

GCS: BioCompute Object Genomics Compliance Suite User Manual

March 2020

Contents

1 Goals	1
2 Getting Started with the Genomics Compliance Suite (GCS)	1
2.1 App Access and Login	1
2.1.1 Containerized app	1
2.1.2 Seven Bridges hosted app	1
2.1.3 RStudio Server on the Cancer Genomics Cloud	2
2.2 Start Page	3
2.3 Platform Composer	4
2.3.1 Step 1 - Import the Target CWL App	4
2.3.2 Step 2 - Provenance, Usability, and Extension Domains	6
2.3.3 Step 3 - Execution and Parametric Domains	8
2.3.4 Step 4 - Description Domain	10
2.3.5 Step 5 - I/O and Error Domain	10
2.3.6 Step 6 - Review and Export	12
2.4 BioCompute Object (BCO) Validator	19
2.5 Interactive BCO Browser	20
Appendix I : Platform Authentication Guide	21
Authentication Token	21
Locate Authentication Token	21
Appendix II: BioCompute Object Domains	22

1 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.

2 Getting Started with the Genomics Compliance Suite (GCS)

This section describes the basic usage of the GCS app.

2.1 App Access and Login

The GCS app is designed to be easily deployed locally for testing or on any remote servers for production. A containerized (Docker) 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.

2.1.1 Containerized app

Pull or build the Docker image from GitHub (see README.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).

2.1.2 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.sbggenomics.com/gcs/> (behind the VPN) and log in with your username and password.

2.1.3 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.

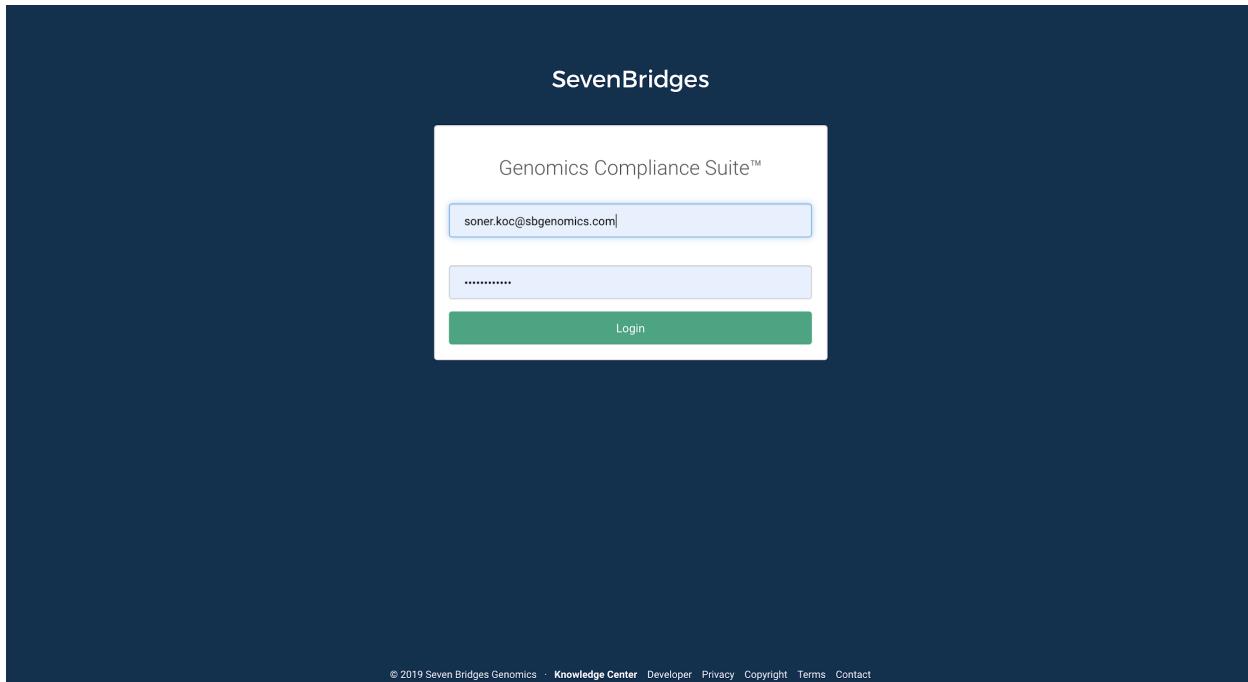


Figure 1: Login Page of the Genomics Compliance Suite

2.2 Start Page

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

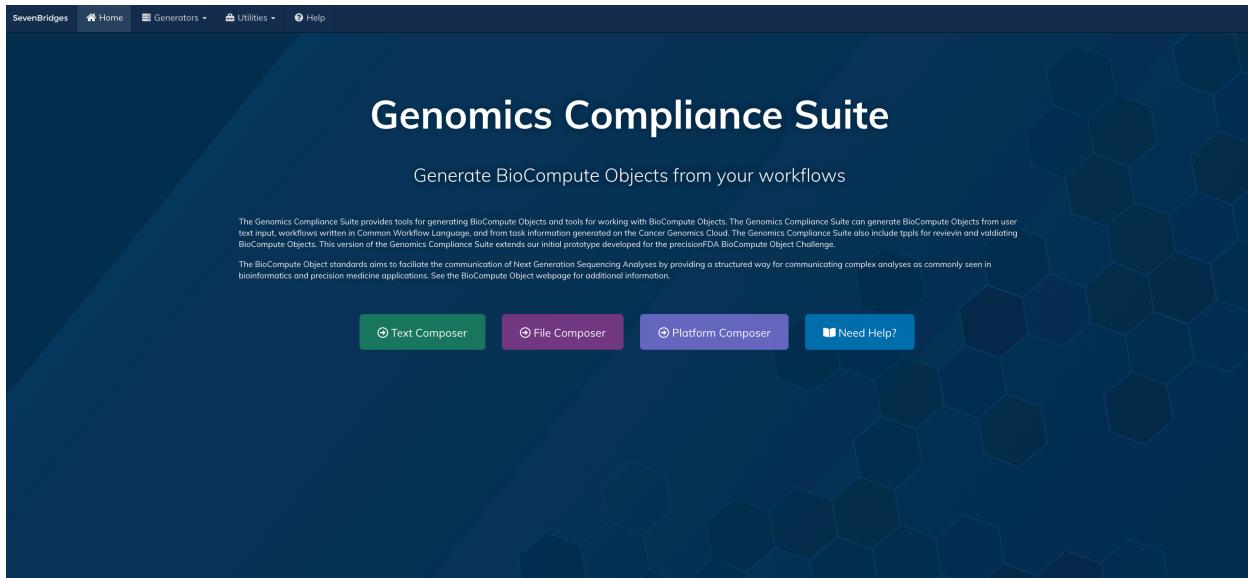


Figure 2: Landing Page of the Genomics Compliance Suite

In the middle of all, you see the quick action buttons; you can quickly start to build your BioCompute (BCO) object file with any of the composer buttons: “**Test Composer**”, “**File Composer**”, and “**Platform Composer**”, 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, **Generators** which includes all composers, **Utilities** which includes “**Browser**”, “**Validator**”, and “**Standard**”, and lastly **Help**.

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

2.3 Platform Composer

2.3.1 Step 1 - Import the Target CWL App

After you click the “**Platform Composer**” page of the GCS platform under “**Generators**” tab, you will see the initial step of the BCO generation, import step (Figure 3).

The screenshot shows the "Step 1/6 - Import" page of the Platform Composer. At the top, there is a navigation bar with links for Home, Generators, Utilities, and Help. Below the navigation bar, a progress bar indicates the current step is "Step 1 Import". The progress bar has six steps labeled: 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, and Step 6 Review & Export. The main content area is titled "Import from Seven Bridges Platforms". It contains fields for "Platform" (set to "Cancer Genomics Cloud (CGC)"), "Auth Token" (a text input field), "Choose Project" (a dropdown menu), "Choose Task in Project (Optional - If Exist)" (another dropdown menu), and "Choose App to Import CWL Workflow" (a third dropdown menu). At the bottom, there are "Previous" and "Next" buttons.

Figure 3: Initial Page of the Platform 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.

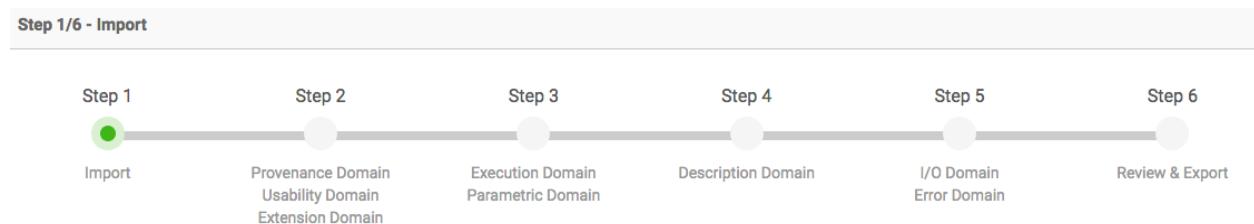


Figure 4: Steps of the Platform Composer

Import from Seven Bridges Platforms

Platform: Cancer Genomics Cloud (CGC)

Paste the Auth Token from the [CGC Developer Dashboard](#)

Choose Project: Choose Task in Project (Optional - If Exist)

Choose App to Import CWL Workflow

Figure 5: Necessary information fields at the beginning of the Platform 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

Platform
Cancer Genomics Cloud (CGC)

Paste the Auth Token from the [CGC Developer Dashboard](#)

Choose Project
pdxnet/pdxnet-datapool

Choose Task in Project (Optional - If Exist)

Choose App to Import CWL Workflow
pdx-wxs-tumor-normal-updatedgatk

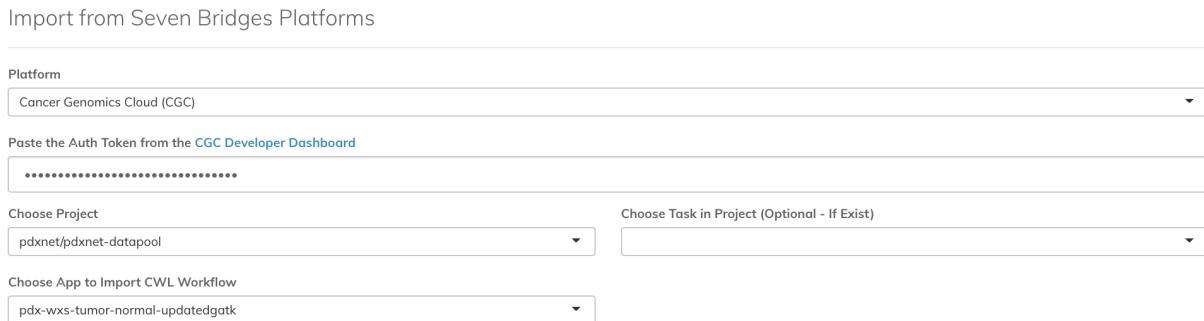


Figure 6: Illustration of the necessary information fields

Import from Seven Bridges Platforms

Platform
Cancer Genomics Cloud (CGC)

Paste the Auth Token from the [CGC Developer Dashboard](#)

Choose Project
pdxnet/pdxnet-datapool

Choose Task in Project (Optional - If Exist)
BCM - PDXnet RNA Expression Estimation Workflow run - 10-30-19 20:21:53

Choose App to Import CWL Workflow
rsem-1-2-31-workflow-with-star-aligner-pe

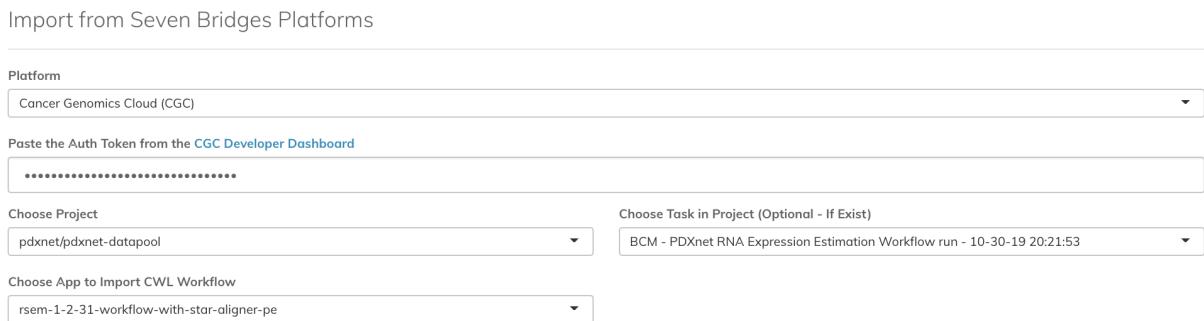


Figure 7: Illustration of the necessary information fields with the optional (task) selection

2.3.2 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.

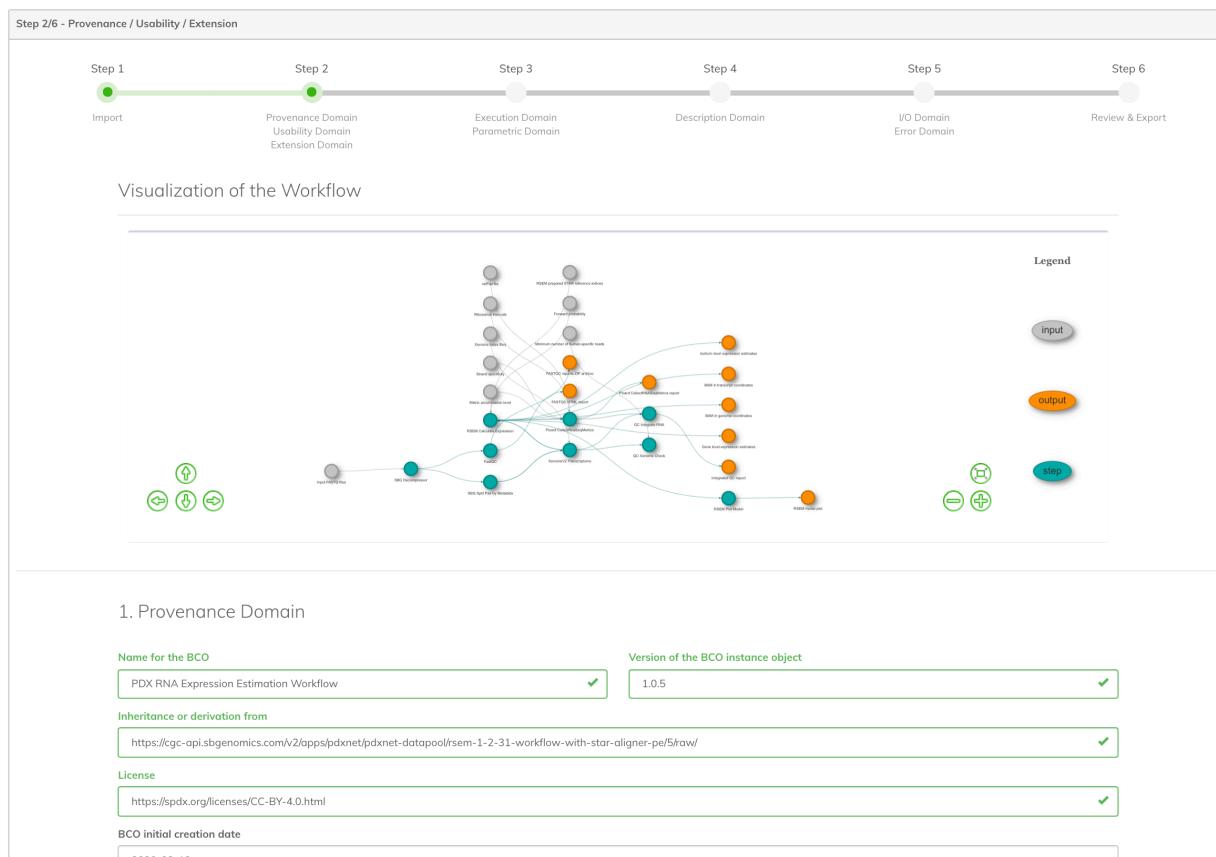


Figure 8: Filled fields, by CWL of the selected application, of domains exist in Step 2.

The screenshot shows a web-based application interface for managing genomic compliance. At the top, there are two sections: '3.1 FHIR Extension' and '3.2 SCM Extension'.
3.1 FHIR Extension: This section includes input fields for 'Endpoint URL of the FHIR server containing the resource:' (e.g., http://fhir.example.com/baseDstu3) and 'The FHIR version used:' (e.g., 3). Below these are buttons for 'New', 'Edit', 'Delete', and 'Copy'. A table view shows columns for 'id' and 'resource', with a note 'No data available in table'. Navigation buttons 'Previous' and 'Next' are at the bottom.
3.2 SCM Extension: This section includes input fields for '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), and 'Revision ID within the scm repository:' (e.g., https://github.com/example/repo/blob/fsrc/src/workflow.cwl). Navigation buttons 'Previous' and 'Next' are at the bottom.

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

2.3.3 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

Step 1 Step 2 Step 3 Step 4 Step 5 Step 6

Import Provenance Domain
Usability Domain
Extension Domain Execution Domain
Parametric Domain Description Domain
I/O Domain
Error Domain Review & Export

4. Execution Domain

Script
https://api.sbggenomics.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

New Edit Delete Copy

Show 10 entries Search: []

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

Showing 1 to 1 of 1 entries Previous 1 Next

New Edit Delete Copy

Show 10 entries Search: []

Figure 10: Execution domain example from Step 3.

New

Name

Version

URI

Access Time

SHA1 Checkksum

Cancel Save

Edit

Name

Version

URI

Access Time

SHA1 Checkksum

Cancel Save

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.

2.3.4 Step 4 - Description Domain

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

step_number	name	description	version
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	#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.

2.3.5 Step 5 - I/O and Error Domain

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

Step 5/6 - IO / Error

Step 1 Step 2 Step 3 Step 4 Step 5 Step 6

Import Provenance Domain
Usability Domain
Extension Domain Execution Domain
Parametric Domain Description Domain I/O Domain
Error Domain Review & Export

7. Input/Output Domain

Input subdomain

Show 10 entries Search:

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

Showing 1 to 4 of 4 entries Previous 1 Next

Output subdomain

Show 10 entries Search:

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

Figure 13: I/O Domain example from Step 5

The screenshot shows the Genomics Compliance Suite interface. At the top, there is a table listing two entries:

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

Below this, a message says "Showing 1 to 9 of 9 entries". There are "Previous" and "Next" buttons, with the number "1" highlighted.

The main content area is titled "8. Error Domain". It contains two sections: "Empirical error subdomain" and "Algorithmic error subdomain".

Empirical error subdomain:

New	Edit	Delete	Copy
Show 10 entries			
Search: <input type="text"/>			
key	value	key	value
No data available in table			

Below this, a message says "Showing 0 to 0 of 0 entries". There are "Previous" and "Next" buttons.

Algorithmic error subdomain:

New	Edit	Delete	Copy
Show 10 entries			
Search: <input type="text"/>			
key	value	key	value
No data available in table			

Below this, a message says "Showing 0 to 0 of 0 entries". There are "Previous" and "Next" buttons.

At the bottom, there are "Previous" and "Next" buttons.

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.

2.3.6 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.

Step 6/6 - Review and Export

Step 1 Step 2 Step 3 Step 4 Step 5 Step 6

Import Provenance Domain
Usability Domain
Extension Domain Execution Domain
Parametric Domain Description Domain I/O Domain
Error Domain Review & Export

Top Level Fields

BCO ID

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

Review & Export

```

4   "checksum": "1a1fe65c543efa87d18a3ddf1a2f89d1f5f0c40051b79daab1bf57409f937749",
5   "provenance_domain": {
6     "name": "RNA-seq Alignment - STAR",
7     "version": "1.0.0",
8     "review": [],
9     "derived_from": "https://api.sbggenomics.com/v2/apps/soners-demo-project/rna-seq-alignment-star/0/raw/",
10    "obsolete_after": "2019-08-08T00:00:00+0000",
11    "embargo": ["2019-08-08T00:00:00+0000", "2019-08-08T00:00:00+0000"],
12    "created": "2019-08-08T00:00:00+0000",
13    "modified": "2019-08-08T00:00:00+0000",
14    "contributors": [],
15    "license": "https://spdx.org/licenses/CC-BY-4.0.html"
16  },
17  "usability_domain": "Alignment to a reference genome and transcriptome presents the first step of RNA-Seq analysis. This pipeline uses STAR, an ultrafast aligner for short sequence reads. It can align short reads (10-100 bp) from various sequencing platforms (Illumina, Ion, SOLiD, PacBio, etc.) and can handle paired-end and single-end reads. STAR aligns reads to a reference genome using a linear reference model, which allows it to align reads across the entire genome without the need for tiling the genome into small pieces. STAR is designed to be fast, accurate, and memory efficient, making it suitable for large-scale genomic analysis. STAR can also handle complex genomic features such as repeats and alternative splicing. The alignment results can be used for downstream analysis such as differential gene expression, variant calling, and genome-wide association studies (GWAS).",
18  "extension_domain": {
19    "fhir_extension": {
20      "fhir_endpoint": "",
21      "fhir_version": "",
22      "fhir_resources": []
23    }

```

Generate & Preview BCO

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

Export as JSON Export as PDF

Save to Platform or GitHub Project

Push to GitHub Upload to Platform

Previous Next

Figure 16: Export and save features

```
{ } rna-seq-alignment-star.bco.json ×  
1  {  
2      "bco_spec_version": "https://w3id.org/biocompute/1.3.0/",  
3      "bco_id": "http://biocompute.sbggenomics.com/bco/82fd0416-5d36-4abb-8fef-df836bc79d84",  
4      "checksum": "1a1fe65c543efa87d18a3ddf1a2f89d1f5f0c40051b79daab1bf57409f937749",  
5      "provenance_domain": {  
6          "name": "RNA-seq Alignment - STAR",  
7          "version": "1.0.0",  
8          "review": [],  
9          "derived_from": "https://api.sbggenomics.com/v2/apps/soner/soners-demo-project/rna-seq-alignmen",  
10         "obsolete_after": "2019-08-08T00:00:00+0000",  
11         "embargo": ["2019-08-08T00:00:00+0000", "2019-08-08T00:00:00+0000"],  
12         "created": "2019-08-08T00:00:00+0000",  
13         "modified": "2019-08-08T00:00:00+0000",  
14         "contributors": [],  
15         "license": "https://spdx.org/licenses/CC-BY-4.0.html"  
16     },  
17     "usability_domain": "Alignment to a reference genome and transcriptome presents the first step o",  
18     "extension_domain": {  
19         "fhir_extension": {  
20             "fhir_endpoint": "",  
21             "fhir_version": "",  
22             "fhir_resources": {}  
23         },  
24         "scm_extension": {  
25             "scm_repository": "",  
26             "scm_type": "git",  
27             "scm_commit": "",  
28             "scm_path": "",  
29             "scm_preview": ""  
30         }  
31     },  
32     "description_domain": {  
33         "keywords": [],  
34         "xref": [],  
35         "platform": "Seven Bridges Platform",  
36         "pipeline_steps": [  
37             {  
38                 "step_number": "1"  
39             }  
40         ]  
41     }  
42 }
```

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).

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 (to the platform) appears near the “**Upload Succeeded**” warning to access the generated file on the platform easily (Figure 20).

Figure 19: Generated BCO exported to the Seven Bridges Platform.

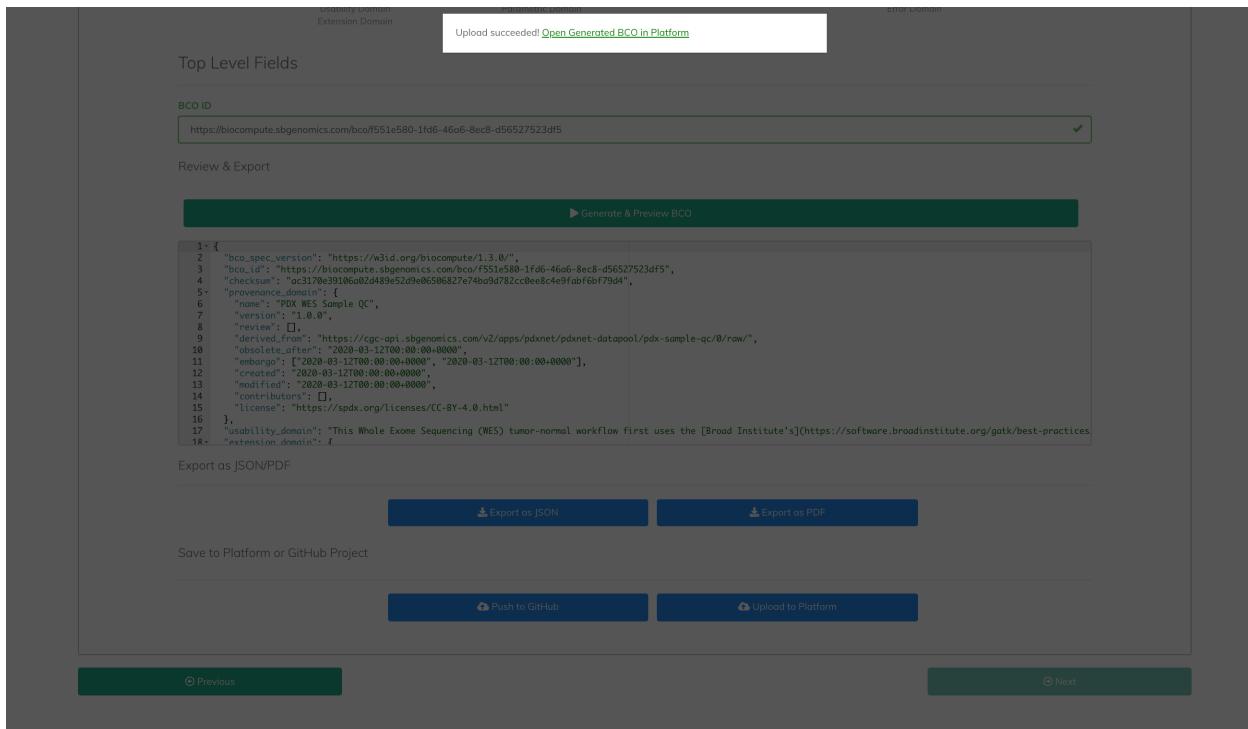


Figure 20: Save to the Seven Bridges Platform with direct links to the platform in Step 6.

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 you 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).

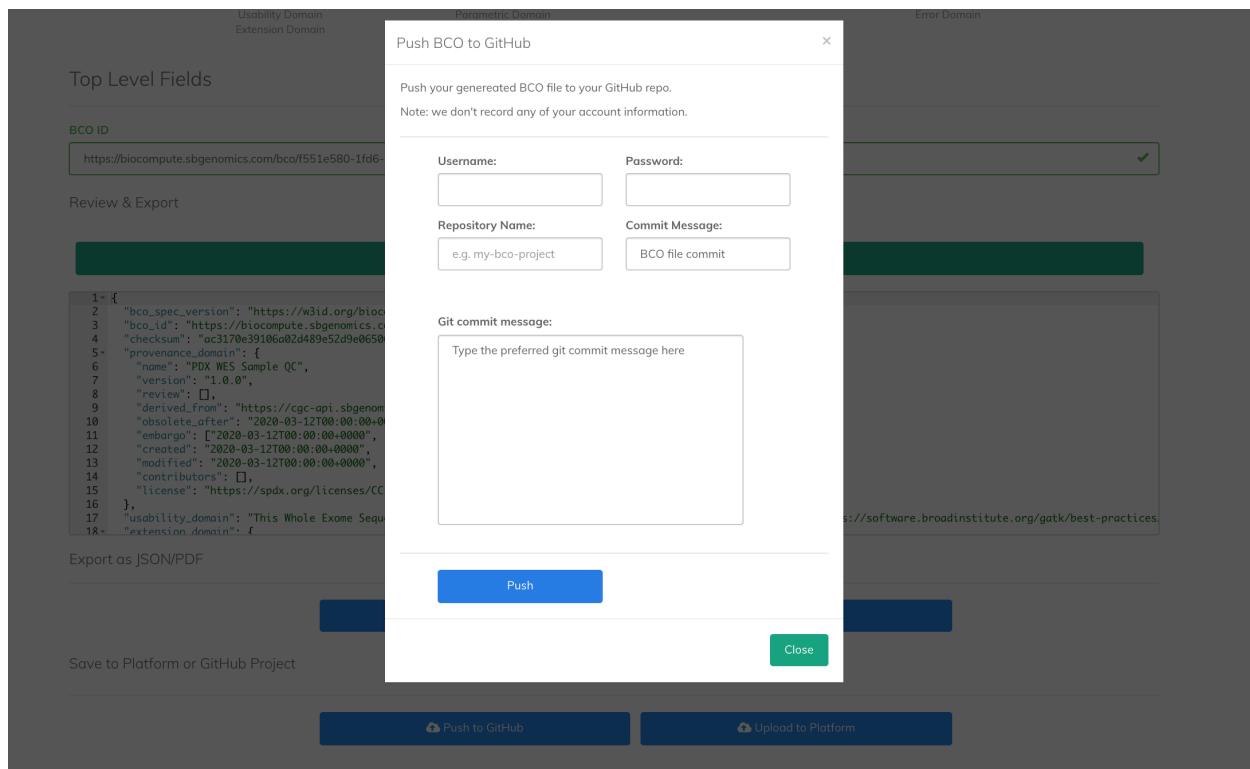


Figure 21: GitHub connection window in Step 6

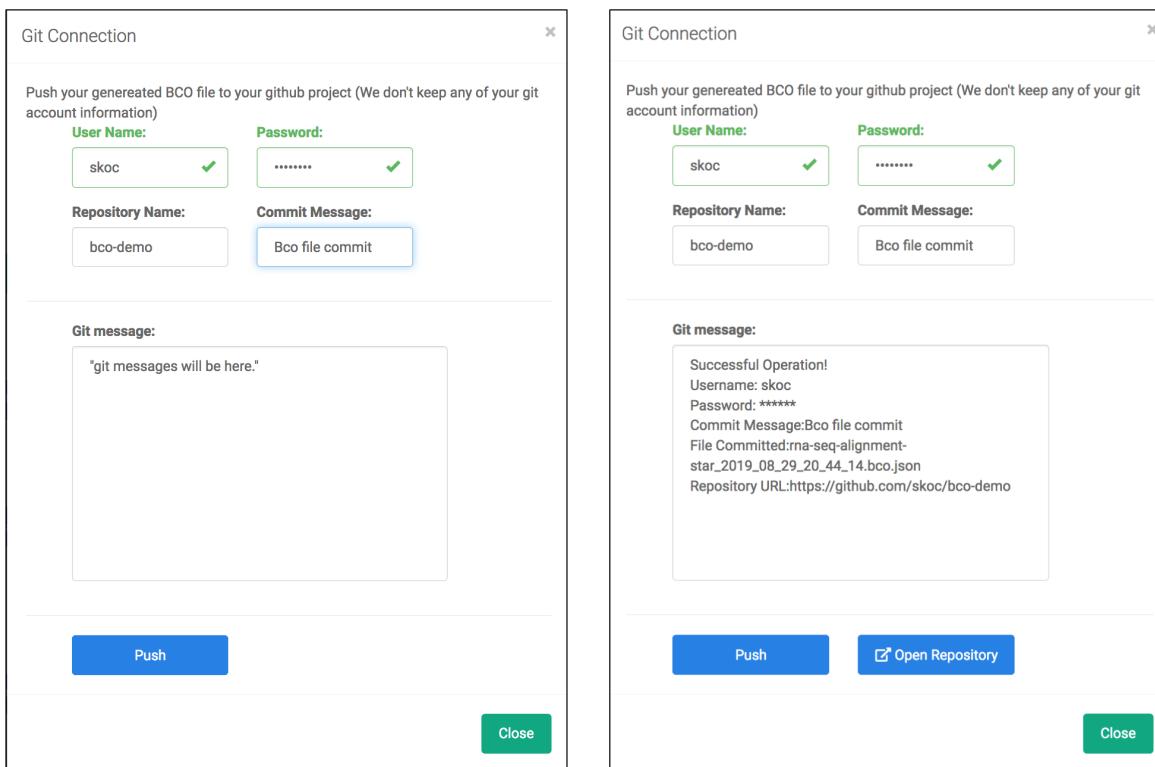


Figure 22: GitHub connection before/after push operation in Step 6

2.4 BioCompute Object (BCO) Validator

The screenshot shows the BCO Validator interface. At the top, there is a navigation bar with links for Home, Generators, Utilities, and Help. Below the navigation bar, the title "BioCompute Object Validator" is displayed. A file input field contains the path "whole-genome-sequencing-bwa-gatk-4.0.bco.json". To the right of the input field is a green button labeled "Validate BCO". Below the file input, there is a section titled "Preview" containing the JSON code of the BCO object. The code includes details such as the BCO specification version, ID, checksum, provenance, version, review status, derivation history, embargo period, creation and modification dates, contributors, license, and usability domain. The "Checksum Validation Results" section shows that the calculated checksum matches the documented one. The "Schema Validation Results" section indicates that the validation was successful with no errors found.

```
1 {  
2   "bco_spec_version": "https://w3id.org/biocompute/1.3.0/",  
3   "bco_id": "http://biocompute.sbggenomics.com/bco/4439045d-6b60-4965-ac15-91cfe69c8d15",  
4   "checksum": "aa4d806c9aa8d52ee17213d611ce56e2b27b70239fa7587bc805854be77d5e14",  
5   "provenance": [{}],  
6   "version": "WGS-BWA-GATK4",  
7   "version": "1.0.",  
8   "review": [],  
9   "derived_from": "",  
10  "embargo": "2020-10-16T00:00:00+0000",  
11  "embargo": ["2019-10-16T00:00:00+0000", "2019-10-31T00:00:00+0000"],  
12  "created": "2019-10-16T00:00:00+0000",  
13  "modified": "2019-10-16T00:00:00+0000",  
14  "contributors": [],  
15  "license": "https://spdx.org/licenses/CC-BY-4.0.html",  
16 },  
17   "usability_domain": "This Whole Genome Sequencing (WGS) workflow identifies variants from a human whole-genome resequencing experiment by using the Broad Institute's best-practices workflow for alignment and vc  
18 }  
  "extension_domain": {}  
  
```

Checksum Validation Results

—— Loading BioCompute Object ——
—— Validating Checksum ——
Documented checksum: aa4d806c9aa8d52ee17213d611ce56e2b27b70239fa7587bc805854be77d5e14
Calculated checksum: aa4d806c9aa8d52ee17213d611ce56e2b27b70239fa7587bc805854be77d5e14
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

Figure 23: A screenshot of the BCO validator UI.

2.5 Interactive BCO Browser

The screenshot shows the SevenBridges Genomics Compliance Suite interface. At the top, there is a dark header bar with the SevenBridges logo and navigation links for Home, Generators, Utilities, and Help. Below the header, the main content area has a title "Interactive Browser for BioCompute Object". A file input field is present with the placeholder "Upload the BCO file (*.json)". A blue button labeled "Explore BCO" is located to the right of the input field. The central part of the screen displays a JSON object representing a BioCompute Object (BCO). The JSON structure includes fields such as "root", "bcf", "spec_version", "id", "checksum", "provenance_domain", "obsolete_after", "embargo", and "created". The "root" field contains nested objects for each of these categories. At the bottom of the page, there is a footer with copyright information: "© 2020 Seven Bridges · Privacy Copyright Terms Contact".

```
{"root": { "bcf": { "spec_version": "https://w3id.org/biocompute/1.3.0/" }, "id": "http://biocompute.sgenomics.com/bco/4439045b-6b60-4965-ac15-91cfef69c8d15", "checksum": "aadd0e6c9a8a52ee17213d61ice56e2b7b70239fe7587bc805854be77d5e14" }, "provenance_domain": { "name": "WGS-BWA-GATK4", "version": "1.0.", "review": [ ], "derived_from": "", "obsolete_after": "2020-10-16T00:00:00+0000" }, "embargo": [ { "date": "2019-10-16T00:00:00+0000" }, { "date": "2019-10-31T00:00:00+0000" } ], "created": "2019-10-16T00:00:00+0000" }}
```

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

Appendix I : Platform Authentication Guide

This section describes how to authenticate and access your Seven Bridges platform resources via the API token.

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).

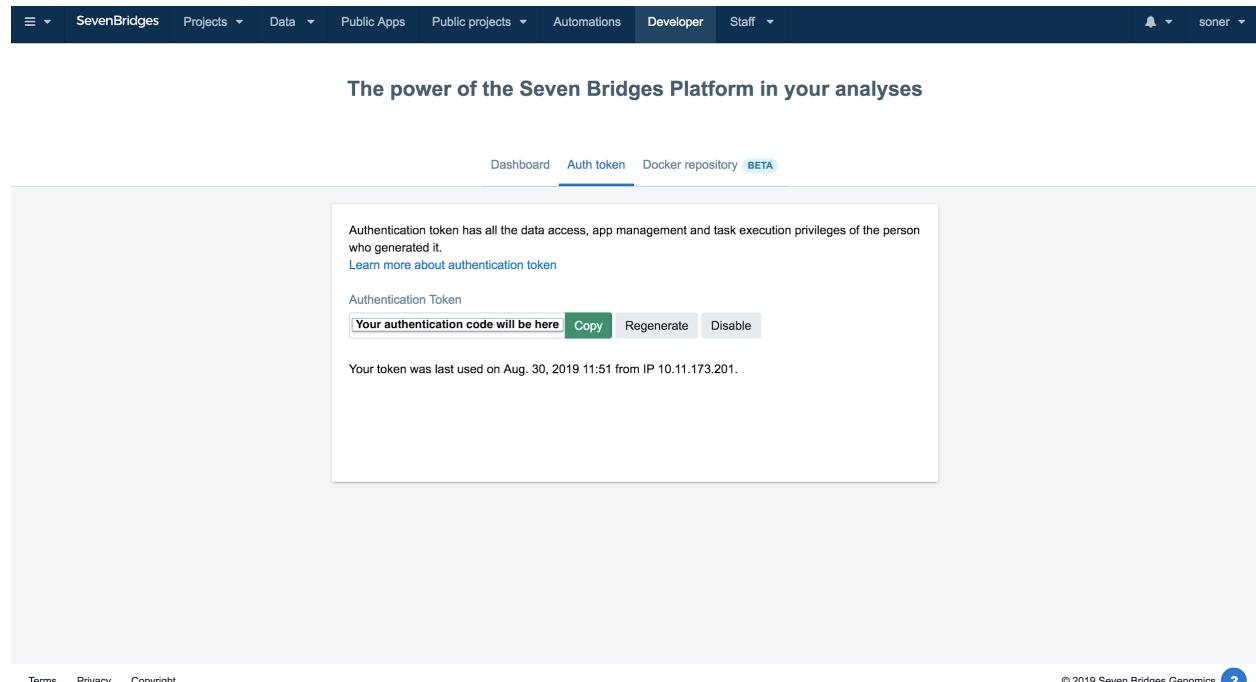


Figure 25: Authentication token generation page on the Seven Bridges Platform.

Appendix II: BioCompute Object Domains

This section summarizes the domains defined in the BioCompute Object specification.

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 Identifier	BCO_id	Unique identifier that should be applied 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 validation.

Figure 26: Top level fields of a BioCompute Object

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 recorded.
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.

Figure 27: Provenance domain of a BioCompute Object

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 searchability and provide a specific description of the object. It helps determine when and how the BCO can be used.

Figure 28: Usability domain of a BioCompute Object

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 pipelines, workflows, protocols, and tool or software source code can be stored, deposited, downloaded.

Figure 29: Extension domain of a BioCompute Object

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).

Figure 30: Description domain of a BioCompute Object

BCO Execution Domain		
	ID	Description
Execution Domain		
Execution Domain	execution_domain	The fields 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 environment.
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.
Environmental Variables	environment_variables	This is an array of key-value pairs useful to configure the execution environment on the target platform.

Figure 31: Execution domain of a BioCompute Object

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.

Figure 32: Parametric domain of a BioCompute Object

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 .

Figure 33: I/O domain of a BioCompute Object

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 outcome. 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 outcome.

Figure 34: Error domain of a BioCompute Object