Supplemental Materials for "On Using Large Language Models to Enhance Clinically-Driven Missing Data Recovery Algorithms in Electronic Health Records"

Sarah C. Lotspeich, Abbey N. Collins, Brian J. Wells, Ashish K. Khanna, Joseph Rigdon, Lucy D'Agostino McGowan

R CODE FOR THE LARGE LANGUAGE MODELS (LLM)

We implemented this *LLM-based roadmap enhancement* using the `ellmer` package in `R`, which enables programmatic interaction with LLM through "tool-calling."[1] That is, the `ellmer` package allows LLM to request execution of `R` functions as external tools. Specifically, we used Gemini-2.5-Flash[2] to test two LLM enhancements.

First, we provided the roadmap's structure as a dataframe "tool" that the LLM could access and modify when prompted. Specifically, we created a function that takes search terms for the ALI components (as vectors) and combines them following the clinical roadmap's structure (a 10×2 dataframe with a rows per component and a column for search terms). Next, we defined a tool that built on this function by adding a description of the tool's purpose,

> "Create a dataframe with text ICD description search terms separated by ; for ICD descriptions to match diagnoses when missing from chart review,"

and arguments for each component of the roadmap framework, like

> "Search terms for ICD text descriptions to detect diagnoses that would suggest that a patient's creatinine clearance is at an unhealthy (low) level, separated by a semicolon."

Then, through the `ellmer` package's tool functionality, we registered this tool (i.e., gave Gemini access to use it) and prompted the LLM to generate relevant terms for each ALI component. Essentially, we requested updates to this dataframe from the LLM, which `ellmer` executed within our `R` session based on either Prompt 1 or 2 from the main text. In this way, we obtained a programmatically-updated roadmap object that could be immediately used in our missing data recovery algorithm pipeline. The follow `R` code implements both of our LLM-based roadmap enhancements as described.

```
library(ellmer)
library(dplyr)

roadmap <- readr::read_csv(here::here("data-raw/audit_roadmap.csv"))

## No context (no examples) -----

c <- chat_google_gemini()

make_data <- function(
  creat_c,
  alb,
  bmi,
  sbp,
  dbp,
  a1c,
  chol,
  trig,
  crp,
  hcst,
```

```
  df_name
) {
  df <- data.frame(
    Variable_Name = c(
      "CREAT_C",
      "ALB",
      "BMI",
      "BP_SYSTOLIC",
      "BP_DIASTOLIC",
      "A1C",
      "CHOL",
      "TRIG",
      "CRP",
      "HCST"
    ),
    If_Missing_Search_For = c(
      creat_c,
      alb,
      bmi,
      sbp,
      dbp,
      a1c,
      chol,
      trig,
      crp,
      hcst
    )
  )
  assign(df_name, df, envir = .GlobalEnv)
}

tool_data <- tool(
  make_data,
  description = "Create a dataframe with text ICD description search terms separated by ;
  for ICD descriptions to match diagnoses when missing from chart review",
  arguments = list(
    creat_c = type_string(
      "Search terms for ICD text descriptions to detect diagnoses that would suggest
      that a patient's creatinine clearance is at an unhealthy (low) level, separated
      by a semicolon."
    ),
    alb = type_string(
      "Search terms for ICD text descriptions to detect diagnoses that would suggest that
      a patient's serum albumin is at an unhealthy (high) level, separated by a semicolon."
    ),
    bmi = type_string(
      "Search terms for ICD text descriptions to detect diagnoses that would suggest that
      a patient's body mass index (BMI) is at an unhealthy (high) level, separated by a
      semicolon."
    ),
    sbp = type_string(
      "Search terms for ICD text descriptions to detect diagnoses that would suggest that
      a patient's systolic blood pressure is at an unhealthy (high) level, separated by a
      semicolon."
    ),
```

```
    dbp = type_string(
      "Search terms for ICD text descriptions to detect diagnoses that would suggest that
      a patient's diastolic blood pressure is at an unhealthy (high) level, separated by a
      semicolon."
    ),
    a1c = type_string(
      "Search terms for ICD text descriptions to detect diagnoses that would suggest that
      a patient's hemoglobin A1c (HbA1c) is at an unhealthy (high) level, separated by a
      semicolon."
    ),
    chol = type_string(
      "Search terms for ICD text descriptions to detect diagnoses that would suggest that
      a patient's total cholesterol is at an unhealthy (high) level, separated by a
      semicolon."
    ),
    trig = type_string(
      "Search terms for ICD text descriptions to detect diagnoses that would suggest that
      a patient's trigylcerides is at an unhealthy (high) level, separated by a semicolon."
    ),
    crp = type_string(
      "Search terms for ICD text descriptions to detect diagnoses that would suggest that
      a patient's C-reactive protein is at an unhealthy (high) level, separated by a
      semicolon."
    ),
    hcst = type_string(
      "Search terms for ICD text descriptions to detect diagnoses that would suggest that
      a patient's homocysteine is at an unhealthy (high) level, separated by a semicolon."
    ),
    df_name = type_string("Name of the dataframe")
  )
)

c$register_tool(tool_data)
c$chat(
  "Please propose an exhaustive list of terms (avoiding acronyms) that will be used to
  search ICD descriptions to identify each of the missing biomarkers and create a data
  frame with these codes. I want you to repeat this process 20 times, creating a new data
  frame each time with each having a unique name starting with `df_nocontext`. Each time
  you repeat this, be sure to make as exhaustive a list as possible. These lists can vary."
)


## With context (examples) -----

c_context <- chat_google_gemini()

tool_data <- tool(
  make_data,
  description = "Create a dataframe with text ICD description search terms separated by ;
  for ICD descriptions to match diagnoses when missing from chart review",
  arguments = list(
    creat_c = type_string("Search terms for ICD text descriptions to detect diagnoses that
    would suggest that a patient's creatinine clearance is at an unhealthy (low) level
    (e.g., renal failure, renal insufficiency, acute kidney injury, and chronic renal
    failure), separated by a semicolon."),
```

```
    alb = type_string("Search terms for ICD text descriptions to detect diagnoses that
    would suggest that a patient's serum albumin is at an unhealthy (high) level,
    separated by a semicolon."),
    bmi = type_string("Search terms for ICD text descriptions to detect diagnoses that
    would suggest that a patient's body mass index (BMI) is at an unhealthy (high) level
    (e.g., Obesity, morbid obesity, Grade I obesity, Grade II obesity, Grade III obesity),
    separated by a semicolon."),
    sbp = type_string("Search terms for ICD text descriptions to detect diagnoses that
    would suggest that a patient's systolic blood pressure is at an unhealthy (high) level
    (e.g., hypertension), separated by a semicolon."),
    dbp = type_string("Search terms for ICD text descriptions to detect diagnoses that
    would suggest that a patient's diastolic blood pressure is at an unhealthy (high)
    level (e.g., hypertension), separated by a semicolon."),
    a1c = type_string("Search terms for ICD text descriptions to detect diagnoses that
    would suggest that a patient's hemoglobin A1c (HbA1c) is at an unhealthy (high) level
    (e.g., diabetes, impaired glycemic control), separated by a semicolon."),
    chol = type_string("Search terms for ICD text descriptions to detect diagnoses that
    would suggest that a patient's total cholesterol is at an unhealthy (high) level
    (e.g., hypercholesterolemia), separated by a semicolon."),
    trig = type_string("Search terms for ICD text descriptions to detect diagnoses that
    would suggest that a patient's trigylcerides is at an unhealthy (high) level (e.g.,
    hypertriglyceridemia), separated by a semicolon."),
    crp = type_string("Search terms for ICD text descriptions to detect diagnoses that
    would suggest that a patient's C-reactive protein is at an unhealthy (high) level
    (e.g., sepsis, infection, autoimmune inflammatory syndrome), separated by a
    semicolon."),
    hcst = type_string("Search terms for ICD text descriptions to detect diagnoses that
    would suggest that a patient's homocysteine is at an unhealthy (high) level (e.g.,
    hyperhomocysteinemia, vitamin deficiency), separated by a semicolon."),
    df_name = type_string("Name of the dataframe")
  )
)

c_context$register_tool(tool_data)
c_context$chat(
  "Please propose an exhaustive list of terms (avoiding acronyms) that will be used to
  search ICD descriptions to identify each of the missing biomarkers and create a data
  frame with these codes. I want you to repeat this process 20 times, creating a new data
  frame each time with each having a unique name starting with `df_context`. Each time you
  repeat this, be sure to include the examples given in (e.g.,) and make as exhaustive a
  list as possible. These lists can vary."
)
```

ADDITIONAL FIGURES

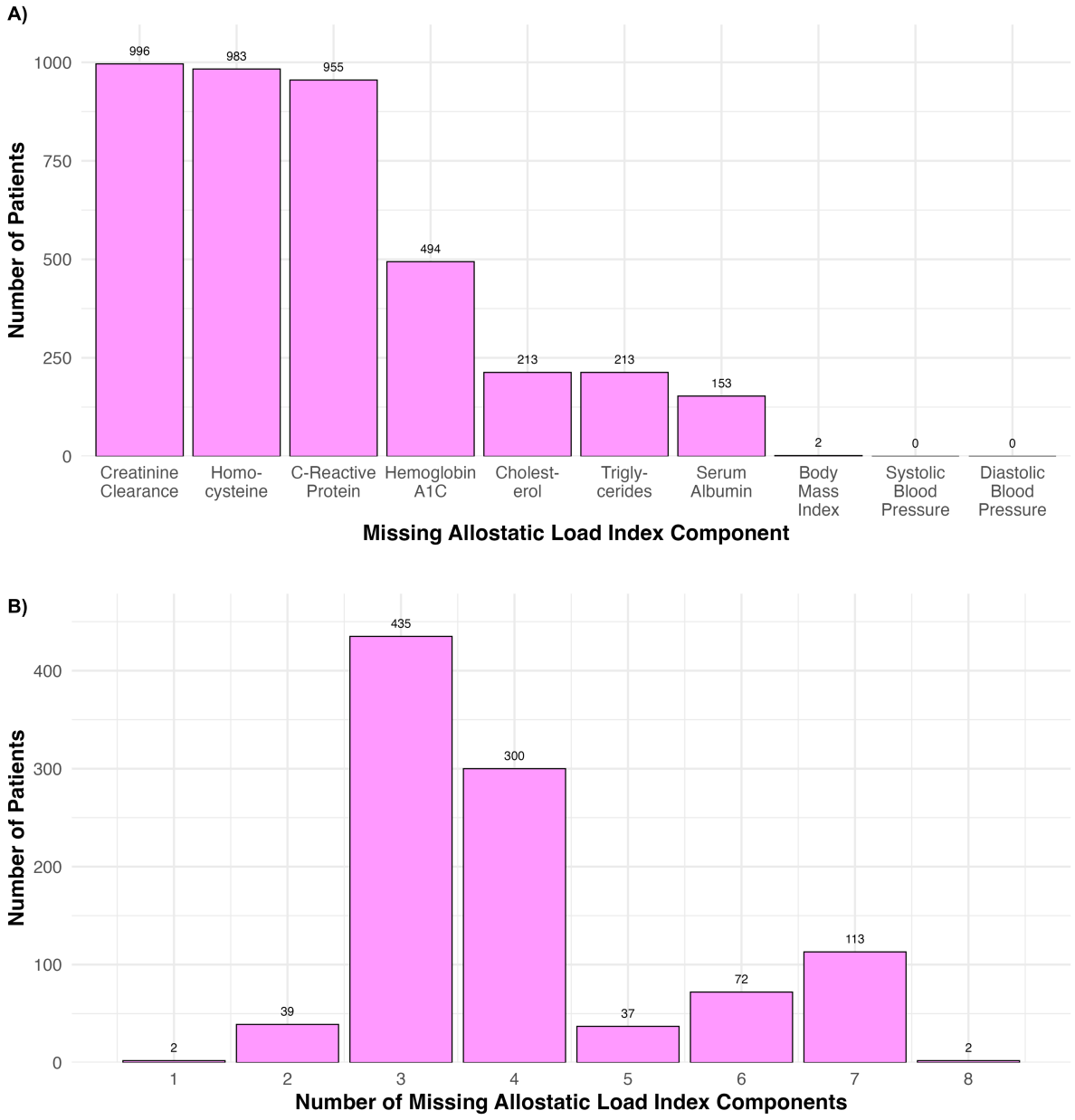

Figure S1. Counts of missing allostatic load index components **A)** per component and **B)** per patient from the original extracted electronic health records (EHR) data ($N = 1000$ patients).

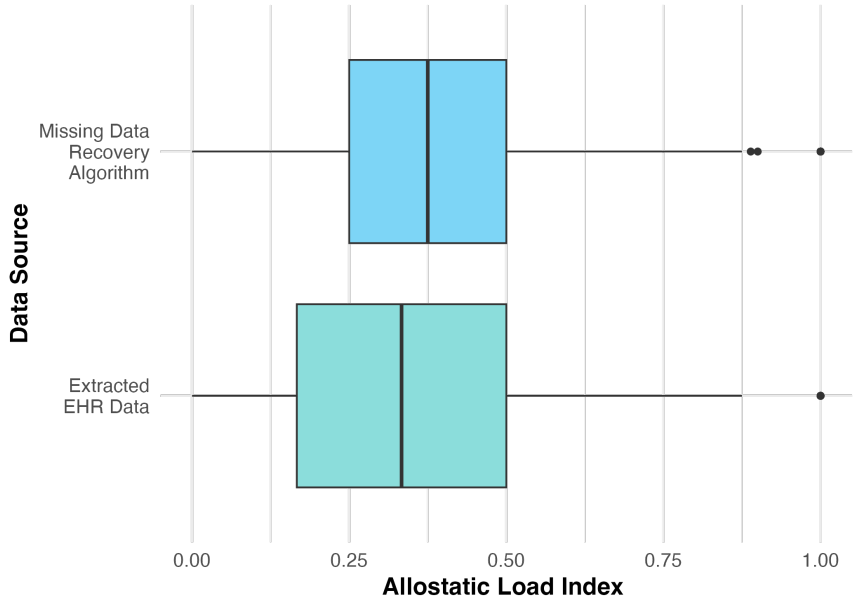

Figure S2. The median allostatic load index (ALI) for the 1000-patient study was slightly higher after applying the missing data recovery algorithm (0.375 versus 0.333 in the extracted EHR data). The distribution of the ALI after recovery was also slightly more symmetric and less variable (IQR $= [0.25, 0.50]$ versus $[0.17, 0.50]$).

REFERENCES

1. Wickham H, Cheng J, Jacobs A, Aden-Buie G, Schloerke B. ellmer: Chat with Large Language Models; 2025. R package version 0.3.0. Available from: `https://ellmer.tidyverse.org`.

2. Gemini Team Google. Gemini: A Family of Highly Capable Multimodal Models. arXiv preprint arXiv:231211805. 2023.