ClinGenDB Infrastructure

This page provides information about ClinGenDB infrastructure and its roles within the ClinGen system.

ClinGenDB infrastructure is not accessed directly by most curators or end-users. The curators and end-users who access ClinGenDB directly generally do that using the UIs provided by GenboreeKB for the purpose of examining ClinGenDB document models providing suggestions for their improvements. GenboreeKB UIs may also be used for curation/editing of genes, variants, and any other document collections in ClinGenDB but are not specifically optimized for such use. ClinGenDB infrastructure (Genboree REST APIs specifically) support development of UIs that are specifically optimized for curation and for access by various audiences via the ClinGen Portal. The APIs are also used by other applications ClinGen consortium members may develop and for interfacing with other ClinGen components as outlined in the Use Case section below. In addition to database functions, APIs provide additional web services and are the cornerstone of the ClinGenDB infrastructure.

ClingenDB FAQs

Question 1: How do I get access into ClinGenDB so I can search and look around?

Answer:

- To access ClinGenDB for interactive use via the GenboreeKB UI or for use by a computer program via REST APIs:
  - You will first need to establish Genboree credentials (user name and password) at www.genboree.org.
  - You will then need to send your Genboree user name to Xin Feng (xinf@bcm.edu) with a request to be granted access to ClinGenDB. Xin will then enable your access to GenboreeKB.
  - Login at GenboreeKB: http://genboree.org/genboreeKB/projects . This step enables Genboree to link your Genboree account to the GenboreeKB account. You will still not see ClinGenDB projects.
  - Send another email to Xin Feng (xinf@bcm.edu) saying that you logged into GenboreeKB successfully.
  - Xin will then add your account to the member list of the ClinGenDB project in GenboreeKB.
  - After receiving a confirmation email from Xin, you will be able to access ClinGenDB data collections via both the GenboreeKB UI and the APIs.

- Access is controlled per group, following well established practice. You will be granted access by being added as a member to a group that has access to ClinGenDB. Group members may have different roles (administrator, author, subscriber), each role having different authorizations (adding new group members, read/write, read-only respectively).
  - Once you are a member, you will have interactive access to current ClinGenDB document collections via the GenboreeKB UI.
    - By clicking at http://genboree.org/genboreeKB/projects you will see the public projects in GenboreeKB. (GenboreeKB is a Redmine plugin – what you are seeing are Redmine project pages.)
    - Sign in by clicking on the button in the top right corner. Provide your Genboree credentials. You will now be seeing the projects that your group(s) have access to, including the ClinGenDB project.
    - By clicking on the ClinGenDB project, you will get to the ClinGenDB project page (implemented using Redmine project page).
    - To access the ClinGenDB document collections via the GenboreeKB UI, click on the GenboreeKB button (to the right of the “Overview” button).
    - You should now be seeing the ClinGenDB collections in the GenboreeKB UI, including “acmg-lit” collection that contains information about variants on the ACMG 56 gene list and a number of versions of gene document collections linking genes to specific conditions. Select a collection to work on using the “Collections” pulldown.
    - The buttons on the right of the “Collection” pulldown allow you to download, upload, and query documents within the collection.
    - Query of the documents within a collection, click on the “Query Collection” icon. Each queryable property is defined as “Identifier” within the Document Model. Select a queryable property for the Document Collection of interest and perform

a query. A list of zero or more single-line “Views” of matching Documents appears. Click on the desired Document to open it for full viewing and possibly for editing. (Example: query the “acmg-lit” collection using the “DocumentID” property using “a” as a query in the “keyword” mode. At least one document should match that query.)

- Views and Queries are also first-class objects in GenboreeKB -- they can be managed via GenboreeKB UI and accessed via the APIs. (Yes, they are also represented as Documents!)
- For access to ClinGenDB via REST APIs, see the answer to Question 7 below. API access uses the same credentials and group-based authorization as interactive access. In fact, GenboreeKB UI (a Redmine plugin) communicates to the Genboree server exclusively through the same APIs. Consequently, you should be able to write UIs similar to GenboreeKB UI yourself.This is how curation interfaces may be written.ClinGen portal and other components of the ClinGen system may ClinGenDB via the same APIs.

Question 2: What data have been loaded and with what schema?

Answer:

- In GenboreeKB, a schema is referred to as Document Model. Each Document Collection within GenboreeKB requires a Document Model. The user may create one or more Document Models and instantiate corresponding Document Collections. The GenboreeKB UI may be used to edit and browse Document Models, create Document Collections, and search, browse, create and edit Documents within Collections. The same operations may also be performed programmatically via the REST APIs.
- REST APIs are also immediately available to programmatically access each newly created collection. This allows the users a full programmatic control of their schemas and data.
- Using these features, we created three types of document collections for ClinGenDB: (1) variant curation; (2) gene curation; and (3) protein domain information.
    - Variant curation documents (Document Collection named “acmg-lit”) include data such as variant ID from ClinVar, Locus-Specific Database (LSDB) links and few additional annotations such as allele frequencies from 1KG and other population databases.
    - Several consecutive versions of gene curation document models have been developed based on user feedback. Corresponding collections are named “GeneDiseasePair-2”, GeneDiseasePair-3.1”, “GeneDiseasePair-3.2”, and “GeneDiseasePair-3.3”.
    - “GeorgetownUniProt-0.1” collects algorithmic annotations of protein domains by our collaborator Peter McGarvey at Georgetown University. The protein domain information from this document models will be used as evidence to support curation of variants that fall within the annotated protein domains.

Question 3: What data do you plan to link out to? How do you decide what to store in ClinGenDB and what to link out to?

Answer:

- As a general rule, if a piece of evidence is used to infer an assertion of pathogenicity of a variant or to establish a link of a gene to a disease, ClinGenDB should ensure that the piece of evidence will be accessible for examination. This generally means storing the summary of evidence within the ClinGenDB document, with a link to the outside source.
- Much external data needs to be examined in the course of the curation process or for the purpose of providing access to various algorithms via the APIs. Not all of this will end up being used as evidence. Services and approaches that “vacuum” the contents of other databases rather than linking need to have their local/loaded contents updated when the remote source database is updated. By linking to data in select remote databases, the changes at the remote service are always available.
    - Thus in certain regards, the vacuum approach is less ideal in the face of changing data, on top of failing to leverage the distributed nature of data and requiring resources to hold the redundant data.
    - Which data to load/copy and which to link to depends on the specific project. For example, Variant documents may link to LSDBs. Gene curation documents may link out to pubmed IDs, but capture select key article metadata [this can also be done automatically by GenboreeKB at document save-time]; Protein documents may link out to the UniProt page; content from unreliable links may be cached locally.
    - In general, a remote database with a well-defined resource representation, large number of stored resources, and which is frequently updated is a good candidate for linking.
    - But at the same time, sufficient context information—possibly select resource metadata—should be captured for the document, and/or information that will be useful for searching, partitioning, or even mining the documents should be considered as candidates for loading/copying into the document. (More sophisticated miners would follow links to the uniform and well-defined representations available at key remote databases.)

Question 4: What data sources are planned to be regularly updated and with what frequency?

Answer:

- We would need input on this from the users.

Question 5: If a curator adds new data how does the system know to link it up with other information already in the system?

Answer:

- It is possible to search certain properties defined in the model—and non-items—and see the list of document “hits”. What fields from the document are used in the “hits table” is defined via Views, which can be defined by the user using the API. These [non-items] properties in the View should help users choose documents of possible interest for their match. Do consider having multiple Views that give users some options such as: (a) light/key-info-only View vs (b) more detailed Views that emphasize different kinds of information for folks interested in different aspects of the doc. Then view those documents by clicking them in the “hit list”. Once the key document(s) are found, edit them and save your edits.
- What is not yet available is more sophisticated nested Boolean search, nor the ability of the basic search to consider items in an items list (or sub-docs in a sub-doc list, if you prefer).

Question 6: What are the JSON schemas you have implemented and how do we find them?

Answer:

- The schema(s) are project-specific and are represented as Document Models in GenboreeKB. GenboreeKB allows the projects such as ClinGen to define the Document Models for variant curation, gene curation, proteins, etc.
- GenboreeKB stores Documents in Document Collections; collections are “homogenous” in that they will contain documents which conform to a specific Document Model. Documents and Document Models can be downloaded or uploaded in JSON or Tabbed formats using GenboreeKB UI.
- You can also view, interact with and edit Document Models using the *View Model (Tree)* option in GenboreeKB UI.
  - Note there are collapse/expand all toolbar buttons to ease exploration. These are also available at the –per-branch level to fully expand or collapse specific branches.
  - For example, the variant curation model for the ACMG 56 set of genes (“acmg-lit”) and versions of gene curation document models (e.g., “GeneDiseasePair-3.2”) are available.
  - Each Document Model (as any other Document in GenboreeKB) is also available via a unique URL.

Question 7: Is there documentation for using the REST API? Do you have example scripts available?

Answer:

- GenboreeKB UI (a Redmine plugin) communicates to the Genboree server exclusively through the same APIs. Consequently, you should be able to write UIs similar to the GenboreeKB UI yourself.
- APIs and their documentation are currently under constant development -- more online documentation about the API in particular will be added in the coming weeks. A comprehensive, still incomplete (not including the most recently added GenboreeKB functionality) is here: http://www.genboree.org/java-bin/showHelp.jsp?topic=restAPIOverview
- Instructions on how to operate on users, groups, and projects can be found here: http://www.genboree.org/java-bin/showHelp.jsp?topic=restUniformMethods
- ClinGenDB API tutorial and sample scripts are available here:
  - https://github.com/clinvar/apidemo
  - More generic information about Genboree REST APIs (on which ClinGenDB APIs are built) is available here:
    - http://www.genboree.org/java-bin/showHelp.jsp?topic=restAPIOverviewAPI access uses the same credentials and group-based authorization as interactive access.
- Brief summary of operation and corresponding scripts:

Purpose	Script for the purpose	Arguments
Create a new collection for storing documents in ClinGenDB	https://github.com/clinvar/apidemo/blob/master/createCollection.rb	#1: Name of new collection #2: JSON file describing document model $
Upload a document to an existing collection	https://github.com/clinvar/apidemo/blob/master/uploadDoc.rb	#1: Name of the collection #2: JSON doc to be uploaded
Search/Get a specific document in collection	https://github.com/clinvar/apidemo/blob/master/genericQuery.rb	#1: Property to search #2: Value to match
Update specific record in collection	The same one for uploading a doc	The same one for uploading a doc
Delete specific record in a collection	https://github.com/clinvar/apidemo/blob/master/deleteDoc.rb	#1: Name of the collection #2: document ID
Retrieve records for queries using wild cards	Will be uploaded soon on github	Will be uploaded soon on github

$ Please looks at the answer of Question 2 for JSON schema

Question 8: How is the data and schema versioned?

Answer:

- Both documents and schemas are versioned using an internal versioning function.
- In the UI, you could get the history of a document using the "History" button located at the left side panel.
- Using the UI or the API, one can retrieve the version history list for a given document or schema and can view a previous version.
    - The UI additionally lets you “rollback” to a previous version of a document. This is effected by retrieving the previous version and then saving that as the latest version—and thus becomes part of the version history as well.
    - Rolling back schemas is not supported in the UI at this time, as the implication for documents in the collections can be very severe: you may have documents that don’t match the older schema. Due to such risks, model changes should be done in coordination with BRL staff at this time.
    - Here is one example of a versioned data model:
        - http://genboree.org/genboreeKB/genboree_kbs?project_id=acmg-apitest&showModelVersionsGrid=true&coll=GeneDiseasePair-3.2

Question 9: What is your plan for managing change as the data model is refined?

Answer:

- Schema change should be carefully managed within any system or project.
- Document models themselves are editable. Changes that do not disrupt the document model in any major way do not require data migration. Addition of new fields—new sibling fields or new sub-document fields—is a minor change and doesn’t invalidate existing documents.
- A major change would be rearrangement of subdocuments within a document. Such change affects the hierarchical structure of the document and can easily invalidate existing documents.
    - This kind of schema change would require a “migration” phase to migrate existing documents under the old model to versions that are valid in the new model.
    - We’ve currently been approaching this by performing the following steps: (1) editing the document model; (2) instantiating a new document collection with the new model; and (3) migrating the data into the newly instantiated data collection. Even the major change that involves migration may be performed via the APIs.
    - Other kinds of changes may or may not require migrations. Consider a value-type change for a given field from a regexp pattern to the more generic string type. Existing documents would be compatible with this regexp à string typing change. However, the reverse schema change—string à regexp—may invalidate some existing documents and should be approached more carefully.
    - As with any storage model, model changes should not be approach cavalierly. Because users, albeit sophisticated ones, can define their own models and create document collections, we currently do not allow *ad hoc* change of models for existing collections via the UI or via the API and will not until further checks are in place to prevent collection corruption / invalidation. Furthermore, model changes involving migrations may take some time. For this and other reasons GenboreeKB UI are contraindicated – the best way is to use the APIs.
        - Schema migration should be performed via the APIs because it does not incur the risks of using GenboreeKB UI. The old collection still exists, in the case of problems with the new model or migration.

Question 10: How are you using JSON-LD within your system?

Answer:

- This feature is planned but not yet implemented.
- The design of Genboree (including GenboreeKB) REST APIs and document-oriented data models put us close to compliance with the emerging Linked Data Platform 1.0 standard http://www.w3.org/TR/ldp/ , the first draft of which was released in 2014 by the World Wide Web Consortium (W3C). We intend to be compliant with this standard at soon after it reaches W3C Recommendation status. This standard refers to RDF serialization and JSON-LD.

Question 11: How are new collections added to ClinGenDB using the APIs?

Answer can be found at this link.

Question 12: How is a document validated in ClinGenDB using the APIs?

Answer can be found at this link.

ClinGenDB Overview Presentation

ClinGen Overview presentation given at the ClinGen Steering Committee Meeting in San Diego on October 23rd 2014: PDF.

Document Modeling in ClinGenDB

ClinGenDB is utilizes MongoDB database, a document-oriented database that is gaining wide adoption. Document-orient modeling methodology is illustrated here: PDF PPT.

ClinGenDB's position within the ClinGen System

ClinGenDB interfaces with a number of components within the ClinGen system as illustrated here: PDF PPT.

ClinGenDB Use Cases

Use Case 1: Data transfer, warehousing, and linking: PDF PPT.

Use Case 2: Gene curation PDF PPT.

Use Case 3: Variant curation: PDF PPT.

Use Case 4: Algorithmic annotation: PDF PPT.

Use Case 5: ClinGenDB Portal (in development)

Use Case 6: EHR and Patient Registries (in development)

ClinGenDB Tutorials

How to contribute my datasets into ClinGenDB?