



August 2018

reana

USER DASHBOARD FOR REUSABLE ANALYSIS PLATFORM

AUTHOR:

Sinclert Pérez

Digital Repositories (IT-CDA-DR)

SUPERVISOR:

Tibor Simko

Digital Repositories (IT-CDA-DR)



ABSTRACT



REANA is a reusable analysis platform which offers physicists the ability to structure their research data analysis and run their computational workflows in a containerized computing cloud.

The goal of this project was to develop a web interface dashboard in order to improve user interaction with the platform. The dashboard will include an overview of the previously submitted workflows, inspection of their running status and outputs, and job monitoring capabilities.





TABLE OF CONTENTS



1. INTRODUCTION	5
2. REQUIREMENTS	6
3. TECHNOLOGY SELECTION	7
3.1. INTERFACE BUILDING	7
3.2. INTERFACE STYLE	7
3.3. WORKFLOW VISUALIZATION	9
3.4. ADDITIONAL TECHNOLOGIES	10
4. DEVELOPMENT	11
4.1. LOGIN PAGE	11
4.2. WORKFLOW LIST PAGE	12
4.3. WORKFLOW DETAILS PAGE	13
5. CONCLUSION	15
6. REFERENCES	16





LIST OF FIGURES



FIGURE 1: REPRODUCIBILITY FAILURES IN SCIENCES	5
FIGURE 2: LOGIN PAGE WIREFRAME	6
FIGURE 3: WORKFLOW-DETAILS PAGE WIREFRAME	6
FIGURE 4: WORKFLOW-LIST PAGE WIREFRAME	6
FIGURE 6: MATERIAL-UI WORKFLOW-LIST PAGE TECHNICAL MOCK-UP	8
FIGURE 7: SEMANTIC-UI WORKFLOW-LIST PAGE TECHNICAL MOCK-UP	8
FIGURE 5: GROMMET WORKFLOW-LIST PAGE TECHNICAL MOCK-UP	8
FIGURE 8: PART OF A COMPUTATIONAL GRAPH USED IN BSM SEARCH	9
FIGURE 9: EXAMPLE OF A SIMPLE COMPUTATIONAL GRAPH	9
FIGURE 10: SCREENSHOT OF IMPLEMENTED LOGIN PAGE	11
FIGURE 11: SCREENSHOT OF IMPLEMENTED WORKFLOW-LIST PAGE	12
FIGURE 12: SCREENSHOT OF IMPLEMENTED WORKFLOW-DETAILS PAGE	13
FIGURE 13: SCREENSHOT OF ONLINE FILE VISUALIZATION	14





1. INTRODUCTION

Over the last few years there has been a reproducibility crisis in most of the sciences. According to a 2016 Nature’s paper [1]: “More than 70% of researchers have tried and failed to reproduce another scientist’s experiments, and more than half have failed to reproduce their own experiments”. This problem is far from being isolated due to its presence in a variety of fields:

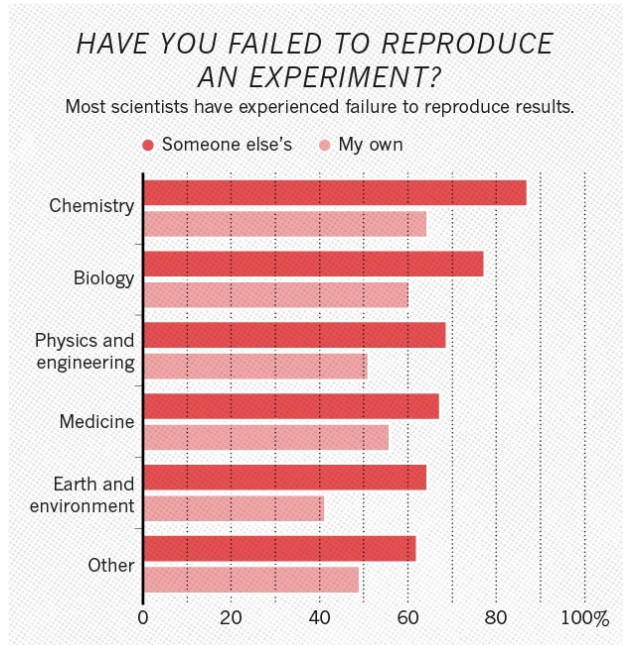


Figure 1: Reproducibility failures in sciences [1]

In order to solve this problem in the area of High Energy Physics (HEP), the Digital Repository section at CERN has developed REANA: a reusable and reproducible research data analysis platform that offers tools to particle physicists to structure their data analysis, so they can be instantiated and run on remote containerized compute clouds.

REANA is composed of a set of components, from which *REANA-Client* [2] is the one providing a Command Line Interface (CLI) in order to create and run data analysis workflows. In addition to the CLI, there was a clear need for workflow status visualization over the web.

This necessity set up the reason for the creation of a new module, *REANA-UI* [3], which will communicate through REST API with the already existing *REANA-Server* [4] module. My responsibility concerned the design and development of this new component.



2. REQUIREMENTS

The first step of the project was to define the basic user requirements that the User Interface (UI) needed to have. In order to do so, several wireframes were created and discussed allowing the team to define the interface views and structure.

From the discussion, we concluded with an Information Architecture including the following pages:

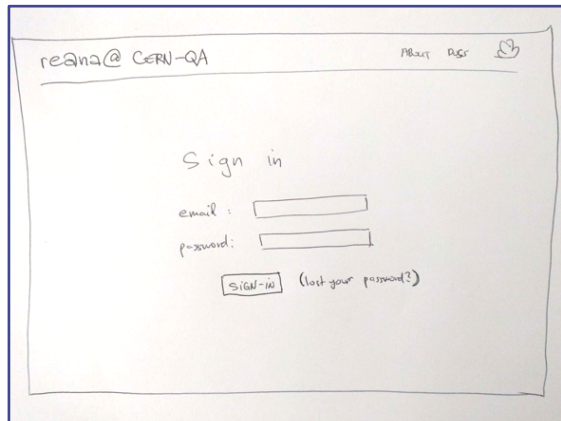


Figure 2: Login page wireframe

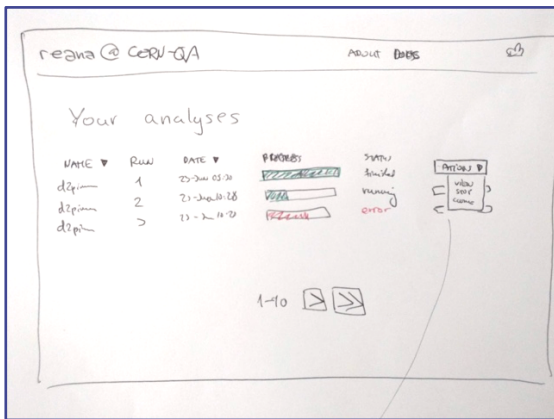


Figure 4: Workflow-list page wireframe

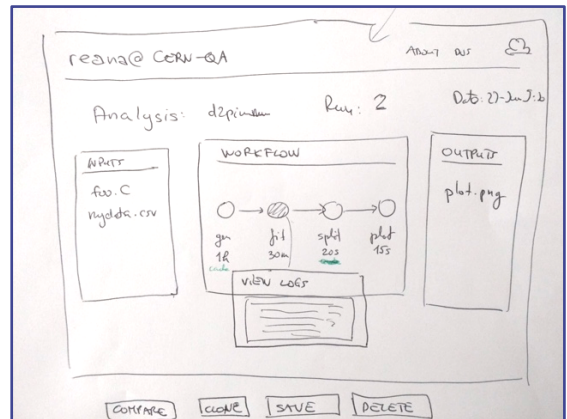


Figure 3: Workflow-details page wireframe

From those wireframes, a list of core UI elements was extracted:

- Progress bar.
- Sorting tables.
- Pagination.
- Graph visualization.
- Buttons with disability option.



3. TECHNOLOGY SELECTION



3.1. INTERFACE BUILDING

Regarding the interface building framework, we selected *React* over its competitors (*Angular* and *Vue*) due to the synergies and similarities with other CERN projects using this framework.

The preference for *React* is shared among several projects within the Digital Repositories section, including the CERN Analysis Preservation project with whom REANA shares the data analysis preservation and reusability goals.

3.2. INTERFACE STYLE

The UI style framework decision was important due to the long perspective life time and complexity of this project. The chosen framework needed to be mature enough, so it could provide most of the core requirements defined in section 2.

The frameworks to consider in this section were: *Grommet* [6], *Material-UI* [7] and *Semantic-UI* [8]. In order to select the most appropriate one, different comparisons were made.

The first comparison consisted on creating a UI elements table to discover the frameworks advantages and disadvantages:

Required element	Grommet	Material-UI	Semantic-UI
Disabled buttons	No	Yes	Yes
Progress bar labels	Yes	No	Yes
Progress bar colors	Yes	No	Yes
Nodes graphs	Yes	No	No
Table pagination	No	Yes	Yes
Table sorting	Yes	Yes	Yes
Documentation quality	Medium	Medium	Good
Examples quality	Medium	Good	Good





Additionally, several technical mock-ups were implemented using a platform called *StoryBooks* [5] for fast and easy prototyping. The next three figures are examples of how the *workflow-list* page technical mock-ups looked:



REANA						
Workflow	Run	Date ↓	Progress	Status	Actions	
Data-Analysis	1	10/07/2018	1 steps 0 / 10	Progress	👁️	⏸️ ▶️ ↺️
Data-Analysis	2	10/07/2018	4 steps 0 / 10	Error	👁️	⏸️ ▶️ ↺️
Data-Analysis	3	10/07/2018	10 steps 0 / 10	Finished	👁️	⏸️ ▶️ ↺️

Figure 7: Grommet workflow-list page technical mock-up



Workflow	Run	Date	Progress	State	Actions	
Data Analysis	1	10/07/2018	Progress	Progress	👁️	⏸️ ▶️ ↺️
Data Analysis	2	10/07/2018	Error	Error	👁️	⏸️ ▶️ ↺️
Data Analysis	3	10/07/2018	Progress	Progress	👁️	⏸️ ▶️ ↺️
Data Analysis	4	10/07/2018	Finished	Finished	👁️	⏸️ ▶️ ↺️
Data Analysis	5	10/07/2018	Finished	Finished	👁️	⏸️ ▶️ ↺️

Figure 5: Material-UI workflow-list page technical mock-up



Workflow	Run	Date	Progress	Status	Actions
D2-pimumu	3	17/07/2018	4 / 10	Running	👁️ ⏸️ ▶️ ↺️
ttH	2	12/07/2018	12 / 40	Error	👁️ ⏸️ ▶️ ↺️
Roofit	1	09/07/2018	36 / 36	Finished	👁️ ⏸️ ▶️ ↺️
H4l	5	01/07/2018	10 / 10	Finished	👁️ ⏸️ ▶️ ↺️

Figure 6: Semantic-UI workflow-list page technical mock-up

From the different frameworks, **Semantic-UI** was chosen due to the maturity of its development and the good quality of its documentation and examples.





3.3. WORKFLOW VISUALIZATION

In the case of the data analysis workflows, they are defined as directed acyclic graphs (DAG) [9]. REANA currently supports either the *Common Workflow Language* (CWL) format, or the *Yadage* format. These workflows may reach high complexity levels in real world use-cases.

The following example shows a typical computational workflow used in the *BSM search* analyses:

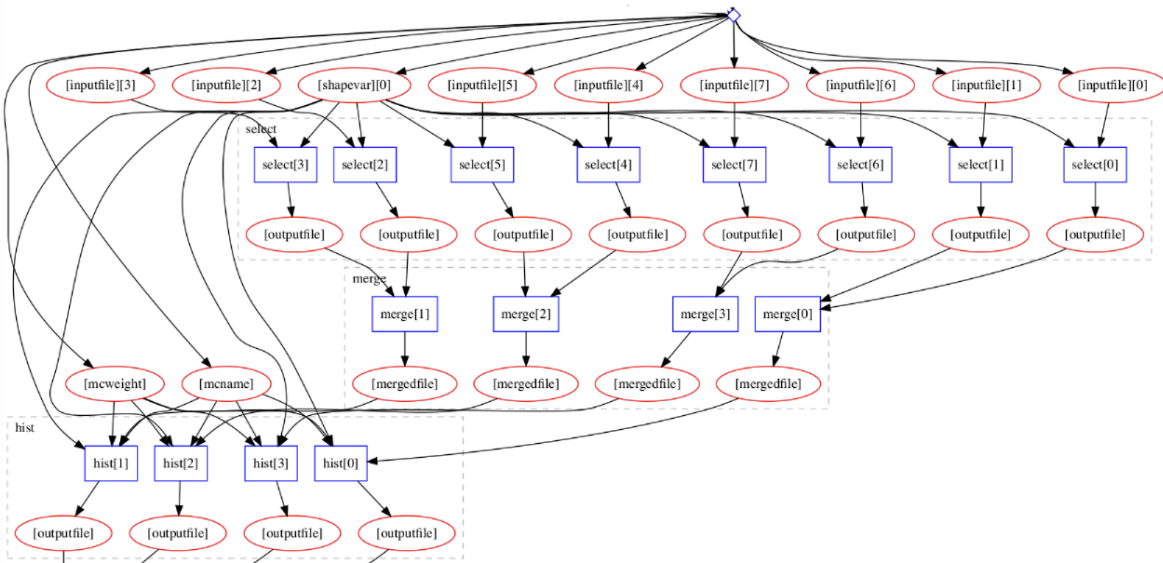


Figure 8: Part of a computational graph used in BSM search

Regarding the visualization of these workflows, **Vis.js** [10] was the selected library of choice due to its simplicity and easy integration with React. A simple example of how to define a graph in Vis.js can be seen in the following example figures:

```
graph: {
  nodes: [
    { id: 1, label: "Node 1" },
    { id: 2, label: "Node 2" },
    { id: 3, label: "Node 3" },
    { id: 4, label: "Node 4" },
    { id: 5, label: "Node 5" }
  ],
  edges: [
    { from: 1, to: 2 },
    { from: 1, to: 3 },
    { from: 2, to: 4 },
    { from: 2, to: 5 }
  ]
}
```

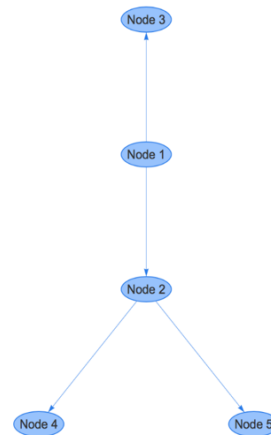


Figure 9: Example of a simple computational graph



3.4. ADDITIONAL TECHNOLOGIES

In addition to the main technology decisions, there were additional packages needed to support extended web-interface functionalities. These packages are:

- ***Axios***: to perform HTTP requests to the deployed server. Axios handles promises in a better manner than the default “*fetch*” method from React.
- ***Lodash***: to provide sorting functionality over an array of objects.
- ***Universal-cookies***: to provide session persistency after a user has logged in.
- ***History***: to ensure browser navigation, and to create our own custom navigation between pages.





4. DEVELOPMENT

4.1. LOGIN PAGE

The first page of the user interface dashboard is the login page. Given that this project is in a private pilot state, users can only register via invitation: they will need to send an email to the project administrator (info@reana.io) and ask for a token.

The received token is used as a user account associated password. It is mainly used to obtain *JSON Web Tokens* [11] (RFC-7519) which provide additional security over common tokens due to their temporary nature (JWT tokens expire after a certain time).

The fully working prototype of the *login* page can be seen in the following figure:

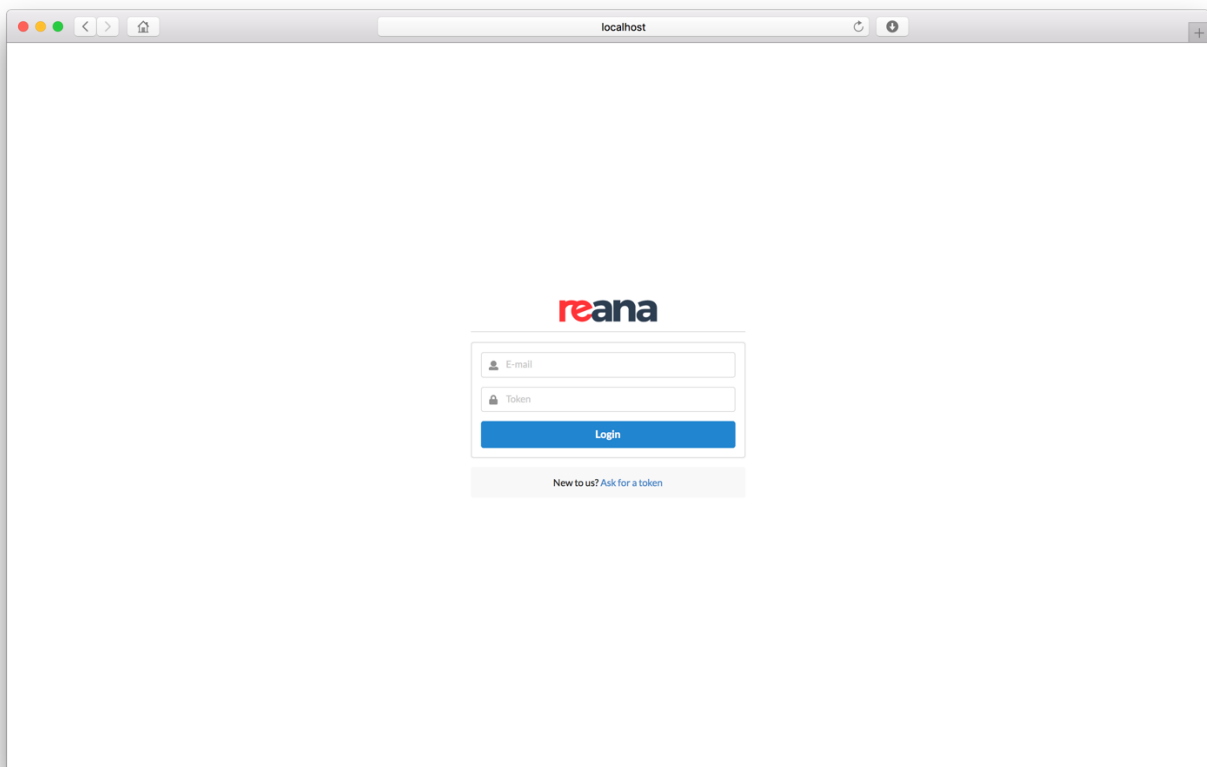


Figure 10: Screenshot of implemented login page



4.2. WORKFLOW LIST PAGE

This page will function as a home after a user has logged in. It contains information about the list of workflows that the user has created together with their status:

- The name and run.
- The creation date.
- The duration.
- The progress (updated every 5 secs).
- The current status.
- The list of available actions.

The fully working prototype *workflow-list* page can be seen in the following figure:

Name ^	Run	Created	Duration	Progress	Status	Actions
Roofit	2	2018-08-07 09:17:44	22d 02h 30m 40s	<div style="width: 50%;"><div style="width: 50%;"></div></div> 1/2	Running	View Pause Resume Rerun
Roofit	1	2018-08-07 09:17:44	22d 02h 30m 40s	<div style="width: 100%;"><div style="width: 100%;"></div></div> 2/2	Finished	View Pause Resume Rerun
CMS	1	2018-08-07 09:17:44	22d 02h 30m 40s	<div style="width: 33%;"><div style="width: 33%;"></div></div> 3/7	Failed	View Pause Resume Rerun
LHCB	2	2018-08-08 06:34:21	21d 05h 14m 43s	<div style="width: 88%;"><div style="width: 88%;"></div></div> 30/34	Running	View Pause Resume Rerun
LHCB	1	2018-08-07 09:17:44	22d 02h 30m 55s	<div style="width: 100%;"><div style="width: 100%;"></div></div> 34/34	Finished	View Pause Resume Rerun

Figure 11: Screenshot of implemented workflow-list page

In terms of **further improvements** to this current prototype page:

1. The progress update could be done over web-sockets instead of simple pooling. It will require changes in the *REANA-Server* module.
2. The pagination at the footer of the list depends on server-side pagination future implementation (GitHub issue [\[12\]](#)).



4.3. WORKFLOW DETAILS PAGE

The workflow details page is the most complex one of the three. It contains the detailed workflow overview information, including:

- The workflow basic information:
 - The name.
 - The run.
 - The creation date.
 - The current status.
- The workflow steps graph.
- The workflow input files.
- The workflow output files.
- The workflow logs.

The fully working prototype *workflow-details* page can be seen in the following figure:

The screenshot shows a web browser window displaying the REANA interface. The page title is 'reana'. The main content area shows the details for a workflow named 'RootFit', Run 1, created on 2018-08-07 09:17:44, with a status of 'finished'. The workflow graph consists of two steps: 'Generate data' and 'Fit data'. The 'Inputs' section lists two files: 'code/fitdata.C' and 'code/gendata.C', both modified on 2018-08-07 09:17:44. The 'Outputs' section lists four files: 'gendata.log', 'data.root', 'fitdata.log', and 'plot.png', all modified on 2018-08-07 09:18:00. The 'Logs' section shows the output of the 'Fit data' step, which includes the ROOT version (6.02/12) and the build information.

Figure 12: Screenshot of implemented *workflow-details* page



Additionally, there exists an online file visualization feature. Once a file is selected, the user can see its content and download it to their local computers.

The online file visualizer can be seen in the next figure:

```

gendata.log

-----
| Welcome to ROOT 6.08/12      http://root.cern.ch |
| Built for linux64sgp       (c) 1995-2014, The ROOT Team |
| From tag v6-02-12_24 June 2015 |
| Try ".help", ".demo", ".license", ".credits", ".quit"/"-q" |
|-----|

Processing code/gendata.c(2000,"data.root")...

[InfoFit v3.60 --- Developed by Wouter Verkerke and David Kirkby@
Copyright (C) 2000-2013 NIKHEF, University of California & Stanford University
All rights reserved. Please read http://root.cern.ch/license.txt

#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooAddPdf::model
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooChebychev::bhg
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x0
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x1
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x2
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x3
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x4
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x5
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x6
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x7
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x8
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x9
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x10
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x11
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x12
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x13
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x14
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x15
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x16
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x17
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x18
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x19
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x20
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x21
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x22
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x23
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x24
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x25
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x26
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x27
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x28
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x29
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x30
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x31
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x32
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x33
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x34
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x35
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x36
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x37
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x38
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x39
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x40
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x41
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x42
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x43
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x44
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x45
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x46
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x47
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x48
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x49
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x50
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x51
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x52
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x53
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x54
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x55
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x56
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x57
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x58
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x59
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x60
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x61
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x62
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x63
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x64
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x65
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x66
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x67
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x68
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x69
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x70
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x71
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x72
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x73
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x74
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x75
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x76
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x77
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x78
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x79
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x80
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x81
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x82
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x83
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x84
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x85
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x86
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x87
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x88
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x89
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x90
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x91
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x92
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x93
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x94
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x95
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x96
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x97
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x98
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x99
#[] TMathObjectHandling -- RooWorkspace::import(w) importing RooHistVar1x100

RooWorkspace(w) workspace contents

variable
-----
(a0,a1,mean,bhkg,nsig,signal,x)

p.d.f.s
-----
RooChebychev::bhg[ a0 a1(a0,a1) ] = 1,2
RooAddPdf::model[ bhg + sig * sig ] = 1,1
RooAddPdf::sig[ nsig * sig ] = 1
RooGaussian::sig[ a0 a1(a0,a1) ] = 1

datasets
-----
RooDataSet::modelData(x)

-----

```

Figure 13: Screenshot of online file visualization

In terms of **further improvements** to this current prototype page:

1. The ability to filter both the list of files and the logs depending on the selected workflow step. It will require changes in the *REANA-Server* module.
2. The ability to expand and collapse nodes from the workflow graph visualization. It would be really useful for complex graphs.



5. CONCLUSION



The present work delivered a first fully functional prototype of the REANA user dashboard. The newly developed *REANA-UI* module, open sourced on GitHub [3], offers users an easy interactive way to inspect and visualize their running workflows over a web interface.

The developed component sets the basic architecture of the REANA web frontend, validating the technology choices and creating a robust foundation for implementing further improvements and future user interface needs.





6. REFERENCES

1. Baker, Monya. “1,500 scientists lift the lid on reproducibility”. 25 May 2016.
<https://www.nature.com/news/1-500-scientists-lift-the-lid-on-reproducibility-1.19970>
2. Reusable Analysis platform, *REANA-Client* component.
<https://github.com/reanahub/reana-client>
3. Reusable Analysis platform, *REANA-UI* component.
<https://github.com/reanahub/reana-ui>
4. Reusable Analysis platform, *REANA-Server* component.
<https://github.com/reanahub/reana-server>
5. Storybook community. *StoryBooks* framework.
<https://github.com/storybooks/storybook>
6. Grommet. *Grommet* framework.
<http://grommet.io>
7. Material-UI Organization. *Material-UI* framework.
<https://material-ui.com>
8. Semantic Organization. *Semantic-UI* framework.
<https://react.semantic-ui.com>
9. Wikipedia, Wikimedia Foundation. “Directed Acyclic Graph”.
https://en.wikipedia.org/wiki/Directed_acyclic_graph
10. Almende. *Vis.js* visualization framework
<http://visjs.org>
11. Auth0. JSON Web Tokens (RFC-7519).
<https://jwt.io>
12. Reusable Analysis platform. “*reana-ui*” repository, issue number 12. “*implementation: pagination*”. <https://github.com/reanahub/reana-ui/issues/12>