

Data Network Dashboards

This document is currently under construction

2020-03-03

Contents

Preface	5
Goals	5
Contributors	5
License	6
Acknowledges	6
1 Introduction	7
1.1 Data Network Dashboard	7
2 Installation	9
2.1 Insert Concepts	10
2.2 Import dashboards	11
2.3 Dummy data	11
3 General	13
3.1 Database Type Filter	13
3.2 Country Filter	14
3.3 General - World Map	14
3.4 General - Network Growth (Summary)	15
3.5 General - Network Growth by Date	16
3.6 General - Patients per Country	17
3.7 General - Database Types per Country	17
4 Person	19
5 Population Characteristics	21
6 Visit	23
7 Death	25
8 Concepts	27
9 Data Domains	29

Preface

This document is currently under construction

Automated Characterization of Health Information at Large-scale Longitudinal Evidence Systems (ACHILLES) is a profiling tool developed by the OHDSI community to provide descriptive statistics of databases standardized to the OMOP Common Data Model. These characteristics are presented graphically in the ATLAS tool. However, this solution does not allow for database comparison across the data network. The proposed solution aggregates ACHILLES results files from databases in the network and displays the descriptive statistics through graphical dashboards. The tool is helpful to gain insight in the growth of the data network and is useful for the selection of databases for specific research questions. In the software demonstration we show a first version of this tool that will be further developed in EHDEN in close collaboration with all our stakeholders, including OHDSI.

Goals

This manual aims to document the procedures to install and configure the charts and dashboards chosen to compare the OHDSI databases.

....

Contributors

To develop this tool, EHDEN organized a hack-a-thon (Aveiro, December 2-3, 2019), where we defined and implemented a series of charts and dashboards containing the most relevant information about the databases. The team involved in this task were composed by the following members:

- João Rafael Almeida¹
- André Pedrosa¹
- Peter R. Rijnbeek²

- Marcel de Wilde²
- Michel Van Speybroeck³
- Maxim Moinat⁴
- Pedro Freire¹
- Alina Trifan¹
- Sérgio Matos¹
- José Luís Oliveira¹

1 - Institute of Electronics and Informatics Engineering of Aveiro, Department of Electronics and Telecommunication, University of Aveiro, Aveiro, Portugal

2 - Erasmus MC, Rotterdam, Netherlands

3 - Janssen Pharmaceutica NV, Beerse, Belgium

4 - The Hyve, Utrecht, Netherlands

License

The system is open-source under the license

The book is written in RMarkdown using the bookdown package.

Acknowledges

This work has been conducted in the context of EHDEN, a project that receives funding from the European Union's Horizon 2020 and EFPIA through IMI2 Joint Undertaking initiative, under grant agreement No 806968.

Chapter 1

Introduction

The OHDSI research network has been growing steadily which results in an increasing number of healthcare databases standardized to the OMOP CDM format. The OHDSI community created the ACHILLES tool (Automated Characterization of Health Information at Large-scale Longitudinal Exploration System) to characterize those databases. The results are available to the data custodian in their local ATLAS tool and helps them to gain insights in their data and helps in assessing the feasibility of a particular research questions.

ACHILLES was designed to extract the metadata from a single database, which by itself does not allow the comparison with the remaining databases in the network. However, we believe there is even more value in sharing this information with others to enable network research in a Data Network Dashboard.

1.1 Data Network Dashboard

The European Health Data and Evidence Network (EHDEN) project therefore designed a Data Network Dashboard tool, a web application to aggregate information from distributed OMOP CDM databases. It uses the ACHILLES results files to construct graphical dashboards and enables database comparison (Figure 1.1). The tool is built on Apache Superset, which is an open-source enterprise-ready business intelligence web application that can provide powerful and fully customizable graphical representations of data. Achilles results can be uploaded through the EHDEN Database Catalogue using the dashboards plugin but can also be directly uploaded in the tool. Figure 1. Example of a dashboards tool presenting age and gender distributions (simulated data).

In this tools, we defined and implemented a series of charts and dashboards containing the most relevant information about the databases, such as:

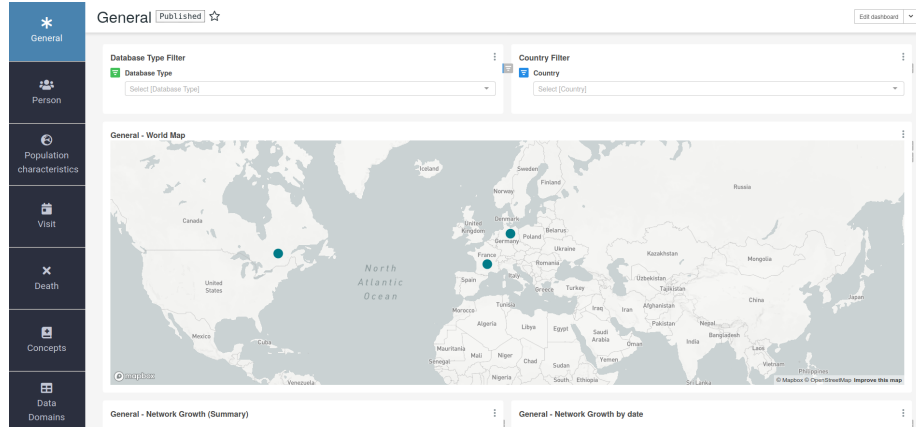


Figure 1.1: Example of a dashboards tool presenting the databases available in the network (simulated data)

- **General:** dashboards that shows the databases types per country, the distribution of data source types, the growth of the Network including the number of database and the number of patients in the databases over time;
- **Person:** representing the number of patients per country, age distribution at first observation, year of birth distribution and normalized gender distribution;
- **Population characteristics:** dashboard with the cumulative patient time, persons with continuous observation per month, and the start and end dates of those periods;
- **Visit:** chart to compare the number and type of visit occurrence records;
- **Death:** information about the number of death records by month, and the patient age at time of death;
- **Concepts:** bubble chart which shows the number of patients and records per concept over the databases;
- **Data domains:** heat map visualization of the major data domains in each database.

Chapter 2

Installation

Make sure that you have docker and docker-compose installed in your machine. Then, please follow these steps:

- Please enter in the “docker” directory and create your `.env` file here, using `.env-example` as reference. For local installation, you can just copy the `.env-example` content to a new file. Note: In case of port errors in the next steps, the problem could be related to a port already in use by your system that you defined here and it is busy, chose other.
- Tip the following commands in the command line:
 1. Clone the Apache Superset repository:

```
git clone https://github.com/apache/incubator-superset
  ./superset
cp ../superset/contrib/docker/superset_config.py ../superset
```
 2. Init the Apache Superset (This creates a user, so it is necessary to interact with the console):

```
docker-compose run --rm superset ./docker-init.sh
```
 3. Init the Dashboard Layout (This creates a user, so it is necessary to interact with the console):

```
docker-compose run --rm dashboard_viewer ./docker-init.sh
```
 4. Finally, bring up the containers

```
docker-compose up -d
```

To check if everything is ok, please wait 2 minutes and tip `docker ps` and the following containers need to be running:

```
... 0.0.0.0:8088->8088/tcp    dashboard_viewer_superset_1
```

```
... 0.0.0.0:8000->8000/tcp    dashboard_viewer_dashboard_viewer_1
... 0.0.0.0:6379->6379/tcp    dashboard_viewer_redis_1
... 5432/tcp                  dashboard_viewer_postgres_1
```

Now, you have a clean setup running in your machine. To try the application using synthetic data, please continue to follow the steps in the “Demo” section.

2.1 Insert Concepts

The concepts table are not in the repository due to its dimension. Therefore, to insert this table in the installation, you should perform the following steps:

1. Download concept.csv file from here (todo)
2. Copy the file to the /tmp directory inside of the postgres container

```
docker cp concept.csv dashboard_viewer_postgres_1:/tmp/
```

3. Enter in the dashboard_viewer_postgres_1 container:

```
docker exec -it dashboard_viewer_postgres_1 bash
```

4. Enter in the achilles database:

```
psql achilles
```

5. Create the table in the database using this command:

```
CREATE TABLE concept (
    concept_id        INTEGER        NOT NULL,
    concept_name       VARCHAR(255)   NOT NULL,
    domain_id         VARCHAR(20)    NOT NULL,
    vocabulary_id      VARCHAR(20)    NOT NULL,
    concept_class_id   VARCHAR(20)    NOT NULL,
    standard_concept    VARCHAR(1)    NULL,
    concept_code        VARCHAR(50)    NOT NULL,
    valid_start_date   DATE           NOT NULL,
    valid_end_date     DATE           NOT NULL,
    invalid_reason     VARCHAR(1)     NULL
);
```

6. Copy the CSV file content to the table (this could take a while):

```
COPY public.concept from '/tmp/concept.csv' WITH DELIMITER ','
CSV HEADER;
```

7. Alter table ownership:

```
-- <user> : defined in the .env file
ALTER TABLE public.concept OWNER TO <user>;
```

8. Create index in table:

```
CREATE INDEX achilles_results_analysis_id_index ON
  achilles_results (analysis_id);
CREATE INDEX achilles_results_source_index ON achilles_results
  (data_source_id);
CREATE INDEX concept_concept_id_index ON concept (concept_id);
CREATE INDEX concept_concept_name_index ON concept
  (concept_name);
```

2.2 Import dashboards

TO DO

2.3 Dummy data

TO DO

Chapter 3

General

Discuss the goal of this dashboard... TO DO

3.1 Database Type Filter

This filter which is a type of chart in Superset was designed to be used in the dashboard aiming the filtering of the data based on the field “database_type” from the table “data_source”. It is important to give the alias “Type” to this field in the select operations because Superset does not recognize as the same field otherwise. ### SQL query

```
-- Country and database type filters
SELECT source.name,
        country.country AS Country,
        database_type AS Type,
        source.slug
FROM public.data_source AS source INNER JOIN public.country
     AS country ON source.country_id=country.id;
```

3.1.1 Chart settings

The main characteristics of this chart are presented in Figure 3.1, being the following:

- todo

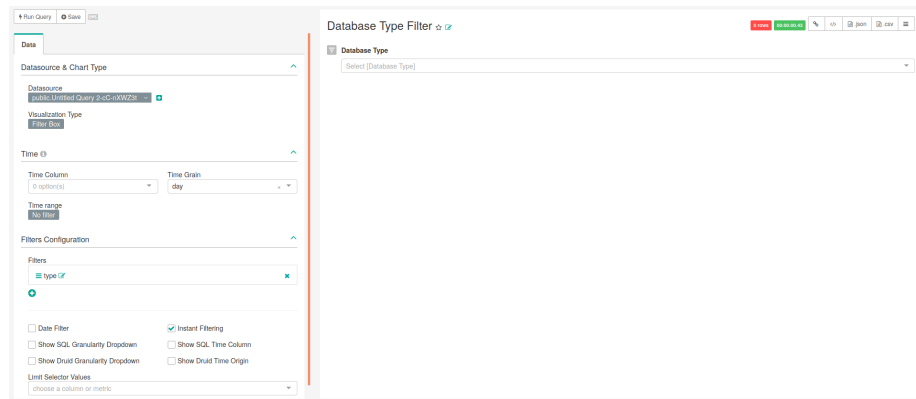


Figure 3.1: Superset chart creation: Settings for creating the database type filter.

3.2 Country Filter

Discuss what is important to see in this chart... TO DO

3.2.1 SQL query

```
-- Country and database type filters
SELECT source.name,
       country.country AS Country,
       database_type AS Type,
       source.slug
FROM public.data_source AS source INNER JOIN public.country
     AS country ON source.country_id=country.id;
```

3.2.2 Chart settings

TO DO

3.3 General - World Map

Discuss what is important to see in this chart... TO DO

3.3.1 SQL query

```
--      General - World Map
SELECT  name,
        slug,
        release_date,
        database_type AS Type,
        latitude,
        longitude,
        link,
        country AS Country,
        continent
FROM public.data_source AS source INNER JOIN public.country
     AS country ON source.country_id=country.id;
```

3.3.2 Chart settings

TO DO

3.4 General - Network Growth (Summary)

Discuss what is important to see in this chart... TO DO

3.4.1 SQL query

```
-- 108      General - Network Growth (Summary)
SELECT data.source,
        data.country AS Country,
        data.database_type AS Type,
        --cast(stratum_1 as INTEGER )*30 AS Days,
        data.release_date - cast(stratum_1 AS INTEGER) * INTERVAL
            '1 month' as Time,
        count_value                AS count
FROM (
    SELECT source.name                AS source,
           achilles.analysis_id      AS analysis_id,
           achilles.stratum_1,
           achilles.stratum_2,
           achilles.stratum_3,
           achilles.stratum_4,
           achilles.stratum_5,
```

```

        achilles.count_value,
        country.country,
        source.database_type,
        source.release_date
FROM public.achilles_results AS achilles INNER JOIN
    public.data_source AS source ON
    achilles.data_source_id=source.id
INNER JOIN public.country AS country ON
    source.country_id=country.id
) data
WHERE analysis_id = 108;

```

3.4.2 Chart settings

TO DO

3.5 General - Network Growth by Date

Discuss what is important to see in this chart... TO DO

3.5.1 SQL query

```

-- 108    General - Network Growth by Date
SELECT data.source,
       data.country,
       data.database_type,
       cast(stratum_1 as Integer)*30 AS DAY,
       count_value                    AS count
FROM (
    SELECT source.name                AS source,
           achilles.analysis_id      AS analysis_id,
           achilles.stratum_1,
           achilles.stratum_2,
           achilles.stratum_3,
           achilles.stratum_4,
           achilles.stratum_5,
           achilles.count_value,
           country.country,
           source.database_type
    FROM public.achilles_results AS achilles INNER JOIN
        public.data_source AS source ON

```



```
    achilles.data_source_id=source.id
  INNER JOIN public.country AS country ON
    source.country_id=country.id
) data
WHERE analysis_id = 108;
```

3.5.2 Chart settings

TO DO

3.6 General - Patients per Country

Discuss what is important to see in this chart... TO DO

3.6.1 SQL query

```
-- 1    General - Patients per Country
SELECT source.name,
       country.country,
       source.database_type,
       count_value as patient_count,
       source.slug
FROM public.achilles_results AS achilles INNER JOIN
  public.data_source AS source ON achilles.data_source_id=source.id
  INNER JOIN public.country AS country ON
    source.country_id=country.id
WHERE analysis_id = 1;
```

3.6.2 Chart settings

TO DO

3.7 General - Database Types per Country

Discuss what is important to see in this chart... TO DO

3.7.1 SQL query

```
-- 1    General - Database types per Country
SELECT source.name,
       country.country as Country,
       database_type as Type,
       count_value as "Nr_patients",
       source.slug
FROM public.achilles_results AS achilles INNER JOIN
     public.data_source AS source ON achilles.data_source_id=source.id
     INNER JOIN public.country AS country ON
         source.country_id=country.id
WHERE analysis_id = 1;
```

3.7.2 Chart settings

TO DO

Chapter 4

Person

Chapter 5

Population Characteristics

Chapter 6

Visit

Chapter 7

Death

Chapter 8

Concepts

Chapter 9

Data Domains