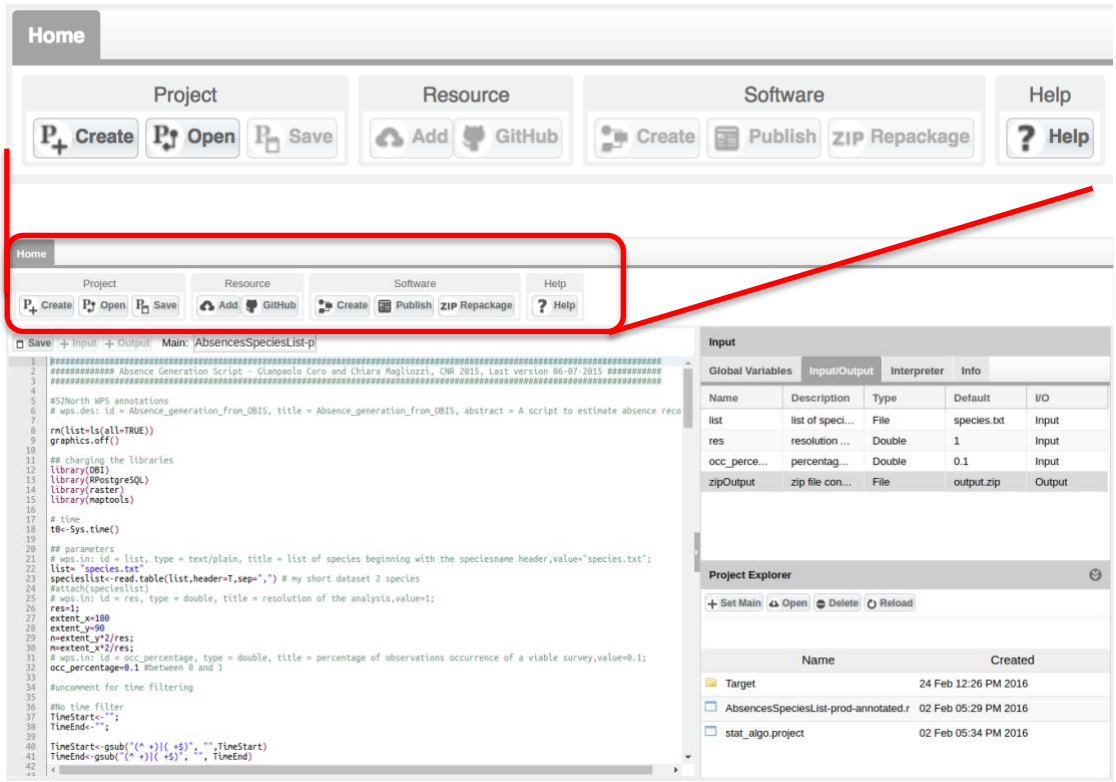

FIGURE 4. INTERFACE TO IMPORT NEW METHODS IN DATAMINER

The *Project* button allows creating, opening and saving a working session. A user uploads a set of files and data on the workspace area (lower-right panel). Upload can be done by dragging and dropping local desktop files. As a next step, the user indicates the "main script", i.e. the script that will be executed on DataMiner and that will use the other scripts and files. After selecting the main script, the left-side editor panel visualises it with R syntax highlighting and allows modifying it. Afterwards, the user indicates the input and output of the script by highlighting variable definitions in the script and pressing the *+Input* (or *+Output*) button: behind the scenes the application parses the script strings and guesses the name, description, default value and type of the variable. This information is visualised in the top-right side *Input/Output* panel, where the user can modify the guessed information. Alternatively, SAI can automatically compile the same information based on WPS4R[8] annotations in the script. Other tabs in this interface area allow setting global variables and adding metadata to the process. In particular, the *Interpreter* tab allows indicating the R interpreter version and the packages required by the script and the *Info* tab allows indicating the name of the algorithm and its description. In the *Info* tab, the user can also specify the VRE in which the algorithm should be available.

Once the metadata and the variables information have been compiled, the user can create a DataMiner as-a-Service version of the script by pressing the *Create* button in the Software panel. The term "software", in this case indicates a Java program that implements an as-a-Service version of the user-provided scripts. The Java software contains instructions to automatically download the scripts and the other required resources on the server that will execute it, configure the environment, execute the main script and return the result to the user. The computations are orchestrated by the DataMiner computing platform that ensures the program has one instance for each request and user. The servers will manage concurrent

[8] 52North. WPS4R. https://wiki.52north.org/Geostatistics/WPS4R

requests from several users and execute code in a closed sandbox folder, to avoid damage caused by malicious code.

Based on the SAI Input/Output definitions written in the generated Java program, DataMiner automatically creates a Web GUI (cf. Section 2.2).

By pressing the *Publish* button, the application notifies DataMiner that a new process should be deployed. DataMiner will not own the source code, which is downloaded on-the-fly by the computing machines and deleted after the execution. This approach meets the policy requirements of those users who do not want to share their code.

The *Repackage* button re-creates the software so that the computational platform will be using the new version of the script. The repackaging function allows a user to modify the script and to immediately have the new code running on the computing system. This approach separates the script updating and deployment phases, making the script producer completely independent on e-Infrastructure deployment and maintenance issues. However, deployment is necessary again whenever Input/Output or algorithm's metadata are changed.

To summarise, the SAI Web application enables an R script with as-a-Service features. SAI reduces integration time with respect to direct Java code writing. Additionally, it adds (i) multi-tenancy and concurrent access, (ii) scope and access management through Virtual Research Environments, (iii) output storage on a distributed, high-availability file system, (iv) graphical user interface, (v) WPS interface, (vi) data sharing and publication of results, (vii) provenance management, and (viii) accounting facilities.

2.4 Exploitation Scenarios

DataMiner is conceived to support the following scenarios / exploitation models:

- ***Full platform as-a-Service***: the entire DataMiner platform is operated by a service provider (e.g. D4Science) and the community / Research Infrastructure establishes a collaboration agreement to use it by well-defined service level agreements. In this scenario, the service provider can establish its own collaboration agreements with other research infrastructures to operate the service it is responsible for (e.g. this is the case of D4Science that has established a collaboration agreement with EGI to deploy some DataMiner instances on EGI sites);
- ***Full platform as-a-Software***: a community / Research Infrastructure can decide to exploit the DataMiner technology (open source) to set up its own instance of the technology. In this case, the community / Research Infrastructure faces hardware resource costs needed to operate the platform, as well as IT personnel costs needed to deploy and operate the technology. In this case, the community can set up an agreement with other infrastructures (e.g. EGI) thus to reduce the costs related with hardware resources. Still, the costs related to DataMiner technology deployment and operation remain;
- ***Platform as-a-Service with Community Contribution***: the DataMiner core components are operated by a service provider and the platform is complemented by some instances (namely, the workers) operated by the Research Infrastructure on its own resources. This make it possible for RIs, to deploy these nodes close to where the data to be processed are actually stored.

3 ENVRIplus Data Processing in Action

The ENVRIplus Data Processing technology has been developed and released in several versions. Regarding the software, Section 3.1 describes how it has been made available via several repositories. Regarding "prototypes", Sections 3.2-3.6 describes how the technology has been deployed by the D4Science infrastructure to provide ENVRIplus communities with fully operational Virtual Research Environments enabling to experience and make use of the technology.

3.1 Software Documentation and Availability

The ENVRIplus Data Processing solution is part of a larger software system, gCube [Assante et al. 2018]. The primary components of this solution are DataMiner and Statistical Algorithms Importer. A complete and constantly updated documentation on such technologies is made available by dedicated Wiki pages hosted by the gCube Wiki. In particular:

- DataMiner Wiki page https://wiki.gcube-system.org/gcube/Data_Mining_Facilities
- Statistical Algorithm Importer https://wiki.gcube-system.org/gcube/Statistical_Algorithms_Importer

The software has been made available by several channels including:

- The gCube System website: https://www.gcube-system.org/software-releases
- The GitHub gCube Releases Repository https://github.com/gcube-team/gcube-releases/releases
- The Zenodo gCube System community https://zenodo.org/communities/gcube-system

3.2 The EISCAT Virtual Research Environment

The EISCAT Virtual Research Environment has been deployed to implement a demonstrator [Chen et al. 2018] addressing a requirement of the EISCAT RI community, namely to allow individual scientists to process their experimental data using their own algorithms. The challenge is common to many ENVRIplus RIs, where data is often processed using standard models and methods. As researchers want to use different analysis models, easily modify parameters or algorithms, and collaborate with each other, they need a Virtual Research Environment (VRE).

The Virtual Research Environment is available at https://services.d4science.org/group/eiscat while a screenshot of the DataMiner instance serving this VRE is in Figure 5.

A demo showcasing the VRE in action is available at https://youtu.be/YEEMUvnSHUM

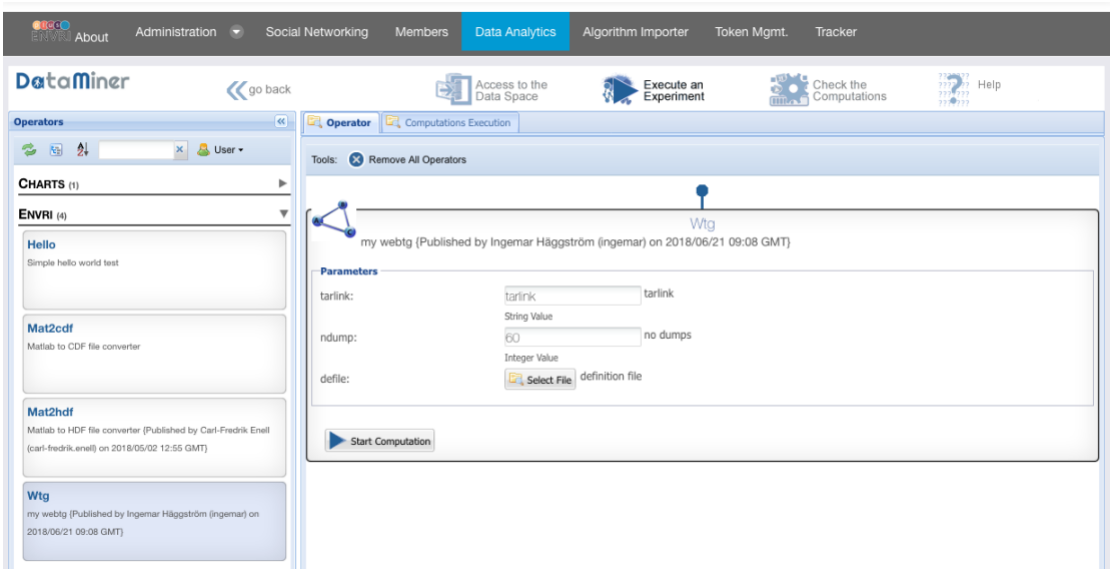

FIGURE 5. EISCAT VRE: DATAMINER SCREENSHOT

3.3 The ENVRIplus Virtual Research Environment

The ENVRIplus Virtual Research Environment has been deployed since the early phases of the ENVRIplus project to provide the project members with an environment showcasing the latest developments of the proposed solution and enact them to experience with it without the need to install any technology or acquire any computing capacity. Moreover, the DataMiner serving this VRE has been configured to offer a number of ready to use algorithms including data clustering methods, time series analysis methods, and geospatial data processing including:

- ESRI Grid Extraction: An algorithm to extract values associated to an environmental feature repository (e.g. NETCDF, ASC, GeoTiff files). A grid of points at a certain resolution is specified by the user and values are associated to the points from the environmental repository. It accepts as one geospatial repository ID or a direct link to a file and the specification about time and space. The algorithm produces one ESRI GRID ASCII file containing the values associated to the selected bounding box.
- Maps comparison: An algorithm for comparing two OGC/NetCDF maps in seamless way to the user. The algorithm assesses the similarities between two geospatial maps by comparing them in a point-to-point fashion. It accepts as input the two geospatial maps and some parameters affecting the comparison such as the z-index, the time index, the comparison threshold. Note: in the case of WFS layers it makes comparisons on the last feature column.
- Time extraction: An algorithm to extract a time series of values associated to a geospatial features repository (e.g. NETCDF, ASC, GeoTiff files). The algorithm analyses the time series and automatically searches for hidden periodicities. It produces one chart of the time series, one table containing the time series values and possibly the spectrogram.
- Time extraction table: An algorithm to extract a time series of values associated to a table containing geospatial information. The algorithm analyses the time series and automatically searches for hidden periodicities. It produces one chart of the time series, one table containing the time series values and possibly the spectrogram.
- X Y extractor: An algorithm to extract values associated to an environmental feature repository (e.g. NETCDF, ASC, GeoTiff files). A grid of points at a certain resolution is specified by the user and values are associated to the points from the environmental repository. It accepts as one geospatial repository ID or a direct link to a file and the

specification about time and space. The algorithm produces one table containing the values associated to the selected bounding box.

- X Y extractor table: An algorithm to extract values associated to a table containing geospatial features (e.g. Vessel Routes, Species distribution maps). A grid of points at a certain resolution is specified by the user and values are associated to the points from the environmental repository. It accepts as one geospatial table and the specification about time and space. The algorithm produces one table containing the values associated to the selected bounding box.
- Z extraction: An algorithm to extract the Z values from a geospatial features repository (e.g. NETCDF, ASC, GeoTiff files). The algorithm analyses the repository and automatically extracts the Z values according to the resolution wanted by the user. It produces one chart of the Z values and one table containing the values.
- Z extraction table: An algorithm to extract a time series of values associated to a table containing geospatial information. The algorithm analyses the time series and automatically searches for hidden periodicities. It produces one chart of the time series, one table containing the time series values and possibly the spectrogram.
- Raster data publisher: This algorithm publishes a raster file as a maps or datasets in the D4Science SDI. NetCDF-CF files are encouraged, as WMS and WCS maps will be produced using this format. For other types of files (GeoTiffs, ASC etc.) only the raw datasets will be published.
- Shapefile publisher: An algorithm to publish shapefiles under WMS and WFS standards in the D4Science SDI. The produced WMS, WFS links are reported as output of this process.

The Virtual Research Environment is available at https://services.d4science.org/group/envriplus while a screenshot of the DataMiner instance serving this VRE is in Figure 6.

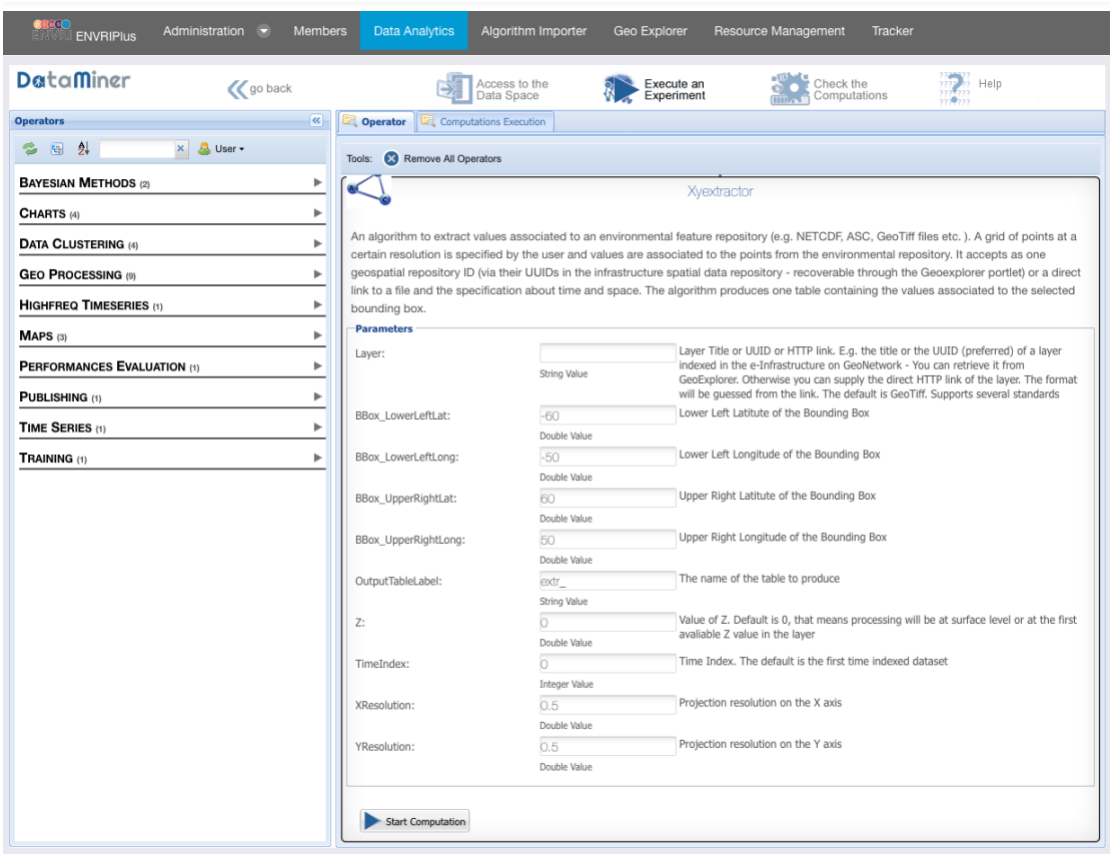

FIGURE 6. ENVRIPLUS VRE: DATAMINER SCREENSHOT

3.4 The ENVRIplus Data4Science Virtual Research Environment

The ENVRIplus Data4Science has been deployed to play the role of long term showcasing environment for all the solutions stemming from ENVRIplus Theme 2 activities. In fact, this VRE is equipped also with a catalogue aiming at enabling the users to browse the portfolio of the available services.

The Virtual Research Environment is available at https://services.d4science.org/group/envriplusdata4science while a screenshot of the DataMiner instance serving this VRE is in Figure 7.

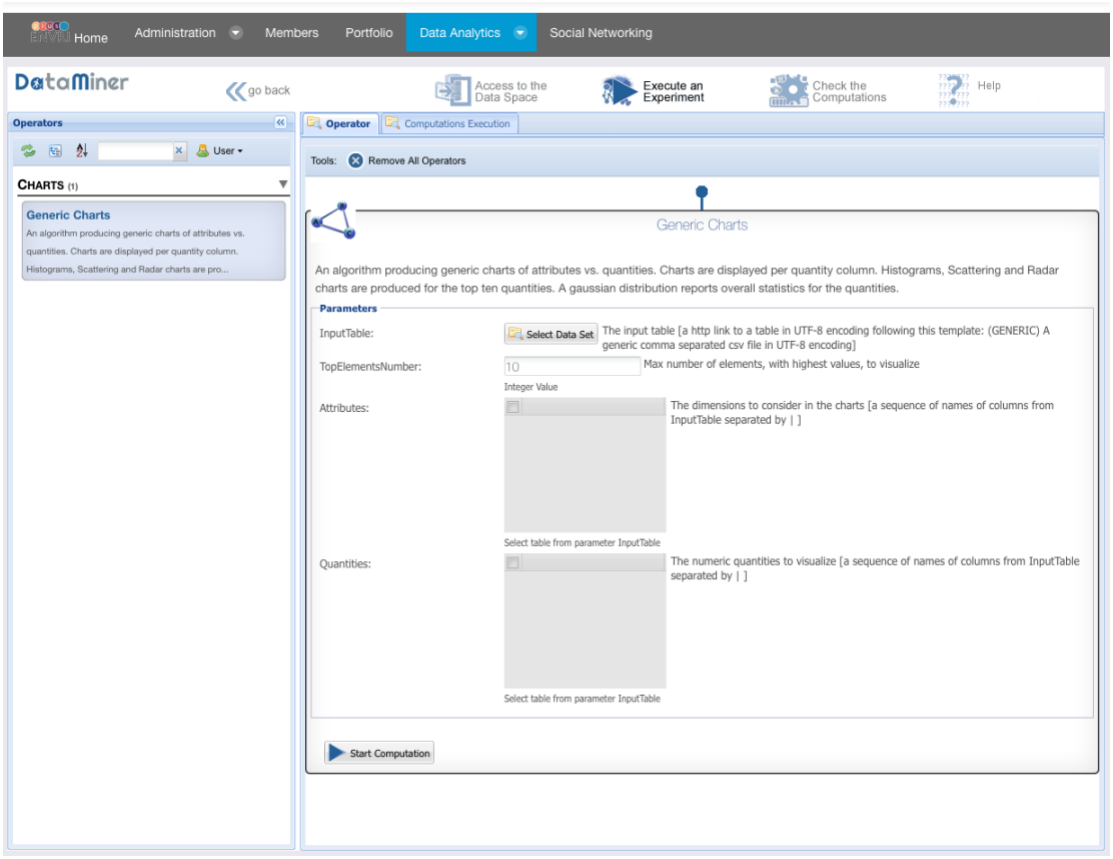

FIGURE 7. ENVRIPLUS DATA4SCIENCE VRE: DATAMINER SCREENSHOT

3.5 The ICOS Eddy Covariance Processing Virtual Research Environment

The ICOS Eddy Covariance Processing Virtual Research Environment has been deployed to implement a demonstrator [Chen et al. 2018] showcasing a novel implementation of a computationally efficient tool for processing of Eddy Covariance (EC) data which offers to users the possibility to calculate EC fluxes through the EddyPro® software according to 4 processing schemes resulting from a different combination of existing methods.

The Virtual Research Environment is available at https://services.d4science.org/group/icoseddycovarianceprocessing while a screenshot of the DataMiner instance serving this VRE is in Figure 8.

A demo showcasing the VRE in action is available at https://youtu.be/hod2WksKzV8

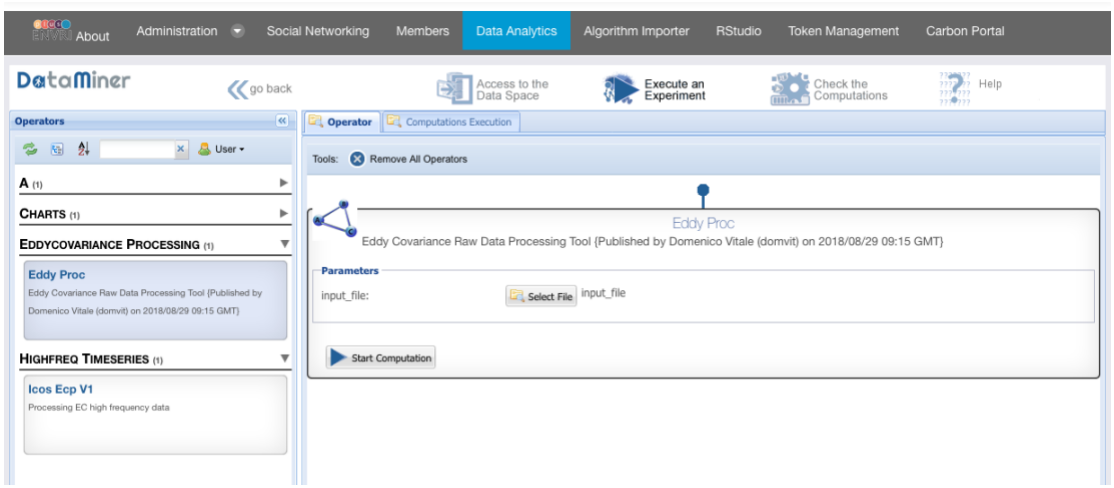

FIGURE 8. ICOS EDDY COVARIANCE PROCESSING VRE: DATAMINER SCREENSHOT

3.6 The Particle Formation Virtual Research Environment

The Particle Formation Virtual Research Environment has been deployed to implement a demonstrator [Chen et al. 2018] supporting aerosol scientists in studying new atmospheric particle formation events by moving data analysis from local computing environments to interoperable infrastructures, thus harmonizing data analysis itself and more importantly the syntax and semantics of data derived from analysis. As researchers interpret primary data and thus gain information and transfer information into knowledge, we are studying and advancing in particular some technical aspects of a knowledge infrastructure i.e., a robust network of scientists, artefacts such as virtual research environments and research data, and institutions such as research infrastructures and e-Infrastructures that acquire, maintain and share scientific knowledge about the natural world. The science demonstrator showcases a possible architecture of a socio-technical infrastructure that "transforms data into knowledge." The proposed approach highlights a range of novel possibilities, in particular enabling researchers to focus on data analysis and interpretation while leaving data access and transformation from and to systems to interoperable infrastructure. It significantly contributes to implementing the global agenda of FAIR data by promoting the notion of "FAIR by Design", weaving data FAIRness into the fabric of infrastructures. It builds on the principle not to leave making data FAIR to researchers but to guarantee it by design of well-engineered infrastructures. The demonstrator is first and foremost of primary interest to a specific scientific community, namely the various aerosol research groups that study new particle formation events.

Regarding analytics, a two steps workflow has been implemented by invoking two Data Miner algorithms from within a Jupyter Notebook, programmatically via a WPS (OGC Web Process Service) interface [Chen et al. 2018].

The Virtual Research Environment is available at https://services.d4science.org/group/particleformation while a screenshot of the DataMiner instance serving this VRE is in Figure 9.

A demo showcasing the VRE in action is available at https://youtu.be/ra9W7b5DbgI

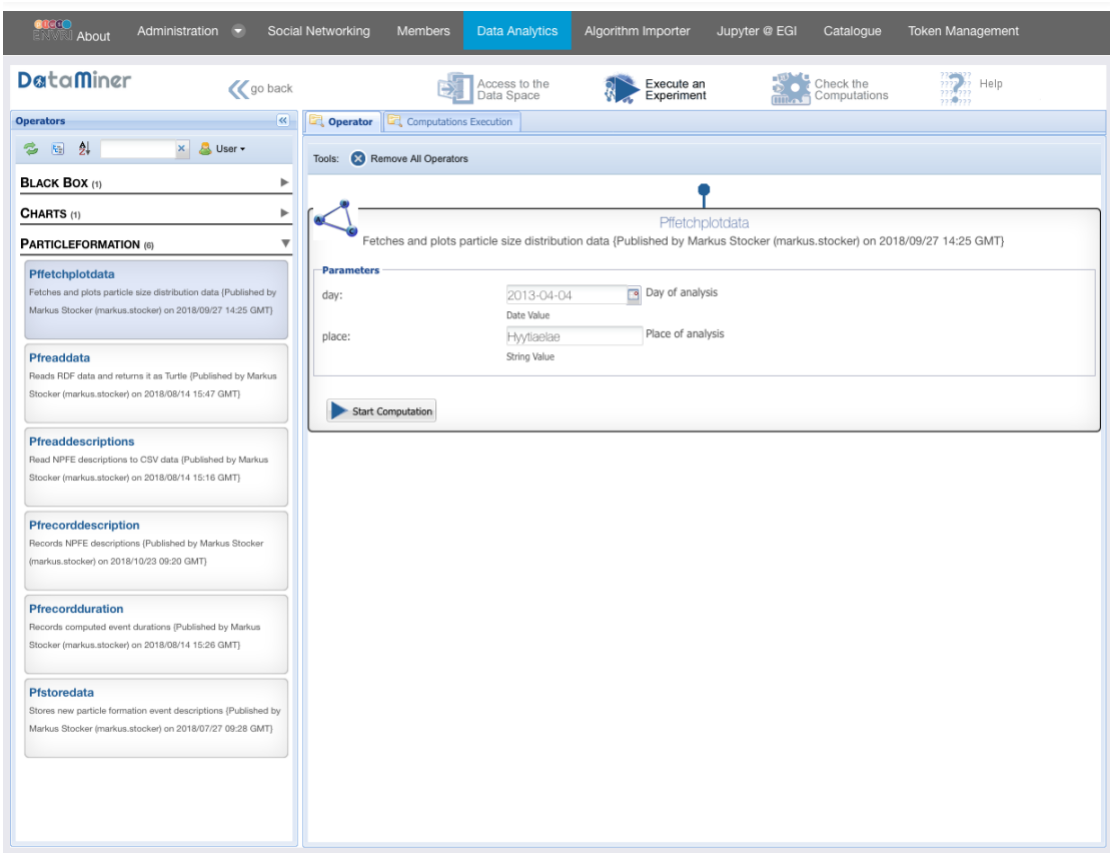

FIGURE 9. PARTICLE FORMATION VRE: DATAMINER SCREENSHOT

4 Concluding remarks

This deliverable documents the implementation of the solution envisaged in D7.1 to provide the ENVRIplus community with a data analytics platform with the following distinguishing features (i) *be extensible*, i.e., the platform is "open" with respect to the analytics techniques it offers / support and the computing infrastructures and solutions it relies on to enact the processing tasks; (ii) *promote distributed processing*, i.e. the platform executes processing tasks by relying on "local engines" / "workers" that can be deployed in multiple instances and execute tasks in parallel and seamlessly; (iii) *be offered by multiple interfaces*, i.e., the platform offers its facilities by both a (web-based) graphical user interface and a (web-based) programmatic interface (aka API) in OGC WPS; (iv) *cater for scientific workflows*, i.e., the platform is exploitable by existing WFMS as well as should support the execution of a processing task captured by a workflow specification; (v) *be easy to use*, i.e., the platform is easy to use for both algorithms / method providers and algorithms / method users; (vi) be *open science friendly, i.e.*, the platform transparently inject open science practices (provenance recording, repeatability) in the processing tasks executed through it.

Such a solution is part of the gCube software system [Assante et al. 2018] and made available as-a-Service by the D4Science.org infrastructure.

References

[Assante et al. 2018] Assante, M.; Candela, L.; Castelli, D.; Cirillo, R.; Coro, C.; Frosini, L.; Lelii, L.; Mangiacrapa, F.; Marioli, V.; Pagano, P.; Panichi, G.; Perciante, C.; Sinibaldi, F. (2018) The gCube system: Delivering Virtual Research Environments as-a-Service, Future Generation Computer Systems, doi: 10.1016/j.future.2018.10.035.

[Atkinson et al. 2016] M. Atkinson, A. Hardisty, R. Filgueira, C. Alexandru, A. Vermeulen, K. Jeffery, T. Loubrieu, L. Candela, B. Magagna, P. Martin, Y. Chen, M. Hellström (2016) A consistent characterisation of existing and planned RIs. ENVRIplus Project Deliverable D5.1

[Bordawekar et al. 2014] R. Bordawekar, B. Blainey, C. Apte (2014) Analyzing analytics. *SIGMOD Rec.* 42, 4 (February 2014), 17-28. doi: 10.1145/2590989.2590993

[Candela et al 2013 b] L. Candela, D. Castelli, P. Pagano (2013) Virtual Research Environments: An Overview and a Research Agenda. Data Science Journal, Vol. 12, p. GRDI75-GRDI81 DOI: 10.2481/dsj.GRDI-013

[Candela et al. 2014] L. Candela; D. Castelli; G. Coro; L. Lelii; F. Mangiacrapa; V. Marioli; P. Pagano (2014) An Infrastructure-oriented Approach for supporting Biodiversity Research. Ecological Informatics, Elsevier, 2014, doi: 10.1016/j.ecoinf.2014.07.006

[Candela et al. 2017] Candela, L.; Coro, G.; Pagano, P.; Panichi, G.; Atkinson, M.; Filgueira, R.; Bailo, D.; Enell, C.-F.; Fiebig, M.; Haslinger, F.; Hellström, M.; Vermeulen, A.; Lankreijer, H.; Huber, R.; Joussaume, S.; Guglielmo, F.; Mendez, V. (2017) Interoperable data processing for environmental RI projects: system design. ENVRIplus Project Deliverable D7.1

[Chen et al. 2018] Chen, Y.; Haggstrom, I.; Buck, J.; Stocker, M.; Carval, T.; Vitale, D.; Huber, R.; Hellström, M.; Candela, L.; Haslinger, F. (2018) Service deployment in computing and data e-Infrastructures – Version 2. ENVRIplus Project Deliverable D9.2