HCA Orange Box Data Browser: Functional Specification and Use Cases

What is the Data Browser

The Data Browser provides a basic web UI for researchers to quickly filter data stored in the DSS. It is not an analysis or visualization tool itself, the Data Browser focuses on finding data in the DSS, enabling users to download for analysis in other systems. The Data Browser is part of the overall Data Portal.

Related Documents

Document	Link
UI/UX Mocks and Interviews	https://docs.google.com/document/d/17KRUW8s091AcJsQdfXUipXlK6q-ukHn_btHWT8L3owY/edit
Project Roadmap for HI, H2 and beyond	https://docs.google.com/document/d/1ydgx5YGTAHZv3rb7VVdx_2Zp6YLBeNgLGRpEL7GjyAk/edit#
Lower Instances	http://explore.dev.data.humancellatlas.org/ http://explore.integration.data.humancellatlas.org/ http://explore.staging.data.humancellatlas.org/
Zenhub Board	https://app.zenhub.com/workspace/o/humancellatlas/data-browser/boards?repos=124946326
Github Repo	https://github.com/HumanCellAtlas/data-browser
Data Portal (not browser) Spec	https://docs.google.com/document/d/1WReZHrghEeHeE1UAx_FWQbt-BJIwmHyja3tExE3AqL0/edit
ICA Bundle Correction Document	https://docs.google.com/document/d/1VnCsOIK7U3DWmXdmmZpxbFlvVL8jd0NJU09cRt0kxpE/edit

High-Level Description

The Data Browser provides a basic web UI for researchers to quickly filter data stored in the DSS. This will be consistent with HCA branding/UX. It will include a UI to browse data releases/freezes, query/download from the Data Storage System (DSS e.g. Blue Box), and provide a consistent interface to “hand off” search results to other analytical systems. The Orange Box is not a Red Box; it is not an analysis, visualization, or exploration portal beyond simple data browsing; it is important that we not discourage a robust tool ecosystem.

We envision a tool that enables researchers to quickly filter and identify data stored in the DSS using a faceted browser. In that way, it’s very similar to the ICGC Portal, specifically the file browser element of this portal, the GDC Portal, and Boardwalk.

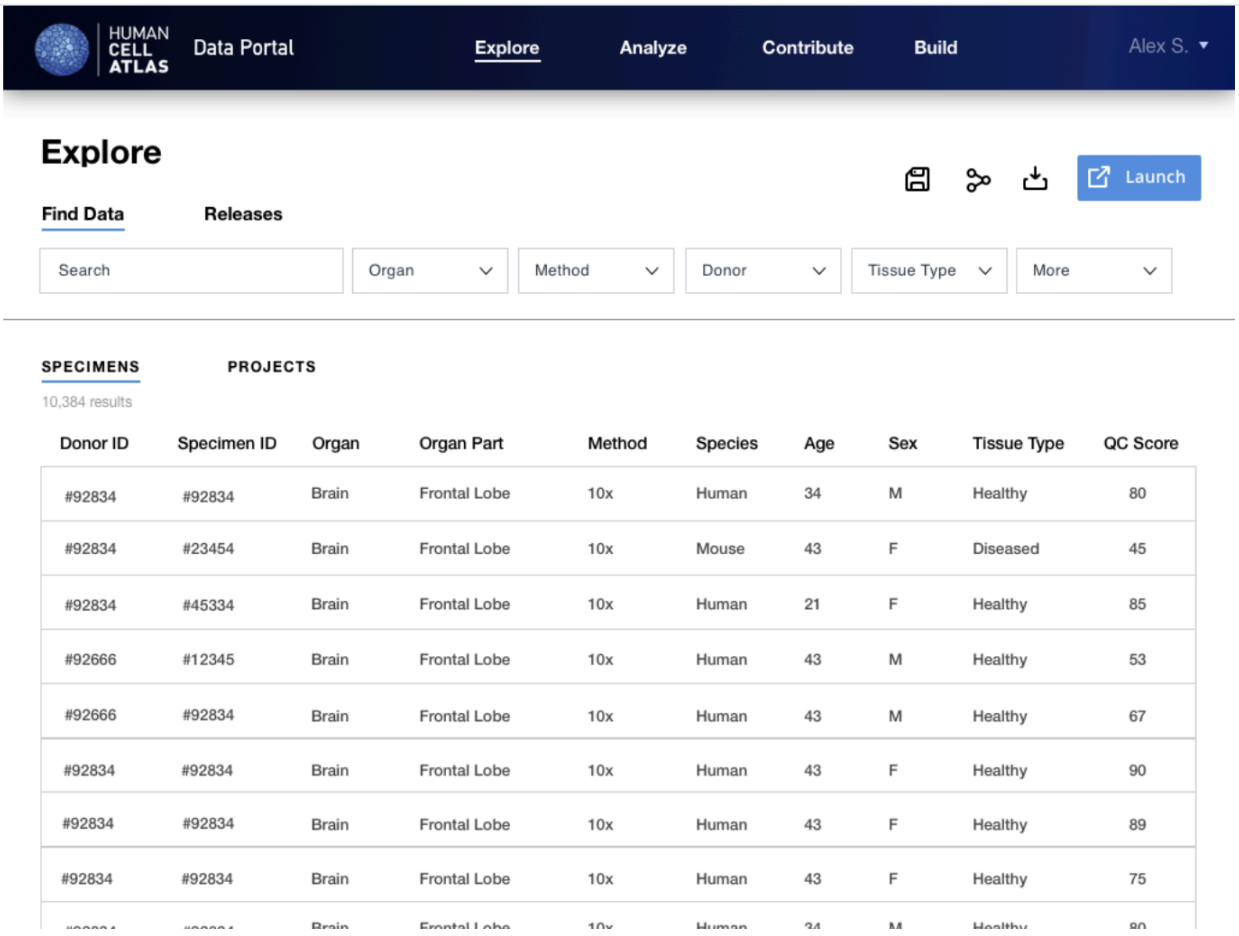


Figure: A mock of the Data Browser file/release browser provides a faceted browser to find data files or a variety of types from specific HCA data releases.

Users

Researcher - Computational
Motivation: As a computational researcher, I want to select and download a subset of data in the Atlas that is similar to what I am working with for my research, so that I can compare my data with a larger set of data.

Researcher - Experimental
Motivation: As an experimental researcher, I want to be able to view a summary of the data available for the HCA. I want to be able to narrow this down to data my group has submitted and use the Data Browser to retrieve analytical results from this subset of data.

Red portal developers
Motivation: As a “Red Box” Tertiary Analysis Portal developer, I want to 1) access Orange box REST services such as the “collections” service (allows you to save search results and share them), the file and other indexes. These will allow me to build my portal faster. Similarly 2) I want to have the Data Browser integrate with my site, so users of the Data Browser can find data they are interested in and then send it to my portal for analysis or visualization.

Data release curators (DCC staff)
Motivation: As a member of the data release working group, I want to generate a manifest of what data is in the Atlas so that I can select which datasets/bundles will be included in the latest release. The process of selecting which datasets/bundles will be included happens outside of the Data Browser, but the Data Browser provides the starting point. We will be creating releases every 3-6 months.

Detailed Basic Use Cases

Prioritization Rubric
Each use case and functional requirement will be tagged with a priority level.
A - Must have
B - Should have
C - Nice to have

Features that are “Must have” must be included in the product in order for the product to be shipped. Features that are “Should have” are ones that are necessary for smooth functioning of the product. We will strive to have as many of the “Should have” features as possible, but accept that some may not make it in and will require workaround.

Features that are “Nice to have” are ones that we will add if we have extra time after “Must haves” and “Should haves” are completed. They will likely be incorporated after the initial production release.

Use Case: Query Files

Priority: Must Have (A)
Primary Actor: HCA researchers
Brief: Researchers want to be able to query for a subset of data in the Atlas that match certain attributes based on metadata provided about the donor/organism, sample, assay, and analysis.

- **Query using a fixed list of core facets based on metadata.** For our initial version, we will have a limited set of core metadata fields that are available for querying. In a future version, we will enable querying on many or all metadata fields.
- **Query based on current data (default) or by release version.** Users by default will be querying all current data. They can optionally select only data from a particular release.

Use Case: Query Biomaterials (Specimens)

Priority: Must Have (A)
Primary Actor: HCA researchers
Brief: Researchers want to be able to query for a subset of biomaterial (specimens) in the Atlas.

Use Case: Query Projects

Priority: Must Have (A)
Primary Actor: HCA researchers
Brief: Researchers want to be able to query for a subset of projects in the Atlas. Projects are a grouping of related datasets that have been submitted by a researcher or lab. As part of the ingestion process, submitters associate their submission with a project.

- **Query using a fixed list of core facets based on metadata.** For our initial version, we will have a limited set of core metadata fields that are available for querying. In a future version, we will enable querying on all metadata fields.

- **Query based on current data (default) or by release version.** Users by default will be querying all current data. They can optionally select only cells from a particular release.

For the HCA PIlot Release we propose to develop a basic version of the Projects tab comprised of the project listing and the project detail view and to conserve effort for other areas of the data browser such as the shopping cart, naming/creating/sharing collections, refining the current project, specimens, files view on the data browser, and working towards the search/collections handoff feature to third party portals.

View Projects List

Project Listing Mockups

When the user selects the project tab, the application displays the paginated list of projects containing data matching the selected facet terms with the following columns:

a. Project Name/Title
b. Organ
c. Experimental Approach
d. Species
e. Diseased
f. File Types
g. Donor Count
h. Estimated Cell Count

View Project Detail

Project Detail Mockup

When the user selects a row in the projects list the project detail/entity page displays showing:

.

1. The project tab shows:
    a. Description
    b. Contact
        i. Contact Name
        ii. Contact Institute
        iii. Contact email for the project.
    c. List of publications with links.
    d. Author List
    e. Collaborating institutes
    f. Download Project LinkLink

Download Project Manifest

1. Selecting a download project link on the project detail page selects the current project as facet and takes the user to the files tab where they can refine their search

Use Case: Advanced Query

Priority: Nice to have (C)
Primary Actor: HCA researchers, data release curators, red portal developers
Brief: Researchers and curators want to be able to query the metadata in the Atlas in an unconstrained way. This includes searching beyond the limited set of metadata fields represented by the faceted Data Browser. At the same time, researchers and curators will want to perform more sophisticated search logic including joins and complex booleans. Ideally, this can be done with a language that is familiar to them, such as the ubiquitous SQL language. Similarly, red box tertiary portal developers may find this service useful in building their own analysis and visualization portals.

Use Case: Download manifest of IDs

Priority: Must Have (A)
Primary Actor: HCA researchers
Brief: Once a researcher has queried and identified the subset of HCA data they are interested in, in order to access that data the Data Browser can create a downloadable “manifest” file that the researcher can use with the HCA client (CLI) to access data from the DSS (e.g. download files to their system). The Data Browser will document how to get the client, set it up, and configure it with a credentials token (optional) obtained after login. Search result manifests, such as the sample below, can then be used to access files. These are created when a user performs a faceted search in the Data Browser.

```
Program      Project      Center Name  Submitter Donor ID  Donor UUID   Submitter
Donor Primary Site  Submitter Specimen ID      Specimen UUID Submitter Specimen Type
```

```
Submitter Experimental Design     Submitter Sample ID Sample UUID   Analysis Type
Workflow Name Workflow Version    File Type     File Path     Upload File ID      Data
Bundle UUID   Metadata.json
Treehouse     Expression Analysis R      THR14  THR14_0320
027a0bd9-5df8-5ab3-815f-af42b0909f6c    BTO:0000042   THR14_0320_S01
4cde4028-4112-5996-8e28-a0f806916932    Primary tumour - other     RNA-Seq
THR14_0320_S01      3398874d-0f93-5737-bf19-32a36c9191dc    sequence_upload
spinnaker     1.0.0  fastq.gz      THR14_0320_S01_R1.fq.gz
8fba946d-183c-5d53-9c71-e50cde2ad45e    6ae93c4b-cf7a-56da-bcff-46597bd45fc5
54101d9b-48d1-54ae-8776-5a85eee1e861
```

Figure: a sample download manifest from http://ucsc-cgp.org

Use Case: Data Browser Logging

Priority: Must Have (A)
Primary Actor:
Brief: The Data Browser will generate logs (including access logs), and submit them to the common logging system.

Use Case: User Login

Priority: Must Have (A)
Primary Actor: HCA researchers (downloaders, submitters), DCP/DCC staff (developers, release curators, etc)
Brief: The Data Browser will include a login feature most likely using OpenID Connect for user authentication (the goal here is to use that authentication system, and possibly authorization system but that's TBD, as the rest of the DCP).

There are a variety of primary actors here and different reasons one would want to log in based on the user role and task at hand. These include:

- HCA researchers - downloaders: By default, all data in the Human Cell Atlas is open and does not require login to access. However, we will provide the option to login which will allow the user to access future premium features such as search result saving/sharing that will require information to be associated with the user's account.
- HCA researcher - submitters: users that are submitters may have additional information show up linking data in the Data Browser with their submissions. This is TBD but possible.
- DCP staff - developers: by logging in the user identity is established. This means he or she can generate a token for the API/client to access data in the DSS. Developers with an admin role may have additional abilities such as viewing non-public dashboards and admin features developed using a web UI for the DSS.
- DCP staff - release curators: staff with a release curator role will be able to perform a release through this site.

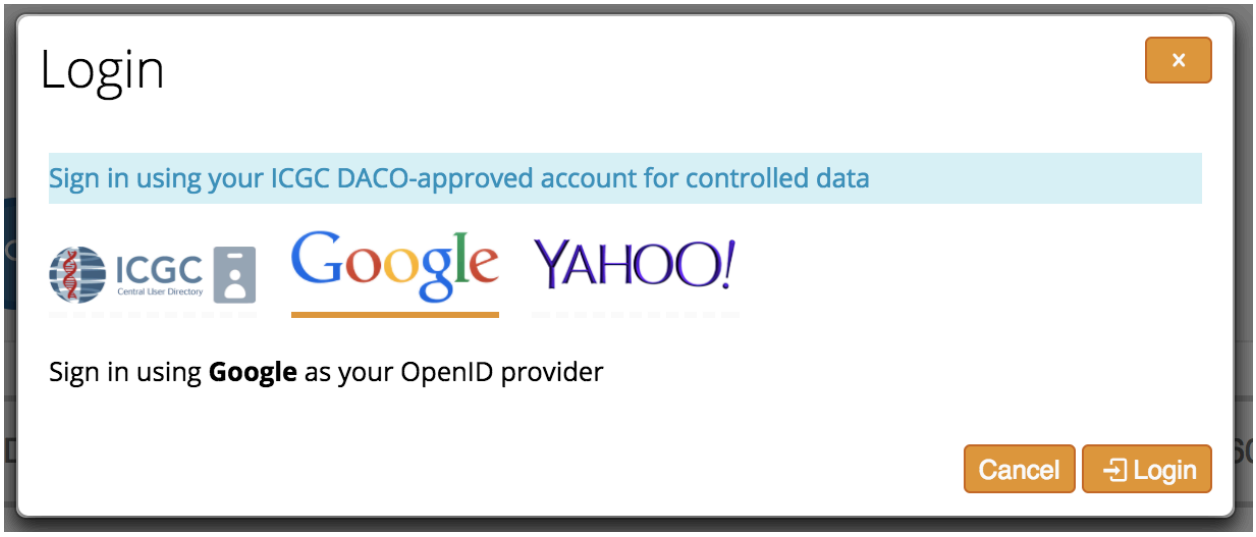


Figure: The OpenID Connect-based login for the ICGC Portal. A similar Web-UI would be used for the Data Browser.

This login functionality may seem to be optional for the Data Browser but systems like the DSS do not have web UIs (at least currently). So the Data Browser is a natural place to consolidate the user web interface and provide a UI for authenticating and generating tokens for use with the DSS.

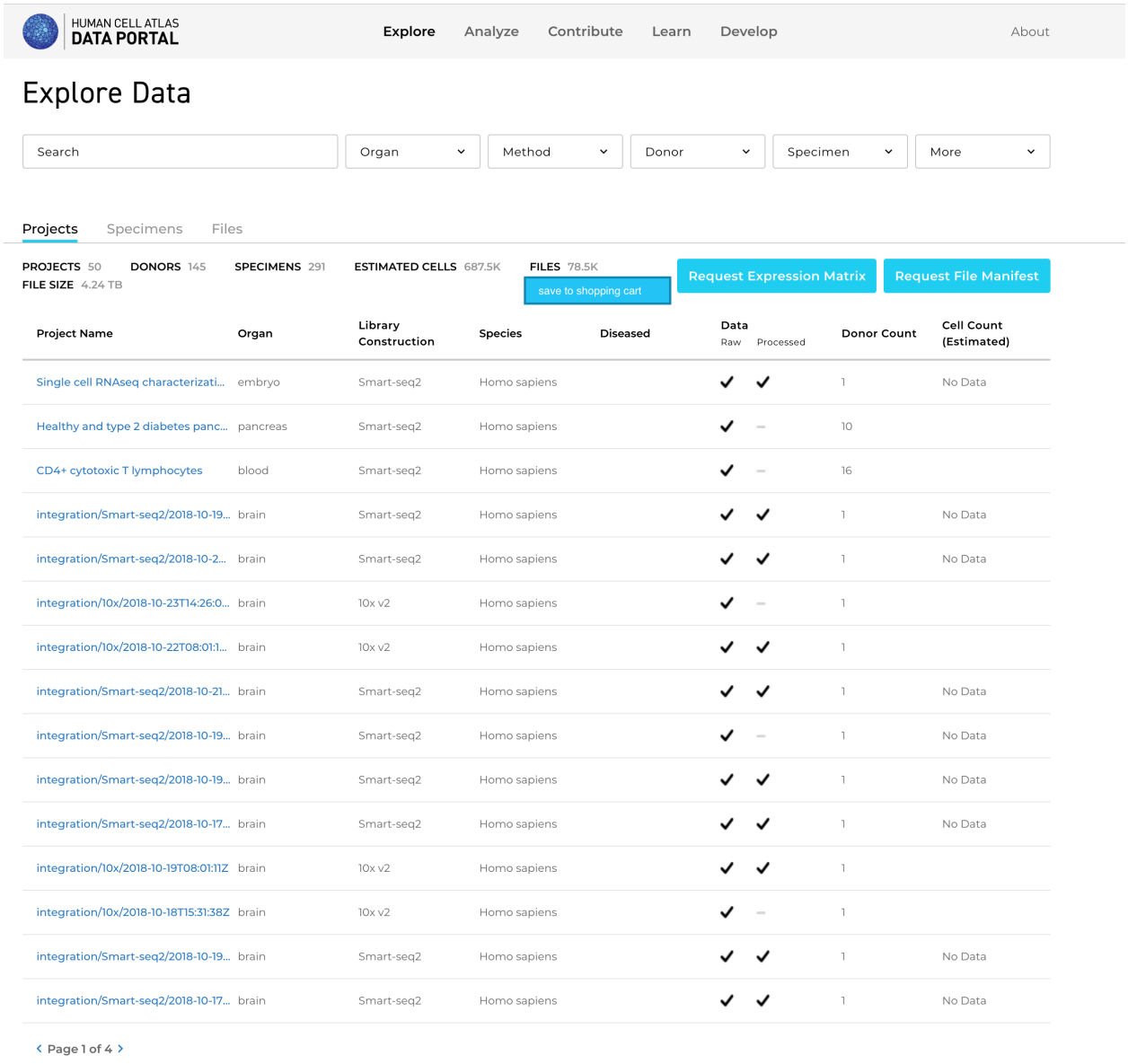
Saved Searches, Shopping Cart and Handoff

Jira link for more information.

Use Case: Putting items in a Shopping Cart

Priority: Must Have (C)
Primary Actor: HCA researchers
Brief: Researchers will be able to save and share results from a query run in the Data Browser (note: not the same as a shared saved query because the results of a query will change! A saved set of results is static). This is akin to what we are calling the “personal release” -- a selected subset of data in the Atlas that has been curated by an individual. These personal releases should be durable and shareable with other researchers of the creator’s choosing.

One idea seen on the GDC site that could be quite applicable here is the idea of a “shopping cart”. When browsing different entities (projects, donors, files, etc) you can opt to select one or all of you search results to a shopping cart. We can think the future about a shopping cart for different entity types, e.g. you could add projects to a projects shopping cart and that bears no relationship to your files shopping cart. However, for simplicity, I think the shopping cart should focus on files for now. So when you’re browsing search results in the portal:



And you click “save to shopping cart” what’s happening here is the files, bundles, versions, projects associated, specimens associated, etc are all saved in your shopping cart. And we can render this shopping cart like we do the files tab, with info about files, bundles, versions, projects, etc.

File Name	Specimen Id	Organ	Organ Part	Library Construction	Species	Age	Sex	Diseased
f3e971f1-d7d4-...	embryo_WAe0...	embryo	blastocyst	Smart-seq2	Homo sapiens		male	
edc9e39d-ad3e...	embryo_WAe0...	embryo	blastocyst	Smart-seq2	Homo sapiens		male	
SRR5175067_2.f...	embryo_WAe0...	embryo	blastocyst	Smart-seq2	Homo sapiens		male	
6b456c47-87b5...	embryo_WAe0...	embryo	blastocyst	Smart-seq2	Homo sapiens		male	
01607d19-40eb...	embryo_WAe0...	embryo	blastocyst	Smart-seq2	Homo sapiens		male	

So, we start with a files-based shopping cart but, since we're tracking project, specimen, etc in the future we can have project, specimen, and files views of the shopping cart. For now, though, a files-focused shopping cart makes sense since our primary activity is making collections of files for download or use through an API.

Once a user is happy with their shopping cart (they can add/remove individual entities) they can decide to make a collection so they can click a button in the shopping cart that it will now make a named collection and empty the cart. Once they have made a collection they can share it, rename it, or delete it but they are not able to alter its content (see Granular Editing of a Collection below, versioning would be a nice to have feature that would allow for "editing" from a user's perspective).

Viewing a collection or a shopping cart, a user can directly click on a file link to download the file. At any time they can use the "Request File Manifest" or "Request Expression Matrix" for those files. Also note, a user can download from a shopping cart, without creating a collection. A user can hand off from a shopping cart, without creating a collection.

See Also

See https://ucsc-cgl.atlassian.net/browse/AZUL-64

See https://ucsc-cgl.atlassian.net/browse/AZUL-318

The API spec from DNAstack:
https://github.com/DataBiosphere/azul/blob/97c57a52a0ae1ac8dfd51d92e4a20f532290bec2/rfc/cart-api-v1.md

See also: Collections API Functional Specification and Use Cases. The goal is to have the Orange Box Data Browser use the common Collections message format and use the Collection API services.

Use Case: Collection Management

Priority: Should Have (B)
Primary Actor: HCA researchers, Red Boxes
Brief:

As a user I wish to manage my Collections by being able to:

- create multiple Collections (via the Shopping Cart)
- view my list of Collections
- delete a Collection (just the Collection, not the underlying data)
- Get a URL to a collection (see Collection Sharing)
- Hand off the collection to another portal for visualization

Use Case: Collection Sharing

Priority: Should Have (B)
Primary Actor: HCA researchers, Red Boxes, staff (curators), scientists.
Brief: Scientists/curators will share a collection saved as a manifest or a URL with collaborators.

As a user I wish to manage my Collections by being able to:

- share a Collection with another user (via a url)
- Get a manifest file for the Collection and send that file around

Use Case: Granular Editing of a Collection

Priority: Nice to Have (C)
Primary Actor: staff (curators), scientists
Brief: Scientists/curators will edit their collection at a very granular level. Actions could include taking out entr(y/ies), or adding (annotating) extra information to their collection. This means scientists need to be able to edit their collections in a human friendly way.

Keep in mind, editing a Collection actually makes a new version of that collection. So the UI will need to show previous versions of that collection and make those accessible for viewing, manifest generation, handoff, etc.

Use Case: Remembering the purpose of the Collection

Priority: Nice to Have (C)
Primary Actor: staff (curators), scientists

Brief: Scientists/curators will potentially save many iterations of a single manifest. Scientist/curators will use information present in the manifest/collection to infer the purpose of that particular manifest/collection. So add the ability to have a name and description for a Collection.

Use Case: Comparing Collections

Priority: Nice to Have (C)
Primary Actor: staff (curators), scientists
Brief: Scientist/curators will want to see how a collection might differ from another one. For example, two scientists might want to see how their own collection might be different. Another example could be of a scientists/curators looking at how various snapshots of their personal collections saved as manifests have evolved over time.

Look at the functionality on the ICGC portal for a good example of what scientists would want:

https://dcc.icgc.org/analysis/view/set/46ec3ea7-a57c-430b-8be0-ef4dd3c2898d

Collections Use Case Discussions

From March 2018 Hinxton meeting: Breakout Topic 4: Release/collection APIs:
Use cases

- As a researcher I want to do a search and share the results with my colleagues without telling them how to do the search (which could change over time)
- By saving it as a collection the list is fixed and it doesn't change over time
- User would like to annotate the collection with information (collection notes)
- User would like to name the collection
- User would like to have a description of the collection
- User would like to version the collection
- User might want to create a collection from multiple searches
- Users would like to have a comment thread for discussion
- Users would like to save history of how the collection was created (good for release)
- Users would like to use a collection to document the input or output of their studies
- Users would like to give their collections to tertiary portals
- User wants to be able to start with a collection and add to it (edit it)
- User would like a mechanism to cite a dataset (with some form of a collection handle?)
- As a user I would like to share editorial rights of my collection to another user
- User would like to be able to script the creation of a collection (be able to re-run later)
- as a user I would like to specify membership in a collection as a search criteria

Definitions:

- Collections are arbitrary groups of entities from the DCP
- Entities are: DSS data files fully qualified, project, biomaterial, process, protocol

Notes:

- Is there anything we would like to do with Collections that we would not want to do with releases? Or vice versa?
- Why not just send around a list of UUIDs?
- Need to ask users how useful this is

Notes from ad-hoc design discussion:

- dynamic collections in the orange box are stored in dynamodb until it is saved. Persistent collections are saved in S3 in the Blue Box

Summary from Hannes in Slack:
This may look like I'm hijacking the manifest topic with collections but bear with me. I think it'll become apparent that collections and manifests are related.

@akislyuk, @kozbo, @caaespin,@tsmith12 and I discussed the manifest/collection issue this afternoon. We agreed that OrangeBox should let its users create and manage their own collections (for example backed by DynamoDB as in @briandoconnor's prototype). They can then either save those collections to BlueBox as a durable, versioned and replicated object on blob storage. They can also export a collection as a manifest for the purpose of sharing or processing it externally. I think we should see manifests as serialized/exported incarnations of collections. For example, the manifest for the CLI (see below) could use the same format as the durable collections in BlueBox.

@akislyuk thinks that batch downloads via the CLI should not be done by creating and downloading a manifest and feeding it to the CLI. He'd rather save the collection as an entity in BlueBox or OrangeBox and hand the CLI a reference to that entity. I think that batch downloads via a manifest would be easier to implement and more tractable for users by letting them slice and dice it before feeding it to the CLI. However, I also see the point that this becomes intractable and inconvenient with large collections. So this remains a subject of ongoing discussions.

We did not have a chance to discuss other use cases for releases / collections. The use case, for example, that mentions that a collection may consist not only of bundles, but of entities of other types, too. Or the use cases around community features such as collaboratively creating and commenting on collections. Or the use case about collection provenance, i.e tracking every modification to a collection.

Use Case: Query Sharing

Priority: Should Have (B)
Primary Actor: HCA researchers, Red Boxes

Brief: Users will use the Data Browser to find a particular subset of HCA data that is useful for them. This feature will allow them to share their query with others using a simple, shortened web sharing link (such as what's present in the ICGC portal). The URL just points back to the Data Browser such that a user clicking on the link will be taken to the facets selected and deselected as they were when the sharing link was created. There's no guarantee that search results will be the same over time.

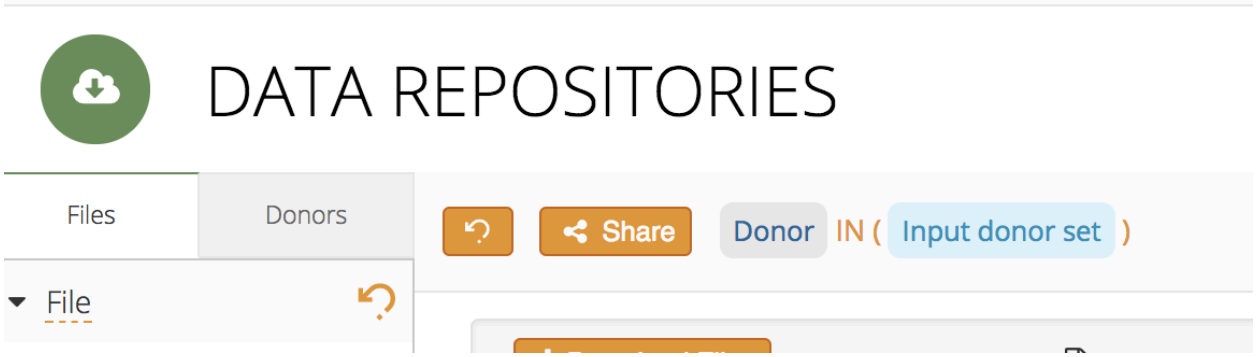


Figure: search result sharing in ICGC

For example, a query is encoded in the GET string:
https://staging.data.humancellatlas.org/explore/files?filter=%5B%7B%22facetName%22%3A%22biologicalSex%22%2C%22terms%22%3A%5B%22male%22%5D%7D%2C%7B%22facetName%22%3A%22organ%22%2C%22terms%22%3A%5B%22embryo%22%5D%7D%2C%7B%22facetName%22%3A%22laboratory%22%2C%22terms%22%3A%5B%22Molecular+Atlas%22%5D%7D%2C%7B%22facetName%22%3A%22fileFormat%22%2C%22terms%22%3A%5B%22matrix%22%5D%7D%5D

I think what we can do to make this a little easier to deal with, something like:
http://bit.ly/2q5oETk

That shortened URL is way easier and more reliable to send around.

In the future, we can think about saving these queries much like a shopping cart can be shared as a collection. And a user can come back to the portal, view and manage their shared queries etc. However, I think the feature of making collections from a shopping cart is way more important. In the short run a user can use their browsers bookmarking feature to accomplish the use case of saving queries and managing them.

Use Case: Red Box Handoff

Priority: Should Have (B)
Primary Actor: Red boxes

Brief: Users will use the Data Browser to identify a subset of data that they want to analyze using a Red Box. We need to make it easy for a user to handoff a selected list of bundles, files, donors, assays, etc to a Red Box of their choosing, such as running workflows in the Firecloud system. This will likely need to be achieved by passing a manifest of IDs, vs passing a query since the query results may change over time. We will need an API that we can provide Red Box developers so they can advertise themselves to the orange box and get specific about the data types they support handoff for. For example, a visualization tool may expect specimen IDs while an analysis portal (FireCloud) will expect File IDs.

As a user I can create a Collection (as above) and pass a reference to the Collection url to a selected "Red Box" (portal analysis application) for further analysis (see https://app.zenhub.com/workspace/o/humancellatlas/data-browser/issues/6).

Is this a collection in the blue box?

Is this something else using the same Collection JSON format?

Use Case: FireCloud Handoff

Priority: Should Have (B)
Primary Actor: Red boxes
Brief: Since we have this already working with Boardwalk, can we get the FireCloud launch working with the existing BDbag functionality before the generic solution on Red Box handoff?

Use Case: Matrix Handoff

Priority: Should Have (B)
Primary Actor: Red boxes
Brief: As a user I can create a Collection (as above) and pass the Collection url to the Matrix Service for further analysis.
Phase 1 implementation: Pass a file manifest to the Matrix Service which parses out the bundle uuids and then extracts the matrix format files from the bundles.
Long term implementation: Calvin Nhieu (Matrix Service dev): I think a list of bundle uuids will remain to service simple and smaller requests, and for large requests a blue box collection would be a more accessible solution than a URL.

Appendix

Metadata v5 and Entities/Facets

See https://github.com/HumanCellAtlas/metadata-schema/tree/v5_prototype and https://github.com/HumanCellAtlas/metadata-schema/tree/v5_prototype/schema_test_files for specific example docs. See:

- https://github.com/HumanCellAtlas/metadata-schema/blob/v5_prototype/json_schema/type/project/project.json
- See https://github.com/HumanCellAtlas/metadata-schema/blob/v5_prototype/spreadsheet/v5/Empty_template_v5.0.0_spreadsheet.xlsx
- This is a really nice overview of the schema (see Clay if you need access): https://www.mindomo.com/mindmap/9798925e255141b088b8db4dbbb9fdb1

Questions

- How are biomaterials linked together? I don't see a parent ID field
- When do we get sample v5 data bundles in the staging system?

Entities

In the first pass we focus on Projects and Files, a future version of the portal can add additional entities in this order.

- **Projects**
- **Files**
- Protocol
- Process
- Biomaterial

Facets

FACETS V5:

>Facets for Biomaterial

biomaterials|content|is_living -> "checkbox"
biomaterials|content|biological_sex -> "checkbox"
biomaterials|content|development_stage|text -> "checkbox"
biomaterials|content|genus_sepecies|text -> "checkbox"
biomaterials|content|biomaterial_id -> "search box"
biomaterials|content|biomaterial_name -> "search box"

>Facets for Project

project|content|project_core|project_id -> "search box"
project|content|project_core|project_title -> "checkbox"

project|content|contributors|contact_name -> "search box"
project|content|contributors|email -> "search box"
Project|content|publications -> could be very useful for another data browser or -> we could have publication browser! More to come.
Insdc_project
geo_series
array_express_investigation
insdc_study

>Facets for Process:

NOTE: The schema for process.json is still not out at the moment. So the lines below here will be guesstimates until such time the metadata group releases the v5 metadata schema
Process|content|analysis|
Process|content|biomaterial_collection
Process|content|imaging
Process|content|sequencing

> Facets for File:

File type

For comparison, here are the facets that CIRM uses in their browser:
Lab
Dataset
Assay
Cell type <--This one is hard.
Sex
Healthy/normal vs diseased
iPSC vs ES
Enrichment in exon/intron/etc.
Organoid vs. primary
Organism (human, mouse, etc.)
Single cell vs Bulk
Single cell methodology (Drop-seq, 10X, Fluidigm, etc.)
Read size?
Access (protected (must request access), embargo (will become open), and open)
Date (submission, added, other?)
Age
QC metrics?
Size? Max/min

To Move to Matrix Service Spec

Use Case: Download concatenated matrix (non-transformative)

Priority: Must Have (A)
Primary Actor: HCA researchers
Brief: Once a researcher has queried and identified the subset of HCA data they are interested in, they can download a single expression matrix for all the cells. This also holds for QC metrics. Note: we need to make it clear to users that this is non-transformative.

- **Download matrix of expression counts.** The user is able to download a single big expression matrix that aggregates the individual expression matrices for the cells in a non-transformative way.
- **Download matrix of QC metrics.** Similar to expression, the user is able to download a single matrix of QC metrics for the selected cells.
- **Download matrix of metadata.**

Open questions:

- What is the file format?
- Asynchronous vs real-time?
- Which metadata should be included?
- Since this is non-transformative, what other information will the user need in order to normalize the data themselves?

Use Case: Download concatenated matrix (transformative)

Priority: Should Have (B)
Primary Actor: HCA researchers
Brief: Once a researcher has queried and identified the subset of HCA data they are interested in, they can download a single big matrix that aggregates the individual expression matrices for the cells where the data has been normalized (transformed). Note: orange can interact with green to get green to do the transformation and store the needed output in the blue box for orange to work with.

Use Case: White/grey/black listing of cells based on quality data