International Journal of Hybrid Information Technology
Vol.2, No.1 (2022), pp.1-12
http://dx.doi.org/10.21742/ijhit.2653-309X.2022.2.1.01

Extending the Power of Problem Oriented Medical Record with Disease Association Discovery: The Case Study of Empowering QL4POMR with OpenTargets

Sabah Mohammed[1*] and Jinan Fiaidhi[2]

[1,2]*Department of Computer Science, Lakehead University, Canada*
[1*]*mohammed@lakeheadu.ca,* [2]*jfiaidhi@lakeheadu.ca*

Abstract

Medical informatics was profoundly influenced by clinical case presentations for providing evidence-based medicine. However, concerns about weak inferences and the high likelihood of bias associated with such reports have resulted in minimal attention being devoted to developing frameworks for approaching, appraising, synthesizing, and applying evidence derived from case reports/series. Nevertheless, the nature of being in a connected world through the emerging information technologies presented a wind of change to link these clinical cases to create together with other knowledge sources from basic science to enhance human health and well-being as well as to build stronger evidence-based medicine. This article is an attempt to describe how to link clinical cases described through the SOAP (Subjective, Objective, Assessment, and Plan) note to the electronic healthcare record and the other associated clinical information from diverse courses. It is part of our efforts to extend our developed QL4POMR problem-oriented medical record to provide more associated clinical information on related drugs and evidence that strengthen the clinician's decision for diagnosis and prognosis. Providing such clinical association was made through the incorporation of a data layer using the Gatsby API and the external biomedical associations through the OpenTargets API.

Keywords*: Translational medical informatics, GraphQL, QL4POMR, Problem oriented medical record, SOAP, Disease associations*

1. Introduction

Clerkships, physicians, and nurses follow the SOAP note in describing, presenting, and planning for patient care. Although the SOAP note was created nearly 60 years ago by Lawrence Weed, MD, it is still commonly used in medical and health science schools [1]. SOAP is the acronym derived from the Problem Oriented Medical Record (POMR) recipe for a progress note about a care problem: (1) subjective, (2) objective, (3) assessment, and (4) plans. While SOAP excels at organizing care information around problem lists, the Electronic Healthcare Record (EMR) industry is not taking advantage [2] and for this reason, the documentation of patient care remains neglected. The health industry's main objective should be centered on medical diagnosis which starts with teaching, learning, and clinical problem-solving. The use of EMR alone neither helps the physician to represent and plan for patient

Article history:
Received (June, 10 2022), Review Result (July 15, 2022), Accepted (September 10, 2022)

care nor does it provide the analytical approach to learning disease patterns. Understanding these patterns helps the physician to determine how best to help the patient. Integrating relevant information to the patient case from various resources can tremendously assist physicians to identify the best care plan and target the most effective drugs and treatments associated with diseases and phenotypes. Such association can be found over a diverse source of information such as medical literature (e.g. Europe PMC [7]), Medical Portals (e.g. DisGeNET [8] for relating genes to human diseases and DISEASES [9]), catalogs (e.g. GWAS [10] for Human Genome-Wide Association) and clinical archives (e.g. ClinVar [11]). However, extracting such associations from these sources requires sophisticated text mining techniques including NER, Controlled Vocabularies, Onologies, Dictionaries, and Corpora, Retrieving and Annotation Techniques [3]. Extraction of insights from all the gathered information requires techniques that go beyond text mining as text is lacking normalization [4]. Modern extraction techniques, however, aggregate citation of information with content-based textual mining using classification schema [5]. The use of classification schema like SOAP provides dynamic mapping of interrelationships from pieces of knowledge across the care practice. QL4POMR described a working implementation of a system designed to enhance the knowledge discovery process through graph-based representation and interaction API to mine and map care findings and their associated methods [6]. The API used by QL4POMR is GraphQL [12] which have a significant role in extracting knowledge from diverse resources. The following list of GraphQL capabilities within the QL4POMR system:

- Call the relevant APIs and obtain the needed data from the requested sources
- Format the data based on the frontend representation
- Send the formatted data to the frontend
- Creates a uniform API across the entire application without being limited by a specific storage engine.
- Add new fields and types to your GraphQL API without impacting existing queries.

GraphQL APIs are organized in terms of types and fields, not endpoints. GraphQL queries access not just the properties of one resource but also smoothly follow references between them. GraphQL APIs get all the data the interface needs in a single request. As a result, there will be minimal logic on the front end. Therefore, GraphQL helps to streamline data representation and takes up the responsibility of providing a well-focused interface for the front. However, given the amount of information contained in EMR and the associations that can be added to the patient case, it is important not only to file the extracted information according to one large schema as GraphQL mandates but also to have the ability to interact with external sources where the GraphQL can have the ability interrogate the external sources without integrating their schema. To incorporate this capability, we have empowered the QL4POMR with a data layer that can sniff the schema from the external resources using the Gatsby API [13]. The Gatsby data layer encompasses both Gatsby's internal GraphQL API and the data source plugins, which together collect data and define a GraphQL schema that traverses that data. Whether this data comes from the surrounding filesystem or a REST or other GraphQL, Gatsby's internal GraphQL API facilitates the single-file co-location of data requirements, extraction and data rendering. The Gatsby v2.1.0 API introduces a new feature called useStaticQuery to provide the ability to use a React Hook to query with GraphQL at build time. It allows your React components to retrieve data via a GraphQL query that will be parsed, evaluated, and injected into the component. However, useStaticQuery is a hook rather

than a component that takes a render prop! Having this ability we can interrogate external resources like the HL7 FHIR EMR system.

```
import { useStaticQuery, graphql } from "gatsby"
export const useSiteMetadata = () => {
  const { site } = useStaticQuery(
    graphql`
      query SiteMetaData {
        site {
          siteMetadata {
            title
            siteUrl
            headline
            description
            image
            name
            logo
          }
        }
      }
    `
  )
  return site.siteMetadata
}
```

Then we can implement the hook as follows to extract information from the source site like its title as follows:

```
import React from "react"
import { useSiteMetadata } from "../hooks/use-site-metadata"
export default function Home() {
  const { title, siteUrl } = useSiteMetadata()
  return <h1>welcome to {title}</h1>
}
```

[Figure 1] illustrates the overall architecture of our QL4POMR with the data layer for extracting data from external sources.

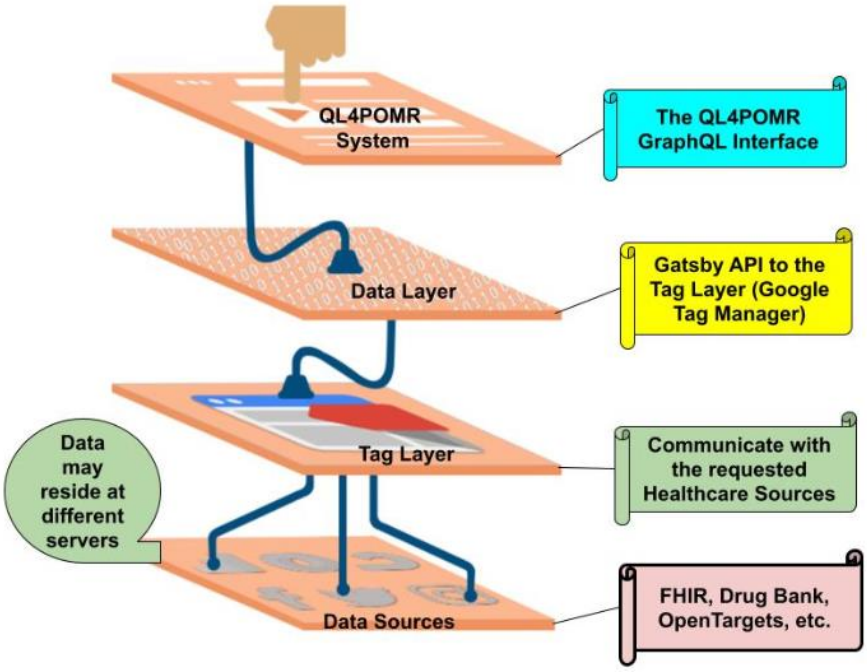

Figure 1. QL4POMR with Gatsby data layer

2. Empowering Ql4pomr with Disease Associations Capabilities

QL4POMR empowerment to identify disease associations has been made possible through the use of the Open Targets Platform [14] API to programmatically access the data sources like drugs and diseases through the APIs. Based on the OpenTargets API, we can query interesting disease associations like:

- Target drug profile (protein information, pathways, tractability, baseline expression, etc)
- Disease associations for a targeted drug
- Evidence linking a drug target to a disease
- Gene summary (associated studies for a given gene)
- Study summary (associated loci, lead variants, linked genes):
- Variant summary (variant details, assigned genes, PheWAS, linked variants)
- Association summary (gene prioritization, colocalization, credible set overlap)

By setting the OpenTargets Platform GraphQL browser one can interrogate huge disease association data sources which include autocompleting capabilities and documentation which are useful for exploring the data, getting to know the schemas, and testing out queries. The OpenTargets browser playground can be accessed through this link:

https://api.platform.opentargets.org/api/v4/graphql/browser

In our design QL4POMR through the data, layer constructs a query by including the relevant fields that we want to seek its association (e.g. disease to get information about a

disease, or target for information about a drug target). Bypassing these variables to the OpenTargets GraphQL we can extract useful associations. The OpenTargets uses the Experimental Factor Ontology (EFO). Each disease is linked to an EFO ID (e.g. the EFO ID for “coronary artery disease” is EFO_0001645). Here’s a query that will do that, which you can execute in the interactive browser:

```
query diseaseAndDrugs {
  disease(efoId: "EFO_0001645"){
    id
    name
    knownDrugs {
      uniqueDrugs
      rows {
        drug {
          id
          name
          isApproved
        }
      }
    }
  }
}
```

The response to this query may bring the following drug targets associated with this disease:

```
{
  "data": {
    "disease": {
      "id": "EFO_0001645",
      "name": "coronary artery disease",
      "knownDrugs": {
        "uniqueDrugs": 218,
        "rows": [
          {
            "drug": {
              "id": "CHEMBL1200694",
              "name": "SEVOFLURANE",
              "isApproved": true
            }
          },
          {
            "drug": {
              "id": "CHEMBL2103749",
              "name": "BIVALIRUDIN",
              "isApproved": true
            }
          },
```

```
{
 "drug": {
  "id": "CHEMBL79",
  "name": "LIDOCAINE",
  "isApproved": true
 }
},
{
 "drug": {
  "id": "CHEMBL1431",
  "name": "METFORMIN",
  "isApproved": true
 }
},
{
 "drug": {
  "id": "CHEMBL1311",
  "name": "ISOSORBIDE MONONITRATE",
  "isApproved": true
 }
},
{
 "drug": {
  "id": "CHEMBL1201657",
  "name": "HEPARIN SODIUM",
  "isApproved": true
 }
},
{
 "drug": {
  "id": "CHEMBL1200694",
  "name": "SEVOFLURANE",
  "isApproved": true
 }
},
{
 "drug": {
  "id": "CHEMBL932",
  "name": "DIPYRIDAMOLE",
  "isApproved": true
 }
},
{
 "drug": {
  "id": "CHEMBL742",
  "name": "KETAMINE",
  "isApproved": true
 }
```

},
{
 "drug": {
  "id": "CHEMBL742",
  "name": "KETAMINE",
  "isApproved": true
 }
},
{
 "drug": {
  "id": "CHEMBL1396",
  "name": "VARENICLINE",
  "isApproved": true
 }
},
{
 "drug": {
  "id": "CHEMBL932",
  "name": "DIPYRIDAMOLE",
  "isApproved": true
 }
},
{
 "drug": {
  "id": "CHEMBL113",
  "name": "CAFFEINE",
  "isApproved": true
 }
},
{
 "drug": {
  "id": "CHEMBL25",
  "name": "ASPIRIN",
  "isApproved": true
 }
},
{
 "drug": {
  "id": "CHEMBL1431",
  "name": "METFORMIN",
  "isApproved": true
 }
},
{
 "drug": {
  "id": "CHEMBL3",
  "name": "NICOTINE",
  "isApproved": true

```
  }
 },
 {
  "drug": {
   "id": "CHEMBL526",
   "name": "PROPOFOL",
   "isApproved": true
  }
 },
 {
  "drug": {
   "id": "CHEMBL1201476",
   "name": "ENOXAPARIN SODIUM",
   "isApproved": true
  }
 },
 {
  "drug": {
   "id": "CHEMBL681",
   "name": "ETOMIDATE",
   "isApproved": true
  }
 },
 {
  "drug": {
   "id": "CHEMBL1200694",
   "name": "SEVOFLURANE",
   "isApproved": true
  }
 },
 {
  "drug": {
   "id": "CHEMBL25",
   "name": "ASPIRIN",
   "isApproved": true
  }
 },
 {
  "drug": {
   "id": "CHEMBL1201584",
   "name": "ABCIXIMAB",
   "isApproved": true
  }
 },
 {
  "drug": {
   "id": "CHEMBL1200402",
   "name": "AMLODIPINE BESYLATE",
```

```
            "isApproved": true
           }
          },
          {
           "drug": {
            "id": "CHEMBL1200733",
            "name": "DESFLURANE",
            "isApproved": true
           }
          },
          {
           "drug": {
            "id": "CHEMBL932",
            "name": "DIPYRIDAMOLE",
            "isApproved": true
       }
      }
     ]
    }
   }
  }
}
```


Figure 2 illustrates a screenshot from the OpenTargets GraphQL Browser.

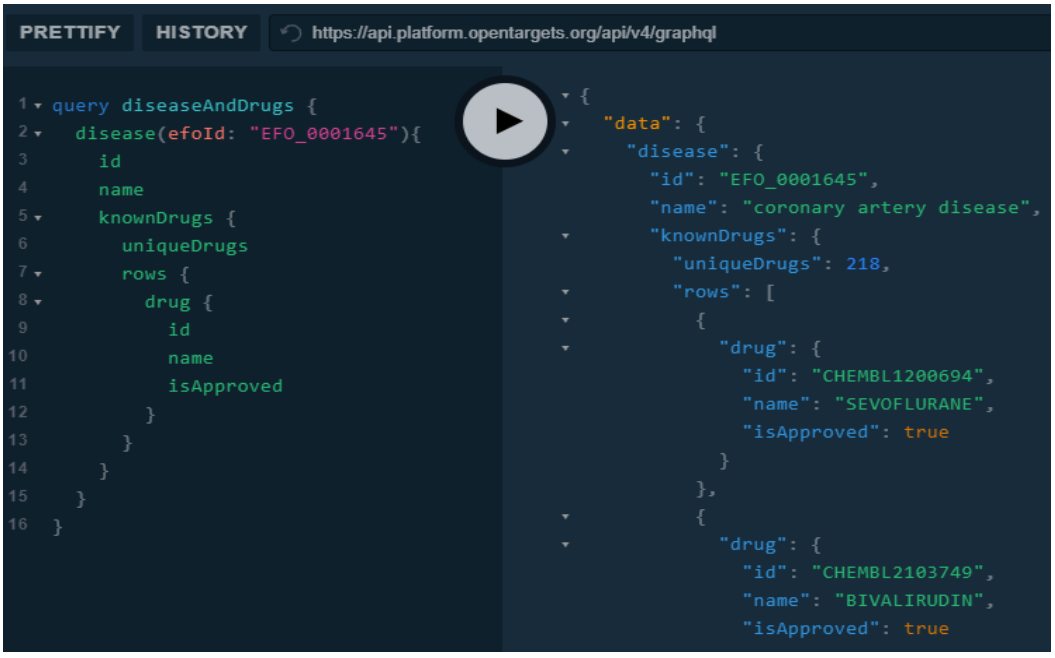

Figure 2. The OpenTargets GraphQL browser querying disease to drugs associations

3. Conclusions

Industry 4.0 initiative and the connected world using the Internet have provided healthcare with infrastructure to strengthen the evidence-based nature of medicine. The ultimate goal of this massive scale connectivity is to end with translational medicine to provide new robust evidence-based treatments and insights towards the improvement of health across populations. However, the first step starts with the presentation of clinical cases classified around a problem list of what patient encounters. This step starts with a clinical note like SOAP and ends with a clinical record like the HL7 FHIR that supports interoperability and ends with larger clinical knowledge associations to the assessment and plans the clinician may make supporting diagnosis and prognosis. [Figure 3] illustrates our effort in building an extended QL4POMR with the help of OpenTargets API to provide associated clinical knowledge with drugs and diseases.

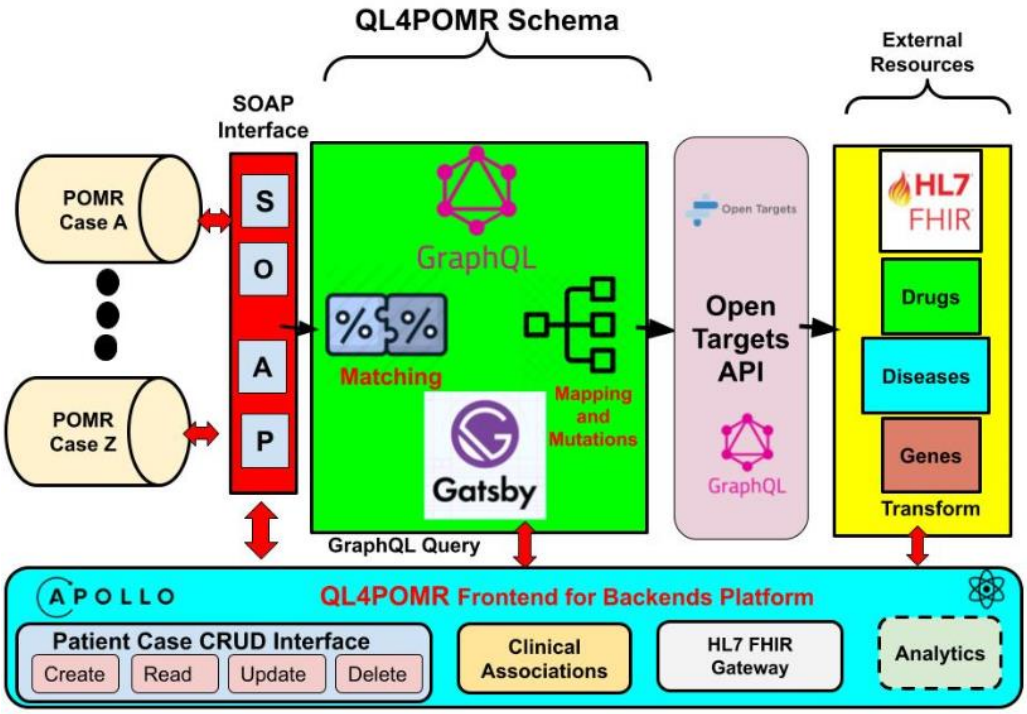

Figure. 3. The overall architecture of QL4POMER with OpenTargets integration

We are continuing our efforts to tab on the other clinical associations that the OpenTargets provides and especially with the genetic information and proteins that physicians may target to treat their cases. This will be left to our future studies and publications. [Figure 4] illustrates all the venues the OpenTargets can provide to enhance the evidence-based association of clinical cases.

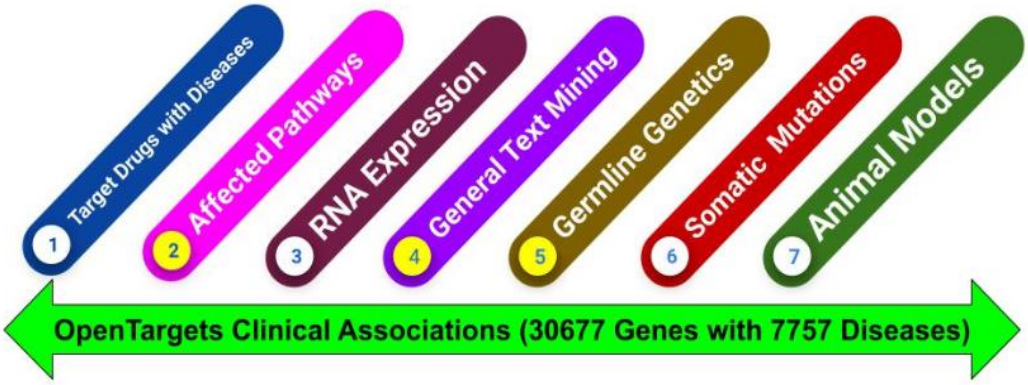

Figure 4. Clinical associations provided through the OpenTargets API

Acknowledgments

This research is funded by the first author NSERC Discovery Grant 2021 (Discovery Development Grant DDG-2021-00014) and the MITACS research project of both authors (MITACS_IT22305)

References

[1] J. M. Faragher, "Bungalow and ranch house: The architectural backwash of California," Western Historical Quarterly, vol.32, no.2, pp.149-173, **(2018)**
[2] Jaroudi, Sarah, and J. Drew Payne, "Remembering Lawrence Weed: A pioneer of the SOAP note," Academic Medicine, vol.94, no.1, pp.11, **(2019)**
[3] S. Pletscher-Frankild, A. Pallejà, K. Tsafou, J. X. Binder, and L. J. Jensen. "DISEASES: Text mining and data integration of disease-gene associations," Methods, vol.74, pp.83-89, **(2015)**
[4] C. Luque, J. M. Luna, M. Luque, and S. Ventura, "An advanced review on text mining in medicine," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol.9, no.3, pp.e1302, **(2019)**
[5] S. Lamparter, M. Ehrig, and C. Tempich, "Knowledge extraction from classification schemas," In OTM Confederated International Conferences on the Move to Meaningful Internet Systems, pp.618-636. Springer, Berlin, Heidelberg, **(2004)**
[6] S. Mohammed and J. Fiaidhi, "Establishment of a mindmap for medical e-Diagnosis as a service for graph-based learning and analytics," Neural Computing and Applications, pp.1-12, **(2021)**
[7] https://europepmc.org/
[8] https://www.disgenet.org/
[9] https://diseases.jensenlab.org/Search
[10] https://www.ebi.ac.uk/gwas/
[11] https://www.ncbi.nlm.nih.gov/clinvar/intro/
[12] https://graphql.org/
[13] https://www.gatsbyjs.com/docs/api-specification/
[14] https://platform.opentargets.org/

This page is empty by intention.