SevenBridges

GCS: BioCompute Object
Genomics Compliance Suite User Manual

October 2019

Table of contents

Goals

Hello, and welcome to the Genomics Compliance Suite (GCS). This manual walks through several steps involved in BioCompute object (BCO) file generation in the GCS. The app guides a user for the generation of BCO manually, or by first selecting bioinformatics workflows written in the Common Workflow Language (CWL), and then by allowing the user to enter text for BioCompute fields described in the BioCompute specification.

Briefly, the goals of this user manual are:

- Learn about the application
- Fetch the CWL workflows in your project, identify and manipulate to generate BCO files
- Download the generated BCO JSON file
- Generate PDF reports for the generated BCO file
- Upload generated BCO file to any of the Seven Bridges platforms or your GitHub account

Getting Started with the Genomics Compliance Suite (GCS)

App Access and Login

The GCS app is designed to be easily deployed locally for testing or on any remote servers for production. A Dockerized version is available on GitHub. Seven Bridges employees can access the app from the internal portal. We are also testing access with academic groups through the Cancer Genomics Cloud.

Dockerized app

Pull or build the Docker image from GitHub (see REAME.md) for details. Run the Docker container. Go to the app served by the Docker container at http://127.0.0.1:3838 in your web browser, and login with the default username and password `sevenbridges/sevenbridges` (Figure 1).

Seven Bridges hosted app

If you are logging on within Seven Bridges, the following instructions are pertinent to those logging on from within the Seven Bridges VPN.

Go to the hosted version https://shiny-internal.sbgenomics.com/gcs/ (behind the VPN) and log in with your username and password.

RStudio Server on the Cancer Genomics Cloud

Getting the app

Copy the ‘biocompute-composer.zip’ to your project folder. Please request access to the zip file if you do not currently have access.

Running the app

To run the app:

- Click "Interactive Analysis", then open "Data Cruncher".

- Click "Create your first analysis", select "RStudio" and start the analysis.

- When the editor is launched, switch to the "Terminal" tab in the lower left panel, run the following commands to copy the app from the project to the analysis session:

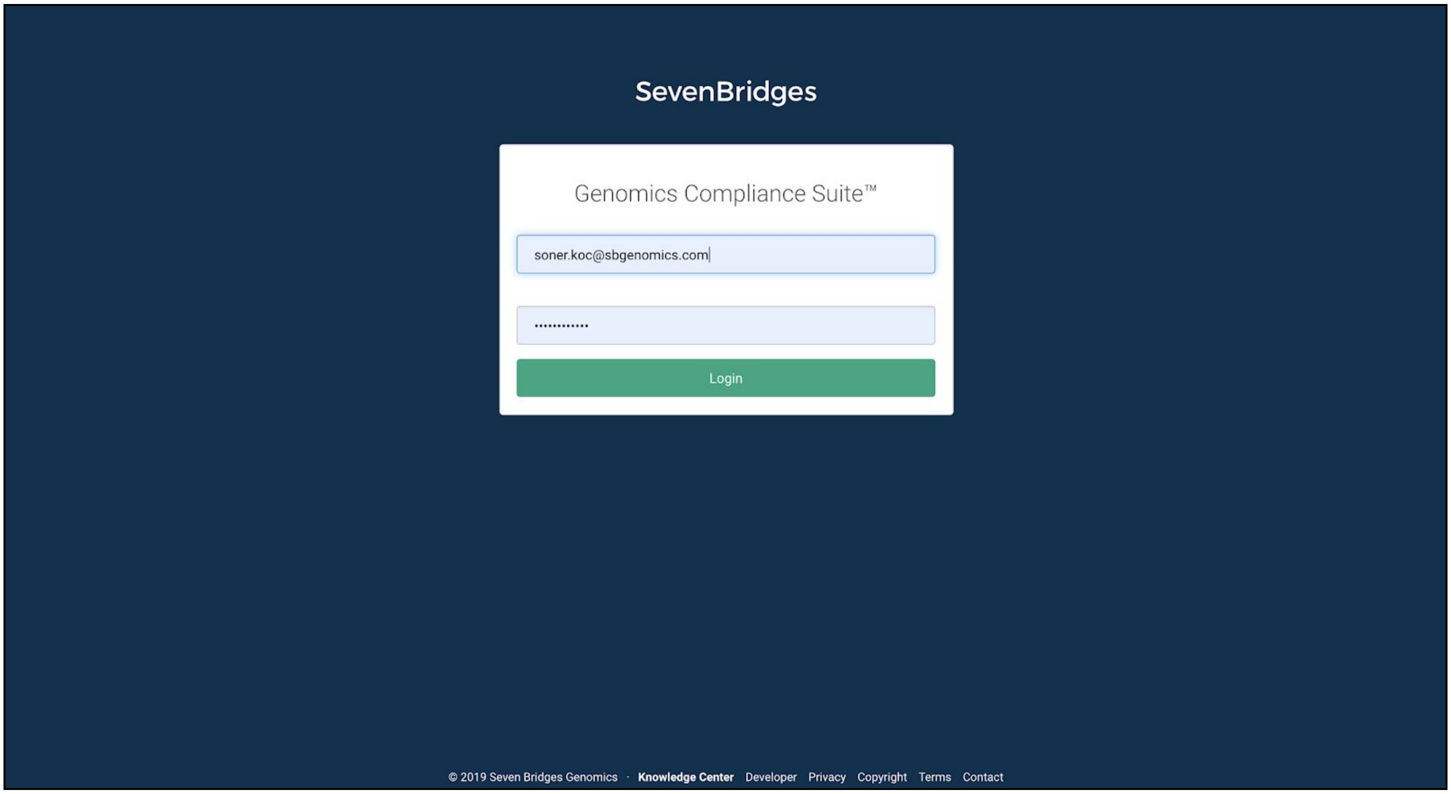
  ```
  cp /sbgenomics/project-files/biocompute-composer.zip /sbgenomics/workspace/
  unzip -q /sbgenomics/workspace/biocompute-composer.zip
  rm /sbgenomics/workspace/biocompute-composer.zip
  ```

- Go to the biocompute-composer folder in the files panel (lower right), open and run through the code in app-setup.R to setup the environment (use Ctrl + Enter to run the code line-by-line).

- Open app.R, click the "Run App" button in the code editor panel to run the app.

- A dialog about popup windows may appear if you're running this for the first time, click "Try Again" to proceed.

- Note: if running the app from a restarted analysis session, remember to run app-setup.R again before running the app.



SevenBridges

Genomics Compliance Suite™

soner.koc@sbgenomics.com

••••••••••••

© 2019 Seven Bridges Genomics · **Knowledge Center** Developer Privacy Copyright Terms Contact

Figure 1: Login Page of the Genomics Compliance Suite

Start Page

You will see the welcome page of the GCS app after your first login:

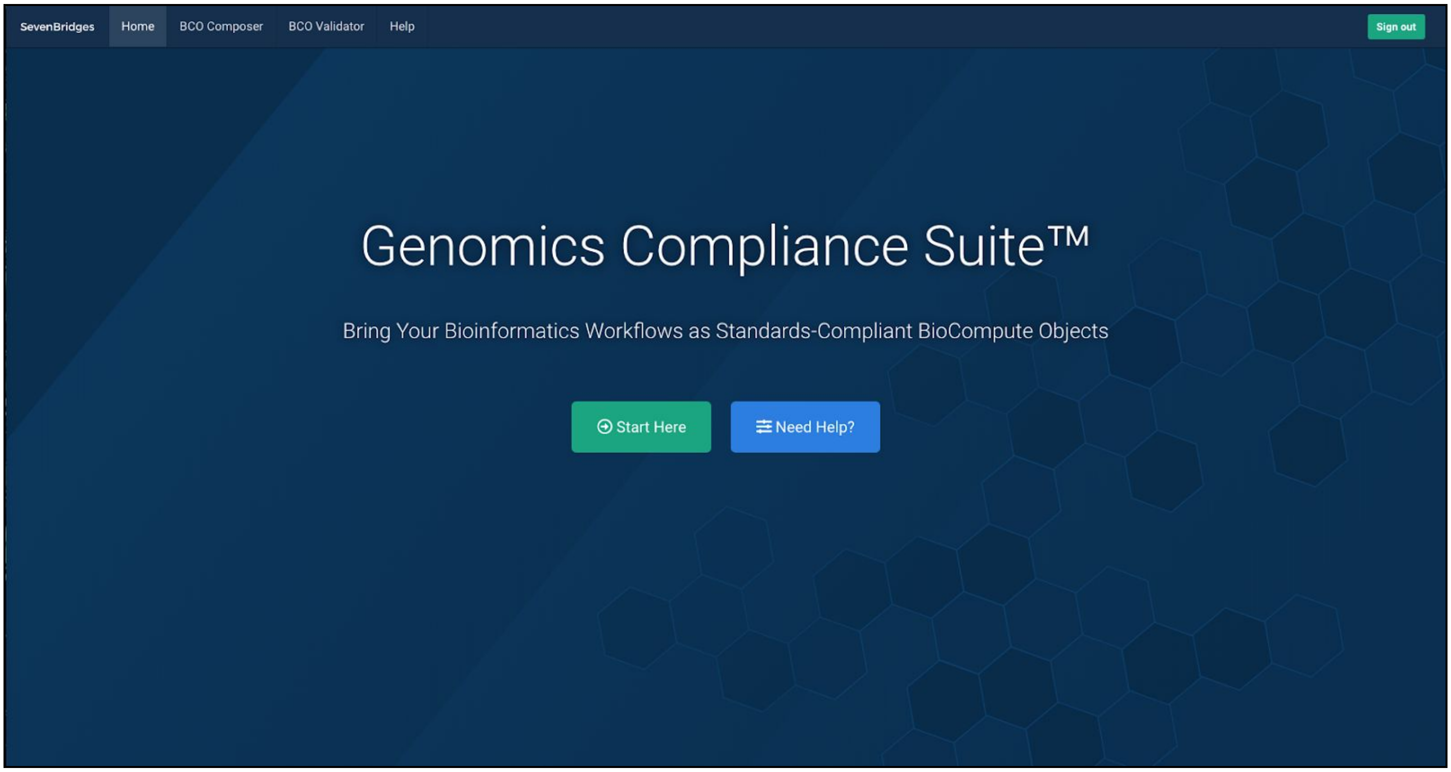

SevenBridges
Home
BCO Composer
BCO Validator
Help
Sign out
Genomics Compliance Suite™
Bring Your Bioinformatics Workflows as Standards-Compliant BioCompute Objects
Start Here
Need Help?

Figure 2: Welcome Page of the Genomics Compliance Suite

In the middle of all, you see the quick action buttons; you can quickly start to build your BioCompute (BCO) object file with "**Start Here**" button, or you can access the informative page about the BCO with "**Need Help?**" button.

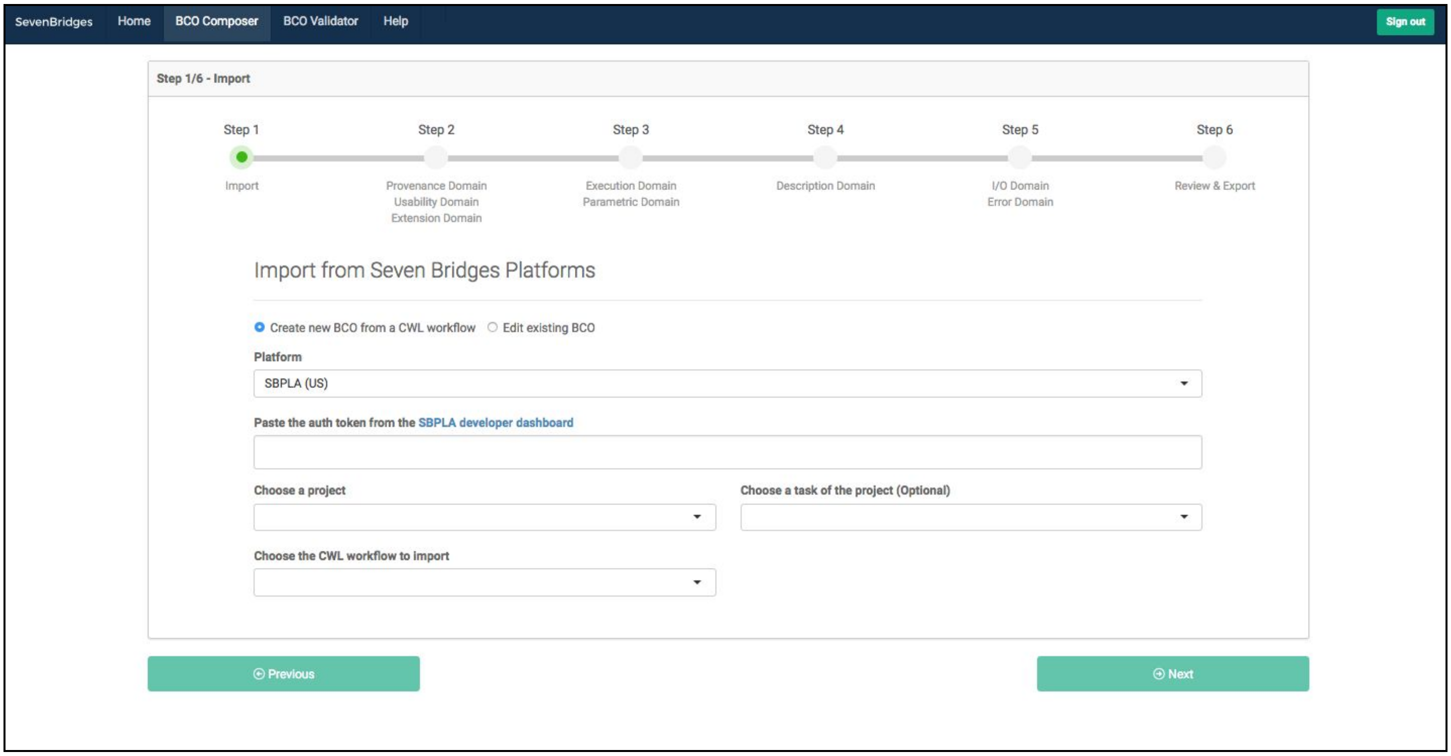
In the navigation bar, you will see the list of tabs for the main features, such as **BCO Composer (manual mode), BCO Composer (from CWL)**, **BCO Validation, Interactive BCO Browser**, and **Help**.

To save us some time, we will walk through the BCO Composer (from CWL) feature below in detail. All the other features should follow the same principles and are intuitive to use.

BioCompute Object (BCO) Composer

Step 1 - Import the Target CWL App

After you click the "**BCO Composer**" page of the GCS platform, you will see the initial step of the BCO generation, import step (Figure 3).



Step 1/6 - Import

Step 1 – Import
Step 2 – Provenance Domain, Usability Domain, Extension Domain
Step 3 – Execution Domain, Parametric Domain
Step 4 – Description Domain
Step 5 – I/O Domain, Error Domain
Step 6 – Review & Export

Import from Seven Bridges Platforms

● Create new BCO from a CWL workflow ○ Edit existing BCO

Platform

SBPLA (US)

Paste the auth token from the SBPLA developer dashboard

Choose a project

Choose a task of the project (Optional)

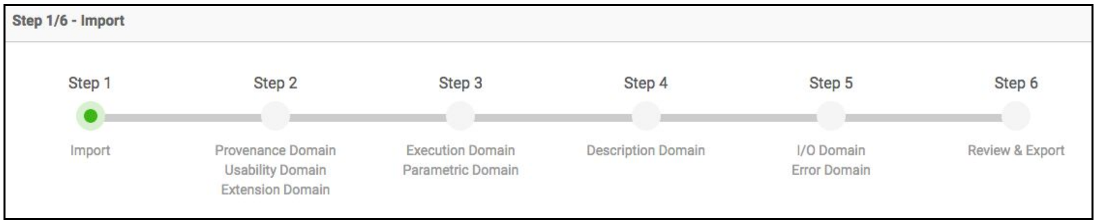
Choose the CWL workflow to import

Figure 3: Initial Page of the BCO Composer

The BCO Creator/Composer has six different steps to generate a new BioCompute Object (BCO) to cover each of the BCO domains from the ground up (Figure 4).

It begins with Common Workflow Language (CWL) importing step then goes with Step 2, which has information about Provenance, Usability, and Extension Domains. In Step 3, you can see the Execution Domain and Parametric Domain fields of a BCO sample. Then it goes with Step 4 that has Description Domain. Step 5 follows with I/O Domain and Error Domain. In the end, the user can review the generated BCO and take action like download the BCO file, upload it to your Seven Bridges platform accounts or GitHub accounts.

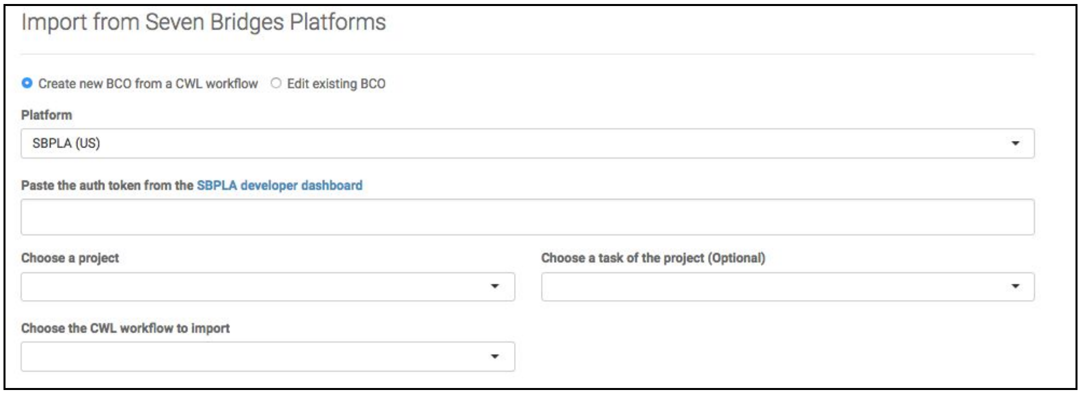
The "Import" step collects some information from a user to initiate the generation process. These are: **Platform**, **Authentication Token**, **Project, Task of the Project** as an optional, and **Application Name** to import the CWL of the selected application.



Step 1/6 - Import

Step 1 – Import
Step 2 – Provenance Domain, Usability Domain, Extension Domain
Step 3 – Execution Domain, Parametric Domain
Step 4 – Description Domain
Step 5 – I/O Domain, Error Domain
Step 6 – Review & Export

Figure 4: Steps of the BCO Composer



Import from Seven Bridges Platforms

● Create new BCO from a CWL workflow ○ Edit existing BCO

Platform: SBPLA (US)

Paste the auth token from the SBPLA developer dashboard:

Choose a project:

Choose a task of the project (Optional):

Choose the CWL workflow to import:

Figure 5: Necessary information fields at the beginning of the BCO Composer

Select a name for a "**Platform**" that you want to fetch a CWL file of an application (Figure 5).

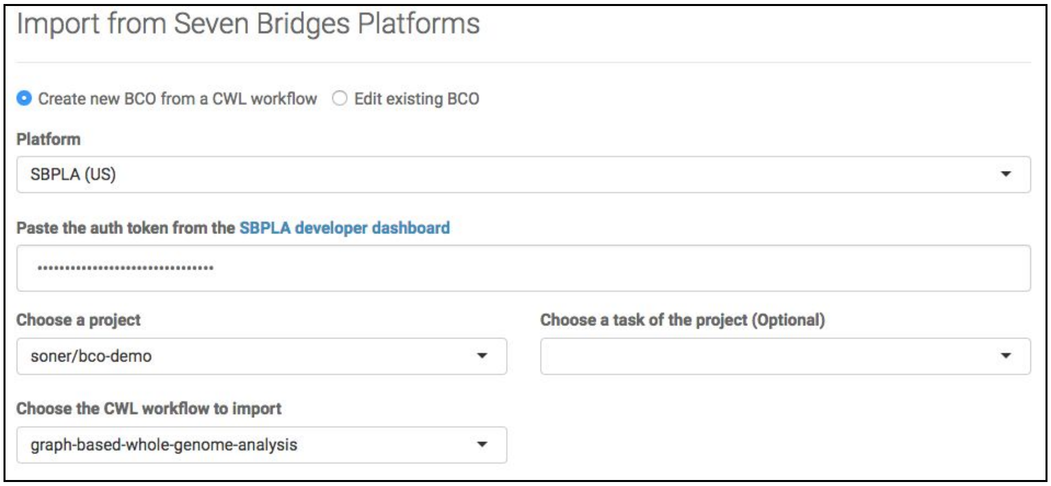
After that, paste your authentication token to "**Paste the auth token from** ..." field. Colored text in the name of the field is a direct link to the authentication token page of the selected platform (Please check **Appendix I** to learn more about how to generate authentication token).

When the authentication token of the user is pasted, projects, tasks, and applications that exist in the account of the user will be fetched.

User can select a project to focus on in "**Choose a project**" field. Then, a task can be selected among the existing tasks in the selected project from "**Choose a task of the project**" field, which is optional.

Finally, the target application can be selected from "**Choose the CWL workflow to import**" field to fetch a CWL of a target application to work with to generate a BioCompute Object.

Filled entries of the page are illustrated in Figure 6 and Figure 7.



Import from Seven Bridges Platforms

● Create new BCO from a CWL workflow ○ Edit existing BCO

Platform: SBPLA (US)

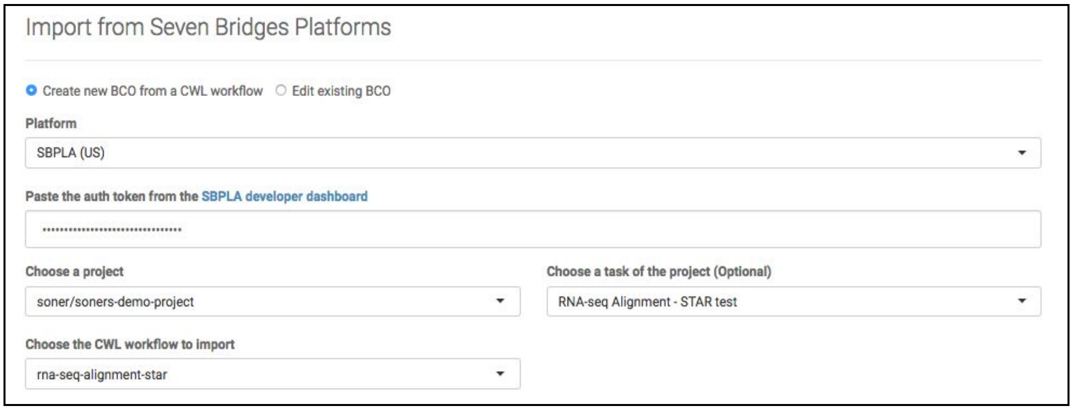
Paste the auth token from the SBPLA developer dashboard: ································

Choose a project: soner/bco-demo

Choose a task of the project (Optional):

Choose the CWL workflow to import: graph-based-whole-genome-analysis

Figure 6: Illustration of the necessary information fields



Import from Seven Bridges Platforms

● Create new BCO from a CWL workflow ○ Edit existing BCO

Platform: SBPLA (US)

Paste the auth token from the SBPLA developer dashboard: ••••••••••••••••••••••••••••••••

Choose a project: soner/soners-demo-project

Choose a task of the project (Optional): RNA-seq Alignment - STAR test

Choose the CWL workflow to import: rna-seq-alignment-star

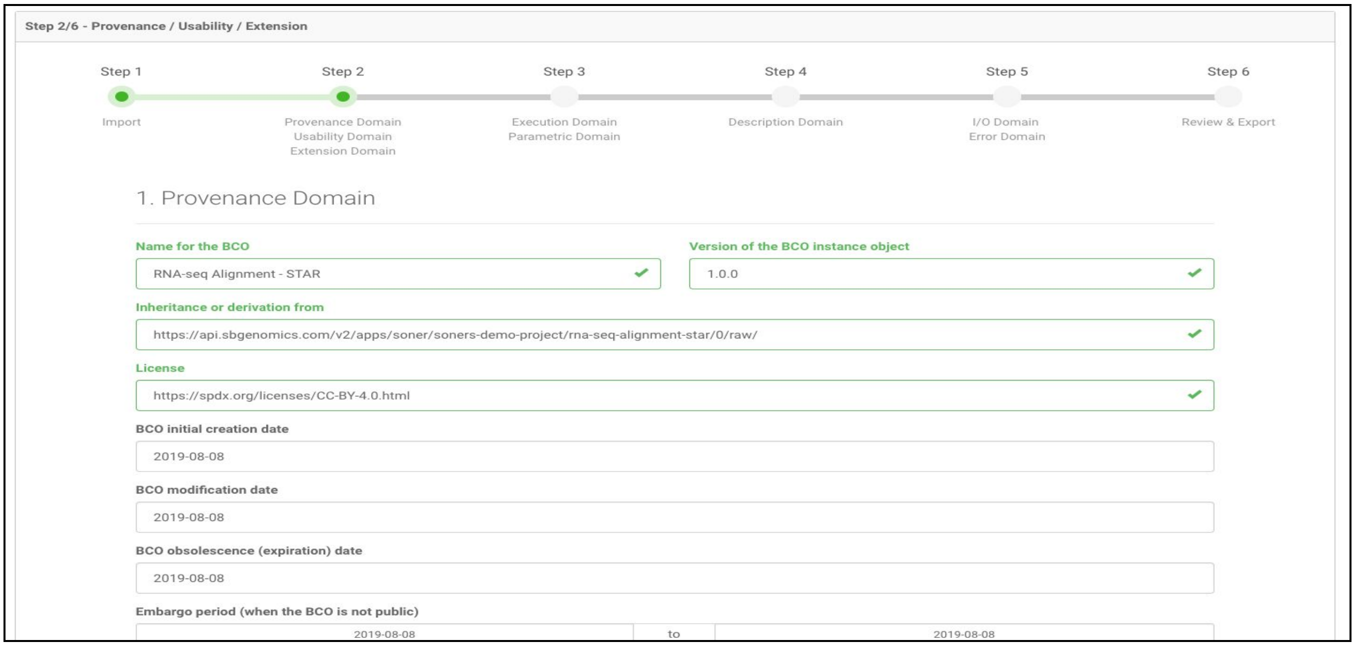
Figure 7: Illustration of the necessary information fields with the optional (task) selection

Step 2 - Provenance, Usability, and Extension Domains

In step 2, Provenance, Usability, and Extension Domains of BCO are listed with their identification fields.

After selecting the target application's CWL file, the GCS app fetches and parses the selected CWL in the background to pre-fill standard information fields between CWL and BCO fields. It is illustrated in Figure 8. On the other hand, some of the fields of BCO cannot be filled by the information parsed from CWL, as shown in Figure 9. Users can add new information into these fields manually. Users can also edit the pre-filled fields according to their needs.

You can find the complete detailed list of Provenance, Usability, and Extension Domain fields in Appendix II, Table 2, and Table 4.



1. Provenance Domain

Name for the BCO: RNA-seq Alignment - STAR ✓

Version of the BCO instance object: 1.0.0 ✓

Inheritance or derivation from: https://api.sbgenomics.com/v2/apps/soner/soners-demo-project/rna-seq-alignment-star/0/raw/ ✓

License: https://spdx.org/licenses/CC-BY-4.0.html ✓

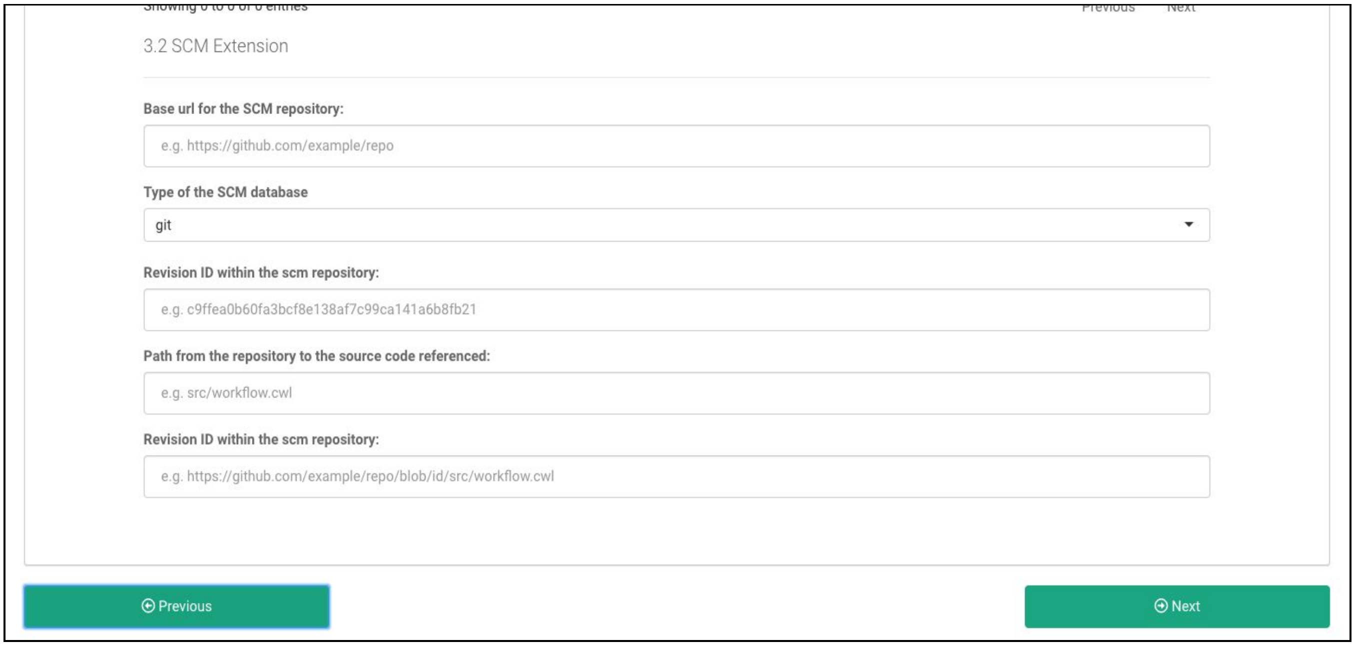
BCO initial creation date: 2019-08-08

BCO modification date: 2019-08-08

BCO obsolescence (expiration) date: 2019-08-08

Embargo period (when the BCO is not public): 2019-08-08 to 2019-08-08

Figure 8: Filled fields, by CWL of the selected application, of domains exist in Step 2



Showing 0 to 0 of 0 entries

3.2 SCM Extension

Base url for the SCM repository:

e.g. https://github.com/example/repo

Type of the SCM database

git

Revision ID within the scm repository:

e.g. c9ffea0b60fa3bcf8e138af7c99ca141a6b8fb21

Path from the repository to the source code referenced:

e.g. src/workflow.cwl

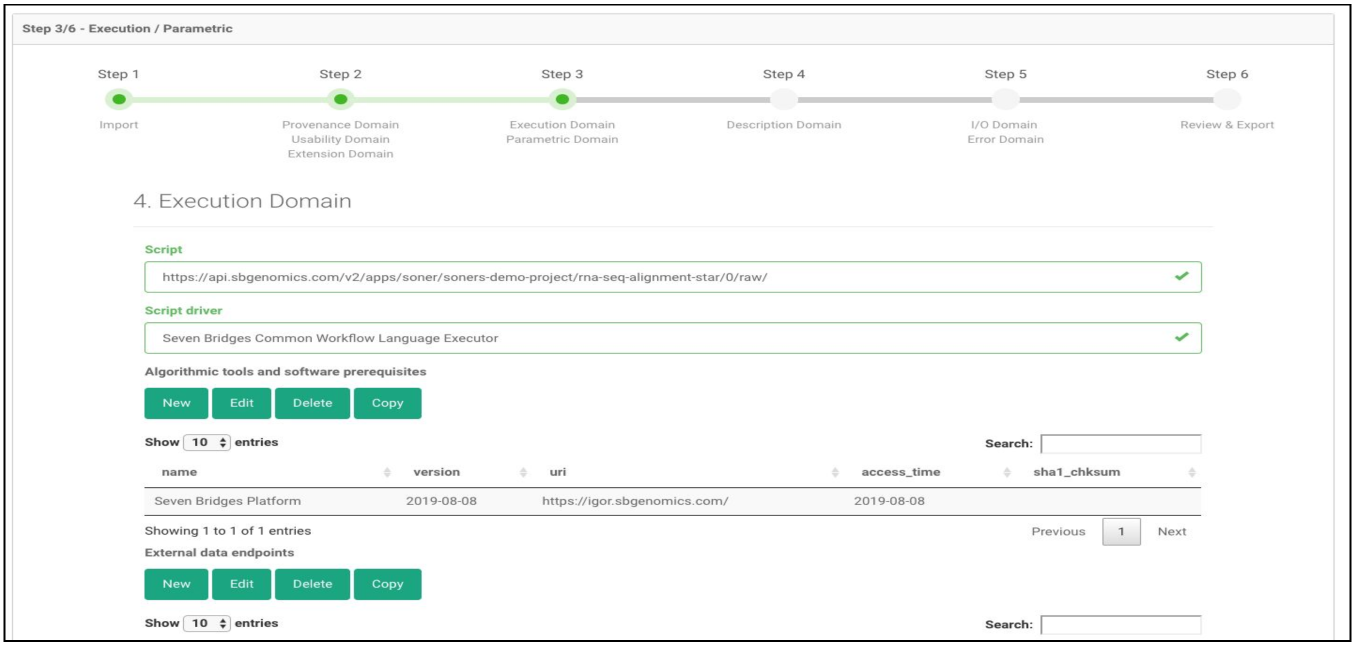
Revision ID within the scm repository:

e.g. https://github.com/example/repo/blob/id/src/workflow.cwl

Figure 9: Unfilled fields, by CWL of the selected application, of domains exist in Step 2

Step 3 - Execution and Parametric Domains

In step 3, Execution and Parametric Domains of BCO are listed with their identification fields.



Step 3/6 - Execution / Parametric

4. Execution Domain

Script

https://api.sbgenomics.com/v2/apps/soner/soners-demo-project/rna-seq-alignment-star/0/raw/ ✓

Script driver

Seven Bridges Common Workflow Language Executor ✓

Algorithmic tools and software prerequisites

Show 10 entries

Search:

name	version	uri	access_time	sha1_chksum
Seven Bridges Platform	2019-08-08	https://igor.sbgenomics.com/	2019-08-08	

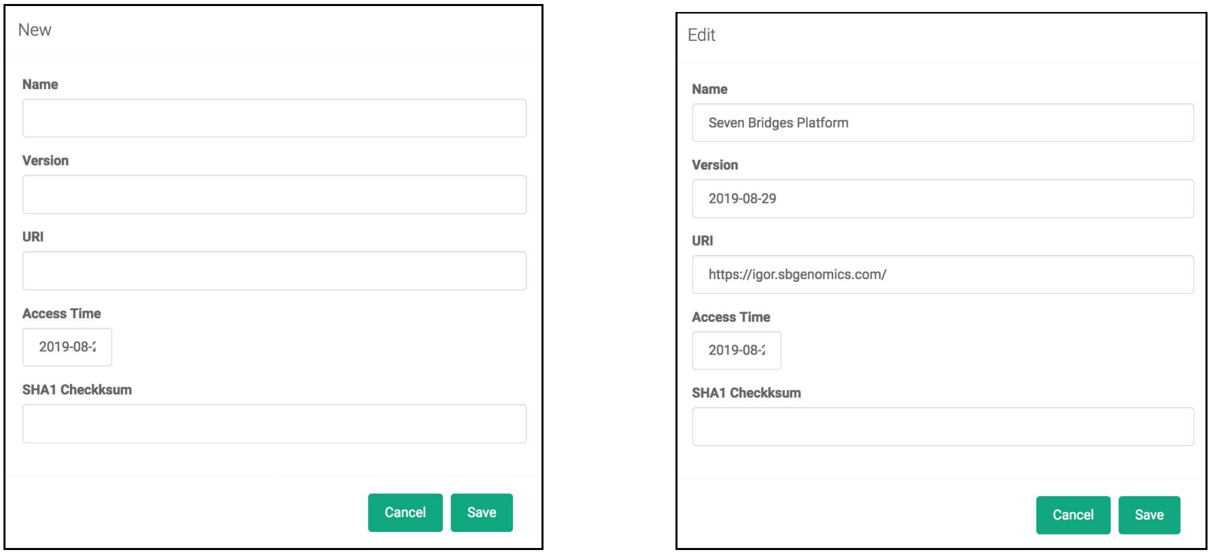
Showing 1 to 1 of 1 entries

External data endpoints

Show 10 entries

Search:

Figure 10: Execution Domain example from Step 3



New

Name	
Version	
URI	
Access Time	2019-08-2
SHA1 Checkksum	

Edit

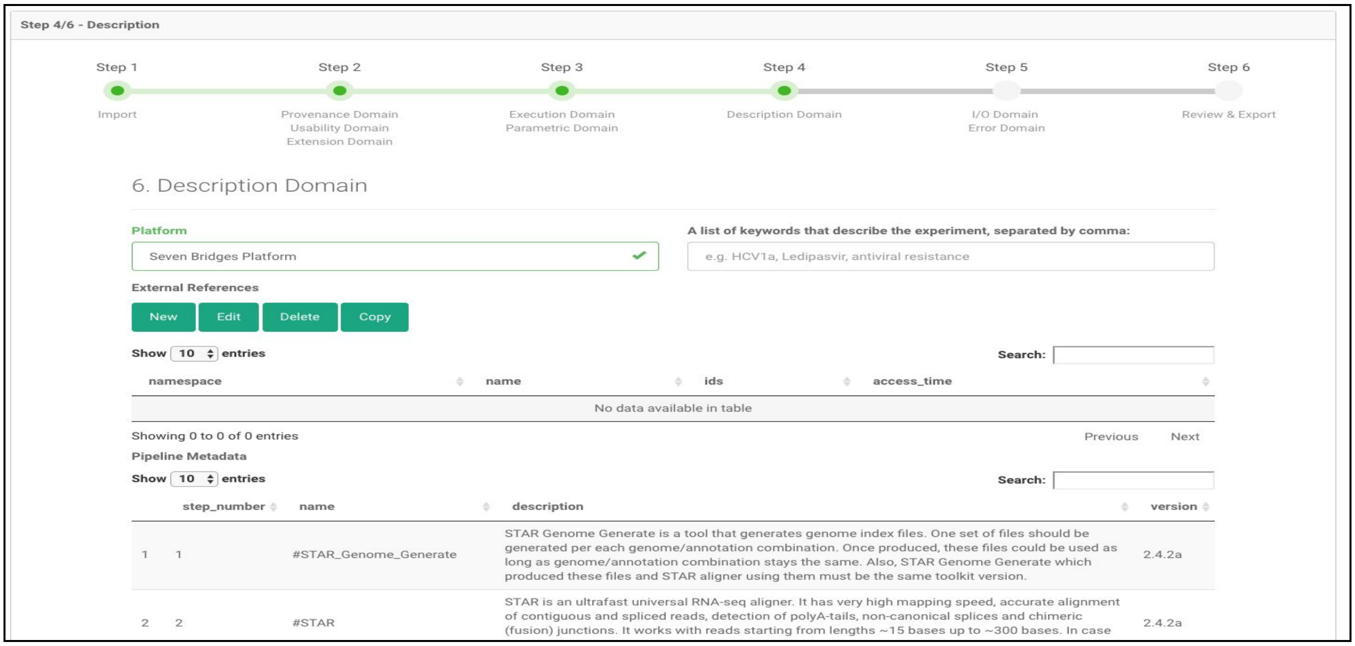
Name	Seven Bridges Platform
Version	2019-08-29
URI	https://igor.sbgenomics.com/
Access Time	2019-08-2
SHA1 Checkksum	

Figure 11: Example of adding a new entry or editing existence one in the field

You can find the complete detailed list of Execution and Parametric Domains' fields in Appendix II, Table 6, and Table 7.

Step 4 - Description Domain

In this step, Description Domain of BCO is listed with its identification fields.



Step 4/6 - Description

6. Description Domain

Platform: Seven Bridges Platform ✓

A list of keywords that describe the experiment, separated by comma: e.g. HCV1a, Ledipasvir, antiviral resistance

External References

Show 10 entries   Search:

namespace	name	ids	access_time
No data available in table			

Showing 0 to 0 of 0 entries   Previous Next

Pipeline Metadata

Show 10 entries   Search:

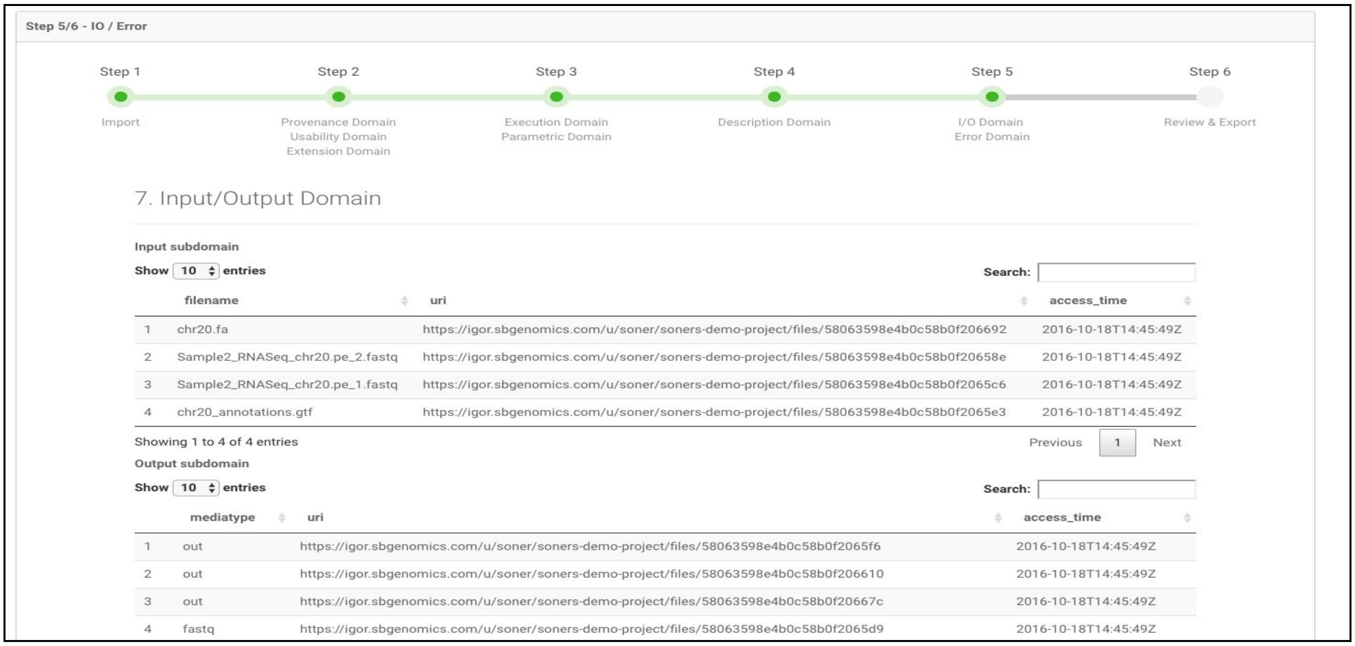
	step_number	name	description	version
1	1	#STAR_Genome_Generate	STAR Genome Generate is a tool that generates genome index files. One set of files should be generated per each genome/annotation combination. Once produced, these files could be used as long as genome/annotation combination stays the same. Also, STAR Genome Generate which produced these files and STAR aligner using them must be the same toolkit version.	2.4.2a
2	2	#STAR	STAR is an ultrafast universal RNA-seq aligner. It has very high mapping speed, accurate alignment of contiguous and spliced reads, detection of polyA-tails, non-canonical splices and chimeric (fusion) junctions. It works with reads starting from lengths ~15 bases up to ~300 bases. In case	2.4.2a

Figure 12: Description Domain example from Step 4

You can find the complete list of the details of the Description Domain fields in Appendix II and Table 5.

Step 5 - I/O and Error Domain

In step 5, Input/Output and Error Domains of BCO is listed with their identification fields.



7. Input/Output Domain

Input subdomain

Show 10 **entries** **Search:**

	filename	uri	access_time
1	chr20.fa	https://igor.sbgenomics.com/u/soner/soners-demo-project/files/58063598e4b0c58b0f206692	2016-10-18T14:45:49Z
2	Sample2_RNASeq_chr20.pe_2.fastq	https://igor.sbgenomics.com/u/soner/soners-demo-project/files/58063598e4b0c58b0f20658e	2016-10-18T14:45:49Z
3	Sample2_RNASeq_chr20.pe_1.fastq	https://igor.sbgenomics.com/u/soner/soners-demo-project/files/58063598e4b0c58b0f2065c6	2016-10-18T14:45:49Z
4	chr20_annotations.gtf	https://igor.sbgenomics.com/u/soner/soners-demo-project/files/58063598e4b0c58b0f2065e3	2016-10-18T14:45:49Z

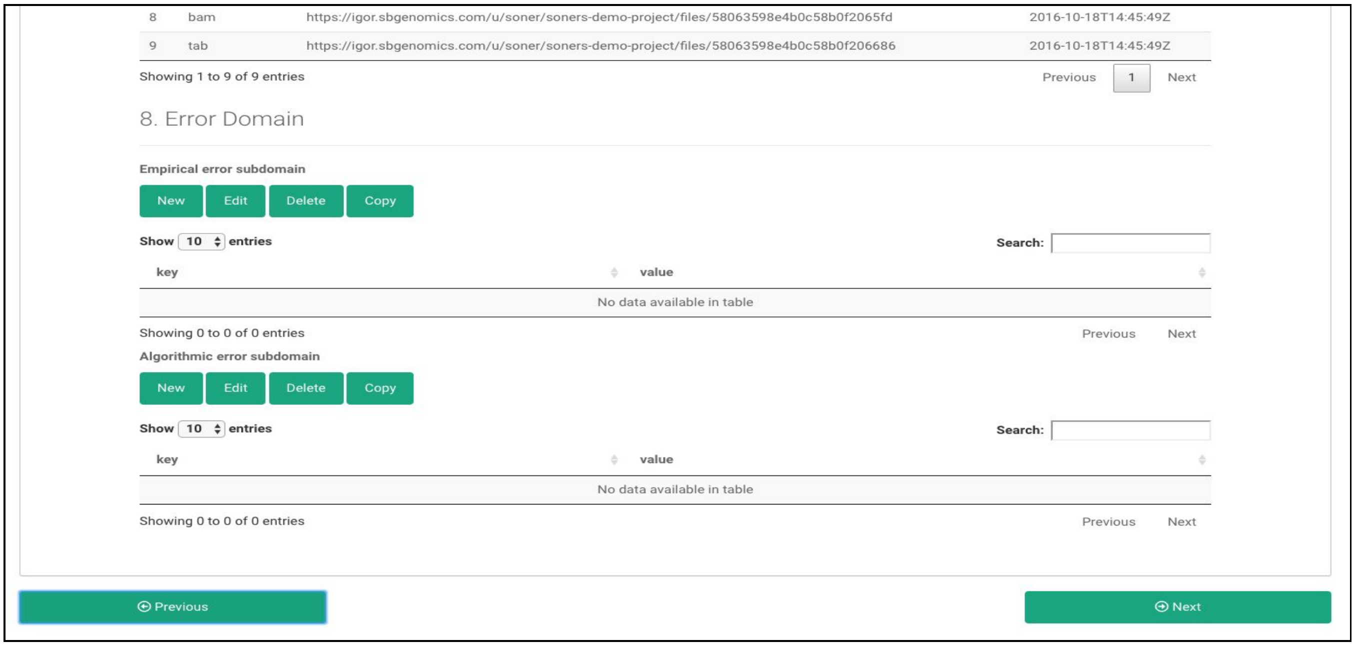
Showing 1 to 4 of 4 entries

Output subdomain

Show 10 **entries** **Search:**

	mediatype	uri	access_time
1	out	https://igor.sbgenomics.com/u/soner/soners-demo-project/files/58063598e4b0c58b0f2065f6	2016-10-18T14:45:49Z
2	out	https://igor.sbgenomics.com/u/soner/soners-demo-project/files/58063598e4b0c58b0f206610	2016-10-18T14:45:49Z
3	out	https://igor.sbgenomics.com/u/soner/soners-demo-project/files/58063598e4b0c58b0f20667c	2016-10-18T14:45:49Z
4	fastq	https://igor.sbgenomics.com/u/soner/soners-demo-project/files/58063598e4b0c58b0f2065d9	2016-10-18T14:45:49Z

Figure 13: I/O Domain example from Step 5



8	bam	https://igor.sbgenomics.com/u/soner/soners-demo-project/files/58063598e4b0c58b0f2065fd	2016-10-18T14:45:49Z
9	tab	https://igor.sbgenomics.com/u/soner/soners-demo-project/files/58063598e4b0c58b0f206686	2016-10-18T14:45:49Z

Showing 1 to 9 of 9 entries

8. Error Domain

Empirical error subdomain

Show 10 **entries** **Search:**

key	value
No data available in table	

Showing 0 to 0 of 0 entries

Algorithmic error subdomain

Show 10 **entries** **Search:**

key	value
No data available in table	

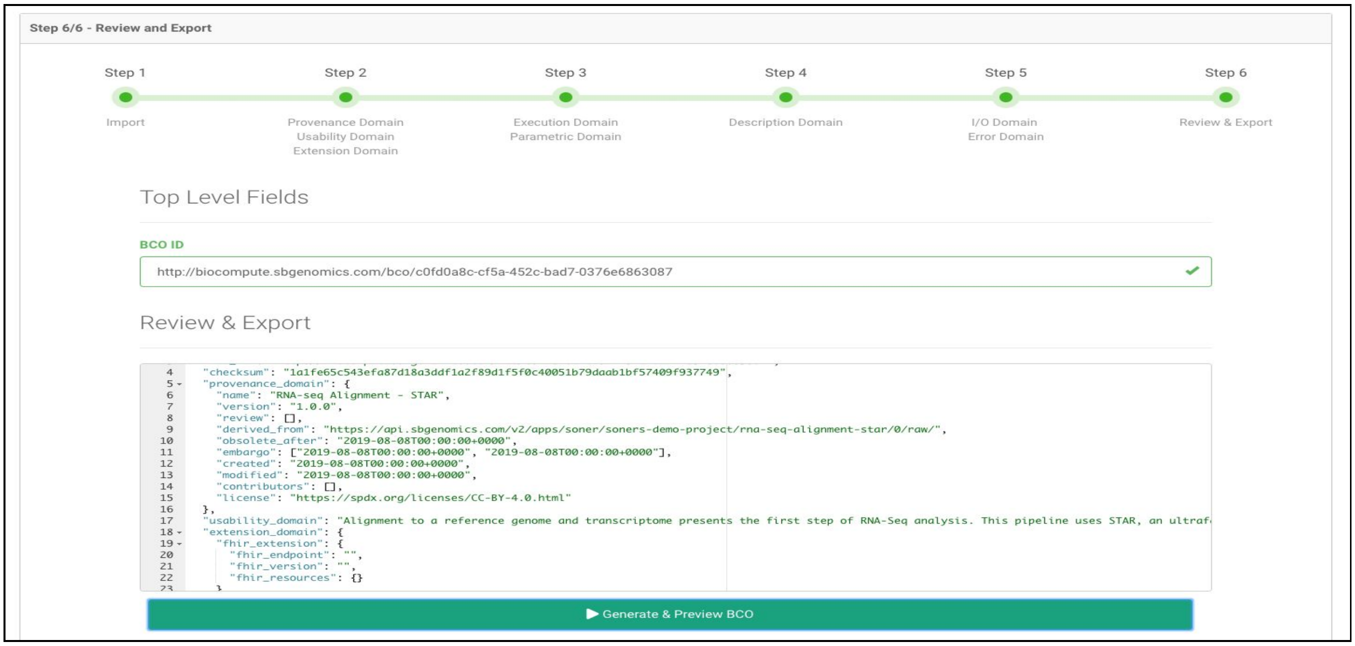
Showing 0 to 0 of 0 entries

Figure 14: Error Domain example from Step 5

You can find the complete detailed list of I/O and Error Domains' fields in Appendix II, Table 8, and Table 9.

Step 6 - Review and Export

As the last step of BCO Composer, you can review your generated BCO file in "**Review & Export**" after "**Generate & Preview BCO**" button is clicked. At the top of them, the "**BCO ID**" of the generated BCO file is automatically assigned.



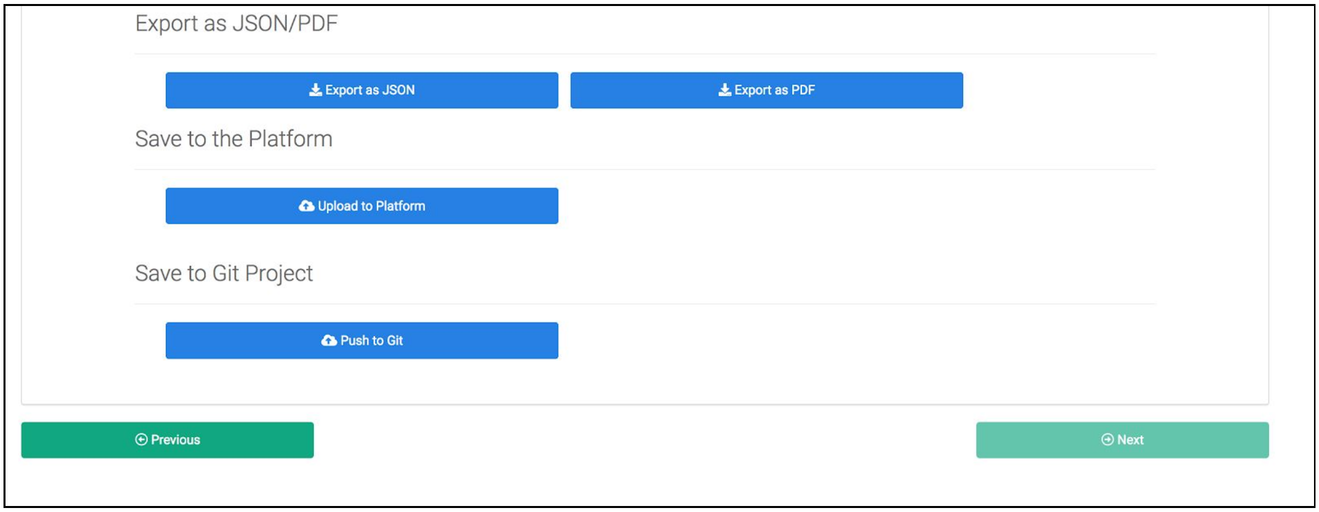
BCO ID

http://biocompute.sbgenomics.com/bco/c0fd0a8c-cf5a-452c-bad7-0376e6863087 ✓

```
  "checksum": "1a1fe65c543efa87d18a3ddf1a2f89d1f5f0c40051b79daab1bf57409f937749",
  "provenance_domain": {
    "name": "RNA-seq Alignment - STAR",
    "version": "1.0.0",
    "review": [],
    "derived_from": "https://api.sbgenomics.com/v2/apps/soner/soners-demo-project/rna-seq-alignment-star/0/raw/",
    "obsolete_after": "2019-08-08T00:00:00+0000",
    "embargo": ["2019-08-08T00:00:00+0000", "2019-08-08T00:00:00+0000"],
    "created": "2019-08-08T00:00:00+0000",
    "modified": "2019-08-08T00:00:00+0000",
    "contributors": [],
    "license": "https://spdx.org/licenses/CC-BY-4.0.html"
  },
  "usability_domain": "Alignment to a reference genome and transcriptome presents the first step of RNA-Seq analysis. This pipeline uses STAR, an ultraf
  "extension_domain": {
    "fhir_extension": {
      "fhir_endpoint": "",
      "fhir_version": "",
      "fhir_resources": {}
    }
```

Figure 15: Review the generated BCO file

At the bottom of the review part, users can access some export and save features (Figure 16). The first one is the feature to export the generated BCO file in JSON file format, "**Export as JSON**". When it is clicked, the generated BCO file is downloaded as a JSON file. You can see the example of the exported file in Figure 17.

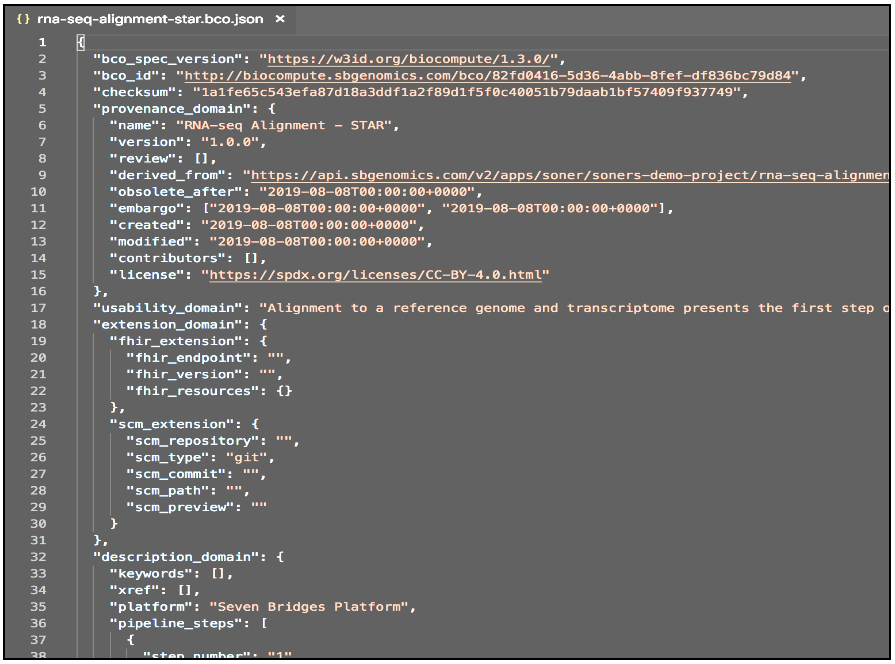


Export as JSON/PDF

Save to the Platform

Save to Git Project

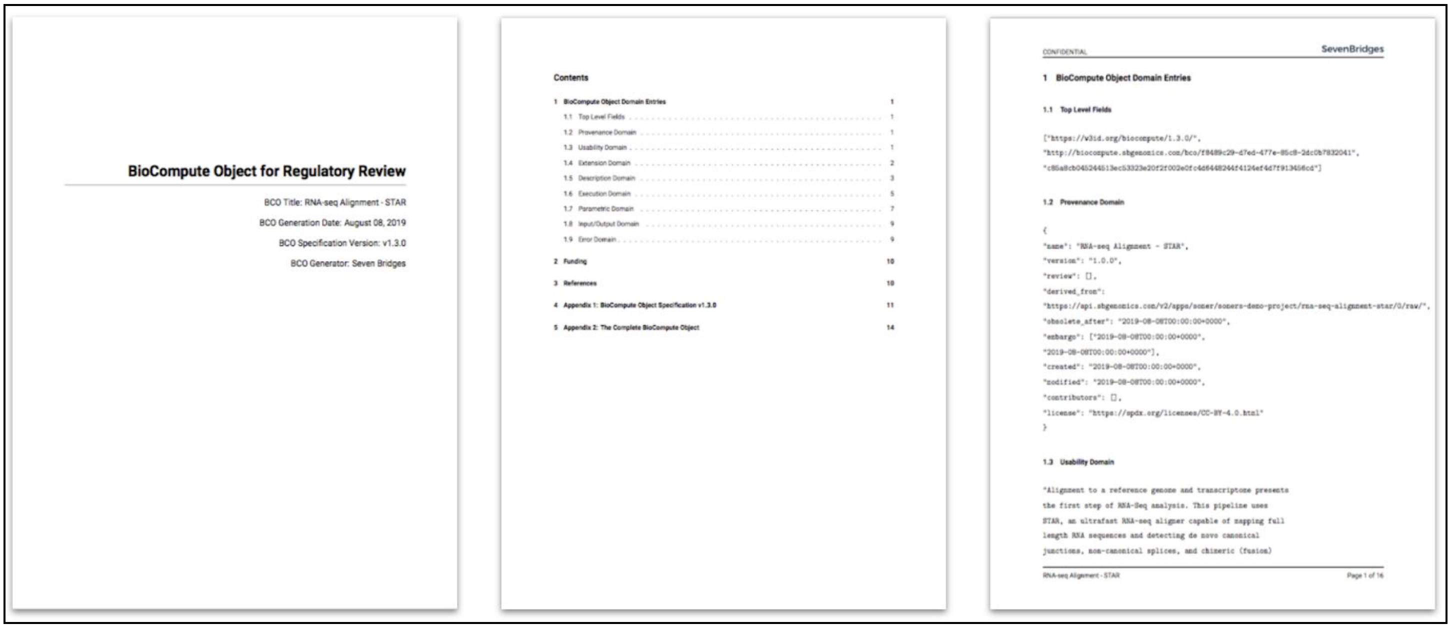
Figure 16: Export and Save Features in Step 6



```
{
  "bco_spec_version": "https://w3id.org/biocompute/1.3.0/",
  "bco_id": "http://biocompute.sbgenomics.com/bco/82fd0416-5d36-4abb-8fef-df836bc79d84",
  "checksum": "1a1fe65c543efa87d18a3ddf1a2f89d1f5f0c40051b79daab1bf57409f937749",
  "provenance_domain": {
    "name": "RNA-seq Alignment - STAR",
    "version": "1.0.0",
    "review": [],
    "derived_from": "https://api.sbgenomics.com/v2/apps/soner/soners-demo-project/rna-seq-alignmen
    "obsolete_after": "2019-08-08T00:00:00+0000",
    "embargo": ["2019-08-08T00:00:00+0000", "2019-08-08T00:00:00+0000"],
    "created": "2019-08-08T00:00:00+0000",
    "modified": "2019-08-08T00:00:00+0000",
    "contributors": [],
    "license": "https://spdx.org/licenses/CC-BY-4.0.html"
  },
  "usability_domain": "Alignment to a reference genome and transcriptome presents the first step o
  "extension_domain": {
    "fhir_extension": {
      "fhir_endpoint": "",
      "fhir_version": "",
      "fhir_resources": {}
    },
    "scm_extension": {
      "scm_repository": "",
      "scm_type": "git",
      "scm_commit": "",
      "scm_path": "",
      "scm_preview": ""
    }
  },
  "description_domain": {
    "keywords": [],
    "xref": [],
    "platform": "Seven Bridges Platform",
    "pipeline_steps": [
      {
        "step_number": "1"
```

Figure 17: Example View of the Downloaded BCO JSON file

As the second feature, users can generate a PDF reports for the BioCompute Object with "**Export as PDF**" button. In the report, users can find all domains of the generated BCO file, BioCompute Object specifications, and typed raw JSON of the generated BCO, with white spaces for reviewers to review (Figure 18).



BioCompute Object for Regulatory Review

BCO Title: RNA-seq Alignment - STAR

BCO Generation Date: August 08, 2019

BCO Specification Version: v1.3.0

BCO Generator: Seven Bridges

Contents

1 **BioCompute Object Domain Entries** 1
- 1.1 Top Level Fields . . . 1
- 1.2 Provenance Domain . . . 1
- 1.3 Usability Domain . . . 1
- 1.4 Extension Domain . . . 2
- 1.5 Description Domain . . . 3
- 1.6 Execution Domain . . . 5
- 1.7 Parametric Domain . . . 7
- 1.8 Input/Output Domain . . . 9
- 1.9 Error Domain . . . 9

2 **Funding** 10

3 **References** 10

4 **Appendix 1: BioCompute Object Specification v1.3.0** 11

5 **Appendix 2: The Complete BioCompute Object** 14

CONFIDENTIAL SevenBridges

1 BioCompute Object Domain Entries

1.1 Top Level Fields

```
["https://w3id.org/biocompute/1.3.0/",
"http://biocompute.sbgenomics.com/bco/f8489c29-d7ed-477e-85c8-2dc0b7832041",
"c85a8cb045244513ec53323e20f2f002e0fc4d6448244f4124ef4d7f913456cd"]
```

1.2 Provenance Domain

```
{
"name": "RNA-seq Alignment - STAR",
"version": "1.0.0",
"review": [],
"derived_from":
"https://api.sbgenomics.com/v2/apps/soner/soners-demo-project/rna-seq-alignment-star/0/raw/",
"obsolete_after": "2019-08-08T00:00:00+0000",
"embargo": ["2019-08-08T00:00:00+0000",
"2019-08-08T00:00:00+0000"],
"created": "2019-08-08T00:00:00+0000",
"modified": "2019-08-08T00:00:00+0000",
"contributors": [],
"license": "https://spdx.org/licenses/CC-BY-4.0.html"
}
```

1.3 Usability Domain

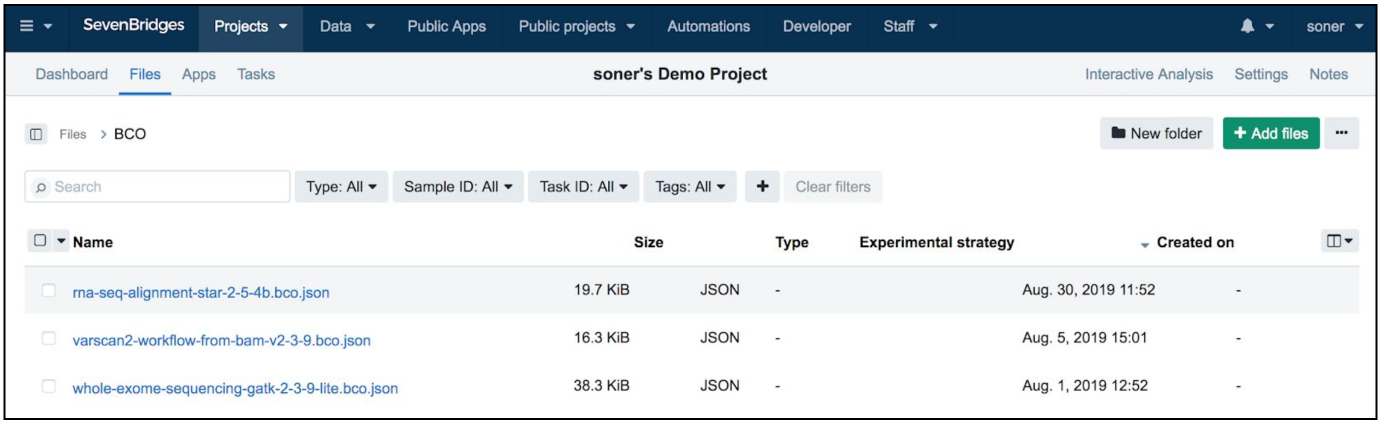
```
"Alignment to a reference genome and transcriptome presents
the first step of RNA-Seq analysis. This pipeline uses
STAR, an ultrafast RNA-seq aligner capable of mapping full
length RNA sequences and detecting de novo canonical
junctions, non-canonical splices, and chimeric (fusion)
```

RNA-seq Alignment - STAR Page 1 of 16

Figure 18: Example View of the Generated PDF Report in Step 6

In addition to export features, there are two very useful save features in the Genomics Compliance Suite: **Save to the Platform** and **Save to GitHub**.

The first one is **Save to the Platform**, automatically saves the generated BCO file into "**BCO**" folder under "**Files**" of your project. You can see the example of the generated files in the project's "**Files/BCO**" folder in Figure 19. Also, there is no need to open the platform in the external browser window. When "**Save to the Platform**" button is clicked, the direct link button (to the platform) appears near the "**Save to the Platform**" button to access the generated file on the platform easily (Figure 20).

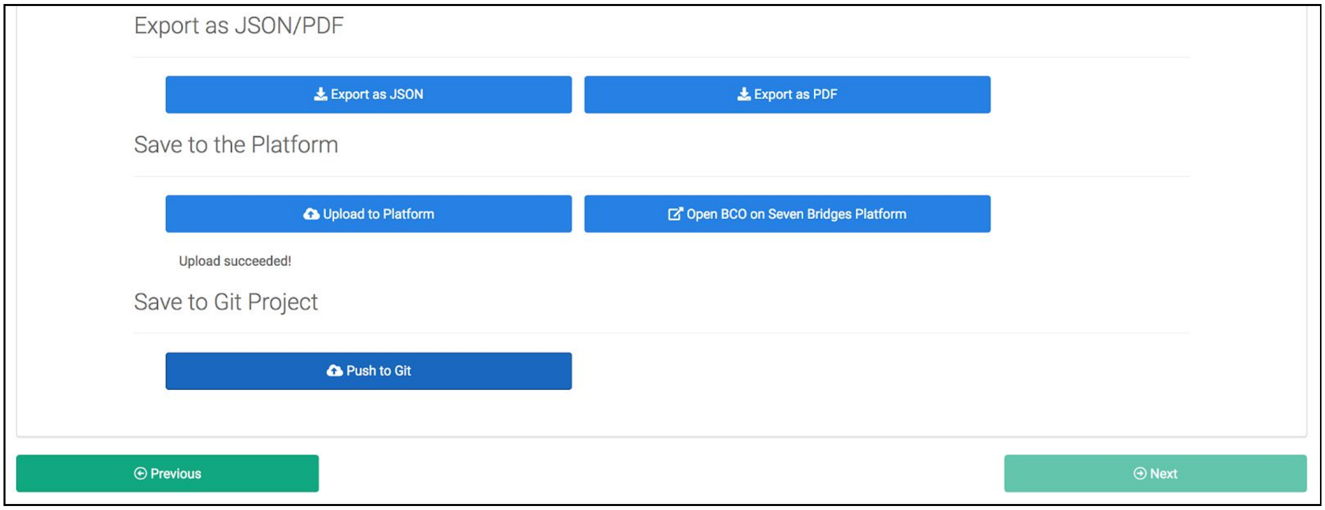


soner's Demo Project

Files > BCO

□ Name	Size	Type	Experimental strategy	Created on	
□ rna-seq-alignment-star-2-5-4b.bco.json	19.7 KiB	JSON	-	Aug. 30, 2019 11:52	-
□ varscan2-workflow-from-bam-v2-3-9.bco.json	16.3 KiB	JSON	-	Aug. 5, 2019 15:01	-
□ whole-exome-sequencing-gatk-2-3-9-lite.bco.json	38.3 KiB	JSON	-	Aug. 1, 2019 12:52	-

Figure 19: Generated BCO Exported to the Platform

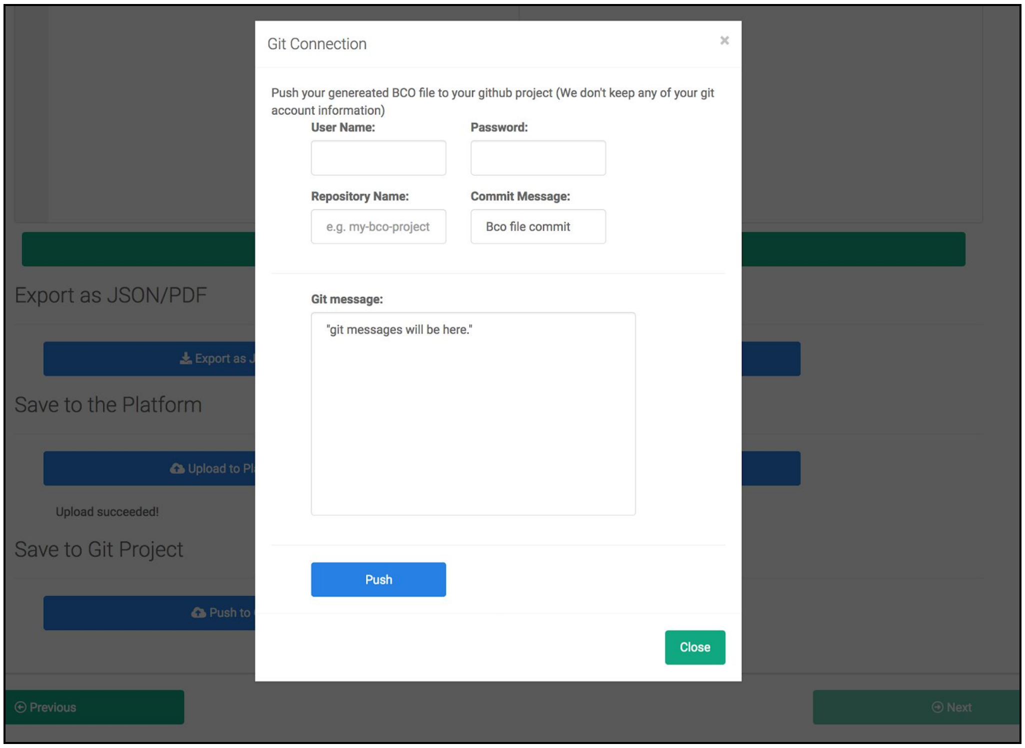


Upload succeeded!

Figure 20: Save to the Platform with Direct Link to the Platform in Step 6

As a second, **Save to GitHub**, automatically saves the generated BCO file into your Github account to make your projects up to date and easy-to-access to your BCO files. You can see the **GitHub Connection** popup window, when your click the **Push to GitHub** button (Figure 21).

The window will need the **username** and **password** of the user's GitHub account to make the connection and push operations. Then, the user can type the target **Repository Name** to push the generated BCO file. Also, there is the **Git Commit Message** field that is the explanatory message for the push operation. In the end, when the user clicks on **Push**, results of the push operation will be listed in **Git commit message** area (Figure 22).



Push your genereated BCO file to your github project (We don't keep any of your git account information)

User Name:

Password:

Repository Name: e.g. my-bco-project

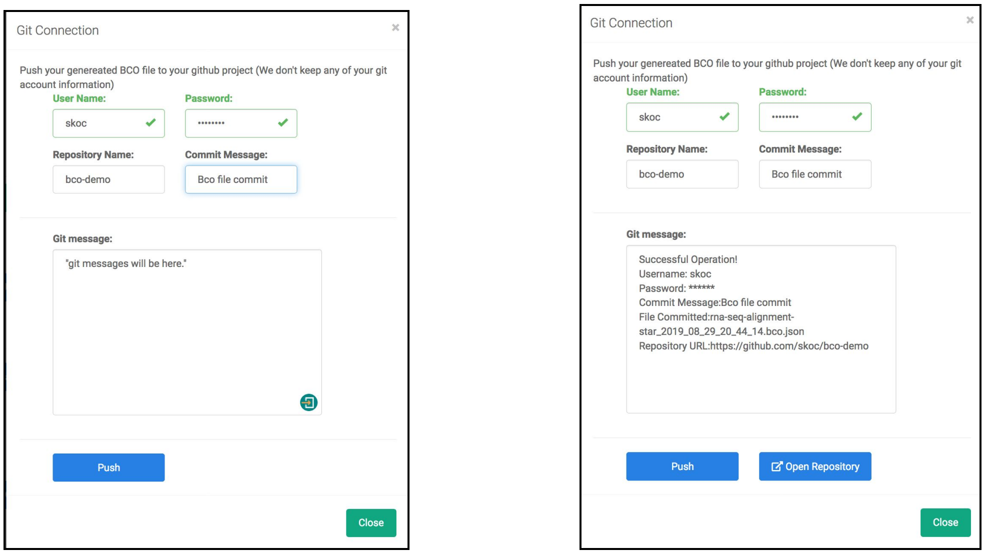
Commit Message: Bco file commit

Git message:

"git messages will be here."

Upload succeeded!

Figure 21: Git Connection Window in Step 6



Push your genereated BCO file to your github project (We don't keep any of your git account information)

User Name: skoc

Password: ••••••••

Repository Name: bco-demo

Commit Message: Bco file commit

Git message:

"git messages will be here."

Push your genereated BCO file to your github project (We don't keep any of your git account information)

User Name: skoc

Password: ••••••••

Repository Name: bco-demo

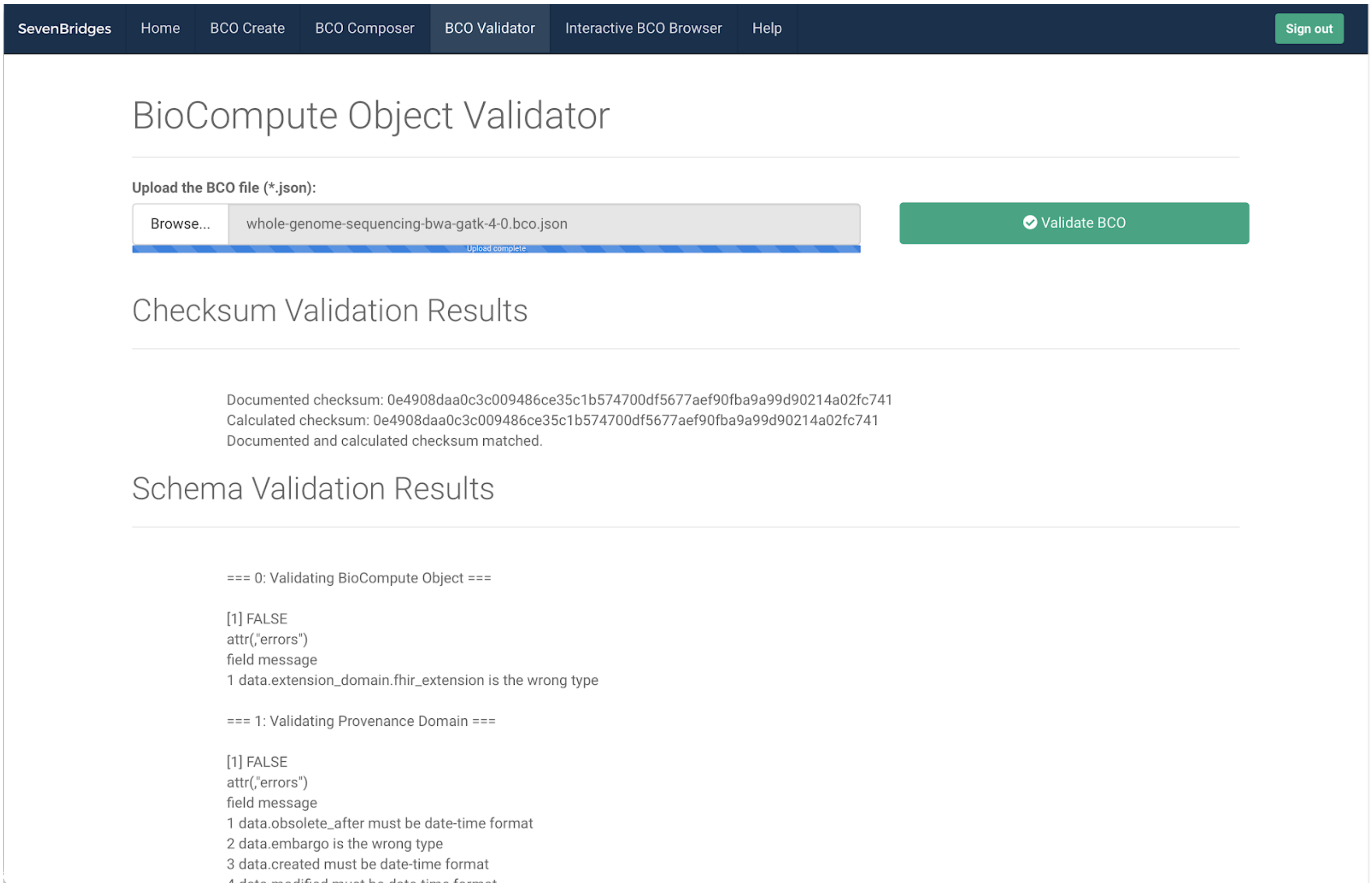
Commit Message: Bco file commit

Git message:

```
Successful Operation!
Username: skoc
Password: ******
Commit Message:Bco file commit
File Committed:rna-seq-alignment-
star_2019_08_29_20_44_14.bco.json
Repository URL:https://github.com/skoc/bco-demo
```

Figure 22: Git Connection Before/After Push Operation in Step 6

BioCompute Object (BCO) Validator



BioCompute Object Validator

Upload the BCO file (*.json):

Checksum Validation Results

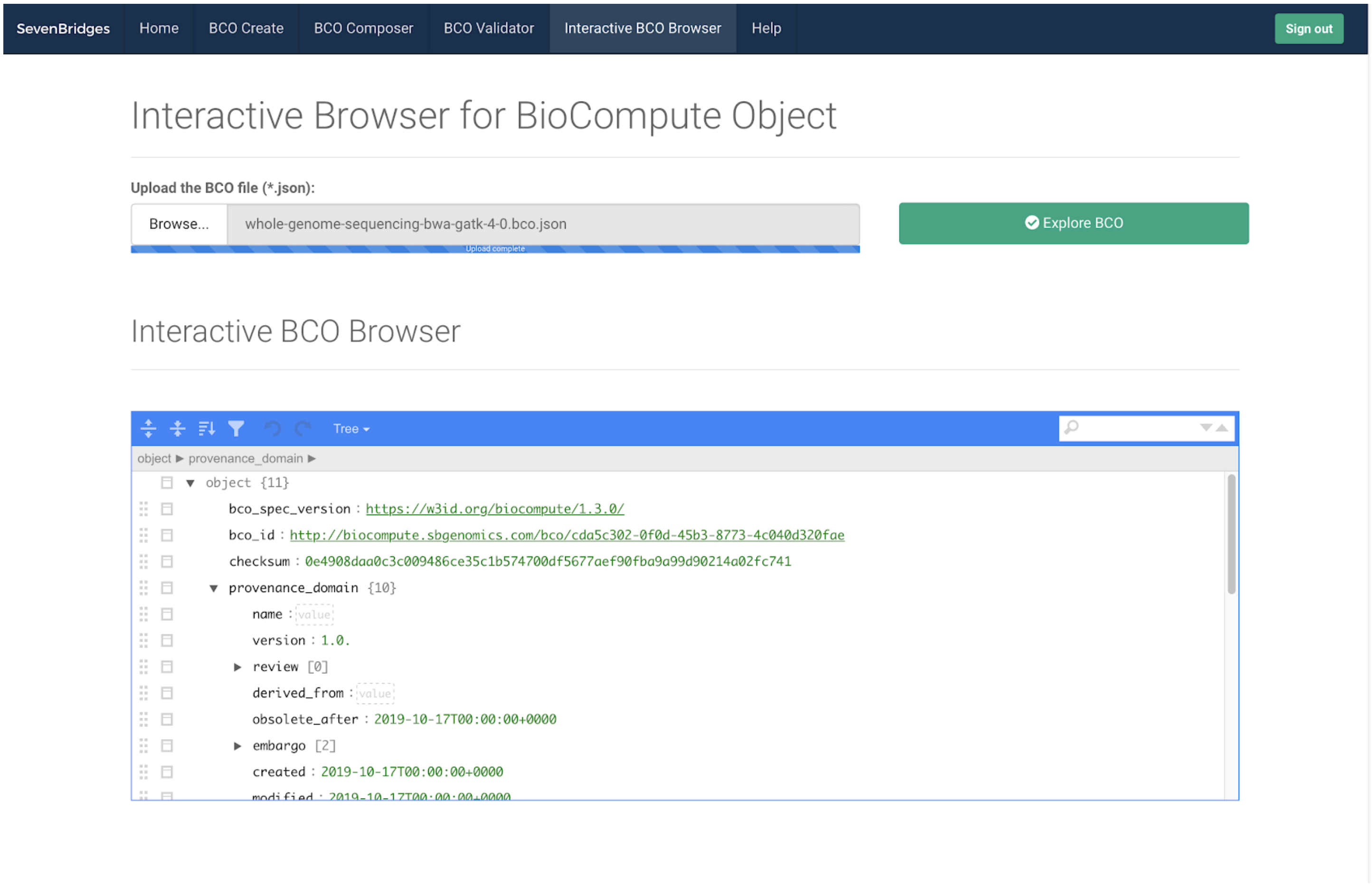
```
Documented checksum: 0e4908daa0c3c009486ce35c1b574700df5677aef90fba9a99d90214a02fc741
Calculated checksum: 0e4908daa0c3c009486ce35c1b574700df5677aef90fba9a99d90214a02fc741
Documented and calculated checksum matched.
```

Schema Validation Results

```
=== 0: Validating BioCompute Object ===

[1] FALSE
attr(,"errors")
field message
1 data.extension_domain.fhir_extension is the wrong type

=== 1: Validating Provenance Domain ===

[1] FALSE
attr(,"errors")
field message
1 data.obsolete_after must be date-time format
2 data.embargo is the wrong type
3 data.created must be date-time format
4 data.modified must be date-time format
```

Figure 23: A screenshot of the BCO validator UI.

Interactive BCO Browser



Interactive Browser for BioCompute Object

Upload the BCO file (*.json):

Browse... whole-genome-sequencing-bwa-gatk-4-0.bco.json

Upload complete

Explore BCO

Interactive BCO Browser

```
▼ object {11}
    bco_spec_version : https://w3id.org/biocompute/1.3.0/
    bco_id : http://biocompute.sbgenomics.com/bco/cda5c302-0f0d-45b3-8773-4c040d320fae
    checksum : 0e4908daa0c3c009486ce35c1b574700df5677aef90fba9a99d90214a02fc741
  ▼ provenance_domain {10}
        name : value
        version : 1.0.
      ▶ review [0]
        derived_from : value
        obsolete_after : 2019-10-17T00:00:00+0000
      ▶ embargo [2]
        created : 2019-10-17T00:00:00+0000
        modified : 2019-10-17T00:00:00+0000
```

Figure 24: A screenshot of the interactive BCO browser UI.

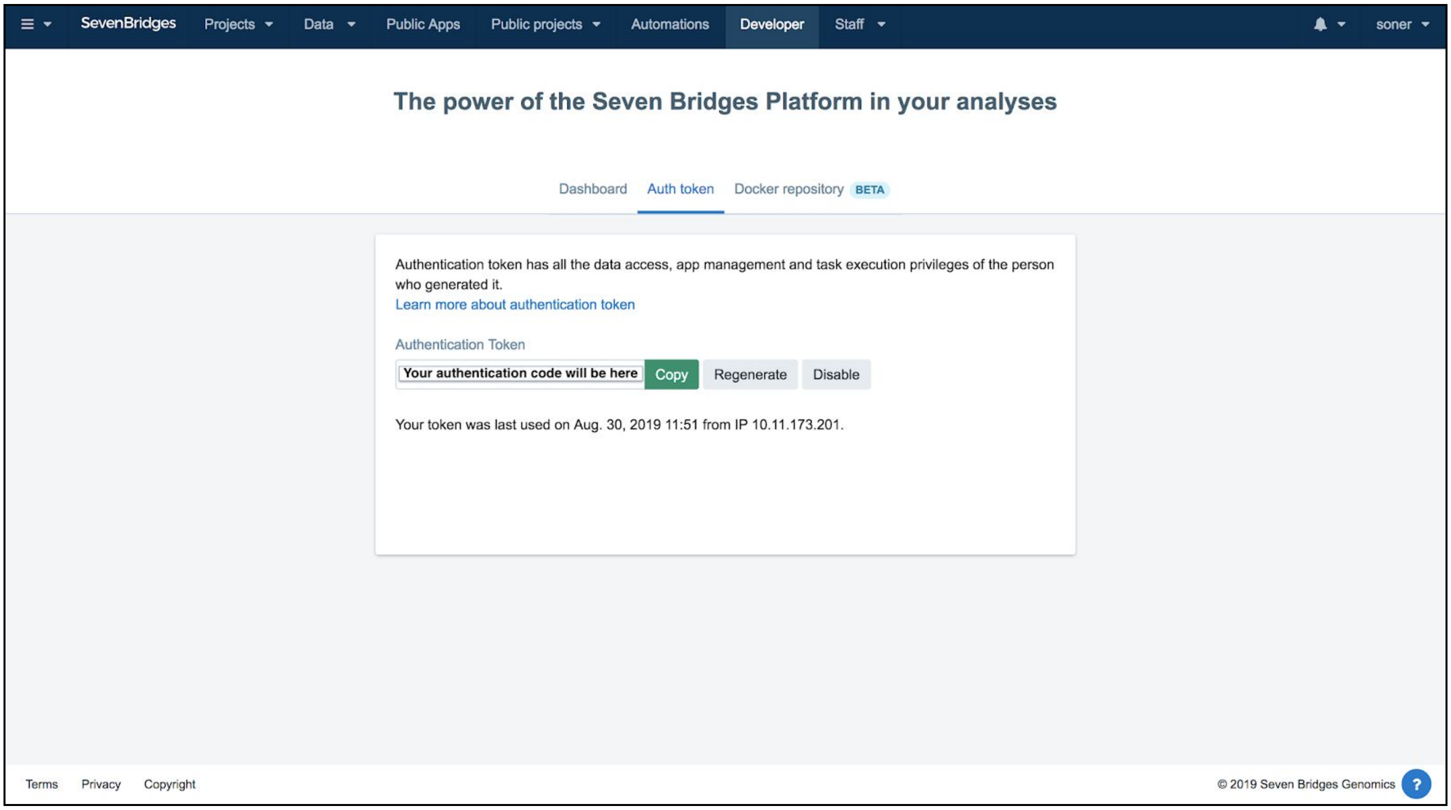
Appendix I : Platform Authentication Guide

Authentication Token

Your authentication token encodes your Seven Bridges Platform credentials and uniquely identifies you on the Seven Bridges Platform. It can be used with a number of API and automation features of the Seven Bridges Platform without the need of a manual, GUI authentication process.

Locate Authentication Token

You can get your authentication token from the developer dashboard. For example, for Seven Bridges Platform users, click here to go to the developer dashboard to generate a new token or access your token (Figure 23).



The power of the Seven Bridges Platform in your analyses

Authentication token has all the data access, app management and task execution privileges of the person who generated it.
Learn more about authentication token

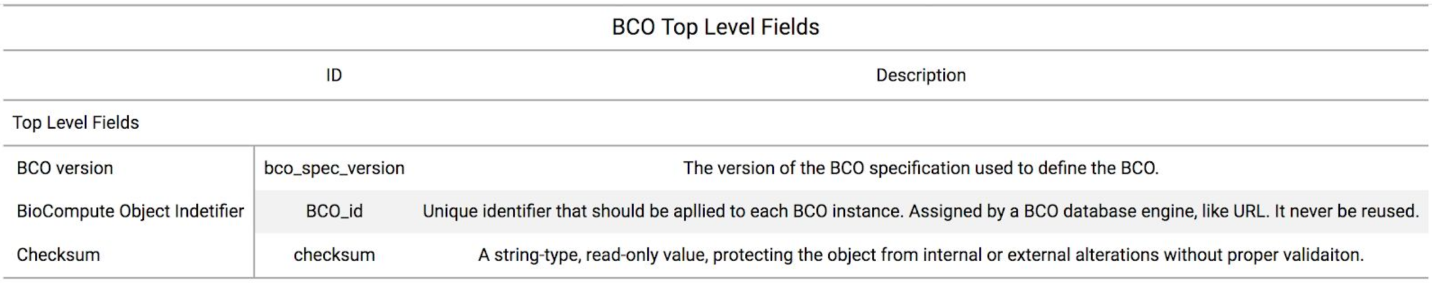
Authentication Token: Your authentication code will be here

Your token was last used on Aug. 30, 2019 11:51 from IP 10.11.173.201.

© 2019 Seven Bridges Genomics

Figure 25: Authentication Token Generation Page of Seven Bridges Platform

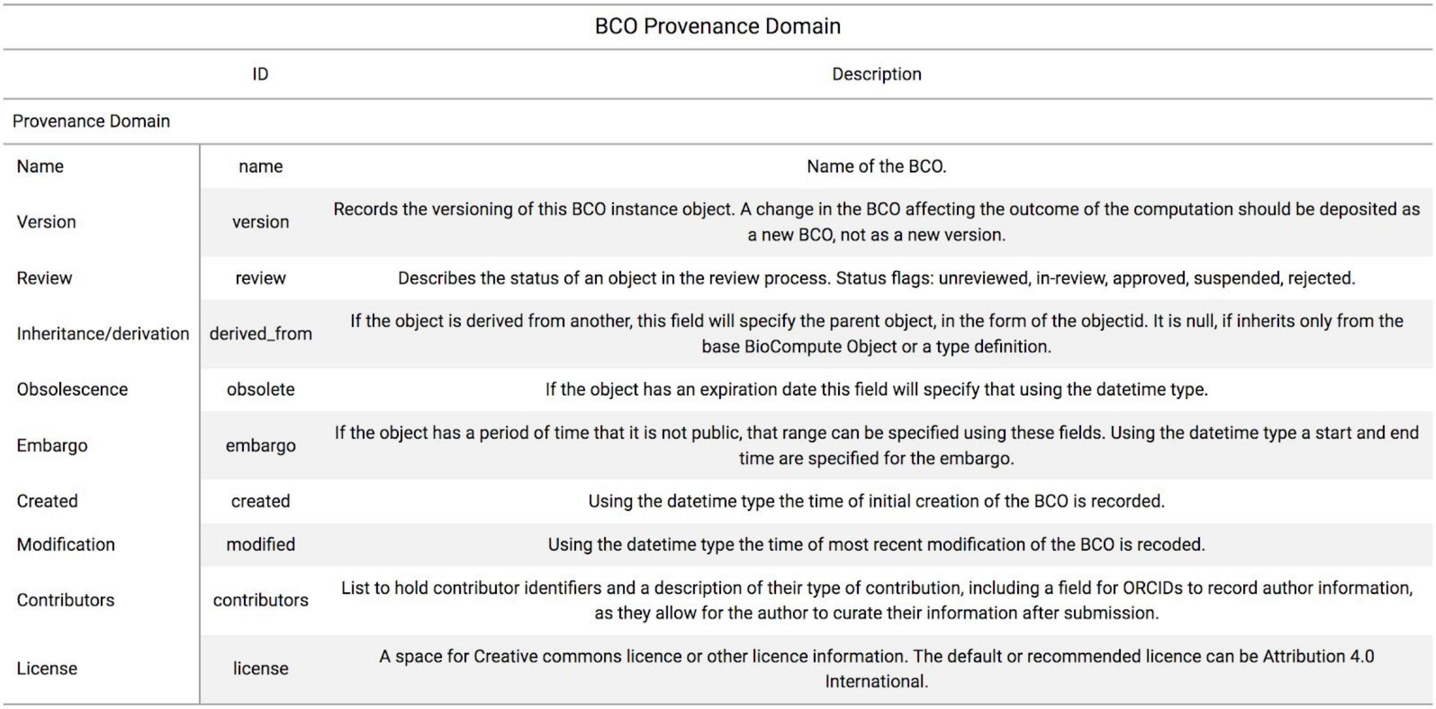
Appendix II: BioCompute Object Domains



BCO Top Level Fields

ID		Description
Top Level Fields		
BCO version	bco_spec_version	The version of the BCO specification used to define the BCO.
BioCompute Object Indetifier	BCO_id	Unique identifier that should be apllied to each BCO instance. Assigned by a BCO database engine, like URL. It never be reused.
Checksum	checksum	A string-type, read-only value, protecting the object from internal or external alterations without proper validaiton.

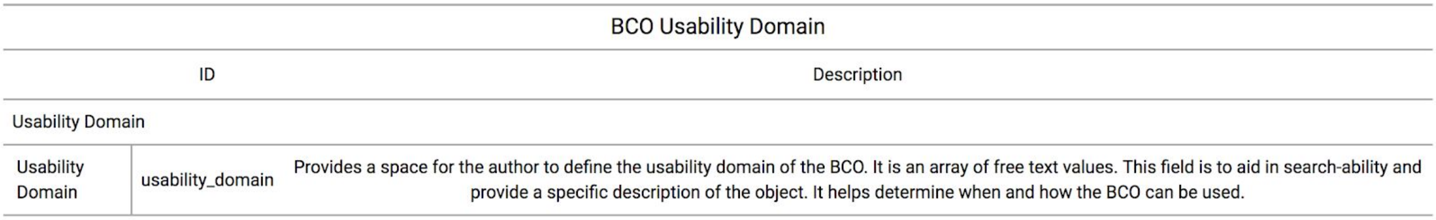
Table 1: Top Level Fields of BCO



BCO Provenance Domain

ID		Description
Provenance Domain		
Name	name	Name of the BCO.
Version	version	Records the versioning of this BCO instance object. A change in the BCO affecting the outcome of the computation should be deposited as a new BCO, not as a new version.
Review	review	Describes the status of an object in the review process. Status flags: unreviewed, in-review, approved, suspended, rejected.
Inheritance/derivation	derived_from	If the object is derived from another, this field will specify the parent object, in the form of the objectid. It is null, if inherits only from the base BioCompute Object or a type definition.
Obsolescence	obsolete	If the object has an expiration date this field will specify that using the datetime type.
Embargo	embargo	If the object has a period of time that it is not public, that range can be specified using these fields. Using the datetime type a start and end time are specified for the embargo.
Created	created	Using the datetime type the time of initial creation of the BCO is recorded.
Modification	modified	Using the datetime type the time of most recent modification of the BCO is recoded.
Contributors	contributors	List to hold contributor identifiers and a description of their type of contribution, including a field for ORCIDs to record author information, as they allow for the author to curate their information after submission.
License	license	A space for Creative commons licence or other licence information. The default or recommended licence can be Attribution 4.0 International.

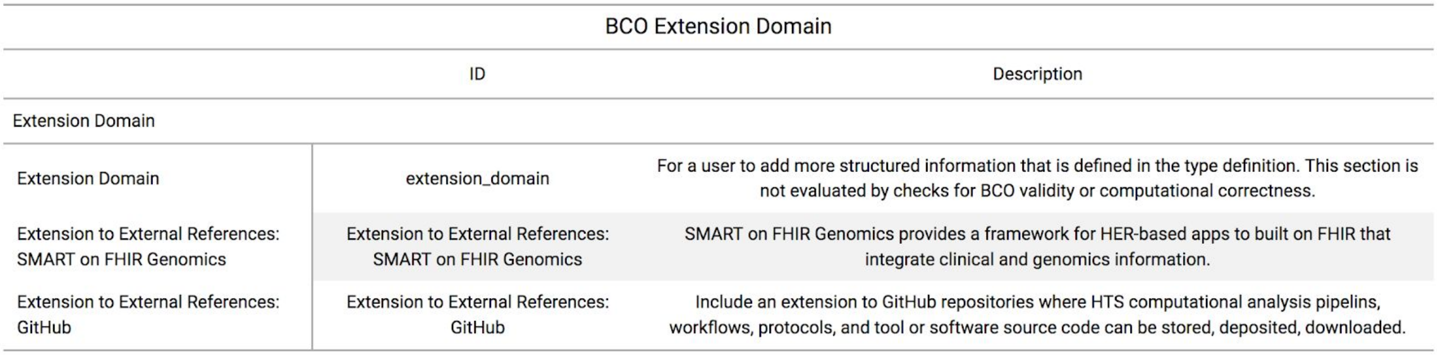
Table 2: Provenance Domain of BCO



BCO Usability Domain

ID		Description
Usability Domain		
Usability Domain	usability_domain	Provides a space for the author to define the usability domain of the BCO. It is an array of free text values. This field is to aid in search-ability and provide a specific description of the object. It helps determine when and how the BCO can be used.

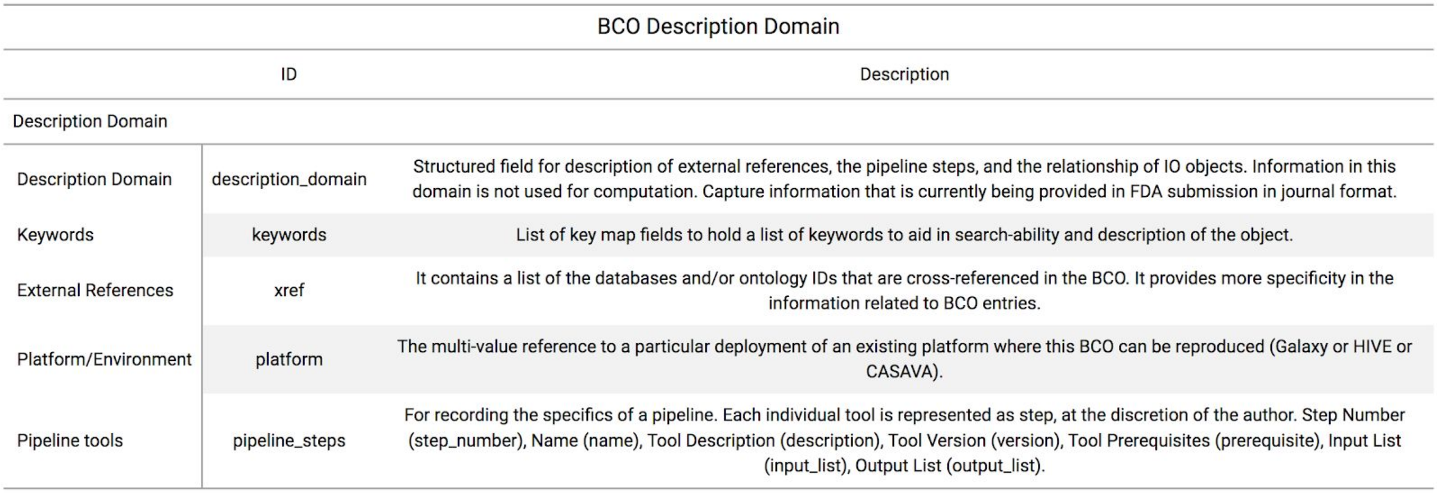
Table 3: Usability Domain of BCO



BCO Extension Domain

	ID	Description
Extension Domain		
Extension Domain	extension_domain	For a user to add more structured information that is defined in the type definition. This section is not evaluated by checks for BCO validity or computational correctness.
Extension to External References: SMART on FHIR Genomics	Extension to External References: SMART on FHIR Genomics	SMART on FHIR Genomics provides a framework for HER-based apps to built on FHIR that integrate clinical and genomics information.
Extension to External References: GitHub	Extension to External References: GitHub	Include an extension to GitHub repositories where HTS computational analysis pipelins, workflows, protocols, and tool or software source code can be stored, deposited, downloaded.

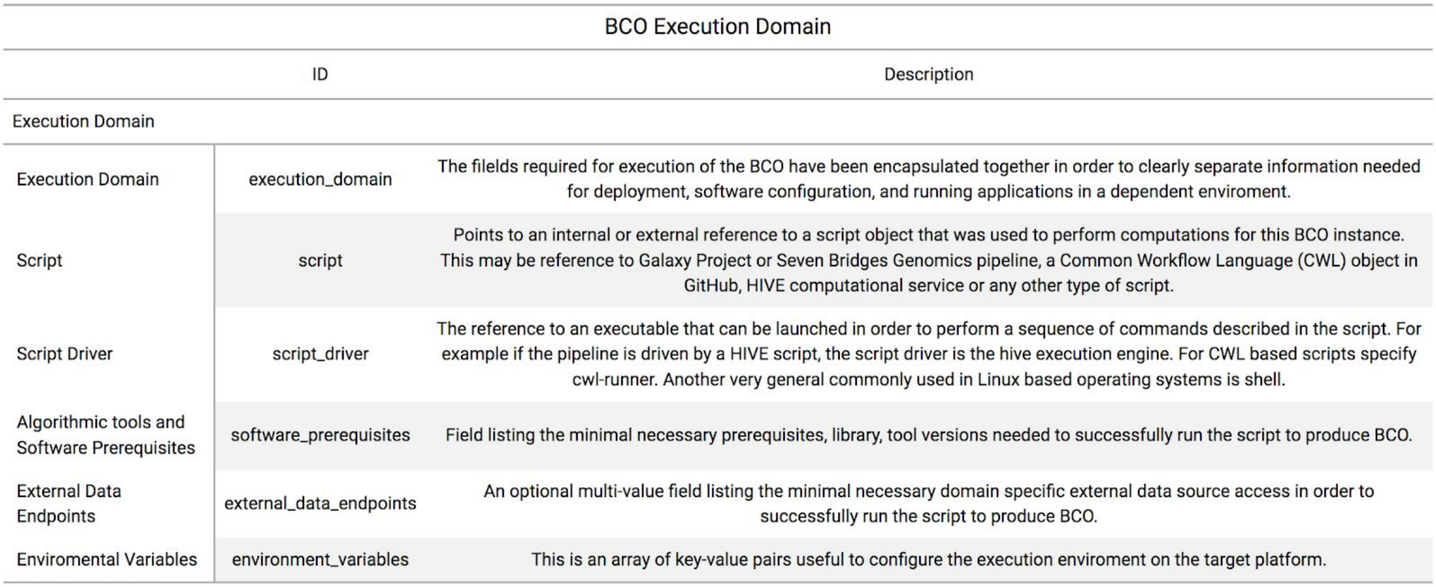
Table 4: Extension Domain of BCO



BCO Description Domain

	ID	Description
Description Domain		
Description Domain	description_domain	Structured field for description of external references, the pipeline steps, and the relationship of IO objects. Information in this domain is not used for computation. Capture information that is currently being provided in FDA submission in journal format.
Keywords	keywords	List of key map fields to hold a list of keywords to aid in search-ability and description of the object.
External References	xref	It contains a list of the databases and/or ontology IDs that are cross-referenced in the BCO. It provides more specificity in the information related to BCO entries.
Platform/Environment	platform	The multi-value reference to a particular deployment of an existing platform where this BCO can be reproduced (Galaxy or HIVE or CASAVA).
Pipeline tools	pipeline_steps	For recording the specifics of a pipeline. Each individual tool is represented as step, at the discretion of the author. Step Number (step_number), Name (name), Tool Description (description), Tool Version (version), Tool Prerequisites (prerequisite), Input List (input_list), Output List (output_list).

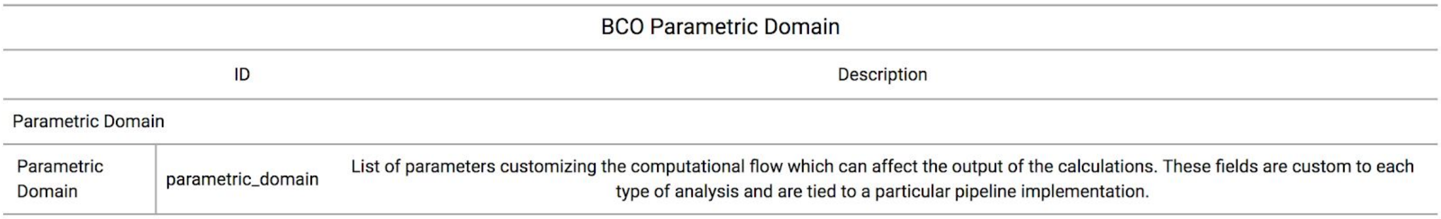
Tabler 5: Description Domain of BCO



BCO Execution Domain

ID		Description
Execution Domain		
Execution Domain	execution_domain	The filelds required for execution of the BCO have been encapsulated together in order to clearly separate information needed for deployment, software configuration, and running applications in a dependent enviroment.
Script	script	Points to an internal or external reference to a script object that was used to perform computations for this BCO instance. This may be reference to Galaxy Project or Seven Bridges Genomics pipeline, a Common Workflow Language (CWL) object in GitHub, HIVE computational service or any other type of script.
Script Driver	script_driver	The reference to an executable that can be launched in order to perform a sequence of commands described in the script. For example if the pipeline is driven by a HIVE script, the script driver is the hive execution engine. For CWL based scripts specify cwl-runner. Another very general commonly used in Linux based operating systems is shell.
Algorithmic tools and Software Prerequisites	software_prerequisites	Field listing the minimal necessary prerequisites, library, tool versions needed to successfully run the script to produce BCO.
External Data Endpoints	external_data_endpoints	An optional multi-value field listing the minimal necessary domain specific external data source access in order to successfully run the script to produce BCO.
Enviromental Variables	environment_variables	This is an array of key-value pairs useful to configure the execution enviroment on the target platform.

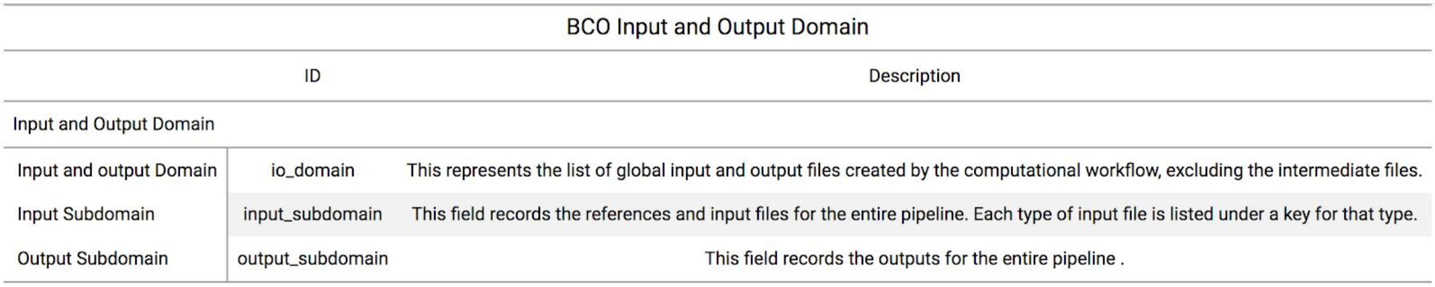
Table 6: Execution Domain of BCO



BCO Parametric Domain

ID		Description
Parametric Domain		
Parametric Domain	parametric_domain	List of parameters customizing the computational flow which can affect the output of the calculations. These fields are custom to each type of analysis and are tied to a particular pipeline implementation.

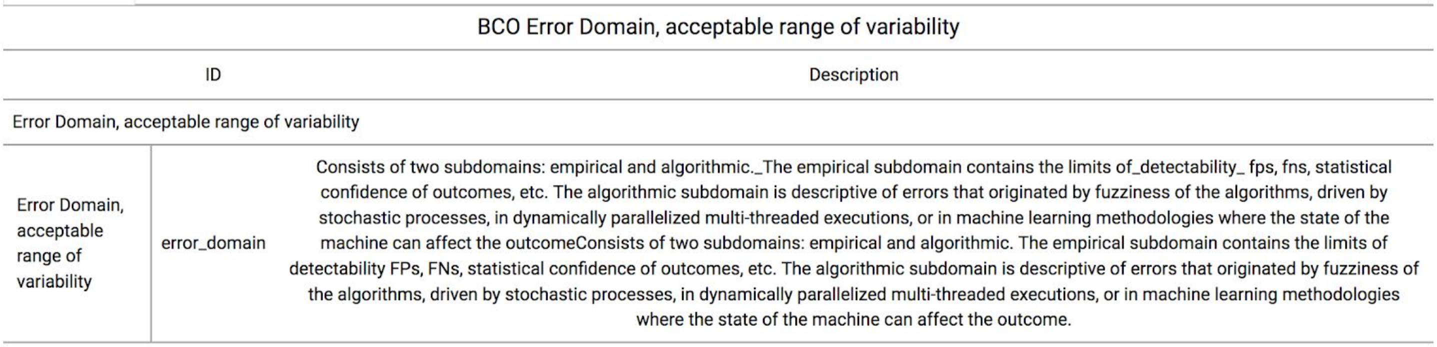
Table 7: Parametric Domain of BCO



BCO Input and Output Domain

ID		Description
Input and Output Domain		
Input and output Domain	io_domain	This represents the list of global input and output files created by the computational workflow, excluding the intermediate files.
Input Subdomain	input_subdomain	This field records the references and input files for the entire pipeline. Each type of input file is listed under a key for that type.
Output Subdomain	output_subdomain	This field records the outputs for the entire pipeline .

Table 8: I/O Domain of BCO



BCO Error Domain, acceptable range of variability

ID		Description
Error Domain, acceptable range of variability		
Error Domain, acceptable range of variability	error_domain	Consists of two subdomains: empirical and algorithmic._The empirical subdomain contains the limits of_detectability_ fps, fns, statistical confidence of outcomes, etc. The algorithmic subdomain is descriptive of errors that originated by fuzziness of the algorithms, driven by stochastic processes, in dynamically parallelized multi-threaded executions, or in machine learning methodologies where the state of the machine can affect the outcomeConsists of two subdomains: empirical and algorithmic. The empirical subdomain contains the limits of detectability FPs, FNs, statistical confidence of outcomes, etc. The algorithmic subdomain is descriptive of errors that originated by fuzziness of the algorithms, driven by stochastic processes, in dynamically parallelized multi-threaded executions, or in machine learning methodologies where the state of the machine can affect the outcome.

Table 9: Error Domain of BCO