The Use of the Julia Programming Language for Global Health Informatics and Observational Health Research

Jacob S. Zelko 1, 2, 3; Varshini Chinta 1,2; Malina Hy 1,2; Fareeda Abdelazeez 4

1. Georgia Tech Research Institute, Georgia Institute of Technology
2. Roux Institute, Northeastern University
3. OneFact Foundation
4. Zewail University

Background

Within the fields of global health informatics and observational health research, the use of real world data continues to grow. (1,2) Alongside that, the demands for the effective utilization of real world evidence based on such data increases – potentially outpacing the current speed at which such studies can generate results. (3) One aspect of conducting real world data-based studies is the current technological landscape of tools available for such research on a global scale. For example, popular languages such as R, Java, and SQL are used in many of these studies and are provisioned through the OHDSI HADES Suite. (4)

However, growing pains within running globally federated network studies continue to emerge. This software demo emphasizes a possible complement to the OHDSI community to perhaps mitigate some of these issues: conducting network studies via the Julia programming language. (5) Julia was first released from MIT in 2012 where it was designed to solve problems within high performance, numerical, and computational computing. With its emphasis on speed, composability, and expressivity, Julia makes an ideal complement to handling the tremendous amounts of data generated in observational health scenarios to generate meaningful real world data insights.

Methods

This software demonstration spans the current state of the art within the JuliaHealth ecosystem and examines tools made for observational health research. (6) A brief introduction to Julia as well as some of its noteworthy features will be covered initially. Then, particular JuliaHealth software and tools will be discussed such as:

- OMOPCDMCohortCreator.jl: a tool for building cohorts off of OMOP CDM databases (7)
- DBConnector.jl: tooling to support connection to a variety of databases using ODBC/JDBC over a common interface
- OHDSIAPI.jl: REST API that provides an interface to OHDSI web server deployments
- OHDSICohortExpressions.jl: re-implementation of OHDSI's Circe JSON to SQL Compiler

If time permits, some additional demonstrations of software that can be shown will be:

- Patient Level Prediction in Julia: using Julia machine learning implementations for patient level prediction
- Coarse Treatment Pathways in Julia: using interactive web-based Julia dashboards to explore coarse treatment pathways on OMOP CDM databases

Results

This demonstration will highlight the ability for packages within the JuliaHealth ecosystem to compose together to create rapid analyses on OMOP CDM databases as exemplified in Figures 1 - 2. Additionally, potential results from a “dummy” study will be explored using an interactive web interface (an example of such an interface is shown in Figure 3).

```julia
import OMOPCDMCohortCreator as occ

import Base: Fix2, Matrix
import Chain: @chain
import DataFrames: Not, outerjoin, groupby, combine
import DBConnector: connect
import HealthSampleData: Eunomia

eunomia = connect(Eunomia())

GetPatientRace₂ = Fix2(occ.GetPatientRace, eunomia)
GetPatientAgeGroup₂ = Fix2(occ.GetPatientAgeGroup, eunomia)
GetPatientGender₂ = Fix2(occ.GetPatientGender, eunomia)

disease_concept_id = 28060

@chain occ.ConditionFilterPersonIDs(disease_concept_id, eunomia) begin
    GetPatientAgeGroup₂(_.person_id)
    outerjoin(_, GetPatientRace₂(_.person_id), on = :person_id => :person_id)
    outerjoin(_, GetPatientGender₂(_.person_id), on = :person_id => :person_id)
    _[:, Not(:person_id)]
    groupby(_, names(_))
    combine(_, nrow => :count)
    occ.ExecuteAudit
end
```

Figure 1: Conducting a characterization study through Julia package composition. It is

natural within the Julia ecosystem to have multiple smaller, compact, and well-maintained packages. These packages are then able to compose together to create novel and useful functionalities highly applicable in an observational health research context. For example, this code block uses 6 different Julia packages that were created for different purposes and have not, to the authors' knowledge, been used in this research context before. Yet, due to consistent interfaces, we were able to create a workflow that produced a full characterization study on the Eunomia dataset looking at patients with *streptococcal pharyngitis* (Concept ID: 28060) and suppressed patient subpopulation counts.

```
ConditionFilterPersonIDs₂ = Fix2(occ.ConditionFilterPersonIDs, eunomia)

CohortAge(x) = (GetPatientAgeGroup₂ ∘ Matrix ∘ ConditionFilterPersonIDs₂)(x)

CohortAge(disease_concept_id)
```

Figure 2: Julia function composition. Similarly, functions within Julia can themselves be composed using the mathematical composition operator, ∘. In this example, three functions (from two different packages) are used to produce a novel and compact function that accepts a single disease Concept ID and returns back all patients ever having that condition and their associated age group. Due to this ability to compose functions, further packages can be developed to quickly and rapidly answer domain specific questions leveraging the pre-existing Julia ecosystem in novel ways.

Figure 3: Interactive Julia dashboards. Speaking further to the maturity of the Julia data science and visualization ecosystem, it is fully possible to develop web-based dashboards to visualize study results similar to Plotly or Shiny dashboards.

Conclusion

By the end of this demonstration, attendees will have attained not only a broad understanding of what Julia is but also its capabilities for participating in observational health and global health informatics research. It should be clearly seen that Julia can be a viable option for global health informatics and strengthens not only the goals of research groups but also existing OMOP CDM based ecosystems. Finally, this demonstration highlights a potentially strong complement to existing architecture within the OHDSI community that can further push the OHDSI community in novel research directions and spaces.

References:

1. DevCon 2023 Workshop: Network studies users' group [Internet]. 2023 [cited 2023 Jun 16]. Available from: https://www.youtube.com/watch?v=Pkf41w1vgs0
2. U.S. Food and Drug Administration. Real-World Evidence [Internet]. 2021 [cited 2021 Oct 27]. Available from: https://www.fda.gov/science-research/science-and-research-special-topics/real-world-evidence
3. Collaborators – OHDSI [Internet]. [cited 2023 Jun 16]. Available from: https://www.ohdsi.org/who-we-are/collaborators/
4. HADES [Internet]. [cited 2023 Jun 16]. Available from: https://ohdsi.github.io/Hades/index.html
5. Bezanson J, Karpinski S, Shah V, Edelman A. Why We Created Julia [Internet]. [cited 2023 Jun 16]. Available from: https://julialang.org/blog/2012/02/why-we-created-julia/

6. JuliaHealth [Internet]. [cited 2021 Aug 13]. Available from: https://juliahealth.org/
7. Zelko J, Chinta V. OMOPCDMCohortCreator 0.2.2 [Internet]. 2023 [cited 2023 Jun 16]. Available from: https://github.com/JuliaHealth/OMOPCDMCohortCreator.jl