
CHORD

David Lougheed

May 11, 2020

INTRODUCTION

1	What is CHORD?	3
2	Project Provenance	5
3	CHORD Singularity Server	7
3.1	Overview	7
3.2	What's Bundled?	7
4	chord_services.json	9
4.1	Overview	9
4.2	Justification	9
4.3	Configuration Parameters	9
5	CHORD Search Format	11
6	chord_workflows.json	13
7	CHORD Service Types	15
8	Creating a Service	17
9	Bundled Services	19
10	Adding Data to the Drop Box	21

This project was funded by CANARIE.

WHAT IS CHORD?

“CHORD is a project to build a federated, Canadian, national data service for privacy- sensitive genomic and related health data. It allows the technologies and services being built by CanDIG [a GA4GH driver project sharing PIs] and its international partners to be made available to the Canadian health research community more broadly.”

Quote from the CHORD SOW; square brackets added for clarity.

In more technical terms, CHORD is a set of microservices that allow the sharing, management, and discovery of genomic and health data across a federated network. CHORD heavily benefits from and utilizes GA4GH’s efforts to create standard APIs and formats for genomic and health data transfer and storage.

PROJECT PROVENANCE

Releases are authorized by a committee composed of CHORD and shared platform software developers and project managers.

Before publication, release candidates *currently* go through the following validation process:

- Comprehensive service- and library-level testing suites
- Testing by hand using synthetic datasets on local machines
- Advance deployment on select instances that opt into testing newer versions

As part of the release process, documentation is included in the form of tagged versions of the this website, and service-level README files for service-specific technical details.

The developers of the platform are constantly monitoring for the latest patches to dependencies used in the project. Any updates that are of critical importance (bug fixes, security flaws) will warrant a patch release of the software itself, which will pass through the standard release vetting process.

CHORD SINGULARITY SERVER

3.1 Overview

To run on the GenAP platform, CHORD services can be bundled together into a single Singularity image, which provides a portable and easy-to-run format for the CHORD suite of microservices.

See the repository for additional details: https://github.com/c3g/chord_singularity

3.2 What's Bundled?

The CHORD Singularity container includes several services, tools, and libraries other than what's provided by CHORD services themselves:

- NodeJS 10
- Python 3.7
- Java 11
- A Redis instance (for message-passing and fast key-value storage) at `/chord/tmp/redis.sock`
- A PostgreSQL 11 instance at `/chord/tmp/postgresql/.s.PGSQL.5433` with service-specific credentials stored in environment variables.
- `htslib`

CHORD_SERVICES.JSON

4.1 Overview

The `chord_services.json` file specifies a list of services for the single-Singularity container server, and is akin to Docker files for each service, although in this case each service is loaded into the same container.

4.2 Justification

CHORD is designed to be ran as an application on the GenAP platform. GenAP applications are instances of one Singularity container, meaning that although CHORD is microservice-based, all microservices must run in the same container.

The `chord_services.json` file specifies configuration for each of these services to allow them to co-operate within the same container.

4.3 Configuration Parameters

TODO

CHORD SEARCH FORMAT

TODO

CHORD_WORKFLOWS.JSON

TODO

CHORD SERVICE TYPES

Artifacts should be unique by themselves within a node, since it allows for a more readable overall API / service file system layout and prevents confusion.

Organization	Artifact	Repository
ca.c3g.chord	service-registry	https://github.com/c3g/chord_service_registry
ca.c3g.chord	drop-box	https://github.com/c3g/chord_drop_box_service
ca.c3g.chord	drs	https://github.com/c3g/chord_drs
ca.c3g.chord	wes	https://github.com/c3g/chord_wes
ca.c3g.chord	federation	https://github.com/c3g/chord_federation
ca.c3g.chord	notification	https://github.com/c3g/chord_notification_service
ca.c3g.chord	event-relay	https://github.com/c3g/chord_event_relay
ca.c3g.chord	metadata	https://github.com/c3g/chord_metadata_service
ca.c3g.chord	variant	https://github.com/c3g/chord_variant_service

CREATING A SERVICE

TODO

BUNDLED SERVICES

TODO

ADDING DATA TO THE DROP BOX

Currently, no upload mechanism is available for CHORD. Files should be copied via some out-of-band mechanism to the CHORD node and placed in whichever folder is bound to `/chord/data/drop-box` inside the container. These files and folders will appear in the front end's "Files" section.