Thermal Liquid Biopsy SQL Database Creation

Avery Bell, Dr. Robert Buscaglia
Department of Mathematics and Statistics

NAU Northern Arizona University — College of the Environment, Forestry, and Natural Sciences
NAU Northern Arizona University — Department of Mathematics and Statistics

Abstract

Thermal liquid biopsy (TLB) is a growing biochemistry field that holds potential to revolutionize the way diseases are detected and diagnosed. TLB results are derived from the thermal denaturation of human blood plasma and provide a characteristic signature termed a thermogram. Thermograms present differently depending on the health status of a patient. There has been over 10 years of active TLB research, and in that time, datasets for lung cancer, melanoma, and a mixture of autoimmune diseases have been compiled. This data consists of thermogram results, and related patient meta information for each thermogram sample. Work has been done to clean, document, and store this TLB data in an R package, and the next step for the organization of this data is to centralize it in a Structured Query Language (SQL) database. This project designed and implemented a relational database for the storage of this TLB information. This project created an entity relation (ER) diagram based on the existing TLB data, translated that ER-diagram into a relational model, and then implemented that model in SQL to build a database. This project culminated in a functioning SQL database for dissemination to TLB investigators.

Introduction

What is TLB?

Clinical Thermal Liquid Biopsy (TLB) is a biophysical technique the involves the denaturation of human blood plasma. It provides measurements of excess specific heat capacity (C_p^{ex}), that are the result of thermal denaturation of the plasma proteome. Measuring the specific excess heat capacity across a range of temperatures results in a signature termed a thermogram (Figure 1)[1]. Thermograms are of interest because the curve characteristics are different depending on the health status of the patient that provided the sample.

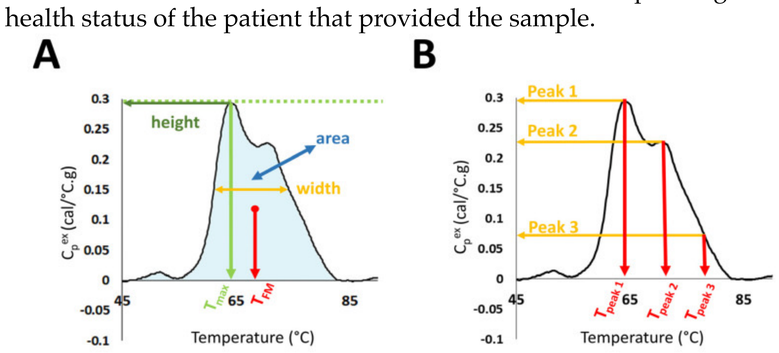


Figure 1. Panel A: Diagram of thermogram characteristics including the maximum height, total area under the curve, full width at half height, maximum peak height, and temperature at peak maximum. Panel B: Position of thermogram peaks 1, 2, 3.

What is a SQL Database?

A database is a collection of related information stored on a computer. SQL is a programming language that is used to create and manage a database. SQL databases store data in tables that can be selected from and merged to obtain desired information.

TLB Data

A single TLB dataset consists of 2 parts. The excess specific heat capacity measurements of a blood plasma sample, and meta information that describes the patient who provided the sample. This meta information includes age, gender, medication, race, and disease specific attributes. This project created a database for 3 TLB datasets, lung cancer, melanoma, and a mixture of autoimmune diseases. These datasets were previously stored in an R environment, and were transferred to SQL. Along with creating a functional database, it also established the blueprint to store other TLB datasets in the SQL database.

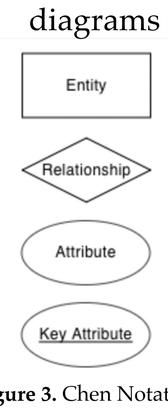
Patient_id	Sample_id	Race	Location	Smoker
LUN123	LUN123_1	White	Lung	Current
LUN456	LUN456_1	Black	Lung, Bone	Ex

Sample_id	T40	T40.1
LUN123_1	0.001	0.00143
LUN456_1	0.0001	0.00152

Figure 2. Example of how TLB datasets are stored in R and Excel files. One file holds patient meta information, and another holds the TLB results.

Database Terminology

ER Diagram: flow chart that defines components of a database, and how they fit together. ER diagrams can be translated into SQL tables.



On object or thing that exists in the real world.

Describes how entities fit together.

Characteristics that entities possess.

Unique characteristic that can identify an entity.

Figure 3. Chen Notation representation for components of an ER diagram. Words inside of shapes are replaced with what they represent. Entities, attributes, key attributes, and relationships are used to build the conceptual framework for relational SQL models.

1. Schneider G, Kaliappan A, Nguyen TQ, **Buscaglia R**, Brock GN, Hall MB, DeSpirito C, Wilkey DW, Merchant ML, Klein JB, Wiese TA, Rivas-Perez HL, Kloecker GH, Garbett NC. The Utility of Differential Scanning Calorimetry Curves of Blood Plasma for Diagnosis, Subtype Differentiation and Predicted Survival in Lung Cancer. Cancers (Basel). 2021 Oct 23;13(21):5326. doi: 10.3390/cancers13215326. PMID: 34771491; PMCID: PMC8582427.

ER Diagram

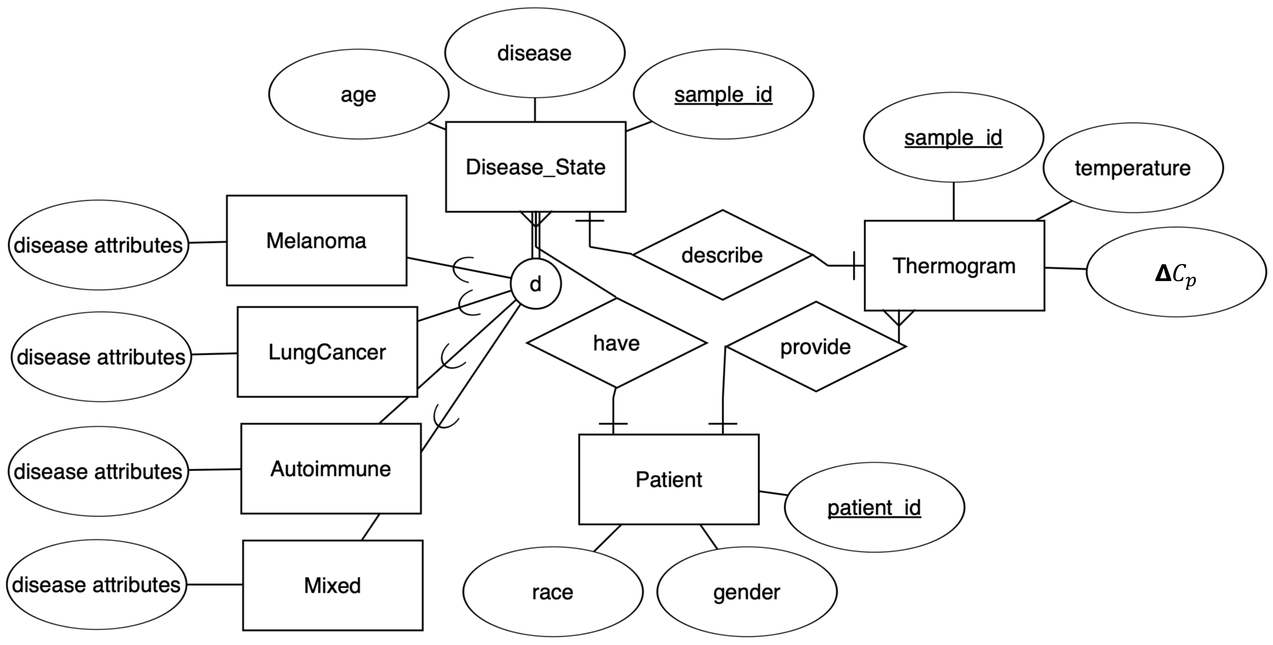


Figure 4. ER Diagram for Thermogram database. Entities are thermograms, patients, and disease states for each disease present in the dataset. The patient information was stored separately from the disease states because patient attributes such as gender and race do not change across samples, whereas diseases can progress and change. Patients with multiple thermogram samples, have multiple disease states, and separating the patient attributes from disease state attributes prevents storing redundant information, creating a memory efficient design for this database.

Database Table Structure

Lung Cancer

Patient			Disease State			Thermogram		
Gender	Race		Location	Smoker		Sample_id	T40	T40.1
Male	White		Lung	Current		LUN123_1	0.001	0.00143
Female	Black		Lung, Bone	Ex		LUN456_1	0.0001	0.00152

Melanoma

Gender	Race		Treatment	Tissues		Sample_id	T40	T40.1
Male	White		Ipilimumab	2		MEL123_1	0.00152	0.00091
Female	Black		GMCSF	1		MEL456_1	0.00109	0.001523

Autoimmune

			Disease	Code		Sample_id	T40	T40.1
			ALS	ALS		AI123_1	0.00149	0.00091
			Lupus	LUP		AI456_1	0.00117	0.001496

All **Patient** information condensed into a single table →

Gender	Race
Male	White
Female	Black
Male	White
Female	Black

Separate **Disease State** tables for each dataset →

Treatment	Tissues
Ipilimumab	2
GMCSF	1

Location	Smoker
Lung	Current
Lung, Bone	Ex

Disease	Code
ALS	ALS
Lupus	LUP

All **Thermogram** information condensed to a single table →

Sample_id	Temperature	ΔC_p
LUN123_1	40	0.001
LUN123_1	40.1	0.00143
MEL123_1	40.	0.00152
MEL123_1	40.1	0.001523
AI123_1	40	0.00149
AI123_1	40.1	0.001496

Figure 5. The structural changes between the TLB data stored in R and the SQL database. Patient and Thermogram tables were condensed, and individual Disease State tables were defined for each dataset. Not all columns present in each table are shown.

Database Population

The tables in each database were populated one row at a time. The function **generate_sql_to_insert_row** (Figure 6) was called for each row of an R data frame, and returned insertion code to add that row to a SQL table. Figure 7 demonstrates how the algorithm populates tables.

1. Connect to the Thermogram database
2. Generate insertion code for each row of the R data frame
3. Repeat until all rows of a table have been pushed
4. Disconnect from database

```
generate_sql_to_insert_row <- function(colnames, inputrow, tablename)
  {

      # get valid data indices
      valid_cols <- which(!is.na(inputrow))

      # get valid data
      valid_input_data <- inputrow[valid_cols]

      valid_input_data_types <- lapply(valid_input_data, typeof)

      # get corresponding column names
      valid_col_names <- paste0(colnames[valid_cols], collapse = ",")

      # build sql statement
      insert <- sprintf("insert into %s (%s)", tablename, valid_col_names)

      # wrap strs in quotes
      input_data <- insert_quotes(valid_input_data_types, valid_input_data)

      value <- sprintf(" values (%s)", paste0(input_data, collapse = ",") )

      return(str_c(insert, value))

  }
```

Figure 6. Image of code that was used to generate a SQL insertion statement for each row in an R data frame.

R Data Frame

Patient_id	Gender	Race
LUN123	Male	White
LUN456	Female	Black
MEL789	Female	White

SQL Table

Patient_id	Gender	Race

Patient_id	Gender	Race
LUN123	Male	White
LUN456	Female	Black
MEL789	Female	White

Patient_id	Gender	Race
LUN123	Male	White

Patient_id	Gender	Race
LUN123	Male	White
LUN456	Female	Black
MEL789	Female	White

Patient_id	Gender	Race
LUN123	Male	White
LUN456	Female	Black

Patient_id	Gender	Race
LUN123	Male	White
LUN456	Female	Black
MEL789	Female	White

Patient_id	Gender	Race
LUN123	Male	White
LUN456	Female	Black
MEL789	Female	White

Figure 7. Example of the algorithm used to populate SQL database tables. Insertion code was generated for each row of an R data frame, and executed in SQL.

GitHub

GitHub is a code management software that can be easily accessed and managed. It holds all the data files, and scripts that created and populated the SQL database. It was used to manage data file, code scripts, and version history.

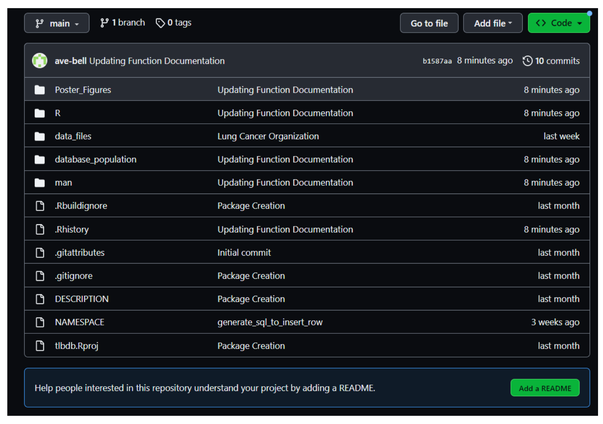


Figure 8. Image of private GitHub Repository that stores the database.

Final Product

The result of this project is a SQL database that uses 5 tables to store 3 thermogram datasets. Figure 9 lists the tables found in the database, and a summary of the database content

Tables in Database	Case/Study	Total Patients	Total Thermograms
lungcancer	Lung Cancer	226	396
melanoma	Melanoma	156	156
mixed	Mixed	921	921
patient			
thermogram			

Figure 9. Summary of existing tables in thermogram database, and counts of patients and thermograms included in the database. Figure represents the finished thermogram SQL database.

Acknowledgements

This is a Data Science Capstone project, completed as part of the Department of Mathematics and Statistics. Thank you to the Office of Undergraduate Research and Creative Activities. Data was generously made available by Dr. Nichola C. Garbett, Associate Professor of Medicine, University of Louisville.