CPIC DATABASE

STANDARD OPERATING PROCEDURE

Clinical Pharmacogenetics Implementation Consortium (CPIC)

February 2021

Version 1.0

Table of Contents

Database Procedures and Operations

Transparency and public accessibility

Transparency and public accessibility are key aspects of CPIC's mission to facilitate the implementation of pharmacogenetic testing into clinical care. The CPIC Pharmacogene Curation SOP describes the process used to generate the assertions made by CPIC experts about allele clinical function assignment, translation of diplotype to phenotype, and the evidence assessed. Information from this process is compiled in (1) the Allele Functionality table and (2) the table that maps allele function to phenotype terms that is created as part of the guideline manuscript process. We refer to the content of these tables as "data" in this document. The data are uploaded, stored, versioned and made accessible through the CPIC database (DB) and application programming interface (API) both of which are freely and publicly accessible. The data are also available through Excel files generated from the DB (see the "Data formats and nomenclature" section) and posted on the CPIC website.

To facilitate use of the DB by the public, a recorded webinar provides an overview of the DB and instructions on how to obtain the DB information by downloading the contents or accessing it through the API. Data and updates are stored in the DB as described in the "Data Preservation" section which allows the public to track when new or revised data are available. Updates also result in the creation of versioned Excel files, as described in the "Version control" section, posted to the CPIC website.

Version control

Data are version controlled. Any modification done to the source tables listed above requires an addition to the "Change log" for that table by a CPIC staff member. Each table contains a "Change log" with the date, description of modifications, and rationale. The change log is uploaded to the DB with each data upload. This information is available through the API. Tables updated as part of a new guideline or guideline update are uploaded to the DB when the guideline is published. Updates to tables that are not associated with a guideline publication (e.g. reassessment, correction) are uploaded to the DB as needed.

The structure of the data model that contains the data in the DB is also version controlled. These changes are tracked in the GitHub code repository used to manage and maintain the data structure. All changes are stored as commits in the repository and track the date, author, and description of any change applied.

CPIC SOPs are stored in a git version control system. This is used to track the date, author, and description of every modification that is made to the SOP. This git repository is made public through GitHub, a collaborative version control provider. SOPs are reviewed annually and every committed change to the document will contain a description of implications of the change to current procedure.

Data Preservation

The primary source of structured CPIC data is a PostgreSQL instance. The instance is hosted by Amazon Web Services (AWS) and managed by their Relational Database Service (RDS). This service manages the provisioning and a large portion of the maintenance of the PostgreSQL software. This service runs on AWS cloud services which ensures a high degree of availability and stability. The AWS system also includes security tooling to restrict database access to only approved applications and maintainers. These restrictions are done via the AWS Identity and Access Management (IAM) system, network firewall restrictions, and database-level role security.

The DB is configured for automatic backups. The RDS service is configured to take daily snapshots of the database and retain 7 days' worth of snapshots within the RDS system. Additionally, DB copies are exported and saved when the codebase that manages the DB has a new release version. DB exports are stored with the code release which ensures retention of all structured content needed to manage the CPIC system.

Periodically exported files derived from the DB that describe allele function and phenotype translation are stored in the AWS Simple Storage Service (S3) service for long-term storage and availability to the public. These files are also copied to separate servers on the Stanford University campus for off-site redundancy. Records are kept in the database itself when these periodically exported files are created and uploaded for general availability. This creates an audit log and enables the public to track when new data are available through the API and S3 service.

The DB does not directly link to secondary databases and therefore does not necessitate additional procedures to recognize changes to a secondary database.

CPIC has contingency plans in place for temporary service outages and cessation of the project. In the unlikely scenario of temporary outage of the AWS system, the CPIC technology stack can be migrated to a different hosting provider. This is possible due to all CPIC services relying exclusively on open source software and all information needed to set up the CPIC services being redundantly copied to other services like GitHub and other hosting platforms such as Stanford University's IT infrastructure. CPIC software is not dependent on any particular hosting platform. CPIC exported files can also be distributed outside of the S3 system via GitHub hosting services.

In the case of cessation of the project, CPIC will publish all relevant data and code to the Zenodo service for long-term, cold storage. Zenodo is committed to open science and abides by FAIR (Findable, Accessible, Interoperable, Reusable) principles. Their system is widely adopted by the data science community and offers outstanding infrastructure for cataloguing data. Once the data are published to Zenodo there will be a stable, citable resource available for public access for the foreseeable future.

Security and Privacy

No personally identifiable or protected health information is evaluated as evidence for CPIC guidelines, so no such data are stored in the DB or used in the CPIC Pharmacogene Curation SOP in any way; HIPAA, privacy, and security training are not applicable to the CPIC DB.

Data formats and nomenclature

The canonical location for all data created through the CPIC Pharmacogene Curation SOP is the DB. The data are split between multiple tables that are defined by SQL scripts in a publicly available code repository. The scripts define the table organization, the specific formats of each property, acceptable values, and the precise relations between tables. The data can be accessed directly from a copy of the DB, via the API, or from Excel sheets generated on demand. The API uses a standard RESTful interface and can either serve JSON or CSV data depending on the request from the user. The generated Excel sheets use standard Excel data formatting functions. The sheets are generated by code defined in the previously mentioned code repository and will retain a format consistent between releases. Any changes to format are tracked in commits to the code repository with author, timestamp, and description metadata.

Data Quality

Metadata

The CPIC Pharmacogene Curation SOP details the process for evidence inclusion and assessment. All evidence sources (e.g. PMIDs) and evidence summaries are found in the Allele Functionality table as described in the CPIC Pharmacogene Curation SOP. These evidence sources and summaries are stored as metadata in the allele table of the DB. The DB also has a table for storing metadata on the evidence sources, including cross-references to PubMed Central IDs (PMCID) and Digital Object Identifiers (DOI). All metadata is available in the DB and the code repository used to manage the data.

Metadata related to testing and reporting of the variant as well as variant characteristics are out of the scope of CPIC's process for assigning allele clinical function and diplotype to phenotype translation.

Data Uniqueness

Double entry of data is prevented by both the DB and the software used to interact with the DB. First, database-level constraints are included in the definitions of the tables used to store the data. These constraints ensure both uniqueness of appropriate properties and consistent references between tables when they are related to each other. This guarantees that modification of a child table that references a parent table will not accept a value that does not also exist in the parent value. The uniqueness constraints ensure, for example, that the gene alleles in the diplotype-phenotype table are unique for that combination of gene + allele descriptors. The Java application software used to import data also has a layer of preconditions “above” the DB constraints that validates for domain-specific logic. These preconditions ensure particular controlled vocabularies are used or naming conventions are adhered to. Preconditions are part of the codebase and version-controlled in the git repository for the codebase.