

Title: *PhenotypeRefiner: A transparent, end-to-end workflow that uses LLMs to produce and characterize executable cohort definitions from human-readable descriptions of clinical phenotypes.*

Authors: Jared Houghtaling¹, Azza Shoaibi¹, Joel Swerdel¹, James Gilbert¹, Martijn Schuemie¹, Patrick Ryan¹

Affiliations:

1. Observational Health Data Analytics, Global Epidemiology, Johnson & Johnson Research and Development, Titusville, NJ, USA

Background

Translating a clinical phenotype description into an executable OMOP cohort definition (also known as phenotype development) is an iterative process that requires reconciling clinical terminology with OMOP vocabularies, encoding temporal and occurrence logic in tools such as CAPR or the Atlas UI, and validating the resulting definition empirically. This process can take days to weeks per phenotype, demands both clinical and informatics expertise, and is prone to inconsistency due to variation in human interpretation of logic and clinical terms. Existing automated approaches address narrow sub-problems (e.g. concept set construction, rule extraction), but few integrate concept mapping, executable code synthesis, and empirical validation in a single auditable pipeline. In this work, we describe PhenotypeRefiner, a multi-agent large language model (LLM) workflow that automates each of these steps through purpose-built system prompts, structured concept catalog construction, and a closed-loop diagnostic refinement cycle.

Methods

PhenotypeRefiner coordinates two LLM calls via the ellmer R package against a Microsoft Azure OpenAI endpoint (model o3), with definitions in this work executed against a real-world OMOP CDM claims database (Merative™ MarketScan® Commercial Claims and Encounters Database, N > 100M patients). Figure 1 diagrams the inputs, outputs, and process elements described in detail in the subsections below.

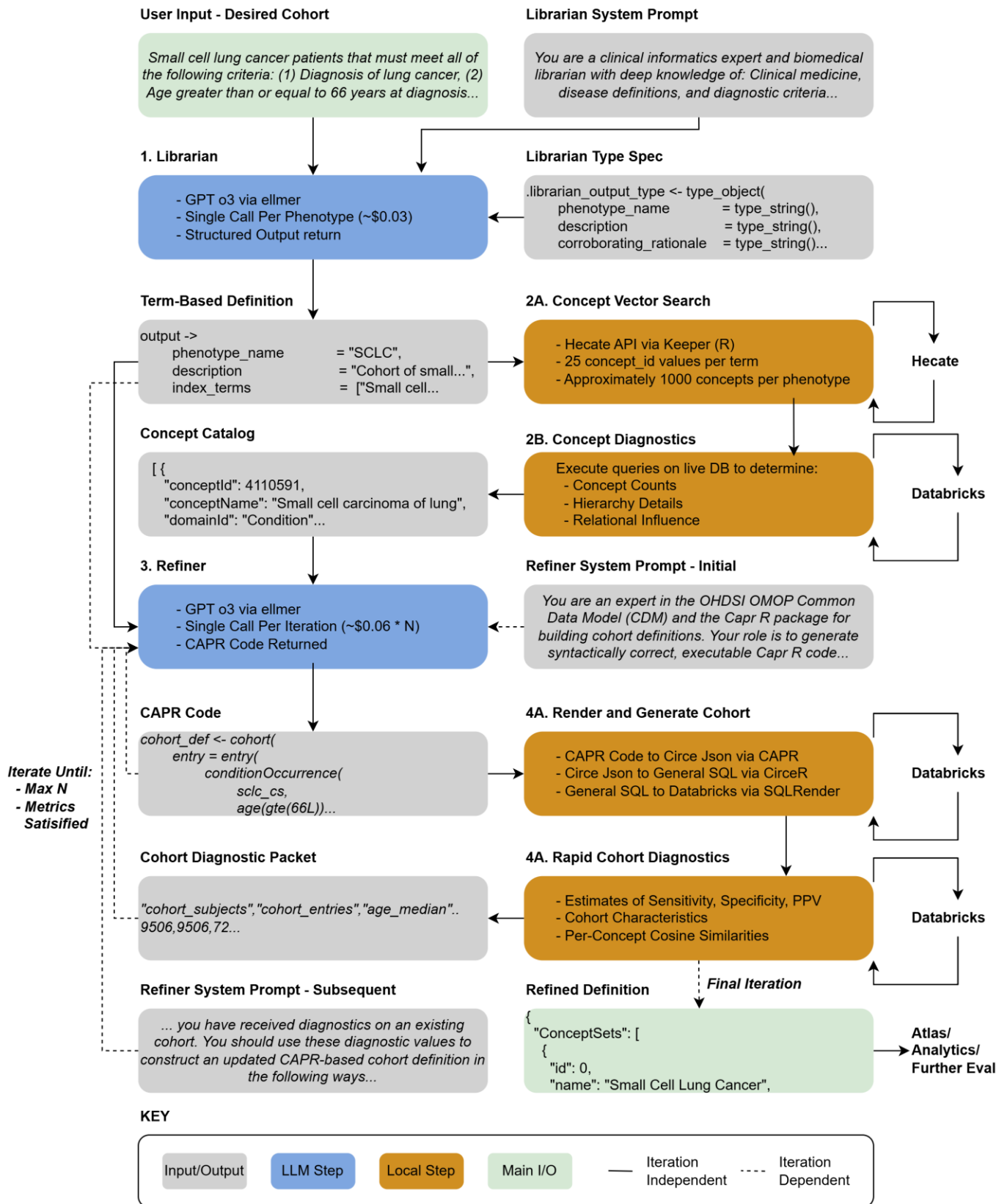


Figure 1. Overview of main steps and associated input and output that make up the PhenotypeRefiner workflow.

Librarian. The Librarian Agent receives a free-text clinical description (i.e. user prompt) about a phenotype of interest and is guided by a detailed system prompt to produce a structured phenotype specification. The prompt instructs the agent to: (1) identify an index event domain (condition, drug, procedure, visit), clinical terms of interest, and incident framing; (2) specify inclusion and exclusion criteria with clinical rationale, temporal windows, and occurrence thresholds (e.g. n or more observations of a given event) ; (3) define demographic constraints and observation window requirements; (4) nominate conflicting concepts that are mutually exclusive with the target phenotype; and (5) identify corroborating clinical evidence (drugs, measurements) expected in true cases. Critically, the prompt instructs the agent to return clinical terms rather than concept IDs -- all term-to-concept resolution is deferred to a downstream vector search step, which then queries a local OMOP concept index and returns standard concept IDs, names, domains, and record counts. This design separates clinical knowledge (LLM) from vocabulary resolution (deterministic search), reducing hallucinated concept IDs and enabling the catalog to reflect only concepts present in the target CDM vocabulary.

Concept Search and Diagnostics. For each clinical term returned by the Librarian, a semantic vector search retrieves up to 25 candidate standard concepts per term. Each candidate is then evaluated against the CDM with a lightweight diagnostic query that returns the number of persons with at least one record and the total (and descendant) record count for that concept. The resulting catalog -- concept ID, name, domain, vocabulary, standard status, descendant count, CDM person count, and CDM record count -- is passed to the Refining Agent as structured context. Concepts with more than 100 unique descendants are flagged as at-risk for over-inclusion, and the catalog entry notes this explicitly. A separate catalog JSON file is written alongside each workflow output for auditability.

Refiner. The Refining Agent receives the Librarian specification and the concept catalog and is guided by a second detailed system prompt to produce R code using the CAPR package (Cohort API in R). The prompt encodes: CAPR syntax rules for concept set construction (`cs()`, `descendants()`, `exclude()`); entry event, inclusion rule, and cohort exit patterns; temporal window and occurrence logic with worked examples; date adjustment patterns for index events derived from related codes; and an explicit priority hierarchy for refinement (broaden concept sets before adding rules; add rules before removing them). The generated CAPR code is compiled to Circe-format JSON and then parameterized SQL via CirceR, rendered and translated to executable SQL via SqlRender, and executed via CohortGenerator with cohort checksum deduplication to avoid redundant executions.

Cohort Generation and Diagnostics. After each cohort generation, three proxy metrics are computed against the CDM: (1) PPV proxy -- proportion of cohort members with cross-domain corroborating evidence (e.g., disease-specific drug or measurement) within a clinically defined window; (2) Sensitivity proxy -- cohort size relative to a CDM denominator estimated from corroborating concept prevalence; (3) Specificity proxy -- composite of clean observation window rate, post-index confirmation rate, and negative-control-free rate (see codebase for additional details). A suite of additional diagnostic queries characterizes the cohort: attrition cascade, index code variety, temporal consistency, and comorbidity profile. The Refining Agent receives all diagnostic output together with the prior CAPR code, the original Librarian specification, and a change registry (description and delta percentage per iteration) and generates a revised definition. The system prompt instructs the agent to prioritize concept set changes and occurrence threshold adjustments before considering structural rule addition or removal, and to treat contradictory concepts as hard exclusions only when they are mutually exclusive with the target condition. The loop runs for a configurable number of iterations; cohort convergence can also be assessed against user-defined thresholds for each metric.

Applications

PhenotypeRefiner is currently in active use in two primary contexts:

Literature-to-executable phenotype conversion. Clinical study teams frequently need to implement phenotypes that are described in peer-reviewed literature or clinical guidelines but have never been expressed as OMOP cohort definitions. PhenotypeRefiner accepts the relevant text -- a methods section excerpt, a table of inclusion/exclusion criteria, relevant code sets (if provided), or a protocol description -- as input to the Librarian Agent and produces an executable OMOP definition within minutes. This has been applied to phenotypes spanning rare diseases, oncology indications, and complex multi-criterion disease definitions, where manual concept identification and logic encoding would otherwise require multiple rounds of expert review.

Research request phenotyping. For prospective research requests (pharmacoepidemiology studies, drug safety signals, comparative effectiveness analyses), PhenotypeRefiner provides a structured, clinically grounded starting point for cohort development. The Librarian Agent's structured output documents the clinical rationale for each criterion and the associated concept set, providing a transparent audit trail. The

concept catalog step ensures that all concepts used in the definition are empirically present in the target CDM before the first cohort is generated. Note that excluding 0-count concepts tailors the definition to a particular database; this feature can be switched off to improve generalizability.

PhenotypeRefiner has meaningfully reduced the time from protocol to executable cohort definition for studies requiring detailed and domain-specific phenotypes. A typical refinement process with three to five iterations generates a working definition in under 10 minutes with an average total LLM cost of about \$0.30, across input descriptions that range from incident drug users with diagnosis anchors, complex psychiatric conditions, and rare autoimmune disease definitions.

Conclusion

PhenotypeRefiner demonstrates that a structured multi-agent approach – separating clinical knowledge elicitation (Librarian), vocabulary grounding (concept catalog), code synthesis (Refining Agent), and empirical feedback (diagnostic loop) into distinct, prompt-governed steps -- can substantially reduce the time and expertise required to produce executable, auditable OMOP phenotype definitions. The pipeline is open-source, database-agnostic, and extensible to additional clinical domains and data environments. Further analysis is required to evaluate the validity of the phenotypes produced by the pipeline.

References

1. Hripcsak G, Duke JD, Shah NH, et al. Observational Health Data Sciences and Informatics (OHDSI): Opportunities for observational researchers. *Stud Health Technol Inform.* 2015;216:574-578.
2. Kohler S, Bard J, Oellrich A, et al. The Human Phenotype Ontology in 2017. *Nucleic Acids Res.* 2017;45(D1):D865-D876.