BioDepot CRDC Integrator: Genomic and image analyses from the Cancer Research Data Commons using a web-based platform

Varik Hoang[1,2], Ling-Hong Hung[1], Zachary Colburn[2], Ka Yee Yeung[1,2]
[1]School of Engineering and Technology, University of Washington Tacoma
[2]Biodepot LLC, Seattle, WA

ABSTRACT

We present a browser-based standalone program to install a web server that facilitates the download, visualization and analysis of clinical, genomic, and image data from the National Cancer Institute (NCI)'s Cancer Research Data Commons (CRDC). CRDC provides access to data type specific repositories, including the Genomic Data Commons (GDC) and Imaging Data commons (IDC). Each data commons provides a web-based user interface to query and explore data. GDC and IDC also provide APIs to systematically retrieve data from its own data commons. However, integrating data from the same patients across repositories requires substantial technical expertise in these APIs. We present a web server that systematically retrieves data corresponding to the same patients across multiple data commons and presents the result in an accessible and interactive table format. We leveraged IDC's API to retrieve metadata values from all DICOM (Digital Imaging and Communications in Medicine) images and transformed these metadata into a SQLite3 database, which was used to map data from IDC with data from GDC. Our web server also offers visualization of DICOM images using IDC viewers (SliM, OHIF) as well as seamless integration with Fiji for additional image processing.

INTRODUCTION

To enable data sharing across studies and data types, the National Cancer Institute (NCI) launched a cloud-based data science infrastructure called the Cancer Research Data Commons (CRDC) that connects datasets with analytical tools [1]. The CRDC provides access to data-type specific repositories, including the Genomic Data Commons (GDC) [2, 3] and the Imaging Data Commons (IDC) [4, 5]. The GDC is a repository for raw sequencing data and derived results. Experimental strategies in the GDC include RNA sequencing (RNA-Seq), microRNA sequencing (miRNA-Seq), whole genome sequencing (WGS), whole exome sequencing (WXS) and targeted sequencing. As of the v39.0 data release on December 4, 2023, the GDC consists of data from over 44 thousand cases across 79 projects [2]. The IDC is

a repository for images, image annotations and analysis results. As of the v17.0 data release on December 4, 2023, the IDC consists of over 51 TB of data across over 63 thousand cases. Many different types of cancer imaging data are available from the IDC, such as radiology, digital pathology, and multispectral data. The IDC also provides access to data standardized using the Digital Imaging and Communication in Medicine (DICOM) standard.

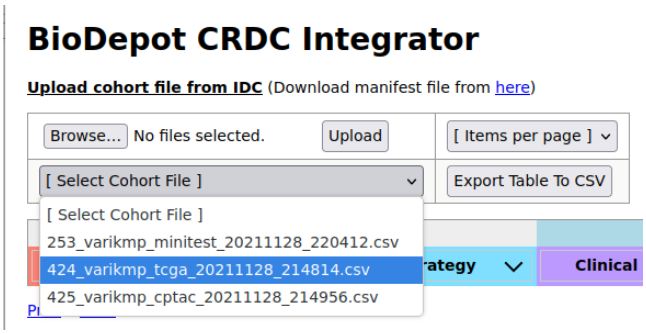
Integrated analysis of genomic and imaging data across different data commons within the CRDC is non-trivial. Each data common provides a web-based user interface to query and explore data. Each data common has their own application programming interfaces (APIs). In particular, GDC and IDC provide APIs to systematically retrieve data from its own data commons. Specifically, the GDC RESTful API can be used to query and search data with a wide range of parameters, download files with tokens from the GDC data portal, slice BAM files to receive a portion of files, and submit data to the GDC [6]. A Bioconductor package "GenomicDataCommons" also provides basic infrastructure in R to query and access genomic data and metadata (including clinical, demographics, annotations) available from the GDC [7, 8]. The IDC is built on the Google Cloud Platform (GCP), including the Google compute engine, BigQuery and the Healthcare API [9]. The IDC provides programmatic interfaces in addition to an interactive data portal, including a RESTful API and SQL interface with example Colab notebooks for developers to query and retrieve IDC data sources. GCP's Storage and BigQuery components can be leveraged for advanced data exploration capabilities [9].

In this paper, we present an open-source software tool that systematically retrieves data corresponding to the same patients across multiple data commons and presents the results in an accessible and interactive table format. We leveraged IDC's API to retrieve metadata values from all DICOM images and transformed these metadata into a database, which was used to map data from IDC with data from GDC. **Figure 1** shows the output of filtering and mapping patient IDs to clinical, metadata, genomic data from the GDC and imaging data from the IDC. Users can choose filtering criteria from primary sites and experimental strategy, select the fields of interest from clinical and metadata to be displayed in the output table and then export the table to a CSV file for download. Not only do the users have access to the IDC data viewers (SliM for slide microscopy images and Open Health Imaging Foundation for radiology images) directly from our application, users can also visualize and leverage QuPath (CITE) or Fiji [10], open source well-established image packages with extensive plugins, to analyze DICOM data. In addition to facilitating data integration across data commons, our web server allows the user

to automatically download DICOM images and analyze these images in Fiji *without any installation* within the provided container. Our web server provides API for users to query the data from the IDC data common without requiring any Google Cloud credential from the clients of our web server.

<add a paragraph about running the Integrator) within Bwb>

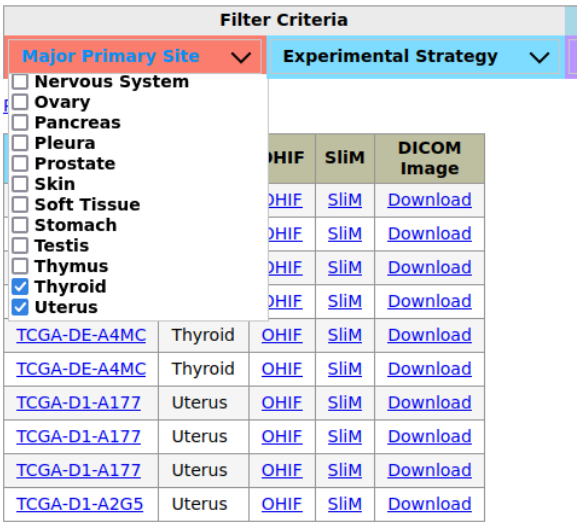
In the example, we use the data set from IDC by downloading the cohort file in CSV format. From the IDC portal web page, we have embraced the enriching TCGA dataset sourced from IDC by obtaining the cohort file in the IDC portal web page. From our web server, we select the uploaded file from the presented dropdown menu, where the file "424_varikmp_tcga_20211128_214814.csv" is uploaded.



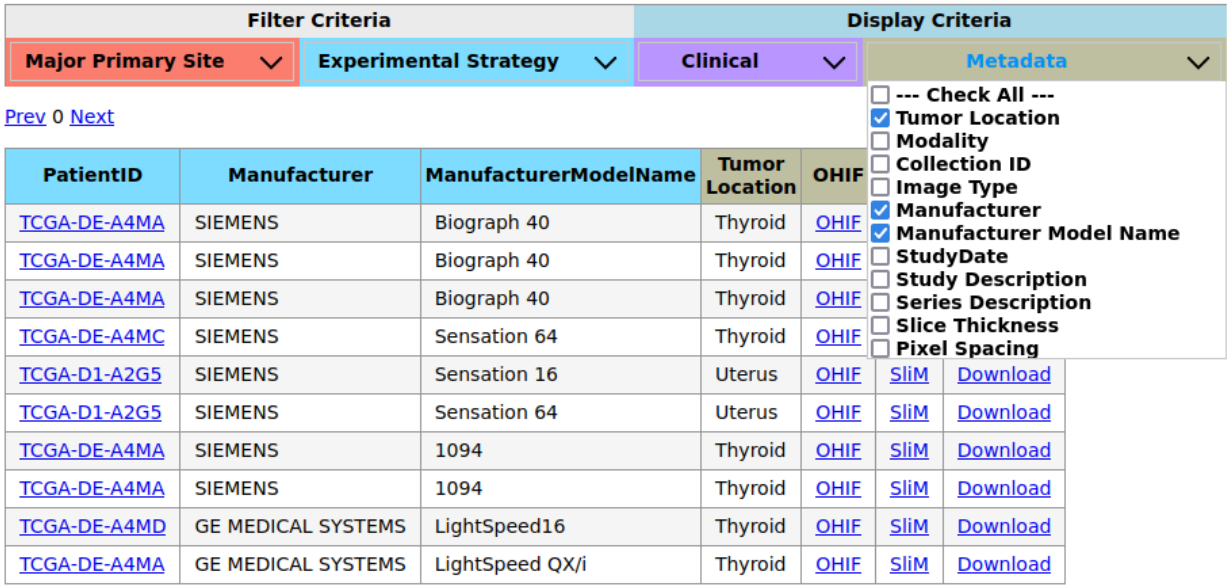
After we choose the cohort file from the list, our application shows us a neat table of results. This table makes it easy to see and understand the data we have.

PatientID	Tumor Location	OHIF	SliM	DICOM Image
TCGA-E2-A1IE	Breast	OHIF	SliM	Download
TCGA-E2-A109	Breast	OHIF	SliM	Download
TCGA-E2-A1IE	Breast	OHIF	SliM	Download
TCGA-E2-A109	Breast	OHIF	SliM	Download
TCGA-E2-A109	Breast	OHIF	SliM	Download
TCGA-E2-A109	Breast	OHIF	SliM	Download
TCGA-E2-A109	Breast	OHIF	SliM	Download
TCGA-E2-A1LG	Breast	OHIF	SliM	Download
TCGA-E2-A14N	Breast	OHIF	SliM	Download
TCGA-E2-A1LG	Breast	OHIF	SliM	Download

There's a section called "Filter Criteria" which helped us refine our results. We ticked the boxes for "Thyroid" and "Uterus" to see only the results that matched these tumor locations. This made the information we got even more useful.



Then, there's another section called "Display Criteria". We check boxes labeled "Manufacturer" and "Manufacturer Model Name". This step gives us extra info about the records we had in the results table. It's like getting a closer look at the details behind the data.



Our application and the GDC/IDC data to make things easy for us. We pick data, put it on our server, and got a clear table of results. We narrow down the results to what we need and learn more about the records in a simple way.

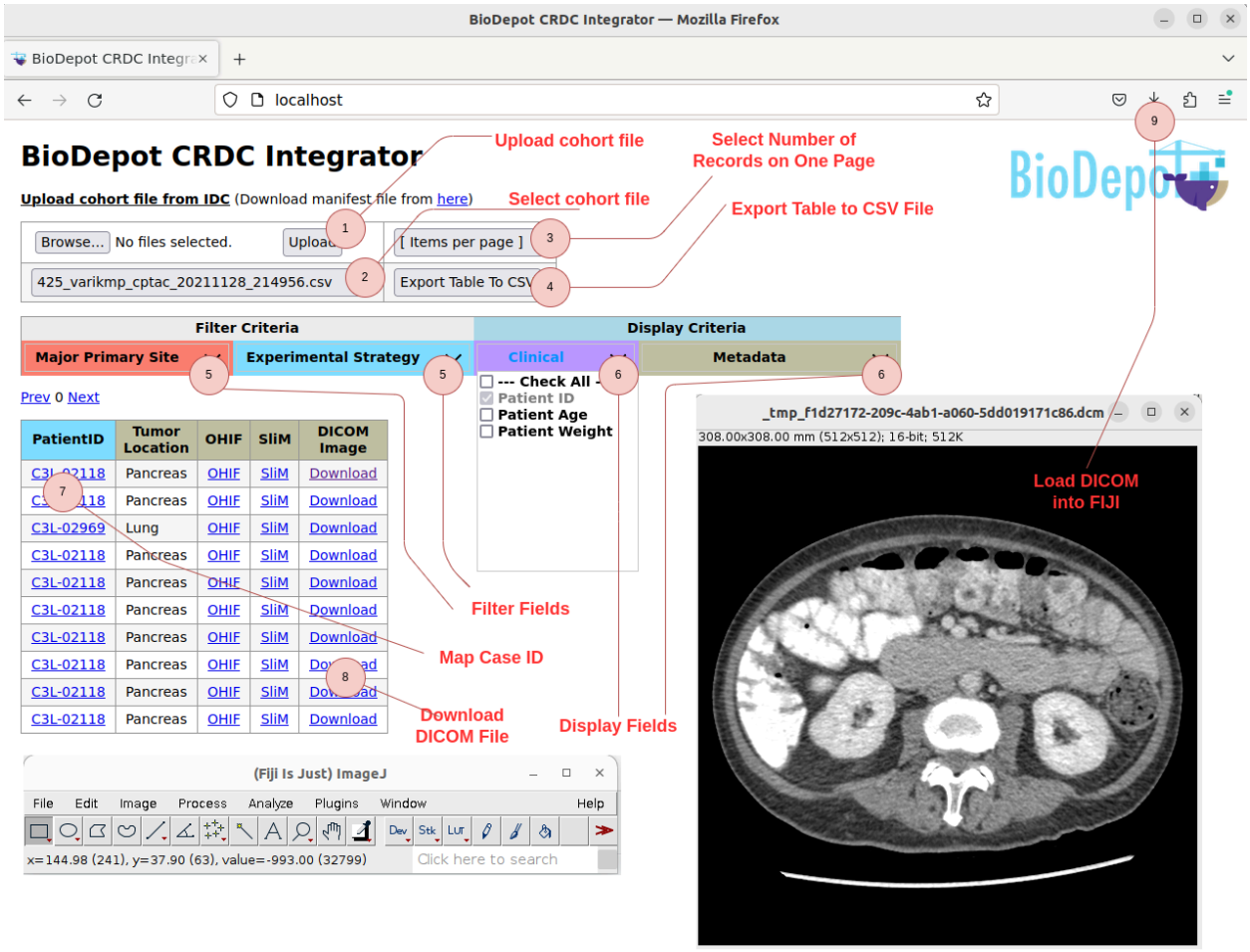


Figure 1: The Biodepot CRDC Integrator is a containerized software tool that systematically filters and retrieves data corresponding to the same patients across the Genomic Data Commons (GDC) and Imaging Data Commons (IDC), presents the result in an accessible and interactive table format, allowing the user to visualize the downloaded images using the IDC data viewers (SliM, OHIF) as well as to perform additional analyses using QuPath or Fiji. Annotated interactive steps for the user include:

1. Select and upload the cohort file (CSV format) downloaded from the IDC.
2. Select the IDC cohort file from the drop-down menu.
3. Select the number of records displayed in the table of results.
4. Export the table of results as a CSV file via browser download.
5. Select filtering criteria from primary sites and experimental strategy.
6. Select fields to display from clinical and metadata in the table of results.
7. Open a new page from the IDC to view the meta data of a selected patient from the table of results.

8. Download the selected DICOM image from the table of results to the local machine.
9. Analyze the selected DICOM image using the QuPath or Fiji image processing package integrated in the Firefox browser provided by our distributed container.

There is a stand-alone service that allows users to select the software they prefer to run the DICOM image. Both Fiji (default selection) and QuPath are open-source and have active communities that contribute to their development and improvement.

RESULTS

The Cancer Imaging Archive (TCIA) dataset is favored for bioinformatics analysis based on the numerous advantages [Link]. TCIA provides a vast and diverse collection of medical images from various cancer types and imaging modalities, making it adaptable to different research needs [Link] [Link]. The data is meticulously curated including essential clinical information and annotations like tumor segmentation, facilitating in-depth analysis [Link]. The format of TCIA is standardized and it has open-access nature to promote collaboration and reproducibility [Link]. We can leverage TCIA for actual clinical cases with clinical relevance. Also, TCIA supports longitudinal studies, offering insights into disease progression and treatment responses [Link] [Link]. The TCIA community is active and has a plenty of available resources to support researchers. For those reasons, we choose TCIA for our examples in this paper.

Figure 2 shows the output of mapping patient IDs to clinical, genomic and imaging data from the GDC and IDC. Users can select the fields of interest to be displayed in the output table and then export the table to a CSV file for download. Not only do the users have access to the IDC data viewers (SliM for slide microscopy images and Open Health Imaging Foundation for radiology images) directly from our web server, users can also visualize and analyze DICOM images using Fiji, an open source well-established image package with extensive plugins. Our web server allows the user to automatically download DICOM images and analyze these images in Fiji (no installation needed if Docker deployment is used). These image analyses can be integrated with the genomic and clinical data corresponding to the same patients.

Illustrate how to use the web server with an example that showcases the differences between our Integrator vs. (GDC, IDC)

TCIA https://wiki.cancerimagingarchive.net/pages/viewpage.action?pageId=70226903
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6134102/

Breast cancer genomic + imaging data
GDC
https://portal.gdc.cancer.gov/repository?facetTab=cases&filters=%7B%22content%22%3A%5B%7B%22content%22%3A%7B%22field%22%3A%22cases.project.project_id%22%2C%22value%22%3A%5B%22TCGA-BRCA%22%5D%7D%2C%22op%22%3A%22in%22%7D%2C%7B%22content%22%3A%7B%22field%22%3A%22files.experimental_strategy%22%2C%22value%22%3A%5B%22Diagnostic%20Slide%22%5D%7D%2C%22op%22%3A%22in%22%7D%5D%2C%22op%22%3A%22and%22%7D&searchTableTab=cases
IDC https://portal.imaging.datacommons.cancer.gov/explore/

Our goal: to showcase the features of the CRDC Integrator that users cannot get from GDC and IDC

We select the TCGA breast data collection BRCA (Breast Invasive Carcinoma) that appear both in GDC and IDC. We keep 1,079 BRCA cases and leave off 1 DLBC (Lymphoid Neoplasm Diffuse Large B-cell Lymphoma) case in GDC to focus on similar dataset for the use case. In IDC, we have 1,098 cases. We choose a specific case TCGA-3C-AAAU to illustrate the results between GDC, IDC, and the CRDC Integrator. GDC provides the information of the case with genomic focus while IDC has the focus on the image information. GDC does not have the metadata information of the images, but CRDC Integrator with the information collecting based on IDC has full metadata of IDC.

The CRDC Integrator presents an innovative and comprehensive solution that helps research in bioinformatics to analyze the imaging data. With an emphasis on providing seamless integration and advanced capabilities, the CRDC Integrator has been developed to fostering a collaborative and user-friendly environment for researchers and professionals in the genomics field. Our software tool represents a significant advancement in the genomics research landscape including integration with QuPath and Fiji, advanced data filtering based on genomics primary sites and experimental strategy, direct access to DICOM images, and flexible output formats from API calls. These features collectively empower researchers to extract valuable insights and accelerate their genomics research endeavors. Below is a table of comparisons between our web server, GDC, and IDC.

Features	**CRDC Integrator**	**GDC**	**IDC**
Integration with QuPath/Fiji	Yes		
Filtering by genomics primary sites	Yes	Yes	Yes
Direct link to download DICOM images	Yes		
Ouptut formats from API calls	Yes		

GDC: allow the users to access the genomic data (both controlled and open access), also visualize the slide image
IDC: filter, display metadata of the images, can visualize the OHIP viewer, slim viewer? (will check)

Integrator: Fiji integration

Spreadsheet summarizing and comparing features
https://docs.google.com/spreadsheets/d/1LSSf9vIsVHYNWZbreOKnXXUGQGJfZcJTAXQKWEEUmjM/edit?usp=sharing

Figure 2: Steps of using the CRDC Integrator. (a)

The uploaded file will be displayed in the drop down list right below the upload button. We select

the CSV file that just uploaded and then wait for seconds for loading the data into the table in the main part. The performance of this process depends on how fast internet connection we have. The middle table in **Figure 1** with annotation #5 allows users to select the features to display. There are three types of features: clinical, genomics, and imaging features. Selecting the feature DICOM will enable a column in the result table to allow users to download the DICOM image to the local machine. Each image for the first time downloading will be caches by the web server to improve the performance of downloading. The feature SliM & OHIF will enable two columns in the result table to allow users directly link to the IDC web page with the selected DICOM image. This feature is useful if the users do not have a DICOM viewer on their local machines. The web page also has an option to select the number of records displayed on one page and another option is to export the result table to CSV file for downloading.

The web page load the table of result set right after we select the features. In this example, we have PatientID (default), OHIF & SliM, and DICOM URL links to download. When we click on the “TCGA-G4-6323”, it open a new tab navigating to the GDC web page that consists of meta data of the case “TCGA-G4-6323”. The web page also allows users to view the DICOM images directly from IDC by clicking on the “OHIF” or “SliM” links. We can also download the DICOM images to the local host for analysis purpose by clicking on the “Download”.

The benchmarking results of runtime: (time in seconds)

We use gCloud credential JSON file to directly connect to the IDC database through BigQuery, a software that is provided by Google Cloud Platform. The experiments include the database with and without indexing to see how the web server performs along with the number of fields filtered out (one field and five fields). There are two environments for each test: one is on a local machine and one is on a gCloud VM to see how the network latency impacts the performance of the web server. We repeat each test case five times to produce the average runtimes (in seconds).

Techniques Used	Test Environment	25 Items per page		100 items per pages	
		1 field	5 fields	1 field	5 fields
BigQuery	Local Machine	< 1	3.869	< 1	5.148
BigQuery	gCloud VM	< 1	< 1	< 1	< 1

Table 1: Benchmarking Results

Using the BigQuery ensures the metadata is always up-to-date. The reason why we choose Google Cloud Platform (GCP) over the other cloud providers is because the main IDC database

resides on GCP which can help us reduce the network latency while retrieving the data from the web server. The performance of a query is significantly improved when we deploy the web server on a GCP VM. The configuration of the local machine in the experiments was Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz and 32 GB RAM.

METHOD (1/9/24 NEED TO UPDATE)

The source code of our web server, video tutorials and documentation are publicly available on GitHub (https://github.com/Biodepot-LLC/CRDC_Integrator). In this section, we will describe how the user can set up and deploy the web server. We will also describe how patients are mapped across the GDC and IDC, and how images are retrieved from the IDC. The integration of Fiji is also reported. **Figure 3** summarizes the implementation design, and relationship between user interaction and php scripts.

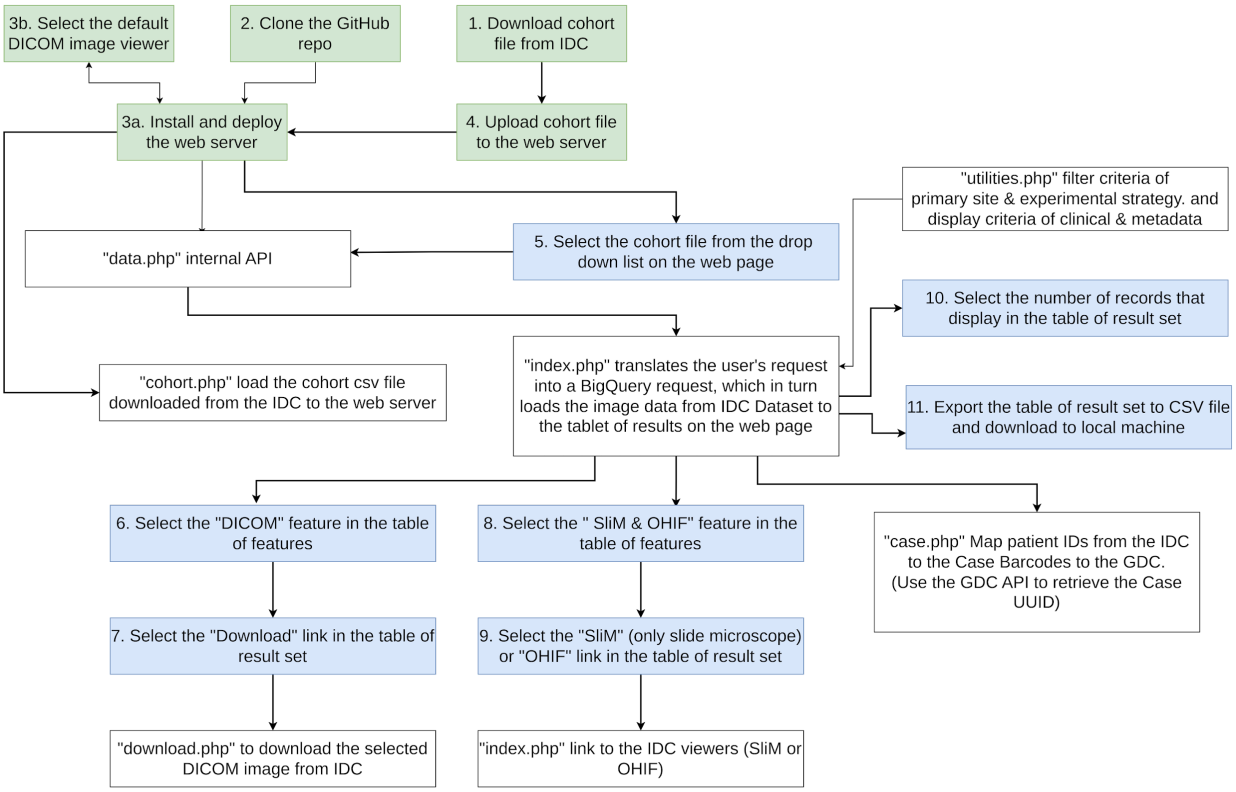


Figure 3: Overview of implementation design of the CRDC Integrator. Steps required to install and deploy the web server are highlighted in green. Steps after the web server is started are shown in blue. Various php scripts are summarized in white boxes.

1. Download the cohort file in CSV format from the IDC.

2. Clone the repository of CRDC Integrator from GitHub at

https://github.com/Biodepot-LLC/CRDC_Integrator.

3. The cohort.php script load all uploaded cohort files into a drop down list component (2) in Figure 1. The data.php script has its main function to take the request from the user and transfer to the IDC data set then retrieve the response and send back to the user.

4. The index.php script takes the cohort file uploaded and stores in the web server.

5. The index.php script will display and format the response in the bottom table of result set in Figure 1. Before sending back the response to the client, the web server use the script "utilities.php" to do the criteria filtering of primary sites and experimental strategies.

6. The data.php translates the response from the IDC to construct an endpoint with required arguments for DICOM image downloading.

7. The user clicks on the "Download" link that is constructed in the step 6 to download one DICOM image.

8. The data.php translates the response from the IDC to construct an endpoint with required arguments for DICOM image viewers.

9. The user clicks on either "SliM" or "OHIF" (different viewers) to view DICOM image(s) from the IDC directly.

Installation and deployment of the CRDC Integrator

A few software tools need to be installed before installing and deploying the web server. Since the IDC is built on the Google Cloud Platform (GCP) [9], gCloud tools and resources are needed to query the IDC. Therefore, the user needs to install the gCloud CLI, a set of command-line tools to create and manage Google Cloud resources, before installing the web server. The user needs to register for a Google Cloud account to access these resources, but downloading these tools are free. Before running the web server, the user also needs to download the gCloud credential file in JSON format and provide it in the Docker command as an input. In addition, the user needs to install PHP, a scripting language for web development. Next, the user should download the source code from our GitHub repository and run the build.sh script to install the Docker image of the web server. The server takes approximately 5 to 10 seconds to set up all configurations before it starts. See the README on our GitHub repository for the exact commands to pull and build the Docker container, as well as to deploy the web application. A demo video illustrating all these steps are available at URL.

Mapping of case barcodes across the GDC and IDC

In order to map patients across the GDC and IDC, the user needs to download the cohort file

consisting of all cases (as identified by “Patient IDs”) in the CSV (comma separated values) format from the IDC Data Portal at https://portal.imaging.datacommons.cancer.gov/. Note that the user needs to sign into the IDC using a Gmail-based account, select all scopes and save new cohort as shown in the annotated screenshot in **Figure S1**. This CSV file is then uploaded to the web server after the web server is started as shown in step (1) and step (2) in **Figure 1**.

A challenge of mapping patients across different data commons is the use of different terminologies. We contacted the GDC support team who recommended the use of “Case Barcodes” to map patients across the GDC and IDC. In addition, we observe that “Patient IDs” in the IDC are the same as “Case Barcodes” in the GDC. A UUID (Universal Unique Identifier) is a 128-bit number used to uniquely identify an object or entity in the GDC [11]. Next, we use the GDC API [6], along with the input Patient ID from the IDC (also known as the “Case Barcodes” in the GDC), to retrieve the Case UUID. This Case UUID is needed to query metadata from the GDC (see **Figure 3**). Specifically, we construct a curl command that will query the GDC server, save the response from the GDC server in the JSON format, and subsequently extract the Case UUID from the JSON file. We append this Case UUID to the endpoint https://portal.gdc.cancer.gov/cases/ such that the corresponding metadata from the GDC will be displayed in the web browser when the user can click on the Patient ID in the results table of the web server. This metadata from the GDC will be displayed as a new tab from the GDC web page https://portal.gdc.cancer.gov/cases/CASE_UUID. This mapping of IDs corresponds to step (6) in **Figure 1**.

We illustrate the mapping of IDs using an example Case Barcode (or Patient ID) TCGA-02-0003. After executing the following curl command

```
$ curl -s
"https://api.gdc.cancer.gov/cases?filters=%7B%22op%22%3A%22and%22%2C%22content%22%3A%5B%7B%22op%22%3A%22in%22%2C%22content%22%3A%7B%22field%22%3A%22submitter_id%22%2C%22value%22%3A%5B%22TCGA-02-0003%22%5D%7D%7D%5D%7D&format=json&size=10000" | jq -r '.data.hits[].id'
```

The Case UUID df3c1d61-79c1-43e9-971a-8029497ffeab is returned. Subsequently, clinical metadata is extracted from https://portal.gdc.cancer.gov/cases/df3c1d61-79c1-43e9-971a-8029497ffeab, shown in **Figure S2**.

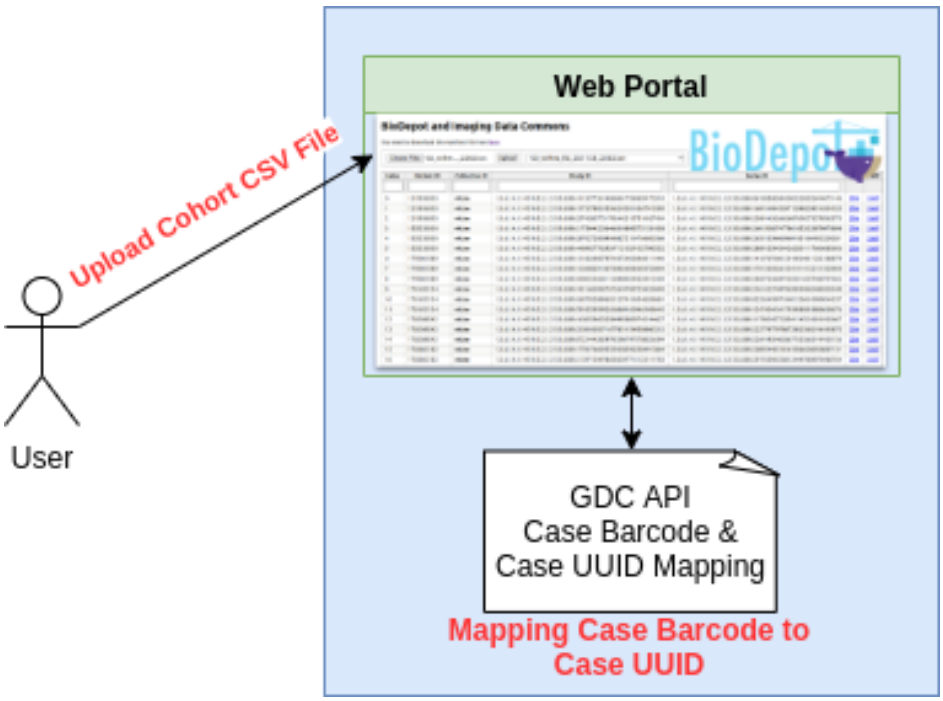


Figure 4: Summary of mapping of identifiers across IDC and GDC. The first step is to download and upload the cohort CSV file from the IDC consisting of a list of all Patient IDs. The second step is to construct a curl command to retrieve the Case UUID, which is required to retrieve the metadata from the GDC.

Filtering and display criteria

The cohort CSV file include all major primary sites such as brain, breast, eye, etc. The users may only choose specific primary sites related to their studies. To keep the results more concise to the users, the web server offer an option to filter only primary sites that users focus on. The web server uses “utilities.php” script to send a request in GraphQL format to GDC API (https://api.gdc.cancer.gov/cases) to limit the number of records relative to the primary sites that users select. The “utilities.php” script makes a request to GDC API as following:

curl -s --data 'filters={\"op\":\"and\",\"content\":[{\"op\":\"in\",\"content\":{\"field\":\"cases.submitter_id\",\"value\":[@patient_ids]}},{\"op\":\"in\",\"content\":{\"field\":\"cases.primary_site\",\"value\":[@primary_sites]}}]}&fields=case_id,submitter_id,primary_site&size=10' 'https://api.gdc.cancer.gov/cases' | jq -r '.data.hits | map(.submitter_id) | join(\",\")' where @patient_ids is a list of Patient IDs or Case BarCodes and @primary_sites is a list of primary sites we want to include in the results. Behind

the “utilities.php” script, we create a function that takes a first list of primary sites (Breast, Lung, Eyes, etc.) that we want to include and a second list of patient IDs. The function uses the curl command to send a request to the GDC API to filter all patients IDs that have primary sites as included in the first list. With this filtering feature, our web server display only the data that have the primary sites specified from the drop down list of primary sites. The web server uses the same way to filter experimental strategies.

Accessing DICOM images from the IDC (@varik: elaborate this paragraph with respect to Fig 4)

We upgrade our web server to a new architecture that can maintain the up-to-date database by using BigQuery to connect to the IDC database directly.

Explain our API

How you constructed queries to query Google Cloud to retrieve the images

Explain annotations A, B, C, D for different components in the text and figure 4 caption

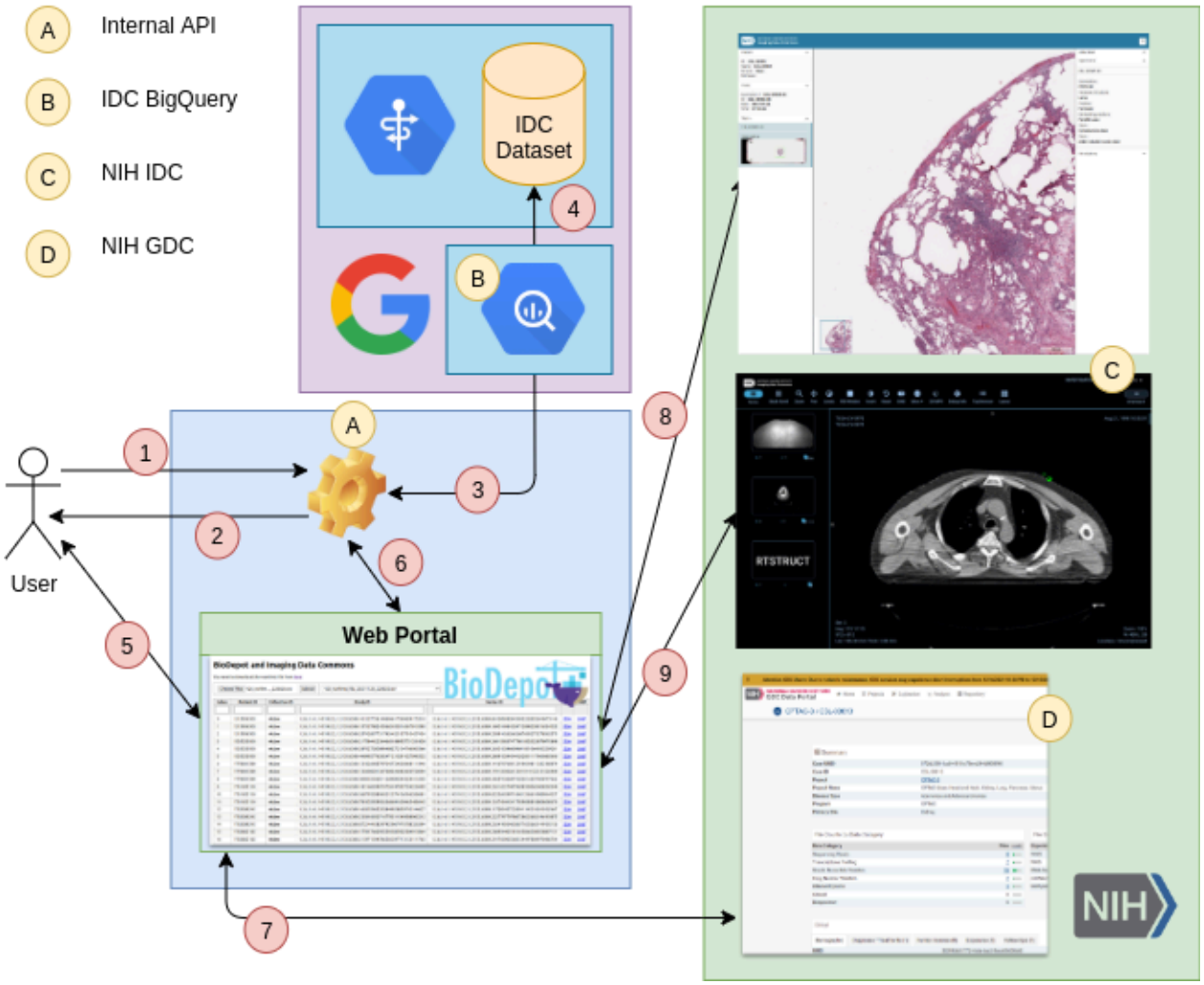


Figure 4: Annotated API calls in the CRDC Integrator implementation:

1. The user sends a direct request (not from the browser) to our API
2. The user receives a direct response from our API
3. The API sends request to BigQuery (gCloud)
4. The BigQuery executes the statement and sends the result back to our API
5. The user runs the web browser to access our web server
6. The web browser sends a request and receives response from our API
7. The web browser opens a new tab to display meta data of the Case barcode from GDC
8. The web browser opens a new tab to display DICOM image(s) using SliM from IDC
9. The web browser opens a new tab to display DICOM image(s) using OHIF from IDC

Integration with QuPath and Fiji

Fiji is a widely used open-source software suite with extensive plugins for scientific image processing (cite Fiji). Our CRDC Integrator web server is distributed as a Docker image with both Fiji and Mozilla Firefox applications installed. Most importantly, we have integrated Fiji in the Firefox browser. Since the Firefox browser has default applications to run through its operating system, we changed the default application in our Docker image to allow Firefox to recognize the DICOM image files by specifying the .dcm extension. With this configuration, the downloaded DICOM image file will be loaded and opened in Fiji when the user clicks on the DCM file from the download section of the browser. This corresponds to step (8) in **Figure 1**.

To view downloaded DICOM images using Fiji, the CRDC Integrator web server must be deployed using the Docker command with the browser option. **Figure 5** summarizes the steps involved in generating a Docker image for the CRDC Integrator web server. After this Docker image is generated, there are two options to deploy the web server in the Docker container: *server-only* mode and *browser-integrated* server mode (see **Figure 6**). In the *server-only mode*, the CRDC Integrator web server will be deployed without the Mozilla Firefox browser, such that Fiji is not automatically integrated. The user needs to manually install the browser's plugin/extension on the local host to have this feature enabled. On the contrary, a Firefox browser window is automatically opened after starting the web server in the *browser-integrated* mode. The user does not have to perform any additional installation because the Fiji plugin/extension is already integrated in the Firefox, which are both included in the container.

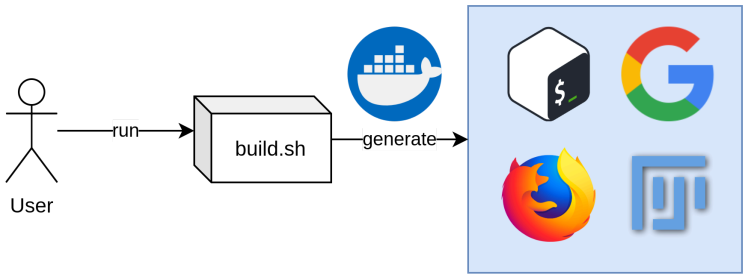


Figure 5: Overview of building a Docker image for the CRDC Integrator web server. To build the web server for the CRDC Integrator, the user runs a shell script called "build.sh", which generate a Docker image containing the terminal, Mozilla Firefox, Google Cloud utilities and Fiji applications.

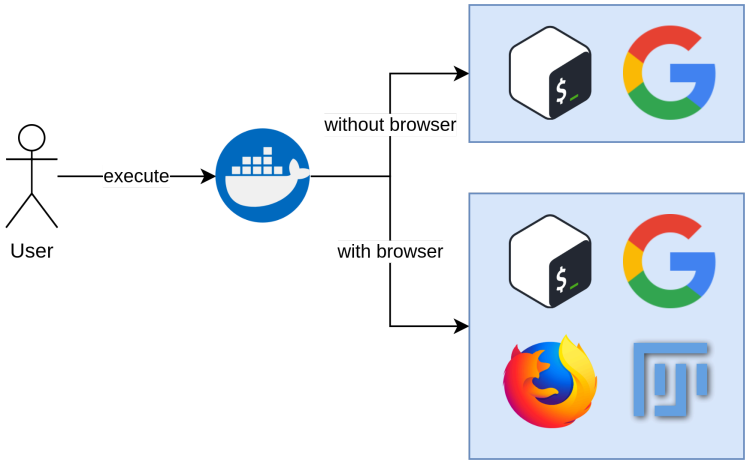


Figure 6: A user can run the Docker container to deploy CRDC Integrator web server in two different modes: *server-only* mode (without the Firefox browser) and *browser-integrated* server (with the Firefox browser).

Integration with the Biodepot-workflow-builder

The Biodepot-workflow-builder (Bwb) is a containerized and cloud-enabled graphical workflow engine {Hung, 2019 #29}. Users can create multi-step analytical workflows with graphics support in the Bwb. Each modular step in a Bwb workflow is encapsulated in a Docker container with systematic version tags, thus enhancing reproducibility of analyses. We previously extended the Bwb platform to support interactive and reproducible image analyses using the Fiji suite (cite Scientific reports paper).

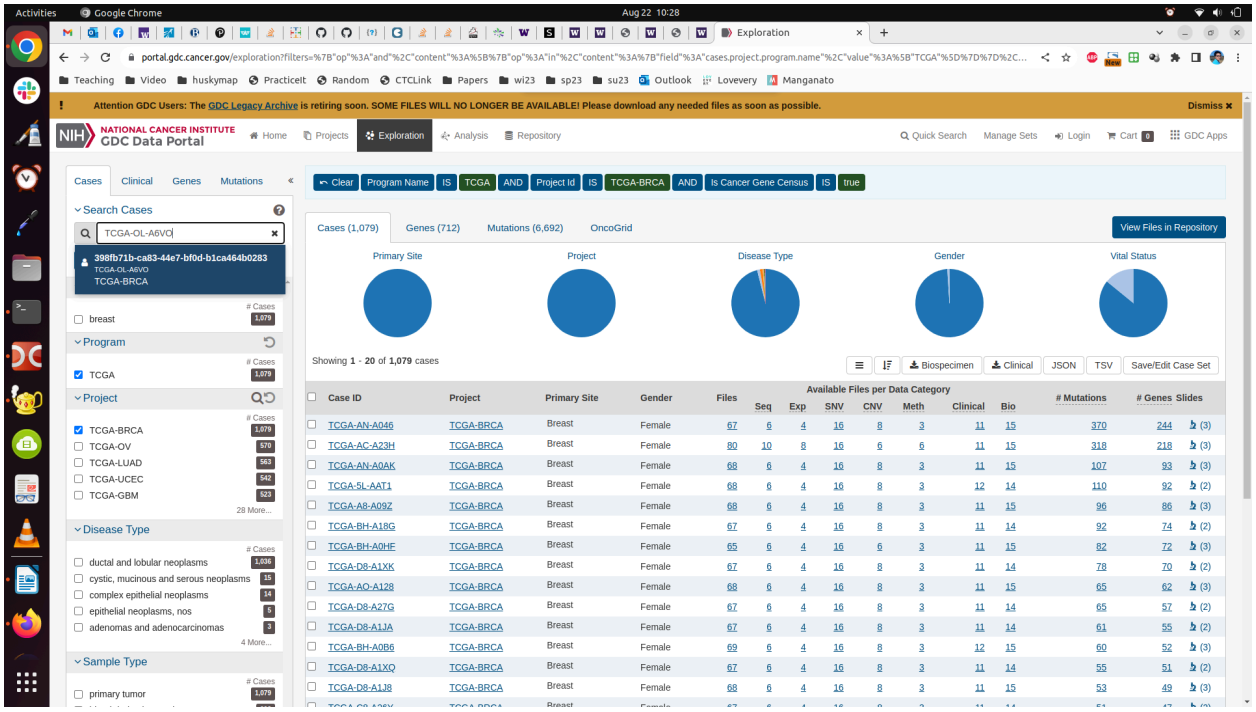
We can use this web server to download DICOM images from IDC gCloud Storage for many purposes.

Bwb supports drag-and-drop feature to help users easily design the widgets in workflow. We will have a widget to hold our web server that has firefox integrated to display all information from the given IDC cohort file (CSV) , especially the direct links to view DICOM images from the IDC web portal. For machine learning purposes, we can use another widget with Jupyter notebook to download all necessary DICOM images and train a model.

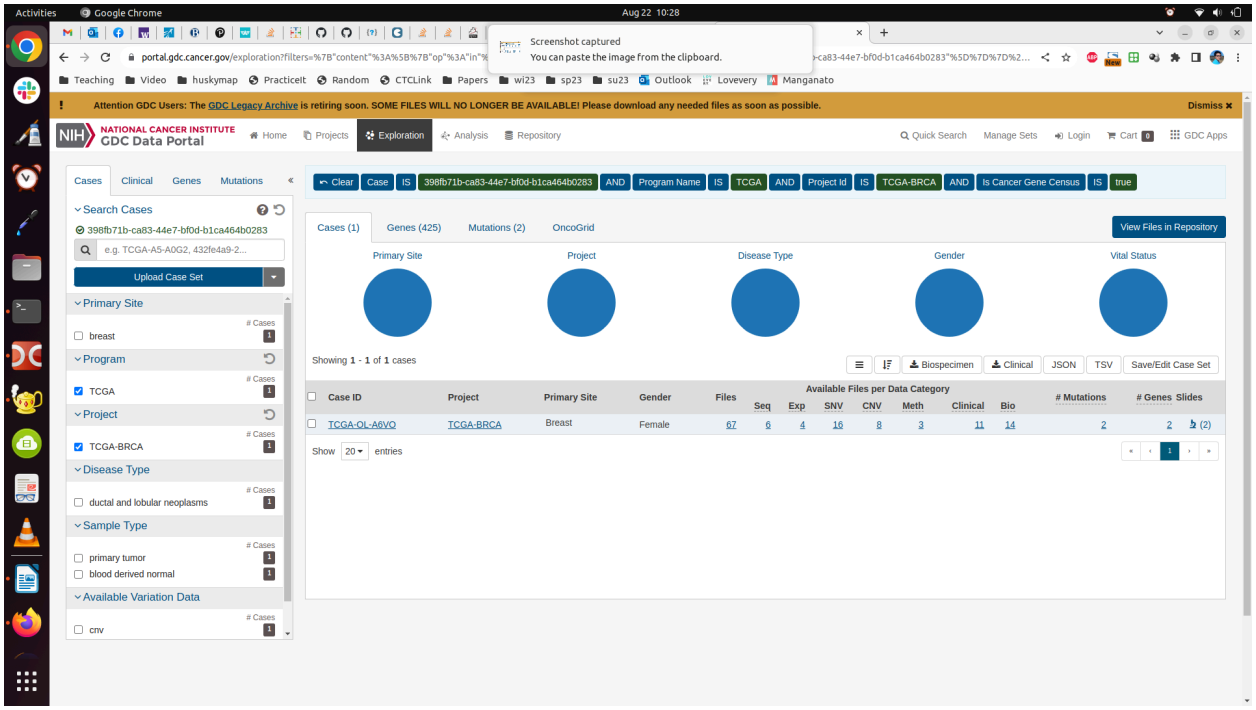
Figure 2 shows the output of mapping patient IDs to clinical, genomic and imaging data from the GDC and IDC. Users can select the fields of interest to be displayed in the output table and then export the table to a CSV file for download. Not only do the users have access to the IDC data viewers (SliM for slide microscopy images and Open Health Imaging Foundation for radiology images) directly from our web server, users can also visualize and analyze DICOM images using Fiji, an open source well-established image package with extensive plugins. Our web server allows the user to automatically download DICOM images and analyze these images in Fiji (no installation needed if Docker deployment is used). These image analyses can be integrated with the genomic and clinical data corresponding to the same patients.

Illustrate how to use the web server with an example that showcases the differences between our Integrator vs. (GDC, IDC)

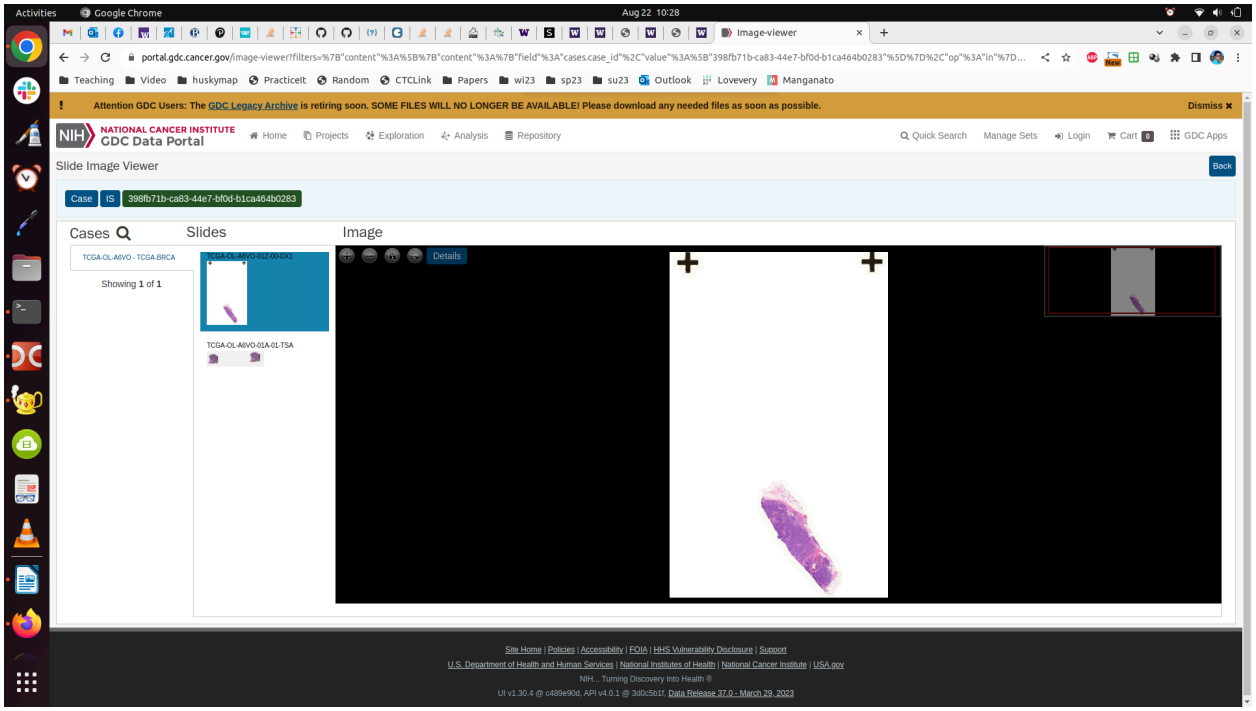
We select a TCGA collection of images to have an example showing the differences between our integrator, GDC, and IDC. The figure below showing how we select or search for the case of images TCGA-OL-A6VO from the collection TCGA-BRCA.



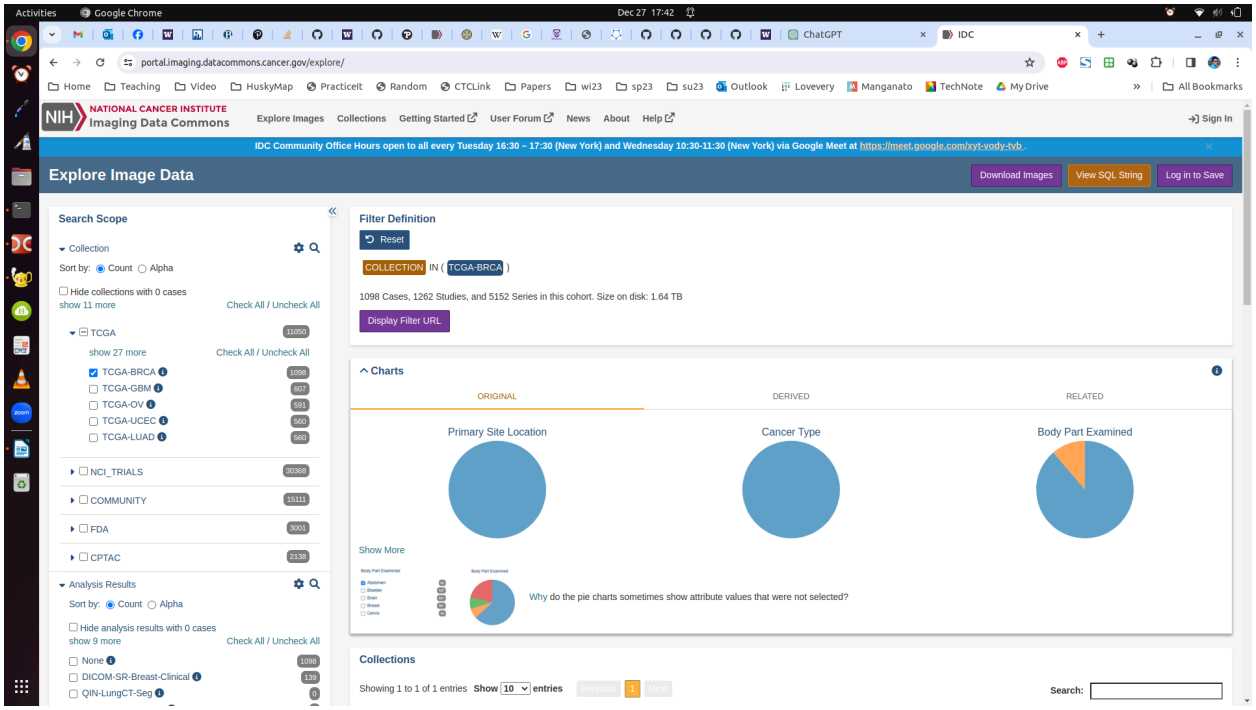
There is one record displayed in the bottom table that indicating two slide images we can view.



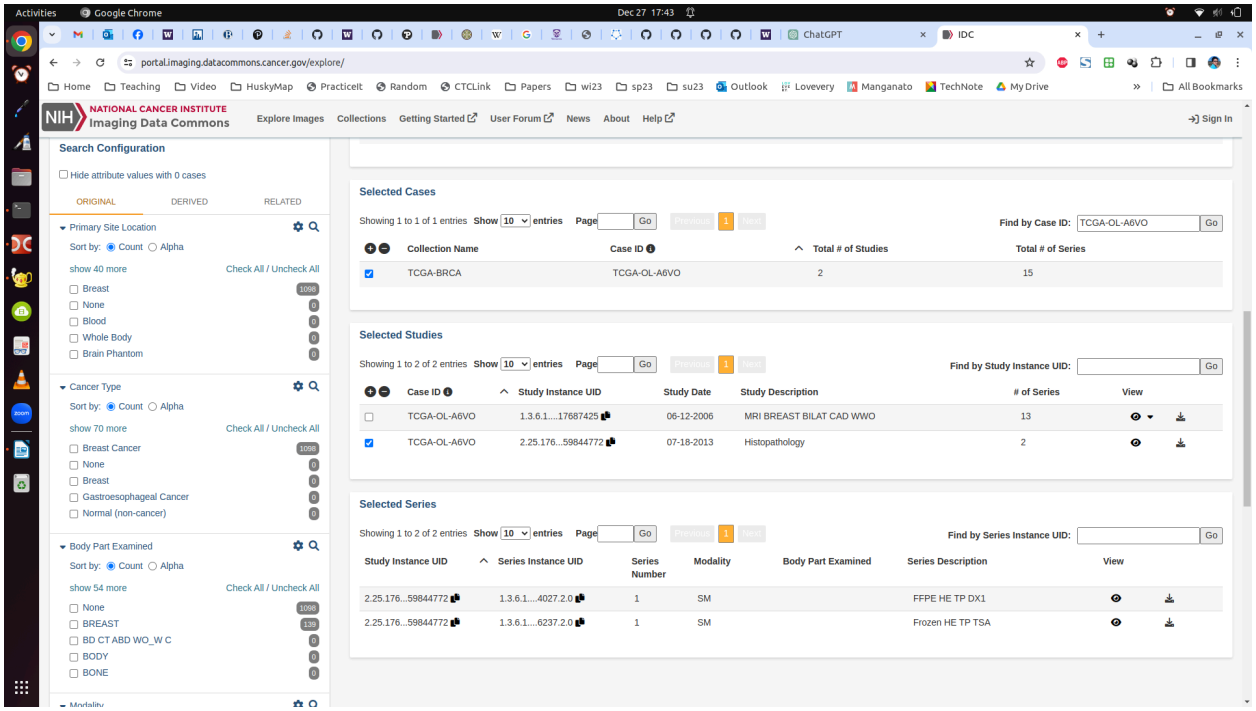
The GDC open an browser-integrated image viewer that has a list of images. This is only for the images that are viewed by slide microscope.



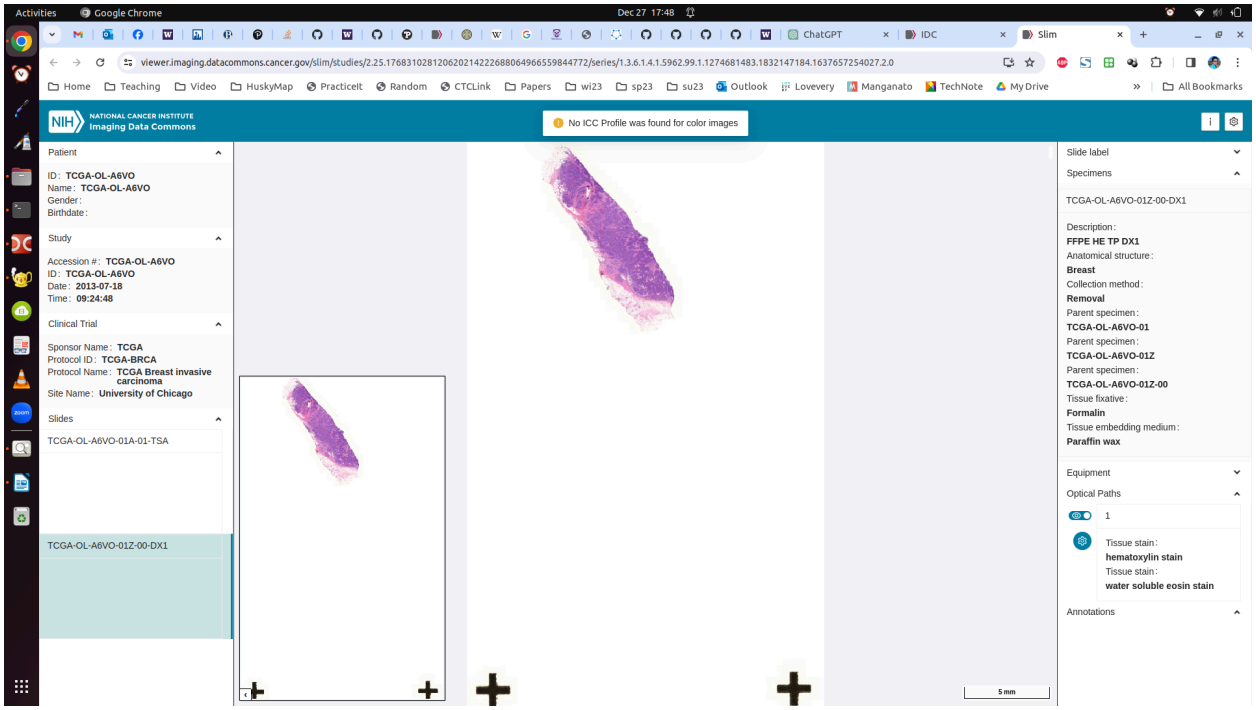
Similar to GDC, IDC provides a search tool that allows us to navigate to the images of TCGA-OL-A6VO.



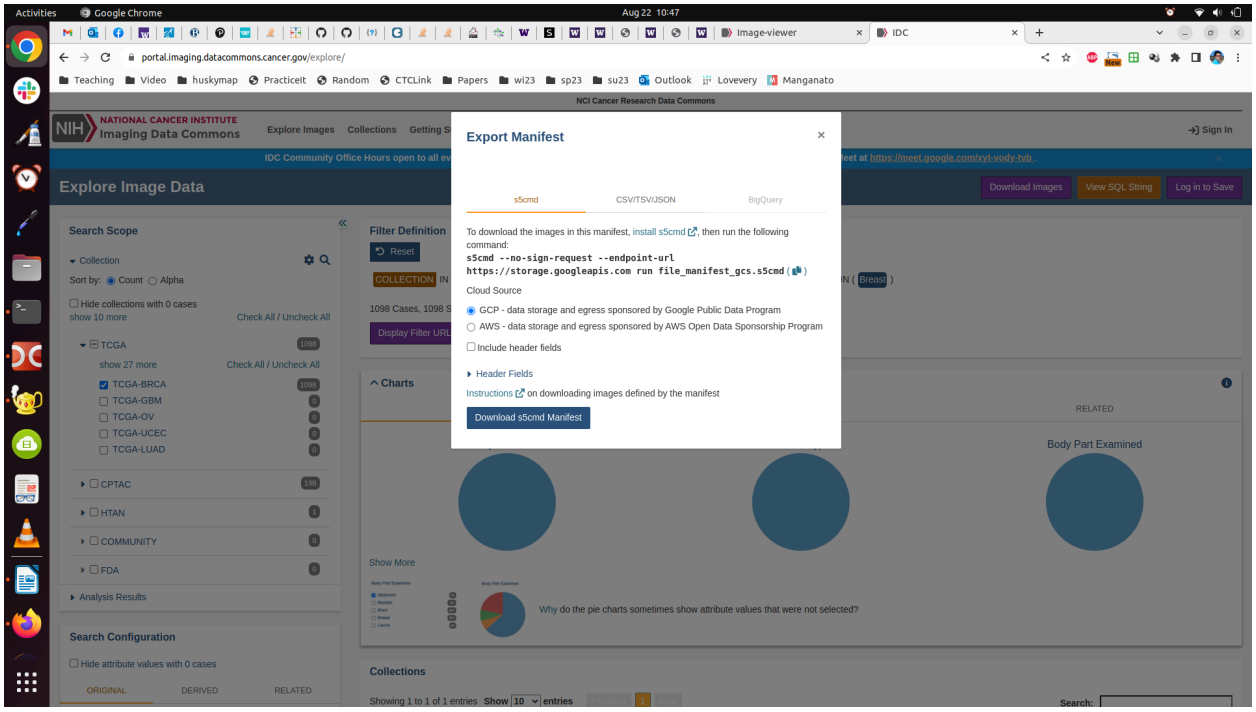
Since IDC provides more images than GDC, we want to show the same image in the study we pick. We select the “TCGA-OL-A6VO” with the study description “Histopathology” and the series description “FFPE HE TP DX1”



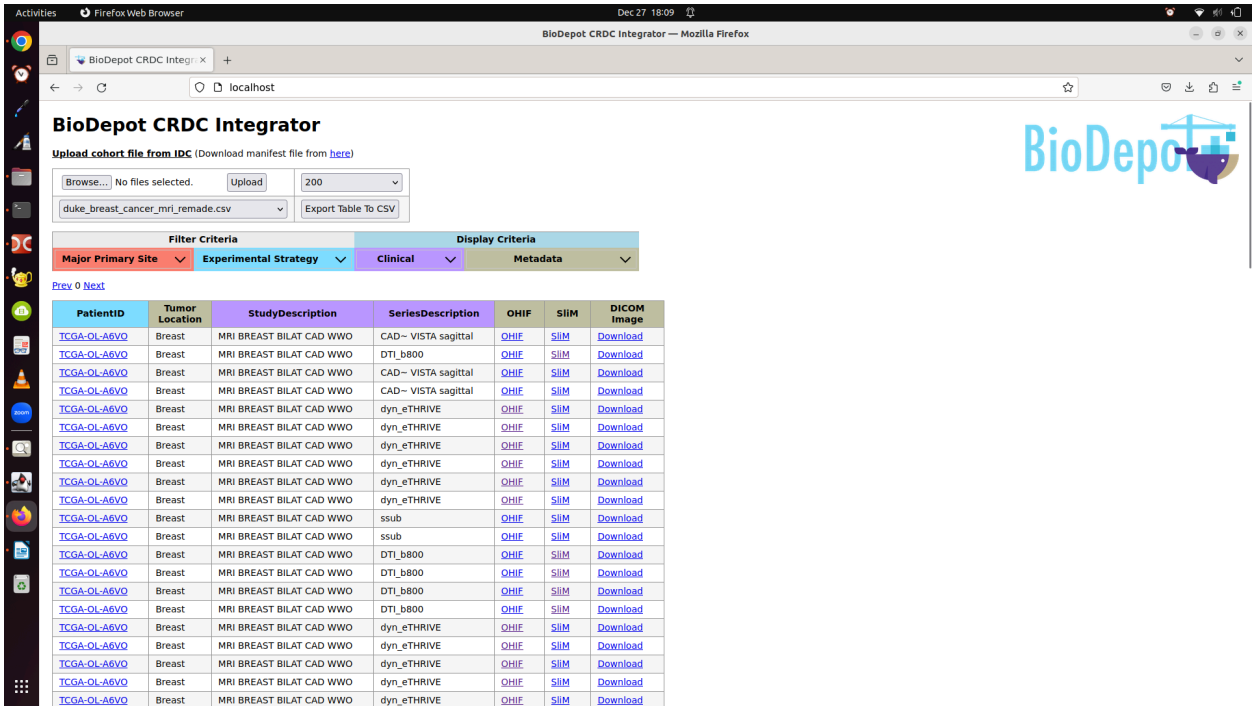
After we click on the “eye” icon, IDC opens the image in a new tab using the SliM viewer



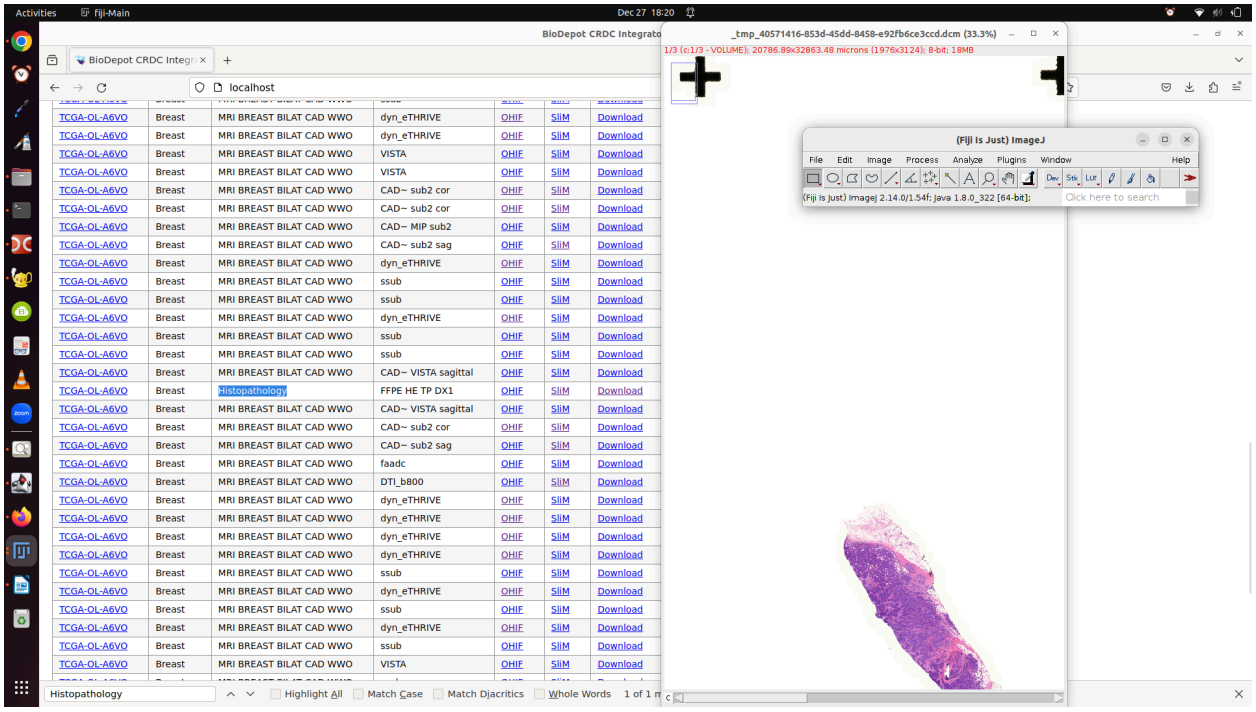
Moreover, the platform allows us to download images with some known cloud techniques that requires users to understand to download manually.

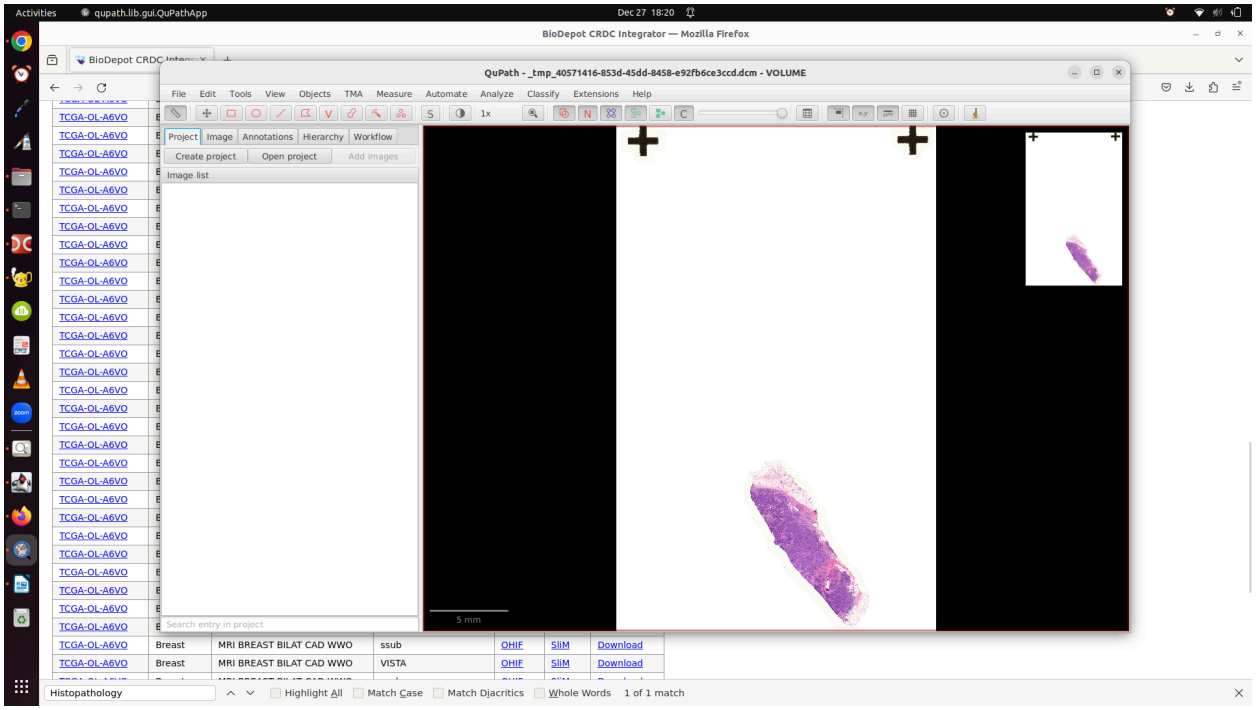


Our integrator can help users view the images not just using the slide microscope but also 3D image (mostly DICOM). The users can also download the images without knowing anything about cloud techniques by one single click on the links.



Moreover, the user can open the images from the browser to enable the Fiji or QuPath integration for analysis if the user chooses to use the Docker version.





TCIA https://wiki.cancerimagingarchive.net/pages/viewpage.action?pageId=70226903
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6134102/

Breast cancer genomic + imaging data
GDC
https://portal.gdc.cancer.gov/repository?facetTab=cases&filters=%7B%22content%22%3A%5B%7B%22content%22%3A%7B%22field%22%3A%22cases.project.project_id%22%2C%22value%22%3A%5B%22TCGA-BRCA%22%5D%7D%2C%22op%22%3A%22in%22%7D%2C%7B%22content%22%3A%7B%22field%22%3A%22files.experimental_strategy%22%2C%22value%22%3A%5B%22Diagnostic%20Slide%22%5D%7D%2C%22op%22%3A%22in%22%7D%5D%2C%22op%22%3A%22and%22%7D&searchTableTab=cases
IDC https://portal.imaging.datacommons.cancer.gov/explore/

Our goal: to showcase the features of the CRDC Integrator that users cannot get from GDC and IDC

GDC: allow the users to access the genomic data (both controlled and open access), also visualize the slide image
IDC: filter, display metadata of the images, can visualize the OHIP viewer, slim viewer? (will

check)

Integrator: Fiji integration

Spreadsheet summarizing and comparing features
https://docs.google.com/spreadsheets/d/1LSSf9vIsVHYNWZbreOKnXXUGQGJfZcJTAXQKWEEUmjM/edit?usp=sharing

CONCLUSIONS

We relied on the cloud database, BigQuery, where we could retrieve the data always up-to-date. We ran the experiments for both cases on the local machine to observe the differences in run time between standalone database and cloud database.

REFERENCES

1.
2.
3.
4. https://api.imaging.datacommons.cancer.gov/v1/swagger#/queries/queryMetadata
5. https://www.sqlite.org/index.html
6. https://www.researchgate.net/publication/349764039_Analyzing_Performance_Differences_Between_MySQL_and_MongoDB

1. **NCI Cancer Research Data Commons (CRDC)** [https://datascience.cancer.gov/data-commons]
2. **NCI Genomic Data Commons (GDC)** [https://gdc.cancer.gov/]
3. Grossman RL, Heath AP, Ferretti V, Varmus HE, Lowy DR, Kibbe WA, Staudt LM: **Toward a Shared Vision for Cancer Genomic Data**. *N Engl J Med* 2016, **375**(12):1109-1112.
4. **National Cancer Institute Imaging Data Commons** [https://portal.imaging.datacommons.cancer.gov/]
5. Fedorov A, Longabaugh WJR, Pot D, Clunie DA, Pieper S, Aerts H, Homeyer A, Lewis R, Akbarzadeh A, Bontempi D *et al*: **NCI Imaging Data Commons**. *Cancer Res* 2021, **81**(16):4188-4193.
6. **GDC Application Programming Interface (API)** [https://gdc.cancer.gov/developers/gdc-application-programming-interface-api]

7. **GenomicDataCommons: NIH / NCI Genomic Data Commons Access** [https://bioconductor.org/packages/GenomicDataCommons]
8. Morgan MT, Davis SR: **GenomicDataCommons: A bioconductor interface to the NCI Genomic Data Commons**. *bioRxiv* 2017:117200.
9. **IDC User Guide** [https://learn.canceridc.dev/api/getting-started]
10. Schindelin J, Arganda-Carreras I, Frise E, Kaynig V, Longair M, Pietzsch T, Preibisch S, Rueden C, Saalfeld S, Schmid B *et al*: **Fiji: an open-source platform for biological-image analysis**. *Nat Methods* 2012, **9**(7):676-682.
11. **GDC Documentation: Universally Unique Identifier (UUID)** [https://docs.gdc.cancer.gov/Encyclopedia/pages/UUID/]
12. **IDC API** [https://api.imaging.datacommons.cancer.gov/v1/swagger#/queries/queryMetadata]
13. **SQLite** [https://www.sqlite.org/index.html]

SUPPLEMENTARY INFORMATION

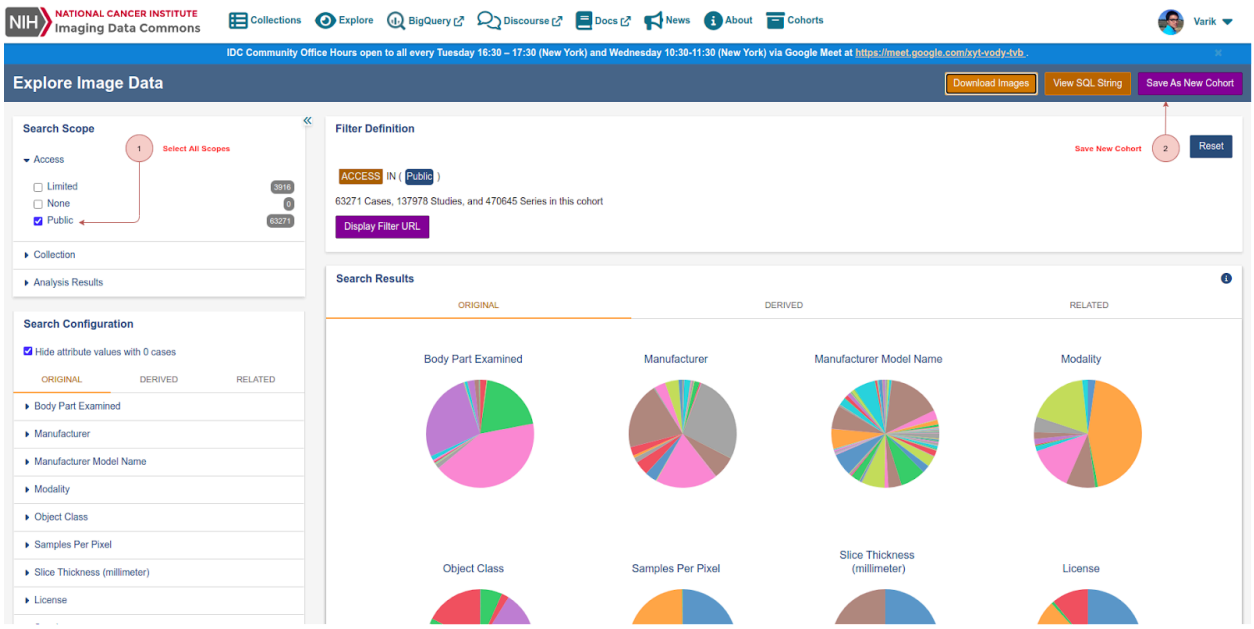


Figure S1: After signing in to the IDC using a Gmail account, the user should select all scopes (step 1) and click “save new cohort” (step 2).

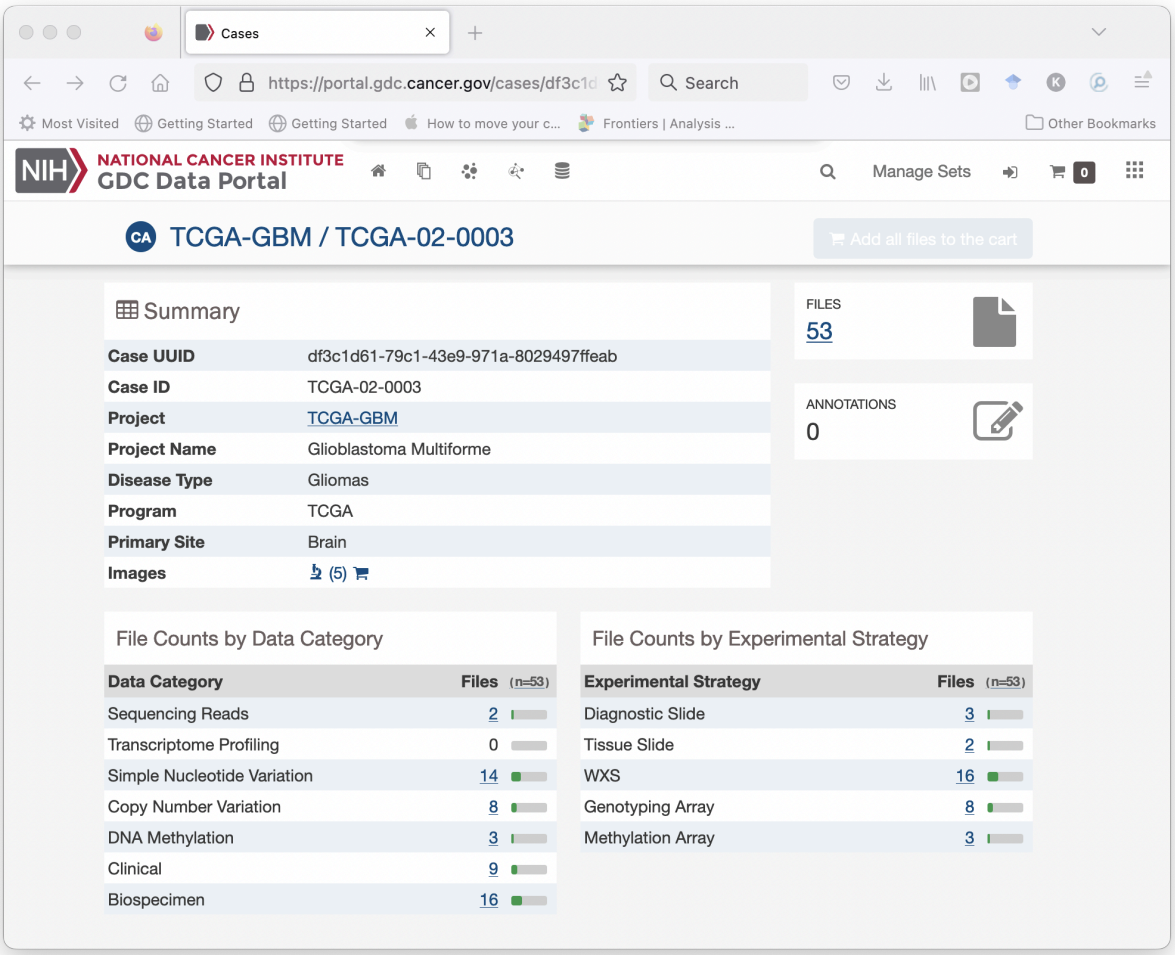


Figure S2: Metadata from GDC for Case Barcode TCGA-02-0003 is available at https://portal.gdc.cancer.gov/cases/df3c1d61-79c1-43e9-971a-8029497ffeab