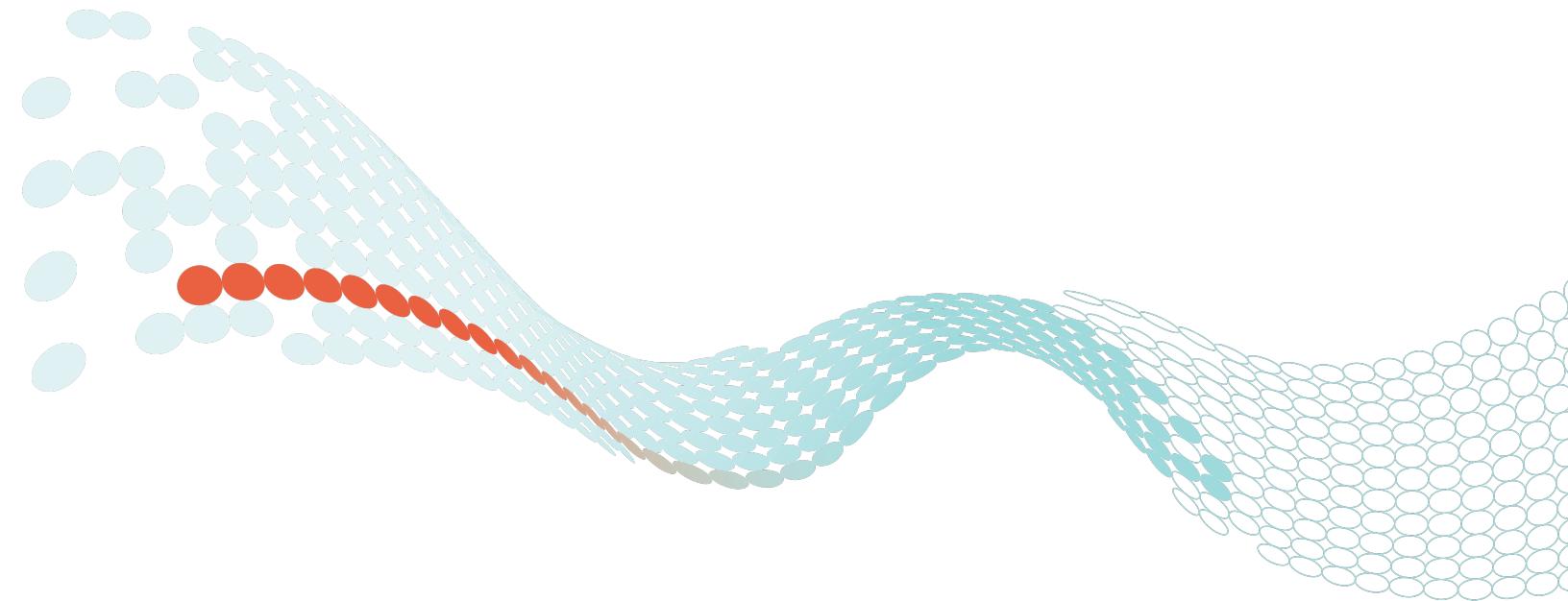


BioCompute Object App:

BCO App User Manual

March 2020



Contents

1	Introduction	1
2	BioCompute Objects	1
2.1	BCO App Overview	1
2.2	Menu Options	2
2.3	Installing and Accessing the BCO App	2
2.3.1	Containerized app	2
2.3.2	Cancer Genomics Cloud hosted app	3
2.4	Getting Started with the BCO App	3
2.4.1	Login Page (Deployment Specific)	3
2.4.2	Landing Page	4
2.5	Generating BioCompute Objects with BCO App	4
2.5.1	Platform Composer	5
2.5.2	Step 1 - Import the Target CWL App	5
2.5.3	Step 2 - Provenance, Usability, and Extension Domains	6
2.5.4	Step 3 - Execution and Parametric Domains	8
2.5.5	Step 4 - Description Domain	9
2.5.6	Step 5 - I/O and Error Domain	10
2.5.7	Step 6 - Review and Export	12
2.6	BioCompute Object Utilities	18
2.6.1	BioCompute Object (BCO) Validator	18
2.6.2	Interactive BCO Browser	19
Appendix I : Platform Authentication Guide		20
Authentication Token		20
Locate Authentication Token		20
Appendix II: BioCompute Object Domains		21

1 Introduction

The BCO App provides tools for generating, reviewing, and validating BioCompute Objects. The BCO App can generate BioCompute Objects from user text input, workflows written in Common Workflow Language (CWL), and from CWL and task information generated on the Cancer Genomics Cloud. The BCO App also include tools for reviewing and validating BioCompute Objects stored locally or on the cloud. Our goals in providing the BCO App are to encourage further development of the standard, promoting the use of CWL to document complex bioinformatics workflows, and to facilitate the use of BCOs in operational settings; including developing the application to support FDA regulatory submissions.

This user manual provides a brief introduction into BioCompute Objects, instructions for installing the BCO App, and an overview of BCO App features.

In reviewing the user manual you will learn how to:

- Generate BCOs from text or CWL files
- Generate BCOs from CWL workflows or Task information from your Seven Bridges Platform project
- Download generated BCO JSON file to your local machine
- Generate PDF reports from BCO JSON files
- Upload BCO file to any of the Seven Bridges platforms
- Publish BCOs to your GitHub account.

2 BioCompute Objects

The BioCompute Object standard is an emerging standard that aims to facilitate the communication of Next Generation Sequencing (NGS) analyses by providing a structured way for documenting and communicating complex analyses as commonly seen in bioinformatics and precision medicine applications. The BCO standard defines 33 fields across nine domains (ex. identification and execution domains). The BCO JSON file is designed to be both human and machine readable. The BCO App supports BioCompute specification v1.3.0. See the appendix or the BioCompute Object website (<https://biocomputeobject.org/index.html>) for additional details.

2.1 BCO App Overview

The current version of the BCO App extends our initial prototype developed for the precisionFDA BioCompute Object Challenge. The challenge required the submitted tools to support the display, creation, and validation. Our current tool also includes support for selecting inputs from both your local machine and from files stored on the Cancer Genomics Cloud, as well as selecting inputs from the suite of Seven Bridges Platforms. The BCO App supports interactive as well as semi-automated

BCO construction. For example, users have the options to select a CWL workflows on the CGC as a way to pre-populate workflows related BCO fields.

2.2 Menu Options

The following menu options are available from the BCO App

Menu	
Entry	Description
Home	App landing page with links to generators and help screen
Generators	Tools for generating BCO JSON Files, generating a BCO PDF, and publishing BCO to a GIT page
Text	Interface allows users to interactively enter text into BCO fields. The composer steps the user through completing each of the BCO domains.
Composer	Interface allows a user to select a CWL file from their local machine to pre-populate the BCO fields with workflow related fields. Once loaded, the interface is identical to the Text Composer.
CWL	Interface allows a user to select a CWL Workflow or Task on the Cancer Genomics Cloud to pre-populate the BCO fields with workflow/task related fields. Once loaded, the interface is identical to the Text Composer
Platform	Interface allows a user to select a CWL Workflow or Task on the Cancer Genomics Cloud to pre-populate the BCO fields with workflow/task related fields. Once loaded, the interface is identical to the Text Composer
Composer	Interface allows a user to select a CWL Workflow or Task on the Cancer Genomics Cloud to pre-populate the BCO fields with workflow/task related fields. Once loaded, the interface is identical to the Text Composer
Utilities	Tools that support BCO use.
Browser	The browser facilitates interactive review of a BCO JSON file
Validator	The validator allows a user to compare a BCO JSON file to the BCO specification.
Standard	The Standard feature provides an interface to review the BCO field definitions by domain.
Help	Brief description of Menu items

2.3 Installing and Accessing the BCO App

The BCO App can be deployed locally and on a remote server. Both the source code and a containerized (Docker) version is available on GitHub. The source code can also be executed from the Cancer Genomics Cloud, as well as any of the Seven Bridges Platforms, through the R Studio extension on our platforms. Additional BCO App deployment details follow.

2.3.1 Containerized app

Running the BCO App from a containerized app will allow you to execute the app without having to install the BCO App requirements/dependencies on your local machine or server. Instructions for pulling, building, and running the Docker image can be found on the BCO App GitHub page (<https://github.com/sbg/bco-app>). The default username and password sevenbridges/sevenbridges (Figure 1).

2.3.2 Cancer Genomics Cloud hosted app

The BCO App can be executed from within the Cancer Genomics Cloud through R Studio. Execution instructions follow.

Go to the hosted version and log in with your username and password.

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 (support.at.sbggenomics.com).

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.

2.4 Getting Started with the BCO App

Executing the BCO App will result in an optional login page or the BCO App landing page depending on the BCO App settings.

2.4.1 Login Page (Deployment Specific)

The login credentials are installation dependent. The default login credentials for user and password are respectively sevenbridges and sevenbridges.

2.4.2 Landing Page

You will see the landing page of the BCO App after you login (See Figure 1).

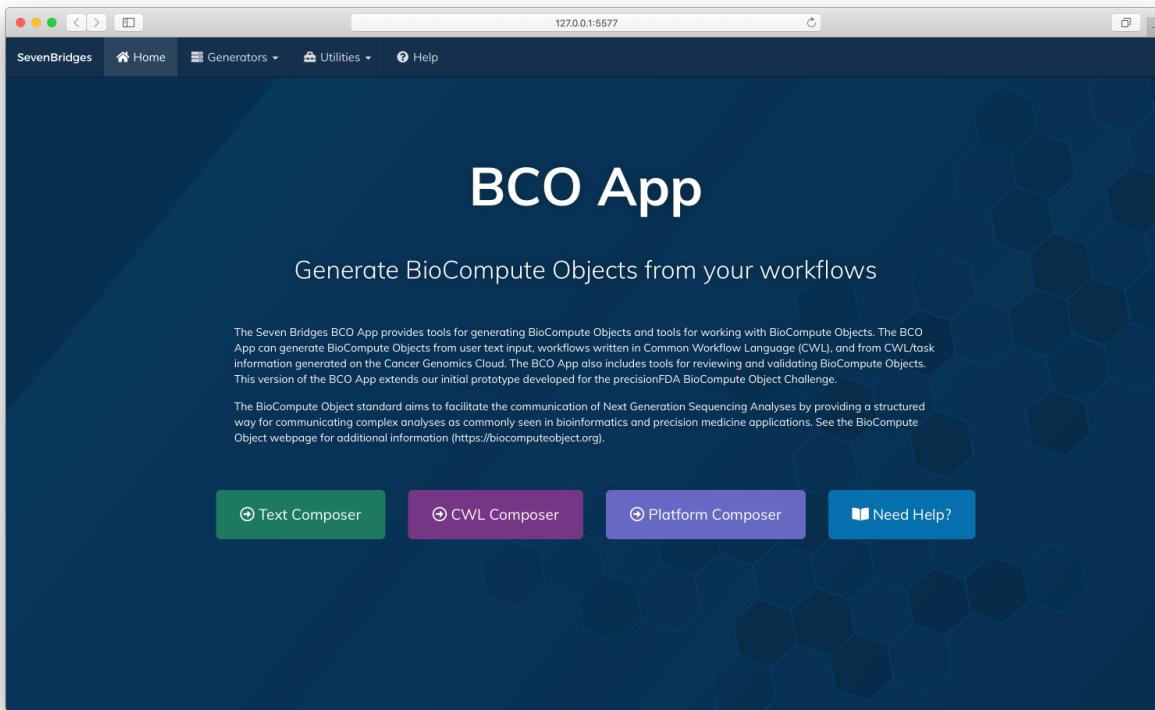


Figure 1: Landing Page of the BCO App

The navigation bar along the top of the screen provides access to primary BCO App features including Generators and Utilities. Buttons along the bottom of the screen provide single click access to BCO App generators and to the BCO App help screen.

2.5 Generating BioCompute Objects with BCO App

The following example details generating a BCO from a CWL workflow on the Cancer Genomics Cloud. We selected the example since the Platform Composer includes elements of the text composer and the CWL composer. Reading through the example will outline the steps required to generate a BCO object.

2.5.1 Platform Composer

2.5.2 Step 1 - Import the Target CWL App

After you click the “**Platform Composer**” page of the BCO App under “**Generators**” tab, you will see the initial step of the BCO generation, import step (Figure 2).

The screenshot shows the "Step 1/6 - Import" page of the Platform Composer. The top navigation bar includes links for Home, Generators, Utilities, and Help. The main content area has a title "Import from Seven Bridges Platforms" and a sub-section "Import". It contains several input fields: "Platform" dropdown set to "Cancer Genomics Cloud (CGC)", "Auth Token" input field with a link to "Paste the Auth Token from the CGC Developer Dashboard", "Choose Project" dropdown, "Choose Task in Project (Optional - If Exist)" dropdown, and "Choose App to Import CWL Workflow" dropdown. At the bottom are "Previous" and "Next" buttons.

Figure 2: Initial Page of the Platform Composer

The BCO Creator/Composer includes six steps, each step corresponding to a BCO domain, to generate a new BioCompute Object (BCO):

Step 1: Import workflow written Common Workflow Language (CWL).

Step 2: Set BCO fields in the Provenance, Usability, and Extension Domains.

Step 3: Set BCO fields in the Execution and Parametric Domains.

Step 4: Set the Description Domain fields.

Step 5: Set the fields in the I/O and Error Domain.

Step 6: Generate BCO

Once user entry is completed, the user can generate and review the BCO associated with the workflow. A generated BCO can be downloaded, published to a Git page, and uploaded it to your Seven Bridges platform.

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. Select a name for a “Platform” that you want to fetch a CWL file of an application (Figure 3).

The screenshot shows a user interface for importing a CWL workflow from a Seven Bridges Platform. At the top, it says "Import from Seven Bridges Platforms". Below that is a "Platform" dropdown menu set to "Cancer Genomics Cloud (CGC)". There is a text input field for "Paste the Auth Token from the [CGC Developer Dashboard](#)". Below that are two dropdown menus: "Choose Project" and "Choose Task in Project (Optional - If Exist)". At the bottom is another dropdown menu for "Choose App to Import CWL Workflow".

Figure 3: Required Platform Composer

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 (See Appendix I for additional information). Projects, tasks, and applications that exist in the account of the user will be fetched once a valid authentication token is entered. 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. Optionally, a user could selected an completed task at this step which would include the workflow information in the generated CWL.

2.5.3 Step 2 - Provenance, Usability, and Extension Domains

In step 2, User enters Provenance, Usability, and Extension Domains of BCO.

The BCO App automatically enters some fields from information parsed from the selected CWL workflow. Users are prompted to complete the remaining fields. You can find the complete detailed list of Provenance, Usability, and Extension Domain fields in Appendix II. See Figure 4 and 5 for Provenance Domain Entry fields.

Step 2/6 - Provenance / Usability / Extension

The diagram illustrates the workflow across six steps:

- Step 1: Import** (green circle)
- Step 2: Provenance Domain / Usability Domain / Extension Domain** (green circle)
- Step 3: Execution Domain / Parametric Domain** (grey circle)
- Step 4: Description Domain** (grey circle)
- Step 5: I/O Domain / Error Domain** (grey circle)
- Step 6: Review & Export** (grey circle)

Visualization of the Workflow

The central part of the visualization is a complex network graph representing the workflow. Nodes are represented by circles of different colors and sizes, connected by arrows indicating data flow. The nodes include:

- Input nodes (grey): "Import BCO File", "SBOL Document", "BioPAX File", "FASTA File", "FASTQ File", "BED File", "GFF File", "VCF File", "Genomic Coordinates", "Protein Coordinates", "Gene Level Expression Estimates", "Protein Level Expression Estimates", "Protein-DNA Interaction Report", "Protein-DNA Interaction Data", "Q3 Interaction Data", "Q3 Interaction Report", "Q3 Interaction Coordinates", "Q3 Interaction Estimates", "Integrative Q3 Report", "Kappa ROC Curve", "AUC ROC Curve".
- Output nodes (orange): "Provenance Domains", "Parameter Domains", "Execution Domains", "Description Domains", "I/O Domains", "Error Domains".
- Step nodes (teal): "Provenance Domains", "Parameter Domains", "Execution Domains", "Description Domains", "I/O Domains", "Error Domains".

Legend

- input**: Grey circle
- output**: Orange circle
- step**: Teal circle

1. Provenance Domain

Name for the BCO

PDX RNA Expression Estimation Workflow ✓

Version of the BCO instance object

1.0.5 ✓

Inheritance or derivation from

<https://cgc-api.sbggenomics.com/v2/applications/pdxnet/pdxnet-database/rsem-1-2-31-workflow-with-star-aligner-pe/> ✓

License

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

BCO initial creation date

2020-03-12 ✓

Figure 4: Example of Provenance Domain fields automatically filled by the BCO App.

The screenshot shows the BCO App interface with two main sections: FHIR Extension and SCM Extension.

3.1 FHIR Extension:

- Endpoint URL of the FHIR server containing the resource:
e.g. http://fhir.example.com/baseDstu3
- The FHIR version used:
e.g. 3
- FHIR Resources table:
 - Buttons: New, Edit, Delete, Copy
 - Show 10 entries
 - Search: Search field
 - Columns: id, resource
 - No data available in table
 - Showing 0 to 0 of 0 entries
 - Previous, Next buttons

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/fid/src/workflow.cwl

Navigation buttons at the bottom:
Previous, Next

Figure 5: Example of Provenance Domain Fields to be entered by the user.

2.5.4 Step 3 - Execution and Parametric Domains

In step 3, You are prompted to enter text in the Execution and Parametric Domains fields. Some entries will spark a pop up box as shown in Figure 6. See Appendix II for a complete list of fields in the Execution and Parametric Domain.

New	
Name	<input type="text"/>
Version	<input type="text"/>
URI	<input type="text"/>
Access Time	<input type="text" value="2019-08-14"/>
SHA1 Checksum	<input type="text"/>
<button>Cancel</button> <button>Save</button>	

Edit	
Name	<input type="text" value="Seven Bridges Platform"/>
Version	<input type="text" value="2019-08-29"/>
URI	<input type="text" value="https://igor.sbggenomics.com/"/>
Access Time	<input type="text" value="2019-08-14"/>
SHA1 Checksum	<input type="text"/>
<button>Cancel</button> <button>Save</button>	

Figure 6: Example of adding a new entry or editing existence one in the field.

2.5.5 Step 4 - Description Domain

The user is prompted to enter information from the in the Description Domain. Some of the fields will be pre-populated as shown in Figure 7. See Appendix II for a complete list Description Domain fields.

Step 4/6 - Description

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

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

New Edit Delete Copy

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	#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 7: 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.

2.5.6 Step 5 - I/O and Error Domain

The user is prompted to enter information for the I/O and Error Domains. Some of the fields will be pre-populated as shown in Figure 8 and 9. See Appendix II for a complete list fields for the I/O and Error Domains.

Step 5/6 - IO / Error

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

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 8: I/O Domain example from Step 5

The screenshot shows the BCO App interface for Step 5, specifically the "Error Domain" section. 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, the "8. Error Domain" section is displayed. It contains two sub-sections: "Empirical error subdomain" and "Algorithmic error subdomain". Each sub-section has a table with the following columns: key, value, and a search bar. Both tables show "No data available in table".

At the bottom of the page, there are navigation buttons: "Previous" and "Next".

Figure 9: 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.

2.5.7 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 (See figure 10).

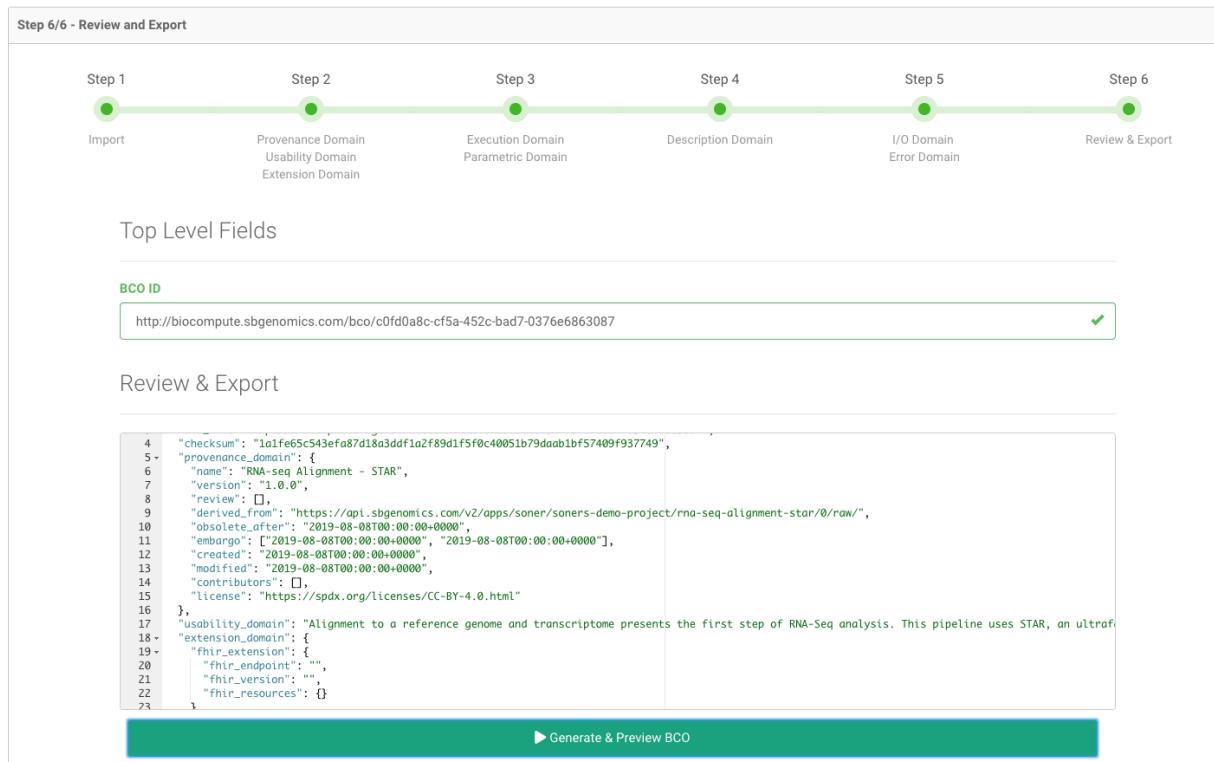


Figure 10: Review the generated BCO file

The BCO App includes several BCO export features (Figure 11). A user can export a BCO file in JSON file format, “**Export as JSON**”. The generated BCO file is downloaded as a JSON file (Figure 12).

Export as JSON/PDF

 Export as JSON

 Export as PDF

Save to Platform or GitHub Project

 Push to GitHub

 Upload to Platform

Figure 11: Export and save features

```
{ } rna-seq-alignment-star.bco.json x
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 12: Example View of the Downloaded BCO JSON file

Users can also generate a PDF reports for the BioCompute Object with “**Export as PDF**” button. The report includes human readable summaries for all the BCO domains, the BioCompute Object specification, and the raw JSON of the generated BCO (Figure 13).

Figure 13: Example View of the Generated PDF Report in Step 6

In addition to export features, there are two very useful save features in the BCO App: **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 14.

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 15).

Figure 14: Generated BCO exported to the Seven Bridges Platform.

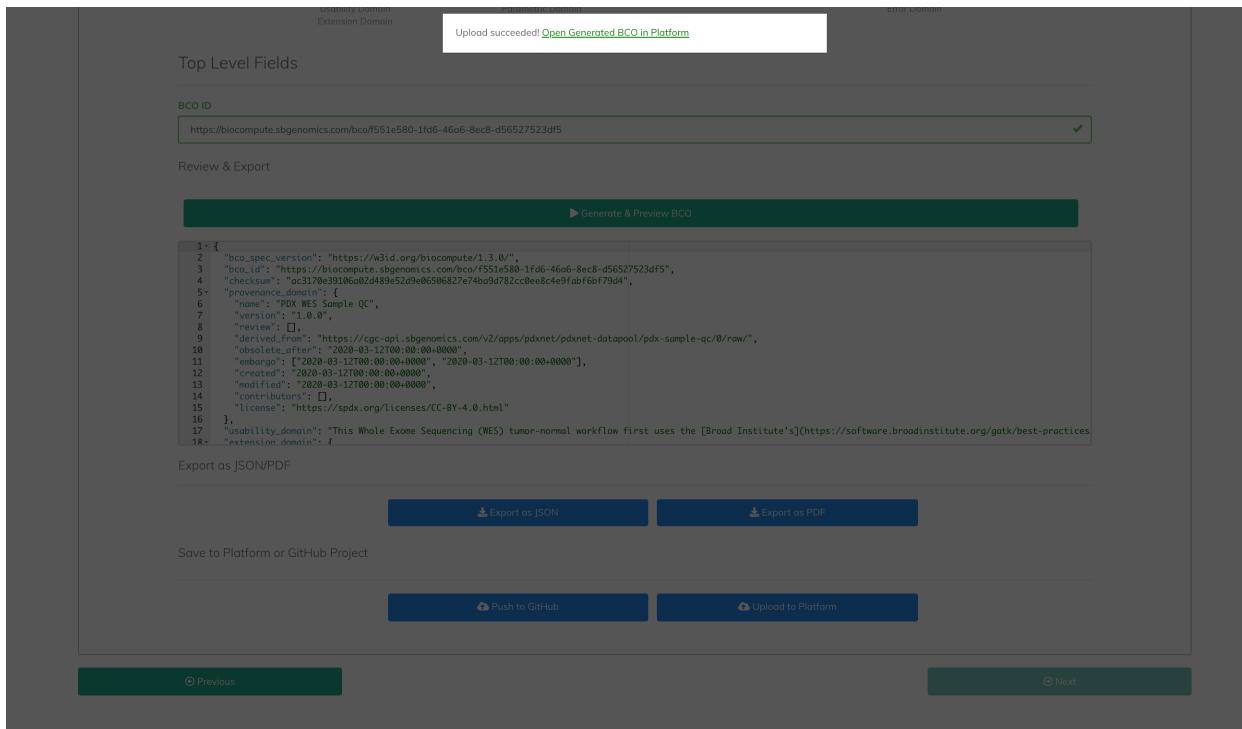


Figure 15: Save to the Seven Bridges Platform with direct links to the platform in Step 6.

A second way to save the generated BCO is to publish the BCO to GitHub. “**Save to GitHub**”, automatically saves the generated BCO file into your GitHub account. “**GitHub Connection**” popup window streamlines the process of publishing the BCO JSON file to your git page (Figure 16).

The “**Push to GitHub**” popup window requires the **username** and **password** of the user’s GitHub account to complete the connection and push operations. The user also enters the target “**Repository Name**” to push the generated BCO file. The “**Git Commit Message**” contains GitHub response messaging that may include indication of a successful or failed operation.

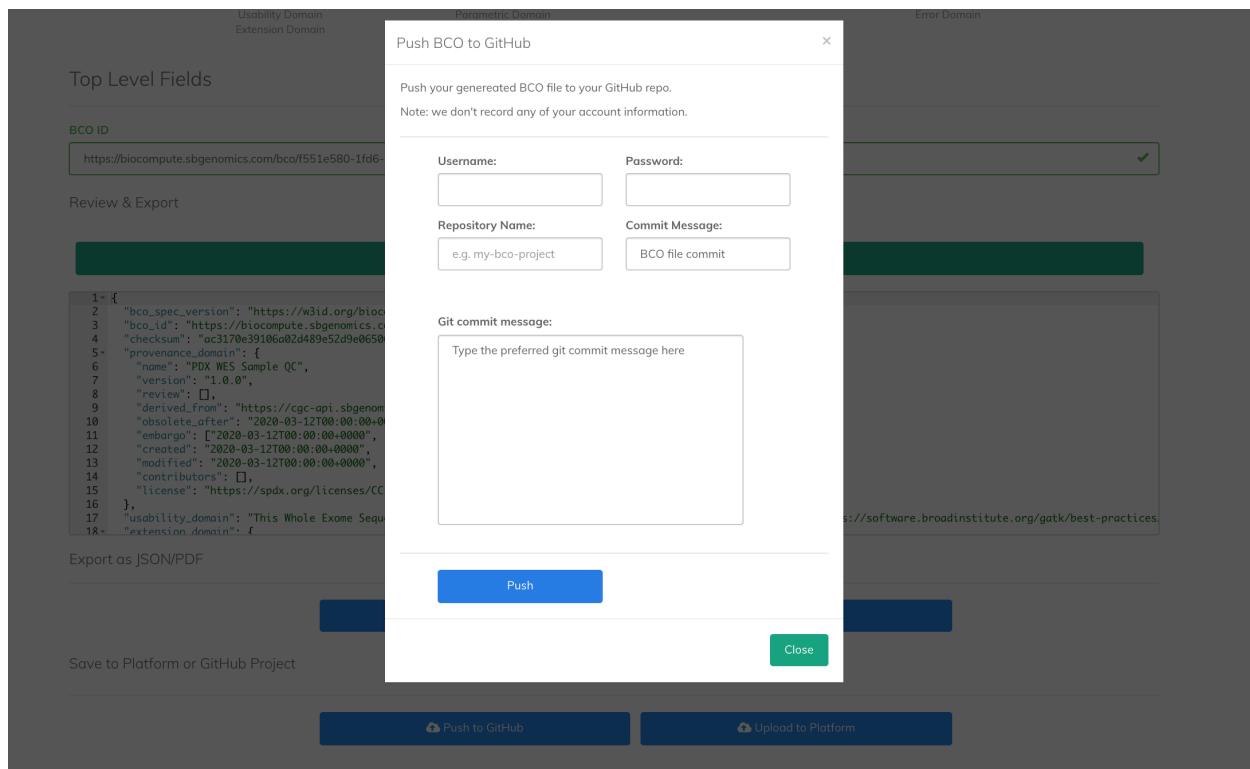


Figure 16: GitHub connection window in Step 6

2.6 BioCompute Object Utilities

2.6.1 BioCompute Object (BCO) Validator

The BCO validator checks an uploaded BCO JSON file with the BCO specification. Results of the check include a Checksum Validation and the Schema Validation.

The screenshot shows the SevenBridges BioCompute Object Validator interface. At the top, there's a navigation bar with links for Home, Generators, Utilities, and Help. Below the navigation is a title "BioCompute Object Validator". A "Upload the BCO file (*.json):" input field contains the path "whole-genome-sequencing-bwa-gatk4-4.bco.json". To the right of the input is a green "Validate BCO" button. Below the input field is a "Preview" section containing the BCO JSON code. The code is as follows:

```
1- {
2   "bc0_spec_version": "https://w3id.org/biocompute/1.3.0/",
3   "bc0_id": "https://biocompute.bnphomics.com/bco/44390459-6be0-4965-ac15-91cfe69c8d15",
4   "checksum": "aa4d0a6c9aa8a52ee17213d611ce56e2b27b70239fa7587bc805854be77d5e14",
5   "provenance_domain": {
6     "name": "WGS-BWA-GATK4",
7     "version": "4.1.0",
8     "review": [],
9     "derived_from": "",
10    "obsolete_after": "2020-10-16T00:00:00+0000",
11    "created": "2019-10-16T00:00:00+0000",
12    "modified": "2019-10-16T00:00:00+0000",
13    "contributors": [],
14    "license": "https://spdx.org/licenses/CC-BY-4.0.html"
15  },
16  "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 variant calling."
17}
18-
```

Below the preview is a "Checksum Validation Results" section with a status message: "Documented checksum: aa4d0a6c9aa8a52ee17213d611ce56e2b27b70239fa7587bc805854be77d5e14 Calculated checksum: aa4d0a6c9aa8a52ee17213d611ce56e2b27b70239fa7587bc805854be77d5e14 Documented and calculated checksum matched." The final section is "Schema Validation Results" which shows a JSON object with one error message: "field 'message' 1 data.extension_domain.fhir_extension is the wrong type".

Figure 17: A screenshot of the BCO validator UI.

2.6.2 Interactive BCO Browser

The Interactive browser allows the user to open and review a BCO file stores on a local machine (See Figure 18).

```
root": { "id": "whole-genome-sequencing-bwa-gatk-4-0.bco.json", "bio_spec_version": "https://w3id.org/biocompute/1.3.0/", "bco_id": "http://biocompute.sigenomics.com/bco/4439045b-6b60-4965-ac15-91cfed9c8d15", "checksum": "aad0a6c9a8a52ee17213d61ce5e2b7b70239fa7587bc805854be77d5e14", "provenance_domain": { "name": "WGS-BWA-GATK4", "version": "1.0.", "review": [ { "id": "1", "derived_from": "", "obsolete_after": "2020-10-16T00:00:00+0000" } ], "embargo": [ { "id": "1", "date": "2019-10-16T00:00:00+0000" }, { "id": "2", "date": "2019-10-17T00:00:00+0000" } ], "created": "2019-10-16T00:00:00+0000" }
```

Figure 18: 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 19).

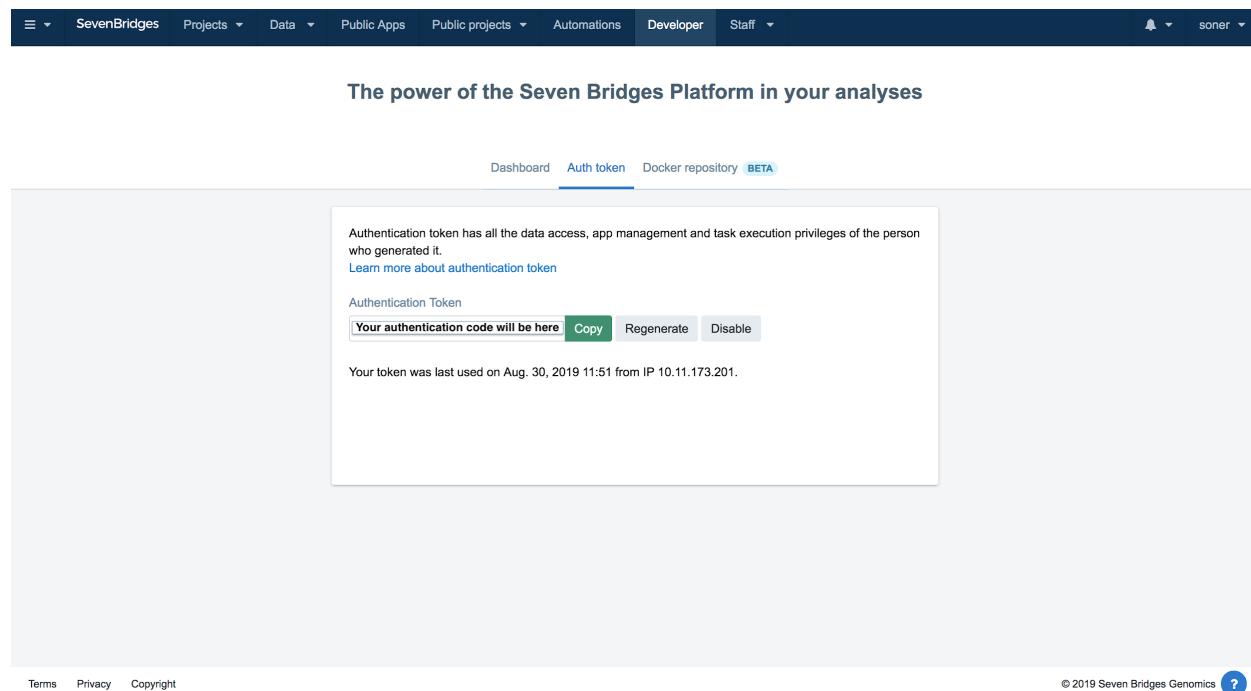


Figure 19: Authentication token generation page on the Seven Bridges Platform.

Appendix II: BioCompute Object Domains

This section summarizes the domains defined in the BioCompute Object specification (v1.3.0).

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 20: 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 21: 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 22: 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 23: 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 24: 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 25: 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 26: 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 27: 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 28: Error domain of a BioCompute Object