

Generalizable Analysis Pipelines for Handling Large HTA Submissions

Matthew Phelps & Nicolai Skov Johnsen

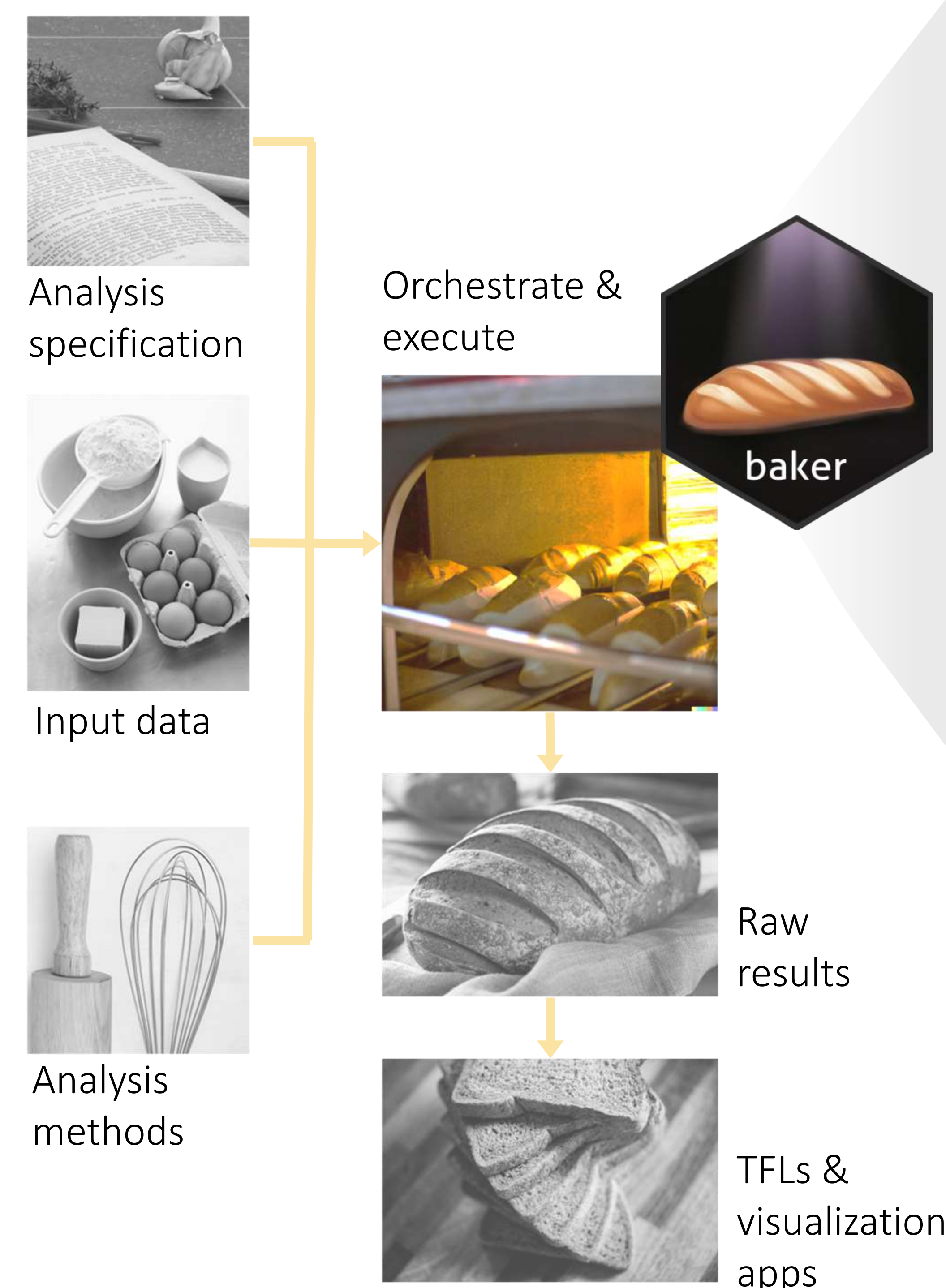
Motivation

Some HTA submissions require thousands of analyses per submission. Producing these is intensive in both programmer and compute time, while often requiring large re-writes of code for each submission despite significant functional overlap.

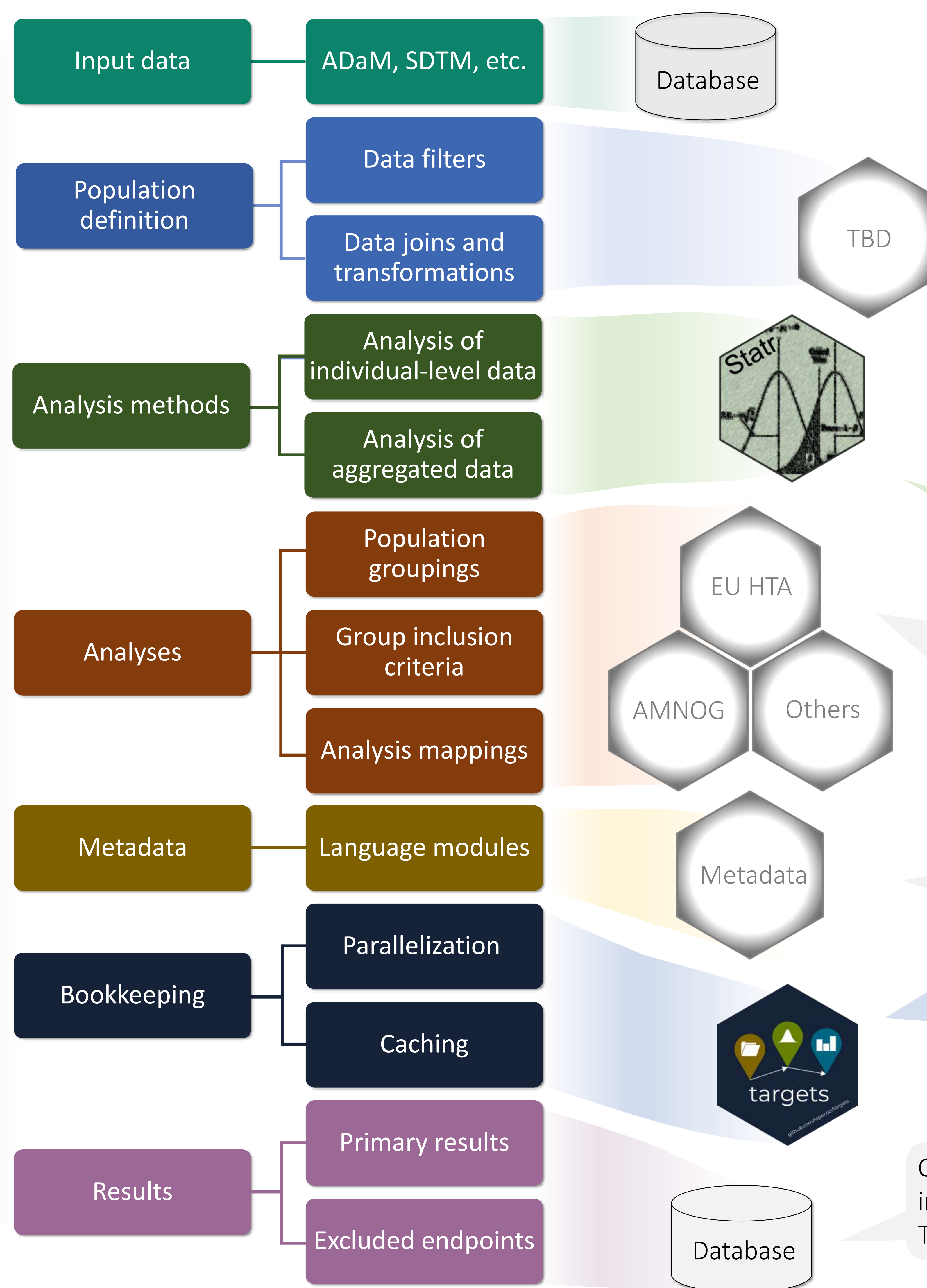
Solution

R-based modular framework that:

- 1) reduces programmer time with greater code re-use between submissions and increased code quality
- 2) reduces compute time via caching, dependency tracking and parallel computing.



Execution handled by R package **baker** that coordinates inputs, outputs, and deploys relevant modules.



Business Gains

Currently, a **50% reduction** in programmer/statistician time in pilot AMNOG submissions. Increased standardization.

**Modules in greyscale not yet implemented*

Modules to transform input data into analysis tables ready for HTA analysis. Could include Pharmaverse (admiral, etc.), or others – please share your ideas with us!

Library of validated statistical methods tailored to needs of HTA analyses.

Logic specifying how to group the population into different groupings, which groups to drop due to e.g., too few samples, and which analyses to apply to each group.

Modules for managing metadata. Include labels, footnotes, translations into other languages (German, Chinese etc.), program logs, etc.

A Make-like tool by Will Landau that avoids redundant runtime, facilitates parallelization, and ensures results always reflect current state of the program [1].

Output results as database, allowing independent downstream formatting into TFLs, apps, or others submission deliverables.

Framework model

Execution stage data model

Implementation modules